Микросхема интегральная 1888ТХ018

Руководство по эксплуатации
ЮФКВ.431268.017РЭ

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	Справ.№
34992-2	13.05.2019	34992-1			

Содержание

1 Описание и работа микросхемы 4
1.1 Назначение микросхемы....4
1.2 Электрические, динамические и конструктивные характеристики микросхемы.19
1.2.1 Состав и расположение внешних выводов микросхемы....19
1.2.2 Конструктивные характеристики 28
1.2.3 Электрические характеристики микросхемы 29
1.3 Устройство и работа микросхемы 32
1.3.1 Структурная схема и шинная организация микросхемы 32
1.3.2 Карты памяти микросхемы 36
1.3.3 Центральный процессорный узел CPU 40
1.3.4 Узел цифровой обработки сигналов....65
1.3.5 Подсистема видео....95
1.3.6 Подсистема кодирования и декодирования....214
1.3.7 Комплект СФ-блока аудиоконтроллера....546
1.3.8 Подсистема контроллера внешней динамической памяти DDR3 579
1.3.9 Подсистема высокоскоростных интерфейсов HSIF 651
1.3.10 Подсистема суперскоростных интерфейсов XHSIF0 и XHSIF1 686
1.3.11 Подсистема низкоскоростных интерфейсов LSIF0 911
1.3.12 Подсистема низкоскоростных интерфейсов LSIF1 993
1.3.13 Подсистема общего управления микросхемы....1449
1.3.14 Подсистема аппаратного тестирования микросхемы....1516
1.4 Маркировка и пломбирование 1524
1.5 Упаковка....1525
2 Использование по назначению микросхемы 1526
2.1 Процедура начальной инициализации 1526
3 Хранение 1531
4 Транспортирование 1532
5 Утилизация....1533
Приложение А. Детальная структурная схема ЦПУ подсистемы....1534
Приложение Б. Ядро процессора PowerPC 470S....1535
Приложение В. Описание регистра граничного сканирования 1648

Удостоверен ЮФКВ.431268.017РЭ-УД

Микросхема интегральная высокопроизводительная, энергоэффективная, мультимедийная (сокращенное название – СБИС МИВЭМ, далее по тексту - микросхема), обеспечивает выполнение функций центрального вычислителя, обработку, распознавание, кодирование, декодирование и выдачу видеосигнала.

Микросхема предназначена для использования в качестве центрального процессора для устройств, требующих высокой производительности и высокой энергоэффективности в вычислительных системах и системах обработки мультимедийной информации, а также в таких устройствах, как блоки преобразования телевизионных сигналов для бортовой аппаратуры и бортовые управляющие машины.

В состав микросхемы входят процессорные ядра PowerPC с кэш памятью 2-го уровня; процессорные ядра NMC3 с кэш памятью 2-го уровня, мультистандартный кодер/декодер видеосигналов, цифровые видео и аудио интерфейсы, интерфейсы с внешней памятью типа DDR3, SRAM и NAND, широкий набор интерфейсных узлов различного типа и производительности.

Документ содержит описание архитектуры и структуры микросхемы, карты памяти для всех активных устройств, входящих в состав микросхемы, и подробное описание всех ее основных подсистем.

1 Описание и работа микросхемы

1.1 Назначение микросхемы

Микросхема предназначена для применения в носимых и стационарных системах обработки мультимедийной информации.

Объектами, для которых предназначено изделие, являются мультимедийные системы гражданского и двойного назначения. Рассматривается несколько классов объектов, для которых возможно применение изделия:

- стационарное устройство приема и отображения мультимедийных сигналов – для приложений типа «видео стена», стационарных терминалов;
- стационарное устройство приема и передачи видеоданных - для организации систем видеоконференций;
- стационарное устройство записи и обработки видеосигналов – для приложений видеокамеры с функцией распознавания изображений, видеорегистраторов;
- носимое мультимедийное устройство – для приложений типа «планшет».

В состав микросхемы входят следующие основные функциональные узлы:

- Два 32-разрядных суперскалярных процессорных ядра с архитектурой PowerPC 470S, каждое из которых содержит управляющий RISC-процессор, 64-разрядный сопроцессор с плавающей точкой, кэш-память команд первого уровня – 32 КБ, кэш-память данных первого уровня объемом 32 КБ, кэш-память второго уровня - 512 КБ. Система команд версии – PowerISA v2.05. Процессорные ядра PowerPC предназначены для общего управления системой и вторичной обработки сигналов.
- Четыре 32-разрядных процессорных ядра NMC3 с архитектурой NeuroMatrix, каждое из которых содержит управляющий RISC-процессор, 64-разрядный векторно-матричный сопроцессор, статическую память объемом 128 КБ, контроллер ПДП, контроллер прерываний, таймеры. Каждая пара ядер NMC3 имеет общую кэш-память второго уровня объемом 128 КБ. Процессорные ядра NMC3 работают на частоте 400 МГц и служат для цифровой обработки аудио-, видео- и других сигналов, поступающих в систему по различным интерфейсам.
- Банк внутренней статической памяти объемом 256 КБ.
- Контроллер ПДП с функциями шифрации.
- Аппаратный кодер/декодер видеосигнала, обеспечивающий кодирование и декодирование видеоданных для форматов потоков видео стандартной четкости по стандартам ISO/IEC 13818-2 (профиль MP/ML) и ISO/IEC 14496-10 (ITU-T H.264) (MP/L3.0), а так же для форматов потоков видео высокой четкости по стандартам ISO/IEC 13818-2 (профиль MP/HL) и ISO/IEC 14496-10 (ITU-T H.264) (MP/L4.0).
- Два 24-разрядных интерфейса цифрового видео выхода (VOUT0, VOUT1), обеспечивающих вывод видео стандартной и высокой четкости.
- Два 24-разрядных интерфейса цифрового видео входа (VIN0, VIN1), обеспечивающих захват видеоизображения от источника видеосигнала стандартной и высокой четкости.
- 4-разрядный интерфейс цифрового аудио выхода (AUD).
- Два 32-разрядных интерфейса с внешней памятью типа DDR3 (EM0, EM1), каждый с возможностью подключения до 8 ГБ внешней памяти, пропускной способностью до 6,4 ГБ/с и байтовым кодом коррекции ошибок (ECC).
- 32-разрядный интерфейс с внешней памятью типа SRAM/NOR (EM2) с возможностью подключения до 1,5 ГБ внешней памяти, пропускной способностью до 400 МБ/с и поддержкой ECC. Для эффективной работы с различными

микросхемами памяти, в интерфейсе предусмотрена программная настройка динамических параметров для циклов чтения/записи.

- 8-разрядный асинхронный интерфейс с энергонезависимой внешней памятью типа NAND, поддерживающий набор команд стандарта ONFI 2.3, два алгоритма аппаратной коррекции ошибок (4 ошибки на 512 байт и 24 ошибки на 1024 байт), аппаратную защиту NAND-памяти от записи, работу с микросхемами MLC и SLC флэш-памяти, конфигурируемый размер страницы от 0,5 Кбайт до 128 Кбайт, программную настройку временных диаграмм чтения/записи. Интерфейс содержит внутренний буфер на 2 КБайта и контроллер ПДП с настройкой порядка следования байтов.
- Два интерфейса PCIe2.0x4 (PCI0, PCI1), каждый из которых может программно настраиваться на работу по 1, 2 или 4 высокоскоростным дуплексными линиям. Максимальная пропускная способность одного интерфейса 2 ГБ/с на прием и 2 ГБ/с на передачу.
- Шесть дуплексных последовательных интерфейсов ARINC 818 (HSVI0,…, HSVI5 – High Speed Video Interface). Пропускная способность одного интерфейса – до 1 ГБ/с в каждом направлении. Интерфейсы предназначены для передачи цифрового видео.
- Четыре дуплексных последовательных интерфейса FibreChannel (FC0,…,FC3). Пропускная способность одного интерфейса – до 1 ГБ/с в каждом направлении. Интерфейсы предназначены для подключения к высокоскоростным сетям хранения данных.
- Два дуплексных последовательных интерфейса SpaceWire (HSCB0, HSCB1 - High Speed Control Bus) с аппаратной поддержкой протокола RMAP (Remote memory access protocol) в качестве ведомого устройства. Пропускная способность одного интерфейса – до 400 Мбит/с в каждом направлении. Интерфейсы предназначены для подключения к телекоммуникационной сети бортовых систем.
- Два 8-разрядных дуплексных интерфейса GMII (GBETH0, GBETH1 – Giga Bit Ethernet) с поддержкой последовательного управляющего интерфейса MDIO для конфигурации параметров работы внешней микросхемы физического интерфейса Gigabit Ethernet и поддержкой отладочного EDCL-протокола, позволяющего внешнему отладочному устройству получить доступ к внутреннему адресному пространству микросхемы. Пропускная способность одного интерфейса – до 1 Гбит/с в каждом направлении. Интерфейсы предназначены для работы в высокоскоростных проводных локальных сетях.
- Три 2-разрядных дуплексных интерфейса RMII (ETH0,…, ETH2 –Ethernet) с поддержкой отладочного EDCL-протокола. Пропускная способность одного интерфейса – до 100 Мбит/с в каждом направлении. Интерфейсы предназначены для подключения к устройствам, реализующим физический уровень интерфейсов Fast Ethernet в проводных локальных сетях.
- Два последовательных дуплексных интерфейса МКИО (LSCB0, LSCB1 – Low Speed Control Bus). Пропускная способность одного интерфейса – 1 Мбит/с в каждом направлении. Интерфейсы предназначены для использования в бортовых системах повышенной надежности.
- Три последовательных дуплексных интерфейса UART (UART0,…, UART2), каждый из которых содержит программируемый генератор тактового сигнала для формирования битового интервала различной длины, FIFO-буфера для приема и передачи данных для снижения количества прерываний. Интерфейсы поддерживают автоматическое обнаружение ложного стартового бита и независимое маскирование прерываний от предающего FIFO-буфера, принимающего FIFO-буфера, тайм-аутов при приеме данных, ошибок приема и пр. Пропускная способность одного интерфейса – до 921600 бит/с в каждом

направлении. Интерфейсы предназначены для организации экономичной связи с другими цифровыми устройствами.

- Три последовательных дуплексных интерфейса SPI (SPI0, SPI1, GSPI) с программируемой частотой работы (встроенный делитель синхросигнала) и поддержкой различных форматов обмена данными (Motorola SPI, Texas Instruments synchronous serial frame format, National Semiconductor Microwire frame format). Интерфейс GSPI отличается от двух других наличием контроллера ПДП и линии прерывания от внешнего ведомого устройства. Пропускная способность одного интерфейса – до 50 Мбит/с в каждом направлении. Интерфейсы предназначены для простого и экономичного сопряжения с внешними периферийными устройствами.
- 4-разрядный интерфейс карты памяти SDIO со встроенным контроллером ПДП и пропускной способностью 200 Мбит/с.
- Последовательный полудуплексный интерфейс USB 2.0 для работы с высокоскоростными (High Speed, до 480 Мбит/сек), полноскоростными (Full Speed, до 12 Мбит/сек) и низкоскоростных (Low Speed, до 1 Мбит/сек) устройствами. Поддерживает расширение OTG спецификации USB 2.0, позволяющее в режиме соединения «точка-точка» работать как в качестве хост-контроллера шины USB, так и в качестве периферийного устройства. Поддерживает до 15 конечных устройств для передачи и приема данных и обеспечивает аппаратную диспетчеризацию всех USB-транзакций. Содержит встроенный физический интерфейс (USB-PHY) и контроллер ПДП.
- Три последовательных полудуплексных интерфейса I2C (I2C0,…, I2C2) с пропускной способностью – 100 и 400 Кбит/с. Интерфейсы предназначены для подключения низкоскоростных периферийных устройств.
- 18 байтовых интерфейсов общего назначения (GPIO0, GPIO1, MGPIO0,…, MGPIO15) с программной настройкой каждого бита на режим входа, выхода или прерывания.
- Интерфейс JTAG, поддерживающий подключение внешнего отладочного устройства.

Все высокоскоростные интерфейсы (PCI0, PCI1, HSVI0,…, HSVI5, FC0,…, FC3, HSCB0, HSCB1, GBETH0, GBETH1, ETH0,…, ETH2 и USB) содержат встроенные контроллеры ПДП.

Функциональные выводы микросхемы приведены в таблице 1.1.

С целью уменьшения размеров корпуса и кристалла многие функциональные выводы используются в мультиплексном режиме. Только мультимедийные интерфейсы (AUD, VOUT0, VOUT1, VIN0, VIN1), интерфейсы с внешней памятью DDR3 (EM0, EM1), а также интерфейсы USB, I2C0,…, I2C2, GPIO0 и GPIO1 подключены к внешним выводам напрямую. Остальные интерфейсы – через мультиплексоры, управляемые с помощью программируемых конфигурационных регистров, входящих в состав основных интерфейсных подсистем.

Программная установка конфигурационных регистров должна осуществляться пользователем на этапе начальной загрузки микросхемы в соответствии с применяемой схемой ее включения в систему. При разработке схемы включения необходимо учитывать, что во время системного сброса все выводы, которые могут менять направление передачи информации, устанавливаются в состояние входа.

Микросхема содержит 8 физических PHY-интерфейсов (SERDES0,…, SERDES7), каждый из которых служит для обеспечения работы одной дуплексной линии сверхскоросного интерфейса. К SERDES0,…, SERDES7 через мультиплексоры подключены интерфейсы PCI0, PCI1, HSVI0,…, HSVI5, FC0,…, FC3.

Интерфейс с внешней SRAM-памятью (EM2) совмещен по выводам с интерфейсами NAND, GBETH0, GBETH1, ETH0, ETH1 и MGPIO0,…, MGPIO8.

Интерфейс GBETH1 может быть подключен к другой группе выводов, которую могут использовать также интерфейсы ETH2, MGPIO9, и MGPIO10.

Интерфейс SDIO делит выводы с интерфейсом GSPI и старшими разрядами интерфейса MGPIO15.

Любой из интерфейсов LSCB0, LSCB1, UART0, UART1, UART2, SPI0, SPI1 может быть заменен на соответствующие разряды интерфейсов MGPIO11,…, MGPIO15.

Таблица 1.1 - – Функциональные выводы микросхемы

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
Интерфейс EM0 с памятью типа DDR3				
EM0_CK0, EM0_CKB0, EM0_CK1, EM0_CKB1	4	O	8	Дифференциальные тактовые сигналы памяти
EM0_CKE0, EM0_CKE1	2	O	8	Разрешение тактирования
EM0_A0, …, EM0_A15	16	O	8	Шина адреса
EM0_B0, …, EM0_B2	3	O	8	Выбор банков памяти
EM0_CSB0, EM0_CSB1	2	O	8	Выбор кристалла
EM0_RASB	1	O	8	Строб адреса строки памяти
EM0_CASB	1	O	8	Строб адреса столбца памяти
EM0_WEB	1	O	8	Признак операции записи
EM0_RESETN	1	O	8	Асинхронный сброс микросхем памяти
EM0_A_DM0, EM0_A_DM1, EM0_B_DM0, EM0_B_DM1, EM0_C_DM0	5	O	8	Сигналы маскирования записи байтов данных
EM0_A_DQ0,…,EM0_A_DQ7, EM0_A_DQ8,…,EM0_A_DQ15, EM0_B_DQ0,…,EM0_B_DQ7, EM0_B_DQ8,…,EM0_B_DQ15, EM0_C_DQ0,…,EM0_C_DQ7	40	I/O	8	Шина данных (пять байтов)
EM0_A_DQS0,EM0_A_DQSB0, EM0_A_DQS1,EM0_A_DQSB1, EM0_B_DQS0,EM0_B_DQSB0, EM0_B_DQS1,EM0_B_DQSB1, EM0_C_DQS0,EM0_C_DQSB0	10	I/O	8	Дифференциальные стробы байтов данных
EM0_ODT0, EM0_ODT1	2	O	8	Управление согласующими резисторами памяти
EM0_DDR_BUFF_EN	1	I		Управление выходными буферами интерфейса
Интерфейс EM1 с памятью типа DDR3				
EM1_CK0, EM1_CKB0, EM1_CK1, EM1_CKB1	4	O	8	Дифференциальные тактовые сигналы памяти
EM1_CKE0, EM1_CKE1	2	O	8	Разрешение тактирования
EM1_A0, …, EM1_A15	16	O	8	Шина адреса
EM1_B0, …, EM1_B2	3	O	8	Выбор банков памяти
EM1_CSB0, EM1_CSB1	2	O	8	Выбор кристалла
EM1_RASB	1	O	8	Строб адреса строки памяти
EM1_CASB	1	O	8	Строб адреса столбца памяти
EM1_WEB	1	O	8	Признак операции записи
EM1_RESETN	1	O	8	Асинхронный сброс микросхем памяти
EM1_A_DM0, EM1_A_DM1, EM1_B_DM0, EM1_B_DM1, EM1_C_DM0	5	O	8	Сигналы маскирования записи байтов данных
EM1_A_DQ0,…,EM1_A_DQ7, EM1_A_DQ8,…,EM1_A_DQ15, EM1_B_DQ0,…,EM1_B_DQ7, EM1_B_DQ8,…,EM1_B_DQ15, EM1_C_DQ0,…,EM1_C_DQ7	40	I/O	8	Шина данных (пять байтов)
EM1_A_DQS0,EM1_A_DQSB0, EM1_A_DQS1,EM1_A_DQSB1, EM1_B_DQS0,EM1_B_DQSB0, EM1_B_DQS1,EM1_B_DQSB1, EM1_C_DQS0,EM1_C_DQSB0	10	I/O	8	Дифференциальные стробы байтов данных

Продолжение таблицы 1.2

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
EM1_ODT0, EM1_ODT1	2	O	8	Управление согласующими резисторами памяти
EM1_DDR_BUFF_EN	1	I		Управление выходными буферами интерфейса
Интерфейсы SERDES0				
PCI0_TX0P, PCI0_TX0N / HSVI0_TXP, HSVI0_TXN	2	O O		Дифференциальная линия 0 передачи данных PCIE0 / Дифференциальная линия передачи данных HSVI0
PCI0_RX0P, PCI0_RX0N / HSVI0_RXP, HSVI0_RXN	2	I I		Дифференциальная линия 0 приема данных PCIE0 / Дифференциальная линия приема данных HSVI0
Интерфейсы SERDES1				
PCI0_TX1P, PCI0_TX1N / HSVI1_TXP, HSVI1_TXN	2	O O		Дифференциальная линия 1 передачи данных PCIE0 / Дифференциальная линия передачи данных HSVI1
PCI0_RX1P, PCI0_RX1N / HSVI1_RXP, HSVI1_RXN	2	I I		Дифференциальная линия 1 приема данных PCIE0 / Дифференциальная линия приема данных HSVI1
Интерфейсы SERDES2				
PCI0_TX2P, PCI0_TX2N / HSVI2_TXP, HSVI2_TXN / FC2_TXP, FC2_TXN	2	O O O		Дифференциальная линия 2 передачи данных PCIE0 / Дифференциальная линия передачи данных HSVI2 / Дифференциальная линия передачи данных FC2
PCI0_RX2P, PCI0_RX2N / HSVI2_RXP, HSVI2_RXN / FC2_RXP, FC2_RXN	2	I I I		Дифференциальная линия 2 приема данных PCIE0 / Дифференциальная линия приема данных HSVI2 / Дифференциальная линия приема данных FC2
Интерфейсы SERDES3				
PCI0_TX3P, PCI0_TX3N / FC0_TXP, FC0_TXN	2	O O		Дифференциальная линия 3 передачи данных PCIE0 / Дифференциальная линия передачи данных FC0
PCI0_RX3P, PCI0_RX3N / FC0_RXP, FC0_RXN	2	I I		Дифференциальная линия 3 приема данных PCIE0 / Дифференциальная линия приема данных FC0
Интерфейсы SERDES4				
PCI1_TX0P, PCI1_TX0N / HSVI3_TXP, HSVI3_TXN	2	O O		Дифференциальная линия 0 передачи данных PCIE1 / Дифференциальная линия передачи данных HSVI3
PCI1_RX0P, PCI1_RX0N / HSVI3_RXP, HSVI3_RXN	2	I I		Дифференциальная линия 0 приема данных PCIE1 / Дифференциальная линия приема данных HSVI3
Интерфейсы SERDES5				
PCI1_TX1P, PCI1_TX1N / HSVI4_TXP, HSVI4_TXN	2	O O		Дифференциальная линия 1 передачи данных PCIE1 / Дифференциальная линия передачи данных HSVI4
PCI1_RX1P, PCI1_RX1N / HSVI4_RXP, HSVI4_RXN	2	I I		Дифференциальная линия 1 приема данных PCIE1 / Дифференциальная линия приема данных HSVI4
Интерфейсы SERDES6				
PCI1_TX2P, PCI1_TX2N / HSVI5_TXP, HSVI5_TXN / FC3_TXP, FC3_TXN	2	O O O		Дифференциальная линия 2 передачи данных PCIE1 / Дифференциальная линия передачи данных HSVI5 / Дифференциальная линия передачи данных FC3
PCI1_RX2P, PCI1_RX2N / HSVI5_RXP, HSVI5_RXN / FC3_RXP, FC3_RXN	2	I I I		Дифференциальная линия 2 приема данных PCIE1 / Дифференциальная линия приема данных HSVI5 / Дифференциальная линия приема данных FC3
Интерфейсы SERDES7				
PCI1_TX3P, PCI1_TX3N / FC1_TXP, FC1_TXN	2	O O		Дифференциальная линия 3 передачи данных PCIE1 / Дифференциальная линия передачи данных FC1
PCI1_RX3P, PCI1_RX3N / FC1_RXP, FC1_RXN	2	I I		Дифференциальная линия 3 приема данных PCIE1 / Дифференциальная линия приема данных FC1
Интерфейс HSCB0 (Space Wire 0)				
HSCB0_DIP, HSCB0_DIN	2	I		Дифференциальная линия данных передатчика
HSCB0_SIP, HSCB0_SIN	2	I		Дифференциальная линия стробов передатчика
HSCB0_DOP, HSCB0_DON	2	O		Дифференциальная линия данных приёмника
HSCB0_SOP, HSCB0_SON	2	O		Дифференциальная линия стробов приёмника
Интерфейс HSCB1 (Space Wire 1)				
HSCB1_DIP, HSCB1_DIN	2	I		Дифференциальная линия данных передатчика
HSCB1_SIP, HSCB1_SIN	2	I		Дифференциальная линия стробов передатчика
HSCB1_DOP, HSCB1_DON	2	O		Дифференциальная линия данных приёмника
HSCB1_SOP, HSCB1_SON	2	O		Дифференциальная линия стробов приёмника

Продолжение таблицы 1.3

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
Интерфейс EM2c памятью типа SRAM / Интерфейсы GBETH0, GBETH1 и интерфейс с памятью NAND / Интерфейсы ETH1, ETH0 и интерфейс с памятью NAND / Интерфейсы MGPIO0,…, MGPIO8				
EM2_D0 / GBETH0_ RX_DV / - / MGPIO0_0	1	I/O I I I/O	8 8	0-й разряд шины данных EM2 / Сигнал значимости принимаемых данных GBETH0 / Не используется / 0-й разряд MGPIO0
EM2_D1 / GBETH0_RX_ER / ETH1_RX_ER / MGPIO0_1	1	I/O I I I/O	8 8	1-й разряд шины данных EM2 / Сигнал ошибки принимаемых данных GBETH0 / Сигнал ошибки принимаемых данных ETH1 / 1-й разряд MGPIO0
EM2_D2 / GBETH0_ RX_COL / - / MGPIO0_2	1	I/O I I I/O	8 8	2-й разряд шины данных EM2 / Сигнал обнаружения коллизии GBETH0 / Не используется / 2-й разряд MGPIO0
EM2_D3 / GBETH0_RX_CRS / ETH1_RX_ CRS / MGPIO0_3	1	I/O I I I/O	8 8	3-й разряд шины данных EM2 / Сигнал обнаружения несущей GBETH0 / Сигнал обнаружения несущей ETH1 / 3-й разряд MGPIO0
EM2_D4 / GBETH0_RX_D0 / ETH1_RX_D0 / MGPIO0_4	1	I/O I I I/O	8 8	4-й разряд шины данных EM2 / 0-й разряд шины принимаемых данных GBETH0 / 0-й разряд шины принимаемых данных ETH1 / 4-й разряд MGPIO0
EM2_D5 / GBETH0_RX_D1 / ETH1_RX_ D1 / MGPIO0_5	1	I/O I I I/O	8 8	5-й разряд шины данных EM2 / 1-й разряд шины принимаемых данных GBETH0 / 1-й разряд шины принимаемых данных ETH1 / 5-й разряд MGPIO0
EM2_D6 / GBETH0_RX_D2 / - / MGPIO0_6	1	I/O I I I/O	8 8	6-й разряд шины данных EM2 / 2-й разряд шины принимаемых данных GBETH0 / Не используется / 6-й разряд MGPIO0
EM2_D7 / GBETH0_RX_D3 / - / MGPIO0_7	1	I/O I I I/O	8 8	7-й разряд шины данных EM2 / 3-й разряд шины принимаемых данных GBETH0 / Не используется / 7-й разряд MGPIO0
EM2_D8 / GBETH0_RX_D4 / - / MGPIO1_0	1	I/O I I I/O	8 8	8-й разряд шины данных EM2 / 4-й разряд шины принимаемых данных GBETH0 / Не используется / 0-й разряд MGPIO1
EM2_D9 / GBETH0_RX_D5 / - / MGPIO1_1	1	I/O I I I/O	8 8	9-й разряд шины данных EM2 / 5-й разряд шины принимаемых данных GBETH0 / Не используется / 1-й разряд MGPIO1
EM2_D10 / GBETH0_RX_D6 / - / MGPIO1_2	1	I/O I I I/O	8 8	10-й разряд шины данных EM2 / 6-й разряд шины принимаемых данных GBETH0 / Не используется / 2-й разряд MGPIO1
EM2_D11 / GBETH0_RX_D7 / - / MGPIO1_3	1	I/O I I I/O	8 8	11-й разряд шины данных EM2 / 7-й разряд шины принимаемых данных GBETH0 / Не используется / 3-й разряд MGPIO1
EM2_D12 / GBETH0_TX_CLK / ETH1_CLK / -	1	I/O I I I	8 	12-й разряд шины данных EM2 / Входной синхросигнал передатчика GBETH0 / Синхросигнал ETH1 / Не используется

Продолжение таблицы 1.4

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
EM2_D13 / GBETH0_RX_CLK / - / -	1	I/O I I I	8	13-й разряд шины данных EM2 / Синхросигнал приемника GBETH0 / Не используется / Не используется
EM2_D14 / GBETH1_RX_DV / - / MGPIO1_4	1	I/O I I I/O	8 8	14-й разряд шины данных EM2 / Сигнал значимости принимаемых данных GBETH1 / Не используется / 4-й разряд MGPIO1
EM2_D15 / GBETH1_RX_ER / ETH0_RX_ER / MGPIO1_5	1	I/O I I I/O	8 8	15-й разряд шины данных EM2 / Сигнал ошибки принимаемых данных GBETH1 / Сигнал ошибки принимаемых данных ETH0 / 5-й разряд MGPIO1
EM2_D16 / GBETH1_ RX_COL / - / MGPIO1_6	1	I/O I I I/O	8 8	16-й разряд шины данных EM2 / Сигнал обнаружения коллизии GBETH1 / Не используется / 6-й разряд MGPIO1
EM2_D17 / GBETH1_RX_CRS / ETH0_RX_ CRS / MGPIO1_7	1	I/O I I I/O	8 8	17-й разряд шины данных EM2 / Сигнал обнаружения несущей GBETH1 / Сигнал обнаружения несущей ETH0 / 7-й разряд MGPIO1
EM2_D18 / GBETH1_RX_D0 / ETH0_RX_D0 / MGPIO2_0	1	I/O I I I/O	8 8	18-й разряд шины данных EM2 / 0-й разряд шины принимаемых данных GBETH1 / 0-й разряд шины принимаемых данных ETH0 / 0-й разряд MGPIO2
EM2_D19 / GBETH1_RX_D1 / ETH0_RX_D1 / MGPIO2_1	1	I/O I I I/O	8 8	19-й разряд шины данных EM2 / 1-й разряд шины принимаемых данных GBETH1 / 1-й разряд шины принимаемых данных ETH0 / 1-й разряд MGPIO2
EM2_D20 / GBETH1_RX_D2 / - / MGPIO2_2	1	I/O I I I/O	8 8	20-й разряд шины данных EM2 / 2-й разряд шины принимаемых данных GBETH1 / Не используется / 2-й разряд MGPIO2
EM2_D21 / GBETH1_RX_D3 / - / MGPIO2_3	1	I/O I I I/O	8 8	21-й разряд шины данных EM2 / 3-й разряд шины принимаемых данных GBETH1 / Не используется / 3-й разряд MGPIO2
EM2_D22 / GBETH1_RX_D4 / - / MGPIO2_4	1	I/O I I I/O	8 8	22-й разряд шины данных EM2 / 4-й разряд шины принимаемых данных GBETH1 / Не используется / 4-й разряд MGPIO2
EM2_D23 / GBETH1_RX_D5 / - / MGPIO2_5	1	I/O I I I/O	8 8	23-й разряд шины данных EM2 / 5-й разряд шины принимаемых данных GBETH1 / Не используется / 5-й разряд MGPIO2
EM2_D24 / GBETH1_RX_D6 / - / MGPIO2_6	1	I/O I I I/O	8 8	24-й разряд шины данных EM2 / 6-й разряд шины принимаемых данных GBETH1 / Не используется / 6-й разряд MGPIO2
EM2_D25 / GBETH1_RX_D7 / - / MGPIO2_7	1	I/O I I I/O	8 8	25-й разряд шины данных EM2 / 7-й разряд шины принимаемых данных GBETH1 / Не используется / 7-й разряд MGPIO2
EM2_D26 / GBETH1_TX_CLK / ETH0_CLK / -	1	I/O I I I	8	26-й разряд шины данных EM2 / Входной синхросигнал передатчика GBETH1 / Синхросигнал ETH0 / Не используется

Продолжение таблицы 1.5

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
EM2_D27 / GBETH1_RX_CLK / - / -	1	I/O I I I	8	27-й разряд шины данных EM2 / Синхросигнал приемника GBETH1 / Не используется / Не используется
EM2_D28,…, EM2_D35 / NAND_D0,…, NAND_D7 / NAND_D0,…, NAND_D7 / MGPIO3_0,…, MGPIO3_7	8	I/O I/O I/O I/O	8 8 8 8	28-й,…, 35-й разряды шины данных EM2 / Шина данных NAND / Шина данных NAND / Разряды MGPIO3
EM2_D36 / NAND_READY1n / NAND_READY1n / MGPIO4_0	1	I/O I I I/O	8 8	36-й разряд шины данных EM2 / Сигнал готовности 1-го банка памяти NAND / Сигнал готовности 1-го банка памяти NAND / 0-й разряд MGPIO4
EM2_D37 / NAND_READY2n / NAND_READY2n / MGPIO4_1	1	I/O I I I/O	8 8	37-й разряд шины данных EM2 / Сигнал готовности 2-го банка памяти NAND / Сигнал готовности 2-го банка памяти NAND / 1-й разряд MGPIO4
EM2_D38 / GBETH0_TX_EN / ETH1_TX_EN / MGPIO4_2	1	I/O O O I/O	8 8 8 8	38-й разряд шины данных EM2 / Управление шиной данных передатчика GBETH0 / Управление шиной данных передатчика ETH1 / 2-й разряд MGPIO4
EM2_D39 / GBETH1_TX_EN / ETH0_TX_EN / MGPIO4_3	1	I/O O O I/O	8 8 8 8	39-й разряд шины данных EM2 / Управление шиной данных передатчика GBETH1 / Управление шиной данных передатчика ETH0 / 3-й разряд MGPIO4
EM2_A0 / - / - / MGPIO4_4	1	O I I I/O	8 8	0-й разряд шины адреса EM2 / Не используется / Не используется / 4-й разряд MGPIO4
EM2_A1 / GBETH0_TX_ER / - / MGPIO4_5	1	O O I I/O	8 8 8	1-й разряд шины адреса EM2 / Сигнал ошибки передачи данных GBETH0 / Не используется / 5-й разряд MGPIO4
EM2_A2 / GBETH0_TX_D0 / ETH1_TX_D0 / MGPIO4_6	1	O O O I/O	8 8 8 8	2-й разряд шины адреса EM2 / 0-й разряд шины передаваемых данных GBETH0 / 0-й разряд шины передаваемых данных ETH1 / 6-й разряд MGPIO4
EM2_A3 / GBETH0_TX_D1 / ETH1_TX_D1 / MGPIO4_7	1	O O O I/O	8 8 8 8	3-й разряд шины адреса EM2 / 1-й разряд шины передаваемых данных GBETH0 / 1-й разряд шины передаваемых данных ETH1 / 7-й разряд MGPIO4
EM2_A4 / GBETH0_TX_D2 / - / MGPIO5_0	1	O O I I/O	8 8 8	4-й разряд шины адреса EM2 / 2-й разряд шины передаваемых данных GBETH0 / Не используется / 0-й разряд MGPIO5
EM2_A5 / GBETH0_TX_D3 / - / MGPIO5_1	1	O O I I/O	8 8 8	5-й разряд шины адреса EM2 / 3-й разряд шины передаваемых данных GBETH0 / Не используется / 1-й разряд MGPIO5
EM2_A6 / GBETH0_TX_D4 / - / MGPIO5_2	1	O O I I/O	8 8 8	6-й разряд шины адреса EM2 / 4-й разряд шины передаваемых данных GBETH0 / Не используется / 2-й разряд MGPIO5
EM2_A7 / GBETH0_TX_D5 / - / MGPIO5_3	1	O O I I/O	8 8 8	7-й разряд шины адреса EM2 / 5-й разряд шины передаваемых данных GBETH0 / Не используется / 3-й разряд MGPIO5

Продолжение таблицы 1.6

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
EM2_A8 / GBETH0_TX_D6 / - / MGPIO5_4	1	O O I I/O	8 8 8	8-й разряд шины адреса EM2 / 6-й разряд шины передаваемых данных GBETH0 / Не используется / 4-й разряд MGPIO5
EM2_A9 / GBETH0_TX_D7 / - / MGPIO5_5	1	O O I I/O	8 8 8	9-й разряд шины адреса EM2 / 7-й разряд шины передаваемых данных GBETH0 / Не используется / 5-й разряд MGPIO5
EM2_A10 / GBETH0_GTX_CLK / - / -	1	O O I I	8 8 	10-й разряд шины адреса EM2 / Выходной синхросигнал передатчика GBETH0 / Не используется / Не используется
EM2_A11 / GBETH0_MDC / ETH1_MDC / MGPIO5_6	1	O O O I/O	8 8 8 8	11-й разряд шины адреса EM2 / Синхросигнал линии MDIO для GBETH0 / Синхросигнал линии MDIO для ETH1 / 6-й разряд MGPIO5
EM2_A12 / GBETH0_MDIO / ETH1_MDIO / MGPIO5_7	1	O I/O I/O I/O	8 8 8 8	12-й разряд шины адреса EM2 / Линия MDIO для GBETH0 / Линия MDIO для ETH1 / 7-й разряд MGPIO5
EM2_A13 / - / - / MGPIO6_0	1	O I I I/O	8 8	13-й разряд шины адреса EM2 / Не используется / Не используется / 0-й разряд MGPIO6
EM2_A14 / GBETH1_TX_ER / - / MGPIO6_1	1	O O I I/O	8 8 8	14-й разряд шины адреса EM2 / Сигнал ошибки передачи данных GBETH1 / Не используется / 1-й разряд MGPIO6
EM2_A15 / GBETH1_TX_D0 / ETH0_TX_D0 / MGPIO6_2	1	O O O I/O	8 8 8 8	15-й разряд шины адреса EM2 / 0-й разряд шины передаваемых данных GBETH1 / 0-й разряд шины передаваемых данных ETH0 / 2-й разряд MGPIO6
EM2_A16 / GBETH1_TX_D1 / ETH0_TX_D1 / MGPIO6_3	1	O O O I/O	8 8 8 8	16-й разряд шины адреса EM2 / 1-й разряд шины передаваемых данных GBETH1 / 1-й разряд шины передаваемых данных ETH0 / 3-й разряд MGPIO6
EM2_A17 / GBETH1_TX_D2 / - / MGPIO6_4	1	O O I I/O	8 8 8	17-й разряд шины адреса EM2 / 2-й разряд шины передаваемых данных GBETH1 / Не используется / 4-й разряд MGPIO6
EM2_A18 / GBETH1_TX_D3 / - / MGPIO6_5	1	O O I I/O	8 8 8	18-й разряд шины адреса EM2 / 3-й разряд шины передаваемых данных GBETH1 / Не используется / 5-й разряд MGPIO6
EM2_A19 / GBETH1_TX_D4 / - / MGPIO6_6	1	O O I I/O	8 8 8	19-й разряд шины адреса EM2 / 4-й разряд шины передаваемых данных GBETH1 / Не используется / 6-й разряд MGPIO6
EM2_A20 / GBETH1_TX_D5 / - / MGPIO6_7	1	O O I I/O	8 8 8	20-й разряд шины адреса EM2 / 5-й разряд шины передаваемых данных GBETH1 / Не используется / 7-й разряд MGPIO6
EM2_A21 / GBETH1_TX_D6 / - / MGPIO7_0	1	O O I I/O	8 8 8	21-й разряд шины адреса EM2 / 6-й разряд шины передаваемых данных GBETH1 / Не используется / 0-й разряд MGPIO7

Продолжение таблицы 1.7

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
EM2_A22 / GBETH1_TX_D7 / - / MGPIO7_1	1	O O I I/O	8 8 8	22-й разряд шины адреса EM2 / 7-й разряд шины передаваемых данных GBETH1 / Не используется / 1-й разряд MGPIO7
EM2_A23 / GBETH1_GTX_CLK / - / -	1	O O I I	8 8 	23-й разряд шины адреса EM2 / Выходной синхросигнал передатчика GBETH1 / Не используется / Не используется
EM2_A24 / GBETH1_MDC / ETH0_MDC / MGPIO7_2	1	O O O I/O	8 8 8 8	24-й разряд шины адреса EM2 / Синхросигнал линии MDIO для GBETH1 / Синхросигнал линии MDIO для ETH0 / 2-й разряд MGPIO7
EM2_A25 / GBETH1_MDIO / ETH0_MDIO / MGPIO7_3	1	O I/O I/O I/O	8 8 8 8	25-й разряд шины адреса EM2 / Линия MDIO для GBETH1 / Линия MDIO для ETH0 / 3-й разряд MGPIO7
EM2_CE0 / NAND_REn / NAND_REn / MGPIO7_4	1	O O O I/O	8 8 8 8	Сигнал выбора 0-го банка памяти EM2 / Сигнал разрешения чтения из памяти NAND / Сигнал разрешения чтения из памяти NAND / 4-й разряд MGPIO7
EM2_CE1 / NAND_CLEn / NAND_CLEn / MGPIO7_5	1	O O O I/O	8 8 8 8	Сигнал выбора 1-го банка памяти EM2 / Сигнал защелкивания команды в память NAND / Сигнал защелкивания команды в память NAND / 5-й разряд MGPIO7
EM2_CE2 / NAND_CE2n / NAND_CE2n / MGPIO7_6	1	O O O I/O	8 8 8 8	Сигнал выбора 2-го банка памяти EM2 / Сигнал выбора 2-го банка памяти NAND / Сигнал выбора 2-го банка памяти NAND / 6-й разряд MGPIO7
EM2_CE3 / NAND_CE1n / NAND_CE1n / MGPIO7_7	1	O O O I/O	8 8 8 8	Сигнал выбора 3-го банка памяти EM2 / Сигнал выбора 1-го банка памяти NAND / Сигнал выбора 1-го банка памяти NAND / 7-й разряд MGPIO7
EM2_CE4 / NAND_ALE / NAND_ALE / MGPIO8_0	1	O O O I/O	8 8 8 8	Сигнал выбора 4-го банка памяти EM2 / Сигнал защелкивания адреса в память NAND / Сигнал защелкивания адреса в память NAND / 0-й разряд MGPIO8
EM2_CE5 / NAND_WEn / NAND_WEn / MGPIO8_1	1	O O O I/O	8 8 8 8	Сигнал выбора 5-го банка памяти EM2 / Сигнал разрешения записи в память NAND / Сигнал разрешения записи в память NAND / 1-й разряд MGPIO8
EM2_BHE_H / - / - / MGPIO8_2	1	O I I I/O	8 8	Сигнал выбора 3-го байта данных EM2 / Не используется / Не используется / 2-й разряд MGPIO8
EM2_BHE_L / - / - / MGPIO8_3	1	O I I I/O	8 8	Сигнал выбора 2-го байта данных EM2 / Не используется / Не используется / 3-й разряд MGPIO8
EM2_BLE_H / - / - / MGPIO8_4	1	O I I I/O	8 8	Сигнал выбора 1-го байта данных EM2 / Не используется / Не используется / 4-й разряд MGPIO8
EM2_BLE_L / - / - / MGPIO8_5	1	O I I I/O	8 8	Сигнал выбора 0-го байта данных EM2 / Не используется / Не используется / 5-й разряд MGPIO8

Продолжение таблицы 1.8

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
EM2_OE / - / - / MGPIO8_6	1	O I I I/O	8 8	Сигнал разрешения выдачи данных из памяти EM2 / Не используется / Не используется / 6-й разряд MGPIO8
EM2_WE / NAND_WPn / NAND_WPn / MGPIO8_7	1	O O O I/O	8 8 8 8	Сигнал разрешения записи данных в память EM2 / Сигнал защиты от записи в память NAND / Сигнал защиты от записи в память NAND / 7-й разряд MGPIO8
Интерфейс GBETH1 / Интерфейс ETH2 / Интерфейсы MGPIO9, MGPIO10				
GBETH1_RX_CLK / ETH2_CLK / -	1	I I I		Синхросигнал приемника GBETH1 / Синхросигнал ETH2 / Не используется
GBETH1_ RX_COL / - / MGPIO9_0	1	I I I/O	 8	Сигнал обнаружения коллизии GBETH1 / Не используется / 0-й разряд MGPIO9
GBETH1_RX_CRS / ETH2_RX_ CRS / MGPIO9_1	1	I I I/O	 8	Сигнал обнаружения несущей GBETH1 / Сигнал обнаружения несущей ETH2 / 1-й разряд MGPIO9
GBETH1_RX_D0 / ETH2_RX_D0 / MGPIO9_2	1	I I I/O	 8	0-й разряд шины принимаемых данных GBETH1 / 0-й разряд шины принимаемых данных ETH2 / 2-й разряд MGPIO9
GBETH1_RX_D1 / ETH2_RX_D1 / MGPIO9_3	1	I I I/O	 8	1-й разряд шины принимаемых данных GBETH1 / 1-й разряд шины принимаемых данных ETH2 / 3-й разряд MGPIO9
GBETH1_RX_D2 / - / MGPIO9_4	1	I I I/O	 8	2-й разряд шины принимаемых данных GBETH1 / Не используется / 4-й разряд MGPIO9
GBETH1_RX_D3 / - / MGPIO9_5	1	I I I/O	 8	3-й разряд шины принимаемых данных GBETH1 / Не используется / 5-й разряд MGPIO9
GBETH1_RX_ER / ETH2_RX_ER / MGPIO9_6	1	I I I/O	 8	Сигнал ошибки принимаемых данных GBETH1 / Сигнал ошибки принимаемых данных ETH2 / 6-й разряд MGPIO9
GBETH1_RX_DV / - / MGPIO9_7	1	I I I/O	 8	Сигнал значимости принимаемых данных GBETH1 / Не используется / 7-й разряд MGPIO9
GBETH1_TX_CLK / - / -	1	I I I		Входной синхросигнал передатчика GBETH1 / Не используется / Не используется
GBETH1_TX_EN / ETH2_TX_EN / MGPIO10_0	1	O O I/O	8 8 8	Управление шиной данных передатчика GBETH1 / Управление шиной данных передатчика ETH2 / 0-й разряд MGPIO10
GBETH1_TX_ER / - / MGPIO10_1	1	O I I/O	8 8	Сигнал ошибки передачи данных GBETH1 / Не используется / 1-й разряд MGPIO10
BETH1_TX_D0 / ETH2_TX_D0 / MGPIO10_2	1	O O I/O	8 8 8	0-й разряд шины передаваемых данных GBETH1 / 0-й разряд шины передаваемых данных ETH2 / 2-й разряд MGPIO10
GBETH1_TX_D1 / ETH2_TX_D1 / MGPIO10_3	1	O O I/O	8 8 8	1-й разряд шины передаваемых данных GBETH1 / 1-й разряд шины передаваемых данных ETH2 / 3-й разряд MGPIO10
GBETH1_TX_D2 / - / MGPIO10_4	1	O I I/O	8 8	2-й разряд шины передаваемых данных GBETH1 / Не используется / 4-й разряд MGPIO10

Продолжение таблицы 1.9

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
GBETH1_TX_D3 / - / MGPIO10_5	1	O I I/O	8 8	3-й разряд шины передаваемых данных GBETH1 / Не используется / 5-й разряд MGPIO10
GBETH1_MDC / ETH2_MDC / MGPIO10_6	1	O O I/O	8 8 8	Синхросигнал линии MDIO для GBETH1 / Синхросигнал линии MDIO для ETH2 / 6-й разряд MGPIO10
GBETH1_MDIO / ETH2_MDIO / MGPIO10_7	1	I/O I/O I/O	8 8 8	Линия MDIO для GBETH1 / Линия MDIO для ETH2 / 7-й разряд MGPIO10
Интерфейс LSCB0 / Интерфейс MGPIO11 и младшие разряды интерфейса MGPIO13				
LSCB0_TXA, LSCB0_nTXA / MGPIO11_0, MGPIO11_1	2	O I/O	4 4	Парафазная линия данных “A” передатчика LSCB0 / 0-й и 1-й разряды MGPIO11
LSCB0_RXA, LSCB0_nRXA MGPIO11_2, MGPIO11_3	2	I I/O	 4	Парафазная линия данных “A” приёмника LSCB0 / 2-й и 3-й разряды MGPIO11
LSCB0_TXB, LSCB0_nTXB MGPIO11_4, MGPIO11_5	2	O	4	Парафазная линия данных “B” передатчика LSCB0 / 4-й и 5-й разряды MGPIO11
LSCB0_RXB, LSCB0_nRXB / MGPIO11_6, MGPIO11_7	2	I I/O	 4	Парафазная линия данных “B” приемника LSCB0 / 6-й и 7-й разряды MGPIO11
LSCB0_CLK_IN / MGPIO13_0	1	I I/O	 4	Синхросигнал контроллера LSCB0 / 0-й разряд MGPIO13
LSCB0_TX_INHA MGPIO13_1	1	O I/O	4 4	Сигнал запроса прерывания “A” из LSCB0 / 1-й разряд MGPIO13
LSCB0_TX_INHB MGPIO13_2	1	O I/O	4 4	Сигнал запроса прерывания “B” из LSCB0 / 2-й разряд MGPIO13
Интерфейс LSCB1 / Интерфейс MGPIO12 и средние разряды интерфейса MGPIO13				
LSCB1_TXA, LSCB1_nTXA / MGPIO12_0, MGPIO12_1	2	O I/O	4 4	Парафазная линия данных “A” передатчика LSCB1 / 0-й и 1-й разряды MGPIO12
LSCB1_RXA, LSCB1_nRXA MGPIO12_2, MGPIO12_3	2	I I/O	 4	Парафазная линия данных “A” приёмника LSCB1 / 2-й и 3-й разряды MGPIO12
LSCB1_TXB, LSCB1_nTXB MGPIO12_4, MGPIO12_5	2	O	4	Парафазная линия данных “B” передатчика LSCB1 / 4-й и 5-й разряды MGPIO12
LSCB1_RXB, LSCB1_nRXB / MGPIO12_6, MGPIO12_7	2	I I/O	 4	Парафазная линия данных “B” приемника LSCB1 / 6-й и 7-й разряды MGPIO12
LSCB1_CLK_IN / MGPIO13_3	1	I I/O	 4	Синхросигнал контроллера LSCB1 / 3-й разряд MGPIO13
LSCB1_TX_INHA MGPIO13_4	1	O I/O	4 4	Сигнал запроса прерывания “A” из LSCB1 / 4-й разряд MGPIO13
LSCB1_TX_INHB MGPIO13_5	1	O I/O	4 4	Сигнал запроса прерывания “B” из LSCB1 / 5-й разряд MGPIO13
Интерфейс UART0 / Старшие разряды интерфейса MGPIO13				
UART0_RXD / MGPIO13_6	1	I I/O	 4	Вход принимаемых данных UART0 6-й разряд MGPIO13
UART0_TXD / MGPIO13_6	1	O I/O	4 4	Выход передаваемых данных UART0 7-й разряд MGPIO13
Интерфейс UART1 / Младшие разряды интерфейса MGPIO14				
UART1_RXD / MGPIO14_0	1	I I/O	 4	Вход принимаемых данных UART1 0-й разряд MGPIO14
UART1_TXD / MGPIO14_1	1	O I/O	4 4	Выход передаваемых данных UART1 1-й разряд MGPIO14
Интерфейс UART2 / Средние разряды интерфейса MGPIO14				
UART2_RXD / MGPIO14_2	1	I I/O	 4	Вход принимаемых данных UART2 2-й разряд MGPIO14

Продолжение таблицы 1.10

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
UART2_TXD / MGPIO14_3	1	O I/O	4 4	Выход передаваемых данных UART2 3-й разряд MGPIO14
Интерфейс SPI0 / Старшие разряды интерфейса MGPIO14				
SPI0_SCLK / MGPIO14_4	1	O I/O	4 4	Тактовый сигнал SPI0 / 4-й разряд MGPIO14
SPI0_MOSI / MGPIO14_5	1	O I/O	4 4	Выход данных SPI0 / 5-й разряд MGPIO14
SPI0_MISO / MGPIO14_6	1	I I/O	 4	Вход данных SPI0 / 6-й разряд MGPIO14
Интерфейс SPI1 / Младшие разряды интерфейса MGPIO15				
SPI1_SCLK / MGPIO15_0	1	O I/O	4 4	Тактовый сигнал SPI1 / 0-й разряд MGPIO15
SPI1_MOSI / MGPIO15_1	1	O I/O	4 4	Выход данных SPI1 / 1-й разряд MGPIO15
SPI1_MISO / MGPIO15_2	1	I I/O	 4	Вход данных SPI1 / 2-й разряд MGPIO15
Интерфейс SDIO / Интерфейс GSPI / Старшие разряды интерфейса MGPIO15				
SDIO_CLK / GSPI_SCLK / -	1	O O I	8 8 	Тактовый сигнал SDIO / Тактовый сигнал GSPI / Не используется
SDIO_CMD GSPI_MOSI / MGPIO15_3	1	I/O O I/O	8 8 8	Линия команд SDIO / Выход данных GSPI / 3-й разряд MGPIO15
SDIO_D0 / GSPI_MISO / MGPIO15_4	1	I/O I I/O	8 8	0-й разряд шины данных SDIO / Вход данных GSPI / 4-й разряд MGPIO15
SDIO_D1 / GSPI_INT / MGPIO15_5	1	I/O I I/O	8 8	1-й разряд шины данных SDIO / Вход прерывания GSPI / 5-й разряд MGPIO15
SDIO_D2 / - / MGPIO15_6	1	I/O I I/O	8 8	2-й разряд шины данных SDIO / Не используется / 6-й разряд MGPIO15
SDIO_D3 / GSPI_CS / MGPIO15_7	1	I/O O I/O	8 8 8	3-й разряд шины данных SDIO / Выборка кристалла GSPI / 7-й разряд MGPIO15
Интерфейс USB				
USB0PP, USB0PN	2	I/O		Дифференциальная линия данных
VBUS	1	I		Определение установки соединения
USB0ID	1	I		Определение конца кабеля
USB_CLK	1	I		Опорный синхросигнал
Интерфейс I2C0				
I2C0_SDA	1	I/O		Линия данных
I2C0_SCL	1	O		Линия синхронизации
Интерфейс I2C1				
I2C1_SDA	1	I/O		Линия данных
I2C1_SCL	1	O		Линия синхронизации
Интерфейс I2C2				
I2C2_SDA	1	I/O		Линия данных
I2C2_SCL	1	O		Линия синхронизации
Интерфейс GPIO0				
GPIO0_0,…, GPIO0_7	8	I/O	4	Выводы общего назначения
Интерфейс GPIO1				
GPIO1_0,…, GPIO1_7	8	I/O	4	Выводы общего назначения

Продолжение таблицы 1.11

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение
Интерфейс цифрового аудио выхода AUD				
AUD_SCLK	1	O	4	Синхросигнал последовательной выдачи отсчетов
AUD_WS	1	O	4	Признак принадлежности передаваемых отсчетов к левому или правому каналу
AUD_D0, …, AUD_D3	4	O	4	Шина последовательной передачи отсчетов многоканального звука
SPDIF	1	O	4	Выход аудиосигнала по стандартам IEC 60958 и IEC 61937
Интерфейс цифрового видео выхода VOUT0				
VOUT0_CLKO	1	O	4	Тактовый сигнал
VOUT0_DE	1	O	4	Сигнал подтверждения передачи видимой части кадра
VOUT0_HSYNC	1	O	4	Сигнал синхронизации строк
VOUT0_VSYNC	1	O	4	Сигнал синхронизации кадров
VOUT0_FIELD	1	O	4	Выбор поля кадра для чересстрочной развертки
VOUT0_D0, …, VOUT0_D23	24	O	4	Шина видеоданных
Интерфейс цифрового видео выхода VOUT1				
VOUT1_CLKO	1	O	4	Тактовый сигнал
VOUT1_DE	1	O	4	Сигнал подтверждения передачи видимой части кадра
VOUT1_HSYNC	1	O	4	Сигнал синхронизации строк
VOUT1_VSYNC	1	O	4	Сигнал синхронизации кадров
VOUT1_FIELD	1	O	4	Выбор поля кадра для чересстрочной развертки
VOUT1_D0, …, VOUT1_D23	24	O	4	Шина видеоданных
Интерфейс цифрового видео входа VIN0				
VIN0_CLKI	1	I		Тактовый сигнал
VIN0_DE	1	I		Сигнал подтверждения передачи видимой части кадра
VIN0_HSYNC	1	I		Сигнал синхронизации строк
VIN0_VSYNC	1	I		Сигнал синхронизации кадров
VIN0_FIELD	1	I		Выбор поля кадра для чересстрочной развертки
VIN0_D0, …, VIN0_D23	24	I		Шина видеоданных
Интерфейс цифрового видео входа VIN1				
VIN1_CLKI	1	I		Тактовый сигнал
VIN1_DE	1	I		Сигнал подтверждения передачи видимой части кадра
VIN1_HSYNC	1	I		Сигнал синхронизации строк
VIN1_VSYNC	1	I		Сигнал синхронизации кадров
VIN1_FIELD	1	I		Выбор поля кадра для чересстрочной развертки
VIN1_D0, …, VIN1_D23	24	I		Шина видеоданных
Интерфейс JTAG				
TCK	1	I		Тактовый сигнал
TMS	1	I		Выбор режима тестирования
TDI	1	I		Вход данных
TDO	1	O	4	Выход данных
TRST	1	I		Сброс тестового порта
DEBUG_HALTn	1	I		Режим останова процессора
Общее управление				
TMODE	1	I		Тестовый режим микросхемы
SYS_RST	1	I		Системный сброс
TIMER	1	O	4	Вывод системного таймера
MCHK_IN	1	I		Вход системной ошибки
MCHK_OUT	1	O	4	Вывод системной ошибки
Опорные синхросигналы				
CLK_CPU	1	I		Синхросигнал PLL_CPU
CLK_DDR	1	I		Синхросигнал PLL_DDR
CLK_AUD	1	I		Синхросигнал PLL_AUDIO
CLK_VDU	1	I		Синхросигнал PLL_VDU
SD0_REFCLK	1	O	4	Опорный синхросигнал PCIE
SD1_REFCLK	1	O	4	Опорный синхросигнал ARINC и FC

Примечания:

1) Используемые обозначения типа вывода:

I – вход;

O – выход,

O(Z) – выход с высокоимпедансным состоянием,

I/O – двунаправленный вывод.

2) Мощность каждого КМОП выходного буфера может быть программно задана при начальной загрузке микросхемы путем записи константы в соответствующий регистр системного контроллера. Однако, программируемое значение тока выходного буфера не должно превышать значения, указанного в таблице 1.1. В противном случае, работоспособность микросхемы не гарантируется. При системном сбросе для всех выходов устанавливается значение выходного тока в 4 мА.

1.2 Электрические, динамические и конструктивные характеристики микросхемы

1.2.1 Состав и расположение внешних выводов микросхемы

Микросхема изготавливается в 1024-выводном корпусе HFCBGA (High Performance Flip Chip Ball Grid Array). Расположение внешних выводов микросхемы представлено на рисунке 2.1.

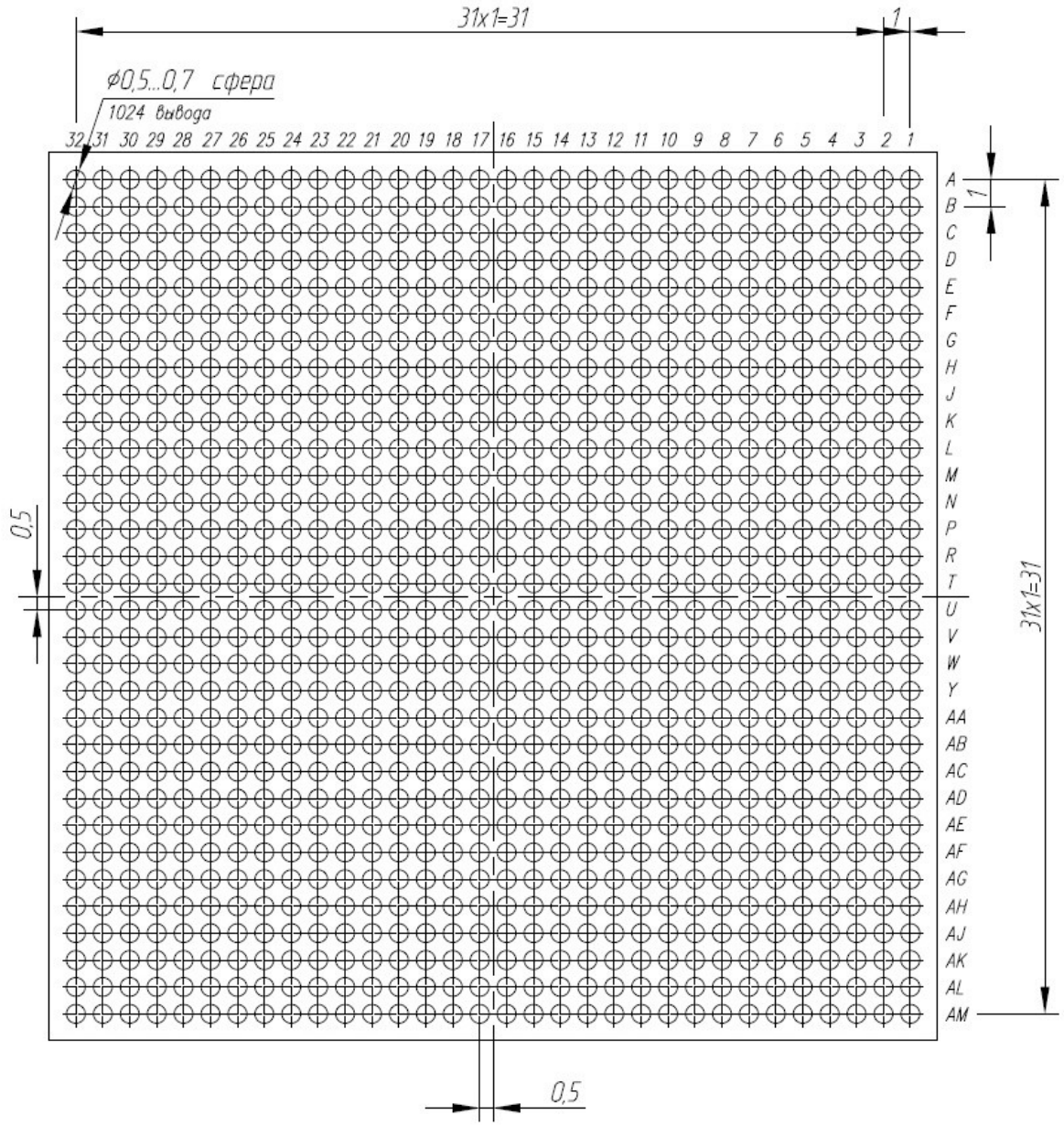


Рисунок 1.1 – Расположение внешних выводов микросхемы

(вид со стороны выводов)

В Таблица 1.12 представлены выводы микросхемы, отсортированные по их функциональному назначению.

Таблица 1.12 - Выводы микросхемы, отсортированные по функциональному назначению

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
Интерфейс с внешней динамической памятью типа DDR3 (EM0)					
EM0_CK0, EM0_CKB0, EM0_CK1, EM0_CKB1	4	O	8	Дифференциальный синхросигнал	AB4, AC4, AB6, AC6
EM0_CKE0,EM0_CKE1	2	O	8	Разрешение тактирования	AJ4, AM7
EM0_A0, …, EM0_A15	16	O	8	Шина адреса	AE5, AL6, AD7, AE6, AJ5,AG6, AK4, AF6, AH7, AC7, AH5, AM5, AL5, AC5, AK6, AJ6
EM0_B0, …, EM0_B2	3	O	8	Выбор банков памяти	AL4, AL7, AE7
EM0_CSB0, EM0_CSB1	2	O	8	Выбор кристалла	AF4, AJ7
EM0_RASB	1	O	8	Строб адреса строки	AE4
EM0_CASB	1	O	8	Строб адреса столбца	AG7
EM0_WEB	1	O	8	Разрешение записи	AG4
EM0_RESETN	1	O	8	Асинхронный сброс микросхем памяти	AM2
EM0_A_DM0,EM0_A_DM1,EM0_B_DM0,EM0_B_DM1,EM0_C_DM0	5	O	8	Сигналы маскирования записи данных	M3, V3, F3, AD3, AK3
EM0_A_DQ0, …, EM0_A_DQ15, EM0_B_DQ0, …, EM0_B_DQ15, EM0_C_DQ0, …, EM0_C_DQ7	40	I/O	8	Шина данных	H1, H2, J1, J2, L1, L2, M1, M2, P1, P2, R1, R2, U1, U2, V1, V2, B1, B2, C1, C2, E1, E2, F1, F2, Y1, Y2, AA1, AA2, AC1, AC2, AD1, AD2, AF2, AK2, AF1, AK1, AJ1, AG1, AG2, AJ2
EM0_A_DQS0, EM0_A_DQSB0, EM0_A_DQS1, EM0_A_DQSB1, EM0_B_DQS0, EM0_B_DQSB0, EM0_B_DQS1, EM0_B_DQSB1, EM0_C_DQS0, EM0_C_DQSB0	10	I/O	8	Дифференциальные стробы данных	J3, K3, R3, T3, C3, D3, AA3, AB3, AG3, AH3
EM0_ODT0, EM0_ODT1	2	O	8	Управление согласующими резисторами	AG5, AD5
EM0_DDR_BUFF_EN	1	I		Управление выходными буферами	AM1
Интерфейс с внешней динамической памятью типа DDR3 (EM1)					
EM1_CK0, EM1_CKB0, EM1_CK1, EM1_CKB1	4	O	8	Дифференциальный синхросигнал	AB29, AC29, AB27, AC27
EM1_CKE0, EM1_CKE1	2	O	8	Разрешение тактирования	AJ29, AM26
EM1_A0, …, EM1_A15	16	O	8	Шина адреса	AE28, AK27, AD26, AE27, AL29, AG27, AJ28, AF29, AH26, AC26, AH28, AL28, AM28, AE26, AJ27, AL27
EM1_B0,…, EM1_B2	3	O	8	Выбор банков памяти	AK29, AL26 AC28
EM1_CSB0, EM1_CSB1	2	O	8	Выбор кристалла	AF27, AJ26
EM1_RASB	1	O	8	Строб адреса строки	AE29
EM1_CASB	1	O	8	Строб адреса столбца	AG26
EM1_WEB	1	O	8	Разрешение записи	AG29
EM1_RESETN	1	O	8	Асинхронный сброс микросхем памяти	AM31
EM1_A_DM0,EM1_A_DM1, EM1_B_DM0,EM1_B_DM1, EM1_C_DM0	5	O	8	Сигналы маскирования записи данных	M30, V30, F30, AD30, AK30

Продолжение таблицы 1.13

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
EM1_A_DQ0, …, EM1_A_DQ15, EM1_B_DQ0, …, EM1_B_DQ15, EM1_C_DQ0, …, EM1_C_DQ7	40	I/O	8	Шина данных	H32,H31,J32,J31,L32,L31,M32, M31,P32,P31,R32,R31,U32,U31, V32,V31,B32,B31,C32,C31,E32, E31,F32,F31,Y32,Y31,AA32, AA31,AC32,AC31,AD32,AD31, AF31,AK31,AF32,AK32,AJ32, AG32,AG31,AJ31
EM1_A_DQS0, EM1_A_DQSB0, EM1_A_DQS1, EM1_A_DQSB1, EM1_B_DQS0, EM1_B_DQSB0, EM1_B_DQS1, EM1_B_DQSB1, EM1_C_DQS0, EM1_C_DQSB0	10	I/O	8	Дифференциальные стробы данных	J30, K30, R30, T30, C30, D30, AA30, AB30, AG30, AH30
EM1_ODT0, EM1_ODT1	2	O	8	Управление согласующими резисторами	AG28, AD28
EM1_DDR_BUFF_EN	1	I		Управление выходными буферами	AM32
Интерфейс внешней статической памяти					
EXT_MEM_D0, …, EXT_MEM_D39	40	I/O	8	Двунаправленная шина данных	AE19,AF19,AA20,AM19,AK18, AD19,AL19,AB20,AK19,AH18, AL18,AG19,AH19,AD18,AC18, AE18,AG18,AJ18,AG17,AA18, AB18,AL17,AH17,AM17,AK17, AA16,AL16,AG16,AF17,AE17, AK16,AH16,AD17,AB17,AD16, AC16,AA17,AB16,AJ16,AE16
EXT_MEM_A0, …, EXT_MEM_A25	26	O	8	Шина адреса	AA23,AB23,AL23,AM23,AG23, AF23,AK23,AH23,AD23,AE23, AL22,AH22,AK22,AJ22,AD22, AK21,AG22,AB22,AA21,AE22, AH21,AB21,AC22,AA22,AL21, AM21
EXT_MEM_WE	1	O	8	Разрешение записи	AC20
EXT_MEM_OE	1	O	8	Разрешение выдачи	AD20
EXT_MEM_BHE_H, EXT_MEM_BHE_L EXT_MEM_BLE_H, EXT_MEM_BLE_L	1	O	8	Выбор байта данных	AE21, AD21, AG21, AF21
EXT_MEM_CE0, …, EXT_MEM_CE5	6	O	8	Выбор микросхемы памяти	AE20, AG20, AL20, AK20, AH20, AJ20
Интерфейс SERDES					

Продолжение таблицы 1.14

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
SD_PHY0_TXP, SD_PHY0_TXN, SD_PHY1_TXP, SD_PHY1_TXN, SD_PHY2_TXP, SD_PHY2_TXN, SD_PHY3_TXP, SD_PHY3_TXN, SD_PHY4_TXP, SD_PHY4_TXN, SD_PHY5_TXP, SD_PHY5_TXN, SD_PHY6_TXP, SD_PHY6_TXN, SD_PHY7_TXP, SD_PHY7_TXN	16	O		Дифференциальные линии передачи данных	A10, B10, A11, B11, A12, B12, A13, B13, A20, B20, A21, B21, A22, B22, A23, B23
SD_PHY0_RXP, SD_PHY0_RXN, SD_PHY1_RXP, SD_PHY1_RXN, SD_PHY2_RXP, SD_PHY2_RXN, SD_PHY3_RXP, SD_PHY3_RXN, SD_PHY4_RXP, SD_PHY4_RXN, SD_PHY5_RXP, SD_PHY5_RXN, SD_PHY6_RXP, SD_PHY6_RXN, SD_PHY7_RXP, SD_PHY7_RXN	16	I		Дифференциальные линии приёма данных	D10, E10, D11, E11, D12, E12, D13, E13, D20, E20, D21, E21, D22, E22, D23, E23
SD_REFCLK0, SD_REFCLK1	2	I		Опорный синхросигнал	L18, L19
Интерфейс LVDS#0					
HSCB0_DIP, HSCB0_DIN	2	I		Дифференциальная линия данных передатчика	A16,B16
HSCB0_SIP, HSCB0_SIN	2	I		Дифференциальная линия стробов передатчика	A17,B17
HSCB0_DOP, HSCB0_DON	2	O		Дифференциальная линия данных приёмника	D16,E16
HSCB0_SOP, HSCB0_SON	2	O		Дифференциальная линия стробов приёмника	D17,E17
Интерфейс LVDS#1					
HSCB1_DIP, HSCB1_DIN	2	I		Дифференциальная линия данных передатчика	A14, B14
HSCB1_SIP, HSCB1_SIN	2	I		Дифференциальная линия стробов передатчика	A15, B15
HSCB1_DOP, HSCB1_DON	2	O		Дифференциальная линия данных приёмника	D14, E14
HSCB1_SOP, HSCB1_SON	2	O		Дифференциальная линия стробов приёмника	D15, E15

Продолжение таблицы 1.15

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
Интерфейс MII					
MII_TX_CLK	1	I		Синхросигнал передатчика	AA9
MII_TXD0, …, MII_TXD3	4	O	8	Шина данных передатчика	AM9, AL9, AJ9, AH9
MII_TX_EN	1	O	8	Управление шиной данных передатчика	AG9
MII_TX_ER	1	O	8	Сигнал ошибки при передаче данных	AE9
MII_RX_CLK	1	I		Синхросигнал приёмника	AF8
MII_RXD0, …, MII_RXD4	4	I		Шина данных приёмника	AM8, AL8, AK8, AH8
MII_RX_DV	1	I		Сигнал значимости принимаемых данных	AD8
MII_RX_ER	1	I		Сигнал ошибки при приёме данных	AG8
MII_COL	1	I		Сигнал обнаружения коллизии	AB8
MII_CRS	1	I		Сигнал обнаружения несущей	AC8
MII_MDC	1	O	8	Синхросигнал интерфейса MDIO	AD9
MII_MDIO	1	I/O	8	Линия данных интерфейса MDIO	AC9
Интерфейс МКИО#0					
LSCB0_CLK_IN0	1	I		Синхросигнал контроллера	W24
LSCB0_TXA0, LSCB0_nTXA0	2	O	4	Парафазная линия данных передатчика	AM24, AL24
LSCB0_RXA0, LSCB0_nRXA0	2	I		Парафазная линия данных приёмника	AD24, AC24
LSCB0_TX_INH_A0	1	O	4	Запрос прерывания	AK24
LSCB0_TXB0, LSCB0_nTXB0	2	O	4	Парафазная дублирующая линия данных передатчика	AH24, AG24
LSCB0_RXB0, LSCB0_nRXB0	2	I		Парафазная дублирующая линия данных приёмника	AB24, AA24
LSCB0_TX_INH_B0	1	O	4	Дублирующий запрос прерывания	AF24
Интерфейс МКИО#1					
LSCB1_CLK_IN1	1	I		Синхросигнал контроллера	W25
LSCB1_TXA1, LSCB1_nTXA1	2	O	4	Парафазная линия данных передатчика	AM25, AL25
LSCB1_RXA1, LSCB1_nRXA1	2	I		Парафазная линия данных приёмника	AD25, AC25
LSCB1_TX_INH_A1	1	O	4	Запрос прерывания	AK25
LSCB1_TXB1, LSCB1_nTXB1	2	O	4	Парафазная дублирующая линия данных передатчика	AH25, AG25
LSCB1_RXB1, LSCB1_nRXB1	2	I		Парафазная дублирующая линия данных приёмника	AB25, AA25
LSCB1_TX_INH_B1	1	I		Синхросигнал контроллера	AF25

Продолжение таблицы 1.16

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
Интерфейс UART#0					
UART0_TXD	1	O	4	Передаваемые данные	AD15
UART0_RXD	1	I		Принимаемые данные	AC15
Интерфейс UART#1					
UART1_TXD	1	O	4	Передаваемые данные	AB15
UART1_RXD	1	I		Принимаемые данные	AA15
Интерфейс UART#2					
UART2_TXD	1	O	4	Передаваемые данные	AM14
UART2_RXD	1	I		Принимаемые данные	AL14
Интерфейс SPI#0					
SPI0_SCLK	1	O	4	Выход синхросигнала	AF13
SPI0_MOSI	1	O	4	Выход данных	AG13
SPI0_MISO	1	I		Вход данных	AH13
Интерфейс SPI#1					
SPI1_SCLK	1	O	4	Выход синхросигнала	AK13
SPI1_MOSI	1	O	4	Выход данных	AL13
SPI1_MISO	1	I		Вход данных	AM13
Интерфейс SDIO					
SDIO_CLK	1	O	8	Сигнал синхронизации	AM15
SDIO_CMD	1	O	8	Линия команды	AL15
SDIO_D0, …, SDIO_D3	4	I/O	8	Шина данных	AK15, AH15, AG15, AF15
Интерфейс USB					
USB0PP, USB0PN	2	I/O		Дифференциальная линия данных	A18, B18
VBUS	1	I		Установка соединения	A19
USB0ID	1	I		Конец кабеля USB	B19
USB_CLK	1	I		Опорный синхросигнал	E18
Интерфейс I2C#0					
I2C0_SDA	1	I/O		Последовательная линия данных	AH14
I2C0_SCL	1	O		Линия синхронизации	AJ14
Интерфейс I2C#1					
I2C1_SDA	1	I/O		Последовательная линия данных	AE14
I2C1_SCL	1	O		Линия синхронизации	AG14
Интерфейс I2C#2					
I2C2_SDA	1	I/O		Последовательная линия данных	AC14
I2C2_SCL	1	O		Линия синхронизации	AD14
Интерфейс GPIO#0					
GPIO0_0, …, GPIO0_7	8	I/O	4	Выводы общего назначения	AE12, AD12, AA12, AC12, AG12, AH12, AB12, AJ12
Интерфейс GPIO#1					
GPIO1_0, …, GPIO1_7	8	I/O	4	Выводы общего назначения	AL12, AL11, AM12, AM10, AK11, AJ10, AL10, AM11
Интерфейс цифрового AUDIO					
AUD_SCLK	1	O	4	Синхросигнал выдачи отсчетов	AB10
AUD_WS	1	O	4	Признак левого / правого канала	AC10
AUD _D0, …, AUD _D3	4	O	4	Шина выдачи отсчетов многоканального звука	AD10, AG10, AE10, AH10
SPDIF	1	O	4	Выход аудиосигнала по IEC 60958 и IEC 61937	AA10

Продолжение таблицы 1.17

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
Интерфейс цифрового видео выхода VOUT#0					
VOUT0_CLKO	1	O	4	Тактовый сигнал	A28
VOUT0_DE	1	O	4	Передачи видимой части кадра	F28
VOUT0_HSYNC	1	O	4	Синхронизация строк	C28
VOUT0_VSYNC	1	O	4	Синхронизация кадров	E28
VOUT0_FIELD	1	O	4	Выбор поля кадра	B28
VOUT0_D0, …, VOUT0_D23	24	O	4	Шина видеоданных	C25,H26,B26,G26,A26,A24, L27,F25,K27,H27,H25,J27,D27, M26,E27,F27,C27,C26,B27, M28,K28,H28,J28,G28
Интерфейс цифрового видео выхода VOUT#1					
VOUT1_CLKO	1	O	4	Тактовый сигнал	E26
VOUT1_DE	1	O	4	Передачи видимой части кадра	K26
VOUT1_HSYNC	1	O	4	Синхронизация строк	D25
VOUT1_VSYNC	1	O	4	Синхронизация кадров	J26
VOUT1_FIELD	1	O	4	Выбор поля кадра	F26
VOUT1_D0, …, VOUT1_D23	24	O	4	Шина видеоданных	H24,G24,K24,J24,E24,C24,D29, C29,F24,M25,B29,B24,J29,J25, F29,H29,M29,E25,L29,K29,E29, B25,L25,K25
Интерфейс цифрового видео входа VIN#0					
VIN0_CLKI	1	I		Тактовый сигнал	J4
VIN0_DE	1	I		Передача видимой части кадра	H4
VIN0_HSYNC	1	I		Синхронизация строк	L4
VIN0_VSYNC	1	I		Синхронизация кадров	K4
VIN0_FIELD	1	I		Выбор поля кадра	M4
VIN0_D0, …, VIN0_D23	24	I		Шина видеоданных	C6,F6,B6,D6,K6,H6,E6,J6,B5, A5,L6,C5,F5,E5,G5,H5,J5,C4, K5,M5,B4,F4,D4,E4
Интерфейс цифрового видео входа VIN#1					
VIN1_CLKI	1	I		Тактовый сигнал	E8
VIN1_DE	1	I		Передача видимой части кадра	K9
VIN1_HSYNC	1	I		Синхронизация строк	G9
VIN1_VSYNC	1	I		Синхронизация кадров	J9
VIN1_FIELD	1	I		Выбор поля кадра	F8
VIN1_D0, …, VIN1_D23	24	I		Шина видеоданных	A9,B9,E9,C9,H8,K8,J8,C8,D8, C7,L8,M8,A7,B7,H7,M7,K7,J7, H9,E7,G7,F7,B8,F9
Интерфейс JTAG					
TCK	1	I		Тактовый сигнал	AD11
TMS	1	I		Режим тестирования	AB11
TDI	1	I		Вход данных	AC11
TDO	1	O	4	Выход данных	AF11
TRST	1	I		Сброс тестового порта	AA11
DEBUG_HALT_N	1	I		Останов процессора	AH11
Общее управление					
TMODE	1	I		Тестовый режим	AG11
SYS_RST	1	I		Системный сброс	AD13
CLK_TMR_MPIC	1	O	4	Вывод системного таймера	AA14
MCHKIN	1	I		Вход системной ошибки	AB13

Продолжение таблицы 1.18

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
MCHKOUT	1	O	4	Выход системной ошибка	AC13
Опорные синхросигналы					
CLK_CPU	1	I		PLL_CPU	L15
CLK_DDR	1	I		PLL_DDR	L16
CLK_VDU	1	I		PLL_VDU	L17
CLK_AUD	1	I		PLL_AUD	L14
Питание					
DDR3_VDDQ	48			Цифровое питание DDR3	A2,B3,D1,G2,H3,K1,N2,P3,T1, U3,W2,Y3,AB1,AB5,AE2,AF3, AH1,AH4,AH6,AJ3,AK5,AK7, AL2,AM3,A31,B30,D32,G31, H30,K32,N31,P30,T32,U30, W31,Y30,AB28,AB32,AE31, AF30,AH27,AH29,AH32,AJ30, AK26,AK28,AL31,AM30
DDR3_VSSQ	50			Цифровая земля DDR3	A1,A3,D2,G1,G3,K2,L3,N1,N3, T2,W1,W3,AB2,AC3,AD4,AD6, AE1,AE3,AF5,AF7,AH2,AL1, AL3,AM4,AM6,A30,A32,D31, G30,G32,K31,L30,N30,N32,T31, W30,W32,AB31,AC30,AD27, AD29,AE30,AE32,AF26,AF28, AH31,AL30,AL32,AM27,AM29
DDR3_VREF	4			Опорное напряжение DDR3	E3, AB7, E30, AB26
DDR3_PLL_VCCA	2			Аналоговое питание DDR3 PLL	AA6, AA27
DDR3_PLL_VSSA	2			Аналоговая земля DDR3 PLL	AA7, AA26
SD_VCCD1P0	24			Аналоговое питание SERDES 1,0В	G10, G11, G12, G13, G20, G21, G22, G23, H10, H11, H12, H13, H20, H21, H22, H23, K20, K21, K22, K23, L10, L11 ,L12, L13,
SD_VCCA1P8	8			Аналоговое питание SERDES 1,8В	J10, J11, J12, J13, J20, J21, J22, J23
SD_VSSA	8			Аналоговая земля SERDES	C10, C11, C12, C13, C20, C21, C22, C23
SD_RBIAS	8			Калибровочный резистор 2кОм ±1%	K10, K11, K12, K13, L20, L21, L22, L23
USB_VCCA3P3	1			Питание USB	D18
USB_VCC18	1			Аналоговое питание USB	E19
USB_VSSA	1			Аналоговая земля для USB	C18
USB_VCCCORE1P0	1			Питание для ФАПЧ в USB	D19
USB_RBIAS	1			Калибровочный резистор 135 Ом ±1%	C19
TEMP_VDDA18	3			Аналоговое питание датчика температуры	AA19, N4, P28
TEMP_VSSA	3			Аналоговая земля датчика температуры	Y18, P5, P29
PLL_CPU_VDDA18	4			Аналоговое питание PLL	G18, J18, H18, K18
PLL_VSSA	8			Аналоговая земля PLL	G17, G19, J17, J19, H17, H19, K17, K19
CLK_DVDD	4			Цифровое питание буферов CLK	G15, J15, H15, K15

Продолжение таблицы 1.19

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
CLK_DVDD_ESD	1			Питание ESD защиты буферов синхронизации	M16
CLK_DVSS_ESD	1			Земля ESD защиты буферов синхронизации	M17
IO_DVDD	7			Цифровое питание КМОП буферов	AA13, AF10, AF12, AF14, AJ11, AJ13, AJ15
MEM_DVDD	8			Цифровое питание буферов интерфейса статической памяти	AC17, AC19, AC21, AC23, AJ17, AJ19, AJ21, AJ23
MII_DVDD	3			Цифровое питание буферов MII	AA8, AF9, AJ8
LSCB_DVDD	3			Цифровое питание буферов LSCB	Y24, AE25, AJ24
VIDEO_DVDD	12			Цифровое питание буферов видео	D5, D7, D9, L5, L7, L9, D24, D26, D28, L24, L26, L28
IO_DVSS	33			Цифровая земля КМОП буферов	A4, A6, A8, A25, A27, A29, G4, G6, G8, G25, G27, G29, AJ25, Y25, AB9, AB14, AE8, AE11, AE13, AE15, AE24, AF16, AF18, AF20, AF22, AK9, AK10, AK12, AK14, AM16, AM18, AM20, AM22,
LVDS_DVDD	4			Цифровое питание буферов LVDS	C14, C16, F15, F17
LVDS_DVSS	4			Цифровая земля буферов LVDS	C15, C17, F14, F16
CORE_VDD	119			Цифровое питание ядра	F20,F21,F22,F23,G16,J16,H16, K16,M9,M12,M13,M20,M21, M24,N5,N8,N9,N12,N13,N16, N17,N20,N21,N24,N25,N28,P6, P7,P10,P11,P14,P15,P18,P19, P22,P23,P26,P27,R6,R7,R10,R11, R14,R15,R18,R19,R22,R23,R26, R27,T4,T5,T8,T9,T12,T13,T16, T17,T20,T21,T24,T25,T28,T29, U4,U5,U8,U9,U12,U13,U16,U17, U20,U21,U24,U25,U28,U29,V6, V7,V10,V11,V14,V15,V18,V19, V22,V23,V26,V27,W6,W7,W10, W11,W14,W15,W18,W19,W22, W23,W26,W27,Y4,Y5,Y8,Y9, Y12,Y13,Y16,Y17,Y20,Y21,Y28, Y29,AA4,AA5,AA28,AA29, AB19

Продолжение таблицы 1.20

Обозначение	Кол-во	Тип	Вых. ток, мА	Функциональное назначение	Выводы корпуса
CORE_VSS	117			Цифровая земля ядра	F10,F11,F12,F13,G14,J14,H14, K14M6,M10,M11,M14,M15, M18,M19,M22,M23,M27,N6, N7,N10,N11,N14,N15,N18,N19, N22,N23,N26,N27, N29,P4,P8, P9,P12,P13,P16,P17,P20,P21, P24,P25,R4,R5,R8,R9,R12,R13, R16,R17,R20,R21,R24,R25,R28, R29,T6,T7,T10,T11,T14,T15, T18,T19,T22,T23,T26,T27,U6, U7,U10,U11,U14,U15,U18,U19, U22,U23,U26,U27,V4,V5,V8, V9,V12,V13,V16,V17,V20,V21, V24,V25,V28,V29,W4,W5,W8, W9,W12,W13,W16,W17,W20, W21,W28,W29,Y6,Y7,Y10,Y11, Y14,Y15,Y19,Y22,Y23,Y26,Y27
NC	2			Не используется	F18, F19

Примечания:

1) Для выводов с префиксом «n» на конце имени сигнала активным является низкий уровень сигнала.

2) Используемые обозначения типов выводов:

I –вход;
O - выход;
I/O – двунаправленный вывод;
S –питание.

1.2.2 Конструктивные характеристики

Микросхема изготавливается в 1024-выводном корпусе типа High-Performance Flip Chip Ball Grid Array (HFCBGA). Внешний вид корпуса микросхемы представлен на рисунке 2.2 с указанием сведений о габаритных и установочных размерах. Для повышения отводимого тепла корпуса HFCBGA дополняют металлической пластиной, которая контактирует с обратной стороной кристалла и уменьшает термическое сопротивление кристалл-среда.

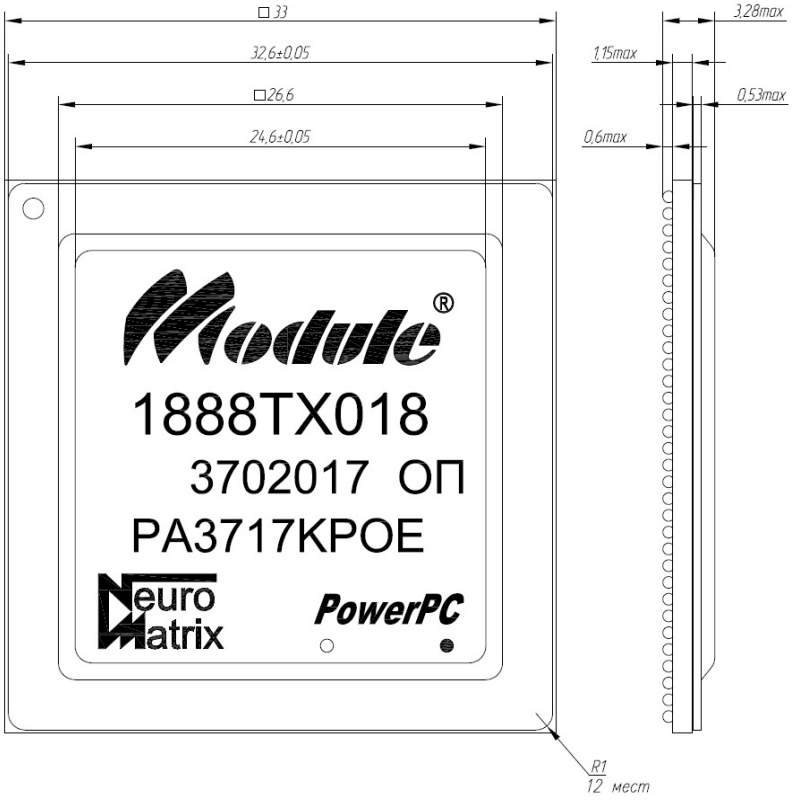


Рисунок 1.2 – Корпус микросхемы

Основные характеристики корпуса:
Механические характеристики:
• размер – 33х33 мм2;
• шаг выводов- 1,00 мм;
• размер вывода – 0,60 мм.

Электрические характеристики:
• индуктивность вывода – не более 6,5 нГн;
• сопротивление вывода – не более 0,270 Ом;
• ёмкость вывода – не более 1,5 пФ.

Тепловые характеристики:
• Тепловое сопротивление при скорости обдува 0 м/с – 5.6 °С/Вт;
• Тепловое сопротивление при скорости обдува 1 м/с – 4,8 °С/Вт;
• Тепловое сопротивление при скорости обдува 2 м/с – 4,10 °С/Вт.

1.2.3 Электрические характеристики микросхемы

Электрические параметры микросхем должны соответствовать нормам, приведенным в таблице 2.3

Таблица 1.21 – Электрические параметры микросхемы

Наименование параметра, единица измерения	Обозна-чение	Норма параметра,		Режим измерения	Темпера-тура среды, 0С
		не менее	не более		
Выходное напряжение низкого уровня буферов КМОП, В	UOL1	-	0,2	UCC1 = 3,45 В, втекающий выходной ток IOL1= 100 мкА	+25 минус 50 + 70
Выходное напряжение высокого уровня буферов КМОП, В	UOH1	UCC1 - 0,2В	-	UCC1 = 3,15 В, вытекающий выходной ток IOH1= -100 мкА	+25 минус 50 + 70
Входной ток утечки низкого уровня, мкА	IILL	-	10	UIL = 0 В	+25 минус 50 + 70
Входной ток утечки высокого уровня, мкА	IILH	-	10	UIH = UCC1	+25 минус 50 + 70
Ток утечки низкого уровня в состоянии «Выключено», мкА	IOZL	-	10	UIL = 0 В	+25 минус 50 + 70
Ток утечки высокого уровня в состоянии «Выключено», мкА	IOZH	-	10	UIH = UCC1	+25 минус 50 + 70

Номинальное значение напряжений питания микросхемы должно быть:

- Напряжение питания UCC1 буферов КМОП 3,3 В ± 0,15 В;
- Напряжение питания UCC2 буферов DDR3 1,5 В ± 0,1 В;
- Напряжение питания UCC3 ядра микросхемы 1,0 В ± 0,5 В;
- Напряжение питания UCC4 буферов LVDS 1,8 В ± 0,09 В.

Значения предельно-допустимых и предельных электрических режимов эксплуатации микросхем в диапазоне рабочих температур среды должны соответствовать нормам, приведенным в таблице.

Таблица 1.22 – Нормы предельно-допустимых и предельных электрических режимов эксплуатации микросхем

Наименование параметра режима, единица измерения	Обозначение	Норма параметра			
		предельно-допустимого режима,		предельного режима,	
		не менее	не более	не менее	не более
Напряжение питания КМОП буферов ввода/вывода, В	U_{CC1}	3,15	3,45	минус 0,4	4,0
Напряжение питания интерфейса с памятью DDR3, В	U_{CC2}	1,40	1,60	минус 0,4	1,975
Напряжение питания ядра, В	U_{CC3}	0,95	1,05	минус 0,5	1,4
Напряжение питания LVDS буферов ввода/вывода, В	U_{CC4}	1,72	1,88	минус 0.4	1,97
Напряжение на выводах «вход-выход» в состоянии «выключено», В	U_{IOZ}	0	3,45	минус 0.4	4,0
Входное напряжение высокого уровня КМОП буферов ввода/вывода, В	U_{IH}	1,8	3,45	-	4,0
Входное напряжение низкого уровня КМОП буферов ввода/вывода, В	U_{IL}	0	0,8	минус 0.4	-
Выходной ток, мА 4-миллиамперный выход 6-миллиамперный выход 8-миллиамперный выход	I_{OUT}	 минус 4 минус 6 минус 8	 +4 +6 +8	 минус 13 минус 13 минус 13	 +13 +13 +13

1.3 Устройство и работа микросхемы

1.3.1 Структурная схема и шинная организация микросхемы

Структурная схема микросхемы приведена на Рис.1.3.

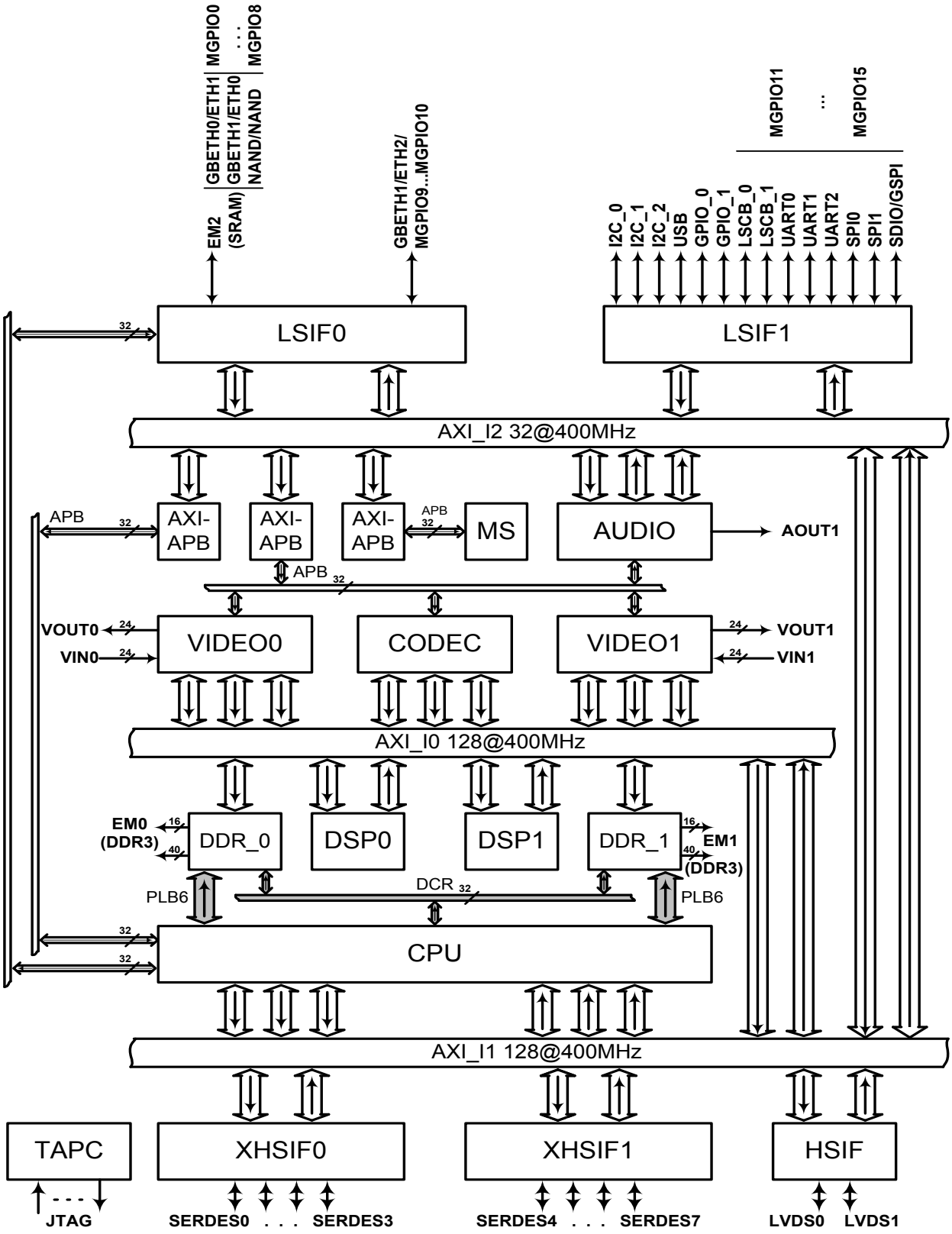


Рисунок 1.3 - Структурная схема микросхемы

Микросхема представляет собой «систему на кристалле», реализованную на базе двух шинных архитектур разных производителей:

1. Система шин AMBA фирмы ARM является основной и используется для реализации большинства межсоединений основных узлов и подсистем микросхемы. Данная система включает в себя:

- 128-, 64- и 32-разрядные системные шины AXI, характеризующиеся высокой пропускной способностью;
- 32-разрядные шины APB для подключения относительно медленных периферийных устройств и конфигурационных регистров различных узлов и подсистем;
- коммутаторы AXI шин и всевозможные мосты, которые связывают различные по типу, разрядности и частоте работы шины.

На Рис.1.3 внутренняя стрелка в обозначении AXI-шинs указывает направление передачи адресов. Устройство, выставляющее адреса, является активным или «master»-устройством, а принимающее адреса – пассивным или «slave»-устройством.

Как показано на Рис.1.1, коммутатор AXI шин верхнего уровня, работающий на частоте 400 МГц, разбит на три части:

AXI_I0 - 128-разрядный коммутатор, соединяющий узлы и подсистемы, участвующие в приеме, хранении, обработке и выдачи мультимедийной информации;

AXI_I1 - 128-разрядный центральный коммутатор, к которому подключены центральное процессорное устройство (CPU), три подсистемы наиболее скоростных интерфейсов и два других шинных коммутатора;

AXI_I2 - 32-разрядный коммутатор, соединяющий медленные периферийные узлы и подсистемы, а также все мосты AXI-APB.

2. Система шин PLB фирмы IBM используется для организации связей между устройствами, входящими в состав центрального процессорного устройства, построенного на базе процессорных ядер PowerPC, а также для подключения к нему интерфейсов с памятью DDR3. Данная система обеспечивает когерентность содержимого L2 кэш-памяти и включает в себя:

- 128-разрядные системные шины PLB6, характеризующиеся высокой пропускной способностью;
- 32-разрядные шины DSR для подключения относительно медленных периферийных устройств и конфигурационных регистров различных узлов и подсистем;
- коммутаторы PLB шин и всевозможные мосты, которые связывают различные по типу, разрядности и частоте работы шины.

Микросхема содержит следующие основные узлы и подсистемы:

CPU (Central Processing Unit) – центральное процессорное устройство, осуществляющее общее управление системой и вторичную обработку сигналов.

CPU содержит:

- два 32-разрядных суперскалярных процессорных ядра PPC 470S, в состав каждого из которых входят управляющий RISC-процессор, 64-разрядный сопроцессор с плавающей точкой, кэш-память команд первого уровня – 32 КБ, кэш-память данных первого уровня объемом 32 КБ, кэш-память второго уровня - 512 КБ;
- банк внутренней памяти (**IM0**) объемом 256 КБ;
- ПЗУ начальной загрузки (**BOOT ROM**);
- контроллер внешней памяти типа SRAM/NOR;
- два контроллера ПДП;
- контроллер прерываний;

- систему внутренних шин фирмы IBM, в состав которой входят системные 128–разрядные шины PLB6, конфигурационные 32–разрядные шины DCR и различные шинные коммутаторы и переходные мосты;
- три переходных моста с шины PLB6 на системную шину AXI (PLB6_AXI_0,…, PLB6_AXI_2);
- три переходных моста с шины AXI на системную шину PLB6 (AXI_PLB6_0,…, AXI_PLB6_2);
- переходной мост с шины APB на шину DCR.

Все блоки процессорных ядер PPC 470S (за исключением кэш-памяти второго уровня) работают на частоте 800 МГц, остальные схемы CPU – на частоте 400 МГц.

CPU подключен к системе шин AMBA микросхемы:

- тремя шинами AXI как устройство-«master»,
- тремя шинами AXI как устройство-«slave»,
- одной шиной APB как устройство-«slave».

CPU напрямую соединен с каждым из интерфейсов внешней памяти DDR0 и DDR1:

- системной шиной PLB6 как устройство-«master»,
- шиной DCR как устройство-«master»,

CPU напрямую соединен с интерфейсом внешней памяти SRAM/NOR, входящим в состав подсистемы LSIF0, специализированной шиной.

DSP0, DSP1 (Digital Signal Processor) – два одинаковых устройства цифровой обработки сигналов, каждое из которых содержит:

- два процессорных ядра NMC3 с архитектурой NeuroMatrix;
- восемь банков внутренней памяти суммарным объемом 256КБ;
- два контроллера ПДП;
- два контроллера прерываний;
- два таймера;
- кэш-память второго уровня объемом 128 КБ.

Каждый DSPx работает на частоте 400 МГц и подключен к системе шин AMBA микросхемы двумя шинами AXI - как устройство-«master» и как устройство-«slave».

VIDEO0, VIDEO1 – две одинаковых подсистемы работы с видеосигналами. Каждая подсистема содержит:

- контроллер вывода видеоизображения на выходной видеоинтерфейс;
- два контроллера захвата видеоизображения (один контроллер захватывает изображение с внешнего входного видеоинтерфейса, другой захватывает изображение, генерируемое контроллером вывода видеоизображения).

Каждая видеоподсистема работает на частоте 400 МГц и подключена к системе шин AMBA микросхемы:

- тремя шинами AXI как устройство-«master»;
- четырьмя шинами APB как устройство-«slave».

CODEC – подсистема кодирования/декодирования видеосигналов и графических изображений, содержащая:

- контроллер кодирования/декодирования видеоизображения;
- контроллер кодирования/декодирования графических изображений.

Подсистема работает на частоте 400 МГц и подключена к системе шин AMBA микросхемы:

- тремя шинами AXI как устройство-«master»;
- двумя шинами APB как устройство-«slave».

AUDIO – подсистема обработки аудиосигналов, содержащая:

- контроллер ПДП;
- контроллер интерфейса I2S;
- контроллер интерфейса SPDIF.

Подсистема работает на частоте 400 МГц и подключена к системе шин AMBA микросхемы:

- двумя шинами AXI – как устройство-«master» и как устройство-«slave»;
- одной шиной APB – как устройство-«slave».

DDR0, DDR1 – два одинаковых интерфейса с внешней памятью типа DDR3, каждый из которых содержит:

- мультиплексор входных системных шин PLB6 и AXI с переходными мостами;
- контроллер DDR3;
- физический интерфейс (PHY) с памятью DDR3.

Каждый интерфейс подключен к узлам системы тремя шинами:

- шиной AXI как устройство-«slave» к системе шин AMBA микросхемы;
- шиной PLB6 как устройство-«slave» к CPU;
- шиной DCR как устройство-«slave» к CPU.

XHSIF0, XHSIF1 (Extra High Speed Interface #0, #1)– две одинаковых подсистемы суперскоростных интерфейсов, каждая из которых содержит:

- интерфейс PCIe2.0x4, который может программно настраиваться на работу по 1, 2 или 4 высокоскоростным дуплексными линиям;
- три дуплексных последовательных интерфейсов ARINC 818;
- два дуплексных последовательных интерфейса FibreChannel;
- систему мультиплексирования, позволяющую выводить на четыре физических PHY-интерфейса SERDES различные комбинации перечисленных интерфейсов.

Каждая подсистема XHSIFx работает на частоте 400 МГц и подключена к системе шин AMBA микросхемы двумя шинами AXI - как «master» и как «slave».

HSIF (High Speed Interface) – подсистема высокоскоростных интерфейсов, содержащая следующие блоки:

- два дуплексных последовательных интерфейса SpaceWire (HSCB0, HSCB1) для подключения к телекоммуникационной сети бортовых систем;
- контроллер ПДП с функциями шифрования.

Подсистема HSIF работает на частоте 400 МГц и подключена к системе шин AMBA микросхемы двумя шинами AXI - как «master» и как «slave».

LSIF0 (Low Speed Interface #0) – подсистема низкоскоростных интерфейсов, в состав которой входят:

- 32-разрядный интерфейс с внешней памятью типа SRAM/NOR (EM2) с байтовым ECC;
- 8-разрядный асинхронный интерфейс с энергонезависимой внешней памятью типа NAND, поддерживающий набор команд стандарта ONFI 2.3, два алгоритма аппаратной коррекции ошибок (4 ошибки на 512 байт и 24 ошибки на 1024 байт), аппаратную защиту NAND-памяти от записи, работу с микросхемами MLC и SLC флэш-памяти;
- два 8-разрядных дуплексных интерфейса GMII (GBETH0, GBETH1), каждый из которых поддерживает последовательный управляющий интерфейс SMI для конфигурации параметров работы внешней микросхемы физического интерфейса Gigabit Ethernet и отладочный EDCL-протокол;
- три 2-разрядных дуплексных интерфейса RMII (ETH0,…, ETH2), каждый из которых предназначен для подключения к устройствам, реализующим физический уровень интерфейсов Fast Ethernet;
- одиннадцать байтовых интерфейсов общего назначения (MGPIO0,…, MGPIO10) с программной настройкой каждого бита на режим входа, выхода или прерывания;
- систему мультиплексирования, позволяющую подключать к внешним выводам микросхемы различные комбинации перечисленных интерфейсов, как указано в таблице 1.1.

Подсистема LSIF0 работает на частоте 100 МГц и подключена к системе шин AMBA микросхемы двумя шинами AXI - как «master» и как «slave».

LSIF1 (Low Speed Interface #1) – подсистема низкоскоростных интерфейсов, в состав которой входят:

- два последовательных дуплексных интерфейса МКИО (LSCB0, LSCB1);
- три последовательных дуплексных интерфейса UART (UART0,…, UART2), каждый из которых содержит программируемый генератор тактового сигнала для формирования битового интервала различной длины и FIFO-буфера для приема и передачи данных;
- три последовательных дуплексных интерфейса SPI (SPI0, SPI1, GSPI) с программируемой частотой работы и поддержкой различных форматов обмена данными (Motorola SPI, Texas Instruments synchronous serial frame format, National Semiconductor Microwire frame format);
- 4-разрядный интерфейс карты памяти SDIO со встроенным контроллером ПДП;
- последовательный полудуплексный интерфейс USB 2.0, который содержит встроенный физический интерфейс (USB-PHY) и контроллер ПДП, может работать с различными по скорости устройствами как в качестве хост-контроллера шины USB, так и в качестве периферийного устройства, поддерживает до 15 конечных устройств для передачи и приема данных и обеспечивает аппаратную диспетчеризацию всех USB-транзакций..
- три последовательных полудуплексных интерфейса I2C (I2C0,…, I2C2);
- семь байтовых интерфейсов общего назначения (GPIO0, GPIO1, MGPIO11,…, MGPIO15) с программной настройкой каждого бита на режим входа, выхода или прерывания;
- систему мультиплексирования, позволяющую подключать к внешним выводам микросхемы различные комбинации перечисленных интерфейсов, как указано в таблице 1.1.

Подсистема LSIF1 работает на частоте 100 МГц и подключена к системе шин AMBA микросхемы двумя шинами AXI - как «master» и как «slave».

MS (Management Subsystem) – подсистема общего управления, в состав которой входят:

- четыре генератора синхросигналов для CPU, DDR, VIDEO и AUDIO;
- блок таймеров;
- сторожевой таймер (Watchdog Timer);
- сенсоры температуры и напряжения;
- системный контроллер.

Подсистема MS подключена к системе шин AMBA 32-разрядной шиной APB как «slave» устройство.

TAPC – контроллер интерфейса JTAG, поддерживающий подключение внешнего отладочного устройства

Каждая из подсистем DSP0, DSP1, XHSIF0, XHSIF1, HSIF, LSIF0, LSIF1 использует внутренний AXI-коммутатор для подключения своих устройств к внешним AXI- шинам данной подсистемы.

1.3.2 Карты памяти микросхемы

Микросхема имеет два основных адресных пространства (по числу используемых шинных архитектур):

- карта памяти активных устройств подключенных к шине PLB6 в CPU;
- карта памяти активных устройств подключенных к шинам AXI микросхемы.

Активными устройствами на шинах являются процессорные ядра, контроллеры ПДП, ряд интерфейсов (например, PCI) и другие устройства, инициирующие обмен данными по шинам.

Имеется третья карта памяти – карта памяти процессорных ядер NMC, которая немного отличается от карты памяти AXI шины и подробно описана в разделе 1.3.5.

1.3.2.1 Карты памяти активных устройств на шине PLB6

Карта памяти шины PLB6 приведена в таблице 1-2.

Таблица 1.23 - Карта памяти активных устройств на шине PLB6

PLB6 Адреса От	PLB6 Адреса До	PLB6 Endian1)	Адресуемое устройство	Выходной порт CPU 2)
0x00_00000000	0x01_FFFFFFFF	BE/LE	DDR3 EM0 (8GB)	
0x02_00000000	0x03_FFFFFFFF	BE/LE	DDR3 EM1 (8GB)	
0x04_00000000	0x05_FFFFFFFF	BE/LE	SRAM EM2 (8GB)	
0x06_00000000	0x07_FFFFFFFF	BE/LE	SRAM EM3 (8GB)	
0x08_00000000	0x10_0003FFFF		Резерв (8GB+256KB)	
0x10_00040000	0x10_0007FFFF	BE/LE	IM0 (256KB)	
0x10_00080000	0x10_0FFFFFFF		Резерв (255,5MB)	
0x10_10000000	0x10_17FFFFFF	LE	PCI0 (128MB)	PLB6_AXI_0
0x10_18000000	0x10_1FFFFFFF	LE	PCI1 (128MB)	PLB6_AXI_0
0x10_20000000	0x10_2FFFFFFF	LE	SRAM EM2 (256MB)	PLB6_AXI_0
0x10_30000000	0x10_37FFFFFF	LE	SRAM EM3 (128MB)	PLB6_AXI_0
0x10_38000000	0x10_3800FFFF	LE	Регистры MS (64KB)	PLB6_AXI_0
0x10_38010000	0x10_38023FFF	LE	Регистры MMS (80KB)	PLB6_AXI_0
0x10_38024000	0x10_380FFFFF		Резерв (880KB)	
0x10_38100000	0x10_381FFFFF	LE	AXI_SYS_CFG (1MB)	PLB6_AXI_0
0x10_38200000	0x10_382FFFFF	LE	AXI_DSP0_CFG(1MB)	PLB6_AXI_0
0x10_38300000	0x10_383FFFFF	LE	AXI_DSP1_CFG(1MB)	PLB6_AXI_0
0x10_38400000	0x10_38FFFFFF		Резерв (12MB)	
0x10_39000000	0x10_3901FFFF	LE	DSP0_NM0 (128KB)	PLB6_AXI_0
0x10_39020000	0x10_3903FFFF	LE	DSP0_NM1 (128KB)	PLB6_AXI_0
0x10_39040000	0x10_3905FFFF	LE	DSP1_NM0 (128KB)	PLB6_AXI_0
0x10_39060000	0x10_3907FFFF	LE	DSP1_NM1 (128KB)	PLB6_AXI_0
0x10_39080000	0x10_39080FFF	LE	I2S (4KB)	PLB6_AXI_0
0x10_39081000	0x10_39081FFF	LE	SPDIF (4KB)	PLB6_AXI_0
0x10_39082000	0x10_3BFFFFFF		Резерв (48MB–520KB)	
0x10_3C000000	0x10_3C00FFFF	LE	Регистры XHSIF0(64KB)	PLB6_AXI_0
0x10_3C010000	0x10_3C01FFFF	LE	Регистры XHSIF1(64KB)	PLB6_AXI_0
0x10_3C020000	0x10_3C02FFFF	LE	Регистры HSIF (64KB)	PLB6_AXI_0
0x10_3C030000	0x10_3C04FFFF	LE	Регистры LSIF0(128KB)	PLB6_AXI_0
0x10_3C050000	0x10_3C06FFFF	LE	Регистры LSIF1(128KB)	PLB6_AXI_0
0x10_3C070000	0x10_3C0FFFFF	LE	Резерв (576KB)	
0x10_3C100000	0x10_3C2FFFFF	LE	AXI_XHSIF0_CFG (2MB)	PLB6_AXI_0
0x10_3C300000	0x10_3C4FFFFF	LE	AXI_XHSIF1_CFG (2MB)	PLB6_AXI_0
0x10_3C500000	0x10_3C6FFFFF	LE	AXI_HSIF_CFG (2MB)	PLB6_AXI_0
0x10_3C700000	0x10_3C8FFFFF	LE	AXI_LSIF0_CFG (2MB)	PLB6_AXI_0
0x10_3C900000	0x10_3CAFFFFF	LE	AXI_LSIF1_CFG (2MB)	PLB6_AXI_0
0x10_3CB00000	0x10_3CB3FFFF	LE	Регистры XHSIF0 (256KB)	PLB6_AXI_0
0x10_3CB40000	0x10_3CB7FFFF	LE	Регистры XHSIF1 (256KB)	PLB6_AXI_0
0x10_3CB80000	0x10_3CB8FFFF	LE	Регистры HSIF (64KB)	PLB6_AXI_0
0x10_40000000	0x10_FFFFFFFF		Резерв (3G+53M–576K)	
0x11_00000000	0x11_0003FFFF	BE	BOOT ROM (256KB)	
0x11_00040000	0x11_FFFFFFFF		Резерв (4G-256KB)	
0x12_00000000	0x12_FFFFFFFF	LE	PCI0 (4GB)	PLB6_AXI_1
0x13_00000000	0x13_FFFFFFFF	LE	PCI1 (4GB)	PLB6_AXI_2

Примечания:

Показана эндианность (BE – big endian, LE – little endian) при обращении процессорных ядер PPC к устройствам через шину PLB6.

Нулевой мост PLB6_AXI_0 транслирует адреса [0x10_10000000:0x10_ FFFFFFFF] шины PLB6 в адреса [0x0_10000000:0x0_FFFFFFFF] шины AXI.

Первый мост PLB6_AXI_1 транслирует адреса [0x12_00000000:0x12_ FFFFFFFF] шины PLB6 в адреса [0x3_00000000:0x3_FFFFFFFF] шины AXI.

Второй мост PLB6_AXI_2 транслирует адреса [0x13_00000000:0x13_ FFFFFFFF] шины PLB6 в адреса [0x4_00000000:0x4_FFFFFFFF] шины AXI.

1.3.2.2 Карты памяти активных устройств на шине AMBA

Карта памяти шин AXI приведена в таблице 1.6.

Таблица 1.24 - Карта памяти активных устройств на шине AMBA.

AXI адреса от	AXI адреса до	AXI устройство	Узел / подсистема (входной порт) 1)
0x0_00000000	0x0_0003FFFF	BOOTROM (256KB)	CPU(AXI_PLB6_0)
0x0_00040000	0x0_0007FFFF	IM0 (256KB)	CPU(AXI_PLB6_0)
0x0_00080000	0x0_00FFFFFF	Резерв (15,5MB)	
0x0_01000000	0x0_03FFFFFF	DCR (48MB)	CPU (APB)
0x0_04000000	0x0_0FFFFFFF	Резерв (192MB)	
0x0_10000000	0x0_17FFFFFF	PCI0 (128MB)	XHSIF0
0x0_18000000	0x0_1FFFFFFF	PCI1 (128MB)	XHSIF1
0x0_20000000	0x0_2FFFFFFF	SRAM EM2 (256MB)	LSIF0
0x0_30000000	0x0_37FFFFFF	SRAM EM3 (128MB)	LSIF0
0x0_38000000	0x0_3800FFFF	Регистры MS (64KB)	MS
0x0_38010000	0x0_38023FFF	Регистры MMS (80KB)	MMS
0x0_38024000	0x0_380FFFFF	Резерв (880KB)	
0x0_38100000	0x0_381FFFFF	AXI_SYS_CFG (1MB)	
0x0_38200000	0x0_382FFFFF	AXI_DSP0_CFG (1MB)	DSP0
0x0_38300000	0x0_383FFFFF	AXI_DSP1_CFG (1MB)	DSP1
0x0_38400000	0x0_38FFFFFF	Резерв (12MB)	
0x0_39000000	0x0_3901FFFF	DSP0_NM0 (128KB)	DSP0
0x0_39020000	0x0_3903FFFF	DSP0_NM1 (128KB)	DSP0
0x0_39040000	0x0_3905FFFF	DSP1_NM0 (128KB)	DSP1
0x0_39060000	0x0_3907FFFF	DSP1_NM1 (128KB)	DSP1
0x0_39080000	0x0_39080FFF	I2S (4KB)	AUDIO (I2S)
0x0_39081000	0x0_39081FFF	SPDIF (4KB)	AUDIO (SPDIF)
0x0_39082000	0x0_3BFFFFFF	Резерв (48MB – 520KB)	
0x0_3C000000	0x0_3C00FFFF	Регистры XHSIF0 (64KB)	XHSIF0
0x0_3C010000	0x0_3C01FFFF	Регистры XHSIF1 (64KB)	XHSIF1
0x0_3C020000	0x0_3C02FFFF	Регистры HSIF (64KB)	HSIF
0x0_3C030000	0x0_3C04FFFF	Регистры LSIF0 (128KB)	LSIF0
0x0_3C050000	0x0_3C06FFFF	Регистры LSIF1 (128KB)	LSIF1
0x0_3C070000	0x0_3C0FFFFF	Резерв (576KB)	
0x0_3C100000	0x0_3C2FFFFF	AXI_XHSIF0_CFG (2MB)	XHSIF0
0x0_3C300000	0x0_3C4FFFFF	AXI_XHSIF1_CFG (2MB)	XHSIF1
0x0_3C500000	0x0_3C6FFFFF	AXI_HSIF_CFG (2MB)	HSIF
0x0_3C700000	0x0_3C8FFFFF	AXI_LSIF0_CFG (2MB)	LSIF0
0x0_3C900000	0x0_3CAFFFFF	AXI_LSIF1_CFG (2MB)	LSIF1
0x0_3CB00000	0x0_3CB3FFFF	Регистры XHSIF0 (256KB)	XHSIF0
0x0_3CB40000	0x0_3CB7FFFF	Регистры XHSIF1 (256KB)	XHSIF1
0x0_3CB80000	0x0_3CB8FFFF	Регистры HSIF (64KB)	HSIF
0x0_3CB90000	0x0_3FFFFFFF	Резерв (53MB – 576KB)	
0x0_40000000	0x0_7FFFFFFF	DDR3 EM0 (1GB) через AXI	DDR3_0 (AXI)
0x0_80000000	0x0_FFFFFFFF	DDR3 EM1 (2GB) через AXI	DDR3_1 (AXI)
0x1_00000000	0x1_FFFFFFFF	DDR3 EM0 / SRAM EM2 (4GB) через PLB6	CPU(AXI_PLB6_1)
0x2_00000000	0x2_FFFFFFFF	DDR3 EM1 / SRAM EM3 (4GB) через PLB6	CPU(AXI_PLB6_2)
0x3_00000000	0x3_FFFFFFFF	PCI0 (4GB)	XHSIF0
0x4_00000000	0x4_FFFFFFFF	PCI1 (4GB)	XHSIF1

Примечание:

Нулевой мост AXI_PLB6_0 при адресации BOOT ROM транслирует адреса [0x0_00000000:0x0_0003FFFF] шины AXI в адреса [0x11_00000000:0x11_0003FFFF] шины PLB6, а при адресации банка внутренней статической

памяти IM0 – адреса [0x0_00040000:0x0_0007FFFF] шины AXI в адреса [0x10_00040000:0x10_ 0007FFFF] шины PLB6.

Первый мост AXI_PLB6_1 транслирует адреса [0x1_00000000:0x1_FFFFFFFF] шины AXI в адреса [0x00_00000000:0x00_FFFFFFFF] или [0x01_00000000:0x01_FFFFFFFF] шины PLB6. Старшие адреса PLB6 задаются программно с помощью конфигурационного регистра моста.

Второй мост AXI_PLB6_2 транслирует адреса [0x2_00000000:0x2_FFFFFFFF] шины AXI в адреса [0x02_00000000:0x02_FFFFFFFF] или [0x03_00000000:0x03_FFFFFFFF] шины PLB6. Старшие адреса PLB6 задаются программно с помощью конфигурационного регистра моста.

1.3.3 Центральный процессорный узел CPU

1.3.3.1 Введение

Структурная схема ЦПУ подсистемы приведена на рисунке 1.

ЦПУ подсистема построена на встраиваемых PowerPC ядрах, которые образуют полностью когерентную симметричную многопроцессорную систему (SMP). Подсистема включает в себя два ядра PowerPC 470S, каждое из которых подключено к собственной кеш памяти второго уровня (L2C). Контроллер шины PLB6 (bus fabric) подключает к себе кеш памяти второго уровня. Многопроцессорный контроллер прерываний MPIC используется для приема 128 входящих прерываний, из которых 32 приходят из ЦПУ подсистемы и 96 из других блоков СнК, и последующего перевода прерываний на процессоры PowerPC.

Два контроллера прямого доступа к памяти (DMAPLB6) используются для пересылки данных между памятью-памятью или между памятью и устройствами ввода-вывода.

Контроллер шины PLB6 подключает к себе 7 мастеров: две кэш памяти второго уровня L2C, два контроллера прямого доступа к памяти (DMAPLB6), три моста между шинами AHB-PLB6 и 5 ведомых устройств (slaves): первую сборку контроллера DDR3 SDRAM (EM0), вторую сборку контроллеров DDR3 SDRAM (EM1), два контроллера SRAM/NOR памяти, четыре моста PLB6PLB4.

Мосты AHB-PLB4, PLB4-PLB6 предоставляют доступ в ЦПУ подсистему со стороны других подсистем СнК, выходящих на шину AXI. Первая группа мостов AHB-PLB4, PLB4-PLB6 используется для доступа к BOOT ROM и SRAM (IM0) памяти со стороны внешних подсистем. Вторая группа мостов AHB-PLB4, PLB4-PLB6 используется для доступа к DDR3 SDRAM (EM0) памяти со стороны внешних подсистем. Третья группа мостов AHB-PLB4, PLB4-PLB6 используется для доступа к DDR3 SDRAM (EM1) памяти со стороны внешних подсистем.

Первый мост PLB6-PLB4 вместе с арбитром шины PLB4 и контроллером SRAM памяти предоставляет возможность процессорам грузиться из BOOT ROM памяти. Второй мост PLB6-PLB4 вместе с арбитром шины PLB4 и контроллером SRAM памяти предоставляет доступ к внутрикристальной SRAM памяти (IM0); через мост PLB4-AHB процессор может обращаться к другим подсистемам СнК. Третья группа мостов PLB6-PLB4, PLB4-AHB предназначена для обращений к высокоскоростным интерфейсам, находящимся в блоке XHSIF0. Четвертая группа мостов PLB6-PLB4. PLB4-AHB предназначена для обращений к высокоскоростным интерфейсам, находящимся в блоке XHSIF1.

Арбитр DCR шины предоставляет доступ к общей DCR шины со стороны первого и второго ядер PowerPC. Общая шина DCR используется для доступа к управляющим регистрам устройств ЦПУ подсистемы.

ЦПУ подсистема также включает в себя:

Возможность трассировки команд

Возможность логической отладки межпроцессорного взаимодействия

Шину DCR для конфигурирования и управления устройствами ЦПУ подсистемы

JTAG отладочные контроллеры, встроенные в процессорные ядра, и работающие с отладчиком RISCWatch

1.3.3.2 Функциональные характеристики ЦПУ подсистемы

Два PowerPC 470S процессорных ядра, каждое с собственной кэш памятью второго уровня L2C

- Суперскалярный процессор с выдачей до 4 команд за такт
- Поддержка когерентности
- Отладочный JTAG интерфейс для RISCWatch
- Трассировочный RISCTrace интерфейс для внутреннего запоминания трассы команд
- Два контроллера L2C кэш памяти размером по 512КБ

PLB6 Коммутатор шины версии 1.0

- Программируемый для работы на частоте 1:2, 1:3, 1:4 относительно PowerPC 470S и на частоте 1:1 относительно L2C

- Используются три подчиненных (slave) сегмента, нулевой сегмент для системной памяти, первый сегмент для интервенции данных (имеет нулевой размер), второй сегмент используется для 64-адресного пространства ввода-вывода

- 128 битные шины данных для Записи и Чтения

- 64 битная адресная шины

- Когерентная поддержка для Сегмента 0

Поддержка PLB6 ПДП(DMA)

- Поддерживаются ПДП(DMA) операции между PLB6 устройствами и AXI устройствами

Два L2C кэш контроллера по 512 КБ

- поддерживают протокол когерентности с 7 состояниями

- одна шина PLB6 Slave для интервенции данных

Поддержка двух интерфейсов PLB4-AHB

Индивидуальные мосты PLB4-AHB используются для каждого PCI Express интерфейса.

Разделение позволяет достичь оптимальной пропускной способности для каждого PCI Express соединения.

DCR шина версии 3.5

- 32 битные шины данных для Записи и Чтения

- Поддержка 32-битных адресов

Многопроцессорный контроллер прерываний (MPIC)

- Поддержка до 4-х процессорных ядер

- Поддержка до 128 источников прерываний

- Реализованы все 128 прерываний, некоторые мультиплексированы.

Генераторы синхросигналов (находятся за пределами подсистемы ЦПУ)

Управление потребляемой мощностью

- управление переходом в спящий режим и пробуждение для ядер PowerPC 470S

Трасса команд

-16Кх64-бит массив памяти для хранения трассы команд (RISCTrace) для одного или двух процессоров

- Программируемые режимы защелкивания

- Программный доступ к буферу трассы команд через DCR шину

Логика для отладки многопроцессорности

- 256x72-бит массив памяти для хранения трассируемых данных

- Программируемые режимы защелкивания

Термодатчики (находятся за пределами подсистемы ЦПУ)

Внутренние датчики напряжения (находятся за пределами подсистемы ЦПУ)

Интерфейс DDR3 SDRAM

- 16 адресных бит и 3 бита выбора банка

- 32 бита данных и 8 бит ECC

- поддержка скорости DDR3 до 1600 MT/s

- поддержка 2 ранков (rank or chip select)
- Настройка внешних ODT для каждого ранка

Замечание: данный чип не поддерживает регистровую динамическую память (registered DIMM).

Внешние входы прерываний

- Чип поддерживает до 11 внешних прерываний, заводимых через входы GPIO.

1149.1 JTAG интерфейс

- Стандартный 5 выводной JTAG интерфейс обеспечивает доступ к основному JTAG контроллеру (находится за пределами подсистемы ЦПУ)
- IEEE 1149.1 JTAG boundary scan поддержка для производителей плат (находятся за пределами подсистемы ЦПУ)
- Обеспечивает доступ к RISCWatch интерфейсу (включая вход останова (halt))

1.3.3.3 Структурная схема ЦПУ подсистемы

Упрощенная структурная схема изображена на рисунке 1.

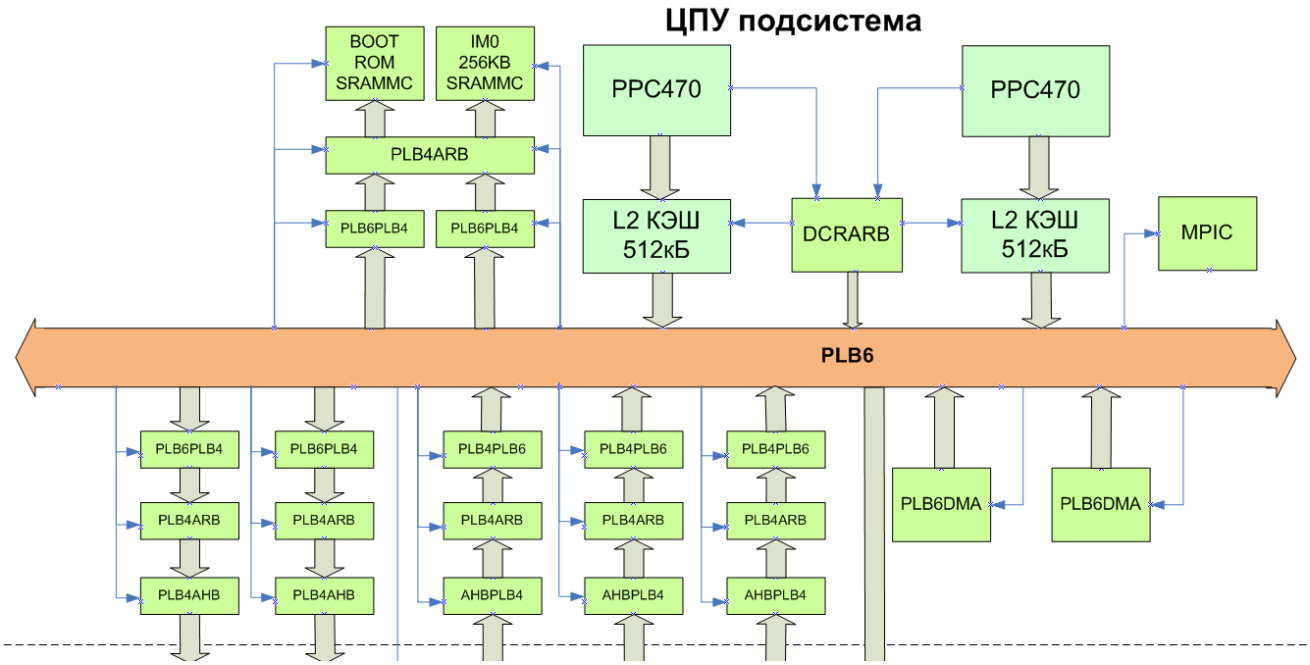


Рисунок 1.4 - Структурная схема ЦПУ подсистемы

Упрощенная структурная схема изображена на рисунке 1.

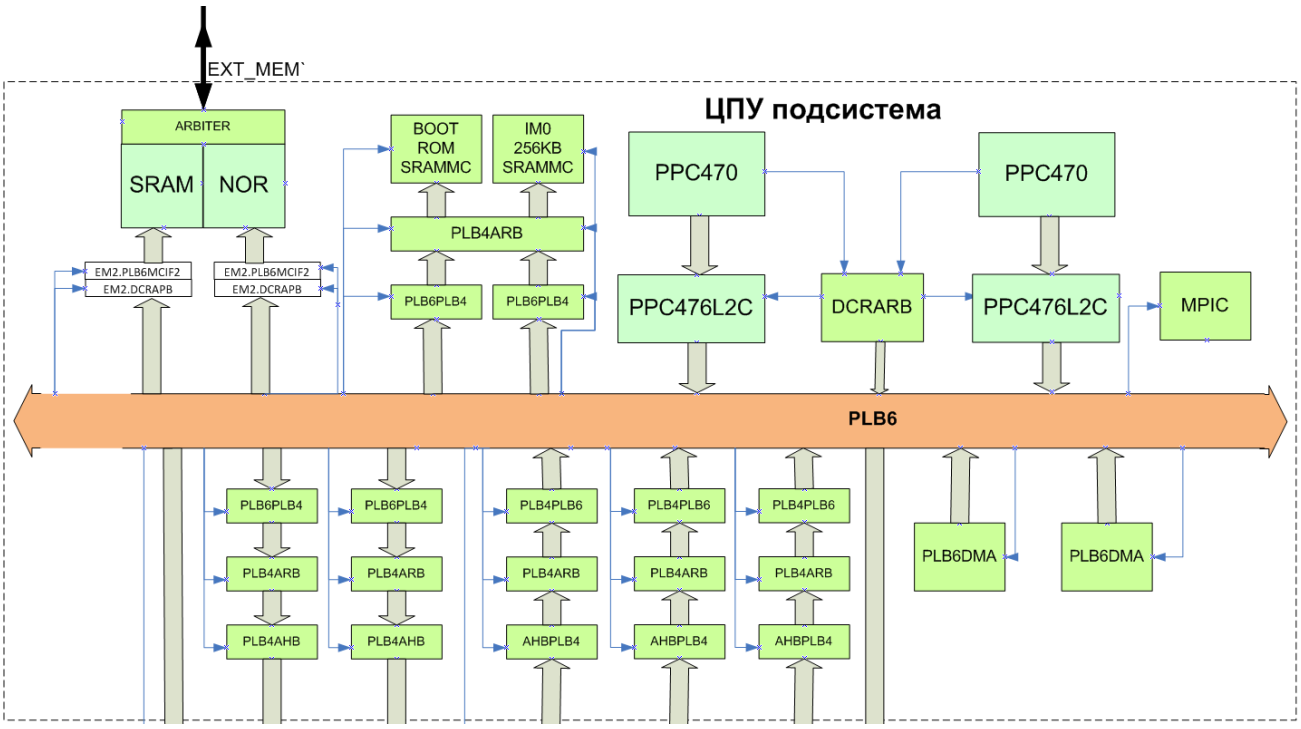


Рисунок 1.5 - Упрощенная структурная схема ЦПУ подсистемы

1.3.3.4 Детальная структурная схема ЦПУ подсистемы

Детальная структурная схема содержит имена и номера блоков, подключения блоков к коммутатору PLB6 и шинам DCR, источники прерываний для MPIC. Детальная структурная схема изображена в Приложении А.

1.3.3.5 Архитектура Power ISA

Ядра PowerPC 470S используют архитектуру Power ISA версии 2.05. Ядра PowerPC 470S являются 32-битными встраиваемыми (embedded) ядрами, поэтому к ним применимы общие разделы описания архитектуры и разделы для встраиваемых ядер.

1.3.3.6 Ядро процессора PowerPC 470S

См. Приложение Б.

1.3.3.7 Кэш инструкций и данных L2C

Обзор

Ядро L2C содержит кеш память второго уровня, которая управляет передачами между L1I кеш памятью команд первого уровня и L1D кеш памятью данных первого уровня одного процессора PowerPC 470S и процессорной локальной шиной PLB6. Кеш второго уровня L2C вместе с PLB6 образуют симметричную многопроцессорную систему (SMP) с когерентной памятью.

L2C является собственной кеш памятью для каждого ядра PowerPC 470S. На Рис. 2 приведен пример конфигурации двухъядерной системы.

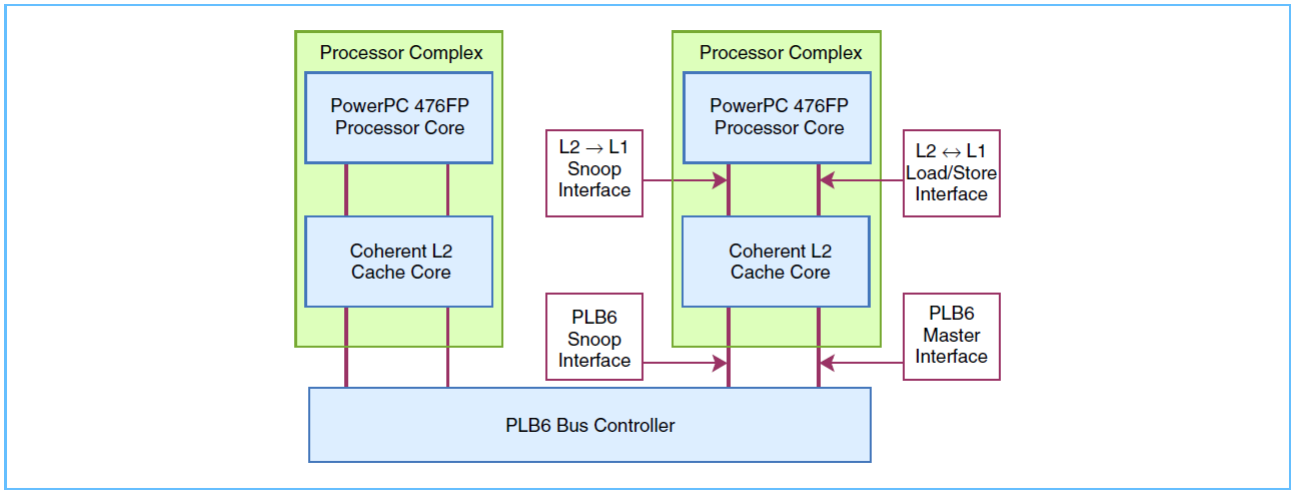


Рисунок 1.6 - Двухъядерная когерентная система на основе ядер PowerPC 470S/476FP

L2C поддерживает присутствие кеш памяти третьего уровня (L3) на шине коммутатора шины PLB6, которая в этом случае находится перед подсистемой памяти (например, DDR3 SDRAM). Присутствие L3 памяти прозрачно для L2 кеш контроллера.

Основные характеристики L2C.

Интерфейс к одному ядру PowerPC через пользовательский (custom) интерфейс. Поддержка соотношения частот PowerPC ядро к L2C 2:1, 3:1 и 4:1.

Размер кеш памяти L2C – 512 КБ

Размер кеш строки составляет 128 байт

Кеш память является 4-канальной множественно-ассоциативной (four-way set associative)

Поддержка протокола когерентности (MESI+T+SL+MU), состояния кеш строк – модифицированная/эксклюзивная/разделяемая/недостоверная, тегированная, последняя разделяемая, не запрашиваемая модифицированная (modified/exclusive/shared/invalid, tagged, shared last, modified unsolicited).

Включает режим работы обратной записи (write-back), сквозной записи (write-through, W=1), некешируемые обращения (I=1).

Все запросы могут обрабатываться как когерентные (M=1).

Включает очередь для Записей на 8 обращений. Поддерживает сбор одиночных Записей процессора в одно обращение размером до 64 байт. Не поддерживает сбор одиночных обращений процессора при пересечении границы в 64 байт.

Используется механизм замещения последней используемой кеш строки (LRU).

Механизм выделения кеш строк при Записи. Выделяется при промахе на операции Записи в память (store) для кешируемых обращений.

Контроль ошибок и их исправление. Исправление одиночных ошибок, обнаружение двойных ошибок на массивах памяти данных и тегов. Биты четности на битах включения (inclusion) и защелкивания (lock).

Унифицированная кеш память L2C для хранения как команд, так и данных.

L2C включает (inclusive) в себя данные кеш памяти первого уровня данных L1D. Включенная информация в границах данных 64 байт (две кещ строки L1D) поддерживается в L2C массиве памяти LRU для того, чтобы фильтровать Запросы Прослушивания (snoop traffic) к L1D.

Включение кода L1 кеш память команд в L2 кеш не навязывается. Инвалидация кеш строки L2C, которая также содержится в L1 кеш памяти команд, не приводит к инвалидации строки в L1 команд.

Сквозная (look-through) работа, нет параллельных запросов к L2C и на PLB6 интерфейс.

Не блокируемая кеш память.

Характеристики интерфейса между CPU L1 и L2C

Пакетные (burst) передачи для Записей и Чтений от CPU, размер кеш строки L1 – 32 байта, размер кеш строки L2 – 128 байт

Поддержка загрузки (load) 64 байт со стороны CPU

Пытается возвратить первым запрашиваемой четверное слово (quadword), затем оставшиеся четверные слова в CPU

Интерфейс со стороны L1 CPU: две ожидающих завершения операции Чтения (load) кода для L1 команд, 4 ожидающих завершения операции Чтения (load) данных для L1 данных, очередь Записи на 8 элементов с поддержкой объединения запросов (store gathering support).

Операции для инвалидации L1 кеш памяти по интерфейсу снупирования (snoop back):

на размере кеш строки L2 – 128 байт,

снупируются до 4 кеш строк L1 на каждый запрос снупирования, приходящий от контроллера L2C

раздельные интерфейсы снупирования для L1 команд, L1 данных и устройства управления памятью (MMU)

поддержка команд iKill, dKill и команд инвалидации буфера ассоциативной трансляции (TLB)

Характеристики PLB6 интерфейса

8 ожидающих (pending) запросов Чтения на PLB6 интерфейсе (включая Чтение-Модификацию-Запись [RWITM] и DCClaim)

Окно частичного ответа при снупировании Tsnoop, 7-8 тактов при соотношении частот (1:1 L2:PLB6), 5-8 тактов при соотношении частот (2:1 L2:PLB6)

42/64 адресных бит

Атрибуты памяти передаются на PLB6 интерфейс с CPU L1 интерфейса

Специальные команды для управления L2C

Синхронизация памяти (msync) управляемая для локального и удаленного процессора; легкая синхронизация (lwsync) и барьер памяти (mbar) для управления локальным процессором.

Поддержка инвалидации TLB (tlbivax/tlbie) и синхронизация TLB (tlbsync).

Поддержка нструкций для касания (touch) кеш строк (dcbt, dcbtst, icbt), закрытия (lock set) кеш строк (dcbtls, dcbtstls, icbtls), открытия (lock clear) кеш строк (dcblc, icblc).

Управления кеш операциями (dcbf, dcbi, dcbst, icbi).

Инвалидация кеш памяти и инициализация с использованием команды инвалидации кеш памяти (dci) или при включении питания.

Регистры кеш памяти и отладочные возможности доступны через DCR интерфейс.

Поддерживается только DCR ведомый (slave) интерфейс, доступ через DCR арбитр.

Управление резервированием памяти

L2C поддерживает управление резервированием памяти на размере кеш строки равной 128 байт.

1.3.3.8 Контроллер прямого доступа к памяти

Контроллер прямого доступа к памяти (DMA) является мастером на шине PLB6, который позволяет осуществлять быстрые передачи данных из памяти в память в отличие от пересылок под программным управлением процессора. Контроллер ПДП позволяет свободно исполнять команды без столкновений на шине, если процессор выполняет команды из кеш памяти. ПДП полезен, когда накладные расходы, связанные с настройкой ПДП контроллера минимальны, по

сравнению со временем, которое занимает пересылка данных процессором с помощью операций Чтения (load) и Записи (Store).

Контроллер ПДП управляет пересылками из памяти в память на шине PLB6. Контроллер имеет 4 канала. Каждый канал имеет независимый набор регистров, для программирования типа исполняемой пересылки.

Каждый канал содержит следующий набор регистров:

- Управляющий регистр канала (DMAx_CRn)
- Регистр счетчика (DMAx_CTCn)
- Регистры адреса источника (DMAx_SAHn и DMAx_SALn)
- Регистры адреса приемника (DMAx_DAHn и DMAx_DALn)
- Регистр шага (stride) (DMAx_DSTRn)

Каждый канал также имеет отдельный набор регистров для разброса/сбора (scatter/gather) данных:

- Регистры адреса дескриптора для разброса/сбора (DMAx_SGHn и DMAx_SGLn)
- Регистр команд для разброса/сбора (DMAx_SGCn)
- Регистр атрибутов команд для разброса/сбора (DMAx_SGCAn)
- Регистр для продолжения работы (resume) канала (DMAx_CHRESn)

Все каналы сообщают их статус в следующем наборе статусных регистров:

- Регистр статуса ПДП (DMAx_SR)
- Регистр адреса ошибки PLB мастера (DMAx_PMEAH и DMAx_PMEAL)
- Регистр ошибки PLB ведомого устройства (DMAx_PSE)
- Регистр адреса ошибки PLB ведомого устройства (DMAx_PSEAH и DMAx_PSEAL)

Регистр управления ПДП используется, чтобы инициализировать канал ПДП и разрешить ПДП передачи и прерывания. Биты [12:31] Регистра Счетчика загружаются количеством пересылок. Регистры адресов Источника и Приемника используются, чтобы определить расположение, откуда данные считываются и куда данные передаются. Регистр Шага ПДП определяет параметры, требуемые для выполнения пересылок с шагом. После того, как регистр ПДП был проинициализирован, канал включается в работу, и каналы ПДП контроллера начинают передавать данные. Регистры Статус и Ошибок разделяются между 4 каналами, каждый канал имеет свой собственный статус внутри регистра.

Регистры ПДП могут быть запрограммированы путем выдачи конфигурационных циклов через DCR ведомый порт или используя функцию разброса/сбора данных. Эта функция разрешает установку множества ПДП операций с помощью таблицы дескрипторов разброса/сбора в системной памяти.

Основные характеристики ПДП контроллера

- Передачи данных из памяти в память на шине PLB
- Один мастер порт на шине PLB6
- Конфигурируемый запрос и выделенная ширина данных (data bandwidth allocation)
- 64- битная адресация на шине PLB6
- Контроль четности на шине PLB6 (четность генерируется на стороне мастера, проверяется на стороне ведомого)
- 4 независимых ПДП канала
- Буфер на 256 байт для каждого канала
- Аппаратное выравнивание
- Одновременное выполнении Чтений и Записей на шине PLB6
- Поддержка операций интерливинга

Улучшенные возможности по разбросу/сбору данных; постановка команд в очередь на лету

Возможность обращений к данным с шагом

1.3.3.9 Многопроцессорный контроллер прерываний

Обзор

Многопроцессорный контроллер прерываний (MPIC) подключается к шине DCR в многопроцессорном окружении.

MPIC использует открытую архитектуру, разработанную совместно компаниями AMD и Cyrix для симметричных многопроцессорных систем, и реализует логику и программные структуры согласно данной спецификации со специальными расширениями и добавленными архитектурными особенностями. Документ, описывающий открытую архитектуру контроллера прерываний, называется “The Open Programmable Interrupt Controller (PIC) Register Interface Specification Revision 1.2”.

MPIC поддерживает до 128 индивидуальных источников прерываний с программируемыми приоритетами прерываний, назначением векторов прерываний и приемниками прерываний на каждый источник. В зависимости от приоритета прерывания источник прерывания может выдать на выходе MPIC прерывание Ошибка Аппаратуры (machine check), критическое прерывание, не критическое прерывание. Каждый источник прерывания может быть индивидуально замаскирован. Также маскирование прерываний может быть на основе приоритета текущей задачи (task priority) для каждого процессора. MPIC поддерживает многопроцессорное окружение (до 4-х ядер) путем поддержки межпроцессорных прерываний, прерываний самому себе, управление инициализацией процессора, выдачей распределенных прерываний и выдачей прямых прерываний так, как это требуется в принятом стандарте.

Основные характеристики MPIC.

- Подключение через DCR шину
- Использование стандарта PIC версии 1.2
- Поддержка от 1 до 4 процессорных ядер
- До 128 источников прерываний
- Встроенные средства для межпроцессорных прерываний и прерываний самому себе
- Четыре встроенных таймера, которые могут генерировать прерывания
- Программирование приоритетов прерываний со справедливостью (fairness)
- Программируемый приемник прерывания, маскирование прерываний, полярность, прерывание по фронту по уровню для каждого источника
- Расширенная поддержка для встраиваемых IBM Power PC процессоров – Ошибка Аппаратуры (Machine Check), Критические, Не критические прерывания для каждого процессора
- Программируемый приоритет задач для каждого процессора
- Базисный и улучшенный метод выдачи прерываний: прямой, широковещательный, распределенный
- Поддержка вложенных (nested) и ложных (spurious) прерываний
- Механизм, сообщающий количество процессоров и поддерживаемых источников прерываний
- Управление инициализацией процессора
- Все регистры 32 битные и выровнены по границе 128 бит для 32/64/128 битных операций на шине
- Блок регистров выровнен по границе 4К
- Поддержка отладки для считывания необработанных прерываний

Рис. 3 показывает использование MPIC в многопроцессорной СнК, где все процессоры имеют доступ к MPIC через разделяемую с помощью арбитра шину DCR.

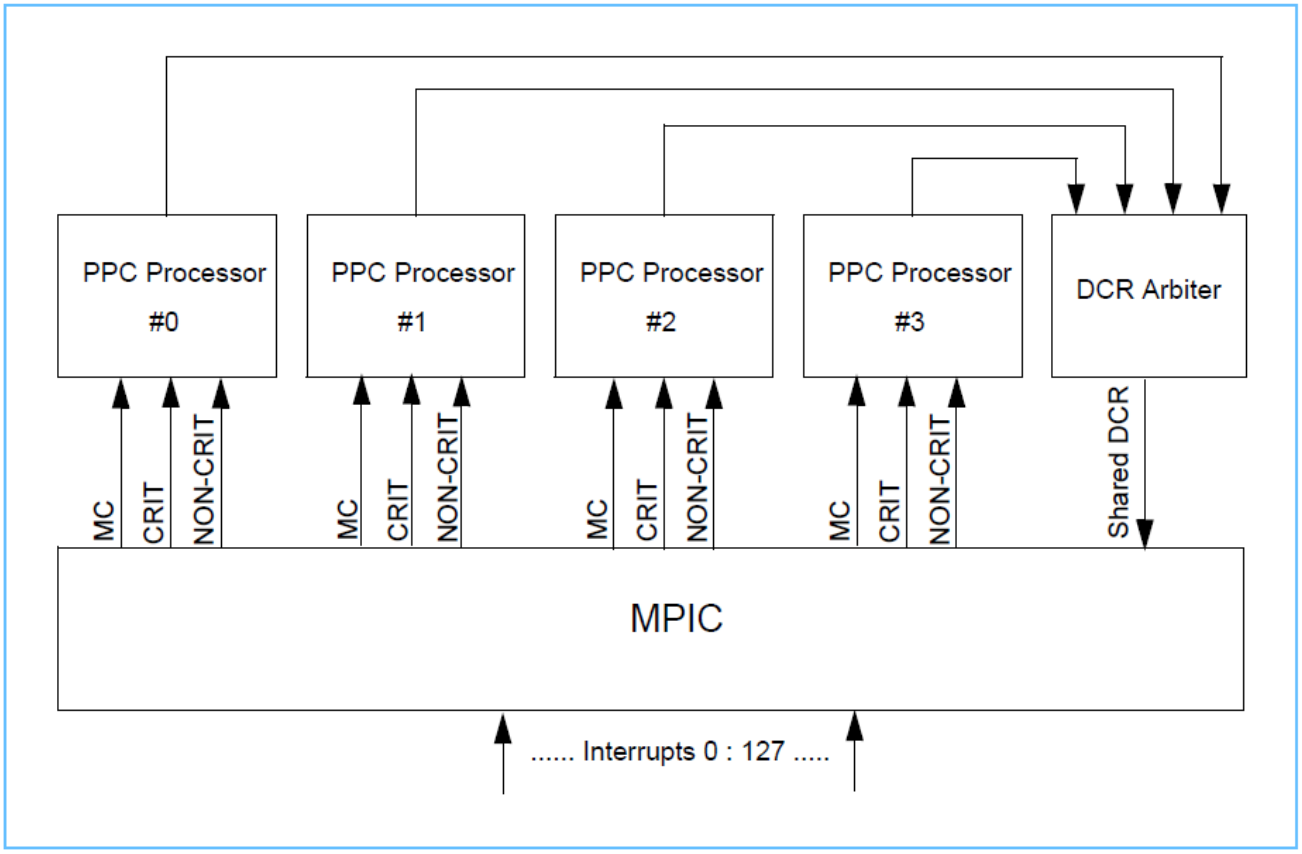


Рисунок 1.7 - Использование MPIC в многопроцессорной системе

На рисунке 3 также показано, что MPIC управляет сигналами Ошибка Аппаратуры (Machine check), критические прерывания и не критические прерывания.

Внутренними прерываниями для MPIC являются прерывания таймера и межпроцессорные прерывания (IPI), MPIC также принимает 128 внешних прерываний. Внешние к MPIC прерывания приходят как от внутренних источников прерываний СнК, так и внешних по отношению к СнК источников прерываний. Внешние к СнК источники прерываний мультиплексированы с входами-выходами общего назначения (GPIO). Прерывания, внешние по отношению к MPIC, могут быть по фронту или по уровню. Прерывание по фронту может быть утеряно, если оно приходит, когда предыдущее прерывание от того же источника находится в обслуживании.

MPIC использует две модели выдачи прерываний: прямая и распределенная. Для прямого режима прерывание будет доставлено в определенный процессор (один приемник) или к множеству выделенных процессоров (широковещательный режим). Для внутренних таймеров и межпроцессорных прерываний поддерживаются оба эти режима.

Для распределенного режима события прерываний из одного источника могут быть распределены среди определенной группы процессоров. Т.е. из специального пула процессоров, назначенных определенному источнику, MPIC выберет наиболее свободного кандидата в справедливой манере без выделения преимуществ какому-нибудь определенному процессору. 128 внешних источников прерываний поддерживаются прямым режимом для назначения одному процессору, если назначен только один процессор, или распределенным режимом, если назначено более одного процессора.

Прерывания в распределенном режиме могут быть доставлены “точно один раз”. Это означает, что прерывание никогда не будет находиться в ожидании или в активном состоянии более, чем к одному процессору в данный момент времени и будет доставлено только один раз на выбранный приемник процессор. Также, после того, как прерывание отправлено

определенному приемнику (некоторый процессор был извещен посредством выставления сигнала на выходе MPIC), дальнейшие события от этого источника не будут отправляться к любому другому процессору, до тех пор, пока обработка исходного события не завершиться прописью бита окончания прерывания (EOI).

Каждое прерывание имеет программируемое значение приоритета в диапазоне от 0 до 15, и каждый процессор имеет программируемый приоритет задачи также в диапазоне от 0 до 15. Прерывания, назначенные одному и тому же процессору, соперничают друг с другом на основе соответствующего программного приоритета. Однако, процессор не будет прерываться, если выигранный приоритет будет больше, чем назначенный приоритет задачи.

MPIC может выставить один из трех типов прерываний процессору. В восходящем порядке приоритета они следующие: не критические прерывания, критические прерывания, ошибка аппаратуры (machine check). Связь между приоритетом прерывания и типом прерывания программируется. Например, все приоритеты прерываний могут оставаться назначенными для генерации не критических прерываний или приоритеты прерываний с 10 по 14 могут быть назначены для генерации критических прерываний и приоритет прерываний 15 может быть назначен для генерации прерывания Ошибка Аппаратуры (Machine Check).

Общий регистр для инициализации процессора поставляется в контроллере, который может быть использован любым процессором для выставления входа инициализации для одного или нескольких других процессоров. Это особенность дает способ стартовать нескольким процессорам и для обеспечения некоторых диагностических возможностей.

Доступ процессора к MPIC осуществляется с помощью DCR шины. Соперничество за доступ к одному MPIC среди нескольких процессоров разрешается с помощью внешнего DCR арбитра. Микросхеме, содержащей один процессор, не обязательно требуется использовать арбитр шины DCR, если процессор использует полный 32 битный DCR адрес. Микросхема, содержащая только один процессор, будет все еще требовать DCR арбитр, если процессор использует 10-битный адрес или, если считается желательным ловить DCR таймауты из-за неверного адреса на шине DCR.

MPIC включает в себя регистр “Кто я” (Who Am I), который используется для индивидуальной идентификации процессора. Этот регистр располагается по одному и тому же DCR адресу для каждого процессора, но возвращает уникальное значение, которое зависит от того, какой процессор произвел Чтение. Этот регистр предназначен для использования, чтобы идентифицировать физическое подключение соединения процессора к MPIC так, чтобы правильные биты поля приемника были использованы для идентификации процессора при прерываниях от устройств ввода-вывода и межпроцессорных прерываний. Значения регистра “Кто я” не обязательно должно быть таким, как идентификатор (ID) логического процессора, назначенного операционной системой для межпроцессорного обмена, или такой же, как 4-битный сигнал DCR_MASTERID, приходящий от процессора. Содержимое регистра “Кто я ” определяется 2-битным входом DCR_MASTERID для MPIC. В данной СнК для Процессора 0 используется значение 0 на входе DCR_MASTERID, для Процессора 1 используется значение 1 на входе DCR_MASTERID.

Открытая спецификация PIC определяет механизм, чтобы обеспечить совместимость чипсетов для персональных компьютеров (PC-AT) при включении питания, которые используют архитектуру контроллера прерываний 8259. После сброса, вызванного включением питания, MPIC будет по умолчанию находиться в сквозном (pass-through) режиме 8259. В этом режиме выход запроса прерывания 8259 будет проходить напрямую через MPIC к входной линии запроса прерываний процессора, и MPIC будет в какой-то момент закрыт. Во время операций симметричной многопроцессорной системы (SMP) режим сквозных прерываний будет закрыт, и MPIC будет распределять все события системных прерываний.

1.3.3.10 Контроллер шины PLB6

Обзор

Контроллер шины PLB6 разрешает до 16 мастерам подключаться к 8 ведомым (slave) сегментам, до 4 устройств на каждый сегмент. Контроллер проводит арбитраж и маршрутизацию (routing) между ведущими и ведомыми устройствами.

Контроллер шины PLB6 обеспечивает арбитраж следующих запросов:

- Запросы ведущего устройства (master), чтобы представить их на правильном сегменте ведомого устройства (slave)
- Запросы ведомого устройства при Чтении данных, чтобы представить считываемые данные требующему данных ведущему устройству
- Запросы ведущего устройства при Записи данных, чтобы представить записываемые данные корректному ведомому устройству

Контроллер шины PLB6 поддерживает когерентных мастеров, которые могут принимать участие в аппаратно реализованном протоколе когерентности кеш памяти.

Основные характеристики контроллера шины PLB6

- Поддержка до 8 когерентных мастеров PLB6 (интерфейс снупирования), когерентные мастера подключаются к контроллеру шины PLB6 на интерфейсах с номерами от 0 до 7.
- Поддержка всего до 16 мастеров
- Поддержка до 8 сегментов ведомых устройств: до 4 ведомых устройств на каждый сегмент, сегмент с номером 0 – это единственный сегмент, способный управлять когерентной памятью, сегмент с номером 0 конфигурируется, как некогерентный, если проект сконфигурирован, как не когерентный
- Обеспечивает интерфейс шины DCR для конфигурирования устройства и доступа к статусным регистрам, включая 7 регистров, определенных в спецификации шины PLB6
- Поддержка аппаратной интервенции данных в протоколе кеш когерентности, где кеш память поставляет данные для запроса Чтения, при попадании запроса прослушивания (snoop) в кеш память
- Высокая производительность: две одновременно работающие 128-битные шины Чтения данных и Записи Данных, конвейеризация Запросов, Чтение данных с изменением порядка (out-of-order)
- Аппаратная поддержка кеш когерентности: поддержка когерентности симметричной многопроцессорной системы (SMP) и когерентности ввода-вывода, до 7 состояний кеш строки, для поддержки сложной интервенции данных, функция соответствия предыдущего выровненного адреса (PAAM)
- 16 конфигурационных и статусных регистров, доступных по шине DCR
- Обнаружение и разрешение подвисания шины
- В границах контроллера команды одного мастера не обгоняют друг друга
- Запросы барьеров памяти (MBar) и синхронизации памяти (MSync)
- Предотвращение столкновений комбинированных ответов (combined response)
- Арбитраж шины для фаз команд и данных: команды ведущего, считываемые данные ведомого, Чтение данных сегмента, Запись данных от ведущего

Типовая схема включения контроллера шины PLB6 изображена на Рис. 4.

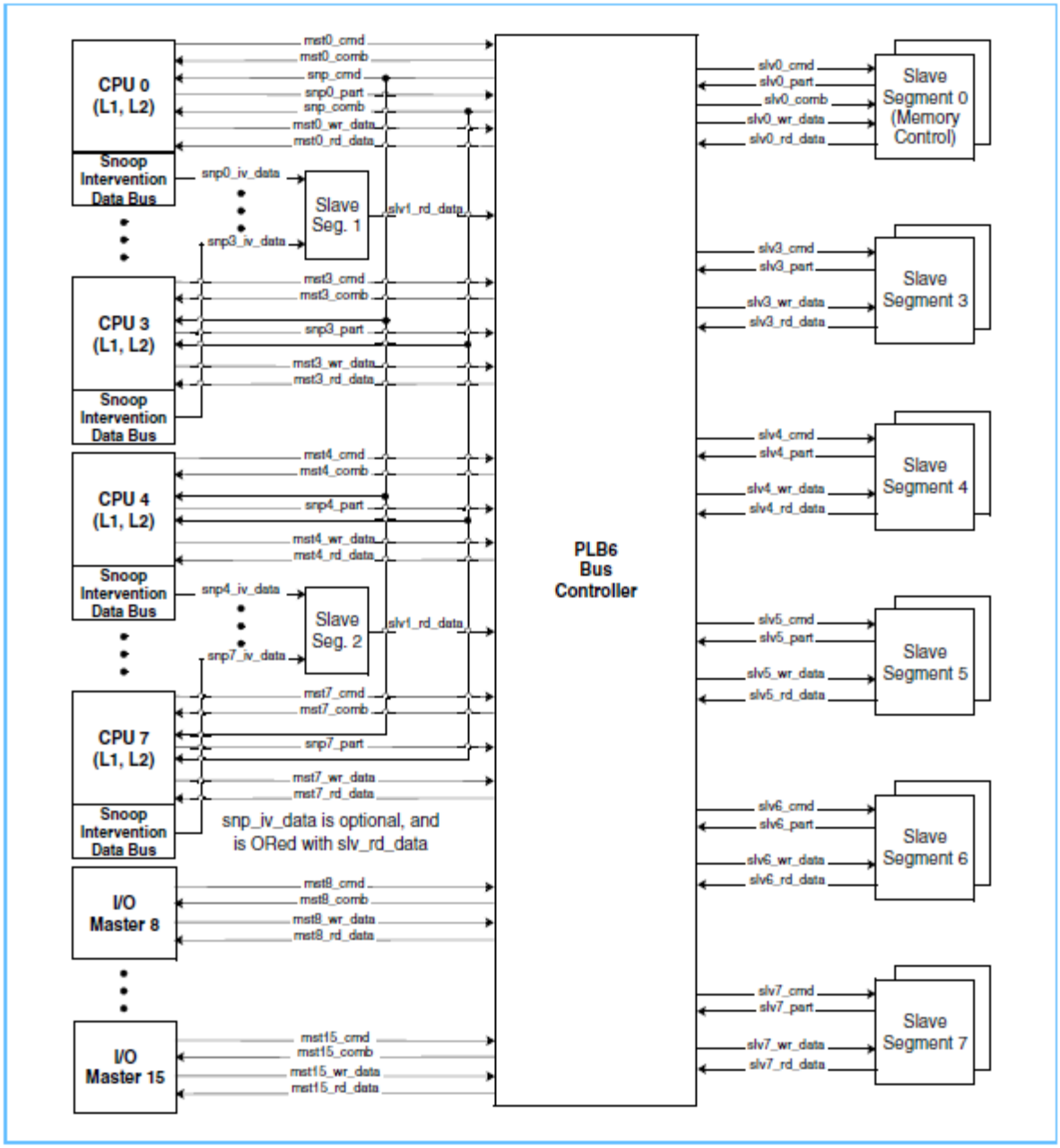


Рисунок 1.8 - Типовая схема включения контроллера шины PLB6

На рисунке 4 PLB6 коммутатор шины связывает 8 когерентных мастеров, процессорные ядра и L2C кеш памяти с номерами от 0 до 7, устройства ввода-вывода с номерами от 8 до 16, сегменты для подключения ведомых устройств с номерами от 0 до 7. Сегмент с номером 0 используется для подключения основной когерентной памяти, сегмент с номером 1 используется для интервенции данных от когерентных мастеров с номерами от 0 до 3, сегмент с номером 2 используется для интервенции данных от когерентных мастеров с номерами от 4 до 7. В текущей СнК используются два ядра PowerPC 470S вместе с кеш памятями L2C, подключаемые как CPU 0 и CPU 1 к интерфейсам ведущих устройств. Интервенция данных происходит на Сегменте 1. Сегмент 2 используется для подключения памяти BOOT ROM, SRAM и мостов PLB6PLB4, которые предоставляют доступ в адресное пространство AXI подсистемы. Другие сегменты не используются. Мастера, которые используют интерфейсы снупирования, имеют независимые интерфейсы данных для подключения к контроллеру PLB6. Интервенция данных при Чтении поддерживается только L2C кеш памятью, которая определяет, что имеет эксклюзивную кеш строку так, что никакое другое ведомое устройство или другая кеш память не будет поставлять

данные. Установка L2C с интерфейсом интервенции данных по снупированию на один из Сегментов ведомых устройств позволяет L2C принимать участие в арбитраже за одну из шин Чтения данных. Шина интервенции данных функционально схожа с любой другой шиной Чтения данных для контроллера шины PLB6.

PLB6 контроллер шины определяет раздельные интерфейсы для ведущих устройств, ведомых устройств и прослушивателей (snoopers). Устройства подключаются к контроллеру, используя следующие интерфейсы:

- Ведущие устройства ввода-вывода подключаются интерфейсу ведущего устройства
- Ведомые устройства ввода-вывода подключаются интерфейсу ведомого устройства (за исключением Сегмента 0, который в когерентных системах используется только для подключения памяти)
- Когерентные ведомые устройства, такие как контроллер памяти, подключаются к Сегменту 0.
- Когерентные процессоры и L2C могут быть подключены к следующим интерфейсам: интерфейс ведущего устройства, интерфейс снупирования, Интерфейс Чтения данных для интервенции данных.

1.3.3.11 Мост с шины PLB6 на шину PLB4

Обзор

Мост с шины PLB6 на шину PLB4 является мостом из некогерентного PLB6 порта ведущего устройства контроллера шины PLB6 на ведущий порт PLB4 контроллера шины PLB4. Мост PLB6-PLB4, и арбитр шины PLB4 позволяют PLB4 ведомым устройствам использоваться в системе на основе шины PLB6.

Основные характеристики моста PLB6-PLB4

- Поддержка выполнения конвейерных операций Чтения и Записи как на стороне PLB6 ведомого устройства, так на стороне PLB4 ведущего устройства.
- Поддержка отложенных Записей
- Поддержка Переповторов (retries), когда буфер или очередь недоступны.
- Обеспечивает интерфейс DCR шины для доступа к конфигурационным и статусным регистрам
- Отношение частот PLB6 к PLB4 N:1 (N>1), синхросигналы синхронные, соотношения 2:1, 3:1, 4:1 верифицированы
- Одновременная работа 128-битных шин Записи данных и Чтения Данных
- PLB6 ведомая сторона
- Конвейер до 4 Запросов Чтения и до 3 запросов Записи
- Поддержка раннего завершения транзакций
- Выдача требуемого слова первым при запросе строки
- Изменение порядка запросов Чтения (out-of-order) в некоторых случаях
- Поддержка соединения множественных данных для операций Чтения и Записи
- Буфер данных
- Интерфейс к внешнему массиву регистров/памяти (register array)
- Верифицированы две конфигурации буферов: большая и малая, в СнК используется большая конфигурация: 4 буфера Чтения и 3 буфера Записи
- PLB4 ведущая сторона
- спроектирована в соответствие со 128-битной спецификацией шины PLB4 без поддержки двойной скорости (double data rate)
- конвейеризация до 4-х запросов Чтения и до 3-х запросов Записи
- поддержка только 128-битных ведомых устройств

Интерфейс к внешнему массиву регистров/памяти

Типовая схема использования моста PLB6-PLB4 показана на Рис. 5.

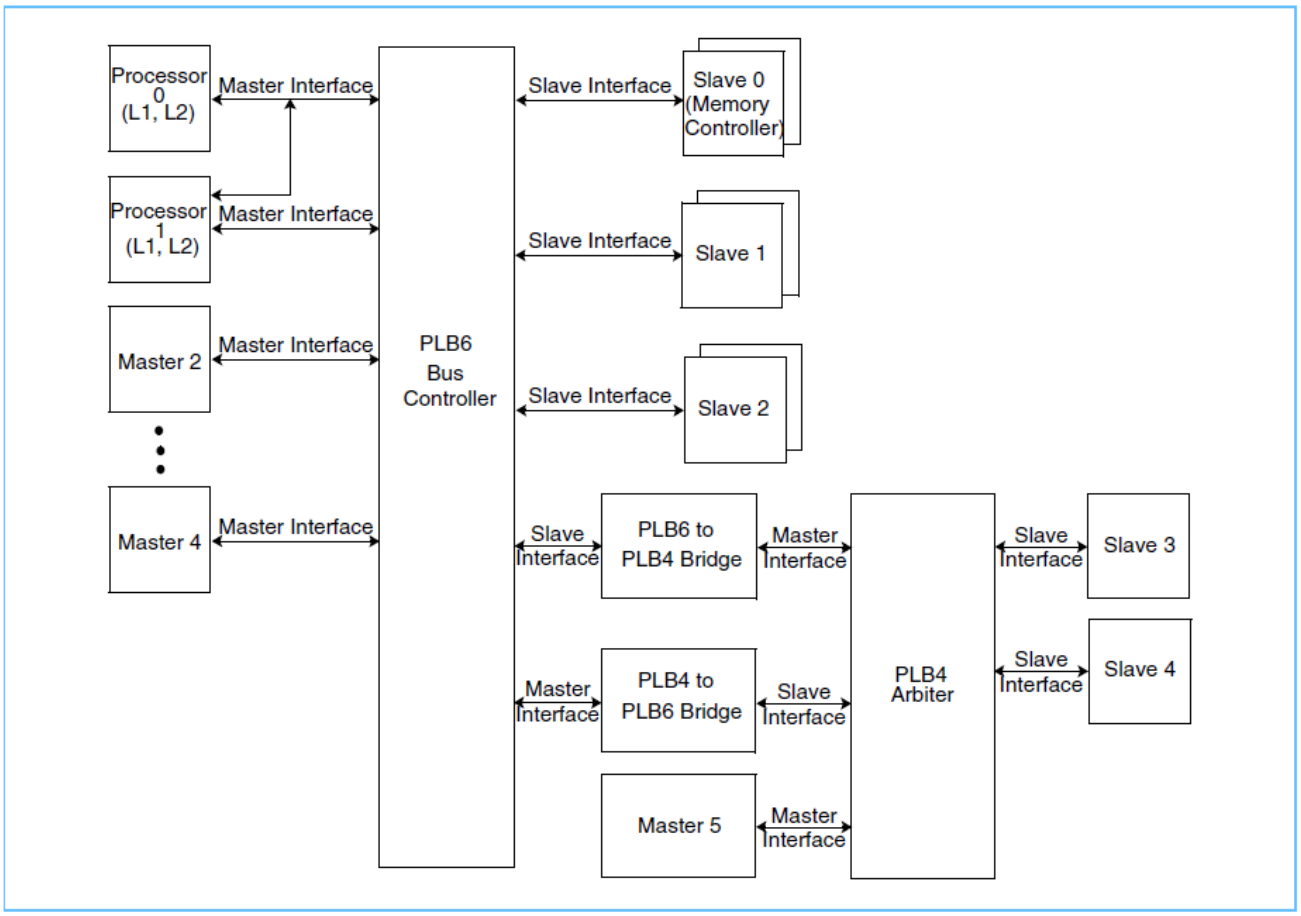


Рисунок 1.9 - Типовая схема использования моста PLB6-PLB4

На рисунке показано, что мост PLB6-PLB4 подключается между Контроллером шины PLB6 и арбитром шины PLB4. Мост PLB6-PLB4 подключает некогерентный интерфейс ведущего устройства контроллера шины PLB6 и интерфейс ведущего устройства арбитра PLB4. Мост разрешает использование PLB4 ведомого устройства в PLB6 системе.

1.3.3.12 Мост с шины PLB4 на шину PLB6

Обзор

Мост с шины PLB4 на шину PLB6 является мостом из PLB4 порта ведущего устройства на ведомый порт PLB6 контроллера шины PLB6. Мост PLB4-PLB и контроллер шины PLB6 позволяют PLB4 мастерам связываться с ведомыми устройствами на шине PLB6.

Основные характеристики моста PLB4-PLB6

- Поддержка конвейеризации операций Чтения и Записи как на стороне PLB6 ведущего устройства, так и на стороне PLB4 ведомого устройства
- Поддержка отложенных Записей
- Поддержка переарбитража, когда буфер и очередь заняты
- Интерфейс DCR шины к конфигурационным и статусным регистрам
- Отношение частот PLB6 к PLB4 N:1 (N>1), синхросигналы синхронные, соотношения 2:1, 3:1, 4:1 верифицированы
- Одновременная работа 128-битных шин Записи данных и Чтения Данных
- PLB64 ведомая сторона
- Разработана в соответствии со Спецификацией 128-битной шины PLB4 без поддержки удвоенной скорости и совместимая со Спецификацией Взаимодействия PLB4 (Interoperability PLB4)
- 128-битная шина данных

- Подтверждение до 4 запросов Чтения и до 3 запросов Записи
- Буфер данных
- Интерфейс к внешнему массиву регистров/памяти (register array), массив в большой конфигурации хранит до 24 четверных слов данных Записи и до 32 четверных слов данных Записи
- Сторона PLB6 ведущего устройства:
- Соответствуют спецификации шины PLB6
- Некогерентный интерфейс PLB6 ведущего устройства
- Конвейер до 4 запросов Чтения (в режиме подтверждения адреса) и до 3 запросов Записи
- Принимает данные с изменением порядка (out-of-order)
- Поддерживает PLB6 ведомое устройство, объединяющее данные от многих запросов

Типовая схема использования моста PLB4-PLB6 изображена на Рис. 5.

Мост PLB4-PLB6 используется как мост между PLB4 арбитром и Контроллером шины PLB6. Мост разрешает использование PLB4 мастера в PLB6 системе.

1.3.3.13 Двунаправленный мост между шинами PLB4 и AHB

Обзор

Двунаправленный мост между шинами PLB4 и AHB позволяет транзакциям переходить с шины PLB4 на шину AHB и обратно. Мост состоит из двух компонентов: мост с шины AHB на шину PLB4, который имеет интерфейс ведомого на шине AHB и интерфейс ведущего на шине PLB4, и мост с шины PLB4 на шину AHB, который имеет интерфейс ведомого на шине PLB4 и интерфейс ведущего на шине AHB.

Мост имеет интерфейс DCR шины для доступа к управляющим и статусным регистрам.

Основные характеристики моста PLB4-AHB.

Интерфейс AHB ведомого устройства
- Ширина шины данных AHB – 128 бит
- Поддерживает все типы передач (HTRANS) - IDLE, BUSY, SEQ, NONSEQ
- Поддерживает все типы пакетных передач (HBURST) для транзакций Чтения и Записи
- Одиночные передачи
- Пакетные передачи с инкрементом адресов (INCR4, INCR8, INCR16)
- Пакетные передачи со сворачиванием адресов (WRAP4, WRAP8, WRAP16)
- Пакетные передачи переменной длины (INCR)
- Поддержка всех размеров передач (HSIZE) – байт, полуслово, слово, двойное слово, четверное слово
- Выдача всех типов ответов (HRESP) – SPLIT, RETRY, OKAY, ERROR
- Работа в системах как многих AHB мастеров, так и одного AHB мастера
- Поддержка досрочного завершения пакетной передачи
- Поддержка атомарных транзакций (locked transactions)

Интерфейс AHB ведущего устройства
- Ширина шины данных AHB – 128 бит
- Поддерживает типы передач (HTRANS) - IDLE, SEQ, NONSEQ
- Поддерживает все типы пакетных передач (HBURST) для транзакций Чтения и Записи
- Одиночные передачи
- Пакетные передачи с инкрементом адресов (INCR4, INCR8, INCR16)
- Пакетные передачи со сворачиванием адресов (WRAP4, WRAP8, WRAP16)
- Пакетные передачи переменной длины (INCR)
- Поддержка всех размеров передач (HSIZE) – байт, полуслово, слово, двойное слово, четверное слово
- Поддерживает все типы ответов (HRESP) – SPLIT, RETRY, OKAY, ERROR
- Работа в системах как многих AHB мастеров, так и одного AHB мастера

- Поддерживает оба типа порядка слов в байте (endian) – большой (big) и маленький (endian). Задается статически. В мостах PLB4-AHB, AHB-PLB4 выбран большой (big) порядок байт.

- Атрибут шины PLB4 (PLB_TAttribute[8]) напрямую транслируется в атрибут шины AHB HPTOT[2]. Когда бит, задающий упорядоченную передачу, установлен в цикле PLB шины, то HPROT[2] бит установлен в 0 (т.е. не буферизированная передача) и противоположное. Этот атрибут рассчитан для передачи только AHB ведомым устройствам. Мост всегда выполняет транзакции в порядке поступления, дополнительно мост имеет управляющий бит в регистре CSR, который заставляет всегда устанавливать HPROT[2] в 0 для всех транзакций, независимо от состояния бита PLB_TAttribute[8].

- Мост не поддерживает циклы BUSY на шине AHB.

Интерфейс PLB ведущего устройства

- Ширина шины данных PLB – 128 бит

- Поддержка одиночных и пакетных передач фиксированной длины

-Конвейеризация адресов допускается для Чтений, следующих за Записями, или Записей, следующими за Чтениями буферизированными передачами

- Поддержка досрочного завершения транзакций со стороны PLB4 ведомых устройств

- Мост записывает детали транзакций в статусный регистр для последующих сценариев разбора ошибок. В регистре CSR запоминается PLB адрес, в котором встретилась ошибка.

- Ошибки PLB таймаута сообщаются сигналами ошибок: PLBMWrErr, PLBMRdErr

- Следующие PLB возможности не реализованы на стороне PLB4 ведущего устройства:

- Не поддерживается сигнал Mn_TAttribte

- Не поддерживается генерация бит четности или их контроль

- Не поддерживаются пакетные передачи переменной длины

- Не поддерживается передачи строк (line)

- Не поддерживается конвейер для Записей, следующих за Записями, или для Чтений, следующих за Чтениями

- Не поддерживается динамическое изменение размера шины данных 32 бита, 64 бита

Интерфейс PLB ведомого устройства

- Поддерживает 64 битных и 128 битных PLB4 мастеров

- Поддержка валидности байт (byte enable) для индикации непрерывных порций (lane) байтовых данных во время транзакций Записи

- Раздельные шины данных для Чтения и Записи. Способность перекрывать транзакции Чтения и Записи.

- Поддержка всех передач PLB за исключением конвейеризации адресов, как для Записи, так и для Чтения

- Одиночные передачи

- Пакетные передачи переменной длины из одной или более порций данных (beats)

- Пакетные передачи фиксированной длины из одной или более порций данных (beats)

- Передача строк (line) размером 4, 8, 16 слов. Всегда целевое слово возвращается первым.

- Операции Записи выставляются PLB мастером. Мост использует буфер (FIFO) для постановки в очередь данных Записи на сторону AHB.

- Поддержка прекращений (abort) передач со стороны PLB мастера либо в тот же самый такт, когда Запрос был подтвержден, или перед подтверждением Запроса со стороны моста.

- Поддержка переарбитража Запроса, если мост не может выполнить текущий Запрос.

- Поддержка 36 битной адресации на шине PLB, используются PLB_ABus[0:31] и PLB_UABus[28:31]. Минимальное пространство AHB – 256 МБ. Максимальное пространство AHB – 64 ГБ.

- Поддержка управления ошибками транзакций через сигналы SL_MRdErr и SL_MIRQ для соответственно транзакций Чтения и Записи, регистры SEAR и SESR доступны через DCR интерфейс. Сигнал SL_MWrErr не поддерживается.
- Следующие особенности PLB не поддерживаются в направлении от PLB4 к AHB:
- Конвейеризация адресных Запросов
- Динамическое изменении размера шины данных
- Любые типы передач PLB, которые не направлены в память

Типовая схема использования двунаправленного моста PLB4XAHB изображена на Рис. 6.

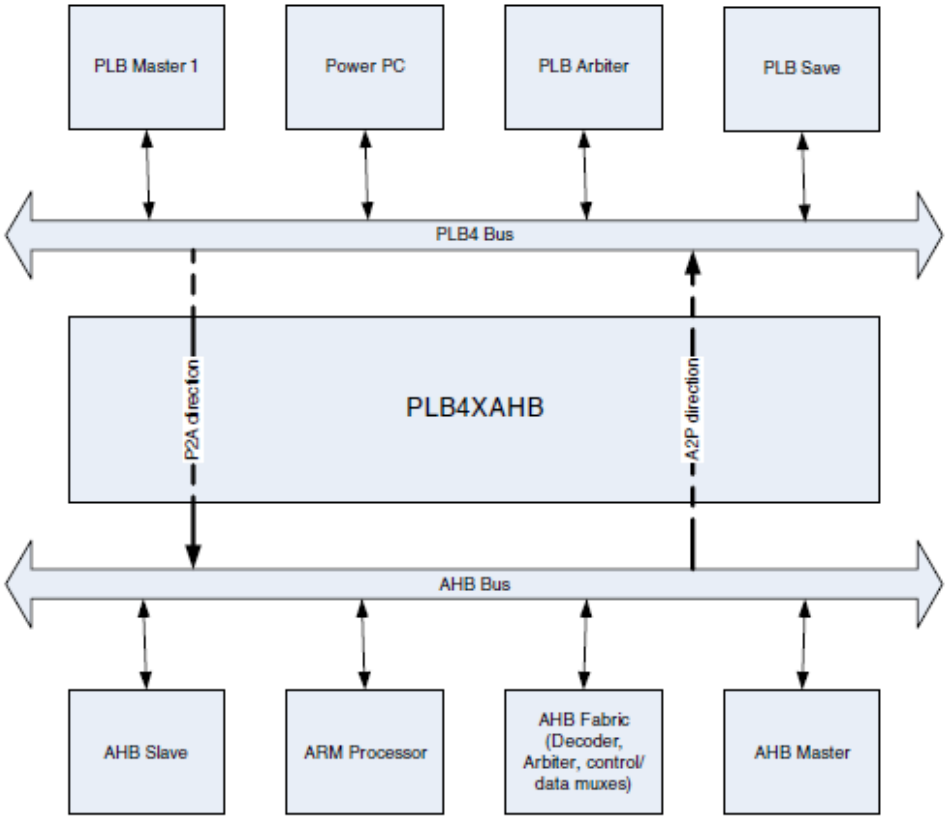


Рисунок 1.10 - Арбитр шины PLB4

Обзор

Высокопроизводительный арбитр шины PLB4 реализует 3-з тактовый протокол подтверждения. Ядро включает в себя управляющее устройство арбитра, сторожевой таймер, модуль адреса, данных Записи и данных Чтения.

Основные характеристики арбитра PLB4

- Поддержка до 8-х ведущих устройств
- Поддержка 128, 64, 32 битных ведущих и ведомых устройств
- Полная 64 битная шина адреса
- 16-битная шина атрибутов
- Программируемый конвейер адресов Чтения до 4-х Запросов
- Программируемый конвейер адресов Записи до 2-х Запросов
- Поддержка до 8 конвейерных ведомых устройств
- Используется 2-х тактовая схема арбитража
- Четыре уровня динамически изменяемого приоритета мастера
- Схемы фиксированного или циклически изменяемого приоритета (rotating), устанавливаются через входы или программируется

- Сторожевой таймер PLB
- Высокочастотный 3-х тактовый временной протокол подтверждения
- Совместим с версией PLB4

1.3.3.14 Контроллер внутренней SRAM памяти

Обзор

Контроллер внутренней SRAM памяти передает данные между шиной PLB4 и встроенной памятью. Контроллер поддерживает 4 банка физической памяти, каждый банк сконфигурирован на 4К слов размером по 128 бит (144 бита включая биты четности).

Контроллер имеет DCR интерфейс для доступа к внутренним регистрам и настройки атрибутом каждого банка памяти.

Основные характеристики контроллера внутренней SRAM памяти

- Поддержка до 4-х неперекрывающихся банков памяти, конфигурируемых по 256, 512, 1К, 2К, 4К или 8К слов по 128 бит (144 бита, включая контроль четности)
- 128 битное ведомое устройство, адресуемое любым 64 битным или 128 битным мастером
- Поддерживаются следующие циклы PLB4 ведомого устройства:
- От 1 до 8 байт одиночные передачи для 64 битных мастеров
- От 1 до 16 байт одиночные передачи для 128 битных мастеров
- Передачи строк из 4 слов
- Передачи строк из 8 слов
- Пакетные передачи фиксированной длины размеры данных двойное слово и четверное слово, завершаемые ведомым устройством
- Пакетные передачи переменной длины размеры данных двойное слово и четверное слово, завершаемые мастером
- Охраняемый (guarded, G - бит) доступ к памяти на границе 4 К
- Один синхросигнал PLB
- Управление потребляемой мощностью
- Функциональное селективное отключение синхросигнала (clock gating)
- Спящий режим
- Обнаружение ошибок с помощью генерация и контроля четности

В СнК используются два контроллера внутренней SRAM памяти один для доступа к BOOT ROM, другой для доступа к SRAM (IM0).

Типовая схема использования контроллера внутренней SRAM памяти приведена на рисунке.

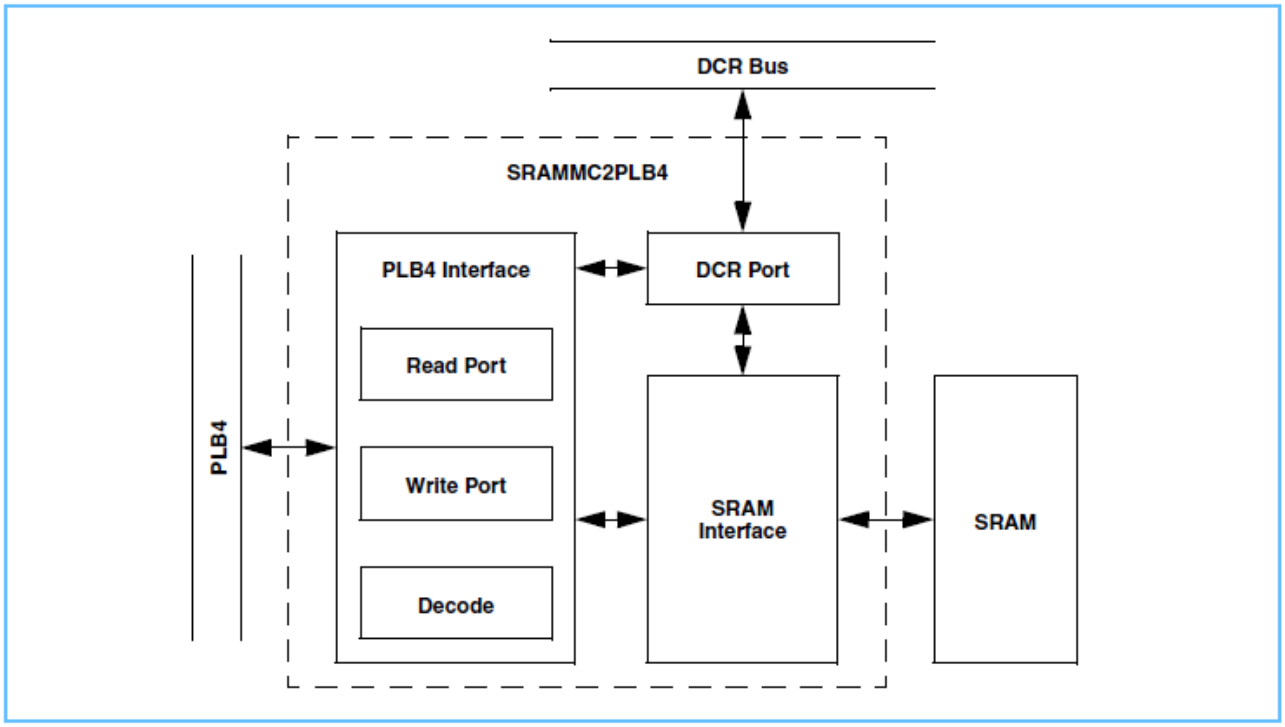


Рисунок 1.11 - Типовая схема использования контроллера внутренней SRAM памяти.

1.3.3.15 Арбитр шины DCR

Обзор

Арбитр шина DCR разработан для арбитража до 4-х DCR мастеров для доступа к общей (shared) DCR шине. Основное использование DCR арбитра в многопроцессорных системах, где всем DCR мастерам в системе требуется доступ к общему набору DCR конфигурационных регистров.

Основные характеристики арбитра шины DCR

- Совместим со спецификацией DCR версии 3.5
- 4 независимых порта для подключения DCR мастеров
- 1 общий порт DCR шины
- Поддержка 10 битной и 32 битной адресации
- 32 битная DCR шина данных
- Схемы арбитража с циклически-изменяемым (round-robin) приоритетом и фиксированным приоритетом
- Поддержка “привилегированных” и “непривилегированных” передач. Приоритет при арбитраже дается привилегированным передачам
- Программируемая трансляция DCR адреса (1 на каждый порт мастера)
- Программируемый счетчик таймаута для DCR передачи с обнаружением ошибки в случае истечения времени счетчика
- Одиночный регистр семафора для программно-управляемого захватывания (locking) DCR ресурса
- Каскадируемое включение, если требуется использовать более 4 DCR мастеров в системе
- Конфигурируемая область легальных (legacy) DCR адресов с привилегированной защитой
- Использование сигнала задержки таймаута (TimeoutWait) на порт мастера при арбитраже
- Поддержка использования сигнала задержки таймаута (TimeoutWait) на общей DCR шине так, что медленные ведомые устройства могут удлинить циклы на DCR шине
- 4-х битовый идентификатор мастера (MasterID) для общей DCR шины
- Работа от одного синхросигнала DCR

- Поддержка возможности использовать персональные синхросигналы для каждого мастера
- Поддержка как последовательного подключения (daisy-chain), так и распределенного (distributed-OR) типа DCR конфигураций
- 6 тактовый минимальный цикл Чтения или Записи, расширяемый ведущим или ведомым устройствами
- Селективное отключение синхросигнала (clock gating) для снижения потребляемой мощности

На Рис. 7 приведена типовая схема использования арбитра шины DCR

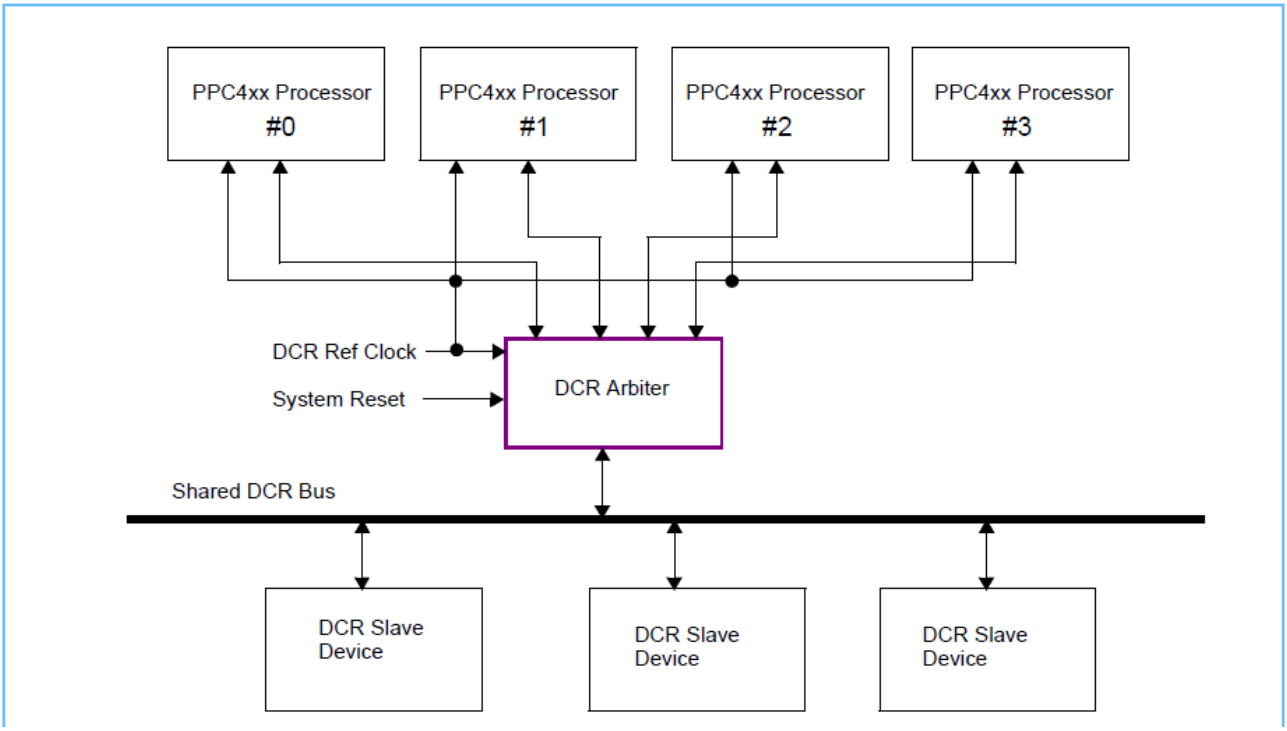


Рисунок 1.12 - Типовая схема использования арбитра шины DCR

1.3.3.16 Контроллер внешней динамической памяти DDR3

DDR3 подсистема предназначена для обеспечения доступа изделия во внешнюю память типа DDR3. Данная подсистема состоит из набора блоков, обеспечивающих необходимые подключения DDR3 контроллера.

Структура DDR3 подсистемы приведена на рисунке Рисунок 1.100 – Структура DDR3 подсистемы .

Состав системы:

контроллер памяти типа DDR3 SDRAM – DDR3MC;

блок моста с шиной типа PLB6 – PLB6MCIF2;

блок моста с шиной типа AMBA AXI – AXIMCIF2;

блок шинного арбитра – MCIF2ARB;

блоки, обеспечивающие управление подсистемой в целом и отдельными ее компонентами по шине управления типа DCR – CRGL, MCLFIR, DCRGRAIF.

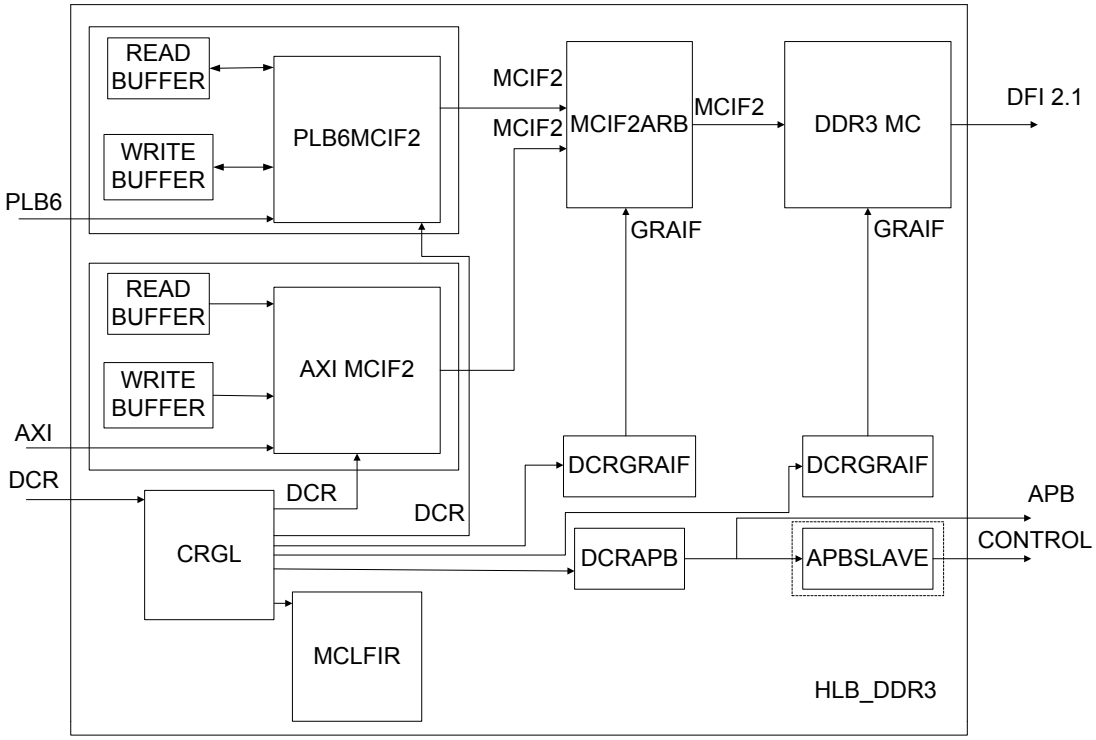


Рисунок 1.13 – Структура DDR3 подсистемы микросхемы

Следует отметить, что DDR3 подсистема не включает в своем составе блоков физического интерфейса DDR3 PHY, так как эти блоки должны быть расположены на верхнем уровне иерархии микросхемы. В то же время, управление блоками DDR3 PHY осуществляется из данной подсистемы по интерфейсу APB. Используется блок моста с шины DCR на шину APB – DCRAPB.

1.3.3.17 Конфигурирование ЦПУ подсистемы

Состояние ЦПУ подсистемы после сброса определяется следующим составляющими:

- Начальное состояние конфигурационных регистров устройств определяется значениями по умолчанию, указанными в документации
- Начальное состояние некоторых конфигурационных регистров устройств задается с помощью начальных значений, задаваемых на входных портах (strap pins).
- Задание некоторых режимов работы устройств задается с помощью начальных значений, задаваемых на входных портах (strap pins).
- Начальное состояние процессора описано в документе “Встроенное процессорное ядро PowerPC 476 FP. Руководство Пользователя” в Разделе 9 “Инициализация”.
- Программой начальной загрузки, исполняемой из постоянной памяти BOOT ROM, которая, в частности, прописывает состояние процессоров PowerPC 470S в состояние, описанное в документе “Встроенное процессорное ядро PowerPC 476 FP. Руководство Пользователя” в Разделе 9.4 “Требования к программной инициализации”.

Имена устройств при установке начальных значений конфигурационных регистров или режимов работы задается согласно детальной структурной схеме, приведенной в Приложении 28. Начальные значения установочных входов (strap pins) приведены в Таблице 1. Базовые DCR адреса в Таблице 1 не приведены, они указаны в Таблице 2. “Размещение устройств в пространстве DCR”. Если установочные входы отображаются на регистры устройств, то регистры и биты регистров указаны в столбце “Имя регистра, биты”.

Таблица 1.25 - Начальные значения регистров устройств

N	Название устройства	Название сигнала	Значение после сброса	Имя регистра, биты	Примечание
1	DCRARB4	DCRARB4M_STRAP_ARBBASEPRIVILEGED	1'b1		
2	DCRARB4	DCRARB4M_STRAP_ARBFAIRNESS	1'b0	DCAR(0)	
3	DCRARB4	DCRARB4M_STRAP_ARBLEGACYPROTECT	1'b0	DACR(1)	
4	DCRARB	M0_DCRMASTERID	4’h0		
5	DCRARB	M1_DCRMASTERID	4’h1		
6	DCRARB	M2_DCRMASTERID	4’h2		
7	DMA2PLB6_1	DMA2PLB6_1_DCR_ONE_TO_ONE_MODE	1'b1		
8	DMA2PLB6_0	DMA2PLB6_DCR_ONE_TO_ONE_MODE	1'b1		
9	DMA2PLB6_1	P6BC_M3_DMA2PLB6_ID	4’h3		
10	DMA2PLB6_0	P6BC_M2_DMA2PLB6_ID	4’h2		
11	L2C_0	I_M0_SELF_ID	5'h0		
12	L2C_1	I_M1_SELF_ID	5'h1		
13	L2C_x	L2C_L2SIZE	2'b01	L2CNFG0[L2Size]	Для обоих L2C
14	L2C_x	L2C_PLBCLKRATIO	2'b00	L2CNFG0[PLBClkRatio]	Для обоих L2C
15	L2C_x	L2C_PLBTSNOOP	3'b000	L2CNFG0[TSnoop]	Для обоих L2C
16	MPIC	MPIC128_STRAP_DCRONETOONEMODE	1'b1		
17	MPIC	MPIC128_STRAP_DEVICEID	8'h50		
18	MPIC	MPIC128_STRAP_TMRCLK_SELECT	mpic_tmrclk_sel		SCTL SYS_CONF 0
19	MPIC	MPIC128_STRAP_VENDORID	8'h14		
20	P6BC	P6BC_I_DCR_ONE_TO_ONE_MODE	1'b1		
21	P6BC	P6BC_I_EBRSD	3'h2		Загрузка с Сегмента 2
27	PLB4ARB8M_0	PGM_0_FAIREN	1'b0	PACR[PPM]	
29	PLB4ARB8M_1	PGM_1_FAIREN	1'b0	PACR[PPM]	
31	PLB4ARB8M_2	PGM_2_FAIREN	1'b0	PACR[PPM]	
33	PLB4ARB8M_3	PLB4ARB8M1_1_PGM_FAIREN	1'b0	PACR[PPM]	
35	PLB4ARB8M_4	PLB4ARB8M1_2_PGM_FAIREN	1'b0	PACR[PPM]	
37	PLB4ARB8M_5	PLB4ARB8M2_1_PGM_FAIREN	1'b0	PACR[PPM]	
39	PLB4ARB8M_6	PLB4ARB8M2_2_PGM_FAIREN	1'b0	PACR[PPM]	
41	PLB4PLB6_1	PLB4PLB61_I_DLYREADMODE	1'b0		
42	PLB4PLB6_1	PLB4PLB61_I_M_SELF_ID	5'h05	P46CR(24:31)	
43	PLB4PLB6_1	PLB4PLB61_I_ONETOONEMODE	1'b0		

Продолжение таблицы 1.26

N	Название устройства	Название сигнала	Значение после сброса	Имя регистра, биты	Примечание
44	PLB4PLB6_1	PLB4PLB61_I_PP_MID0	4'hF		По умолчанию
45	PLB4PLB6_1	PLB4PLB61_I_PP_MID1	4'hF		По умолчанию
46	PLB4PLB6_1	PLB4PLB61_I_PP_MID2	4'hF		По умолчанию
48	PLB4PLB6_2	PLB4PLB62_I_DLYREADMODE	1'b0		
49	PLB4PLB6_2	PLB4PLB62_I_M_SELF_ID	5'h06	P46CR(24:31)	
50	PLB4PLB6_2	PLB4PLB62_I_ONETOONEMODE	1'b0		
51	PLB4PLB6_2	PLB4PLB62_I_PP_MID0	4'hF		По умолчанию
52	PLB4PLB6_2	PLB4PLB62_I_PP_MID1	4'hF		По умолчанию
53	PLB4PLB6_2	PLB4PLB62_I_PP_MID2	4'hF		По умолчанию
55	PLB4PLB6_0	PLB4PLB6_DLYREADMODE	1'b0		
56	PLB4PLB6_0	PLB4PLB6_M_SELF_ID	5'h04	P46CR(24:31)	
57	PLB4PLB6_0	PLB4PLB6_ONETOONEMODE	1'b0		
58	PLB4PLB6_0	PLB4PLB6_PP_MID0	4'hF		По умолчанию
59	PLB4PLB6_0	PLB4PLB6_PP_MID1	4'hF		По умолчанию
60	PLB4PLB6_0	PLB4PLB6_PP_MID2	4'hF		По умолчанию
61	PLB4XAHB_1	PLB4XAHB1_BOOTSTRAP_ADDR_HI	8'h0F	TopAddr(7:0)	
62	PLB4XAHB_1	PLB4XAHB1_BOOTSTRAP_ADDR_LO	8'h01	BotAddr(7:0)	
63	PLB4XAHB_2	PLB4XAHB2_BOOTSTRAP_ADDR_HI	8'h0F	TopAddr(7:0)	
64	PLB4XAHB_2	PLB4XAHB2_BOOTSTRAP_ADDR_LO	8'h01	BotAddr(7:0)	
65	PLB4XAHB_0	PLB4XAHB_BOOTSTRAP_ADDR_HI	8'h0F	TopAddr(7:0)	
66	PLB4XAHB_0	PLB4XAHB_BOOTSTRAP_ADDR_LO	8'h01	BotAddr(7:0)	
67	PLB6PLB4_2	PLB6PLB41_I_ADDRH0	32'h00000012	ADDRH0	
68	PLB6PLB4_2	PLB6PLB41_I_ADDRL0	32'h00000012	ADDRL0	
70	PLB6PLB4_2	PLB6PLB41_I_ONETOONEMODE	1'b1		
71	PLB6PLB4_3	PLB6PLB42_I_ADDRH0	32'h00000013	ADDRH0	
72	PLB6PLB4_3	PLB6PLB42_I_ADDRL0	32'h00000013	ADDRL0	
74	PLB6PLB4_3	PLB6PLB42_I_ONETOONEMODE	1'b1		
75	PLB6PLB4_0	PLB6PLB4_0_ADDRH0	32'h000003FF	ADDRH0	
76	PLB6PLB4_0	PLB6PLB4_0_ADDRL0	32'h000003FF	ADDRL0	
78	PLB6PLB4_0	PLB6PLB4_0_ONETOONEMODE	1'b1		
79	PLB6PLB4_1	PLB6PLB4_1_ADDRH0	32'h00000010	ADDRH0	
80	PLB6PLB4_1	PLB6PLB4_1_ADDRL0	32'h00000010	ADDRL0	
82	PLB6PLB4_1	PLB6PLB4_1_ONETOONEMODE	1'b1		
84	CLDCR	PPC470S_CLDCR_I_DCR_ONE_TO_ONE_MODE	1'b1		
85	SRAMMC_0	SYS_0_DATA_PARITY	1'b0	DPC(0)	

Продолжение таблицы 1.27

N	Название устройства	Название сигнала	Значение после сброса	Имя регистра, биты	Примечание
86	SRAMMC_0	SYS_0_PGM_ALTCR0	32'hFFFC098F	SB0CR	
87	SRAMMC_0	SYS_0_PGM_ALTCR1	32'hFFFD098F	SB1CR	
88	SRAMMC_0	SYS_0_PGM_ALTCR2	32'hFFFE098F	SB2CR	
89	SRAMMC_0	SYS_0_PGM_ALTCR3	32'hFFFF098F	SB3CR	
91	SRAMMC_0	SYS_0_PGM_PB0S	3'b101		64КБ
92	SRAMMC_0	SYS_0_PGM_PB1S	3'b101		64КБ
93	SRAMMC_0	SYS_0_PGM_PB2S	3'b101		64КБ
94	SRAMMC_0	SYS_0_PGM_PB3S	3'b101		64КБ
	SRAMMC_0	SYS_0_ALTMODE	1’b0		
95	SRAMMC_1	SYS_1_DATA_PARITY	1'b0	DPC(0)	
96	SRAMMC_1	SYS_1_PGM_ALTCR0	32'h00040980	SB0CR	
97	SRAMMC_1	SYS_1_PGM_ALTCR1	32'h00050980	SB1CR	
98	SRAMMC_1	SYS_1_PGM_ALTCR2	32'h00060980	SB2CR	
99	SRAMMC_1	SYS_1_PGM_ALTCR3	32'h00070980	SB3CR	
101	SRAMMC_1	SYS_1_PGM_PB0S	3'b101		64КБ
102	SRAMMC_1	SYS_1_PGM_PB1S	3'b101		64КБ
103	SRAMMC_1	SYS_1_PGM_PB2S	3'b101		64КБ
104	SRAMMC_1	SYS_1_PGM_PB3S	3'b101		64КБ
	SRAMMC_1	SYS_1_ALTMODE	1’b0		
105	PPC_0	TIE0_C470SERPNRESET	10'h3FF	RSTCFG[ERPN]	
106	PPC_0	TIE0_C470SPIR	c470s0_pir	PIR	SYS_BOOT0=0, PIR=0, SYS_BOOT0=1, PIR=1
107	PPC_1	TIE1_C470SERPNRESET	10'h3FF	RSTCFG[ERPN]	
108	PPC_1	TIE1_C470SPIR	c470s1_pir	PIR	SYS_BOOT0=0, PIR=1, SYS_BOOT0=1, PIR=0
109	PPC_x	TIE_C470SENDIANRESET	1'b0	RSTCFG[E]	Для обоих PPC
110	PPC_x	TIE_C470SPVR	32'h7FF5_20C0	PVR	Для обоих PPC
111	PPC_x	TIE_C470SUSERRESET	4'h0	RSTCFG[U]	Для обоих PPC

Примечание - Регистры, доступные по интерфейсу DCR, имеют размер 32 биты, самый старший бит (MSB) – 0, самый младший бит (LSB) – 31.

После прохождения процедуры инициализации состояние TLB для обоих процессоров CPU 0 и CPU 1 изображено в таблице 1.8. Расшифровка полей описана в документе “Встроенное процессорное ядро PowerPC 476 FP. Руководство Пользователя” в Разделе 4 “Модуль управления памятью”. Там же описана процедура добавления, удаления и поиска строк TLB.

Таблица 1.28 - Состояние строк TLB после инициализации, выполненной загрузчиком из BOOT ROM

Номер строки	EPN	V	TS	S	T	E+RPN	IL1 ID	U0123	WIMG	E	UXWR	SXWR	BE0_5
000-A	00000000	1	0	07	0000	01000000000	11	0000	0100	0	000	111	1
015-A	FFF00000	1	0	07	0000	3FFFFF00000	11	0000	0101	0	000	101	0
016-A	10000000	1	0	1F	0000	01010000000	11	0000	0101	1	000	011	2
032-A	20000000	1	0	1F	0000	01020000000	11	0000	0101	1	000	011	3
048-A	30000000	1	0	1F	0000	01030000000	11	0000	0101	1	000	011	4

После прохождения загрузки из BOOT ROM необходимо прописать конфигурационные регистры коммутатора шины PLB6. Для этого в регистры BC_SGD1, BC_SGD2 прописывается число 0x00000010, что означает, что начальный адрес второго сегмента (регистр BC_SGD2) начинается с адреса 0x00000010_00000000. Поскольку первый сегмент используется для снупирования, то он не виден для программирования, поэтому в регистр BC_SGD1 прописывается такое же значение, что и в BC_SGD2. Затем бит 0 регистра BC_CR0 устанавливается в 1, что означает отмену начальной карты памяти P6BC, задаваемой входами I_EBRSD[0:2], и использование карты памяти, задаваемой регистрами BC_SGDx.

1.3.4 Узел цифровой обработки сигналов

Микросхема содержит два одинаковых узла цифровой обработки. Каждый узел содержит две процессорные системы на базе DSP ядра NMC3 и общую кэш памятью 2-го уровня объемом 128 кБайт для повышения скорости обмена данными с внешней памятью.

1.3.4.1 Структурная схема и состав узла цифровой обработки сигналов

Структурная схема узла цифровой обработки сигналов приведена на рисунке ниже.

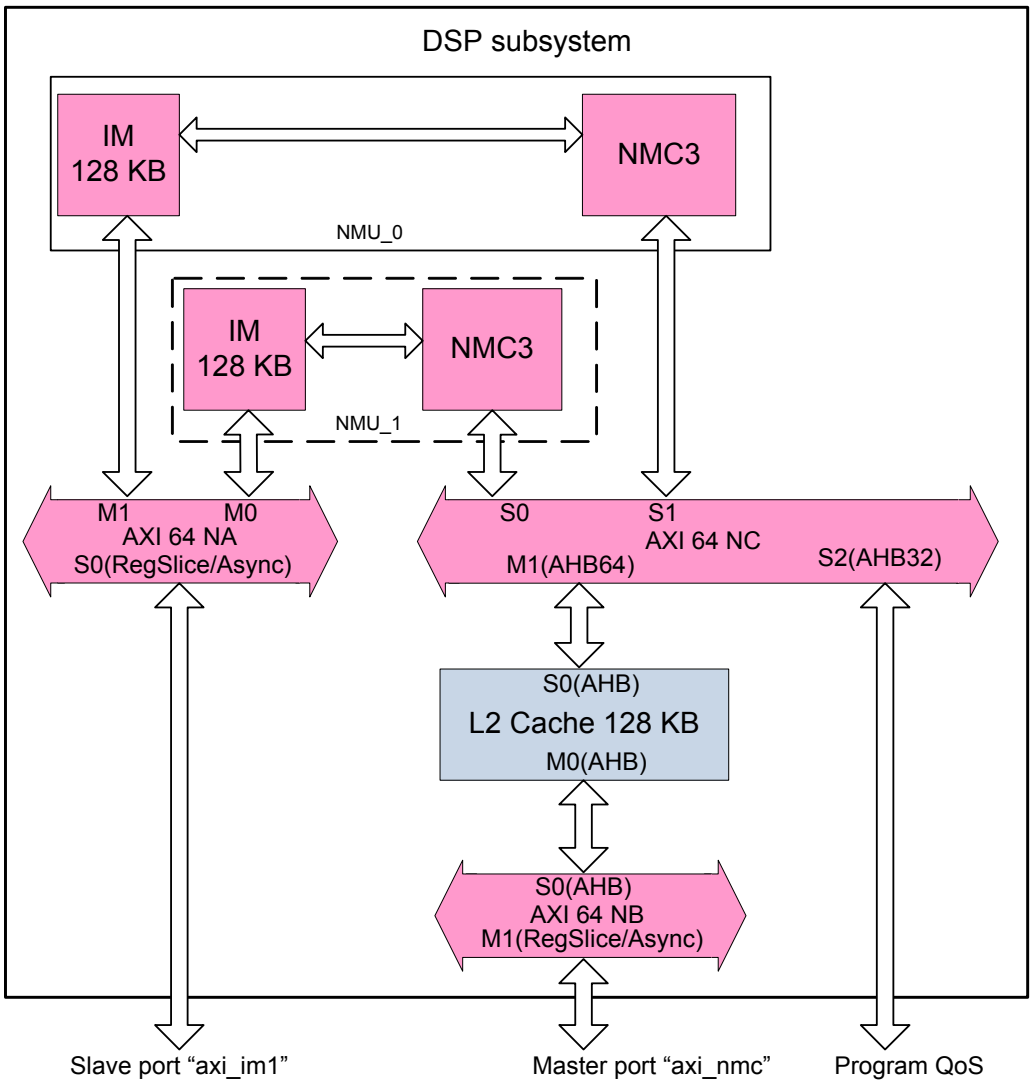


Рисунок 1.14 –Структурная схема узла цифровой обработки сигналов

NMU_0, NMU_1 – процессорные системы на базе DSP ядра NMC3, содержащие внутреннюю память IM общим объемом 128 КБайт.

L2 Cache – контроллер кэш памяти команд для процессорных систем NMU_0 и NMU_1 (128 Кбайт)

AXI 64 NC, AXI 64 NA, AXI 64 NB – шинные коммутаторы

1.3.4.2 Карта памяти для процессорной системы NMU

Ниже в таблице приведена карта памяти для процессорной системы NMU/

Таблица 1.29 – Карта памяти для процессорной системы NMU

адрес со стороны ядра NMC3 (32 разряда) от	адрес со стороны ядра NMC3 (32 разряда) до	AXI устройство
0x0000_0000	0x0000_1FFF	NMB0 (0 - й банк своей памяти)
0x0000_2000	0x0000_3FFF	NMB1 (1 - й банк своей памяти)
0x0000_4000	0x0000_5FFF	NMB2 (2 - й банк своей памяти)
0x0000_6000	0x0000_7FFF	NMB3 (3 - й банк своей памяти)
0x0001_0000	0x0001_FFFF	IM0 (256K)
0x0002_0000	0x0002_0001	PU
0x0010_0000	0x00FF_FFFF	DCR (48M)
0x0400_0000	0x0DFF_FFFF	IFSYS0 (640M)
0x0E00_0000	0x0E00_3FFF	APB0(64K)
0x0E00_4000	0x0E00_8FFF	APB1(80K)
0x0E04_0000	0x0E07_FFFF	NIC301_A_CFG(1M)
0x0E08_0000	0x0E0B_FFFF	NIC301_DSP0_CFG(1M)
0x0E0E_0000	0x0E0F_FFFF	NIC301_DSP1_CFG(1M)
0x0E20_0000	0x0E2F_FFFF	DSP_L2C regs, access only by NMC(4M)
0x0E40_0000	0x0E40_7FFF	DSP0_NM0 (128K)
0x0E40_8000	0x0E40_FFFF	DSP0_NM1 (128K)
0x0E41_0000	0x0E41_7FFF	DSP1_NM0 (128K)
0x0E41_8000	0x0E41_FFFF	DSP1_NM1 (128K)
0x0E42_0000	0x0E42_03FF	I2S(4K)
0x0E42_0400	0x0E42_07FF	SPDIF(4K)
0x0F00_0000	0x0FFF_FFFF	IFSYS1 (64M)
0x1000_0000	0x1FFF_FFFF	DDR3 EM0 1GB,access directly from AXI
0x2000_0000	0x3FFF_FFFF	DDR3 EM1 2GB, access directly from AXI
0x4002_0000	0x4002_FFFF	DMAC контроллер GLG процессорной системы NMU

1.3.4.3 Процессорная система на базе DSP ядра NeuroMatrix Core3 (NMU)

Структурная схема процессорной системы на базе DSP ядра NeuroMatrix Core 3 (NMC3) приведена ниже (см. Рисунок 1.15).

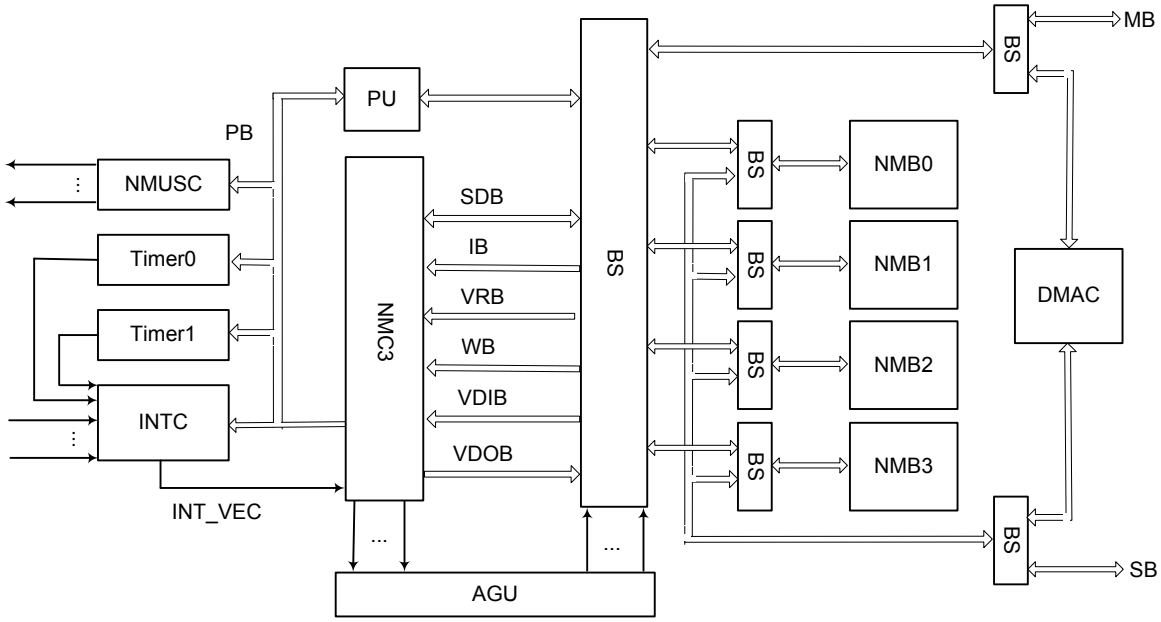


Рисунок 1.15 – Структурная схема процессорной системы на базе DSP ядра NeuroMatrix Core 3

Процессорная система на базе DSP ядра NeuroMatrix Core 3 работает на частоте 400 МГц и содержит следующие функциональные узлы:

NMC3 (NeuroMatrix Core 3) – DSP ядро NeuroMatrix Core 3, которое представляет собой ядро высокопроизводительного векторно-матричного процессора с оригинальной динамической суперскалярной параллельной архитектурой и сверхбольшими словами команд. Обмен ядра с внешним миром осуществляется с помощью шести 64-разрядных шин: команд (IB), скалярных данных (SDB), векторных входных данных (VDIB), весов (WB), векторного регистра (VRB) и векторных выходных данных (VDOB). Описание ядра NMC3 находится в отдельном документе: “СФ - БЛОК NMC3 Техническое описание ЮФКВ.431282.007ТО”.

NMUSC (NMU System Controller) – системный контроллер процессорной системы NMU.

Данный контроллер содержит программно-доступный со стороны NMC3 периферийный регистр, который выполняет функции управления прерываниями процессорному ядру PowerPC и осуществляет вывод на выводы TD тестовых сигналов.

INTC (Interrupt Controller) – контроллер прерываний, который формирует запрос для NMC3 на одно из следующих внешних прерываний:

- высокоприоритетное и низкоприоритетное от системного контроллера SCTL процессорного ядра PowerPC;
- одно прерывание от блока DMAC;
- одно прерывание от блока PU;
- одно прерывание от блока TIMER0;
- одно прерывание от блока TIMER1;

а также соответствующий ему адрес-вектор прерывания (INT VEC).

TIMER0 и TIMER1 – два интервальных 32-разрядных таймера.

Таймеры идентичны по функциям, независимы друг от друга и предназначены для отсчета задаваемых интервалов времени для ядра NMC3. Каждый таймер содержит 32-разрядный счетчик, осуществляющий отсчет временных интервалов. Данный счетчик состоит из рабочего и теневого регистров.

Каждый из таймеров может работать как в непрерывном режиме, так и в режиме однократного запуска. Интервал счета таймера задается программно. В качестве сигнала счета выступает тактовый сигнал процессора. По достижении нулевого значения таймер формирует сигнал прерывания, который может быть обработан блоком прерываний стандартным образом.

PU (Permutation Unit) – блок упаковки/распаковки векторов данных. Данный блок выступает в роли сопроцессора для NMC3, и его задачей является преобразование потоковых данных в такой формат, который более удобен для обработки в NMC3 (распаковка) или для хранения в памяти (упаковка). Соответственно, применение блока PU позволяет резко повысить эффективность ядра NMC3 на таких задачах, как корреляция или фильтрация, при этом хранить коэффициенты для той же корреляции или фильтрации, а также результаты обработки в компактном виде. Блок PU аппаратно поддерживает следующие операции:

Распаковка входного вектора в выходной с элементами заданной большей разрядности (до 64 разрядов), чем в исходном векторе, с возможностью распространения знака или нуля внутри элемента.

Упаковка входного вектора в выходной с элементами заданной меньшей разрядности (до 1 разряда), чем в исходном векторе, с возможностью выбора любых необходимых разрядов (соседних) внутри элементов с отбрасыванием остальных.

Преобразование входного вектора в диагональную матрицу (может использоваться для ускорения скалярного произведения векторов в ядре NMC3).

Запись элементов входного вектора в столбцы выходной матрицы или обратное этому преобразование.

Доступ к векторам, хранящимся в памяти, с адресацией (начальным смещением) до элемента этих векторов (вплоть до 1 разряда).

Возможность работы с блоками данных в памяти как с циклическими буферами (вычисление адреса данных по модулю заданного числа).

Более подробно PU описан в подразделе 1.3.4.3.4.

DMAC (DMA Controller) – контроллер ПДП, предназначен для пересылки данных между внутренней памятью NMU и внешней по отношению к NMU памятью. Управление осуществляется процессорной системой NMU.

AGU (Address Generator Unit) – блок адресных генераторов, который получает запрос на обмен данными с памятью от ядра NMC3 (начальный и конечный адрес, смещение, число обращений) и затем сам формирует адреса, освобождая от этого процессор. Благодаря 6 шинам ядра NMC3 и наличию в блоке 6 адресных генераторов, возможно осуществлять до шести операций ввода-вывода за один такт.

BS (Bus Switch) – шинный коммутатор.

NMB (NeuroMatrix Memory Bank) - банк памяти объёмом 4K×64, который может адресоваться до 32- или 64-разрядного слова.

1.3.4.3.1 Интервальные таймеры TIMER0 и TIMER1.

Таймеры идентичны по функциям, независимы друг от друга и предназначены для отсчета интервалов времени для ядра NMC3. Таймер содержит 32-разрядный счетчик, осуществляющий отсчет временных интервалов. Данный счетчик состоит из рабочего и теневого 32-х разрядных регистров (см. Рисунок 1.16).

Таймер может работать как в непрерывном режиме, так и в режиме однократного запуска. Интервал счета таймера задается программно. В качестве сигнала счета выступает тактовый сигнал процессора. По достижении нулевого значения таймер формирует сигнал прерывания.

Таймер имеет следующие программно доступные регистры: рабочий регистр счетчика TMR_CNTw (доступен на чтение), теневой регистр счетчика TMR_CNTs (доступен на запись), регистр состояния TMR_MODE (доступен на запись и на чтение).

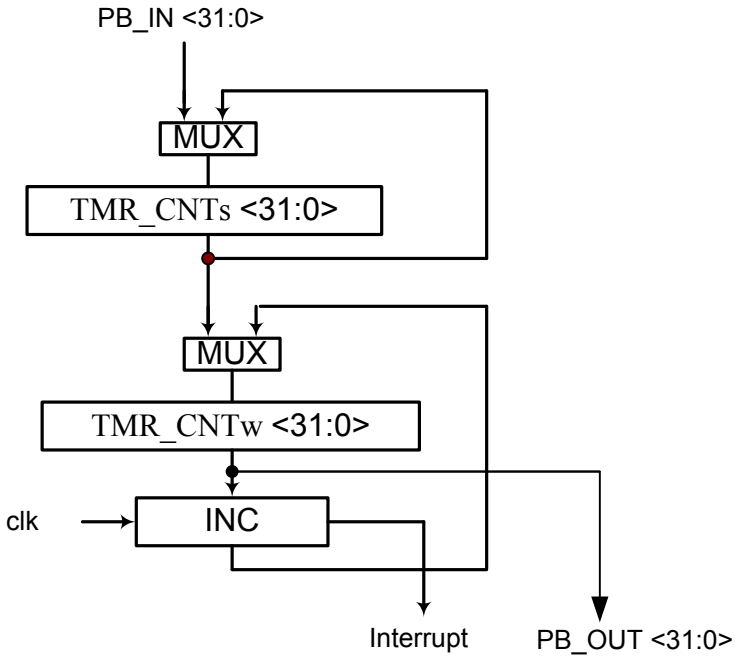


Рисунок 1.16- Структурная схема таймера.

Теневой и рабочий регистр счетчика имеют одинаковый формат (см. Рисунок 1.17).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Count_value																															

Рисунок 1.17 – Регистр счетчика.

Поле Count_value (разряды 31-0) задает интервал счета.
Регистр состояния управляет режимом работы и запуском таймера.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
зарезервировано																														E	M

Рисунок 1.18 – Формат регистра состояния TMR_MODE.

0-й разряд (M) задает режим работы таймера:
0 – режим многократного запуска;

1 – режим однократного запуска.

1-й разряд (E) разрешает или запрещает работу таймера:

0 – таймер не работает;

1 – таймер работает.

В теневой регистр счетчика программно записывается значение, которое одновременно переписывается в рабочий регистр. После запуска таймера рабочий регистр счетчика будет инкрементироваться каждый процессорный такт. Когда рабочий регистр счетчика будет иметь значение 0h, происходит выдача прерывания и в рабочий регистр счетчика переписывается значение теневого регистра. Если таймер работает в режиме однократного запуска, то бит E в регистре состояния сбросится в 0 и таймер остановится. Иначе таймер продолжит свою работу. В процессе работы программно можно прочитать рабочий регистр счетчика, а также остановить таймер, записав 0 в поле E регистра состояния. Если после этого снова в поле E записать 1, то таймер продолжит отсчет с того значения, на котором произошел останов. Таким образом, чтобы отсчитываемый временной интервал составлял:

1 такт надо записать значение FFFF_FFFFh в теневой регистр счетчика;

2 такта надо записать значение FFFF_FFFEh в теневой регистр счетчика;

n тактов надо записать значение “–n” в дополнительном коде в теневой регистр счетчика.

Максимальный временной интервал 232 процессорных такта, для этого надо записать значение 0000_0000h в теневой регистр счетчика.

Регистры таймеров являются периферийными и расположенвы в регистровом окне 0, адреса приведены в таблице 1.10.

Таблица 1.30 – Коды для обращения к регистрам таймеров (периферийные регистры для NMC3)

Код регистра	Регистр - источник	Регистр – приемник
	Регистровое окно 0	
110000	TMR_CNT0	TMR_CNT0
110001	TMR_CNT1	TMR_CNT1
110010	TMR_MODE0	TMR_MODE0
110011	TMR_MODE1	TMR_MODE1

1.3.4.3.2 Системный контроллер процессорной системы NMU (NMCSC), который выполняет функции управления прерываниями

Контроллер управляет запросами на прерывание к процессорному ядру PowerPC и контролирует выводы тестового порта. Контроллер содержит программно-доступный со стороны ядра NMC3 периферийный регистр NMCSCR, который доступен на чтение и запись по адресу периферийного регистра 110111b (pr7) в регистровом окне 0.

31–10	9–8	7–6	5–4	3	2	1	0
Резерв	TD			резерв	PowerPCL	PowerPCH	резерв

Рисунок 1.19 – Формат регистра NMCSCR.

2-й разряд (PowerPCH) - запрос на высокоприоритетное прерывание к процессорному ядру PowerPC. Запись единицы в данный разряд вызовет высокоприоритетное прерывание в системном контроллере (SCTL) системы процессорного ядра PowerPC. Сброс данного разряда происходит автоматически.

3-й разряд (PowerPCL) - запрос на низкоприоритетное прерывание к процессорному ядру PowerPC. Запись единицы в данный разряд вызовет низкоприоритетное прерывание в системном контроллере (SCTL) системы процессорного ядра PowerPC. Сброс данного разряда происходит автоматически.

Поле TD (8-5 разряды) является тестовым портом. Значение, записанное в данное поле, будет через два такта после записи выдано на выводы TD3 - TD0;

1.3.4.3.3 Контроллер прерываний (INTC)

Контроллер прерываний формирует запрос для ядра NMC3 на одно из следующих внешних прерываний:

- высокоприоритетное и низкоприоритетное от процессорного ядра PowerPC;
- одно прерывание от блока DMAC;
- одно прерывание от блока TIMER0;
- одно прерывание от блока TIMER1;
- одно прерывание от блока PU.

Контроллер прерываний формирует соответствующий запросу адрес-вектор прерывания (INTV).

Запросы на прерывания приходят на входы блока INTC IRQ[15:0] (см. Рисунок 1.20). По положительному фронту запроса устанавливается соответствующий бит в регистре запросов IRR. Далее на пути сигнала стоит регистр маски IMR. Значение 0 в соответствующем бите разрешает прохождение сигнала, а значение 1 блокирует. Далее схема приоритетов Priority Unit выбирает запрос с наибольшим приоритетом, формирует запрос на прерывания для ядра NMC3 и соответствующий ему адрес-вектор прерывания. Приоритет запросов фиксированный: IRQ0 имеет наивысший приоритет, IRQ15 – наименее приоритетный запрос.

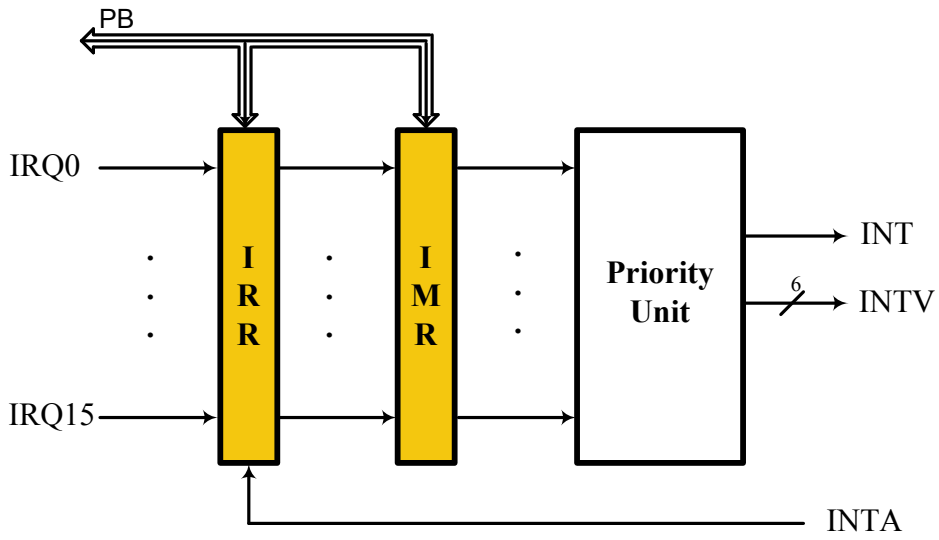


Рисунок 1.20 – Структура контроллера прерываний.

Если сигнал INTA имеет активный уровень, то считается, что процессорное ядро NMC3 зафиксировало прерывание, и соответствующий бит регистра IRR сбрасывается. Если сигнал INTA имеет неактивный уровень, то запрос на прерывание INT будет стоять до прихода сигнала INTA. При этом адрес-вектор INTV на выходе блока INTC не фиксируется, т.е. если на вход IRQx поступит более приоритетное прерывание, то адрес-вектор изменится и будет соответствовать более приоритетному прерыванию.

Регистр запросов IRR доступен на чтение и побитовую установку/сброс, регистр маски IMR доступен на чтение и запись со стороны ядра NMC3 по периферийной шине. Коды периферийных регистров приведены в таблице 1.11.

Таблица 1.31 – Коды для обращения к регистрам блока INTC (периферийные регистры для NMC3)

Код регистра	Регистр – источник	Регистр – приемник
	Регистровое окно 0	
110100	IMR	IMR
110101	IRR	IRRreset
110110	IRR	IRRset

Формат регистра IRR приведен ниже (см. Рисунок 1.21). Единица в соответствующем разряде означает запрос на прерывание.

31–16	15–0
Reserved	IRQ<15:0>

Рисунок 1.21 – Формат регистра IRR.

Формат регистра IMR приведен ниже (см. Рисунок 1.22). Единица в соответствующем разряде означает маскирование запроса (запрос не проходит), 0 – запрос не маскируется (запрос проходит).

31–16	15–0
Reserved	IMR<15:0>

Рисунок 1.22 – Формат регистра IMR.

Примечание. Запись в регистр IMR, сброс регистра IRR и т.д. осуществляется на последней ступени конвейера, а сброса регистра INTR в ядре NMC3 осуществляется на первой ступени конвейера, поэтому если необходимо сбросить какой-либо бит INTR строго после записи/сброса IMR/IRR необходимо прочистить конвейер. Это возможно несколькими способами, например подачей 9 команд vnul с директивой .wait или записью в PC нового значения. Пример см. Ниже:

```
.wait;
        pr5 = 0ffffh;       //clear all request in INT controller
        vnul; vnul; vnul; vnul; vnul; vnul; vnul; vnul; vnul;
        intr clear 03c0h;

или

.wait;
        pr5 = 0ffffh;       //clear all request in INT controller
        pc = INTR_CLEAR;
<INTR_CLEAR>
        intr clear 03c0h;
```

Список прерываний, которые подаются на входы запросов IRQ, и соответствующий им адрес-вектор, приведен в таблице 1.12.

Таблица 1.32 – Список внешних прерываний на входах IRQ блока INTC

№ входа	источник	Адрес вектор		Описание
		bin	hex	
IRQ0	INT_PowerPC_IN	1000_0000b	80h	Высокоприоритетное прерывание от процессорного ядра PowerPC
IRQ1	Резерв	1000_1000b	88h	
IRQ2	INT_TIMER0	1001_0000b	90h	Прерывание от таймера 0
IRQ3	Резерв	1001_1000b	98h	
IRQ4	Резерв	1010_0000b	A0h	
IRQ5	резерв	1010_1000b	A8h	
IRQ6	INT_DMAC	1011_0000b	B0h	Прерывания от блока DMAC
IRQ7	INT_TIMER1	1011_1000b	B8h	Прерывание от таймера 1
IRQ8		1100_0000b	C0h	
IRQ9	INT_ PowerPC_IN	1100_1000b	C8h	Низкоприоритетное прерывание от процессорного ядра PowerPC
IRQ10		1101_0000b	D0h	
IRQ11	INT_PU	1101_1000b	D8h	Прерывание по окончанию работы упаковщика PU
IRQ12	Резерв	1110_0000b	E0h	
IRQ13	резерв	1110_1000b	E8h	
IRQ14	Резерв	1111_0000b	F0h	
IRQ15	резерв	1111_1000b	F8h	

1.3.4.3.4 PU (Permutation Unit) – блок упаковки/распаковки векторов данных

Для преобразования потоков данных в формат, удобный для обработки в NMC3 или для хранения в памяти, потоки данных в первом случае распаковываются, а во втром – упаковываются. Этим занимается устройство упаковки/распаковки потоков данных PU (Permutation Unit). Использование PU позволяет резко повысить эффективность NMC3 на таких задачах, как корреляция и фильтрация, при этом хранить коэффициенты для корреляции и фильтрации, а также результаты обработки в компактном виде.

Устройство выполняет распаковку и упаковку потоков данных, при этом элемент входных и выходных данных может иметь размер от 1 до 64 разрядов, результаты могут выдаваться как начиная с младших элементов, так и со старших. Предусмотрено преобразование матрицы в вектор и распаковка вектора с преобразованием в матрицу.

Блок PU может получать данные от ядра NMC3 или самостоятельно формировать адрес для обращения в память за данными или для записи результата.

Настройка блока осуществляется ядром NMC3 путем записи конфигурационных регистров PU.

1.3.4.3.4.1 Аппаратно поддерживаемые операции

Блок PU обрабатывает длинные последовательности двоичных данных. Элементы данных в последовательности имеют одинаковый размер в диапазоне от 1 до 64 бит. Каждая последовательность данных хранится упакованной в 64-х разрядных словах, размещаемых в последовательных ячейках памяти.

Блок PU выполняет обмен с памятью 64-х разрядными машинными словами.

Обрабатываемые данные могут восприниматься как числа в дополнительном коде или простые наборы бит. Последовательности данных могут образовывать вектора и/или матрицы.

Вектор представляет собой несколько последовательных данных.

Матрица – массив (несколько) векторов.

Упаковка входных данных с возможностью выбора любых необходимых разрядов с отбрасыванием остальных. При упаковке настраиваются следующие параметры:

- X – величина, на которую сначала надо сдвинуть весь поток данных ($63 \geq X \geq 0$);
- N – разрядность входных неупакованных данных (Возможные значения N: 2, 4, 8, 16, 32, 64);
- XS – количество разрядов от начала данного (самого младшего разряда) до первого упаковываемого разряда;
- M – разрядность выходных упакованных данных (N > M. Возможные значения M: 2, 4, 8, 16, 32, 64);
- Использовать при упаковке насыщение или нет. При этом числа считаются как числа представленные в дополнительном коде. Если насыщение включено, и в старших отбрасываемых разрядах были значащие биты, то результатом операции насыщения будет максимальное положительное (011…1b) число или максимальное отрицательное (100…0b) число. Если же в старших отбрасываемых разрядах не было значащих бит, то насыщения не происходит.

Пример упаковки без насыщения, N=16, M=8, XS=2 представлен ниже (см. Рисунок 1.23)

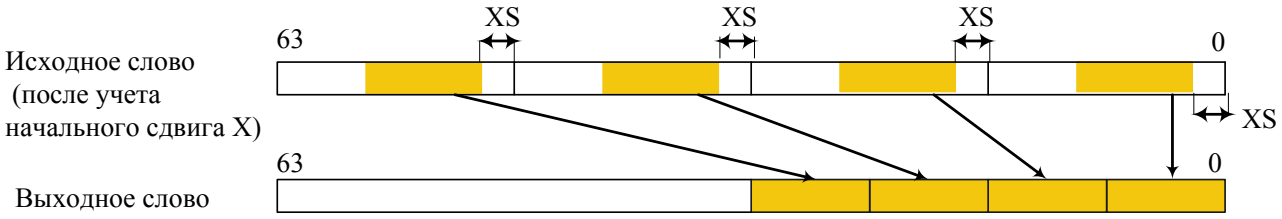


Рисунок 1.23 – Упаковка без насыщения.

Упаковка в слова с одноразрядными данными.

При данной операции входные данные должны быть либо “1”, либо “-1”. При этом единица в выходном слове кодирует значение “-1”, а нуль – “1” При данной упаковке настраиваются следующие параметры:

- N – разрядность входных неупакованных данных (Возможные значения N: 2, 4, 8, 16, 32, 64),
- X – величина, на которую сначала надо сдвинуть весь поток данных, (63 ≥ X ≥ 0).

Пример упаковки в вектора с одноразрядными данными при N = 16 приведен ниже (см. Рисунок 1.24).

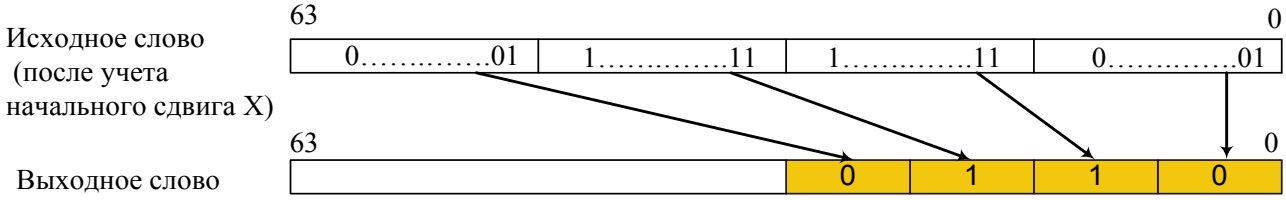


Рисунок 1.24 – Упаковка в вектора с одноразрядными данными

Распаковка. Распаковка осуществляется распространением знакового разряда или нуля в старших разрядах до необходимой разрядности. При распаковке настраиваются следующие параметры:

- X – величина, на которую сначала надо сдвинуть весь поток данных, (63 ≥ X ≥ 0),
- N – разрядность входных упакованных данных (Возможные значения N: 2, 4, 8, 16, 32, 64);
- M – разрядность выходных распакованных данных (N ≤ M, возможные значения M: 2, 4, 8, 16, 32, 64),
- Y – шаг, на который надо сдвинуть входной поток, для получения очередного слова для дальнейшей обработки. Более подробное объяснение данного параметра показано на рисунке ниже (см. Рисунок 1.25).

Распространять знаковый разряд или распространять нуль.

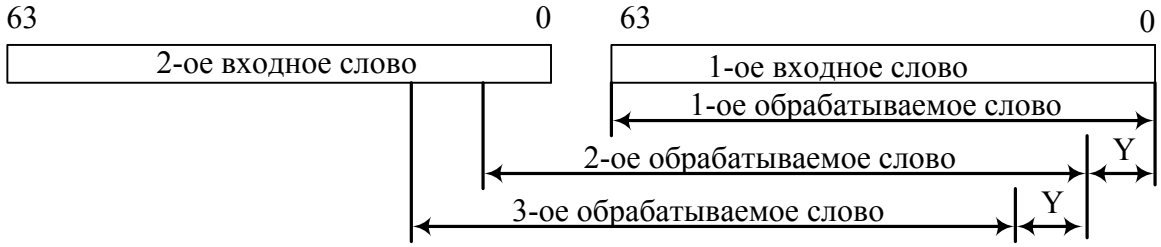


Рисунок 1.25. Получение слов для обработки из входного потока данных

Пример распаковки для N = 8, M = 16, распространение нулем представлен ниже (см. Рисунок 1.26).

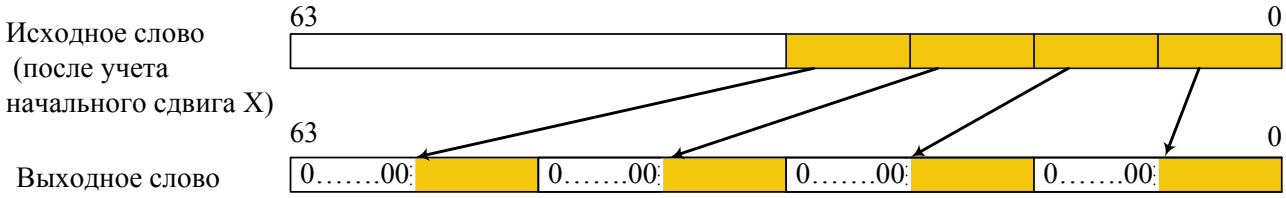
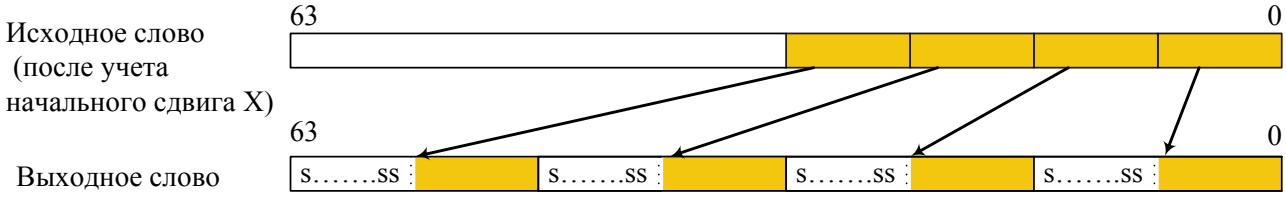


Рисунок 1.26 – Распаковка N=8, M=16, распространение нулем

Пример распаковки для N = 8, M = 16, распространение знаком представлен ниже (см. Рисунок 1.27)



s – знаковый разряд числа

Рисунок 1.27 – Распаковка N=8, M=16, распространение знаком

Распаковка одноразрядных данных. При данной операции единица во входном слове кодирует значение – “-1”, а нуль – “1” При данной распаковке настраиваются следующие параметры:

- M – разрядность выходных неупакованных данных (Возможные значения M: 2, 4, 8, 16, 32, 64),
- X – величина, на которую сначала надо сдвинуть весь поток данных, (63 ≥ X ≥ 0),
- Y – шаг, на который надо сдвинуть входной поток для получения очередного слова для дальнейшей обработки (64 ≥ Y ≥ 1).

Пример распаковки входного слова одноразрядных данных с M = 16 представлен ниже (см. Рисунок 1.28).

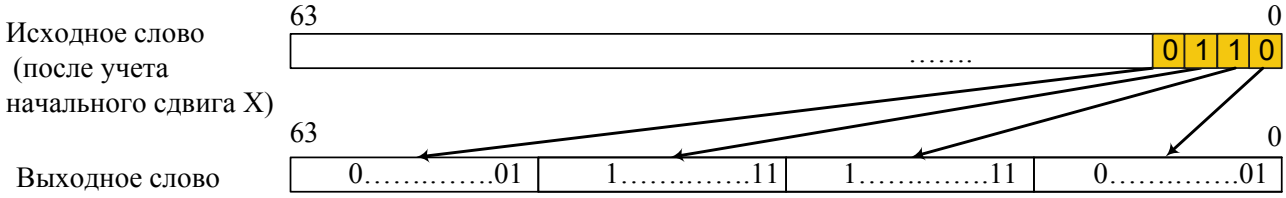


Рисунок 1.28 – Распаковка входного слова одноразрядных данных

Порядок выдачи данных в выходном слове.

Результат может быть выдан, начиная с младших элементов или начиная со старших элементов в выходном слове (см. Рисунок 1.29 и Рисунок 1.30).

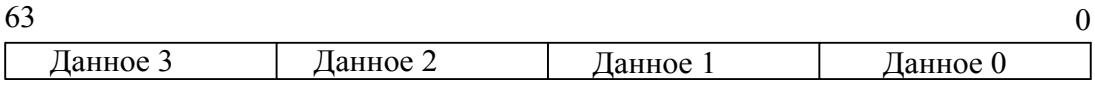


Рисунок 1.29 – Выдача слова, начиная с младших элементов.

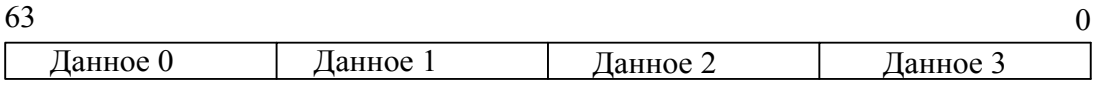


Рисунок 1.30 – Выдача вектора, начиная со старших элементов.

Операции с матрицами.

Получение диагональной матрицы. С помощью PU можно из входного вектора получить матрицу, элементы матрицы можно расположить как в главной, так и в побочной диагонали. Ниже приведен пример для вектора из 4-х элементов (см. Рисунок 1.31). Данное действие выполняется вместе с режимом распаковки.

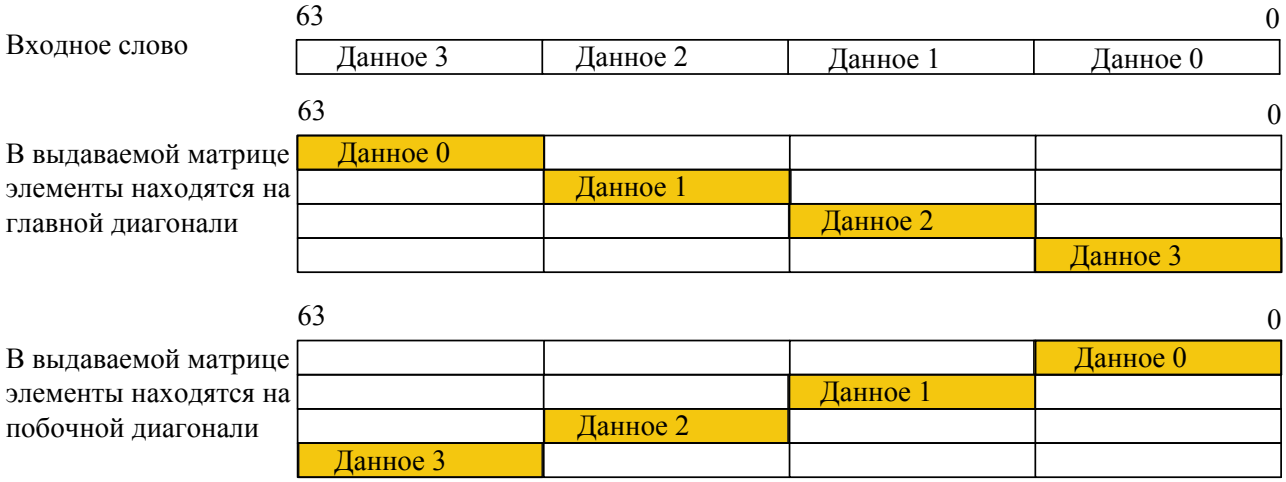


Рисунок 1.31 – Получение диагональной матрицы

Преобразование диагональной матрицы в вектор. Данное действие является обратным к описанному выше действию.

Получение матрицы, в которой элементы находятся в определенном столбце. С помощью PU из слова, состоящего из элементов размером M, можно получить матрицу, в которой элементы исходного слова стоят в определенном столбце. При этом задается номер столбца, в котором будут размещаться элементы (см. Рисунок 1.32). Данное действие выполняется вместе с режимом распаковки.

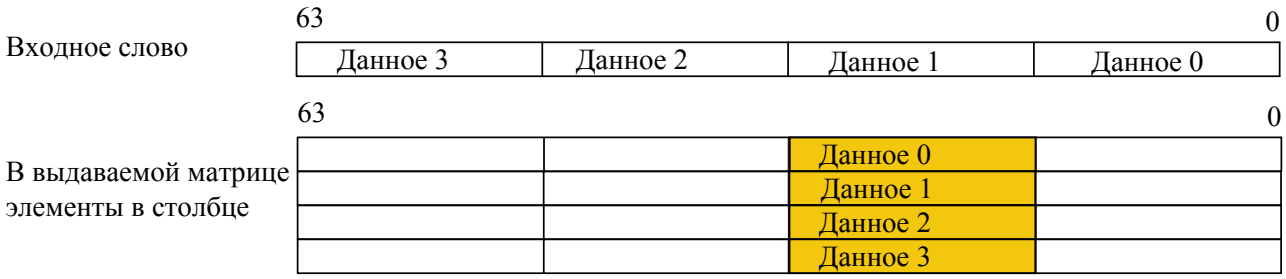


Рисунок 1.32– Размещение входного слова в первом столбце матрицы

Преобразование матрицы, в которой элементы находятся в определенном столбце, в вектор. Данное действие является обратным к описанному выше действию. Данное действие выполняется при помощи обычной упаковки с N=64.

1.3.4.3.4.2 Режимы и порядок работы PU.

Блок PU может получать данные от ядра NMC3 по шине PU_SB или самостоятельно формировать адрес для обращения в память за данными по шине PU_MB. Для передачи данных от ядра NMC3 к PU в адресном пространстве NMC3 выделен специальный адрес 20001h. При записи данных по этому адресу скалярными или векторными командами данные будут переданы в блок PU для обработки. Запись скалярными командами должна производиться 64-х разрядными словами (регистровыми парами). Результат обработки может быть считан ядром NMC3 напрямую из блока PU по шине PU_SB или PU может самостоятельно формировать адрес для записи результата в память по шине PU_MB. Для чтения данных с выхода PU ядро NMC3 использует специально выделенный адрес 20000h. При чтении данных по этому адресу скалярными или векторными командами данные будут считаны из блока PU. Чтение скалярными командами должно производиться 64-х разрядными словами (регистровыми парами). Возможны следующие режимы работы блока PU:

- Блок PU получает данные от ядра NMC3 по шине PU_SB[63:0] и записывает результат обработки в память по шине PU_MB.
- Блок PU забирает данные из памяти по шине PU_MB[63:0] и передает данные по запросу от NMC3 по шине PU_SB[63:0].
- Блок PU получает данные по шине PU_SB[63:0] и передает данные по запросу от NMC3 по шине PU_SB[63:0].
- Блок PU не работает.

Если PU настраивается таким образом, что самостоятельно читает или записывает данные (по шине PU_MB), то необходимо выделить область памяти, в которую он будет обращаться. Границы этой области сообщаются блоку PU через регистры PUR7 и PUR6; начальный (стартовый) адрес внутри области – через регистр PUR5. PU обращается последовательно по адресам в выделенной области и не может выйти за её границы – каждый раз, когда достигается старший адрес области, следующее обращение следует по младшему адресу области.

Предусмотрен следующий порядок работы с блоком PU:

- Настройка конфигурации путем записи в конфигурационные регистры PU процессорным ядром NMC3.
- Запуск блока PU.
- Передача и/или получение данных с помощью команд записи и/или чтения данных ядра NMC3. При этом количество читаемых и записываемых данных должно строго соответствовать заданным настройкам регистров CNT_IN и CNT_OUT.
- Обработка прерывания от устройства. После того, как блок PU заканчивает работу, происходит выдача прерывания. Это говорит о том что блок готов к настройке на обработку следующего пакета данных.

После системного сброса блок PU переходит в состояние “Блок PU не работает” и находится в этом состоянии до момента программного изменения поля, отвечающего за режим работы. После записи в это поле значения, отличного от 0, блок PU начинает работу в соответствии с тем режимом, который был записан, и прочими настройками, заданными в конфигурационных регистрах.

1.3.4.3.4.3 Конфигурационные регистры

Таблица 1.33 – Коды регистров блока PU (периферийные регистры для NMC3) .

Код регистра	Регистр – приемник
	Регистровое окно 1
110000 110001	PUR0 PUR1
111000 111001 111010 111011 111100 111101 111110 111111	PUR2 PUR3 PUR4 PUR5 PUR6 PUR7 PUR8 PUR9

Регистры блока PU доступны только для записи как периферийные регистры ядра NMC3. Коды регистров блока PU приведены в таблице 1.13. Конфигурационные регистры можно разделить на три группы.

Первая группа регистров содержит код операции. Группа состоит из регистров PUR8, PUR9.

Вторая группа регистров PUR7, PUR6, PUR5, PUR4, PUR3 используется для организации счетчиков. Регистры содержат параметры для обращения в память – поля TOP, BOTTOM, FIRST, количество слов, которое нужно принять из процессора или из памяти – поле CNT_IN, количество слов, которое нужно выдать в память или в процессор – поле CNT_OUT.

Третья группа регистров PUR2, PUR1, PUR0 предназначена для экономии аппаратуры и упрощения реализации функциональных блоков. Регистры содержат используемые для разных целей маски. Если неправильно задать значения этих регистров, могут быть получены ошибочные данные.

Регистр PUR9 должен быть записан последним. Порядок записи остальных регистров не важен.

После сброса поле регистра PUR9, отвечающее за режим работы (MODE), принимает значение 0 (блок PU не работает), остальные поля и регистры после сброса не определены.

Структура конфигурационных регистров PU дана ниже.

Регистр PUR9. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR9 является периферийным в регистровом окне 1 и доступен по адресу 111111b (pr15).

31 30 29	28 27 26 25 24 23	22 21 20 19 18 17	16 15 14 13 12 11	10	9	8	7	6	5	4	3	2	1 0
зарезервировано	Xs	SHIFT	COLNUM	COL	DIAG	MATOUT	MATIN	MSB	BIT	SAT	SE	PACKAA	MODE

Рисунок 1.33 – Формат регистра PUR9.

Поле MODE (разряды с 1 по 0) определяет режим работы PU:

00 – PU не работает

01 – Входные данные поступают по шине PU_SB от процессорного ядра NMC3, выходные данные записываются в память по шине PU_MB. Адрес для записи результата вырабатывается блоком PU (PU_SB → PU_MB).

10 – Входные данные поступают по шине PU_MB. Адрес для чтения входных данных вырабатывается блоком PU, выходные данные выдаются на шину PU_SB, которые забирает процессорное ядро NMC3 (PU_MB → PU_SB).

11 – Входные данные поступают по шине PU_SB от процессорного ядра NMC3, выходные данные выдаются на шину PU_SB, которые забирает процессорное ядро NMC3 (PU_SB → PU_SB).

После того, как в поле MODE записано ненулевое значение, и до окончания работы блока (до прерывания от блока) все регистры устройства недоступны для записи.

2-й разряд (PACK) определяет режим работы PU распаковки или упаковки:

0 – распаковка,

1 – упаковка.

3-й разряд (SE) определяет режим расширения при распаковке:

0 – расширить нулем,

1 – расширить знаком.

4-й разряд (SAT) определяет режим насыщения при упаковке:

0 – насыщение выключено,

1 –насыщение включено.

5-й разряд (BIT) определяет тип распаковки/упаковки:

0 – обычная распаковка/упаковка,

1 – распаковка/упаковка битовых данных.

6-й разряд (MSB) определяет положение элементов в выходном векторе:

0 – выдача вектора, начиная с младших элементов

1 – выдача вектора, начиная со старших элементов

7-й разряд (MATIN) определяет тип данных на входе PU:

0 – на входе PU вектор данных,

1 – на входе PU матрица данных.

8-й разряд (MATOUT) определяет тип данных на выходе PU для распаковки. Для упаковки данный бит должен быть нулем.

0 – на выходе PU вектор данных,

1 – на выходе PU матрица данных.

9-й разряд (DIAG). Данное поле определяет тип матрицы на входе или выходе. Имеет значение:

0 – получение побочной диагонали,

1 – получение главной диагонали.

10-й разряд (COL). Выдача столбца матрицы:

0 – нет преобразования данных в столбец матрицы,

1 – есть преобразование данных в столбец матрицы.

Поле COLNUM (16-11 разряды) – номер столбца матрицы, в который надо записать элементы; если в столбце больше одного разряда, то это номер младшего разряда столбца.

Поле SHIFT (22 – 17 разряды) определяет начальный сдвиг. При распаковке это величина X, при упаковке это сумма X+Xs

Поле Xs (28 – 23 разряды) – величина Xs при упаковке, при распаковке следует записать 0. Значение поля Xs не может быть больше значения поля SHIFT.

Возможные режимы работы устройства и соответствующие им настройки приведены в таблице 1.14. Все другие сочетания значений являются запрещенными.

Таблица 1.34 – Список операций блока PU

Тип операции \ управляющее поле	PACK	BIT	MATIN	MATOUT	DIAG	COL
Распаковка/переупаковка	0	0/1	0	0	x*	x
Распаковка с получением элементов в главной диагонали матрицы	0	0/1	0	1	1	0
Распаковка с получением элементов в побочной диагонали матрицы	0	0/1	0	1	0	0
Распаковка с получением элементов в столбце матрицы	0	0/1	0	1	x	1
Преобразование матрицы с элементами в главной диагонали в вектор	0	0	1	0	1	0
Преобразование матрицы с элементами в побочной диагонали в вектор	0	0	1	0	0	0
Упаковка	1	0/1	0	0	x	x

* – значение поля не оказывает влияния на результат.

Регистр PUR8. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR8 является периферийным в регистровом окне 1 и доступен по адресу 111110b (pr14)

31–25	24–19	18–12	11–6	5–0
NPE	NPB	Y	MCODE	NCODE

Рисунок 1.34 – Формат регистра PUR8.

Поле NCODE (5 – 0 разряды). Задает разрядность входных данных, возможные значения поля NCODE:

0 – разрядность входных данных 1 бит;
1 – разрядность входных данных 2 бит;
2 – разрядность входных данных 4 бит;
4 – разрядность входных данных 8 бит;
8 – разрядность входных данных 16 бит;
16 – разрядность входных данных 32 бит;
32 – разрядность входных данных 64 бит;

При подаче на вход одноразрядных элементов NCODE = 0 и BIT = 1.

Поле MCODE (11 – 6 разряды). Задает разрядность выходных данных, возможные значения поля MCODE:

0 – разрядность выходных данных 1 бит;
1 – разрядность выходных данных 2 бит;
2 – разрядность выходных данных 4 бит;
4 – разрядность выходных данных 8 бит;
8 – разрядность выходных данных 16 бит;
16 – разрядность выходных данных 32 бит;
32 – разрядность выходных данных 64 бит;

При выдаче одноразрядных элементов M = 0 и BIT = 1.

Поле Y (18 – 12 разряды). Шаг, на который надо сдвинуть входной поток для получения очередного слова для дальнейшей обработки, при распаковке Y = m*N где m = 1, 2, 64/N, в остальных случаях Y = 64. Значение поля Y не может быть равно 0.

Поле NPB (24 – 19 разряды). Значение, которое вычисляется следующим образом:

при упаковке – 64*M/N,
при распаковке – 64*N/M.

Поле NPE (31-25 разряды) Значение, которое вычисляется:

при упаковке – N/M
при распаковке – 0.

Регистр PUR7. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR7 является периферийным в регистровом окне 1 и доступен по адресу 111101b (pr13)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOP																														

Рисунок 1.35 – Формат регистра PUR7.

Поле TOP (30-0 разряды) содержит сдвинутое на один бит в сторону младшего разряда значение максимального адреса при обращении к памяти по шине PU_MB (т.е. разряды адреса с 31-го по 1-й). Нулевой разряд адреса считается равным 0.

Регистр PUR6. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR6 является периферийным в регистровом окне 1 и доступен по адресу 111100b (pr12)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BOTTOM																														

Рисунок 1.36 – Формат регистра PUR6.

Поле BOTTOM (30-0 разряды) содержит сдвинутое на один бит в сторону младшего разряда значение минимального адреса при обращении к памяти по шине PU_MB (т.е. разряды адреса с 31-го по 1-й). Нулевой разряд адреса считается равным 0.

Регистр PUR5. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR5 является периферийным в регистровом окне 1 и доступен по адресу 111011b (pr11)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIRST																														

Рисунок 1.37 – Формат регистра PUR5.

Поле FIRST (30-0 разряды) содержит сдвинутое на один бит в сторону младшего разряда значение первого адреса при обращении к памяти по шине PU_MB (т.е. разряды адреса с 31-го по 1-й). Нулевой разряд адреса считается равным 0. Ограничение: BOTTOM ≤ FIRST ≤ TOP. Адреса генерируются, начиная с FIRST, каждый следующий адрес получается путем инкрементирования, в случае достижения TOP следующий адрес – BOTTOM.

Регистр PUR4. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR4 является периферийным в регистровом окне 1 и доступен по адресу 111010b (pr10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT_IN																															

Рисунок 1.38 – Формат регистра PUR4.

Поле CNT_IN (31-0 разряды) содержит количество входных 64-х разрядных слов, которое должно выдать ядро NMC3 на шину PU_SB (для режима MODE = 01, 11) или которое блок PU должен считать из памяти по шине PU_MB (для режима MODE = 10).

Регистр PUR3. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR3 является периферийным в регистровом окне 1 и доступен по адресу 111001b (pr9)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT_OUT																															

Рисунок 1.39 – Формат регистра PUR3.

Поле CNT_OUT (31-0 разряды) содержит количество выходных слов, которое должен выдать блок PU на шину PU_SB для ядра NMC3 (для режима MODE = 10, 11) или на шину PU_MB для записи в память (для режима MODE = 01). После выдачи количества слов, определяемого полем CNT_OUT, блок PU заканчивает обработку пакета и выдает прерывание ядру NMC3. После этого можно производить настройку и запуск для обработки следующего пакета данных.

Между полями CNT_IN и CNT_OUT есть соотношение при упаковке:

если поле SHIFT регистра PUR9 = 0, то CNT_IN = CNT_OUT * N/M

если поле SHIFT регистра PUR9 > 0, то CNT_IN = CNT_OUT * N/M +1

Если соотношения не выполнены, то возможна некорректная работа блока PU. Например, всегда присылать лишнее слово нельзя, так как при достижении числа уже выданных устройством слов величины CNT_OUT вырабатывается сигнал прерывания, блок PU останавливается, принимает новые значения в конфигурационные регистры, а дополнительное слово данных для предыдущей команды воспримется как первое слово данных для следующей команды.

Регистр PUR2. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR2 является периферийным в регистровом окне 1 и доступен по адресу 111000b (pr8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASKS																															

Рисунок 1.40 – Формат регистра PUR2.

Поле MASKS (31-0 разряды) содержит маску, которая при упаковке и распаковке имеет разное назначение. При упаковке в поле MASKS единица должна быть в самом старшем разряде каждого числа, в остальных разрядах должны быть нули (см. Рисунок 1.41)

31		2M-1	M	M-1	0
10……..…00	...	10……..…00		10……..…00	

Рисунок 1.41. Формат маски MASKS при упаковке.

При распаковке в поле MASKS единицы должны стоять в N младших разрядах каждого распаковываемого числа, нули в остальных разрядах (см. Рисунок 1.42).

31	2M-1	M+N	M+N-1	M	M-1	N	N-1	0
0……..	0…..…..00		1……...11		0…..…..00		1……...11	

Рисунок 1.42. Формат маски MASKS при распаковке.

Регистры PUR1 и PUR0. Регистры доступны на запись для ядра NMC3. Для NMC3 регистры PUR1 и PUR0 являются периферийными в регистровом окне 1 и 84антупны по адресам 110001b (pr1), 110000b (pr0) соответственно.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASKSAT<31:0>																															

Рисунок 1.43 – Формат регистра PUR1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASKSAT<63:32>																															

Рисунок 1.44 – Формат регистра PUR0.

Регистр PUR1 содержит младшую часть, а регистр PUR0 старшую часть 64-х разрядного поля MASKSAT<63:0>. Данное поле используется при упаковке и при получении матрицы с элементами в столбце или диагонали. При всех других операциях данное поле не используется и может быть не инициализировано.

При упаковке в поле MASKSAT<63:0> содержится 64/N последовательно расположенных одинаковых частей. Каждая часть содержит N бит. M-1 младших разрядов этой части заполняются нулями, следующие N-M-Xs бит заполняются единицами и оставшиеся Xs+1 разрядов заполняются нулями (см. Рисунок 1.45).

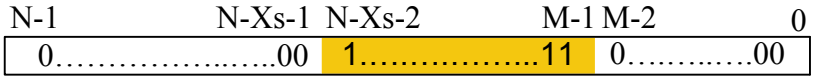


Рисунок 1.45. Формат общей части для получения маски MASKSAT при упаковке.

Оставшиеся разряды маски получаются путем дублирования этой общей части. Ниже приведен пример полученной маски с общей частью 000…111…000 (см. Рисунок 1.46).

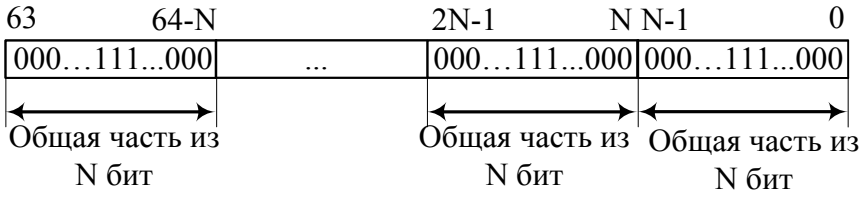


Рисунок 1.46. Пример полученной маски MASKSAT при упаковке.

При получении матрицы с элементами в столбце поле MASKSAT<63:0> содержит единицы в тех разрядах, где должен быть столбец матрицы с данными (см. Рисунок 1.47).

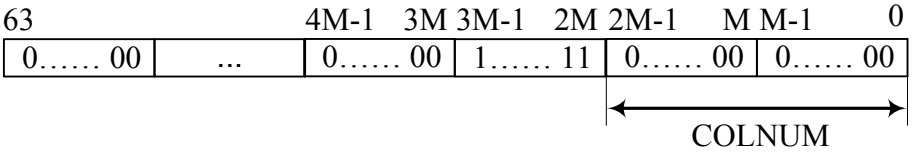


Рисунок 1.47. Пример маски MASKSAT при получении матрицы с элементами в столбце.

При получении матрицы с элементами в главной или побочной диагонали поле MASKSAT<63:0> содержит единицы в M младших разрядах (см. Рисунок 1.48).

63		M	M-1	0
0	……	00	1……	11

Рисунок 1.48. Пример маски MASKSAT при получении матрицы с элементами в главной или побочной диагонали.

1.3.4.3.5 Список периферийных регистров для ядра NMC3

Список всех периферийных регистров для ядра NMC3 приведен в таблице 1.15. Все другие сочетания значений являются запрещенными.

Таблица 1.35 – Список периферийных регистров для ядра NMC3

Код регистра	Регистровое окно 0		Регистровое окно 1	
	Регистр – источник	Регистр – приемник	Регистр – источник	Регистр – приемник
110000	TMR_CNT0	TMR_CNT0		PUR0
110001	TMR_CNT1	TMR_CNT1		PUR1
110010	TMR_MODE0	TMR_MODE0		
110011	TMR_MODE1	TMR_MODE1		
110100	IMR	IMR		
110101	IRR	IRRreset		
110110	IRR	IRRset		
110111	NMCSCR	NMCSCR		
111000				PUR2
111001				PUR3
111010				PUR4
111011				PUR5
111100				PUR6
111101				PUR7
111110				PUR8
111111				PUR9

1.3.4.3.6 Контроллер ПДП (DMAC)

Контроллер ПДП (DMAC) предназначен для пересылки данных между внутренней памятью NMU и внешней по отношению к NMU памятью. Управление осуществляется процессорной системой NMU.

Структурная схема каналов ПДП “память – память” приведена ниже (см. Рисунок 1.49).

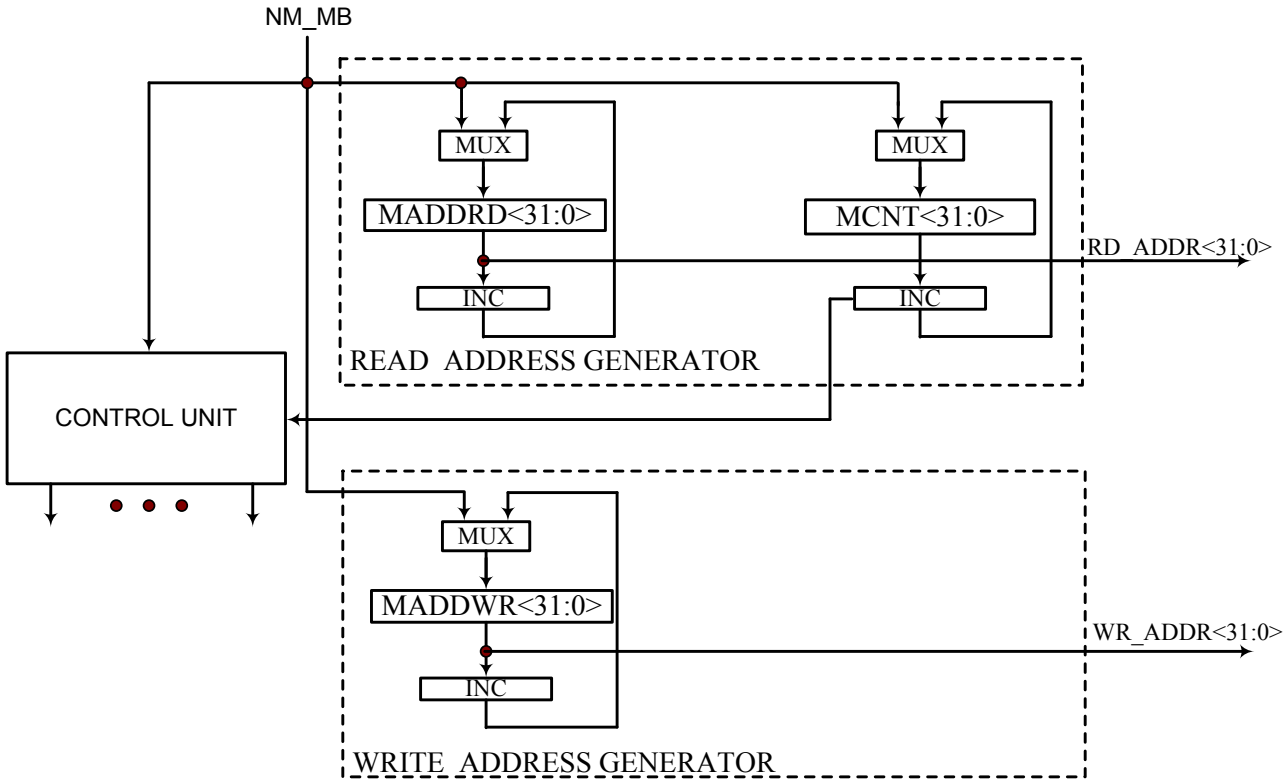


Рисунок 1.49 - Структурная схема каналов ПДП “память – память”.

Каналы ПДП “память – память” содержат следующие функциональные узлы:

READ ADDRESS GENERATOR – генератор адреса для канала ПДП чтения из памяти. Он содержит следующие программно доступные на запись регистры: MADDRD – регистр текущего адреса, который после запуска каналов ПДП формирует в режиме инкремента очередной адрес чтения из памяти, и MCNT – счётчик читаемых 64-х разрядных слов из памяти. Минимальное число слов, которое можно задать для счётчика – 1, максимальное – 232. Чтобы задать число слов для чтения из памяти, равное N, необходимо задать в регистре MCNT число (-1* N) в дополнительном коде. Единице соответствует код FFFF_FFFFh, числу 232 соответствует код 0000_0000h. При работе канала MCNT инкрементируется каждый раз, когда отправлен запрос на чтение от канала ПДП, пока не появится перенос из старшего разряда. В этом случае канал ПДП чтения из памяти останавливается, а MCNT обнуляется.

WRITE ADDRESS GENERATOR – генератор адреса для канала ПДП записи в память. Он содержит только один программно-доступный на запись регистр: MADDWR – регистр текущего адреса, который после запуска каналов ПДП и чтения очередного данного в режиме инкремента каналом чтения, формирует очередной адрес записи в память. Канал ПДП записи в память продолжает свою работу до тех пор, пока не обнулится MCNT и не будут записаны все считанные каналом ПДП чтения из памяти данные. После окончания работы канала записи может быть выдано прерывание процессорной системе NMU.

1.3.4.3.6.1 Конфигурационные регистры блока DMAC

Управление блоком DMAC осуществляться процессорной системой NMU. Чтение и запись конфигурационных регистров блока производится 32-х разрядными скалярными командами

ввода/вывода и равносильно чтению и записи в определенные ячейки памяти. Адреса регистров блока DMAC для процессорной системы NMU приведены ниже в таблице 1.16.

Таблица 1.36 – Адреса регистров блока DMAC в адресном пространстве NMU

	Источник	Адрес вектор hex	Регистр доступен на	
			Чтение	Запись
1	DMACR	4002_0102	+	+
2	MADDRD	4002_00FC		+
3	MADDWR	4002_00FE		+
4	MCNT	4002_0100		+

Регистр управления DMACR (DMA Control Register) (см. Рисунок 1.50) доступен на запись и чтение со стороны процессорной системы NMU. Если было выдано прерывание блоком DMAC, то первая команда записи в регистр DMACR произведет сброс сигнала прерывания. При этом в команде записи должен быть сброшен источник прерывания (бит IRQM) иначе сигнал прерывания взведется в следующем такте.

31–6	5	4	3	2	1	0
зарезервировано	ENM	резерв	MASKM	резерв	IRQM	резерв

Рисунок 1.50 – Формат регистра DMACR

Бит IRQM (1-й разряд) устанавливается после окончания работы ПДП канала записи “память – память” Показывает, что канал ПДП “память – память” закончил свою работу (отправлены последние данные на запись). Если при этом сброшен бит MASKM, то будет выдано прерывание процессорному ядру NMC3. Сброс данного бита осуществляется записью “1” в этот разряд командой записи. Программно установить данный бит в “1” нельзя.

Бит MASKM (3-й разряд) – маска прерывания канала “память – память”.

0 – прерывание не маскируется;

1 – прерывание маскируется.

Бит ENM (5-й разряд) – бит разрешения работы канала “память – память”.

0 – канал “память – память” не отправляет запрос на чтение данных;

1 – канал “память – память” отправляет запрос на чтение данных.

Данный бит устанавливается программно. После этого канал ПДП “память – память” начинает выдавать запросы на чтение данных. После выдачи необходимого числа данных (определяется регистром MCNT) происходит аппаратный сброс данного бита, и канал ПДП больше не выдает запросы на чтение. Сброс данного бита может быть осуществлен программно.

При записи разряды 0-й и 2-й должны быть установлены в единицу, разряд 4 должен быть установлен в ноль.

Регистр MCNT<31:0> (см. Рисунок 1.51) – 32-х разрядный регистр числа читаемых из памяти 64-разрядных слов каналом ПДП “память – память”. Регистр доступен на запись. Минимальное число слов, которое можно задать – 1, максимальное – 232. Чтобы задать число слов для чтения из памяти, равное N, необходимо задать в регистре MCNT число (-1* N) в дополнительном коде. Единице соответствует код FFFF_FFFFh, числу 232 соответствует код 0000_0000h.

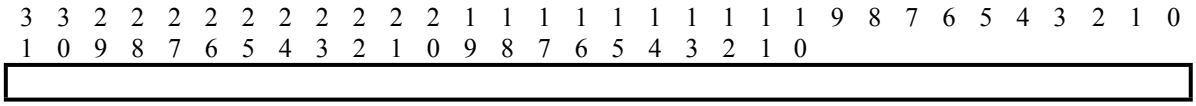


Рисунок 1.51 – Формат регистра MCNT.

Регистр MADDRD<31:0> (см. Рисунок 1.52) – 32-х разрядный регистр текущего адреса чтения канала ПДП “память – память”, который после запуска канала в режиме инкремента формирует очередной адрес чтения ПДП. Регистр доступен на запись.

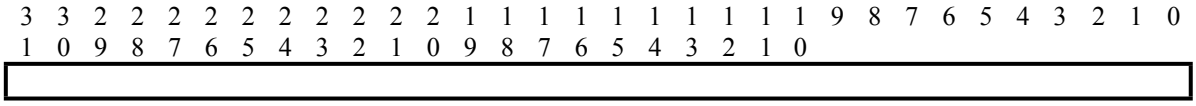


Рисунок 1.52 – Формат регистра MADDRD.

Регистр MADDWR<31:0> (см. Рисунок 1.53) – 32-х разрядный регистр текущего адреса записи. После запуска канала ПДП формирует адрес записи в режиме инкремента Регистр доступен на запись.

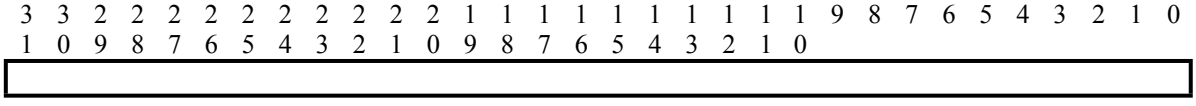


Рисунок 1.53 – Формат регистра MADDWR.

1.3.4.4 Контроллер кэш-памяти узла цифровой обработки сигналов

Для увеличения производительности процессорных ядер NMU в узле цифровой обработки сигналов применяется кэш память второго уровня объёмом 128КБайт.

Ниже приведены его технические характеристики:

размер кэш памяти 128КБайт;

ассоциативный кэш с 4 блоками (4-way);

размер блока 32 КБайт;

длина кэш строки фиксирована и составляет 32 байта;

контроллер имеет набор программно доступных регистров, позволяющих выполнять настройку параметров контроллера.

1.3.4.4.1 Конфигурация кэш памяти

Размер кэш памяти 128Кбайт. Организована кэш память четырьмя блоками (way). В каждом блоке содержится 1024 строки по 32 байта в каждой строке.

1.3.4.4.2 Стратегия замещения

Есть два механизм замещение строк в кэш памяти:

замещается строка, к которой дольше всего не было обращений (LRU);

произвольное замещение.

Выбор механизма замещения осуществляется при помощи поля REPL регистра управления P2CCR

1.3.4.4.3 Стратегия записи данных

Есть два механизма для записи данных:

обратная запись (copy-back);

сквозная запись (write through).

При сквозной записи, происходит модификация данных в кэш памяти и одновременно с этим осуществляется запись во внешнюю память. При обратной записи, происходит модификация данных в кэш памяти, а запись во внешнюю память осуществляется, когда происходит замещение строки. Выбор механизма записи осуществляется при помощи поля WP регистра управления P2CCR.

1.3.4.4.4 Запрет на замещение блока

Один или несколько блоков могут иметь запрет на замещение. Настройка запрета осуществляется при помощи поля LOCK регистра управления P2CCR. Запрет на замещение должен формироваться, начиная со старших блоков. Т.е. сначала запрет устанавливается на блок 4, затем на блок 3 и т.д. После того как запрет на замещение установлен необходимо подать команду полной очистки блока для синхронизации данных.

1.3.4.4.5 Очистка кэш памяти

Есть три варианта очистки кэш памяти:

простая очистка. Сбрасываются разряды, говорящие о том, что данная область кэш памяти содержит какие-либо актуальные данные;

произвести запись из области кэш памяти в основную память (происходит запись только модифицированных данных). При этом данная область кэш памяти остается активной;

полная очистка. При этом происходит запись из области кэш памяти в основную память (происходит запись только модифицированных данных) и сброс разрядов, говорящих о том, что данная область кэш памяти содержит какие-либо актуальные данные.

При этом команда очистки может быть применена ко всей кэш памяти, одному блоку или только к одной строке. При очистке строки, возможно указать как конкретный блок и адрес

строки так адрес памяти, который надо удалить из кэш памяти. Когда необходимо очистить весь кэш, то в поле Mem Addr регистра L2CFM необходимо указать 0.

1.3.4.4.6 Запрет работы с кэш памятью

После системного сброса кэш память не используется. Если необходимо начать использование кэш памяти, то надо установить бит EN регистра P2CCR. Если же в какой-то момент необходимо прекратить использование кэш памяти, то необходимо сделать очистку кэш памяти с установленным битом DI регистра L2CFM или L2CFT.

1.3.4.4.7 Информация о работе кэш памяти

Для получения информации о количестве кэш промахов или попаданий предусмотрены регистры, которые содержать число обращений в кэш память (L2CACR) и число попаданий в кэш память (L2CHCR). При записи значения ноль в регистр L2CACR происходит сброс обоих внутренних счетчиков (числа обращений и промахов) и перезапись значений счетчиков, которое было до сброса, в данные регистры. После этого значения могут быть считаны программно. Возможны два режима работы:

циклический режим;

режим сдвига.

При циклическом режиме после достижения счетчиком числа обращений в кэш память максимального значения, счетчик числа обращений и счетчик числа попаданий в кэш память обнуляются и начинают отсчет заново. В режиме сдвига при достижении счетчиком числа обращений в кэш память максимального значения происходит сдвиг на 16 бит в сторону младших разрядов счетчика числа обращений и счетчика числа попаданий в кэш память.

Аналогично работают счетчик числа тактов шины и счетчик числа активных тактов шины.

1.3.4.4.8 Конфигурационные регистры блока кэш памяти.

В таблице 1.17 приведены программно доступные регистры контроллера кэш памяти.

Таблица 1.37 – Адреса регистров контроллера кэш памяти в адресном пространстве NMU

	Источник	Адрес вектор hex	Регистр доступен на	
			Чтение	Запись
1	регистр управления L2CCR	0E20_0000	+	+
2	регистр конфигурации L2CSR	0E20_0001	+	
3	регистр очистки кэш-памяти по адресу L2CFM	0E20_0002	+	+
4	регистр очистки кэш памяти по номеру блока или строке L2CFT	0E20_0003	+	+
5	регистр счетчика доступов к кэш память L2CACR	0E20_0004	+	+
6	регистр счетчика попаданий в кэш память L2CHCR	0E20_0005	+	
7	регистр счетчика тактов шины между кэш памятью и процессорными системами NMU L2CFsCR	0E20_0006	+	+
8	регистр счетчика активных тактов шины между кэш памятью и процессорными системами NMU L2CFsUCR	0E20_0007	+	

Далее приведено подробное описание полей регистров контроллера кэш памяти.

Таблица 1.38 – Поля регистра управления (L2CCR)

Биты	Поле	Описание
0	резерв	при записи должно быть 0
1	WP	настройка режима записи: 0 - обратная запись (copy-back); 1 - сквозная запись (write through);
2	HC	Режим работы счетчика кэш-попадания 0 - циклический режим; 1- режим сдвига.
3	UC	Режим работы счетчика запросов на шине: 0 - циклический режим; 1- режим сдвига.
7:4	резерв	при записи должно быть 0
11:8	LOCK	Настройка запрета на замещение блока: 0 - замещение i-го блока разрешено: 1 - замещение i-го блока запрещено.
15:12	резерв	при записи должно быть 0
29:28	REPL	Настройка режима замещения блоков: 00: замещается строка, к которой дольше всего не было обращений (LRU); 01: произвольное замещение;
30:	резерв	при записи должно быть 0
31	EN	разрешение работы кэш-памяти: 0: кэш не работает; 1: кэш работает.

Таблица 1.39 – Поля регистра конфигурации (L2CSR)

Биты	Поле	Описание
31:25	резерв	
24	LSIZE	размер строки 32 байта
23	резерв	
22	EDAC	0 - защита памяти в блоке не реализована
21:16	MTRR	2'b0
15:13	BBus_width	2'b10 - ширина шины 64 бита
12:2	резерв	
1:0	Way	2b'11 - ассоциативный кэш с 4 блоками (4-way).

Таблица 1.40 – Поля регистра очистки кэш памяти по адресу (L2CFM)

Биты	Поле	Описание
31:5	Mem Addr	адрес памяти, который надо вычистить из кэша. Если надо вычистить все строки кэша, то необходимо указать 0.
4	резерв	
3	DI	запрет работы кэша (равносилен сбросу бита EN в регистре управления): 0 - кэш работает; 1 - запрет работы кэша
2:0	FLUSH	Режим очистки кэша 3'b001 - очистить строку; 3'b010 - переписать строку из кэша в память (переписываются только модифицированные стоки); 3'b011 - переписать строку из кэша в память (переписываются только модифицированные стоки) и очистить строку; 3'b101 - очистить все строки; 3'b110 - переписать все строки из кэша (переписываются только модифицированные стоки) в память; 3'b111 - переписать все строки из кэша в память(переписываются только модифицированные стоки) и очистить их;

Таблица 1.41 – Поля регистра очистки кэш памяти по номеру блока или строке (L2CFT)

Биты	Поле	Описание
31:10	cash line index/TAG	определяет тэг очищаемого блока или определенную строку в блоке. При очистке блока: [31:10] определяет тэг очищаемого блока, в случае очистки блока или При очистке конкретной сроки: [31:16] определяет конкретную строку, [15:10] = 6'b0;
9	Fetch	выборка из памяти во время очистки: 0 - выборки данных нет, или очищается конкретная строка; 1- данные выбираются из памяти во время очистки блока.
8	Valid	Поле используется только при блочной очистке. 0 - очистить конкретную строку; 1 - очистить только активные строки блока при блочной очистке;
7	Dirty	Поле используется только при блочной очистке. 0 - очистить конкретную строку; 1 - очистить только измененные строки блока при блочной очистке;
6	резерв	
5:4	Way	номер блока (way), в котором надо очистить строку; 2'b00 - очистить way 0; 2'b 01 - очистить way 1; 2'b 10 - очистить way 2; 2'b 11 - очистить way 3;
3	DI	запрет работы кэша (равносилен сбросу бита EN в регистре управления): 0 - кэш работает; 1 - запрет работы кэша.
2	WF	0 - очистка конкретной строки; 1 - очистка блока;
1:0	FLUSH	Режим очистки кэша. очистка строки 2'b00 - очистить строку; 3'b10 - переписать строку из кэша в память; 3'b11 - переписать строку из кэша в память и очистить строку; очистка блочная 2'b00 - очистить в соответствии с установкой [8:7] разрядов; 3'b10 - переписать измененные строки блока из кэша в память; 3'b11 - очистить в соответствии с установкой [8:7] разрядов и переписать измененные строки блока из кэша в память;

Таблица 1.42 – Поля регистра счетчика доступов к кэш память (L2CACR)

Биты	Поле	Описание
31:0	Access counter	счетчик запросов к кэш памяти. Запись нуля в данный регистр обнуляет теневой регистр счетчика доступов к кэш памяти и переписывает значение из теневого в данный регистр.

Таблица 1.43 – Поля регистра счетчика попаданий в кэш память (L2CHCR)

Биты	Поле	Описание
31:0	Hit counter	счетчик числа попаданий в кэш память. Запись нуля в регистр L2CACR обнуляет теневой регистр счетчика попаданий в кэш память и переписывает значение из теневого в данный регистр

Таблица 1.44 – Поля регистра счетчика тактов шины между кэш памятью и процессорными системами NMU (L2CFsCR)

Биты	Поле	Описание
31:0	bus cycle counter	счетчик тактов шины. Запись нуля в данный регистр обнуляет теневой регистр счетчика и переписывает значение из теневого в данный регистр

Таблица 1.45 – Поля регистра счетчика активных тактов шины между кэш памятью и процессорными системами NMU (L2CFsUCR)

Биты	Поле	Описание
31:0	bus usage counter	счетчик активных тактов шины. Запись нуля в регистр L2CFsCR обнуляет теневой регистр счетчика активных тактов шины и переписывает значение из теневого в данный регистр

1.3.5 Подсистема видео

Подсистема видео предназначена для обеспечения вывода на внешнее устройство отображения видео и графической информации с помощью выходного цифрового интерфейса. А также для захвата видео или графического изображения, передаваемого с внешнего устройства с помощью входного цифрового интерфейса.

На рисунке 1.54 приведена структурная схема подсистемы. Подсистема видео состоит из двух типов устройств:

- VDU (видеоконтроллер высокого разрешения);
- GRAB (устройство захвата изображения).

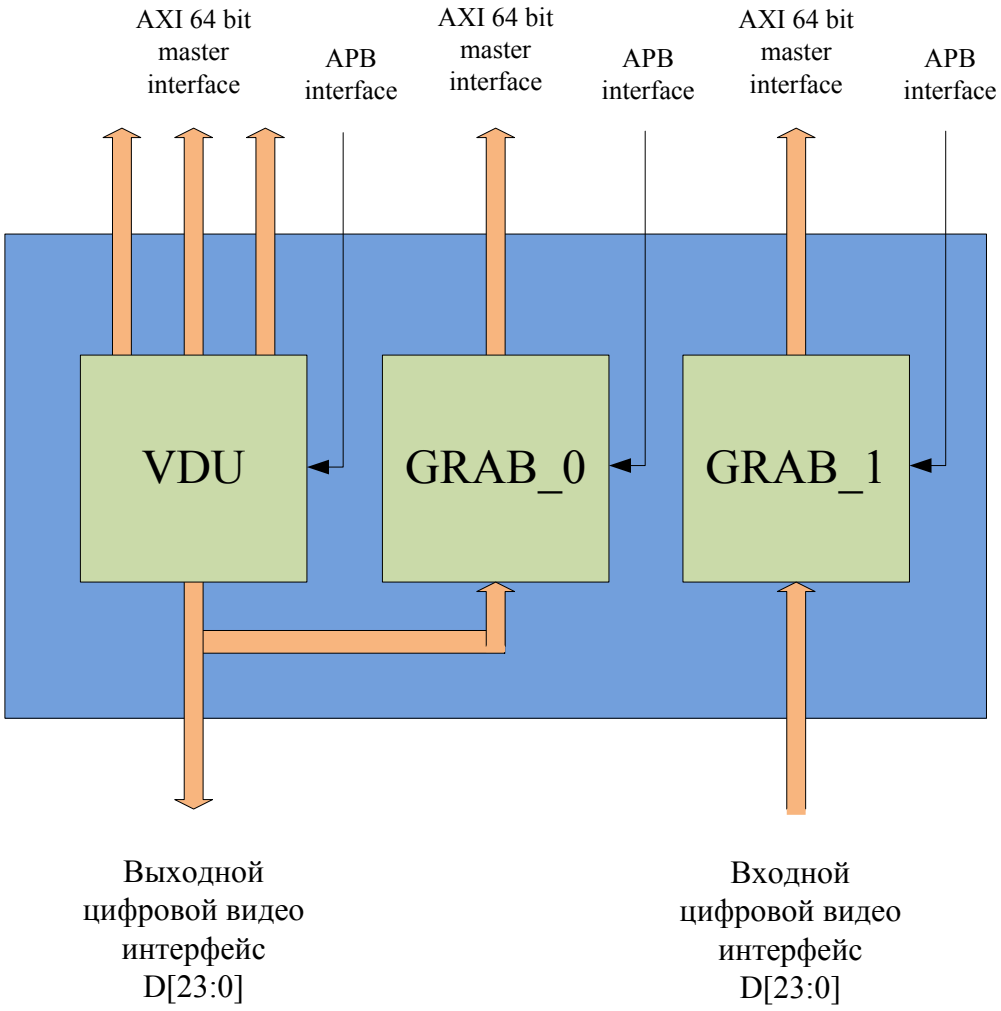


Рисунок 1.54 - Структурная схема подсистемы видео

1.3.5.1 Видеоконтроллер высокого разрешения

1.3.5.1.1 Общее описание

1.3.5.1.1.1 Описание назначения

Видеоконтроллер высокого разрешения рассчитан на применение в составе СнК с системной шиной AMBA 3.0 (AXI) высокой степени интеграции и может использоваться в цифровых фотоаппаратах, видеокамерах, DVD-плеерах, телевизионных приставках и цифровых телевизорах (см. рисунок 1.55).

Контроллер предназначен для формирования видеосигнала во всем спектре телевизионных разрешений, вплоть до разрешений высокой четкости, поддерживает аппаратное наложение слоя графики и меню на основное видеоизображение с программируемым коэффициентом прозрачности, а также позволяет масштабировать видеоизображение с произвольными коэффициентами по вертикали и горизонтали. Контроллер отличается небольшой площадью, занимаемой на кристалле, и по своим характеристикам соответствует образцам IP-блоков, доступным в данный момент на международном рынке интеллектуальной собственности. Контроллер имеет стандартные внешние интерфейсы, позволяющие разработчикам СнК интегрировать его в свои проекты с минимальными затратами.

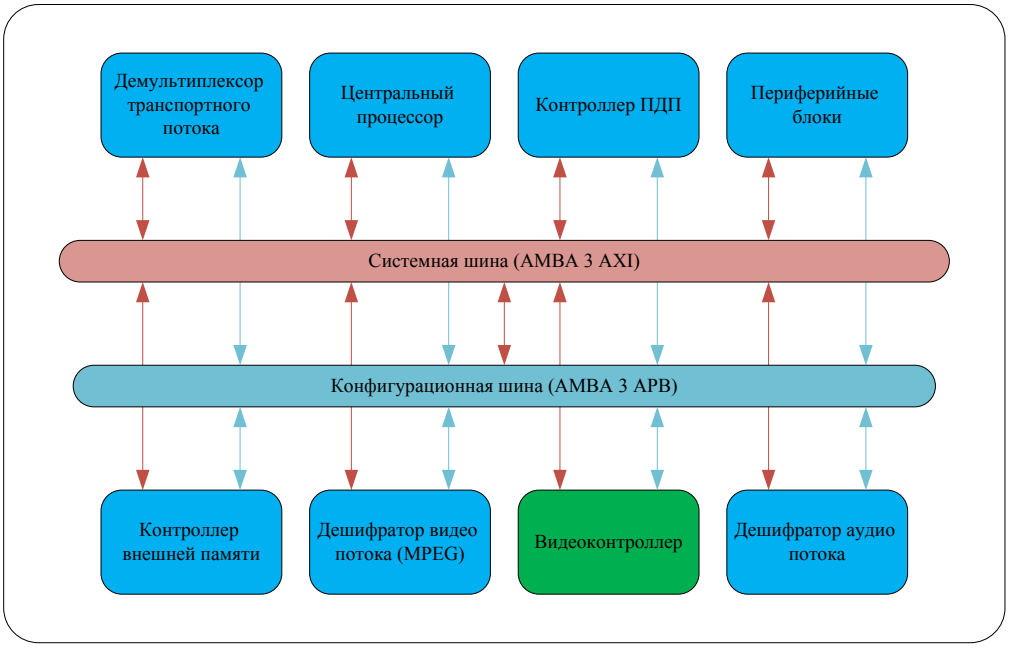


Рисунок 1.55 - Схема типового применения (в типовом применении обязательно наличие видеоконтроллера, контроллера памяти, процессора, а также шин AXI и APB)

На рисунке 1.56 прдставлено типичное изображение, которое может быть выведено на экран дисплея видеоконтроллером.

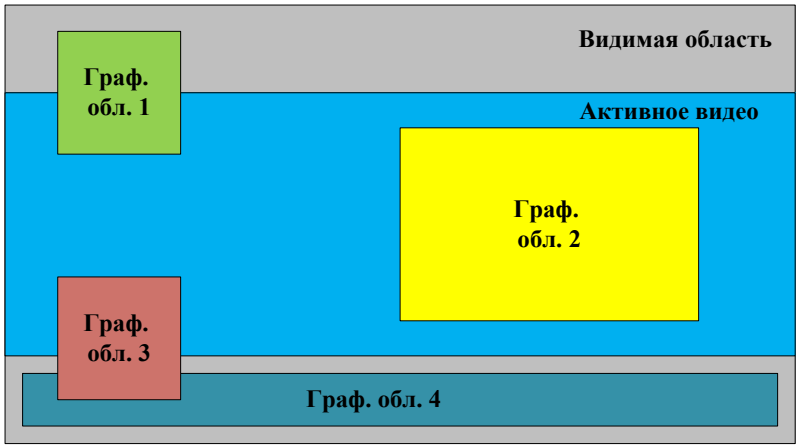


Рисунок 1.56 - Пример построения изображения на экране дисплея

Видимая область определяет размеры итогового изображения на экране средства отображения. При этом цвет видимой области (цвет фона) задается программно в регистрах видеоконтроллера.

Область активного видео – это область, в которую осуществляется вывод изображения из основного видеослоя, эта область непрозрачна и располагается поверх видимой области.

Графические области 1-4 относятся к графическим слоям и располагаются поверх области активного видео. Для графических областей могут быть заданы коэффициенты прозрачности. Размеры и положение областей могут варьироваться, при этом допускается перекрытие графических областей по горизонтали при использовании вывода через разные графические слои.

1.3.5.1.1.2 Технические характеристики

Технические характеристики устройства Видеоконтроллера делятся на две группы:

- технологически не зависимые характеристики;
- технологически зависимые характеристики.

К первой группе относятся следующие характеристики устройства:

- поддерживаемые форматы хранения видеоданных:
  - линейное с двух плоскостным и трех плоскостным хранением компонент,
  - макроблочное с двух плоскостным хранением компонент;
- формат представления компонент YCbCr видеоданных:
  - 4.2.0,
  - 4.2.2;
- поддерживаемые форматы хранения графических данных:
  - 5:6:5,
  - 1:5:5:5,
  - 4:4:4:4,
  - 8:8:8:8;
- поддерживаемые разрешения вывода видеоданных:
  - SDTV 720x480i 60Hz,
  - SDTV 720x576i 50Hz,
  - EDTV 720(1440)x480i 60Hz,
  - EDTV 720(1440)x576i 50Hz,
  - EDTV 720x480p 60Hz,
  - EDTV 720x576p 50Hz,
  - HDTV 1280x720p 60Hz,
  - HDTV 1280x720p 50Hz,
  - HDTV 1920x1080i 60Hz,
  - HDTV 1920x1080i 50Hz,
  - HDTV 1920x1080p 30Hz,
  - HDTV 1920x1080p 25Hz;
  - HDTV 1920x1080p 60Hz,
  - HDTV 1920x1080p 50Hz;
- системная шина данных AMBA AXI с разрядностью по адресам 32 бит, по данным 64 бита;

Ко второй группе относятся следующие характеристики устройства, все значения приведены при условии реализации устройства на технологическом процессе Fujitsu с проектными нормами 65 нм:

- системная частота – 196 МГц;
- частота вывода видеоданных для высокого разрешения – 148,5МГц;
- частота вывода видеоданных для высокого разрешения – 74,25МГц;
- частота вывода видеоданных для стандартного разрешения – 27 МГц;

1.3.5.1.1.3 Структурная схема

Структурная схема СФ блока ВКВР приведена на рисунке 1.57.

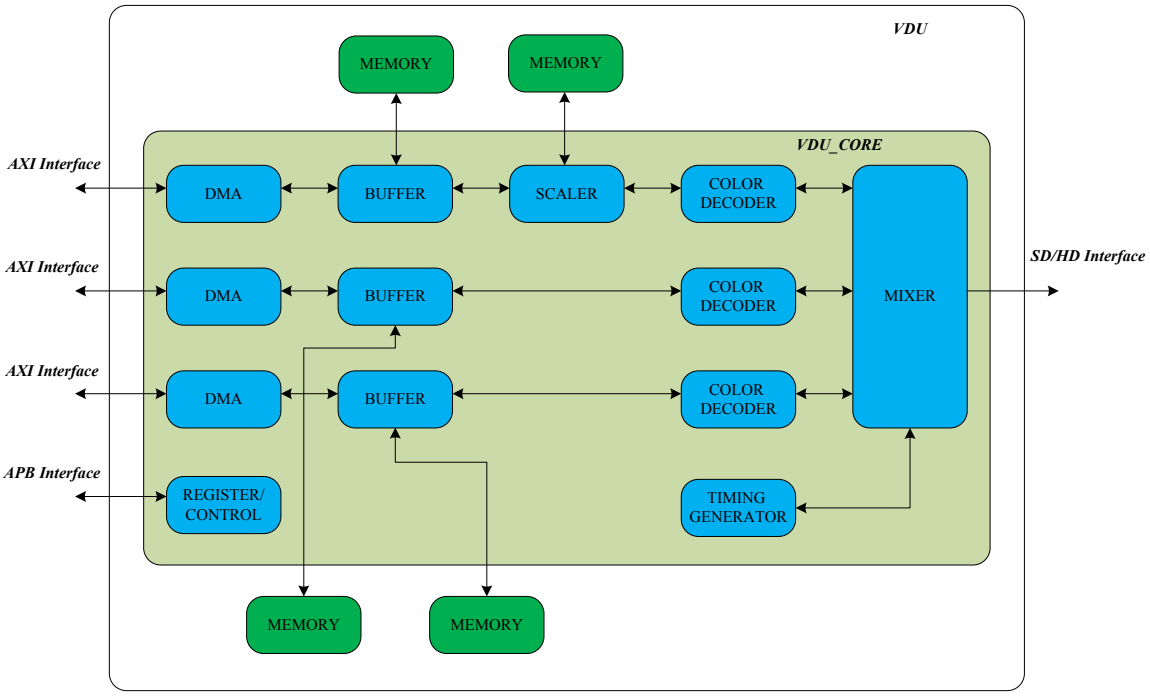


Рисунок 1.57 - Структурная схема видеоконтроллера высокого разрешения

СФ-блок видеоконтроллера высокого разрешения является параметризируемым и построен по модульному принципу, что позволяет на его базе формировать различные конфигурации, оптимально соответствующие различным вариантам его применения. Возможно изменение объемов используемой буферной памяти и исключение из тракта обработки ряда блоков, таких как устройство масштабирования, поддержка разрешений высокой четкости и др.

REGISTER/CONTROL – блок системного интерфейса управления обеспечивает прием и выдачу конфигурационной информации видеоконтроллера по запросу внешнего ведущего устройства. Реализуется контроллером ведомого на шине AMBA APB, в соответствии со спецификацией AMBA™ 3 APB Protocol v1.0. Управляет функционированием блоков видеоконтроллера и их взаимодействием, обеспечивает запись/считывание управляющей информации в/из регистров видеоконтроллера посредством системного интерфейса, управляет режимами работы видеоконтроллера, преобразует внутренние управляющие сигналы и вырабатывает сигналы прерываний. Набор внутренних регистров видеоконтроллера, обеспечивает хранение управляющей информации и внутреннего состояния видеоконтроллера.

DMA – контроллер прямого доступа в память основного видеослоя – обеспечивает генерацию запросов в память для загрузки видеоданных в канал основного видеослоя. На основе внутренней информации генерирует адреса для обращения в видеопамять за порцией видеоданных, определяет размер порции. Имеет интерфейс AMBA® AXI Protocol v1.0.

BUFFER – обеспечивает буферизацию для нивелирования возможной задержки поступления видеоданных по шине. Осуществляет преобразование данных из форматов интерфейсной шины в форматы, используемые при обработке в канале слоя.

COLOR DECODER – преобразователь форматов представления цвета в канале слоя – видеоконтроллер поддерживает режим телевидения стандартной четкости, и режим телевидения высокой четкости, при этом, обеспечивая перевод видеоизображения между ними, данные форматы цифрового телевидения имеют разные колориметрические характеристики. Блок выполняет преобразование формата Y601CbCr в Y709CbCr и наоборот. Внутренним форматом представления цвета в видеоконтроллере является формат YCbCr 4:4:4, поэтому входные данные графического слоя, представленные в форматах (A)RGB необходимо преобразовать в Y601CbCr 4:4:4 (при работе в режиме телевидения стандартной четкости), или Y709CbCr 4:4:4 (в режиме

телевидения высокой четкости). При этом значение прозрачности выносится в отдельный канал и подается на блок наложения и позиционирования видеослоев.

SCALER – устройство масштабирования видеоизображения - осуществляет, как увеличение разрешения исходного изображения (upscaling), так и уменьшение (downscaling). Увеличение разрешения проводится по методу линейной интерполяции, уменьшение - выполняется методом усреднения по нескольким ближайшим точкам. Видеоконтроллер позволяет выполнять масштабирование изображения с произвольными значениями коэффициентов, однако стабильная работа видеоконтроллера гарантируется только с определенным набором коэффициентов, необходимыми для осуществления преобразований между разрешениями форматов входного и выходного изображения, являющихся рабочими для видеоконтроллера. Значения коэффициентов масштабирования ограничены также желаемым качеством выходного изображения.

Для устранения возможных искажений изображения при его масштабировании в системе, производящей масштабирование могут быть применены дополнительные средства, например, предварительная сглаживающая фильтрация перед уменьшением разрешения изображения.

Осуществляет преобразование форматов представления входной видеоинформации из форматов YCbCr 4:2:0 и YCbCr 4:2:2 во внутреннее представление YCbCr 4:4:4.

Осуществляет симметричную обрезку изображения по вертикали и горизонтали.

MEMORY – блок предназначен для временного хранения строк изображения.

MIXER – блок наложения и позиционирования видеослоев. Данный блок реализует совмещение слоев видеоизображения: основного видеослоя, двух графических слоев и фонового слоя (задаваемого в регистрах видеоконтроллера). Для получения итогового изображения необходимо произвести наложения данных слоев изображения друг на друга в следующем порядке: фоновый, основной, графический. Основной слой всегда непрозрачен и накладывается на фоновый слой. Также данный блок производит встраивание результирующего изображения во временную развертку, генерируемую генератором временных разверток.

TIMING GENERATOR – генератор временных разверток; обеспечивает генерацию сигналов в соответствии со стандартами ITU-R BT.656 и EIA/CEA-861-B.

1.3.5.1.1.4 Описание входных форматов основного видео слоя

Основной видеослой (MVL) предназначен для отображения основного видеопотока. Видеоконтроллер выводит видеоданные, хранимые во внешней памяти, на один из своих цифровых видеоинтерфейсов: стандартной или высокой четкости. Для основного видеослоя видеоконтроллера поддерживаются следующие форматы представления входных данных:

- YCbCr 4:2:0;
- YCbCr 4:2:2.

Видеоданные видеопотока, хранимые во внешней памяти, разделены на яркостную и цветоразностную плоскости (плоскости компонент), которые хранятся в видеопамяти независимо. При этом порядок данных внутри плоскости может быть различным.

Видеоконтроллером поддерживается два режима разделения кадра на плоскости компонент:

- режим двойного разделения;
- режим тройного разделения.

Видеоконтроллер поддерживает два режима вывода видео данных:

- прогрессивный;
- чересстрочный.

При чересстрочный режиме, каждая плоскость компонент может храниться в памяти в виде одной области, так называемого кадра (как в прогрессивном режиме), так и в виде двух областей (полукадров).

Видеоконтроллером поддерживаются два режима порядка данных внутри плоскости:

- линейный режим;
- макроблочный режим.

Для макроблочного режима порядка данных внутри плоскости не поддерживается режим тройного разделения кадра на плоскости.

Предполагается, что видеопамять состоит из 64х разрядных слов.

1.3.5.1.1.4.1 Режимы разделения кадра на плоскости

В случае режима двойного разделения кадра на плоскости компонент действительны следующие утверждения, приведенные в данном разделе.

Видеопоток представлен в виде последовательности кадров, которые хранятся в памяти раздельно.

Независимо от того, в каком виде представлена исходная видеопоследовательность (прогрессивном или чересстрочном) и от режима отображения (прогрессивный, чересстрочный) все яркостные составляющие (Y) для кадра хранятся в отдельной яркостной плоскости и все цветоразностные составляющие (Cb, Cr) кадра хранятся в отдельной цветоразностной плоскости. В случае чересстрочного режима строки четного и нечетного полукадров хранятся в чередующемся манере, начиная сверху в одной плоскости.

Яркостная и цветоразностная плоскости называются - плоскости компонент и хранятся в памяти раздельно. Таким образом, видеоконтроллеру требуется задать два базовых адреса (один базовый адрес для яркостной плоскости и один - для цветоразностной плоскости) для обращения к видеокадру.

Внутри цветоразностной плоскости Cb и Cr составляющие хранятся в виде чередующихся CbCr пар. CbCr пары занимают по два байта и имеют выровненные по 2 байта адреса, при этом Cb составляющая находится в младшем (четном) байте, а Cr составляющая расположена в старшем (нечетном) байте.

В случае режима тройного разделения кадра – цветоразностная плоскость разделяется на две плоскости: одну плоскость для Cb составляющей и одну плоскость для Cr составляющей. При этом видеоконтроллеру требуется уже три базовых адреса (один для яркостной плоскости и два для цветоразностных плоскостей) для доступа к видеокадру.

Для каждого базового адреса плоскостей принято выравнивание по 1 КБайтной границе.

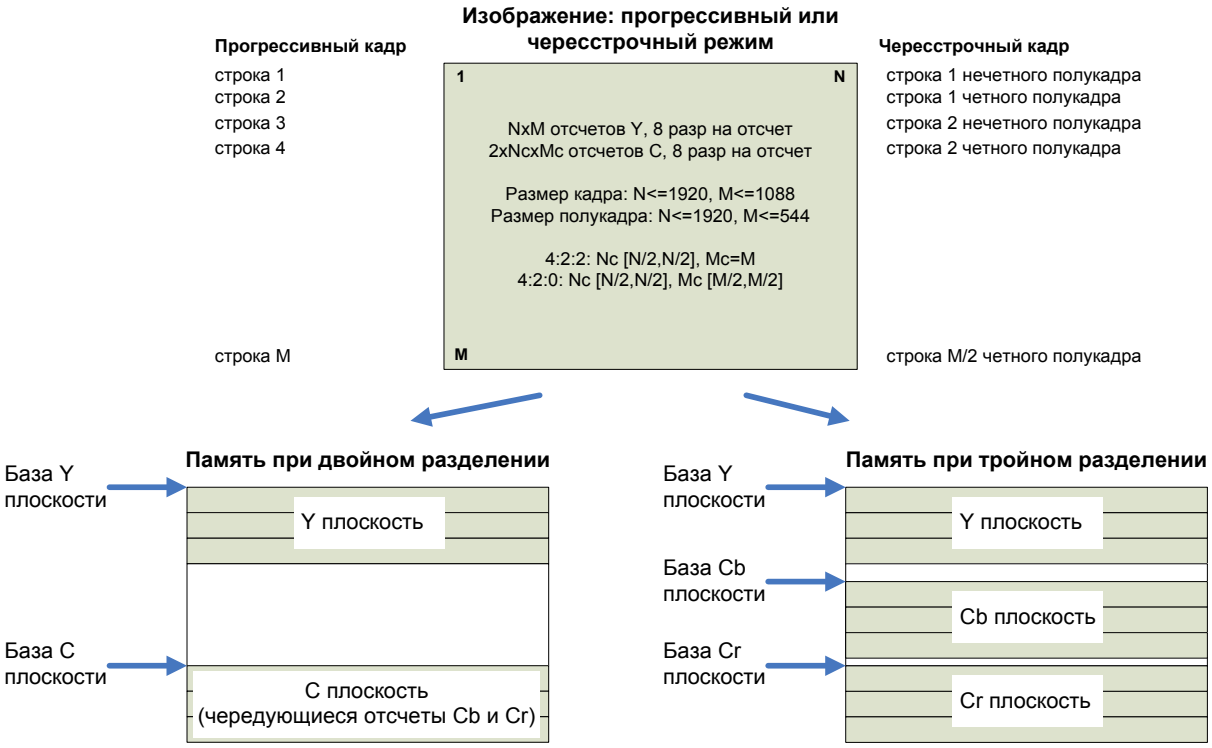


Рисунок 1.58 - Режимы разделения кадра на плоскости компонент

Порядок данных внутри плоскости

1.3.5.1.1.4.2 Линейный режим

Изображения внутри плоскости компоненты расположено следующим образом:
строки последовательно расположены сверху вниз;
внутри каждой строки отсчеты расположены последовательно слева направо.

В случае чересстрочного режима изображения строки четного и нечетного полукадров представлены в одной плоскости в виде четных и нечетных строк одного кадра (четные строки плоскости относятся к четному полукадру, нечетные строки – к нечетному полукадру).

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32
P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47	P48

Рисунок 1.59 - Положение отсчетов на экране

64х разрядное слово видеопамяти может содержать максимально до восьми 8ми разрядных отсчетов для плоскости компоненты. Внутри слова видеопамяти первый отсчет (по правилам чтения изображения) содержится в байте с младшим адресом, т.е. в младшем байте LSB для формата little endian format, и в старшем байте MSB для формата big endian format. Байты по

следующим адресам содержат отсчеты в соответствии с правилами чтения изображения в плоскости.

Рассмотрим порядок данных внутри плоскости на следующем примере (см. рисунок). Базовый адрес плоскости равняется n. Размер плоскости N=16, M=3, Длина строки N равняется физическому размеру строки.

MSB		QWORD = 64 разр				LSB		
0	1	2	3	4	5	6	7	смещение в байтах
P1	P2	P3	P4	P5	P6	P7	P8	n
P9	P10	P11	P12	P13	P14	P15	P16	n + 8
P17	P18	P19	P20	P21	P22	P23	P24	n + 16
P25	P26							n + 24
P33	P34							n + 32
P41	P42					P47	P48	n + 40

Рисунок 1.60 - Отображение плоскости в памяти при линейном режиме (big endian format)

MSB		QWORD=64 разр				LSB		
7	6	5	4	3	2	1	0	смещение в байтах
P8	P7	P6	P5	P4	P3	P2	P1	n
P16	P15	P14	P13	P12	P11	P10	P9	n + 8
P24	P23	P22	P21	P20	P19	P18	P17	n + 16
						P26	P25	n + 24
						P34	P33	n + 32
P48	P47					P42	P41	n + 40

Рисунок 1.61 - Отображение плоскости в памяти при линейном режиме (little endian format)

1.3.5.1.1.4.3 Макроблочный режим

Плоскость представлена в виде макроблоков (см. рисунок 1.62), состоящих из 16 строк по 16 отсчетов в каждой. Макроблоки объединяются в строки, в которых макроблоки содержаться последовательно слева направо. Плоскость состоит из набора строк макроблоков, которые идут последовательно сверху вниз. При этом количество макроблоков в строке Nm = N/16 (N – размер строки плоскости компонента), а количество строк макроблоков Mm=M/16.

Внутри каждого макроблока изображение представлено аналогично линейному режиму.

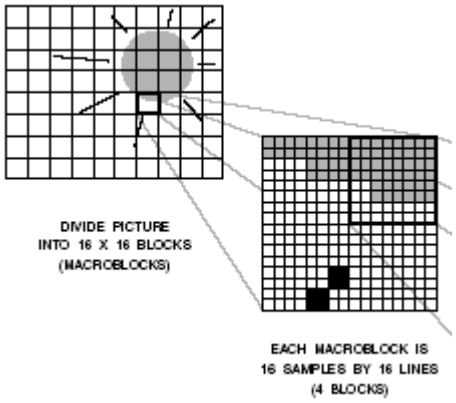


Рисунок 1.62 - Представление изображения макроблоками

В случае чересстрочного режима изображения строки четного и нечетного полукадров представлены в одной плоскости в виде четных и нечетных строк одного макроблока (первые 8 строк (0-7) относятся к четному полукадру, строки 8-15 – к нечетному полукадру).

Отображение плоскости при макроблочном режиме будет иметь вид, представленный на рисунках ниже 1.63 и Рисунок 1.64.

Пример: плоскость, имеющая размеры в макроблоках Nm=2, Mm=2. Длина строки Nm равняется физическому размеру строки. На рисунке 1.63 цветом выделены строки памяти, содержащие первую и семнадцатую строки изображения. Возможно хранение информации в памяти как в формате little endian format, так и в формате big endian format.

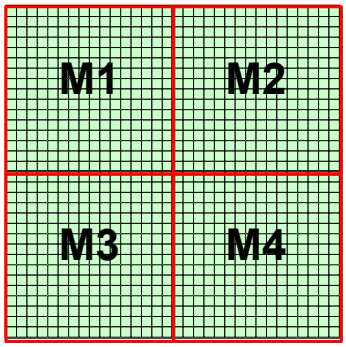


Рисунок 1.63 - Положение макроблоков на экране

MSB		*QWORD=64 разр*				*LSB*		
7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*смещение в байтах*
M1_8	*M1_7*	*M1_6*	*M1_5*	*M1_4*	*M1_3*	*M1_2*	*M1_1*	*n*
M1_16	*M1_15*	*M1_14*	*M1_13*	*M1_12*	*M1_11*	*M1_10*	*M1_9*	*n+8*
M1_256	*M1_255*	*M1_254*	*M1_253*	*M1_252*	*M1_251*	*M1_250*	*M1_249*	*n+32*8*
M2_8	*M2_7*	*M2_6*	*M2_5*	*M2_4*	*M2_3*	*M2_2*	*M2_1*	*n+33*8*
M2_16	*M2_15*	*M2_14*	*M2_13*	*M2_12*	*M2_11*	*M2_10*	*M2_9*	*n+34*8*
M2_256	*M2_255*	*M2_254*	*M2_253*	*M2_252*	*M2_251*	*M2_250*	*M2_249*	*n+64*8*
M3_8	*M3_7*	*M3_6*	*M3_5*	*M3_4*	*M3_3*	*M3_2*	*M3_1*	*n+65*8*
M3_16	*M3_15*	*M3_14*	*M3_13*	*M3_12*	*M3_11*	*M3_10*	*M3_9*	*n+66*8*
M3_256	*M3_255*	*M3_254*	*M3_253*	*M3_252*	*M3_251*	*M3_250*	*M3_249*	*n+96*8*
M4_8	*M4_7*	*M4_6*	*M4_5*	*M4_4*	*M4_3*	*M4_2*	*M4_1*	*n+97*8*
M4_16	*M4_15*	*M4_14*	*M4_13*	*M4_12*	*M4_11*	*M4_10*	*M4_9*	*n+98*8*
M4_256	*M4_255*	*M4_254*	*M4_253*	*M4_252*	*M4_251*	*M4_250*	*M4_249*	*n+128*8*

Рисунок 1.64 - Отображение плоскости в памяти при макроблочном режиме (little endian format)

Режим тройного разделения плоскости на компоненты не доступен при макроблочном режиме.

1.3.5.1.1.4.4 Доступ к плоскостям компонент

Область доступа в системной видеопамяти задается следующими параметрами (при этом базовый адрес (указатель на начало плоскости компоненты видеокадра) не указывается):

- mvl_line_size – размер строки доступной области в плоскости, в пикселах. При макроблочном режиме – кол-во макроблоков = mvl_line_size/16;
- mvl_num_row – количество строк в доступной области плоскости;
- mvl_full_line_size (>= mvl_line_size) – полный размер строки плоскости в пикселах, может быть равен или больше размера строки доступной области;
- mvl_start_pos – адрес, указывающий на начало доступной области в системной памяти;
- field_access – тип изображения: прогрессивный, чересстрочный.

Ситуация, когда внутри плоскости компоненты в памяти существуют недоступные позиции (изображение, считываемое из памяти меньше кадра) возможна в том случае, если выполняется одно из следующих условий:

- mvl_full_line_size > mvl_line_size;
- mvl_num_row < M (M – количество строк в плоскости компонента кадра).

Пояснением вышеуказанных параметров служит рисунок 1.65.

Ограничения для параметров доступа к области памяти:

- Для параметров mvl_line_size и mvl_num_row нет никаких ограничений, кроме того, что они должны быть четными значениями.
- Параметр mvl_full_line_size для линейного режима (при задании в пикселах) должен быть кратен 16 (размеру строки макроблока). При макроблочном режиме

параметр mvl_full_line_size должен быть кратен 64 (Y или 4:2:2 CbCr плоскости) или 128 (4:2:0 CbCr плоскость). Таким образом, начальный (самый левый) макроблок в строке макроблоков имеет адрес, выровненный по 1 КБайтной границе.

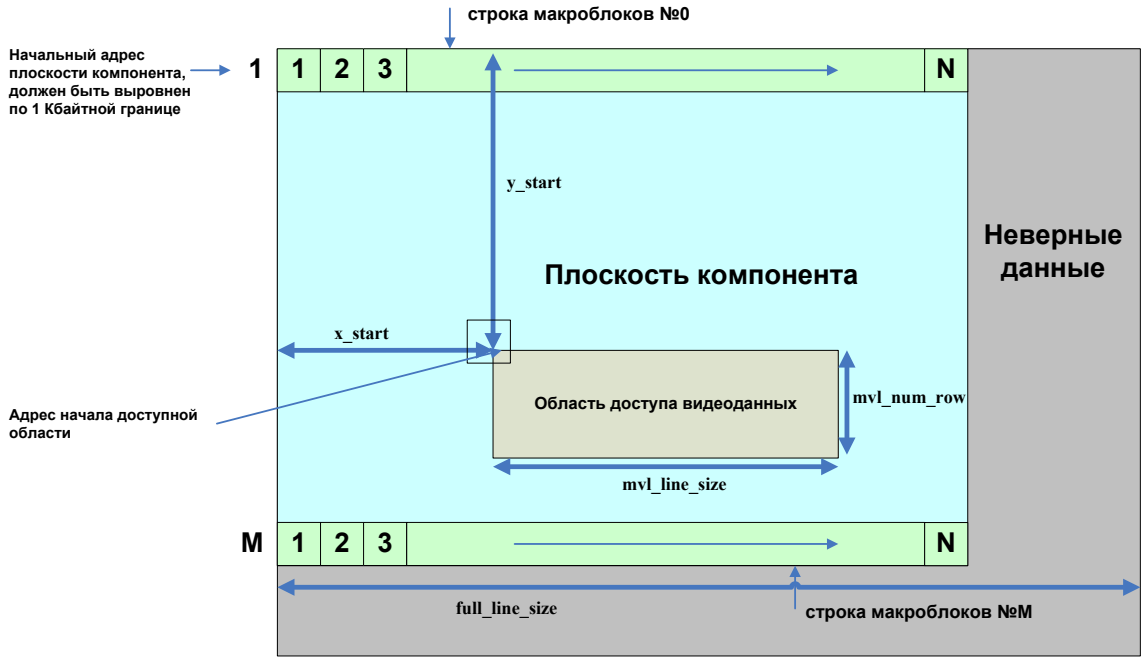


Рисунок 1.65 - Иллюстрация параметров для обращения в область доступа системной видеопамяти

Параметр видеоконтроллера - field_access может быть установлен для доступа только к четным, или нечетным строкам изображения в плоскости компонента (тип считываемых строк определяется: field_access = “00” – чтение в прогрессивном режиме, field_access = “01” – чтение нечетных строк, field_access = “10” – чтение четных строк изображения). Это необходимо для организации чересстрочного доступа (доступа к полукадрам) при прогрессивном хранении изображения в системной видеопамяти. Данный режим требует постоянства значения mvl_full_line_size, так как для вычисления адреса строки используется данное значение. Для пояснения вышесказанного см. рисунок 1.66 и рисунок 1.67.

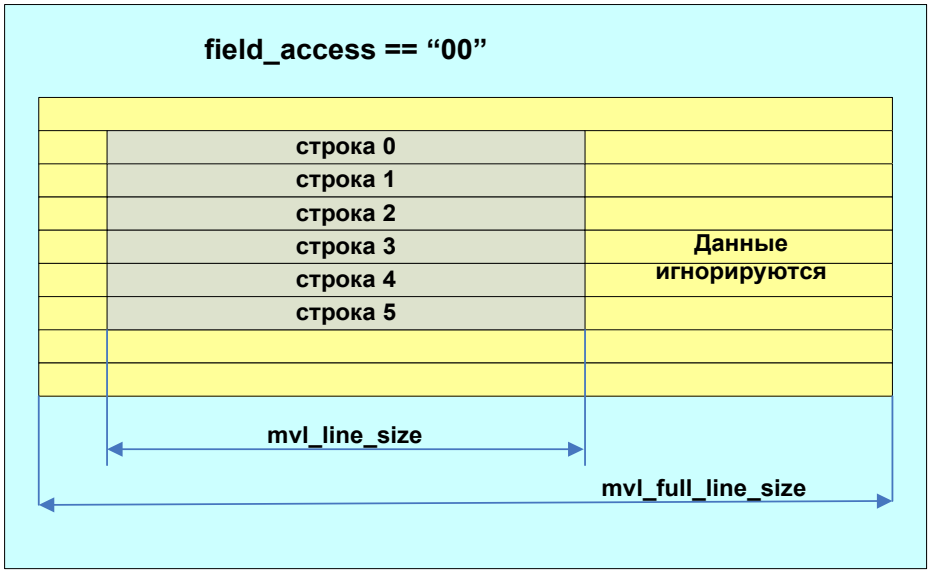


Рисунок 1.66 - Пример использования параметра field_access == “00”

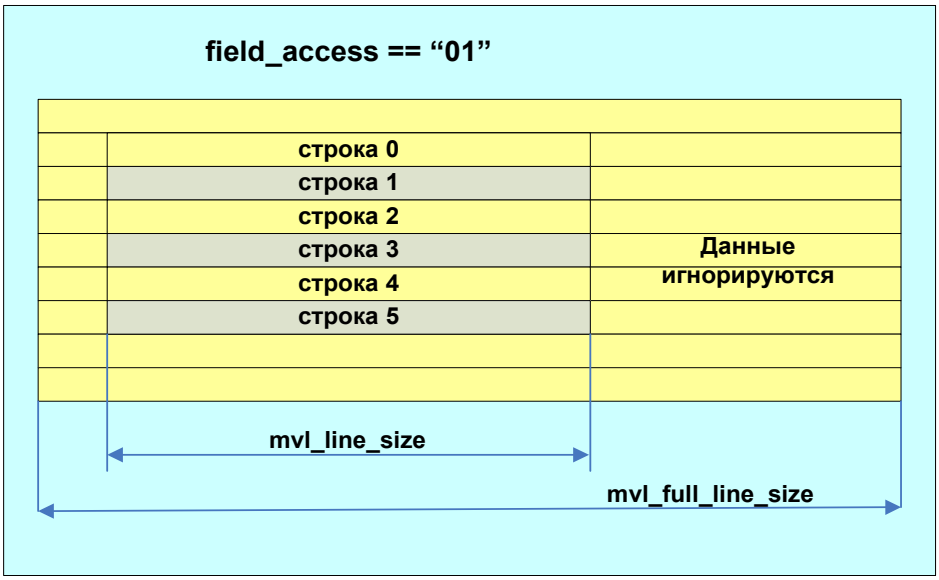


Рисунок 1.67 - Пример использования параметра field_access == “01”

1.3.5.1.1.5 Описание входных форматов графического слоя

Графический слой представляет собой набор графических областей. Каждая графическая область представляет собой прямоугольник, имеющий границы и содержащий набор пикселей, определяющих его содержание. Графические данные располагаются поверх области активного видео с использованием коэффициента прозрачности, определяющего прозрачность графических данных.

Графические области одного слоя не должны перекрываться по горизонтали и только одна область может быть определена для строки. Графические области считываются из системной памяти, каждая графическая область содержит:

- заголовок графической области;
- графические данные.

При хранении в памяти все графические данные хранятся в little endian формате.

1.3.5.1.1.5.1 Заголовок графической области

Заголовок определяет положение области на экране, размер области, формат цвета, указатель на заголовок следующей графической области в памяти (связанный список), указатель на начало графических данных, полный размер строки графических данных. Для форматов представления цвета 5:6:5 и 1:5:5:5 в заголовке задается значение коэффициента прозрачности для данной области.

	63	32	31 16	15 3	0
n	osd_base[31:1]	32	osd_next[31:3]		2...0
n+1	63...50 osd_ver_start[11:0]	47..44 osd_hor_start[11:0]	31..28 osd_ver_size[11:0]	15…12	osd_hor_size[11:0]
n+2			alpha[7:0]	23..18 df[1:0] 15…13	full_line_size[12:0]

Рисунок 1.68 - Структура заголовка графической области

osd_next[31:3] – указатель на заголовок следующей области, хранится в нулевом слове, разряды с 31 по 3, имеет выравнивание адреса на начало строки в памяти.

osd_hor_start[11:0] – горизонтальная координата начальной позиции графической области для отображения на экране, хранится в 1-ом слове, разряды с 43 по 32.

osd_ver_start[11:0] – вертикальная координата начальной позиции графической области для отображения на экране, хранится в 1-ом слове, разряды с 59 по 48.

df [1:0] – формат представления цвета для области, хранится во 2-ом слове, разряды с 17 по 16.

alpha [3:0] – коэффициент прозрачности для всей области, используется для форматов представления цвета 5:6:5 и 1:5:5:5, не используется для формата 4:4:4:4 и 8:8:8:8, хранится во 2-ом слове, разряды с 27 по 24.

osd_hor_size[11:0] – горизонтальный размер считываемой области в пикселах - 1 (размер строки), хранится в 1-ом слове, разряды с 11 по 0, максимальное значение 1920 пикселов.

osd_ver_size[11:0] – вертикальный размер считываемой области в строках -1, хранится в 1-ом слове, разряды с 27 по 16, максимальное значение 1080 строк.

full_line_size[12:0] – полный размер строки графического изображения в пикселах (full_line_size>= osd_hor_size) , хранится во 2-ом слове, разряды с 12 по 0, максимальное значение 1920 пикселов.

osd_base[31:1] – указатель на начало графических данных, хранится в нулевом слове, разряды с 63 по 33, имеет выравнивание на размер слова.

Все указатели хранятся по выровненным по 64 разрядам адресам в системной памяти.

Указатель на заголовок следующей области указывает на следующий заголовок. Значение указателя 0x0 означает конец связанного списка, что эквивалентно окончанию отображения графического слоя для данного кадра изображения.

Для формата представления цвета возможны следующие значения:

Таблица 1.46 - Список форматов графической информации

df[1:0]	Формат
00	8:8:8:8
01	5:6:5
10	1:5:5:5
11	4:4:4:4

Видеоконтроллером поддерживаются следующие форматы представления графических данных в памяти:

5:6:5 (RGB)
1:5:5:5 (ARGB)
5:5:5:1 (RGBA)
4:4:4:4 (ARGB)
4:4:4:4 (RGBA)
8:8:8:8 (ARGB)
8:8:8:8 (RGBA)

Положение бита, определяющего прозрачность – форматы ARGB и RGBA, задается управляющим регистром.

1.3.5.1.1.5.2 Форматы графических данных

Формат 5:6:5

Для компонент R, G, B отводится 5, 6 и 5 разрядов соответственно. Значение коэффициента прозрачности всегда берется из значения, задаваемого в заголовке для области.

Таблица 1.47 - Формат 5:6:5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R[4:0]					G[5:0]						B[4:0]				

Формат 1:5:5:5

На каждую компоненту R,G,B отводится по 5 разрядов. При этом для каждого пикселя задается индивидуальное значение коэффициента прозрачности. ‘0’ – означает полную прозрачность, ‘1’- означает, что для данного пикселя будет использоваться значение коэффициента прозрачности из заголовка области.

Таблица 1.48 - Формат 1:5:5:5 (ARGB)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	R[4:0]					G[4:0]					B[4:0]				

Таблица 1.49 - Формат 5:5:5:1 (RGBA)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R[4:0]					G[4:0]					B[4:0]					A

Формат 4:4:4:4

На компоненты R,G,B отводится по 4 разряда на цвет, при этом для каждого пикселя задается 4-х разрядное значение коэффициента прозрачности.

Таблица 1.50 - Формат 4:4:4:4 (ARGB)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A[3:0]				R[3:0]				G[3:0]				B[3:0]			

Таблица 1.51 - Формат 4:4:4:4 (RGBA)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R[3:0]				G[3:0]				B[3:0]				A[3:0]			

Формат 8:8:8:8

На компоненты R, G, B отводится по 8 разряда на цвет, при этом для каждого пикселя задается 8-ми разрядное значение коэффициента прозрачности.

Таблица 1.52 - Формат 8:8:8:8 (ARGB)

31 : 24	23 : 16	15 : 8	7 : 0
A[7:0]	R[7:0]	G[7:0]	B[7:0]

Таблица 1.53 - Формат 8:8:8:8 (RGBA)

31 : 24	23 : 16	15 : 8	7 : 0
R[7:0]	G[7:0]	B[7:0]	A[7:0]

1.3.5.1.1.5.3 Хранение в памяти графического изображения и графических областей

Обращение в память за данными графического слоя идет по адресу, выровненному для обращения к 64-х разрядной строке.

Данные графической области в памяти располагаются в линейном формате, как это изображено на рисунке.

P1	P2	P3	P4	P5	P6
P7	P8	P9	P10	P11	P12
P13	P14	P15	P16	P17	P18
P19	P20	P21	P22	P23	P24

Рисунок 1.69 - Графическое изображение и область в памяти.

1.3.5.1.1.6 Структура кадра выходного видеоизображения

Формируемое видеоконтроллером результирующее выходное видеоизображение (в дальнейшем активное видеоизображение) вставляется в генерируемую видеоконтроллером развертку, соответствующую одному из поддерживаемых видеоформатов. Развертка представляет собой структурирующую информацию, которая может передаваться как по внешним линиям, так и вставляться в поток видеоданных. Шина видеоданных и внешние сигналы структурирования информации составляют выходной видеоинтерфейс видеоконтроллера, по которому передается структурированное видеоизображение (в дальнейшем кадры видеоизображения).

Структура выходного кадра видеоизображения представлена на рисунках 1.70, 1.71.

Кадр видеоизображения состоит из межстрочных и межкадровых интервалов (Blank), заполняемых черным цветом, и активного видеоизображения.

Существует два основных типа развертки: прогрессивная – при которой все строки активного изображения передаются последовательно друг за другом, и чересстрочная – при которой вначале передаются все четные строки активного видеоизображения, а затем – нечетные, таким образом, кадр состоит из двух полукадров.

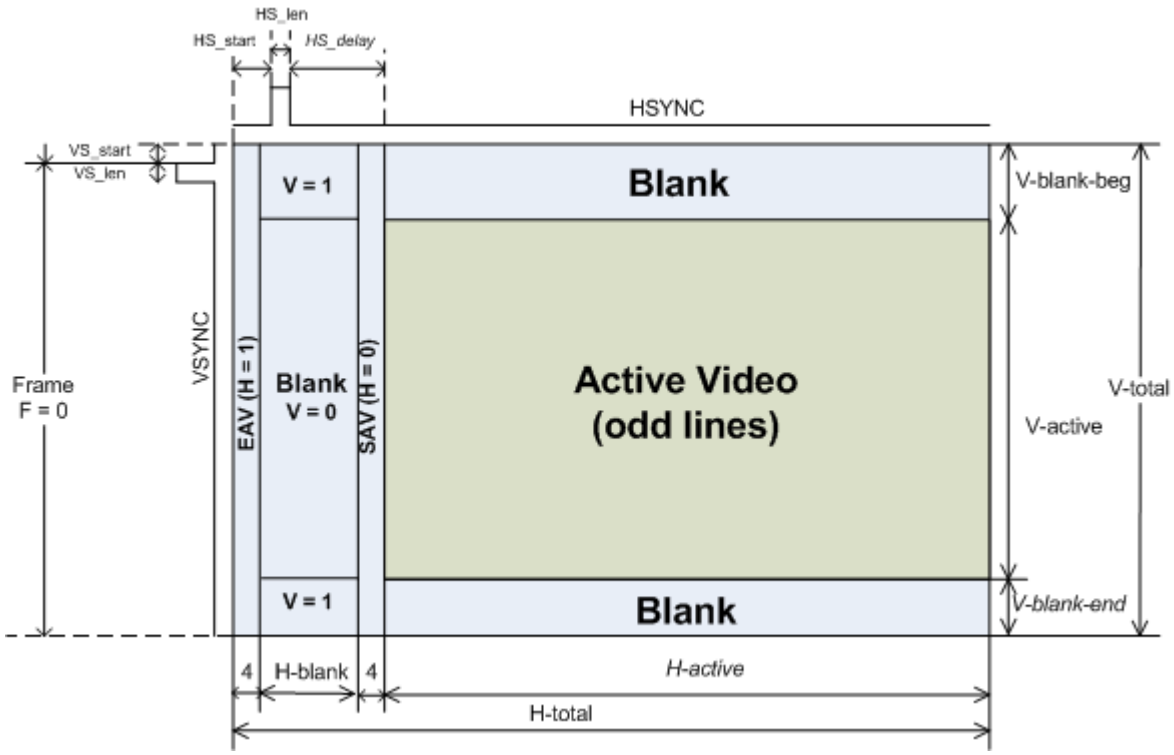


Рисунок 1.70 - Структура кадра при прогрессивной развертке

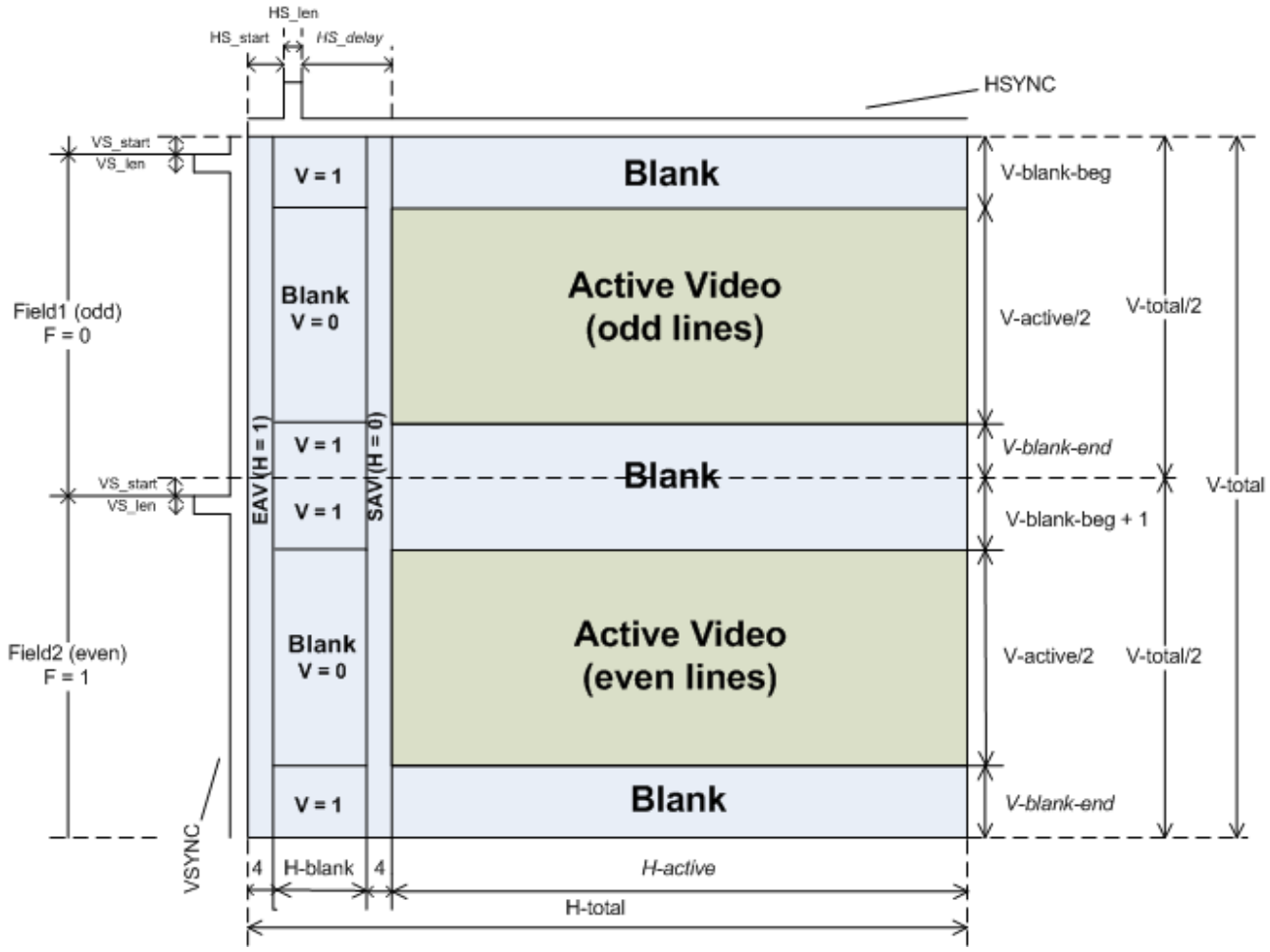


Рисунок 1.71 - Структура кадра при чересстрочной развертке

1.3.5.1.1.7 Выходной видеоинтерфейс

Выходной видеоинтерфейс состоит из 24-разрядной шины данных и линий, по которым передаются сигналы структурирования кадров.

Видеоконтроллер обеспечивает выдачу сигналов структурирования кадра (развертки) в соответствии со стандартом CEA-861-E. К сигналам развертки относятся:

- DataEnable – сигнал подсвечивающий данные активной части видеоизображения;
- HSYNC – сигнал горизонтальной синхронизации;
- VSYNC – сигнал вертикальной синхронизации;
- FIELD – идентификатор выводимого полукадра (‘0’ – четный, ‘1’ – нечетный).

Временные диаграммы формируемых сигналов структурирования кадра имеет общую структуру (см. рисунки 1.72, 1.3) и различаются лишь длительностью их отдельных сегментов, значения которой задаются программно через конфигурационные регистры видеоконтроллера. Полярность сигналов HSYNC, VSYNC также задается путем программирования конфигурационных регистров видеоконтроллера.

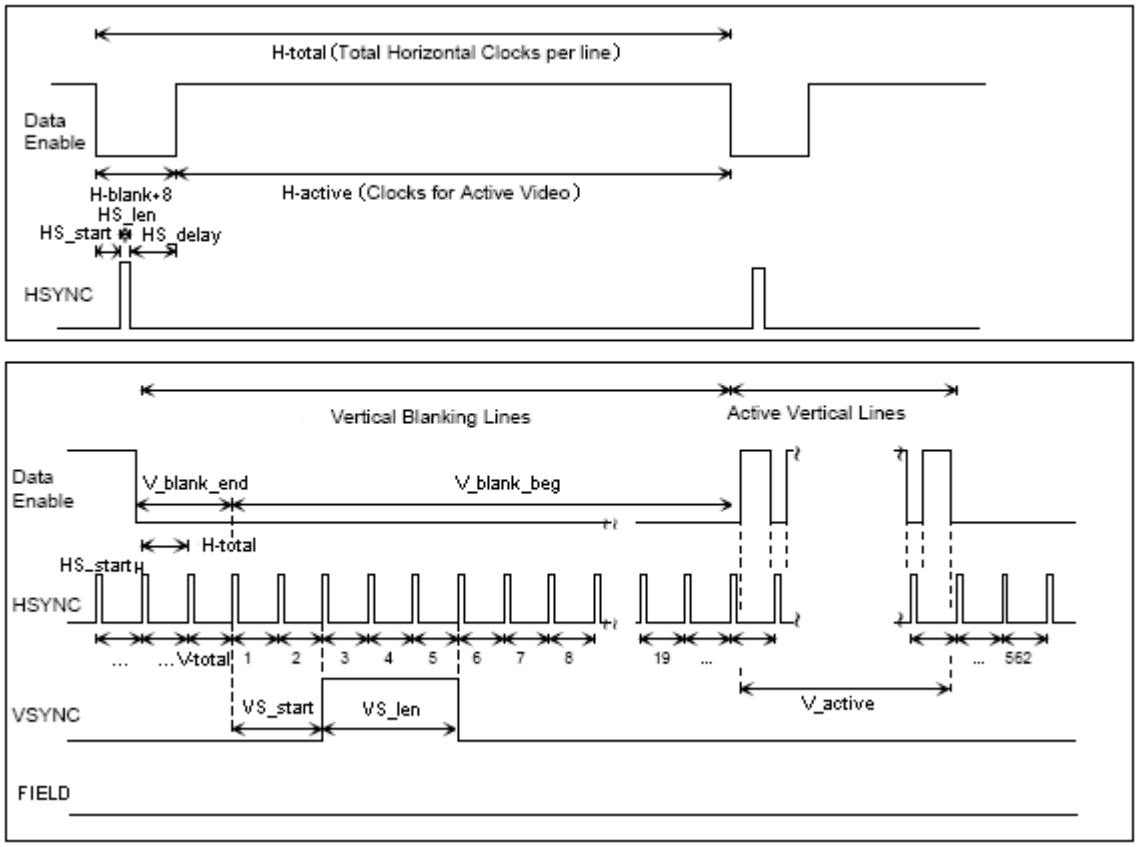


Рисунок 1.72 - Сигналы горизонтальной и вертикальной синхронизации при прогрессивной развертке

Во время активного уровня сигнала DataEnable на шину данных видеоинтерфейса выдаются пиксели активного видеоизображения, в противном случае (во время межстрочных и межкадровых интервалов) выдается значение, соответствующее черному цвету (Y = 0x10; Cb, Cr = 0x80).

Структурирующая информация помимо передачи по отдельным линиям может передаваться и в потоке данных посредством кодов начала и конца активных данных видеоизображения (SAV, EAV).

Коды SAV, EAV – это 4-х байтовые посылки формата: FF,00,00, XY.

XY = 1,F,V,H,P3,P2,P1,P0 – где:

F – бит четности полукадра;

V – бит вертикальной развертки;

H – бит горизонтальной развертки;

P3 = (V + H);

P2 = (F + H);

P1 = (F + V);

P0 = (F + V + H) ,

где – P0-P3 – биты кода четности, “+” – операция XOR (исключающее ИЛИ).

Биты F,V,H являются аналогами внешних сигналов FIELD, VSYNC, HSYNC. Значения, которые они принимают, зависят от расположения в кадре и показаны на рисунках 1.Рисунок 1.70, 1.Рисунок 1.71.

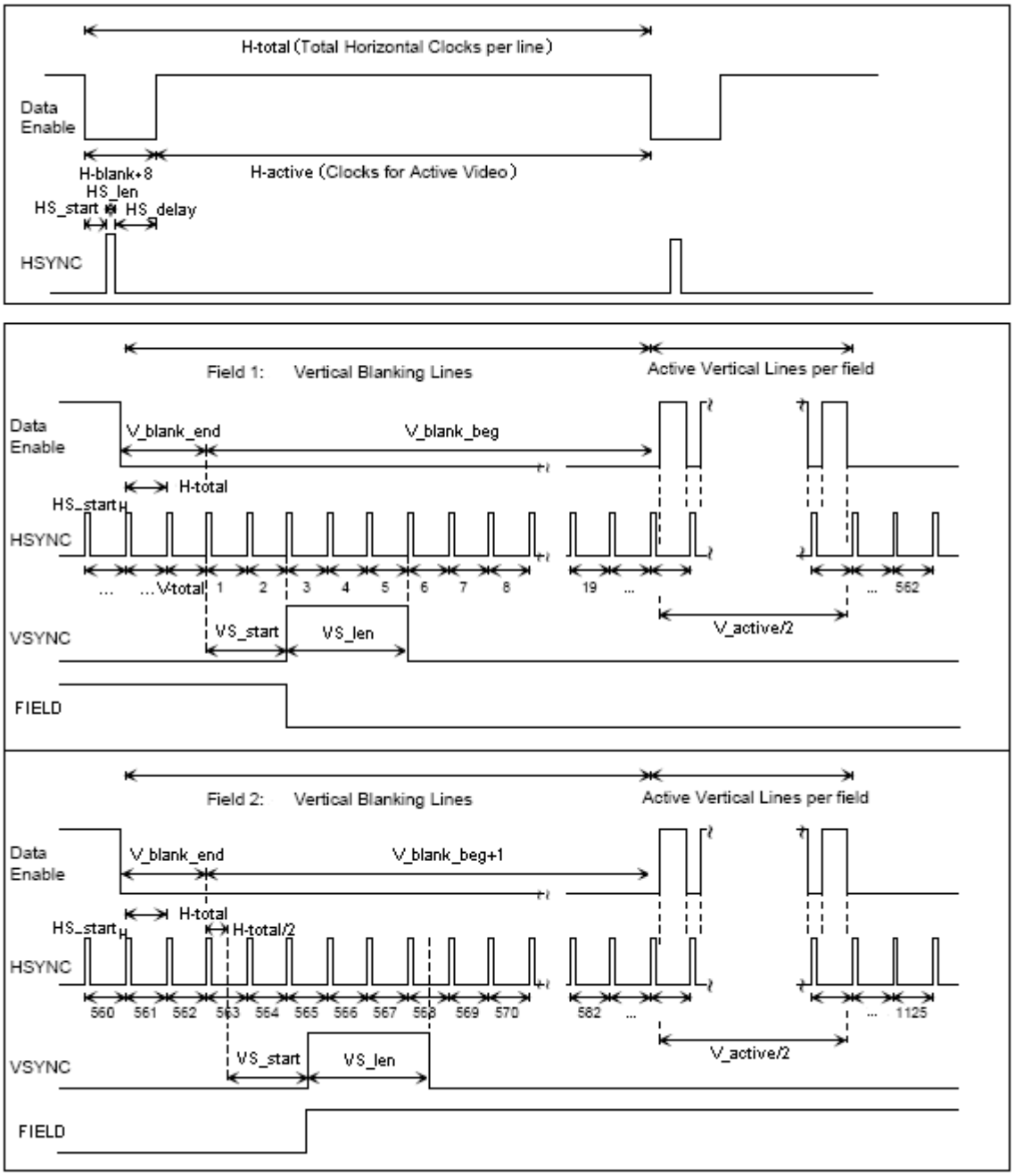


Рисунок 1.73 - Сигналы горизонтальной и вертикальной синхронизации при чересстрочной развертке

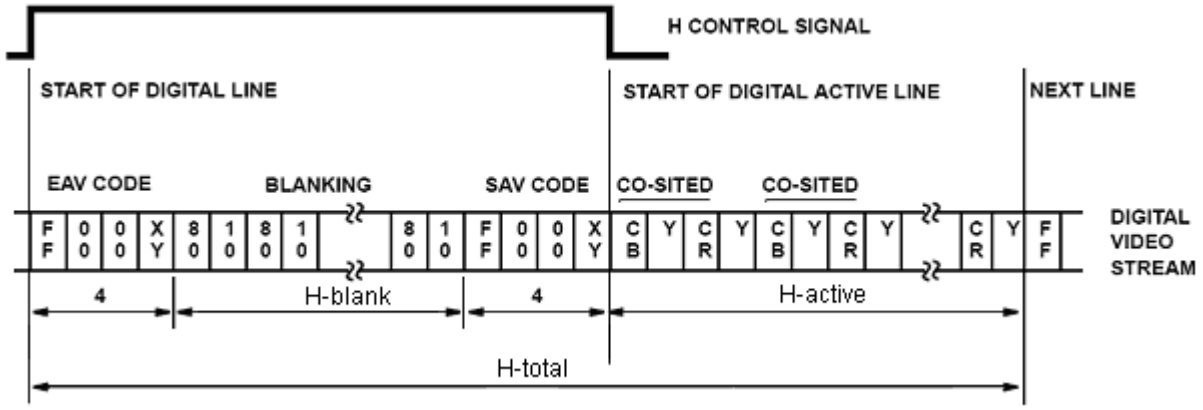


Рисунок 1.74 - Выходной поток видеоданных соответствующий одной линии видеоизображения формата SDTV

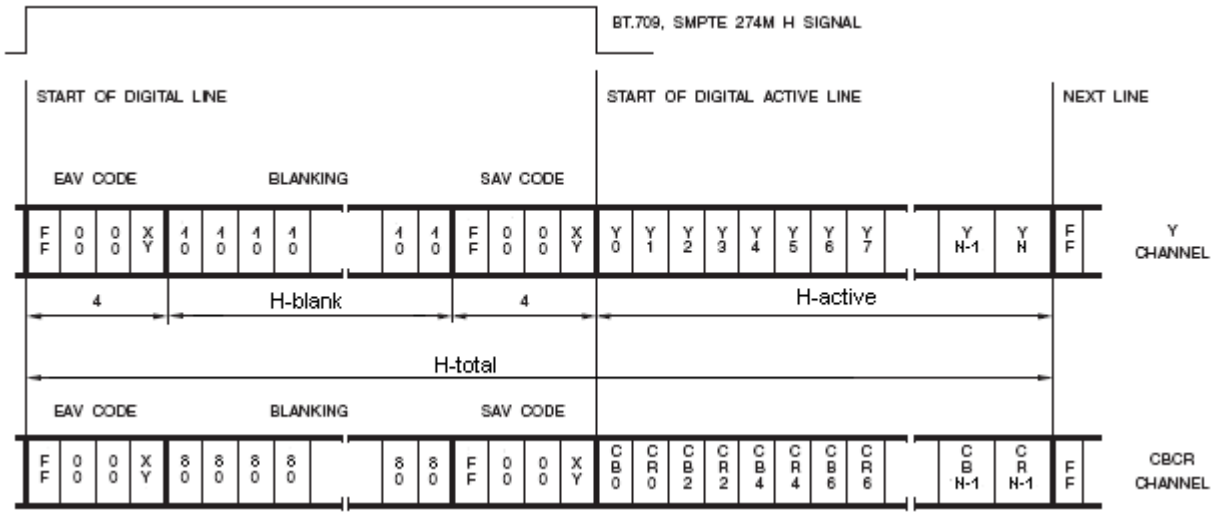


Рисунок 1.75 - Выходной поток видеоданных соответствующий одной линии видеоизображения форматов HDTV

1.3.5.1.1.8 Формат выдаваемых данных

Видеоконтроллер выдает данные форматов YCbCr 4:2:2, YCbCr 4:4:4 по 24-х разрядной шине данных видеоинтерфейса. Разрядность яркостных и цветностных компонент пикселя – 8бит.

Данные SDTV форматов могут выводиться двумя способами:

1) Последовательно по 8-ми разрядной шине (YCbCr 4:2:2). В этом случае яркостная (Y) и цветностные (Cb,Cr) составляющие пикселов передаются по одной шине в следующем порядке Cb0, Y0, Cr0, Y1, Cb2, Y2, Cr2, Y3 … и так далее. Значения передаются на частоте 27 МГц, при этом каждый пиксель (яркостная составляющая) передается с частотой вдвое меньшей – 13,5 МГц.

2) Параллельно по 24-х разрядной шине (YCbCr 4:4:4). В данном случае Y, Cb, Cr

Mode	Frame Hz	Sdandard	Pix Clk MHz	H-blank	H-active	H-Total	HS_start	HS_len	HS_delay	V-Blank beg	V-Active	V-Total	V-Blank end	VS_start	VS_len	VS_P
720x486i	30*2	BT656	27	134	720	858	19	62	57	19	486	525	0	3	3	0
720x576i	25*2	BT656	27	140	720	864	12	63	69	22	576	625	2	0	3	0
720(1440)x480i	30*2	770.2	27	268	1440	1716	38	124	114	21	480	525	1	3	3	0
720(1440)x576i	25*2	BT656	27	280	1440	1728	24	126	138	22	576	625	2	0	3	0
720x480p	60	770.2	27	130	720	858	16	62	60	42	480	525	3	6	6	0
720x576p	50	BT1358	27	136	720	864	12	64	68	44	576	625	5	0	5	0
1280x720p	60	770.3	74,25	362	1280	1650	110	40	220	25	720	750	5	0	5	1
1280x720p	50	296M	74,25	692	1280	1980	440	40	220	25	720	750	5	0	5	1
1920x1080i	30*2	770.3	74,25	272	1920	2200	88	44	148	20	1080	1125	2	0	5	1
1920x1080i	25*2	274M	74,25	712	1920	2640	528	44	148	20	1080	1125	2	0	5	1
1920x1080p	30	274M	74,25	272	1920	2200	88	44	148	41	1080	1125	4	0	5	1
1920x1080p	25	274M	74,25	712	1920	2640	528	44	148	41	1080	1125	4	0	5	1

составляющие передаются по отдельным 8-разрядным шинам. В данном режиме производится репликация пикселей (каждый пиксель передается дважды), таким образом, они передаются с частотой 13,5 МГц.

Данные EDTV, HDTV форматов могут выводиться двумя способами:

По 16-и разрядной шине (YCbCr 4:2:2). Яркостная составляющая (Y) и цветоразностная пара (Cb, Cr) передаются по собственным 8-разрядным шинам.

По 24-х разрядной шине (YCbCr 4:4:4). В данном случае Y, Cb, Cr составляющие передаются по отдельным 8-разрядным шинам. Репликация пикселов не производится.

Таблица 1.54 - Распределение составляющих пикселей в выходной шине данных для различных форматов видеоизображения

VideoData [23:0]	SDTV		HDTV/EDTV	
	4:2:2	4:4:4	4:2:2	4:4:4
dv0[7:0]	Cb0,Y0,Cr0,Y1…	Y0, Y0, Y1, Y1…	Y0, Y1…	Y0, Y1…
dv1[7:0]	-	Cr0,Cr0, Cr1,Cr1…	Cb0,Cr0…	Cr0,Cr1…
dv2[7:0]	-	Cb0,Cb0,Cb1,Cr1…	-	Cb0,Cb1…

Также возможен вывод изображения в формате RGB888, но только при работе исключительно с графическими слоями и фоном. В этом случае необходимо запретить преобразование RGB->YCbCr (установив бит OSD_RGB_YCBCR_BYPASS в единицу) для графических слоев, а значение фона необходимо задавать в цветовом формате RGB. Вывод производится по 24-х разрядной шине: компонента G соответствует Y, R – Cr, B – Cb.

Таблица 1.55 - Параметры и их значения необходимые для построения временной диаграммы выходного видеоизображения заданного формата

1920x1080p	60	274M	148,5	272	1920	2200	88	44	148	41	1080	1125	4	0	5	1
				X1	X2	X3	H1	H2	H3	V1	V2	V3	V4	V5	V6	

VS-P – полярность сигналов синхронизации

Максимальные значения параметров:
H-blank = 1023 H-active = 2047 H-Total = 4095
HS_start = 1023 HS_len = 255 HS_delay = 1023
V-Blank_beg = 63 V-Active = 2047 V-Total = 4095 V-Blank_end = 7
VS_start = 7 VS_len = 7

1.3.5.1.1.9 Программный (регистровый) интерфейс

Задание конфигурационных параметров и отображение текущего состояния видеоконтроллера ведется через регистры. Доступ к регистрам осуществляется посредством системного интерфейса управления (интерфейс APB slave). Каждое обращение к регистрам имеет выравнивание до 32-х разрядов.

При обращении к регистрам видеоконтроллера по зарезервированным адресам никаких ошибок не возникает, все пересылки завершаются нормально. При этом операции записи игнорируются, операции чтения возвращают все нули. При попытке записи в регистры, предназначенные только для чтения через системный интерфейс управления, операция записи завершается успешно, никакой записи не производится, состояние регистров не изменяется.

Внутренние регистры видеоконтроллера имеют связь со всеми блоками видеоконтроллера.

Управление видеоконтроллером и индикация информации о его состоянии ведется посредством регистров. Регистры видеоконтроллера объединены в группы по функциональному признаку. Видеоконтроллер содержит группы регистров, список которых приводится в таблицах 1.36 и 1.37. Доступ к регистрам ведется посредством интерфейса ведомого шины APB (системный интерфейс управления). Адреса регистров имеют выравнивание по 32 разряда.

Контроллер системного интерфейса управления содержит центральный дешифратор, который выделяет соответствующую группу регистров посредством анализа битов с 11 по 8 адреса.

Базовые адреса групп регистров, также приведены в таблицах 1.36 и 1.37. Описания регистров и принципов работы с полями регистров даны ниже.

При обращении к регистрам видеоконтроллера по зарезервированным или неиспользуемым адресам никаких ошибок не возникает, все пересылки завершаются нормально. При этом операции записи игнорируются, операции чтения возвращают все нули.

Видеоконтроллер состоит из двух ядер, каждое из которых имеет свой собственный набор регистров, доступный по собственной шине управления.

Набор групп регистров для каждого ядра может отличаться, это обусловлено различием в выполняемых ядрами функций. Ядро 0 (основное ядро) выполняет вывод видео и графической информации, содержит устройство масштабирования и обеспечивает генерацию сигналов кадровой развертки.

Ядро 1 (графическое ядро) в данной реализации видеоконтроллера обеспечивает вывод только графической информации.

Одноименные регистры внутри ядер имеют идентичную структуру и назначение, если какой-либо параметр из регистра в данном ядре не используется, он является зарезервированным и его изменение не влияет на аппаратуру. Так в качестве примера можно рассмотреть управляющий регистр ядра 1, в нем признаки, относящиеся к видеоданным являются незначащими и не используются.

Таблица 1.56 - Список групп регистров ядра 0 (основного) видеоконтроллера

Имя группы	Название	Базовые адреса групп
VDU Control	управляющие регистры видеоконтроллера	0x0000_0000
reserved	зарезервированные адреса	0x0000_0100
VDU Status	регистры состояния видеоконтроллера	0x0000_0200
VDU Interrupt	регистры прерываний видеоконтроллера	0x0000_0300
VDU Scaler	регистры масштабирования	0x0000_0400
VDU DIF	регистры цифровых видеоинтерфейсов	0x0000_0500
reserved	зарезервированные адреса	0x0000_0600
VDU OSD	регистры графического слоя	0x0000_0700
reserved	зарезервированные адреса	0x0000_0800
VDU MI	регистры интерфейса с памятью	0x0000_0900

Таблица 1.57 - Список групп регистров ядра 1 (дополнительного) видеоконтроллера

Имя группы	Название	Базовые адреса групп
VDU Control	управляющие регистры видеоконтроллера	0x0000_0000
reserved	зарезервированные адреса	0x0000_0100
VDU Status	регистры состояния видеоконтроллера	0x0000_0200
VDU Interrupt	регистры прерываний видеоконтроллера	0x0000_0300
reserved	зарезервированные адреса	0x0000_0400
reserved	зарезервированные адреса	0x0000_0500
reserved	зарезервированные адреса	0x0000_0600
VDU OSD	регистры графического слоя	0x0000_0700
reserved	зарезервированные адреса	0x0000_0800
VDU MI	регистры интерфейса с памятью	0x0000_0900

При обращении к регистрам, имеющим размер меньше ширины шины системного интерфейса управления (32 разряда), данные прижимаются (выравниваются) по правому краю (младшим разрядам).

1.3.5.1.2 Регистры параметров текущего и следующего кадров

Видеоконтроллер выводит на экран и считывает из памяти кадры видеоизображения. Каждый кадр может характеризоваться своими параметрами, которые меняются от кадра к кадру, среди таких параметров можно выделить базовый адрес в памяти для кадра основного видеослоя (для каждой компоненты) и базовый адрес заголовка первой области кадра для графического слоя. Для данных целей в видеоконтроллере реализованы два набора конфигурационных регистров.

Следующий – используется для задания параметров следующего кадра, не виден аппаратуре видеоконтроллера.

Текущий (активный) – в текущий момент времени используется аппаратурой видеоконтроллера.

Разделение на “текущие” и “следующие” регистры является логическим и физически реализуется на двух регистрах регистр_XXX_A(reg0), регистр_XXX_B(reg1), которые могут иметь различный статус: “текущий” или “следующий”, который может циклически переключаться.

Таким образом, изначально регистр_XXX_A является текущим, а регистр_XXX_B – следующим, после того как данные текущего кадра будут все считаны из внешней памяти в интерфейсный буфер видеоконтроллера произойдет переключение регистров и регистр_XXX_B обретет статус – текущий, а регистр_XXX_A – следующий и программист должен будет задать значение базового адреса следующего кадра. Задание следующего адреса должно быть проведено до того, как произойдет следующее переключение регистров.

Переключение регистров возможно только в том случае, если для кадра с заданным базовым адресом подготовлены все видеоданные в памяти. О том, что данные в видеопамяти для данного кадра готовы сигнализирует соответствующий сигнал регистр_XXX_SW_ENA установленный программно в единицу, этот сигнал и разрешает переключение страниц, в том случае если данный сигнал не будет установлен в единицу – переключения регистров не будет, видеоконтроллер будет ожидать разрешения переключения, чтение данных производиться не будет.

О том, какой из регистров - регистр_XXX_A или регистр_XXX_B в данный момент времени имеет статус текущий (активный) свидетельствует признак в регистре STAT_VDU_AVMP. Переключение регистров “текущего” и “следующего” адресов осуществляется по прерываниям MVL_FR_RD_END_ENA и OSD_FR_RD_END_ENA.

При описании регистров видеоконтроллера, для регистров, работающих по данному алгоритму будут даны дополнительные пояснения.

1.3.5.1.3 Особенность регистров состояния видеоконтроллера

При реализации регистров состояния не требуется их реализации в качестве запоминающего элемента в блоке регистров. Схемы выработки сигналов для этой группы регистров находятся в устройствах видеоконтроллера, поэтому в блоке регистров должно быть реализовано только адресное обращение к этим сигналам с возможностью их считывания. Запись в эти регистры не возвращает ошибки, однако сама запись не производится.

1.3.5.1.4 Особенность регистров прерываний

Группа регистров прерывания содержит два регистра – регистр разрешения прерываний и регистр состояний прерываний. При возникновении прерывания в аппаратуре видеоконтроллера оно фиксируется в регистре состояния прерываний вне зависимости от того установлен ли соответствующий бит в регистре разрешения прерываний. Если в разряде регистра разрешения прерываний установлена единица и в регистре состояния зафиксировано возникновение прерывания, то видеоконтроллером генерируется сигнал о возникновении прерывания, который выдается в систему.

Для снятия прерывания в соответствующий разряд регистра статуса прерываний программно должна быть установлена единица. При записи нулевого значения в соответствующий разряд регистра прерываний его значение не изменяется.

1.3.5.1.5 Невозможность изменения регистров при активном видеоконтроллере

Чтобы избежать повреждения изображения переключение между режимами видеоизображения необходимо проводить при обратном ходе растровой развертки, то есть изменения станут активными на следующем кадре, или при деактивированном видеоконтроллере, тогда изменения вступят в силу после его активации. Все связанные с видеорежимом регистры должны быть установлены в соответствующие значения до активирования видеоконтроллера.

1.3.5.1.6 Сброс регистров видеоконтроллера

Сброс регистров видеоконтроллера (запись в них начальных значений) осуществляется исключительно программным способом. То есть программист должен перед использованием регистров произвести программным способом инициализацию всех конфигурационных регистров видеоконтроллера. Исключение составляют регистры прерываний (регистр разрешения прерываний INT_VDU_ENA и регистр состояния прерываний INT_VDU_STAT, а также регистр управления CTRL_SOFT_RESET, а также управляющие регистры в каждой группе регистров, а именно MI_CTRL, DIF_CTRL, SCAL_CTRL), которые переводятся в исходное

состояние при приходе системного сигнала сброса, или сигнала программного сброса видеоконтроллера.

1.3.5.1.6.1.1 VDU Control - управляющие регистры видеоконтроллера

Таблица 1.58 - Управляющие регистры видеоконтроллера (VDU Control Registers)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
CTRL_VDU_ID	Идентификационный номер видеоконтроллера	VDU_CTRL_BASE + 0004H	r	32	ebeb'ab01H
CTRL_VDU_ENA	Разрешение отображения слоев видеоконтроллера	VDU_CTRL_BASE + 0008H	rw	3	0000'0000H
CTRL_SOFT_RESET	Программный сброс видеоконтроллера	VDU_CTRL_BASE + 000CH	rw	1	0000'0000H

Регистр идентификационного номера видеоконтроллера (CTRL_VDU_ID)

Регистр содержит информацию, идентифицирующую видеоконтроллер в системе, версию видеоконтроллера. Смещение VDU_CTRL_BASE + 0004H. Разрядность - 32 разряда.

Регистр разрешения видеоконтроллера (CTRL_VDU_ENA)

Регистр содержит управляющую информацию об активности слоев изображения, разрешая/запрещая их отображение. Смещение VDU_CTRL_BASE + 0008H. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1-39.

Таблица 1.59 - Разряды регистра разрешения отображения слоев видеоконтроллера (CTRL_VDU_ENA)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15-10	Rserved			
9	MVL_BASE_SW_ENA	разрешение переключения страниц видеопамяти для контроллера основного видеослоя 0 – переключение запрещено 1 – переключение разрешено. Когда переключение страниц разрешено и остаток буфера кадра считан из интерфейсного буфера (MVL_BUF), видеоконтроллер переключается между двумя доступными страницами видеопамяти, адреса которых расположены в регистрах базового адреса (покомпонентно) (MI_MVL_XX_BA) текущий/следующий. Контроллер автоматически изменяет флаг, отображающий текущую активную страницу видеопамяти (STAT_MVL_A_REG). При запрете чтение данных из памяти производится не будет.	rw	0H
8	OSD_BASE_SW_ENA	разрешение переключения страниц видеопамяти (базовых адресов первого заголовка) для контроллера графического слоя 0 – переключение запрещено 1 – переключение разрешено. Когда переключение страниц разрешено, и остаток буфера кадра считан из интерфейсного буфера (OSD_BUF), видеоконтроллер переключается между двумя доступными страницами видеопамяти, адреса которых расположены в регистрах “адреса первого заголовка графической области” (OSD_BASE) текущий/следующий. Контроллер автоматически изменяет флаг, отображающий текущую активную страницу видеопамяти (STAT_OSD_A_REG). При запрете чтение данных из памяти производится не будет.	rw	0H
7-3	Rserved			
2	MVL_ENA	отображение основного слоя 0 – основной слой деактивирован (чтение данных для данного слоя из видеопамяти не производится, устройство наложения слоев не запрашивает данные для данного слоя) 1 – основной слой активирован деактивирован (ведется чтение данных для данного слоя из видеопамяти и отображение данного слоя, при наличии VDU_ENA).	rw	0H
1	OSD_ENA	отображение графического слоя 0 – графический слой деактивирован (чтение данных для данного слоя из видеопамяти не производится, устройство наложения слоев не запрашивает данные для данного слоя) 1 – графический слой активирован (ведется чтение данных для данного слоя из видеопамяти и отображение данного слоя, при наличии VDU_ENA).	rw	0H
0	VDU_ENA	активность видеоконтроллера 0 - видеоконтроллер деактивирован (чтение видеоданных из внешней памяти не производится, на выходе видеоконтроллера установлены константные значения) 1 – видеоконтроллер активирован (ведется чтение данных из внешней памяти для активированных слоев, на выходе видеоконтроллера генерируются временные диаграммы соответствующей развертки)	rw	0H

Регистр программного сброса видеоконтроллера (CTRL_SOFT_RESET)

Программная установка единицы в данном регистре вызывает сброс управляющей логики видеоконтроллера. Проведение сброса возможно только при деактивированном видеоконтроллере (анализируется признак INT_VDU_OFF).

При активном видеоконтроллере единица в данном регистре будет сброшена на следующем такте аппаратурой видеоконтроллера и сброса логики видеоконтроллера произведено не будет.

При поступлении данного сигнала в неактивной стадии видеоконтроллера будет произведен сброс логики видеоконтроллера. Сброс будет длиться несколько десятков тактов системной частоты. Во время сброса модификация регистров видеоконтроллера запрещена. В течение этого времени чтение регистра программного сброса будет возвращать единицу. Регистр программного сброса сбрасывается по завершении программного сброса.

Таким образом, дальнейшая работа с видеоконтроллером возможно только при чтении нулевого значения из данного регистра.

Смещение VDU_CTRL_BASE + 000CH. Разрядность - 1 разряд.

1.3.5.1.6.1.2 VDU Status - регистры состояния видеоконтроллера

Таблица 1.60 - Регистры состояния видеоконтроллера (VDU Status Registers)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
STAT_VDU_DISP	Состояние отображения изображения	VDU_STAT_BASE + 0004H	r	3	7H
STAT_VDU_AVMP	Активный набор регистров для базовых адресов	VDU_STAT_BASE + 0008H	r	2	0H
STAT_VDU_FIFO	Состояние встроенной буферной памяти видеоконтроллера	VDU_STAT_BASE + 000CH	r	8	101H

Регистр состояния отображения изображения (STAT_VDU_DISP)

Регистр содержит информацию, отображающую состояние выводимого изображения на выходе видеоконтроллера. Смещение VDU_STAT_BASE + 0004H. Разрядность - 3 разряда. Разбиение на поля приведено в таблице 1.41.

Таблица 1.61 - Разряды регистра состояния отображения изображения (STAT_VDU_DISP)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
4	FIELD	идентифицирует выводимый полукадр в кадре при чересстрочном режиме 0 – четный полукадр (строки 0, 2, 4, …) 1 – нечетный полукадр (строки 1, 3, 5, …) особенность заключена в том, что переключение данного признака осуществляется в момент выработки признака SA (обратный ход развертки), таким образом, если при возникновении SA считать признак FIELD, то он будет отражать значение для следующего отображаемого полукадра	r	0
2	HBB	обратный ход луча по горизонтали 0 – прямой ход 1 – обратный ход	r	1H
1	VBB	обратный ход луча по вертикали 0 – прямой ход 1 – обратный ход	r	1H
0	BLANK_ACT	вывод изображения 0 – во время вывода изображения на экран 1 – когда активен сигнал BLANK	r	1H

Регистр состояния активного набора регистров для базовых адресов (STAT_VDU_AVMP)

Регистр содержит информацию, об активном наборе регистров базовых адресов. Смещение VDU_STAT_BASE + 0008H. Разрядность - 2 разряда. Разбиение на поля приведено в таблице 1.42.

Таблица 1.62 - Разряды регистра состояния активного набора регистров для базовых адресов (STAT_VDU_AVMP)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
1	MVL_AVMP	активный набор регистров для основного видеослоя (определяет текущую видеостраницу - кадр) 0 – активен набор управляющих регистров видеоконтроллера REG0 (A) 1 – активен набор управляющих регистров видеоконтроллера REG1 (B)	r	0H
0	OSD_AVMP	активный набор регистров для графического слоя (определяет текущую видеостраницу – кадр) 0 – активен набор управляющих регистров видеоконтроллера REG0 (A) 1 – активен набор управляющих регистров видеоконтроллера REG1 (B)	r	0H

Регистр состояния встроенной буферной памяти видеоконтроллера (STAT_VDU_FIFO)

Регистр содержит информацию, отображающую состояние внутренней буферной памяти видеоконтроллера. Смещение VDU_STAT_BASE + 000CH. Разрядность - 8 разрядов. Разбиение на поля приведено в таблице 1.43.

Таблица 1.63 - Разряды регистра состояния встроенной буферной памяти видеоконтроллера (STAT_VDU_FIFO)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
25-24	MVL_I_FIFO	интегральный уровень заполнения интерфейсного буфера основного слоя 00 - уровень буфера от 0 до ¼ от количества слов в буфере 01 - уровень буфера от ¼ до ½ от количества слов в буфере 10 - уровень буфера от ½ до ¾ от количества слов в буфере 11 - уровень буфера более ¾ от количества слов в буфере	r	00H
17-16	OSD_I_FIFO	интегральный уровень заполнения интерфейсного буфера графического слоя 00 - уровень буфера от 0 до ¼ от количества слов в буфере 01 - уровень буфера от ¼ до ½ от количества слов в буфере 10 - уровень буфера от ½ до ¾ от количества слов в буфере 11 - уровень буфера более ¾ от количества слов в буфере	r	00H
9-8	MVL_O_FIFO	статус заполненности линейного (выходного) буфера основного слоя 01 – буфер пуст 10 – буфер заполнен 00,11 – недопустимые значения	r	01H
1-0	OSD_O_FIFO	статус заполненности линейного (выходного) буфера графического слоя 01 – буфер пуст 10 – буфер заполнен 00,11 – недопустимые значения.	r	01H

Регистр счетчика импульсов сигнала пиксельной синхронизации (STAT_SA_PIX_CNT)

Регистр содержит информацию, о количестве импульсов сигнала пиксельной синхронизации, возникших с момента выработки очередного сигнала SA. Смещение VDU_STAT_BASE + 0014H. Разрядность - 22 разряда. Разбиение на поля приведено в таблице 1.44.

Таблица 1.64 - Разряды регистра состояния встроенной буферной памяти видеоконтроллера (STAT_SA_PIX_CNT)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
21-0	SA_PIX_CNT	22х разрядное состояние счетчика, указывает количество тактов пиксельной синхронизации, прошедших с момента выработки сигнала SA	r	00H

1.3.5.1.6.1.3 VDU Interrupt - регистры прерываний видеоконтроллера

Таблица 1.65 - Регистры прерываний видеоконтроллера (VDU Interrupt Registers)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
INT_VDU_ENA	Регистр разрешения прерываний	VDU_INT_BASE + 0004H	rw	32	00000000H
INT_VDU_STAT	Регистр состояния прерываний	VDU_INT_BASE + 0008H	rw	32	00000302H

Эти регистры имеют одинаковую структуру. Группа регистров прерывания содержит два регистра – регистр разрешения прерываний и регистр состояний прерываний. При возникновении прерывания в аппаратуре видеоконтроллера оно фиксируется в регистре состояния прерываний вне зависимости от того установлен ли соответствующий бит в регистре разрешения прерываний. Если в разряде регистра разрешения прерываний установлена единица и в регистре состояния зафиксировано возникновение соответствующего прерывания, то видеоконтроллером генерируется сигнал о возникновении прерывания, который выдается в систему.

Для снятия прерывания в соответствующий разряд регистра статуса прерываний (VDU_INT_STAT) программно должна быть установлена единица. При записи нулевого значения в соответствующий разряд регистра прерываний его значение не изменяется.

Регистр разрешения прерываний (INT_VDU_ENA)

Регистр содержит информацию, разрешающую выставление сигнала о прерывании в регистре INT_VDU_STAT. Смещение VDU_INT_BASE + 0004H. Разрядность - 10 разрядов. Разбиение на поля приведено в таблице 1.46.

Таблица 1.66 - Разряды регистра разрешения прерываний (INT_VDU_ENA)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31..26	RESERVED			
25	I_FIFO_EMPTY_ENA	разрешает возникновение прерывания - опустошение входного буфера 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H
24	O_FIFO_EMPTY_ENA	разрешает возникновение прерывания - опустошение выходного буфера 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H
19..23	RESERVED			
18	SA_ENA	разрешает возникновение прерывания - обратный ход кадровой развертки 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H

Продолжение таблицы 1.67

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
17	H_SYNC_ENA	разрешает возникновение прерывания - импульс горизонтальной синхронизации 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H
16	V_SYNC_ENA	разрешает возникновение прерывания - импульс вертикальной синхронизации 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H
11..15	RESERVED			
10	FR_PRC_END_ENA	разрешает возникновение прерывания - окончание обработки активного кадра 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H
9	MVL_FR_RD_END_ENA	разрешает возникновение прерывания - окончание считывания активного кадра для основного видеослоя из памяти 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H
8	OSD_FR_RD_END_ENA	разрешает возникновение прерывания - окончание считывания активного кадра для графического слоя из памяти 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H
2..7	RESERVED			
1	VDU_OFF_ENA	разрешает возникновение прерывания - отключение контроллера 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H
0	INT_SYS_ERR_ENA	разрешает возникновение прерывания - системная ошибка 0 – прерывание запрещено 1 – прерывание разрешено	rw	0H

Регистр состояния прерываний (INT_VDU_STAT)

Регистр содержит информацию, отображающую состояние прерываний видеоконтроллера. При возникновении прерывания аппаратура видеоконтроллера выставляет в соответствующем разряде регистра INT_VDU_STAT единичное значение, свидетельствующее о возникновении прерывания. В дальнейшем анализируется состояние регистра разрешения прерываний и при единичном значении в соответствующем разряде регистра INT_VDU_ENA на основании сборки сигналов данного регистра вырабатывается единый сигнал о запросе прерывания.

Для снятия сигнала прерывания в соответствующий разряд регистра INT_VDU_STAT должно быть записано единичное значение. Учитывая вышесказанное в регистр INT_VDU_STAT должна обеспечиваться возможность записи информации как с шины системного интерфейса управления, так и от внутренних блоков видеоконтроллера.

Смещение VDU_INT_BASE + 0008H. Разрядность - 10 разрядов. Разбиение на поля приведено в таблице 1.47. Назначение полей аналогично полям регистра INT_VDU_ENA.

Таблица 1.68 - Разряды регистра состояния прерываний (INT_VDU_STAT)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31..26	RESERVED			
25	I_FIFO_EMPTY	опустошение входного буфера Возникает при опустошении входного буфера для mvl или osd слоя в рабочем режиме видеоконтроллера. Означает, что из входного буфера было произведено считывание пиксела в тот момент, когда буфер был пуст. Может быть вызвано блокировкой интерфейсной шины, обращением по неправильному адресу видеопамяти или недостаточным размером буфера.	rw	0H
24	O_FIFO_EMPTY	опустошение выходного буфера Возникает при опустошении выходного буфера для mvl или osd слоя в рабочем режиме видеоконтроллера при выводе этих слоев. Означает, что из выходного буфера было произведено считывание пиксела в тот момент, когда буфер был пуст, т.е. отображение данных ведется быстрее их поступления. Может быть вызвано блокировкой интерфейсной шины, обращением по неправильному адресу видеопамяти или недостаточным размером буфера.	rw	0H
23..19	RESERVED			
18	SA	обратный ход кадровой развертки	rw	0H
17	H_SYNC	импульс горизонтальной синхронизации Означает появление импульса горизонтальной синхронизации (HSYNC).	rw	0H
16	V_SYNC	импульс вертикальной синхронизации Означает появление импульса вертикальной синхронизации (VSYNC)	rw	0H
15..11	RESERVED			
10	FR_PRC_END	окончание обработки активного кадра Данное прерывание возникает, когда все видеоданные из активной страницы видеопамяти (кадра) были обработаны и выданы при помощи блока наложения и позиционирования видеослоев (выданы данные как для основного, так и графического слоев кадра). Соответствует окончанию выдачи активной области кадра или полукадра при чересстрочном режиме отображения.	rw	0H
9	MVL_FR_RD_END	окончание считывания активного кадра для основного видеослоя из памяти Данное прерывание возникает, когда все видеоданные из активной страницы видеопамяти (кадра или полукадра при чересстрочном режиме отображения) были считаны из памяти контроллером DMA_MVL.	rw	1H
8	OSD_FR_RD_END	окончание считывания активного кадра для графического слоя из памяти Данное прерывание возникает, когда все видеоданные из активной страницы видеопамяти (кадра или полукадра при чересстрочном режиме отображения) были считаны из памяти контроллером DMA_OSD.	rw	1H
7..2	RESERVED			
1	VDU_OFF	отключение контроллера Возникает при установке признака CTRL_VDU_ENA в ноль после окончания отображения последнего кадра. После установки INT_VDU_OFF все регистры видеоконтроллера могут быть изменены. Рекомендуется менять значения регистров от которых зависит видеорежим при установке признака INT_VDU_OFF. После установки признака INT_VDU_OFF перед последующим использованием видеоконтроллера рекомендуется проводить программный сброс видеоконтроллера посредством программной установки признака CTRL_SOFT_RESET.	rw	1H
0	INT_SYS_ERR	Системная ошибка. Возникает при прихождении от интерфейсных блоков сигнала об ошибке в процессе чтения данных из видеопамяти.	rw	0H

1.3.5.1.6.1.4 VDU Scaler - регистры масштабирования

Таблица 1.69 - Регистры масштабирования (VDU Scaler Registers)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
SCALER_CTRL	регистр управления устройством масштабирования	VDU_SCALER_BASE + 0004H	rw	32	0000_0000H
SCALER _SCH_Y	коэффициент горизонтального масштабирования для яркостной компоненты	VDU_SCALER_BASE + 0008H	rw	20	0000_0000H
SCALER _SCV_Y	коэффициент вертикального масштабирования для яркостной компоненты	VDU_SCALER_BASE + 000CH	rw	20	0000_0000H
SCALER _SCH_C	коэффициент горизонтального масштабирования для цветоразностных компонент	VDU_SCALER_BASE + 0010H	rw	20	0000_0000H
SCALER _SCV_C	коэффициент вертикального масштабирования для цветоразностных компонент	VDU_SCALER_BASE + 0014H	rw	20	0000_0000H
SCALER _SIZE_Y	размер изображения после масштабирования для яркостной компоненты	VDU_SCALER_BASE + 0018H	rw	32	0000_0000H
SCALER _SIZE_C	размер изображения после масштабирования для цветоразностной компоненты	VDU_SCALER_BASE + 001CH	rw	32	0000_0000H
SCALER_FLT_Y_C0	нулевой коэффициент фильтра устройства масштабирования для яркостной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 0020H	rw	32	0000_0000H
SCALER_FLT_Y_C1	первый коэффициент фильтра устройства масштабирования для яркостной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 0024H	rw	32	0000_0000H
SCALER_FLT_Y_C2	второй коэффициент фильтра устройства масштабирования для яркостной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 0028H	rw	32	0000_0000H
SCALER_FLT_Y_C3	третий коэффициент фильтра устройства масштабирования для яркостной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 002CH	rw	32	0000_0000H
SCALER_FLT_Y_C4	четвертый коэффициент фильтра устройства масштабирования для яркостной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 0030H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 0034H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 0038H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 003CH	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 0040H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 0044H	rw	32	0000_0000H
SCALER_FLT_NORM	регистр коэффициентов нормализации горизонтального фильтра для яркостной компоненты	VDU_SCALER_BASE + 0048H	rw	32	0000_0000H

Продолжение таблицы 1.70

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
SCALER_FLT_C_C0	нулевой коэффициент фильтра устройства масштабирования для цветоразностной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 004CH	rw	32	0000_0000H
SCALER_FLT_C_C1	первый коэффициент фильтра устройства масштабирования для цветоразностной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 0050H	rw	32	0000_0000H
SCALER_FLT_C_C2	второй коэффициент фильтра устройства масштабирования для цветоразностной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 0054H	rw	32	0000_0000H
SCALER_FLT_C_C3	третий коэффициент фильтра устройства масштабирования для цветоразностной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 0058H	rw	32	0000_0000H
SCALER_FLT_C_C4	четвертый коэффициент фильтра устройства масштабирования для цветоразностной компоненты (по вертикали и горизонтали)	VDU_SCALER_BASE + 005CH	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 0060H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 0064H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 0068H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 006CH	rw	32	0000_0000H
SCALER_PHS_CUT_V	количество обрезаемых пикселов по вертикали	VDU_SCALER_BASE + 0070H	rw	32	0000_0000H
SCALER_SIZE_CUT	размер изображения после обрезания	VDU_SCALER_BASE + 0074H	rw	32	0000_0000H
SCALER_PHS_CUT_H	количество обрезаемых пикселов по горизонтали	VDU_SCALER_BASE + 0078H	rw	32	0000_0000H
SCALER_MVL_CLR_Y	матрица коэффициентов для преобразования яркостной составляющей	VDU_SCALER_BASE + 007CH	rw	32	0000_0000H
SCALER_MVL_CLR_Cb	матрица коэффициентов для преобразования цветоразностной составляющей Cb	VDU_SCALER_BASE + 0080H	rw	32	0000_0000H
SCALER_MVL_CLR_Cr	матрица коэффициентов для преобразования цветоразностной составляющей Cr	VDU_SCALER_BASE + 0084H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 0088H	rw	32	0000_0000H
reserved		VDU_SCALER_BASE + 008CH	rw	32	0000_0000H

Регистр управления устройством масштабирования (SCALER_CTRL)

Регистр содержит управляющую информацию для устройства масштабирования видеоконтроллера. Смещение VDU_SCALER_BASE + 0004H. Разрядность - 32 разряда.

Таблица 1.71 - Разряды регистра управления устройством масштабирования (SCALER_CTRL)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 26	RSERVED			
25	SCAL_DROP_POLAR_STR	определяет полярность схемы dropper'а (интерлейсера) 0 – отбрасываются нечетные строки кадра (выводимый кадр содержит только четные строки исходного кадра) 1 – отбрасываются четные строки кадра (выводимый кадр содержит только нечетные строки исходного кадра)	rw	0H
24	SCAL_DROP_ENA_STR	разрешает работу устройства dropper (интерлейсер), которое производит удаление четных или нечетных строк в кадре (зависит от признака SCAL_DROP_POLAR_STR), удаление строк осуществляется после масштабирования и обрезания кадра 0 – пропуск строк отключен, данные минуют dropper 1 – производится отбрасывание строк	rw	0H
23 - 18	RSERVED			
17	V_FLT_ENA	включает фильтрацию изображения при его масштабировании по вертикали 0 – фильтрация отключена, данные минуют фильтр 1 – производится фильтрация данных Вертикальная фильтрация изображения допустима только в том случае, если горизонтальный размер изображения после масштабирования (параметр SCALER _SIZE_Y(C)_H) меньше, либо равен 1024 пикселам – это обусловлено размерами буферов, используемых для вертикальной фильтрации.	rw	0H
16	H_FLT_ENA	включает фильтрацию изображения при его масштабировании по горизонтали 0 – фильтрация отключена, данные минуют фильтр 1 – производится фильтрация данных	rw	0H
15 - 12	RSERVED			
11	V_CUT_ENA	включает режим обрезания изображения в вертикальном направлении 0 – изображение сохраняет размер 1 – производится обрезание изображения после его масштабирования		
10	H_CUT_ENA	включает режим обрезания изображения в горизонтальном направлении 0 – изображение сохраняет размер 1 – производится обрезание изображения после его масштабирования		
9	V_UPSIZE	направление масштабирования в вертикальном направлении 0 – производится уменьшение изображения (downscaling) 1 – производится увеличение изображения (upscaling) в данной версии видеоконтроллера этот параметр является опциональным и может быть не установлен, о направлении масштабирования, аппаратура видеоконтроллера судит по значению целой части коэффициентов масштабирования	rw	0H

Продолжение таблицы 1.72

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
8	H_UPSIZE	направление масштабирования в горизонтальном направлении 0 – производится уменьшение изображения (downscaling) 1 – производится увеличение изображения (upscaling) в данной версии видеоконтроллера этот параметр является опциональным и может быть не установлен, о направлении масштабирования, аппаратура видеоконтроллера судит по значению целой части коэффициентов масштабирования	rw	0H
7 - 2	RSERVED			
1	V_SCALER_ENA	включает масштабирование изображения по вертикали 0 – масштабирование изображения не производится 1 – масштабирование изображения производится	rw	0H
0	H_SCALER_ENA	включает масштабирование изображения по горизонтали 0 – масштабирование изображения не производится 1 – масштабирование изображения производится	rw	0H

Регистр коэффициента горизонтального масштабирования для яркостной компоненты (SCALER _SCH_Y)

Регистр содержит коэффициенты для масштабирования плоскости яркостной компоненты изображения по горизонтали. Коэффициент задается в виде шага масштабирования, определяемого по формуле ((кол-во пикселов в строке до масштабирования – 1) / (кол-во пикселов в строке после масштабирования – 1)) Коэффициенты представлены в виде целой и дробной частей, обе в виде беззнакового целого. Дробная часть коэффициента получена умножением исходной вещественной дробной части на 65536 (берется целая часть – дробная отбрасывается).

При увеличении – целая часть коэффициента является нулевой. При уменьшении коэффициент масштабирования больше 1.

Согласно сценариям применения видеоконтроллера, увеличение изображения производится от значений изображения с разрешениями меньше SD (но не менее CIF) до изображения с SD разрешением. А уменьшение изображения от изображений с разрешением HD до изображения с SD.

Под целую часть отводится 4 разряда, под дробную – 16 разрядов.

Смещение VDU_SCALER_BASE + 0008H. Разрядность - 20 разрядов. Разбиение на поля приведено в таблице 1.50.

Таблица 1.73 - Разряды регистра коэффициента горизонтального масштабирования для яркостной компоненты (SCALER _SCH_Y)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
19 – 16	SC_INT_HY	значение целой части коэффициента	rw	0H
15 - 0	SC_FRAC_HY	значение дробной части коэффициента	rw	0000H

Регистр коэффициента вертикального масштабирования для яркостной компоненты (SCALER _SCV_Y)

Регистр содержит коэффициенты для масштабирования плоскости яркостной компоненты изображения по вертикали. Коэффициент задается в виде шага масштабирования, определяемого по формуле ((кол-во строк до масштабирования – 1) / (кол-во строк после масштабирования – 1)). Коэффициенты представлены в виде целой и дробной частей, обе в виде беззнакового целого. Дробная часть коэффициента получена умножением исходной вещественной дробной части на 65536 (берется целая часть – дробная отбрасывается).

При увеличении – целая часть коэффициента является нулевой. При уменьшении коэффициент масштабирования больше 1.

Согласно сценариям применения видеоконтроллера, увеличение изображения производится от значений изображения с разрешениями меньше SD (но не менее CIF) до изображения с SD разрешением. А уменьшение изображения от изображений с разрешением HD до изображения с SD.

Под целую часть отводится 4 разряда, под дробную – 16 разрядов.

Смещение VDU_SCALER_BASE + 000CH. Разрядность - 20 разрядов. Разбиение на поля приведено в таблице 1.51.

Таблица 1.74 - Разряды регистра коэффициента вертикального масштабирования для яркостной компоненты (SCALER _SCV_Y)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
19 – 16	SC_INT_VY	значение целой части коэффициента	rw	0H
15 - 0	SC_FRAC_VY	значение дробной части коэффициента	rw	0000H

Регистр коэффициента горизонтального масштабирования для цветоразностных компонент (SCALER _SCH_C)

Регистр содержит коэффициенты для масштабирования плоскости цветоразностной компоненты изображения по горизонтали. Коэффициент задается в виде шага масштабирования, определяемого по формуле ((кол-во пикселов в строке до масштабирования – 1) / (кол-во пикселов в строке после масштабирования – 1)) Коэффициенты представлены в виде целой и дробной частей, обе в виде беззнакового целого. Дробная часть коэффициента получена умножением исходной вещественной дробной части на 65536 (берется целая часть – дробная отбрасывается).

При увеличении – целая часть коэффициента является нулевой. При уменьшении коэффициент масштабирования больше 1.

Согласно сценариям применения видеоконтроллера, увеличение изображения производится от значений изображения с разрешениями меньше SD (но не менее CIF) до изображения с SD разрешением. А уменьшение изображения от изображений с разрешением HD до изображения с SD.

Под целую часть отводится 4 разряда, под дробную – 16 разрядов.

Смещение VDU_SCALER_BASE + 0010H. Разрядность - 20 разрядов. Разбиение на поля приведено в таблице 1.52.

Таблица 1.75 - Разряды регистра коэффициента горизонтального масштабирования для цветоразностных компонент (SCALER _SCH_C)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
19 – 16	SC_INT_HC	значение целой части коэффициента	rw	0H
15 - 0	SC_FRAC_HC	значение дробной части коэффициента	rw	0000H

Регистр коэффициента вертикального масштабирования для цветоразностных компонент (SCALER _SCV_C)

Регистр содержит коэффициенты для масштабирования плоскости цветоразностной компоненты изображения по вертикали. Коэффициент задается в виде шага масштабирования, определяемого по формуле ((кол-во строк до масштабирования – 1) / (кол-во строк после масштабирования – 1)). Коэффициенты представлены в виде целой и дробной частей, обе в виде беззнакового целого. Дробная часть коэффициента получена умножением исходной вещественной дробной части на 65536 (берется целая часть – дробная отбрасывается).

При увеличении – целая часть коэффициента является нулевой. При уменьшении коэффициент масштабирования больше 1.

Согласно сценариям применения видеоконтроллера, увеличение изображения производится от значений изображения с разрешениями меньше SD (но не менее CIF) до изображения с SD разрешением. А уменьшение изображения от изображений с разрешением HD до изображения с SD.

Под целую часть отводится 4 разряда, под дробную – 16 разрядов.

Смещение VDU_SCALER_BASE + 0014H. Разрядность - 20 разрядов. Разбиение на поля приведено в таблице 1.53.

Таблица 1.76 - Разряды регистра коэффициента вертикального масштабирования для цветоразностных компонент (SCALER _SCV_C)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
19 – 16	SC_INT_VC	значение целой части коэффициента	rw	0H
15 - 0	SC_FRAC_VC	значение дробной части коэффициента	rw	0000H

Регистр размера изображения после масштабирования для яркостной компоненты (SCALER _SIZE_Y)

Регистр содержит информацию о вертикальном (количество строк) и горизонтальном (количество пикселов в строке) размерах изображения после его масштабирования для яркостной компоненты.

При этом все размеры задаются на единицу меньше физических, т.е. для того, чтобы получить реальное количество пикселов или строк изображения необходимо к указанным в регистрах размерам добавить единицу. Также действует ограничение – количество пикселов и количество строк должно быть четным значением.

Пример: физический размер изображения: 720x576, значения параметров в регистрах: SIZE_VY = 576 – 1 = 575, SIZE_HY = 720 – 1 = 719.

Смещение VDU_SCALER_BASE + 0018H. Разрядность - 32 разряда. Разбиение на поля приведено в таблице 1.54.

Таблица 1.77 - Разряды регистра размера изображения после масштабирования для яркостной компоненты (SCALER _SIZE_Y)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 28	RESERVED			
27 – 16	SIZE_VY	количество строк изображения после масштабирования для яркостной компоненты	rw	000H
15 – 12	RESERVED			
11 - 0	SIZE_HY	количество пикселов в строке изображения после масштабирования для яркостной компоненты	rw	000H

Регистр размера изображения после масштабирования для цветоразностной компоненты (SCALER _SIZE_C)

Регистр содержит информацию о вертикальном (количество строк) и горизонтальном (количество пикселов в строке) размерах изображения после его масштабирования для цветоразностной компоненты.

При этом все размеры задаются на единицу меньше физических, т.е. для того, чтобы получить реальное количество пикселов или строк изображения необходимо к указанным в регистрах размерам добавить единицу. Также действует ограничение – количество пикселов и количество строк должно быть четным значением.

Пример: физический размер изображения: 720x576, значения параметров в регистрах: SIZE_VC = 576 – 1 = 575, SIZE_HC = 720 – 1 = 719.

Смещение VDU_SCALER_BASE + 001CH. Разрядность - 32 разряда. Разбиение на поля приведено в таблице 1.55.

Таблица 1.78 - Разряды регистра размера изображения после масштабирования для цветоразностной компоненты (SCALER _SIZE_C)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 28	RESERVED			
27 – 16	SIZE_VC	количество строк изображения после масштабирования для цветоразностной компоненты	rw	000H
15 – 12	RESERVED			
11 - 0	SIZE_HC	количество пикселов в строке изображения после масштабирования для цветоразностной компоненты	rw	000H

Регистр нулевого коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C0)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для яркостной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для нулевого звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Смещение VDU_SCALER_BASE + 0020H. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.56.

Таблица 1.79 - Разряды регистра нулевого коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C0)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C0_VY	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C1_VY/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C0_HY	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C1_HY/ HFLT_Y_N)	rw	00H

Регистр первого коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C1)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для яркостной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для первого звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8-миразрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Смещение VDU_SCALER_BASE + 0024H. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.57.

Таблица 1.80 - Разряды регистра первого коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C1)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C1_VY	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C2_VY/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C1_HY	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C2_HY/ HFLT_Y_N)	rw	00H

Регистр второго коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C2)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для яркостной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для второго звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Смещение VDU_SCALER_BASE + 0028H. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 33.

Таблица 1.81 - Разряды регистра второго коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C2)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C2_VY	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C3_VY/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C2_HY	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C3_HY/ HFLT_Y_N)	rw	00H

Регистр третьего коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C3)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для яркостной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для третьего звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Данный регистр в данной версии видеоконтроллера не используется.

Смещение VDU_SCALER_BASE + 002CH. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.59.

Таблица 1.82 - Разряды регистра третьего коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C3)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C3_VY	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C3_VY/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C3_HY	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C3_HY/ HFLT_Y_N)	rw	00H

Регистр четвертого коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C4)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для яркостной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для четвертого звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Данный регистр в данной версии видеоконтроллера не используется.

Смещение VDU_SCALER_BASE + 0030H. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.60.

Таблица 1.83 - Разряды регистра четвертого коэффициента фильтра устройства масштабирования для яркостной компоненты (SCALER_FLT_Y_C4)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C4_VY	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C4_VY/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C4_HY	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C4_HY/ HFLT_Y_N)	rw	00H

Регистр коэффициентов нормализации горизонтального фильтра для яркостной компоненты (SCALER_FLT_NORM)

Коэффициент нормализации представляет собой восьми разрядное целочисленное значение кратное степени 2. Данный коэффициент задает значение, на которое были домножены вещественные коэффициенты фильтра для получения целочисленного значения. Таким образом, для того, чтобы получить реальное значение коэффициента фильтра надо его целочисленное значение, задаваемое в регистрах разделить на коэффициент нормализации.

Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256. В регистр коэффициента нормализации заносится значение равное коэффициенту нормализации – 1.

Данный регистр в данной версии видеоконтроллера не используется, значение коэффициента нормализации фиксировано и равно 256.

Смещение VDU_SCALER_BASE + 0048H. Разрядность - 32 разряда. Разбиение на поля приведено в таблице 1.61.

Таблица 1.84 - Разряды регистра коэффициентов нормализации горизонтального фильтра для яркостной компоненты (SCALER_FLT_NORM)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 24	VFLT_C_N	значение коэффициента нормализации вертикального фильтра цветоразностной компоненты	rw	00H
23 - 16	HFLT_C_N	значение коэффициента нормализации горизонтального фильтра цветоразностной компоненты	rw	00H
15 - 8	VFLT_Y_N	значение коэффициента нормализации вертикального фильтра яркостной компоненты	rw	00H
7 - 0	HFLT_Y_N	значение коэффициента нормализации горизонтального фильтра яркостной компоненты	rw	00H

Регистр нулевого коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C0)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для цветоразностной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для нулевого звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Смещение VDU_SCALER_BASE + 004CH. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.62.

Таблица 1.85 - Разряды регистра нулевого коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C0)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C0_VC	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C0_VC/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C0_HC	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C0_HC/ HFLT_Y_N)	rw	00H

Регистр первого коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C1)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для цветоразностной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для первого звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Смещение VDU_SCALER_BASE + 0050H. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.63.

Таблица 1.86 - Разряды регистра первого коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C1)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C1_VC	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C1_VC/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C1_HC	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C1_HC/ HFLT_Y_N)	rw	00H

Регистр второго коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C2)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для цветоразностной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для второго звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8-миразрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Смещение VDU_SCALER_BASE + 0054H. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.64.

Таблица 1.87 - Разряды регистра второго коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C2)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C2_VC	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C0_VC/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C2_HC	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C0_HC/ HFLT_Y_N)	rw	00H

Регистр третьего коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C3)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для цветоразностной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для третьего звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Данный регистр в данной версии видеоконтроллера не используется.

Смещение VDU_SCALER_BASE + 0058H. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.65.

Таблица 1.88 - Разряды регистра третьего коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C3)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C3_VC	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C0_VC/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C3_HC	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C0_HC/ HFLT_Y_N)	rw	00H

Регистр четвертого коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C4)

Регистр содержит коэффициенты для усредняющей фильтрации видеоданных на входе устройства масштабирования для цветоразностной компоненты. Данный регистр содержит горизонтальный и вертикальный коэффициенты для четвертого звена фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

Данный регистр в данной версии видеоконтроллера не используется.

Смещение VDU_SCALER_BASE + 005CH. Разрядность - 16 разрядов. Разбиение на поля приведено в таблице 1.66.

Таблица 1.89 - Разряды регистра четвертого коэффициента фильтра устройства масштабирования для цветоразностной компоненты (SCALER_FLT_C_C4)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
15 – 8	FLT_C4_VC	значение коэффициента вертикального фильтра (реальное значение коэффициента определяется FLT_C0_VC/ HFLT_Y_N)	rw	00H
7 - 0	FLT_C4_HC	значение коэффициента горизонтального фильтра (реальное значение коэффициента определяется FLT_C0_HC/ HFLT_Y_N)	rw	00H

Регистр размера изображения после обрезания (SCALER_SIZE_CUT)

Данный регистр содержит информацию о горизонтальном и вертикальном размерах после обрезания изображения (обрезание производится после масштабирования). Горизонтальный размер - количество пикселов в строке. Вертикальный – количество строк в изображении. Так как на выходе scaler'а имеем формат видеоданных YCbCr 4:4:4, то данные значения применимы как для яркостной компоненты, так и для цветоразностной.

При этом все размеры задаются на единицу меньше физических, т.е. для того, чтобы получить реальное количество пикселов или строк изображения необходимо к указанным в регистрах размерам добавить единицу. Также действует ограничение – количество пикселов и количество строк должно быть четным значением.

Смещение VDU_SCALER_BASE + 0074H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.67.

Таблица 1.90 - Разряды регистра размера изображения после обрезания (SCALER_SIZE_CUT)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 27	RESERVED			
26 – 16	SIZE_CUT_V	вертикальный размер	rw	000H
15 – 11	RESERVED			
10 - 0	SIZE_CUT_H	горизонтальный размер	rw	000H

Горизонтальный регистр количества обрезаемых пикселов (SCALER_PHS_CUT_H)

Данный регистр содержит информацию о количестве обрезаемых пикселов изображения. Указанное количество пикселов обрезается от каждой строки с каждой стороны изображения, с каждой по PHASE_CUT_H (слева по PHASE_CUT_H_L, справа по PHASE_CUT_H_R).

Так как на выходе схемы масштабирования имеем формат видеоданных YCbCr 4:4:4, то данное значение применимо как для яркостной компоненты, так и для цветоразностной.

Для данных параметров все размеры задаются равными физическим, т.е. реальное количество пикселов изображения соответствует указанным в регистрах размерам (добавлять единицу не нужно). Действует ограничение – количество обрезаемых пикселов должно быть четным значением.

Смещение VDU_SCALER_BASE + 0078H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 43.

При этом

SCALER_SIZE_H = SCALER_SIZE_CUT_H + SCALER_PHASE_CUT_H_L + SCALER_PHASE_CUT_H_R,

Таблица 1.91 - Разряды регистра количества обрезаемых пикселов (SCALER_PHS_CUT_H)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 27	RESERVED			
26 – 16	PHASE_CUT_H_L	количество пикселов, обрезаемых с левой стороны строки изображения	rw	000H
15 – 11	RESERVED			
10 - 0	PHASE_CUT_H_R	количество пикселов, обрезаемых с правой стороны строки изображения	rw	000H

Вертикальный регистр количества обрезаемых пикселов (SCALER_PHS_CUT_V)

Данный регистр содержит информацию о количестве обрезаемых строк изображения.

Указанное количество строк обрезается сверху и снизу изображения, с каждой по PHASE_CUT_V (сверху - PHASE_CUT_V_T, снизу PHASE_CUT_V_B). Обрезание изображения производится после масштабирования.

Так как на выходе схемы масштабирования имеем формат видеоданных YCbCr 4:4:4, то данное значение применимо как для яркостной компоненты, так и для цветоразностной.

Для данных параметров все размеры задаются равными физическим, т.е. реальное количество строк изображения соответствует указанным в регистрах размерам (добавлять единицу не нужно). Действует ограничение – количество обрезаемых строк должно быть четным значением.

Смещение VDU_SCALER_BASE + 0070H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.69.

При этом

SCALER_SIZE_V = SCALER_SIZE_CUT_V + SCALER_PHASE_CUT_V_T + SCALER_PHASE_CUT_V_B

Таблица 1.92 - Разряды регистра количества обрезаемых пикселов (SCALER_PHS_CUT_V)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 26	RESERVED			
25 – 16	PHASE_CUT_V_T	количество строк, обрезаемых сверху изображения	rw	000H
15 – 10	RESERVED			
9 - 0	PHASE_CUT_V_B	количество строк, обрезаемых снизу изображения	rw	000H

Регистр матрицы коэффициентов для преобразования яркостной составляющей (SCALER_MVL_CLR_Y)

Регистр содержит коэффициенты для проведения колориметрического преобразования видеоданных SD качества (YCbCr601) в HD качество (YCbCr709) и наоборот. Данный регистр содержит коэффициенты для преобразования яркостной компоненты.

Коэффициенты представлены в виде 10 разрядных целых знаковых чисел (старший (десятый) разряд является знаковым). Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на 256.

Примеры значений коэффициентов приведены в документе: Спецификация на блок преобразования форматов цвета в канале основного видеослоя.

Смещение VDU_SCALER_BASE + 007CH. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.70.

Таблица 1.93 - Разряды регистра матрицы коэффициентов для преобразования яркостной составляющей (SCALER_MVL_CLR_Y)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 30	RESERVED			
29 – 20	MVL_CLR_Y_Cr	коэффициент для перемножения со значением компоненты Cr входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H
19 – 10	MVL_CLR_Y_Cb	коэффициент для перемножения со значением компоненты Cb входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H
9 - 0	MVL_CLR_Y_Y	коэффициент для перемножения со значением компоненты Y входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H

Регистр матрицы коэффициентов для преобразования цветоразностной составляющей Cb (SCALER_MVL_CLR_Cb)

Регистр содержит коэффициенты для проведения колориметрического преобразования видеоданных SD качества (YCbCr601) в HD качество (YCbCr709) и наоборот. Данный регистр содержит коэффициенты для преобразования цветоразностной Cb компоненты.

Коэффициенты представлены в виде 10 разрядных целых знаковых чисел (старший (десятый) разряд является знаковым). Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на 256.

Примеры значений коэффициентов приведены в документе: Спецификация на блок преобразования форматов цвета в канале основного видеослоя.

Смещение VDU_SCALER_BASE + 0080H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.71.

Таблица 1.94 - Разряды регистра матрицы коэффициентов для преобразования цветоразностной составляющей Cb (SCALER_MVL_CLR_Cb)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 30	RESERVED			
29 – 20	MVL_CLR_Cb_Cr	коэффициент для перемножения со значением компоненты Cr входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H
19 – 10	MVL_CLR_Cb_Cb	коэффициент для перемножения со значением компоненты Cb входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H
9 - 0	MVL_CLR_Cb_Y	коэффициент для перемножения со значением компоненты Y входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H

Регистр матрицы коэффициентов для преобразования цветоразностной составляющей Cr (SCALER_MVL_CLR_Cr)

Регистр содержит коэффициенты для проведения колориметрического преобразования видеоданных SD качества (YCbCr601) в HD качество (YCbCr709) и наоборот. Данный регистр содержит коэффициенты для преобразования цветоразностной Cr компоненты.

Коэффициенты представлены в виде 10 разрядных целых знаковых чисел (старший (десятый) разряд является знаковым). Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на 256.

Примеры значений коэффициентов приведены в документе: Спецификация на блок преобразования форматов цвета в канале основного видеослоя.

Смещение VDU_SCALER_BASE + 0084H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.72.

Таблица 1.95 - Разряды регистра матрицы коэффициентов для преобразования цветоразностной составляющей Cr (SCALER_MVL_CLR_Cr)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 30	RESERVED			
29 – 20	MVL_CLR_Cr_Cr	коэффициент для перемножения со значением компоненты Cr входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H
19 – 10	MVL_CLR_Cr_Cb	коэффициент для перемножения со значением компоненты Cb входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H
9 - 0	MVL_CLR_Cr_Y	коэффициент для перемножения со значением компоненты Y входного сигнала 9 разряд – знак 8-0 – значение коэффициента	rw	000H

1.3.5.1.6.1.5 VDU DIF - регистры цифровых видеоинтерфейсов

К регистрам цифровых видеоинтерфейсов относятся регистры, управляющие работой таких блоков как: блок наложения и позиционирования видеослоев, генератор временных разверток и выходных цифровых интерфейсов видеоконтроллера.

Таблица 1.96 - Регистры цифровых видеоинтерфейсов (VDU DIF)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
DIF_CTRL	регистр управления цифровых видеоинтерфейсов	VDU_DIF_BASE + 0004H	rw	32	0000_0000H
DIF_BGR	регистр цвета фона	VDU_DIF_BASE + 0008H	rw	24	0000_0000H
DIF_MVL_START	регистр координат основного видеослоя на экране	VDU_DIF_BASE + 000CH	rw	32	0000_0000H
DIF_BLANK	регистр размеров интервалов межстрочной и межкадровой разверток	VDU_DIF_BASE + 0010H	rw	32	0000_0000H
DIF_FSIZE	регистр размеров кадра видеоизображения	VDU_DIF_BASE + 0014H	rw	24	000_0000H
DIF_ASIZE	регистр размеров активной части видеоизображения	VDU_DIF_BASE + 0018H	rw	24	000_0000H
DIF_HSYNC	регистр позиции сигнала горизонтальной синхронизации	VDU_DIF_BASE + 001CH	rw	32	0000_0000H
DIF_VSYNC	позиции сигнала вертикальной синхронизации	VDU_DIF_BASE + 0020H	rw	32	0000_0000H
DIF_CTRL2	регистр управления цифровых видеоинтерфейсов N2	VDU_DIF_BASE + 0024H	rw	32	0000_0000H

Регистр управления цифровым видеоинтерфейсом (DIF_CTRL)

Данный регистр содержит управляющую информацию для цифровых видеоинтерфейсов видеоконтроллера.

Смещение VDU_DIF_BASE + 0004H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.74.

Таблица 1.97 - Разряды регистра управления цифровых видеоинтерфейсов (DIF_CTRL)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31	DIF_444_MODE	Выбор формата данных в случае их вывода по трем шинам (SDTV_FORM = 1): ‘0’ - на выход подаются данные в формате YCBCR 4:2:2; ‘1’ - на выход подаются данные в формате YCBCR 4:4:4. В случае вывода данных по шинам в количестве отличном от трех (SDTV_FORM = 0), бит DIF_444_MODE должен быть установлен в ноль.	rw	0H
30 - 25	RESERVED			
24	DIF_CH_SEL	Выбор канала наложения: ‘0’ – наложение канала OSD_0 на канал MVL. На полученное результирующее изображение впоследствии накладывается OSD_1 (если активно). ‘1’ – наложение канала OSD_1 на канал MVL. Полученное результирующее изображение впоследствии непрозрачно накладывается на OSD_0 (если активно).	rw	0H
23 - 18	RESERVED			
17	VSYNC_P	Полярность сигнала кадровой развертки (VSYNC). Определяет уровень напряжения выходного сигнала синхронизации вертикальной развертки: ‘0’ – активный уровень импульса синхронизации – ноль. В период прямого хода луча по вертикали VSYNC равен единице, а в период обратного – вырабатывается импульс синхронизации длительностью несколько строк с активным уровнем ноль; ‘1’ – активный уровень импульса синхронизации – единица. Во время прямого хода луча по вертикали VSYNC равен нулю, а во время обратного вырабатывается импульс синхронизации длительностью несколько строк с активным уровнем единица.	rw	0H
16	HSYNC_P	Полярность сигнала строчной развертки (HSYNC). Определяет уровень напряжения выходного сигнала синхронизации горизонтальной развертки: ‘0’ – активный уровень импульса синхронизации – ноль. В период прямого хода луча по горизонтали HSYNC равен единице, а в период обратного вырабатывается импульс синхронизации длительностью несколько тактов с активным уровнем ноль; ‘1’ – активный уровень импульса синхронизации – единица. В период прямого хода луча по горизонтали HSYNC равен нулю, а в период обратного вырабатывается импульс синхронизации длительностью несколько тактов с активным уровнем единица.	rw	0H
15 – 11	RESERVED			
10	SDTV_FORM (3BUS_EN)	Разрешение вывода данных по трем шинам: ‘0’ – вывод данных по одной шине для видео формата SD (MODE[4] = 0), либо по двум шинам для видео формата HD (MODE[4] = 1); ‘1’ – вывод данных по трем шинам для видео SD и HD форматов. В случае вывода видео SD формата, каждый пиксель в строке повторяется дважды.	rw	0H

Продолжение таблицы 1.98

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
9	EXT_SYNC_EN	Разрешение выдачи сигналов внешней синхронизации: ‘1’ – видеоданные сопровождаются сигналами развертки field, hsync, vsync, data_en; ‘0’ – сигналы развертки field, hsync, vsync, data_en – не активны (находятся в состоянии нуля).	rw	0H
8	INT_SYNC_EN	Разрешение выдачи кодов внутренней синхронизации: ‘1’ – видеоданные содержат в себе синхроданные в виде кодов SAV, EAV. ‘0’ – видеоданные не содержат в себе коды синхронизации SAV, EAV.	rw	0H
7 – 5	RESERVED			
4 - 0	MODE	Формат выходного видеоизображения. Нумеруется в соответствии с таблицей:	rw	00H

4 S/H	3 2 Size	1 I/P	0 60/ 50	-разряды Mode	pix_clk (МГц)
0	00	0	0	00h - SDTV 720x486i 60Hz	27
0	00	0	1	01h - SDTV 720x576i 50Hz	27
1	00	0	0	10h - EDTV 720(1440)x480i 60Hz	27
1	00	0	1	11h - EDTV 720(1440)x576i 50Hz	27
1	00	1	0	12h - EDTV 720x480p 60Hz	27
1	00	1	1	13h - EDTV 720x576p 50Hz	27
1	01	1	0	16h - HDTV 1280x720p 60Hz	74,25
1	01	1	1	17h - HDTV 1280x720p 50Hz	74,25
1	10	0	0	18h - HDTV 1920x1080i 60Hz	74,25
1	10	0	1	19h - HDTV 1920x1080i 50Hz	74,25
1	10	1	0	1Ah - HDTV 1920x1080p 30Hz	74,25
1	10	1	1	1Bh - HDTV 1920x1080p 25Hz	74,25

другие значения зарезервированы.

Регистр цвета фона (DIF_BGR)

Данный регистр содержит информацию о цвете фонового слоя. Информация задается в формате YCbCr.

Смещение VDU_DIF_BASE + 0008H. Разрядность 24 разряда. Разбиение на поля приведено в таблице 1.75.

Таблица 1.99 - Разряды регистра цвета фона (DIF_BGR)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 24	RESERVED			
23 - 16	BGR_Y	значение яркостной составляющей Y (диапазон допустимых значений 16-235)	rw	00H
15 - 8	BGR_Cb	значение цветоразностной составляющей Cb (диапазон допустимых значений 16-240)	rw	00H
7 – 0	BGR_Cr	значение цветоразностной составляющей Cr (диапазон допустимых значений 16-240)	rw	00H

Регистр координат основного видеослоя на экране (DIF_MVL_START)

Данный регистр содержит информацию о начальных координатах положения основного видеоизображения на экране. Координаты задаются для левого верхнего угла видеоизображения относительно левого верхнего угла видимой области изображения. Рекомендуется устанавливать четные значения координат.

Смещение VDU_DIF_BASE + 000CH. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.76.

Таблица 1.100 - Разряды регистра координат основного видеослоя на экране (DIF_MVL_START)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 27	RESERVED			
26 – 16	MVL_START_V	значение вертикальной координаты	rw	000H
15 – 11	RESERVED			
10 – 0	MVL_START_H	значение горизонтальной координаты	rw	000H

Регистр размеров интервалов межстрочной и межкадровой разверток (DIF_BLANK)

Данный регистр содержит информацию о размере интервалов межстрочной и межкадровой разверток.

Смещение VDU_DIF_BASE + 0010H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.77.

Таблица 1.101 - Разряды регистра размеров интервалов межстрочной и межкадровой разверток (DIF_BLANK)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 27	RESERVED			
26 – 24	VBLANK_END	размер интервала межкадровой развертки (от конца активной части видеоизображения до начала нового кадра), измеряется в линиях		
23 – 22	RESERVED			
21 – 16	VBLANK_BEG	размер интервала межкадровой развертки (от начала кадра до активной части видеоизображения), измеряется в линиях		
15 – 10	RESERVED			
9 – 0	HBLANK	размер интервала межстрочной развертки в пикселах -1	rw	000H

Регистр размеров кадра видеоизображения (DIF_FSIZE)

Данный регистр содержит информацию о размерах полного видеоизображения (включая активную часть видеоизображения и области межстрочной и межкадровой разверток)

Смещение VDU_DIF_BASE + 0014H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.78.

Таблица 1.102 - Разряды регистра размеров кадра видеоизображения (DIF_FSIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 27	RESERVED			
26 – 16	VTOTAL	вертикальный размер (в линиях) кадра видеоизображения	rw	000H
15 – 12	RESERVED			
11 – 0	HTOTAL	горизонтальный размер (в пикселах) кадра видеоизображения	rw	00H

Регистр размеров активной части видеоизображения (DIF_ASIZE)

Данный регистр содержит информацию о размерах полного видеоизображения (включая активную часть видеоизображения и области межстрочной и межкадровой разверток)

Смещение VDU_DIF_BASE + 0018H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.79.

Таблица 1.103 - Разряды регистра размеров активной части видеоизображения (DIF_ASIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 27	RESERVED			
26 – 16	VACTIVE	вертикальный размер (в линиях) видимого изображения	rw	000H
15 – 11	RESERVED			
10 – 0	HACTIVE	горизонтальный (в пикселах - 1) активной части видеоизображения	rw	00H

Регистр позиции сигнала горизонтальной синхронизации (DIF_ HSYNC)

Данный регистр содержит информацию о положении и длительности сигнала синхронизации горизонтальной развертки.

Смещение VDU_DIF_BASE + 001CH. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.80.

Таблица 1.104 - Разряды регистра позиции сигнала горизонтальной синхронизации (DIF_HSYNC)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 24	HSYNC_DELAY	длина промежутка между окончанием импульса горизонтальной развертки и началом активной части видеолинии, измеряется в пикселах	rw	
23 – 16	HSYNC_LEN	длина (количество пикселов) сигнала синхронизации горизонтальной развертки HSYNC, измеряется в пикселах	rw	0000H
15 – 10	RESERVED			
9 – 0	HSYNC_START	позиция (в пикселах) установки сигнала горизонтальной развертки HSYNC в активное состояние относительно начала новой видеостроки (начала интервала blank новой видеостроки)	rw	0000H

Регистр позиции сигнала вертикальной синхронизации (DIF_VSYNC)

Данный регистр содержит информацию о положении и длительности сигнала вертикальной синхронизации развертки.

Смещение VDU_DIF_BASE + 0020H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.81.

Таблица 1.105 - Разряды регистра позиции сигнала вертикальной синхронизации (DIF_VSYNC)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 19	RESERVED			
18 – 16	VSYNC_LEN	длина (количество строк) сигнала синхронизации вертикальной развертки VSYNC	rw	0000H
15 – 3	RESERVED			
2 – 0	VSYNC_START	позиция (в строках) установки сигнала вертикальной развертки VSYNC в активное состояние	rw	0000H

Регистр управления цифровым видеоинтерфейсом N2 (DIF_CTRL2)

Данный регистр содержит управляющую информацию для цифровых видеоинтерфейсов видеоконтроллера.

Смещение VDU_DIF_BASE + 0024H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.82.

Таблица 1.106 - Разряды регистра управления цифровых видеоинтерфейсов (DIF_CTRL2)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 9	RESERVED			
8	REPLIC_MODE	Выбор режима репликации цветностной составляющей пикселов (используется для режимов IEDTV при выводе данных по двум шинам: MODE = 0x10 или 0x11, SDTV_FORM = 0): ‘0’ – производится репликация цветоразностных пар: Cb0Cr0Cb0Cr0; ‘1’ – производится репликация компонент в каждой цветоразностной паре: Cb0Cb0Cr0Cr0; Для всех остальных видеорежимов данный бит должен быть установлен в значение ‘0’.	rw	0H
7 - 1	RESERVED			
0	AVR_MODE	Выбор метода усреднения значений цветности при преобразовании внутреннего формата представления данных YCbCr4:4:4 во внешний – YCbCr4:2:2: ‘0’ – каждой паре пикселов присваивается значение цветности первого пиксела из пары. ‘1’ – каждой паре пикселов присваивается значение цветности равное усредненному значению цветностей каждого пиксела из пары. В случае вывода данных в формате YCbCb4:4:4 (DIF_444_MODE = 1) данный бит должен быть установлен в значение ‘0’.	rw	0H

1.3.5.1.6.1.6 VDU OSD - регистры графического слоя

Таблица 1.107 - Регистры графического слоя (VDU OSD)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
OSD_BASEreg0	адрес первого заголовка графической области, набор регистров reg0 (A)	VDU_OSD_BASE + 0000H	rw	32	0000_0000H
OSD_BASEreg1	адрес первого заголовка графической области, набор регистров reg1 (B)	VDU_OSD_BASE + 0004H	rw	32	0000_0000H
OSD_COLOR_Y	коэффициенты для преобразования формата ARGB в AYCbCr яркостной компоненты	VDU_OSD_BASE + 0008H	rw	32	0000_0000H
OSD_COLOR_Cb	коэффициенты для преобразования формата ARGB в AYCbCr цветоразностной компоненты Cb	VDU_OSD_BASE + 000CH	rw	32	0000_0000H
OSD_COLOR_Cr	коэффициенты для преобразования формата ARGB в AYCbCr цветоразностной компоненты Cr	VDU_OSD_BASE + 0010H	rw	32	0000_0000H
OSD_FIFO_END	размер входного буфера графического слоя	VDU_OSD_BASE + 0014H	rw	32	0000_007FH
OSD_CTRL	управляющий регистр графического слоя	VDU_OSD_BASE + 0018H	rw	32	0000_0000H

Регистр адреса первого заголовка графической области (OSD_BASEreg0, OSD_BASEreg1)

Регистр содержит базовый адрес первого заголовка графической области для данного кадра. По данному адресу из внешней памяти извлекается заголовок графического слоя, который содержит в себе адрес заголовка для следующей графической области данного кадра. Таким образом, заголовки областей организованы в виде связанного списка.

Существует два набора данных регистров с индексом reg0 и reg1. При этом, в текущий момент времени один из наборов, например, _reg0, содержит значение текущего базового адреса – того по которому в данный момент времени ведется чтение видеоданных из внешней памяти, при этом второй набор, в данном случае _reg1 содержит базовый адрес положения в памяти следующего кадра. После окончания чтения видеостраницы по данному базовому адресу статус наборов регистра переключается и reg0 станет следующим, а reg1 – текущим. Индикация того, какой из наборов регистров в данный момент является текущим ведется в регистре STAT_VDU_AVMP.

Базовый адрес заголовка имеет выравнивание по границе 64х разрядного слова. Таким образом, младшие три разряда являются незначащими, адрес содержится в старших 29 разрядах.

Смещение VDU_OSD_BASE + 0000H, 0004H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.84.

Таблица 1.108 - Разряды регистра первого заголовка графической области (OSD_BASEreg0)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 3	OSD_BASE	адрес первого заголовка графической области	rw	0000_0000H
2 – 0	RESERVED			

Регистр коэффициентов для преобразования формата ARGB в AYCbCr яркостной компоненты (OSD_COLOR_Y)

Регистр содержит матрицу коэффициентов для преобразования формата представления цвета пиксела из ARGB в AYCbCr для формирования значения яркостной компоненты.

Смещение VDU_OSD_BASE + 0008H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.85. Коэффициенты задаются 9-ти разрядными значениями из которых 8 значимых разрядов (коэффициент) и старший девятый разряд – знак.

Таблица 1.109 - Разряды регистра коэффициентов для преобразования формата ARGB в AYCbCr яркостной компоненты (OSD_COLOR_Y)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 29	RESERVED			
28 – 20	OSD_COLOR_Y_B	множитель для поля синего цвета	rw	0_0000H
19	RESERVED			
18 – 10	OSD_COLOR_Y_G	множитель для поля зеленого цвета	rw	0_0000H
9	RESERVED			
8 – 0	OSD_COLOR_Y_R	множитель для поля красного цвета	rw	0_0000H

Регистр коэффициентов для преобразования формата ARGB в AYCbCr цветоразностной компоненты Cb (OSD_COLOR_Cb)

Регистр содержит матрицу коэффициентов для преобразования формата представления цвета пиксела из ARGB в AYCbCr для формирования значения цветоразностной компоненты Cb.

Смещение VDU_OSD_BASE + 000CH. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.86. Коэффициенты задаются 9-ти разрядными значениями из которых 8 значимых разрядов (коэффициент) и старший девятый разряд – знак.

Таблица 1.110 - Разряды регистра коэффициентов для преобразования формата ARGB в AYCbCr цветоразностной компоненты Cb (OSD_COLOR_Cb)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 29	RESERVED			
28 – 20	OSD_COLOR_CB_B	множитель для поля синего цвета	rw	0_0000H
19	RESERVED			
18 – 10	OSD_COLOR_CB_G	множитель для поля зеленого цвета	rw	0_0000H
9	RESERVED			
8 – 0	OSD_COLOR_CB_R	множитель для поля красного цвета	rw	0_0000H

Регистр коэффициентов для преобразования формата ARGB в AYCbCr цветоразностной компоненты Cr (OSD_COLOR_Cr)

Регистр содержит матрицу коэффициентов для преобразования формата представления цвета пиксела из ARGB в AYCbCr для формирования значения цветоразностной компоненты Cr.

Смещение VDU_OSD_BASE + 0010H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.87. Коэффициенты задаются 9-ти разрядными значениями из которых 8 значимых разрядов (коэффициент) и старший девятый разряд – знак.

Таблица 1.111 - Разряды регистра коэффициентов для преобразования формата ARGB в AYCbCr цветоразностной компоненты Cr (OSD_COLOR_Cr)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 29	RESERVED			
28 – 20	OSD_COLOR_CR_B	множитель для поля синего цвета	rw	0_0000H
19	RESERVED			
18 – 10	OSD_COLOR_CR_G	множитель для поля зеленого цвета	rw	0_0000H
9	RESERVED			
8 – 0	OSD_COLOR_CR_R	множитель для поля красного цвета	rw	0_0000H

Регистр размера входного буфера графического слоя (OSD_FIFO_END)

Регистр содержит значение конечного адреса входного буфера графического слоя, позволяет программно варьировать объем входного буфера.

Смещение VDU_OSD_BASE + 0014H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.88.

На данный момент регистр зафиксирован – значение 127 – размер буфера графического слова. В текущей версии видеоконтроллера, данный регистр не используется. Задаваемые значения не влияют на работу видеоконтроллера.

Таблица 1.112 - Разряды регистра входного буфера графического слоя (OSD_FIFO_END)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 10	RESERVED			
9 – 0	OSD_FIFO_END	значение конечного адреса	rw	0_007FH

Регистр управления графического слоя (OSD_CTRL)

Регистр содержит управляющую информацию для графического слоя.

Смещение VDU_OSD_BASE + 0018H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.89.

Таблица 1.113 - Разряды регистра управления графического слоя (OSD_CTRL)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 9	RESERVED			
8	OSD_RGB_YCBCR_BYPASS	определяет режим работы декодера цвета графического слоя 0 – декодер ведет преобразование входных ARGB данных в формат (A)YCbCr 1 – преобразование данных отключено, используется для подачи на вход графического слоя данных в формате AYCbCr 8:8:8:8	rw	0_0000H
7 – 1	RESERVED			
0	OSD_ARGB_RGBA	определяет формат данных в OSD слое 0 - RGBA 1 - ARGB	rw	0_0000H

1.3.5.1.6.1.7 VDU MI - регистры интерфейса с памятью

Таблица 1.114 - Регистры интерфейса с памятью (VDU MI)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
MI_CTRL	регистр управления интерфейса с памятью	VDU_MI_BASE + 0004H	rw	32	0000_0000H
MI_MVL_Y_BAreg0	базовый адрес доступной области яркостной компоненты, набор регистров reg0 (A)	VDU_MI_BASE + 0008H	rw	32	0000_0000H
MI_MVL_Cb_BAreg0	базовый адрес доступной области цветоразностной (Cb, CbCr) компоненты, набор регистров reg0 (A)	VDU_MI_BASE + 000CH	rw	32	0000_0000H
MI_MVL_Cr_BAreg0	базовый адрес доступной области цветоразностной (Cr) компоненты, набор регистров reg0 (A)	VDU_MI_BASE + 0010H	rw	32	0000_0000H
MI_MVL_Y_BAreg1	базовый адрес доступной области яркостной компоненты, набор регистров reg1 (B)	VDU_MI_BASE + 0014H	rw	32	0000_0000H
MI_MVL_Cb_BAreg1	базовый адрес доступной области цветоразностной (Cb, CbCr) компоненты, набор регистров reg1 (B)	VDU_MI_BASE + 0018H	rw	32	0000_0000H
MI_MVL_Cr_BAreg1	базовый адрес доступной области цветоразностной (Cr) компоненты, набор регистров reg1 (B)	VDU_MI_BASE + 001CH	rw	32	0000_0000H
MI_MVL_Y_SIZE	размер доступной области плоскости для яркостной компоненты	VDU_MI_BASE + 0020H	rw	32	0000_0000H
MI_MVL_C_SIZE	размер доступной области плоскости для цветоразностных компонент	VDU_MI_BASE + 0024H	rw	32	0000_0000H
MI_MVL_FULL_SIZE	полный (физический) размер строки плоскости	VDU_MI_BASE + 0028H	rw	32	0000_0000H
MI_AXI_MVL_PARAM	параметры контроллера ведущего на шине AXI для основного видеослоя	VDU_MI_BASE + 002CH	rw	32	0031_0000H
MI_MVL_FIFO_START	начальные адреса плоскостей компонент во входном буфере основного слоя	VDU_MI_BASE + 0030H	rw	32	0000_0000H
MI_MVL_FIFO_END	размер входного буфера основного слоя	VDU_MI_BASE + 0034H	rw	32	0000_0000H
MI_AXI_OSD_PARAM	параметры контроллера ведущего на шине AXI для графического слоя	VDU_MI_BASE + 0038H	rw	32	0031_0000H

Регистр управления интерфейса с памятью (MI_CTRL)

Регистр содержит управляющую информацию определяющую порядок нахождения видеоданных во внешней памяти.

Смещение VDU_MI_BASE + 0004H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.91.

Таблица 1.115 - Разряды регистра управления интерфейса с памятью (MI_CTRL)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31		Искусственный интерфейс для масштабирования		
30 - 10	RESERVED			
16	BUF_BIT_ENDIAN	формат представления данных во входном буфере основного слоя (big/little endian format) 0 - little endian format 1 - big endian format	rw	0
15 - 10	RESERVED			
9	YCbCr_FORMAT	формат представления видеоданных в памяти 0 – данные в памяти представлены в формате YCbCr 4:2:0 1 - данные в памяти представлены в формате YCbCr 4:2:2 (предварительно)	rw	0
8	PLANE_NUM	количество плоскостей компонент в памяти 0 – режим двойного разделения (плоскости Y, CbCr) 1 - режим тройного разделения (плоскости Y, Cb, Cr)	rw	0
7 - 3	RESERVED			
2	PLANE_ACCESS	режим доступа внутри плоскости 0 - линейный режим 1 - макроблочный режим (только при PLANE_NUM = ‘0’ и FILD_STORE = ‘0’)	rw	0
1	FILD_ACCESS	режим доступа к видеоизображению 0 - прогрессивный 1 - чересстрочный	rw	0
0	FILD_STORE	режим хранения полукадров при чересстрочном режиме изображения 0 – полукадры хранятся в памяти в одном прогрессивном кадре (чередуются строки) 1 - полукадры хранятся в разных областях памяти (предварительно) в данной версии видеоконтроллера этот параметр задается заданием разных базовых адресов расположения данных в памяти для разных полукадров	rw	0

Регистры базовых адресов доступной области для компонент (MI_MVL_Y_BAreg0, MI_MVL_Cb_BAreg0, MI_MVL_Cr_BAreg0, MI_MVL_Y_BAreg1, MI_MVL_Cb_BAreg1, MI_MVL_Cr_BAreg1)

Данные регистры содержат базовые адреса доступных областей для различных компонент видеоизображения.

Начальный адрес видеостраницы в памяти должен быть выровнен на границу слова памяти, то есть должен быть кратен 8. В том случае, если требуется вывод изображения начиная с не кратного адреса, базовый адрес все равно должен быть установлен кратным 8, а излишнее изображение может быть удалено в тракте обработки видеоинформации (горизонтальное обрезание в устройстве масштабирования).

Существует два набора данных регистров с индексом reg0 и reg1. При этом, в текущий момент времени один из наборов, например, _reg0, содержит значение текущего базового адреса – того по которому в данный момент времени ведется чтение видеоданных из внешней памяти,

при этом второй набор, в данном случае _reg1 содержит базовый адрес положения в памяти следующего кадра.

После окончания чтения видеостраницы по данному базовому адресу статус наборов регистра переключается и reg0 станет следующим, а reg1 – текущим (при разрешенном переключении страниц). Индикация того, какой из наборов регистров в данный момент является текущим ведется в регистре STAT_VDU_AVMP.

Структура регистров обоих наборов идентична, поэтому иллюстрируется только структура регистров набора reg0 (A).

Смещения VDU_MI_BASE + 0008H, 000CH, 0010H, 0014H, 0018H, 001CH соответственно. Разрядность 32 разряда. Разбиение на поля приведено в таблицах 1.92, 1.93, 1.94.

Таблица 1.116 - Разряды регистра базового адреса доступной области яркостной компоненты (MI_MVL_Y_BAreg0, MI_MVL_Y_BAreg1)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 0	MI_MVL_Y_BAreg0	базовый адрес доступной области яркостной компоненты	rw	0000_0000H

Таблица 1.117 - Разряды регистра базового адреса доступной области цветоразностной (Cb, CbCr) компоненты (MI_MVL_Cb_BAreg0, MI_MVL_Cb_BAreg1)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 0	MI_MVL_Cb_BAreg0	базовый адрес доступной области цветоразностной (Cb, CbCr) компоненты при MI_PLANE_NUM = ‘0’ – задает базовый адрес компоненты CbCr; при MI_PLANE_NUM = ‘1’ – задает базовый адрес компоненты Cb.	rw	0000_0000H

Таблица 1.118 - Разряды регистра базового адреса доступной области цветоразностной (Cr) компоненты (MI_MVL_Cr_BAreg0, MI_MVL_Cr_BAreg1)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 0	MI_MVL_Cr_BAreg0	базовый адрес доступной области цветоразностной (Cr) компоненты используется только при MI_PLANE_NUM = ‘1’	rw	0000_0000H

Регистр размера доступной области плоскости для яркостной компоненты (MI_MVL_Y_SIZE)

Регистр содержит горизонтальный и вертикальный размеры доступной области плоскости яркостной компоненты в памяти.

При этом все размеры задаются на единицу меньше физических, т.е. для того, чтобы получить реальное количество пикселов или строк изображения необходимо к указанным в регистрах размерам добавить единицу. Также действует ограничение – количество пикселов и количество строк должно быть четным значением.

Смещение VDU_MI_BASE + 0020H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.95.

Таблица 1.119 - Разряды регистра размера доступной области плоскости для яркостной компоненты (MI_MVL_Y_SIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 27	RESERVED			
26 - 16	MI_MVL_Y_V_SIZE	количество строк в доступной области плоскости для яркостной компоненты	rw	0_0000H
15 - 11	RESERVED			
10 – 0	MI_MVL_Y_H_SIZE	размер строки доступной области изображения в плоскости - количество точек в строке для чтения из памяти для яркостной компоненты	rw	0_0000H

Регистр размера доступной области плоскости для цветоразностных компонент (MI_MVL_C_SIZE)

Регистр содержит горизонтальный и вертикальный размеры доступной области плоскости цветоразностных компонент в памяти.

При этом все размеры задаются на единицу меньше физических, т.е. для того, чтобы получить реальное количество пикселов или строк изображения необходимо к указанным в регистрах размерам добавить единицу. Также действует ограничение – количество пикселов и количество строк должно быть четным значением.

Смещение VDU_MI_BASE + 0024H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.96.

Таблица 1.120 - Разряды регистра размера доступной области плоскости для цветоразностных компонент (MI_MVL_C_SIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 27	RESERVED			
26 - 16	MI_MVL_C_V_SIZE	количество строк в доступной области плоскости для цветоразностных компонент	rw	0_0000H
15 - 11	RESERVED			
10 – 0	MI_MVL_C_H_SIZE	размер строки доступной области изображения в плоскости - количество точек в строке для чтения из памяти для цветоразностных компонент	rw	0_0000H

Регистр полного (физического) размера строки плоскости компоненты (MI_MVL_FULL_SIZE)

Регистр содержит горизонтальный размер доступной области плоскости цветоразностных компонент в памяти.

При этом все размеры задаются равные физическим, т.е. для того, чтобы получить реальное количество пикселов или строк изображения нет необходимости к указанным в регистрах размерам добавить единицу. Также действует ограничение – количество пикселов и количество строк должно быть четным значением.

Значения для яркостной и цветоразностной составляющих задаются одинаковыми, без учета формата 4:2:2.

Смещение VDU_MI_BASE + 0028H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.97.

Таблица 1.121 - Разряды регистра полного (физического) размера строки плоскости компоненты (MI_MVL_FULL_SIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 – 29	Reserved			
28 – 16	MVL_C_FULL_SIZE	полный размер строки плоскости в пикселах, может быть равен или больше размера строки доступной области, для цветоразностной компоненты	rw	0000H
15 – 13	Reserved			
12 – 0	MVL_Y_FULL_SIZE	полный размер строки плоскости в пикселах, может быть равен или больше размера строки доступной области, для яркостной компоненты	rw	0000H

Регистр параметров контроллера ведущего на шине AXI для MVL (MI_AXI_MVL_PARAM)

Регистр содержит конфигурационные параметры, определяющие настройку интерфейса AXI для DMA канала основного видеослоя. Подробно назначение этих параметров описывается в спецификации на соответствующий интерфейс.

Смещение VDU_MI_BASE + 002CH. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.98.

Таблица 1.122 - Разряды регистра параметров контроллера ведущего на шине AXI (MI_AXI_PARAM)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 28	RESERVED			
27 - 24	ARLEN[3:0]	длина пакета задает количество посылок, считываемых при подаче одного адреса	rw	0000H
23	RESERVED			
22 – 20	ARSIZE[2:0]	разрядность каждой посылки в пакете размерность каждой посылки ограничена разрядностью интерфейса и всегда фиксирована, при этом она равна разрядности интерфейса ARSIZE = b011	r	b011
19 - 18	RESERVED			
17 – 16	ARBURST[1:0]	тип пакетной пересылки реализуется только INCR режим пакетной передачи (ARBURST = 2’b01)	r	b01
15 – 10	RESERVED			
9 – 8	ARLOCK[1:0]	параметр блокировки нет поддержки режимов обращения locked и exclusive	rw	0000H
7 – 4	ARCACHE[3:0]	режим работы с кэш нет поддержки системных КЭШей (сигнал ARCACHE не используется = 4’b0000)	rw	b00
3	RESERVED			
2 – 0	ARPROT[2:0]	параметр защищенного режима	rw	0000H

Регистр начальных адресов плоскостей компонент во входном буфере основного слоя (MI_MVL_FIFO_START)

Регистр содержит начальные адреса для различных плоскостей компонент во входном буфере основного видеослоя. Данный регистр позволяет производить программное разделение входного буфера на логические области.

Смещение VDU_MI_BASE + 0030H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.99.

Таблица 1.123 - Разряды регистра начальных адресов плоскостей компонент во входном буфере основного слоя (MI_MVL_FIFO_START)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 24	RESERVED			
23 - 16	MVL_Cr_FIFO_START	начальный адрес в MVL_BUF FIFO для цветоразностной компоненты	rw	0000H
15 - 8	MVL_Cb_FIFO_START	начальный адрес в MVL_BUF FIFO для цветоразностной компоненты	rw	0000H
7 - 0	MVL_Y_FIFO_START	начальный адрес в MVL_BUF FIFO для яркостной компоненты	rw	0000H

Регистр размера входного буфера основного слоя (MI_MVL_FIFO_END)

Регистр содержит значение конечного адреса входного буфера основного слоя, позволяет программно варьировать объем входного буфера.

Смещение VDU_MI_BASE + 0034H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.100.

Таблица 1.124 - Разряды размера входного буфера основного слоя (MI_MVL_FIFO_END)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 10	RESERVED			
9 – 0	MVL_FIFO_END	значение конечного адреса	rw	0_0000H

Регистр параметров контроллера ведущего на шине AXI для OSD (MI_AXI_OSD_PARAM)

Регистр содержит конфигурационные параметры, определяющие настройку интерфейса AXI для DMA канала графического слоя. Подробно назначение этих параметров описывается в спецификации на соответствующий интерфейс.

Смещение VDU_MI_BASE + 0038H. Разрядность 32 разряда. Разбиение на поля приведено в таблице 1.101.

Таблица 1.125 - Разряды регистра параметров контроллера ведущего на шине AXI (MI_AXI_PARAM)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31 - 28	RESERVED			
27 - 24	ARLEN[3:0]	длина пакета задает количество посылок, считываемых при подаче одного адреса	rw	0000H
23	RESERVED			
22 – 20	ARSIZE[2:0]	разрядность каждой посылки в пакете размерность каждой посылки ограничена разрядностью интерфейса и всегда фиксирована, при этом она равна разрядности интерфейса ARSIZE = b011	r	b011
19 - 18	RESERVED			
17 – 16	ARBURST[1:0]	тип пакетной пересылки реализуется только INCR режим пакетной передачи (ARBURST = 2’b01)	r	b01
15 – 10	RESERVED			
9 – 8	ARLOCK[1:0]	параметр блокировки нет поддержки режимов обращения locked и exclusive	rw	0000H
7 – 4	ARCACHE[3:0]	режим работы с кэш нет поддержки системных КЭШей (сигнал ARCACHE не используется = 4’b0000)	rw	b00
3	RESERVED			
2 – 0	ARPROT[2:0]	параметр защищенного режима	rw	0000H

1.3.5.1.7 Функциональное описание

1.3.5.1.7.1 Подготовка видеоконтроллера к использованию

1.3.5.1.7.1.1 Сброс видеоконтроллера и начальная инициализация

Для приведения видеоконтроллера в работоспособное состояние необходимо после включения питания произвести его сброс и начальную инициализацию.

Сброс видеоконтроллера может быть либо программным, либо аппаратным. Аппаратный сброс осуществляется посредством подачи низкого уровня сигнала на порт reset_i. Программный сброс осуществляется путем установки в единицу регистра CTRL_SOFT_RESET, при этом значение разряда VDU_OFF регистра состояния прерываний INT_VDU_STAT должно быть в единице, то есть видеоконтроллер должен находиться в остановленном (выключенном) состоянии, в противном случае программный сброс будет проигнорирован аппаратурой видеоконтроллера. После прохождения операции сброса значение регистра программного сброса CTRL_SOFT_RESET автоматически обнулится. Управляющие схемы видеоконтроллера окажутся обнуленными. Получается, что программный сброс может производиться только при выключенном видеоконтроллере, в то время как аппаратный сброс можно провести в любое время.

Следует отметить, что, как при аппаратном, так и при программном сбросах значения большинства конфигурационных программно доступных регистров видеоконтроллера не будут обнулены.

Сброс регистров видеоконтроллера (запись в них начальных значений) осуществляется исключительно программным способом. То есть программист должен перед использованием регистров произвести программным способом инициализацию всех конфигурационных регистров видеоконтроллера. Исключение составляют регистры прерываний (регистр разрешения прерываний INT_VDU_ENA и регистр состояния прерываний INT_VDU_STAT, а также регистр управления CTRL_SOFT_RESET, а также управляющие регистры в каждой группе регистров, а именно MI_CTRL, DIF_CTRL, SCAL_CTRL), которые переводятся в исходное состояние при приходе системного сигнала сброса, или сигнала программного сброса видеоконтроллера.

Рекомендуется после подачи сигнала системного или программного сброса провести программную инициализацию всех регистров видеоконтроллера.

1.3.5.1.7.1.2 Инициализация и выключение видеоконтроллера

Инициализация: после включения питания и прохождения операции сброса, все блоки видеоконтроллера будут находиться в исходном рабочем состоянии. Следующий шаг - конфигурирование контроллера на требуемый режим работы, для этого требуется записать соответствующие значения во все регистры контроллера, поскольку не все регистры сбрасываются во время операции сброса. Для запуска работы контроллера нужно, после настройки параметров в регистрах, установить единицу в разряде VDU_ENA в управляющем регистре видеоконтроллера VDU_CONTROL.

Выключение: для выключения устройства нужно записать в управляющий регистр VDU_CONTROL в разряд VDU_ENA значение ноль, после этой операции контроллер выведет текущий кадр (если в каналах обработки контроллера данные только текущего выводимого кадра) или текущий и следующий кадр (если в каналах обработки контроллера данные двух кадров, текущего выводимого кадра и следующего кадра) и завершит свою работу, выставив единицу в разряде VDU_OFF регистра состояния прерываний INT_VDU_STAT. После выключения можно произвести программный сброс или новое конфигурирование устройства.

1.3.5.1.7.2 Использование видеоконтроллера

1.3.5.1.7.2.1 Конфигурирование регистров видеоконтроллера

Конфигурирование регистров видеоконтроллера (запись в них значений) разрешено не во все моменты времени. Так изменение значений некоторых регистров во время активной фазы вывода изображения на экран может привести к неправильному функционированию видеоконтроллера и искажению итогового изображения. В этом разделе приводится информация о том, в какие моменты времени допустимо изменять значения конфигурационных регистров видеоконтроллера.

Можно выделить несколько состояний видеоконтроллера:

- видеоконтроллер выключен (неактивен) – признак VDU_OFF == 1;
- неактивная часть изображения (обратный ход луча кадровой развертки) = признак VBB == 1 (регистр STAT_VDU_DISP - Состояние отображения изображения);
- (замечание: в этом состоянии пользователь должен гарантировать, что в тракте обработки видеоконтроллера нет данных, на обработку которых может повлиять изменение данного параметра)
- неактивен основной слой видеоконтроллера (сам видеоконтроллер активен) в любой момент времени;
- неактивен графический слой видеоконтроллера (сам видеоконтроллер активен) в любой момент времени;
- любой момент времени.

Для того, чтобы гарантировать присутствие в канале обработки данных только одного кадра можно отключить признак разрешения переключения страниц (MVL_BASE_SW_ENA, OSD_BASE_SW_ENA), после окончания чтения предыдущего кадра. В этом случае данные для нового кадра в конвейер поступать не будут.

В таблице 1.102 приведен список регистров и даны рекомендации о возможном времени и причинах изменения состояния данных регистров (описание регистров смотри в разделе 1.4.5 Программный (регистровый) интерфейс).

Таблица 1.126 - Порядок изменения значений конфигурационных регистров

Регистр	Разряд	vdu_off == 1	vbb == 1	mvl_ena == 0	osd_ena == 0	all time	Пояснения
CTRL_VDU_ENA	MVL_BASE_SW_ENA	-	-	-	-	да	изменяется в случае готовности/не готовности видеостраницы в памяти
	OSD_BASE_SW_ENA	-	-	-	-	да	изменяется в случае готовности/не готовности видеостраницы в памяти
	MVL_ENA	да	да	нет	нет	нет	
	OSD_ENA	да	да	нет	нет	нет	
	VDU_ENA					да	
CTRL_SOFT_RESET		да	нет	нет	нет	нет	

Продолжение таблицы 1.127

Регистр	Разряд	vdu_off == 1	vbb == 1	mvl_ena == 0	osd_ena == 0	all time	Пояснения
INT_VDU_ENA	I_FIFO_EMPTY_ENA					да	
	O_FIFO_EMPTY_ENA					да	
	H_SYNC_ENA					да	
	V_SYNC_ENA					да	
	FR_PRC_END_ENA					да	
	MVL_FR_RD_END_ENA					да	
	OSD_FR_RD_END_ENA					да	
	SA_ENA					да	
	VDU_OFF_ENA					да	
	INT_SYS_ERR_ENA					да	
INT_VDU_STAT	I_FIFO_EMPTY					да	запись в регистры состояния прерываний производится для снятия соответствующего прерывания
	O_FIFO_EMPTY					да	
	H_SYNC					да	
	V_SYNC					да	
	FR_PRC_END					да	
	MVL_FR_RD_END					да	
	OSD_FR_RD_END					да	
	SA					да	
	VDU_OFF					да	
	INT_SYS_ERR					да	
SCALER_CTRL	FLT_ENA	да	нет	да	нет	нет	
	SCALER_ENA	да	нет	да	нет	нет	
SCALER _SCH_Y	SC_INT_HY	да	нет	да	нет	нет	
	SC_FRAC_HY	да	нет	да	нет	нет	
SCALER _SCV_Y	SC_INT_VY	да	нет	да	нет	нет	
	SC_FRAC_VY	да	нет	да	нет	нет	
SCALER _SCH_C	SC_INT_HC	да	нет	да	нет	нет	
	SC_FRAC_HC	да	нет	да	нет	нет	
SCALER _SCV_C	SC_INT_HC	да	нет	да	нет	нет	
	SC_FRAC_HC	да	нет	да	нет	нет	
SCALER _SIZE_Y	SIZE_VY	да	нет	да	нет	нет	
	SIZE_HY	да	нет	да	нет	нет	
SCALER _SIZE_C	SIZE_VC	да	нет	да	нет	нет	
	SIZE_HC	да	нет	да	нет	нет	

Продолжение таблицы 1.128

Регистр	Разряд	vdu_off == 1	vbb == 1	mvl_ena == 0	osd_ena == 0	all time	Пояснения
SCALER_FLT_Y_C0	FLT_C0_VY	да	нет	да	нет	нет	
	FLT_C0_HY	да	нет	да	нет	нет	
SCALER_FLT_Y_C1	FLT_C1_VY	да	нет	да	нет	нет	
	FLT_C1_HY	да	нет	да	нет	нет	
SCALER_FLT_Y_C2	FLT_C2_VY	да	нет	да	нет	нет	
	FLT_C2_HY	да	нет	да	нет	нет	
SCALER_FLT_Y_C3	FLT_C3_VY	да	нет	да	нет	нет	
	FLT_C3_HY	да	нет	да	нет	нет	
SCALER_FLT_Y_C4	FLT_C4_VY	да	нет	да	нет	нет	
	FLT_C4_HY	да	нет	да	нет	нет	
SCALER_FLT_C_C0	FLT_C0_VC	да	нет	да	нет	нет	
	FLT_C0_HC	да	нет	да	нет	нет	
SCALER_FLT_C_C1	FLT_C1_VC	да	нет	да	нет	нет	
	FLT_C1_HC	да	нет	да	нет	нет	
SCALER_FLT_C_C2	FLT_C2_VC	да	нет	да	нет	нет	
	FLT_C2_HC	да	нет	да	нет	нет	
SCALER_FLT_C_C3	FLT_C3_VC	да	нет	да	нет	нет	
	FLT_C3_HC	да	нет	да	нет	нет	
SCALER_FLT_C_C4	FLT_C4_VC	да	нет	да	нет	нет	
	FLT_C4_HC	да	нет	да	нет	нет	
SCALER_SIZE_CUT	SIZE_CUT_V	да	нет	да	нет	нет	
	SIZE_CUT_H	да	нет	да	нет	нет	
SCALER_PHS_CUT	PHASE_CUT_V	да	нет	да	нет	нет	
	PHASE_CUT_H	да	нет	да	нет	нет	
SCALER_MVL_CLR_Y	MVL_CLR_Y_Cr	да	нет	да	нет	нет	
	MVL_CLR_Y_Cb	да	нет	да	нет	нет	
	MVL_CLR_Y_Y	да	нет	да	нет	нет	
SCALER_MVL_CLR_Cb	MVL_CLR_Cb_Cr	да	нет	да	нет	нет	
	MVL_CLR_Cb_Cb	да	нет	да	нет	нет	
	MVL_CLR_Cb_Y	да	нет	да	нет	нет	
SCALER_MVL_CLR_Cr	MVL_CLR_Cr_Cr	да	нет	да	нет	нет	
	MVL_CLR_Cr_Cb	да	нет	да	нет	нет	
	MVL_CLR_Cr_Y	да	нет	да	нет	нет	

Продолжение таблицы 1.129

Регистр	Разряд	vdu_off == 1	vbb == 1	mvl_ena == 0	osd_ena == 0	all time	Пояснения
DIF_CTRL	VSYNC_P	да	нет	нет	нет	нет	
	HSYNC_P	да	нет	нет	нет	нет	
	SYNC_SEP	да	нет	нет	нет	нет	
	MODE	да	нет	нет	нет	нет	
	EXT_SYBC_EN	да	нет	нет	нет	нет	
	INT_SYNC_EN	да	нет	нет	нет	нет	
DIF_BGR	BGR_Y	да	да	нет	нет	нет	
	BGR_Cb	да	да	нет	нет	нет	
	BGR_Cr	да	да	нет	нет	нет	
DIF_MVL_START	MVL_START_V	да	да	да	нет	нет	
	MVL_START_H	да	да	да	нет	нет	
DIF_BLANK	HBLANK	да	нет	нет	нет	нет	
	VBLANK_BEG	да	нет	нет	нет	нет	
	VBLANK_END	да	нет	нет	нет	нет	
DIF_ASIZE	VACTIVE	да	нет	нет	нет	нет	
	HACTIVE	да	нет	нет	нет	нет	
DIF_HSYNC	HSYNC_LEN	да	нет	нет	нет	нет	
	HSYNC_START	да	нет	нет	нет	нет	
	HSYNC_DELAY	да	нет	нет	нет	нет	
DIF_VSYNC	VSYNC_LEN	да	нет	нет	нет	нет	
	VSYNC_START	да	нет	нет	нет	нет	
OSD_BASE	OSD_BASE					да	по окончанию чтения из памяти текущей страницы с графическими данными osd_fr_rd_end
OSD_COLOR_Y	OSD_COLOR_Y_B	да	нет	нет	да	нет	
	OSD_COLOR_Y_G	да	нет	нет	да	нет	
	OSD_COLOR_Y_R	да	нет	нет	да	нет	
OSD_COLOR_Cb	OSD_COLOR_Y_B	да	нет	нет	да	нет	
	OSD_COLOR_Y_G	да	нет	нет	да	нет	
	OSD_COLOR_Y_R	да	нет	нет	да	нет	
OSD_COLOR_Cr	OSD_COLOR_Y_B	да	нет	нет	да	нет	
	OSD_COLOR_Y_G	да	нет	нет	да	нет	
	OSD_COLOR_Y_R	да	нет	нет	да	нет	
OSD_FIFO_END	OSD_FIFO_END	да	нет	нет	да	нет	

Продолжение таблицы 1.130

Регистр	Разряд	vdu_off == 1	vbb == 1	mvl_ena == 0	osd_ena == 0	all time	Пояснения
MI_CTRL	BUF_BIT_ENDIAN	да	нет	да	нет	нет	
	YCbCr_FORMAT	да	нет	да	нет	нет	
	PLANE_NUM	да	нет	да	нет	нет	
	PLANE_ACCESS	да	нет	да	нет	нет	
	FILD_ACCESS	да	нет	да	нет	нет	
	FILD_STORE	да	нет	да	нет	нет	
MI_MVL_Y_BA	MI_MVL_Y_BA					да	по окончанию чтения из памяти текущей страницы с видео данными mvl_fr_rd_end
MI_MVL_Cb_BA	MI_MVL_Cb_BA					да	
MI_MVL_Cr_BA	MI_MVL_Cr_BA					да	
MI_MVL_Y_SIZE	MI_MVL_Y_V_SIZE	да	нет	да	нет	нет	
	MI_MVL_Y_H_SIZE	да	нет	да	нет	нет	
MI_MVL_C_SIZE	MI_MVL_C_V_SIZE	да	нет	да	нет	нет	
	MI_MVL_C_H_SIZE	да	нет	да	нет	нет	
MI_MVL_FULL_SIZE	MVL_C_FULL_SIZE	да	нет	да	нет	нет	
	MVL_Y_FULL_SIZE	да	нет	да	нет	нет	
MI_AXI_PARAM	ARLEN[3:0]	да	нет	нет	нет	нет	
	ARSIZE[2:0]	да	нет	нет	нет	нет	
	ARBURST[1:0]	да	нет	нет	нет	нет	
	ARLOCK[1:0]	да	нет	нет	нет	нет	
	ARCACHE[3:0]	да	нет	нет	нет	нет	
	ARPROT[2:0]	да	нет	нет	нет	нет	
MI_MVL_FIFO_START	MVL_Cr_FIFO_START	да	нет	нет	нет	нет	
	MVL_Cb_FIFO_START	да	нет	нет	нет	нет	
	MVL_Y_FIFO_START	да	нет	нет	нет	нет	
MI_MVL_FIFO_END	MVL_FIFO_END	да	нет	нет	нет	нет	

1.3.5.1.7.2.2 Отображение фона

Настройка отображения фона сводится к установке его цвета. Цвет устанавливается в формате YCbCr соответствующего качества (SDTV или HDTV) путем записи в регистр DIF_BGR значения для Y в разряд BGR_Y, для значения Cb в разряд BGR_Cb, для значения Cr в разряд BGR_Cr. Если в результирующем изображении требуется только наличие фона, то значение разрядов OSD_ENA и MVL_ENA регистра CTRL_VDU_ENA должно быть ноль.

1.3.5.1.7.2.3 Задание параметров выходного видеоизображения

Параметры выходного видеоизображения зависят от формата выходного видеоизображения, поэтому их следует выбирать в соответствии с таблицей 2.2 (задание других значений параметров может приводить к непредсказуемой работе устройства).

Таблица 1.131 - Параметры и их значения, необходимые для построения временной диаграммы выходного видеоизображения заданного формата

Mode	Frame Hz	Sdandard	PixClkM Hz	H-blank	H-active	H-Total	HS_start	HS_len	HS_delay	V-Blank beg	V-Active	V-Total	V-Blank end	VS_start	VS_len	VS_P
720x486i 720x576i	30*2 25*2	BT656 BT656	27 27	134 140	720 720	858 864	19 12	62 63	57 69	19 22	486 576	525 625	0 2	3 0	3 3	0 0
720(1440)x480i 720(1440)x576i	30*2 25*2	770.2 BT656	27 27	268 280	1440 1440	1716 1728	38 24	124 126	114 138	21 22	480 576	525 625	1 2	3 0	3 3	0 0
720x480p 720x576p	60 50	770.2 BT1358	27 27	130 136	720 720	858 864	16 12	62 64	60 68	42 44	480 576	525 625	3 5	6 0	6 5	0 0
1280x720p 1280x720p	60 50	770.3 296M	74,25 74,25	362 692	1280 1280	1650 1980	110 440	40 40	220 220	25 25	720 720	750 750	5 5	0 0	5 5	1 1
1920x1080i 1920x1080i	30*2 25*2	770.3 274M	74,25 74,25	272 712	1920 1920	2200 2640	88 528	44 44	148 148	20 20	1080 1080	1125 1125	2 2	0 0	5 5	1 1
1920x1080p 1920x1080p	30 25	274M 274M	74,25 74,25	272 712	1920 1920	2200 2640	88 528	44 44	148 148	41 41	1080 1080	1125 1125	4 4	0 0	5 5	1 1
1920x1080p 1920x1080p	60 50	274M 274M	148,5 148,5	272 712	1920 1920	2200 2640	88 528	44 44	148 148	41 41	1080 1080	1125 1125	4 4	0 0	5 5	1 1

Соответствующие параметры записываются в регистр управления цифровых интерфейсов DIF_CTRL, регистр координат основного видеослоя на экране DIF_MVL_START, регистр размера интервала межстрочной развертки DIF_BLANK, регистр размера активной части видеоизображения DIF_ASIZE, регистр позиции сигнала горизонтальной синхронизации DIF_HSYNC, регистр позиции сигнала вертикальной синхронизации DIF_VSYNC. Конфигурирование режима выводимого изображения необходимо осуществлять при выключенном устройстве.

1.3.5.1.7.2.4 Тракт обработки данных видеоконтроллера

При создании выходного видеоизображения видеоконтроллер производит обработку читаемых из внешней памяти видео и графических данных в соответствии со схемой представленной на рисунке Рисунок 1.76.

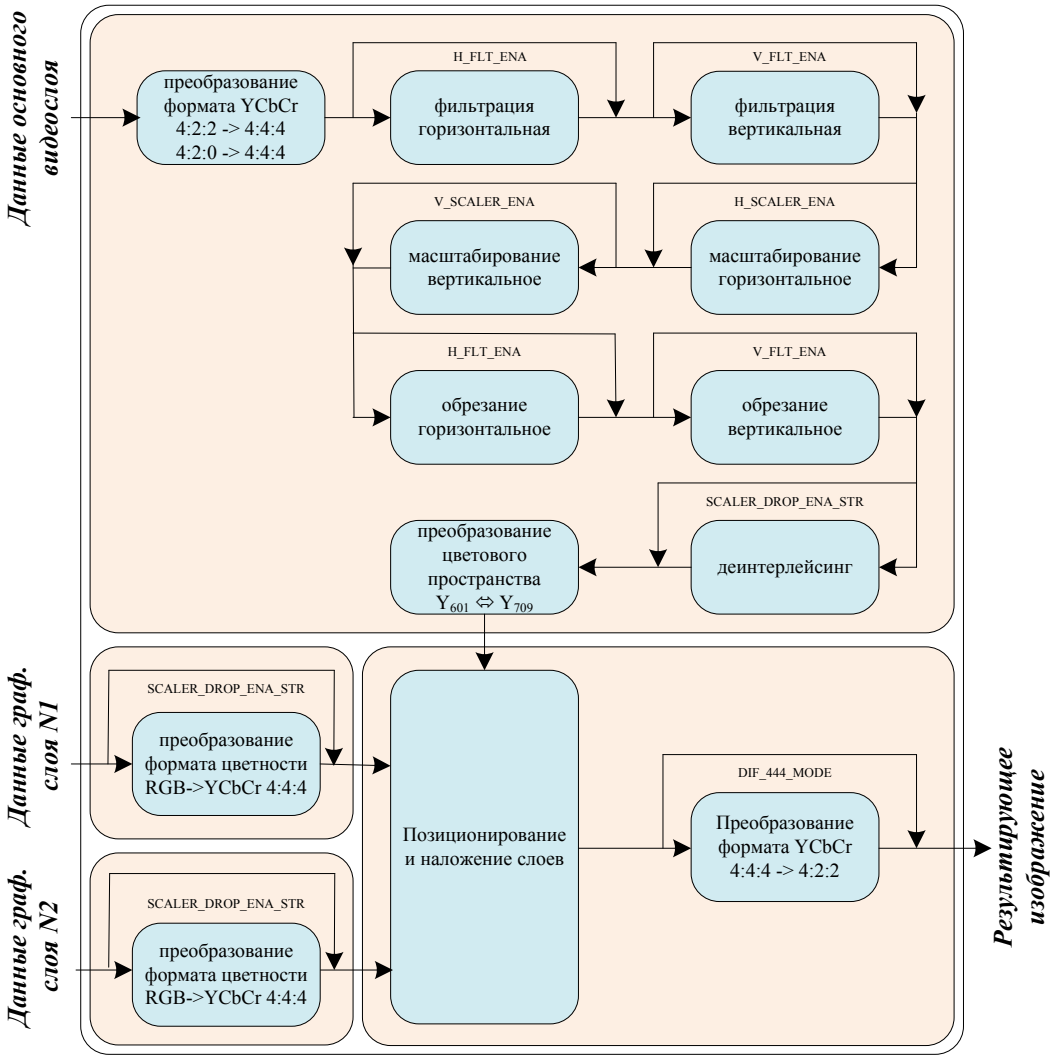


Рисунок 1.76 - Тракт обработки данных видеоконтроллера

Подробное описание настройки видеоконтроллера для выбора пути прохождения данных по тракту видеоконтроллера, а также управления основными функциями преобразования данных, приведено в последующих пунктах документации.

1.3.5.1.7.2.5 Отображение графического слоя

Графический слой представляет собой набор графических областей. Каждая графическая область представляет собой прямоугольник, имеющий границы и содержащий набор пикселей, определяющих его содержание. Графические данные располагаются поверх области активного видео с использованием коэффициента прозрачности, определяющего прозрачность графических данных.

Ограничения, накладываемые на данный слой:

Графические области одного слоя не должны перекрываться по горизонтали и только одна область может быть определена для строки. Устройство управления и контроллер прямого доступа видеоконтроллера должны гарантировать то, что заголовок области будет вовремя считан и декодирован, для обращения к следующей области.

Графические области считываются из системной памяти, каждая графическая область содержит:

заголовок графической области;

графические данные.

При хранении в памяти все графические данные хранятся в little endian формате. Форматы входных данных графической области описаны в разделе 1.4.4 Протокол формирования изображения и форматы данных графического слоя данного руководства по эксплуатации.

1.3.5.1.7.2.5.1 Чтение из памяти

Перед началом чтения графической области из памяти необходимо сначала сконфигурировать контроллер на соответствующий режим. Для этого необходимо записать в регистры OSD_COLOR_Y, OSD_COLOR_Cb, OSD_COLOR_Cr соответствующие значения коэффициентов преобразования для различных режимов качества (SDTV или HDTV). И записать в регистр OSD_BASE1 адрес заголовка первой графической области (первого связанного списка), а в регистр OSD_BASE0 адрес заголовка второй графической области (второго связанного списка).

После этого в разряд разрешения переключения страниц OSD_BASE_SW_ENA регистра управления видеоконтроллера VDU_CONTROL необходимо записать единицу, а также разрешить отображение графического слоя, записав единицу в разряд OSD_ENA регистра управления видеоконтроллера VDU_CONTROL.

Если же изначально готовы данные только одного кадра (полукадра), то следует записать адрес заголовка первой графической области (первого связанного списка) в регистр OSD_BASE1, и включить OSD_BASE_SW_ENA в единицу, после того как появится прерывание окончания считывания активного кадра для графического слоя из памяти OSD_FR_RD_END (появится, если разрешена маска соответствующего прерывания разряд OSD_FR_RD_END_ENA в регистре INT_VDU_ENA) необходимо записать ноль.

Данные второго кадра должны быть подготовлены и адрес следующей графической области должен быть записан в регистр OSD_BASE0 до появления прерывания импульс вертикальной синхронизации V_SYNC (появится, если разрешена маска соответствующего прерывания разряд VSYNC_ENA в регистре INT_VDU_ENA), после появления которого, нужно записать в разряд OSD_BASE_SW_ENA единицу.

Далее рассматриваются различные случаи чередования кадров (случаи рассматриваются для изначально выключенного контроллера):

Первый кадр выводится, а второй не выводится.

в регистр OSD_BASE записывается адрес выводимой графической области (первого связанного списка)

в разряд разрешения переключения страниц OSD_BASE_SW_ENA регистра управления видеоконтроллера VDU_CONTROL записывается единица

в разряд OSD_ENA регистра управления видеоконтроллера VDU_CONTROL записывается единица

установить единицу в разряде VDU_ENA в управляющем регистре видеоконтроллера VDU_CONTROL

по прерыванию окончания считывания активного кадра для графического слоя из памяти OSD_FR_RD_END (появится, если разрешена маска соответствующего прерывания разряд OSD_FR_RD_END_ENA в регистре INT_VDU_ENA) необходимо записать ноль в разряд разрешения переключения страниц OSD_BASE_SW_ENA регистра управления видеоконтроллера VDU_CONTROL

по прерыванию окончания обработки активного кадра FR_PRC_END (появится, если разрешена маска соответствующего прерывания разряд FR_PRC_END_ENA в регистре INT_VDU_ENA) необходимо записать ноль в разряд OSD_ENA регистра управления видеоконтроллера VDU_CONTROL

для следующих кадров необходимо выбрать действия из соответствующих вариантов

Первый кадр не выводится, а второй выводится.

в регистр OSD_BASE записывается адрес выводимой графической области (первого связанного списка), если данные области готовы

в разряд разрешения переключения страниц OSD_BASE_SW_ENA регистра управления видеоконтроллера VDU_CONTROL записывается ноль

в разряд OSD_ENA регистра управления видеоконтроллера VDU_CONTROL записывается ноль

установить единицу в разряде VDU_ENA в управляющем регистре видеоконтроллера VDU_CONTROL

по прерыванию окончания обработки активного кадра FR_PRC_END (появится, если разрешена маска соответствующего прерывания разряд FR_PRC_END_ENA в регистре INT_VDU_ENA) необходимо записать единицу в разряд OSD_ENA регистра управления видеоконтроллера VDU_CONTROL

по прерыванию импульс вертикальной синхронизации V_SYNC (появится, если разрешена маска соответствующего прерывания разряд VSYNC_ENA в регистре INT_VDU_ENA) записать единицу в разряд разрешения переключения страниц OSD_BASE_SW_ENA регистра управления видеоконтроллера VDU_CONTROL

для следующих кадров необходимо выбрать действия из соответствующих вариантов

Первый кадр не выводится и второй не выводится.

в разряд OSD_ENA регистра управления видеоконтроллера VDU_CONTROL записывается ноль

установить единицу в разряде VDU_ENA в управляющем регистре видеоконтроллера VDU_CONTROL

для следующих кадров необходимо выбрать действия из соответствующих вариантов

Первый кадр выводится и второй выводится.

в регистр OSD_BASE записывается адрес выводимой графической области (связанного списка)

в разряд разрешения переключения страниц OSD_BASE_SW_ENA регистра управления видеоконтроллера VDU_CONTROL записывается единица

в разряд OSD_ENA регистра управления видеоконтроллера VDU_CONTROL записывается единица

установить единицу в разряде VDU_ENA в управляющем регистре видеоконтроллера VDU_CONTROL

для следующих кадров необходимо выбрать действия из соответствующих вариантов

1.3.5.1.7.2.5.2 Колориметрические преобразования

В канале графического слоя осуществляется преобразование значений цветовых характеристик (colorimetric characteristics) αRGB (форматы представления графических данных 5:6:5; 1:5:5:5 и 4:4:4:4) в YCbCr 709 4.2.2(HD) высокого качества или преобразование αRGB (форматы представления графических данных 5:6:5; 1:5:5:5 и 4:4:4:4) в YCbCr 601 4.2.2(SD) стандартного качества.

Для задания соответствующего цветового преобразования необходимо записать в регистры OSD_COLOR_Y, OSD_COLOR_Cb, OSD_COLOR_Cr соответствующие значения коэффициентов преобразования для различных режимов качества (SDTV или HDTV).

Для высокого качества 709 4.2.2 (HD) следующие значения коэффициентов

OSD_COLOR_Y_R 0x 2f
OSD_COLOR_Y_G 0x 9d
OSD_COLOR_Y_B 0x 10
OSD_COLOR_Cb_R 0x 11a
OSD_COLOR_Cb_G 0x 157
OSD_COLOR_Cb_B 0x 70
OSD_COLOR_Cr_R 0x 70
OSD_COLOR_Cr_G 0x 166
OSD_COLOR_Cr_B 0x 10a

Для стандартного качества 601 4.2.2 (SD) следующие значения коэффициентов

OSD_COLOR_Y_R 0x 42
OSD_COLOR_Y_G 0x 81
OSD_COLOR_Y_B 0x 19
OSD_COLOR_Cb_R 0x 126
OSD_COLOR_Cb_G 0x 14a
OSD_COLOR_Cb_B 0x 70
OSD_COLOR_Cr_R 0x 70
OSD_COLOR_Cr_G 0x 15e
OSD_COLOR_Cr_B 0x 112

1.3.5.1.7.2.5.3 Наложение графических областей на основной видеослой или фон

На вход блока наложения и позиционирования слоев поступают данные основного видеослоя, а также данные двух графических слоев, каждый из которых состоит из набора графических областей, не перекрывающихся по горизонтали. Внутренний формат представления видео и графических данных – YCbCr4:4:4. Размеры основного видеослоя и графических областей, а также начальная точка их позиционирования в результирующем выходном изображении задается путем программирования регистров конфигурации видеоконтроллера (для видеослоя) и заданием соответствующих параметров в заголовках графических областей, хранящихся во внешней памяти.

Видеоконтроллер может применять две схемы наложения слоев друг на друга в зависимости от значения бита управления DIF_CH_SEL:

в первом случае (см. рисунок 1.77) производится непрозрачное наложение основного видеослоя слоя на фон экрана, затем на полученное изображение накладывается с прозрачностью графический слой 1, затем на полученное значение непрозрачно накладывается графический слой 2.

во втором случае (см. рисунок 1.78) производится наложение графического слоя 1 с прозрачностью на фон, параллельно с этим производится наложение с прозрачностью графического слоя 2 на основной видеослой. Впоследствии результат второго наложения непрозрачно накладывается на результат первого наложения.

Основной видеослой всегда непрозрачен, графические области могут накладываться с коэффициентом прозрачности. Формат прозрачности каждой графический области задается в ее заголовке и может принимать два основных значения: заданный единый коэффициент прозрачности применяется ко всем пикселям графической области или каждый пиксель

графической области, прочитанный из внешней памяти, может нести в себе свое индивидуальное значение коэффициента прозрачности (см. раздел 2.6).

Наложение точек с прозрачностью осуществляется по формуле:

mix_Y = (α*osd_Y + (255-α)*mvl_Y)/ 255 – для яркостной Y составляющей пикселя,

аналогично для составляющих Cb,Cr.

α = 0..255 – коэфф. непрозрачности, точка с коэффициентом 0 – является полностью прозрачной, а с коэффициентом 255 – полностью непрозрачной.

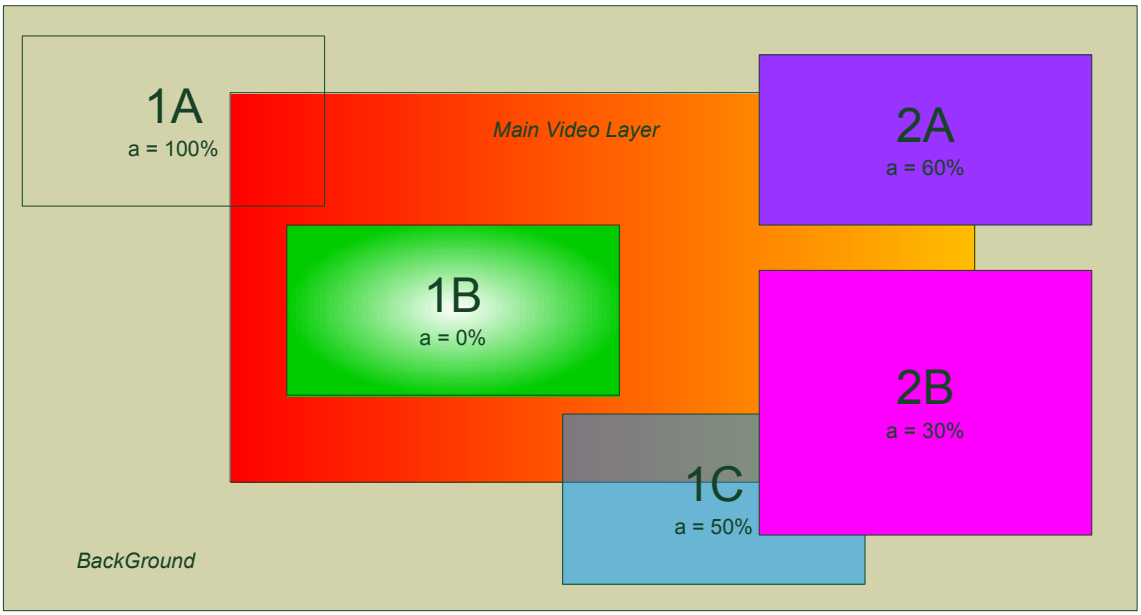


Рисунок 1.77 - Наложения графических областей с разными коэффициентами прозрачности на основной видеослой и фон при значении бита управления DIF_CH_SEL равном 0

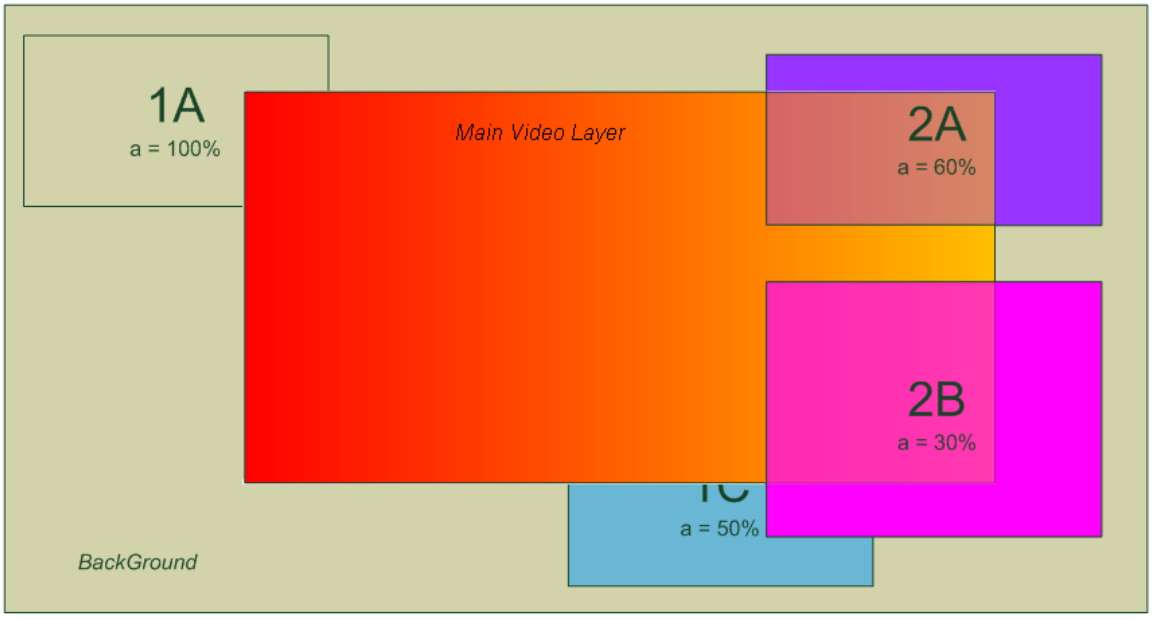


Рисунок 1.78 - Наложения графических областей с разным коэффициентами прозрачности на основной видеослой и фон при значении бита управления DIF_CH_SEL равном 1

Полученное в результате наложения и позиционирования слоев изображение представлено во внутреннем формате YCbCr 4:4:4. В случае необходимости вывода изображения в формате 4:2:2, необходимо активировать процесс преобразования, установив бит DIF_444_MODE конфигурационных регистров видеоконтроллера в ноль. Суть преобразования заключается в уменьшении цветностной информации видеоизображения, а именно присвоению каждой паре пикселов одного и того же значения цветности. Видеоконтроллер реализует два способа преобразования формата 4:4:4 в 4:2:2:

каждой паре пикселов присваивается значение цветности первого пиксела из пары.

каждой паре пикселов присваивается значение цветности равное усредненному значению цветностей каждого пиксела из пары.

Выбор способа преобразования осуществляется программированием бита AVR_MODE конфигурационных регистров видеоконтроллера.

Необходимо отметить, что при выборе второго способа преобразовании формата данных может происходить потеря качества изображения. Так, например, если граница двух вертикальных полос различного цвета попадает на пару усреднения, то цветовой переход получится не точным (см. рисунок 1.79). В то же время, большинство картинок не имеют резких цветовых переходов, а потому данная особенность не будет проявляться. Программистам всё же следует учесть данное положение и при необходимости задавать четные координаты и размеры областей.

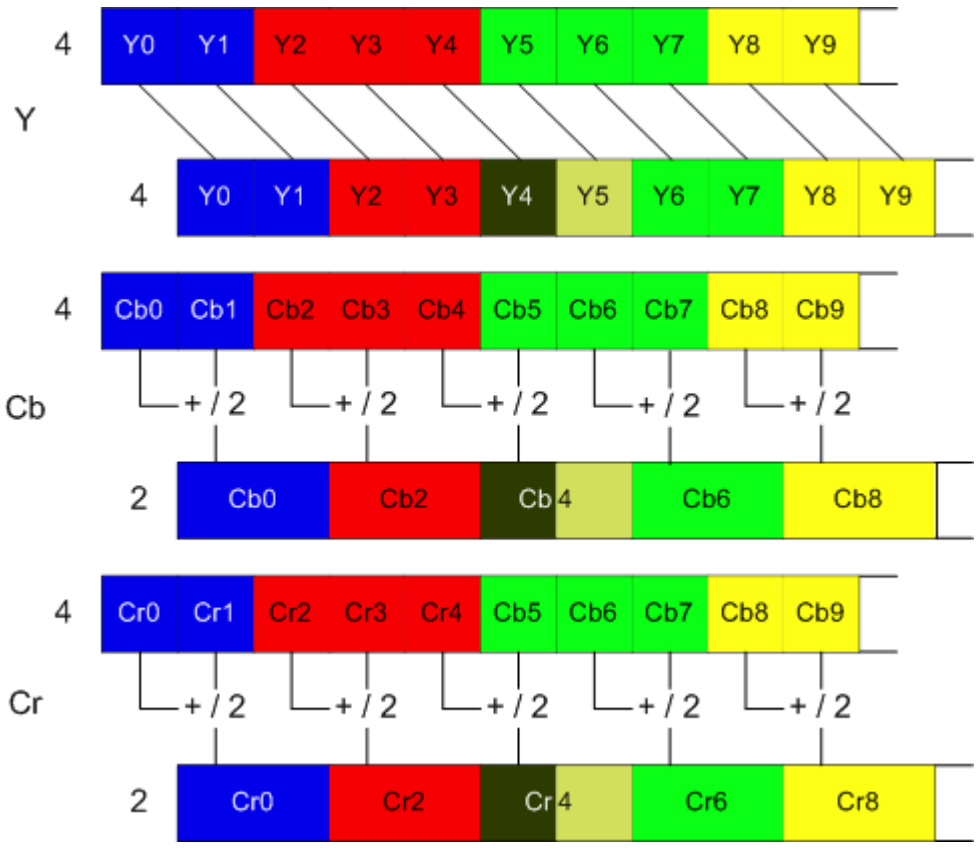


Рисунок 1.79 - Преобразование YCbCr 4:4:4 –> 4:2:2

Если бит управления видеоконтроллера OSD_RGB_YCBCR_BYPASS установлен в единицу, на вход блока наложения и позиционирования слоев данные графических слоев будут поступать в формате RGB. В таком случае наложение графических слоев друг на друга и на фон будут производиться в формате RGB (значение цвета фона также должно задаваться в формате RGB). Описанные выше способы и формулы наложения остаются в силе. Поскольку основной

видеослой имеет постоянный формат YCbCr, то наложение с ним невозможно, и его отображение должно быть отключено (MVL_EN=0). Бит управления DIF_444_MODE должен быть установлен в значение ‘1’. Выходное результирующее изображение будет представлено в формате RGB.

1.3.5.1.7.2.6 Отображение основного видеослоя

Основной видеослой видеоконтроллера предназначен для отображения видеоизображений.

Видеоконтроллер выводит видеоданные, хранимые во внешней памяти, на один из своих цифровых видеоинтерфейсов: стандартной или высокой четкости. Для основного видеослоя видеоконтроллера поддерживаются следующие форматы представления входных данных:

YCbCr 4:2:0

YCbCr 4:2:2

Видеоданные видеопотока, хранимые во внешней памяти, разделены на яркостную и цветоразностную плоскости (плоскости компонент), которые хранятся в видеопамяти независимо. При этом порядок данных внутри плоскости может быть различным.

Форматы входных данных для основного видеослоя подробно описаны в разделе 1.4.3 Протокол формирования изображения и форматы данных основного видеослоя данного руководства по эксплуатации.

1.3.5.1.7.2.6.1 Чтение видеопотока из памяти

Перед началом чтения данных из памяти необходимо сконфигурировать контроллер в требуемый режим работы.

Для этого требуется записать в регистры MVL_COLOR_Y (Cb и Cr) соответствующие коэффициенты преобразования для различных режимов качества (SDTV или HDTV). Информацию о формате входных данных хранимых в памяти необходимо записать в регистр управления интерфейса с памятью (MI_CTRL), это информация о размере пакета на системной шине и т.д. Следующим шагом будет задание базовых адресов, выводимых данных в регистре базовых адресов доступной области для компонент (MI_MVL_Y_BAreg0, MI_MVL_Cb_BAreg0, MI_MVL_Cr_BAreg0, MI_MVL_Y_BAreg1, MI_MVL_Cb_BAreg1, MI_MVL_Cr_BAreg1).

Затем задается размер считываемой области в памяти для компонент в регистрах размера доступной области плоскости для яркостной и цветоразностных компонент (MI_MVL_Y_SIZE, MI_MVL_C_SIZE).

После этого записывается полный размер изображения в памяти (может быть больше размера считываемой области) в регистре полного (физического) размера строки плоскости компоненты (MI_MVL_FULL_SIZE).

В том случае, если нет необходимости производить преобразование размеров изображения в регистры размеров изображения после масштабирования и регистры размеров изображения после обрезания должны быть установлены размеры, соответствующие размерам (MI_MVL_Y_SIZE, MI_MVL_C_SIZE). Также признаки, разрешающие преобразование изображения, должны быть обнулены.

В регистре разрешения видеоконтроллера необходимо установить разряд регистра разрешения отображения основного слоя в единицу. Затем необходимо установить разряд MVL_BASE_SW_ENA в единицу в регистре CTRL_VDU_ENA и запустить работу видеоконтроллера VDU_ENA.

Смена базового адреса следующего кадра (полукадра) происходит по срабатыванию прерывания MVL_FR_RD_END (если оно было разрешено). Если данные основного видеослоя не готовы, а контроллер работает, например, происходит отображение графического слоя или фона, то, когда данные основного слоя будут подготовлены, то при возникновении прерывания V_SYNC (если оно было разрешено) нужно записать в MVL_BASE_SW_ENA единицу.

1.3.5.1.7.2.6.2 Колориметрические преобразования

В канале основного слоя осуществляется преобразование значений цветовых характеристик из стандартного качества YCbCr 601 4.2.2(SD) в высокое качество YCbCr 709 4.2.2(HD) или из высокого качества YCbCr 709 4.2.2(HD) в стандартное качество YCbCr 601 4.2.2(SD) или пропускание без преобразования качества.

Для задания соответствующего преобразования качества необходимо записать в регистры OSD_COLOR_Y, OSD_COLOR_Cb, OSD_COLOR_Cr, соответствующие значения коэффициентов преобразования.

Из высокого качества в стандартное

MVL_CLR_Y_Y	0x24a
MVL_CLR_Y_CB	0x19
MVL_CLR_Y_CR	0x31
MVL_CLR_CB_Y	0x1f
MVL_CLR_CB_CB	0xfd
MVL_CLR_CB_CR	0x21c
MVL_CLR_CR_Y	0x17
MVL_CLR_CR_CB	0x213
MVL_CLR_CR_CR	0xfc

Из стандартного качества в высокое

MVL_CLR_Y_Y	0x53
MVL_CLR_Y_CB	0x21e
MVL_CLR_Y_CR	0x235
MVL_CLR_CB_Y	0x222
MVL_CLR_CB_CB	0x105
MVL_CLR_CB_CR	0x1d
MVL_CLR_CR_Y	0x21a
MVL_CLR_CR_CB	0x13
MVL_CLR_CR_CR	0x107

Без преобразования

MVL_CLR_Y_Y	0x0
MVL_CLR_Y_CB	0x0
MVL_CLR_Y_CR	0x0
MVL_CLR_CB_Y	0x0
MVL_CLR_CB_CB	0x100
MVL_CLR_CB_CR	0x0
MVL_CLR_CR_Y	0x0
MVL_CLR_CR_CB	0x0
MVL_CLR_CR_CR	0x100

1.3.5.1.7.2.6.3 Масштабирование видеоизображений

Видеоконтроллер обеспечивает функцию масштабирования изображения, как по горизонтали, так и по вертикали для реализации следующих режимов работы видеоконтроллера, обусловленных техническим заданием.

Воспроизведение видео с разрешениями меньше SD (но не менее CIF) при выходном разрешении SD.

Без коррекции.

С масштабированием в SD с независимыми коэффициентами по X и Y, и преобразованием к 4:3 (обрезанием границ).

Воспроизведение HD видео с уменьшением до SD и преобразованием к 4:3 (обрезанием границ). Преобразование развертки входного изображения не производится, т.е. тип и частота развертки входного и выходного изображений должны совпадать.

При уменьшении размеров изображения с HD до SD с целью улучшения качества изображения и устранения возможных артефактов (aliasing) в состав видеоконтроллера введен усредняющий (antialiasing) фильтр. При этом вертикальная фильтрация изображения допустима только в том случае, если горизонтальный размер изображения после масштабирования (параметр SCALER _SIZE_Y(C)_H) меньше, либо равен 1024 пикселам – это обусловлено размерами буферов, используемых для вертикальной фильтрации.

Для осуществления операции масштабирования пользователь должен задать значения в регистрах масштабирования (VDU Scaler Registers). Эти значения должны быть заданы при отключенном видеоконтроллере или основном слое видеоконтроллера.

Так в регистре управления устройством масштабирования (SCALER_CTRL) должны быть установлены разрешающие сигналы, определяющие набор операций, которые необходимо выполнить с изображением. Возможный набор операций:

- вертикальная фильтрация;
- горизонтальная фильтрация;
- обрезание изображения по вертикали;
- обрезание изображения по горизонтали;
- вертикальное масштабирование;
- горизонтальное масштабирование.

Также необходимо задать коэффициенты масштабирования, как по вертикали, так и по горизонтали, при этом коэффициенты задаются отдельно для яркостной и цветоразностной компонент (регистры: SCALER_SCH_Y, SCALER_SCV_Y, SCALER _SCH_C, SCALER_SCV_C).

Коэффициент задается в виде шага масштабирования, определяемого по формуле ((кол-во пикселов в строке до масштабирования – 1) / (кол-во пикселов в строке после масштабирования – 1)) Коэффициенты представлены в виде целой и дробной частей, обе в виде беззнакового целого. Дробная часть коэффициента получена умножением исходной вещественной дробной части на 65536 (берется целая часть – дробная отбрасывается).

При увеличении – целая часть коэффициента является нулевой. При уменьшении коэффициент масштабирования больше 1.

Согласно сценариям применения видеоконтроллера увеличение изображения производится от значений изображения с разрешениями меньше SD (но не менее CIF) до изображения с SD разрешением. А уменьшение изображения от изображений с разрешением HD до изображения с SD.

Под целую часть коэффициента отводится 4 разряда, под дробную – 16 разрядов.

Помимо этого, необходимо задать размеры изображения, которые будут после операции масштабирования (в том случае, если масштабирование не производится данные размеры должны совпадать с размерами изображения в памяти). Регистры: SCALER _SIZE_Y, SCALER _SIZE_C

При этом все размеры задаются на единицу меньше физических, т.е. для того, чтобы получить реальное количество пикселов или строк изображения необходимо к указанным в регистрах размерам добавить единицу. Также действует ограничение – количество пикселов и количество строк должно быть четным значением.

При необходимости операции фильтрации в регистры: SCALER_FLT_Y_C0, SCALER_FLT_Y_C1, SCALER_FLT_Y_C2, SCALER_FLT_C_C0, SCALER_FLT_C_C1, SCALER_FLT_C_C2 должны быть прописаны коэффициенты фильтра.

Усредняющая фильтрация применяется при уменьшении размеров изображения для устранения причин возникновения дефектов изображения, вызванных отбросом незначащих пикселов и строк изображения. Устраняет дефект изображения, именуемый термином – aliasing.

Коэффициенты представлены в виде 8 разрядных целых беззнаковых чисел. Данные коэффициенты получены в результате умножения исходных (вещественных) коэффициентов на коэффициент нормализации. Сумма всех коэффициентов должна быть меньше или равняться коэффициенту нормализации. Коэффициенты нормализации представлены числами, являющимися степенями 2. Так для 8ми разрядных коэффициентов фильтра коэффициент нормализации равен 256.

Пример: единичный фильтр, не вносящий искажений – фильтр имеет следующие коэффициенты 1, 254, 1. При увеличении граничных коэффициентов и уменьшении среднего размывающая способность фильтра возрастает.

1.3.5.1.7.2.6.4 Обрезание видеоизображений

Обрезание изображения силами видеоконтроллера ведется только в случае его масштабирования, или при отрезании небольшого количества строк/пикселов, соизмеримых с интервалами межстрочной и межкадровой разверток соответственно.

Горизонтальное обрезание

При горизонтальном обрезании задаются три параметра: количество пикселов, обрезаемое слевой стороны строки изображения, количество пикселов, обрезаемое справой стороны строки (SCALER_PHS_CUT_H) и количество пикселов изображения, которое остается (SCALER_SIZE_CUT).

Вертикальное обрезание

При вертикальном обрезании задаются три параметра: количество строк, обрезаемое сверху изображения, количество строк, обрезаемое снизу (SCALER_PHS_CUT_V) и количество строк изображения, которое остается (SCALER_SIZE_CUT).

При задании размера изображения после масштабирования размеры задаются на единицу меньше физических, т.е. для того, чтобы получить реальное количество пикселов или строк изображения необходимо к указанным в регистрах размерам добавить единицу. Также действует ограничение – количество пикселов и количество строк должно быть четным значением.

При задании размеров обрезаемых областей размеры задаются равными физическим размерам, т.е. реальное количество пикселов/строк изображения соответствует указанным в регистрах размерам (добавлять единицу не нужно). Действует ограничение – количество обрезаемых пикселов/строк должно быть четным значением.

1.3.5.1.7.2.6.5 Позиционирование основного слоя на экране

Пользователем могут быть заданы начальные координаты отображения изображения основного слоя на экране (координаты левого верхнего угла изображения). Данные значения задаются в регистре DIF_MVL_START.

Рекомендуется устанавливать четные значения координат.

1.3.5.1.7.2.6.6 Особенности чтения данных в прогрессивном и чересстрочном режимах

При чтении входных данных в формате YCbCr 420 в чересстрочном режиме, блоками DMA компонент основного слоя осуществляется выборка цветоразностных компонент для каждого полукадра в формате 422.

При чтении входных данных в формате YCbCr 420 в прогрессивном режиме чтение ведется в формате 420, а размножение строк цветоразностных компонент ведется аппаратурой видеоконтроллера внутри.

1.3.5.1.7.2.6.7 Преобразование прогрессивного изображения в чересстрочное

Видеоконтроллер аппаратно поддерживает функцию преобразования прогрессивного изображения в чересстрочное. Данная функция используется тогда, когда на входе видеоконтроллера присутствует прогрессивное изображение при этом выходная развертка установлена как чересстрочная.

Данное преобразование может осуществляться двумя способами.

Непосредственно при чтении данных из видеопамяти. Настройки видеоконтроллера позволяют обеспечить чтение прогрессивного изображения, хранимого в видеопамяти, в чересстрочном режиме. Обеспечивая сначало считывание первого полукадра, а затем второго полукадра. Данный режим допускается применять, когда не требуется преобразование изображения в видеоконтроллере, так как при масштабировании подобного динамического изображения возможно возникновение артефактов на экране отображающего устройства.

Примером применения второго способа может служить следующий сценарий. На вход видеоконтроллера подается видеоизображение прогрессивного типа с размерами большими или меньшими, чем видимая область выводимого изображения. В данном случае необходимо произвести масштабирование входного изображения до размеров видимой области выходного изображения.

Для сохранения качества изображения и предотвращения появления артефактов, масштабирование видеоизображения необходимо вести в прогрессивном режиме с последующим выводом строк определенного полукадра.

Данная процедура осуществляется следующим образом: входные тракты видеоконтроллера и схема масштабирования устанавливаются для работы с изображением в прогрессивном режиме. Для выходных трактов задается требуемый чересстрочный режим отображения изображения. Задаются коэффициенты масштабирования и колориметрического преобразования, обеспечивающие корректное преобразование размеров входного изображения до размеров выходного. Разрешается работа устройства отбрасывания строк – устанавливается в единичное значение признак SCAL_DROP_ENA_STR в регистре SCALER_CTRL. При этом для каждого выводимого полукадра необходимо задавать признак, определяющий принцип его формирования. Признак SCAL_DROP_POLAR_STR в регистре SCALER_CTRL. Определяет полярность схемы dropper’a (интерлейсера) 0 – отбрасываются нечетные строки кадра (выводимый кадр содержит только четные строки исходного кадра), 1 – отбрасываются четные строки кадра (выводимый кадр содержит только нечетные строки исходного кадра).

1.3.5.2 Устройство захвата изображения

1.3.5.2.1 Общее описание

1.3.5.2.1.1 Описание назначения

Устройство захвата видеоизображения (УЗВИ) рассчитано на применение в составе СнК с системной шиной AMBA 3.0 (AXI) высокой степени интеграции.

Устройство видеозахвата предназначено для захвата активной части видеоизображения из телевизионной развёртки стандартной и высокой чёткости, предварительной обработки видеоизображения и записи его во внутреннюю память системы. Устройство видеозахвата работает с видеоизображением прогрессивного и чересстрочного формата. Устройство видеозахвата поддерживает функции гама-коррекции изображения и преобразования цветового пространства.

1.3.5.2.1.2 Технические характеристики

Технические характеристики УЗВИ делятся на две группы:

- Технологически не зависимые характеристики
- Технологически зависимые характеристики

К первой группе относятся следующие характеристики устройства:

- поддерживаемые форматы хранения видеоданных:
  - линейный, с двух плоскостным и трех плоскостным хранением компонент (для данных формата YCbCr),
  - линейный, с одно плоскостным и трех плоскостным хранением компонент (для данных формата RGB);
- формат захватываемых видеоданных:
  - YCbCr 4:4:4,
  - YCbCr 4:2:2,
  - RGB 888;
- поддерживаемые форматы записи графических данных:
  - YCbCr 4:4:4,
  - YCbCr 4:2:2,
  - ARGB 8888;
- поддерживаемые разрешения захватываемых видеоданных:
  - SDTV 720x480i 60Hz,
  - SDTV 720x576i 50Hz,
  - EDTV 720(1440)x480i 60Hz,
  - EDTV 720(1440)x576i 50Hz,
  - EDTV 720x480p 60Hz,
  - EDTV 720x576p 50Hz,
  - HDTV 1280x720p 60Hz,
  - HDTV 1280x720p 50Hz,
  - HDTV 1920x1080i 60Hz,
  - HDTV 1920x1080i 50Hz,
  - HDTV 1920x1080p 30Hz,
  - HDTV 1920x1080p 25Hz;
  - HDTV 1920x1080p 60Hz,
  - HDTV 1920x1080p 50Hz;
- цветовые преобразования захватываемого изображения;
- гамма-коррекция захватываемого изображения;
- системная шина данных AMBA AXI с разрядностью по адресам 32 бит, по данным 64 бита;

- системная шина управления AMBA APB с разрядностью по адресам 32 бита, по данным 32 бита.
-
-

1.3.5.2.1.3 Структурная схема

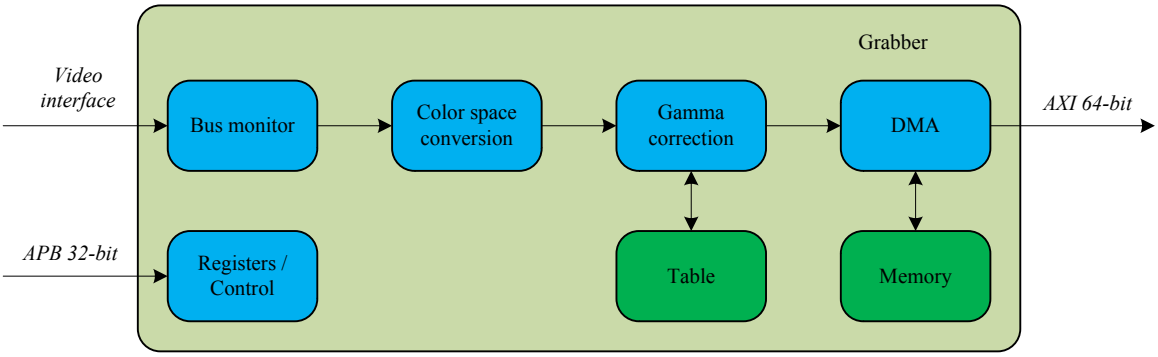


Рисунок 1.80 - Структурная схема устройства захвата видеоизображения

СФ-блок устройства захвата видеоизображения построен по модульному принципу, что позволяет на его базе формировать различные конфигурации, оптимально соответствующие различным вариантам его применения. На рисунке Рисунок 1.80 изображена структурная схема устройства захвата видеоизображения. Возможно исключение из тракта обработки ряда блоков, таких как блок преобразования цвета и блок гама-коррекции.

Registers/Control – блок обеспечивает управление устройством захвата видеоизображения. Обеспечивает прием и выдачу конфигурационной информации устройства видеозахвата по запросу внешнего ведущего устройства. Реализуется контроллером ведомого на APB, в соответствии со спецификацией AMBA™ 3 APB Protocol v1.0. Управляет функционированием блоков устройства видеозахвата и их взаимодействием. Обеспечивает запись/считывание управляющей информации в/из регистров устройства видеозахвата посредством системного интерфейса. Управляет режимами работы устройства видеозахвата. Преобразует внутренние управляющие сигналы и вырабатывает сигналы прерываний. Набор внутренних регистров устройства видеозахвата, обеспечивает хранение управляющей информации и внутреннего состояния устройства видеозахвата.

Bus monitor – блок предназначен для обработки входной цифровой телевизионной развёртки. Данный блок реализует захват активной части видеоизображения, выделение области интереса и преобразование внешнего формата данных во внутренний формат. Также данный блок реализует функцию распознавания параметров телевизионной развёртки: размеров активной части видеоизображения, количество активных шин.

Color space conversion – блок, реализующий преобразования представления цвета из цветовой модели YCbCr в RGB и наоборот для стандартов SDTV и HDTV и осуществления преобразования представления цвета из SDTV в HDTV и наоборот для формата данных YCbCr. Настройка блока осуществляется с помощью конфигурационной шины. Параметры настройки записываются в регистры устройства.

Gamma correction – Блок, реализующий гамма-коррекцию видеоизображения. Гамма-коррекция реализована табличным способом. Запись в таблицу осуществляется с помощью конфигурационной шины в регистры устройства.

Table – блок предназначен для хранения таблицы гамма-коррекции.

DMA – контроллер прямого доступа в память – обеспечивает генерацию запросов в память для записи видеоданных. На основе внутренней информации генерирует адреса для обращения в память для записи порции видеоданных, определяет размер порции. Имеет интерфейс AMBA® AXI Protocol v1.0.

Memory – блок предназначен для временного хранения видеоданных.

1.3.5.2.1.4 Описание формата расположения видеоданных в памяти

Устройство захвата видеоизображения захватывает активную (видимую) часть видеоизображения из кадра стандартной или высокой четкости и записывает её во внутреннюю память системы. Видеопоток представлен в виде последовательности кадров, которые хранятся в памяти раздельно. Устройство видеозахвата поддерживает следующие форматы записи данных в память:

- **YCbCr 4:2:2**
- **YCbCr 4:4:4**
- **ARGB 8888**

Видеоданные формата YCbCr, хранимые во внутренней памяти, разделены на яркостную и цветоразностную плоскости (плоскости компонент), которые хранятся в видеопамяти раздельно. При этом порядок данных внутри плоскости может быть только «little endian».

Устройством видеозахвата поддерживается три режима разделения кадра на плоскости компонент:

- расположение в одной плоскости (для данных формата ARGB 8888);
- режим двойного разделения (для данных форма YCbCr 4:2:2);
- режим тройного разделения (для данных форма YCbCr 4:2:2 и 4:4:4).

Устройство захвата видеоизображение поддерживает два формата видеоизображения:

- прогрессивный;
- чересстрочный.

При чересстрочном режиме, каждая плоскость компонент может храниться в памяти в виде одной области, так называемого кадра (как в прогрессивном режиме), так и в виде двух областей (полукадров).

Устройством видеозахвата поддерживается только линейный режим расположения данных в плоскости

Предполагается, что видеопамять состоит из 64х разрядных слов.

Расположение видеоданных во внутренней памяти системы

Устройство видеозахвата поддерживает расположения данных в памяти формата YCbCr 4:4:4 и YCbCr 4:2:2 в трёхплоскостном распределении и YCbCr 4:2:2 в двухплоскостном распределении, а также расположения данных формата ARGB 8888 в одной плоскости.

При двухплоскостном распределении видеоданных формата YCbCr память разделяется на 2 области. Первая область содержит в себе яркостную компоненту Y. Вторая область содержит цветоразностные компоненты Cb и Cr. В этом случае компоненты Cb и Cr чередуются, то есть компонента Cb располагается в ячейках с чётными адресами, а Cr располагается в ячейках с нечётными адресами. На рисунке 1.81 изображена структура памяти при двухплоскостном распределении данных формата YCbCr.

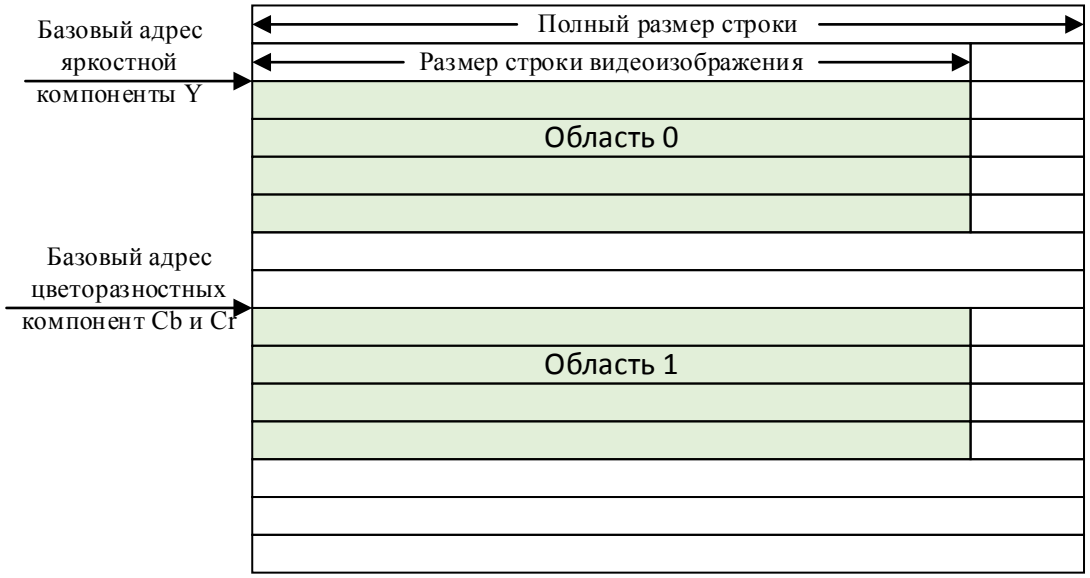


Рисунок 1.81 - Двухплоскостное разделение данных формата YCbCr в памяти

При расположении в памяти данных формата YCbCr 4:2:2, при двухплоскостном разделении, размер строки области для яркостной компоненты Y равен размеру строки области для цветоразностных компонент Cb и Cr. Также равно количество строк в областях яркостной Y и цветоразностных компонент Cb и Cr. Области памяти, в которых содержатся эти компоненты, равны.

При трёхплоскостном разделении видеоданных формата YCbCr память разделяется на 3 области. Каждая область содержит в себе только одну компоненту – Y, Cb или Cr. На рисунке 1.82 изображена структура памяти при трёхплоскостном разделении данных формата YCbCr.

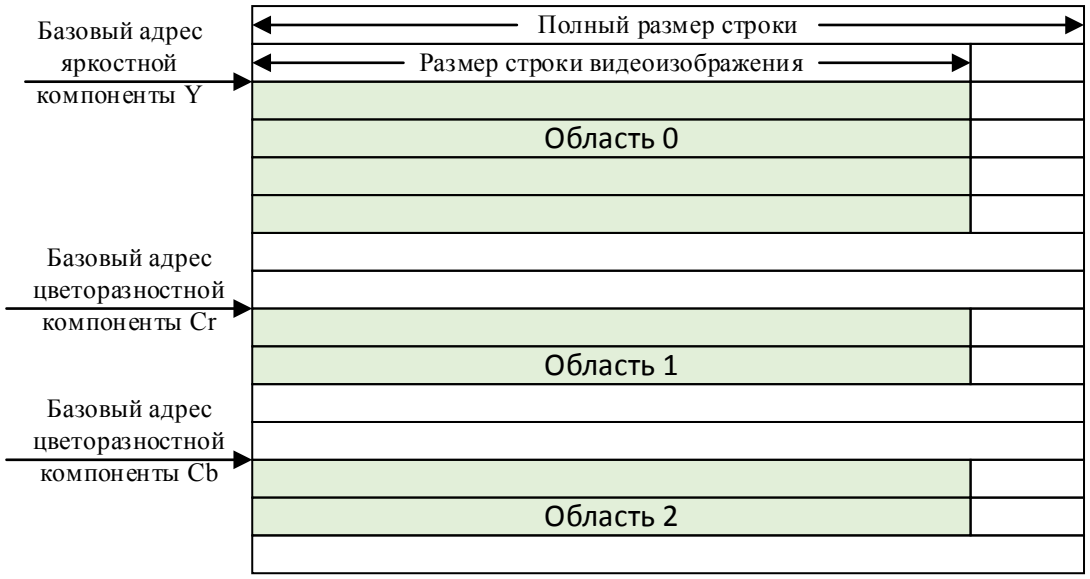


Рисунок 1.82 - Трёхплоскостное разделение данных формата YCbCr в памяти

Для формата YCbCr 4:2:2, при трёхплоскостном разделении, размер строки областей для цветоразностных компонент Cb и Cr меньше в 2 раза размера строки области для яркостной компоненты Y, так как строка видеоизображения для данного формата содержит в 2 раза меньше данных цветоразностных компонент Cb и Cr чем яркостная компонента Y. Количество строк в

областях для яркостной компоненты Y и цветоразностных компонент Cb и Cr равно. При этом области для цветоразностных компонент Cb и Cr равны.

Для формата YCbCr 4:4:4, в трёхплоскостном разделении, размер строк областей яркостной Y и цветоразностных компонент Cb и Cr, равны. Также количество строк в областях яркостной Y и цветоразностных компонент Cb и Cr равны. Из этого следует, что размеры трёх этих областей памяти равны.

Расположение данных формата RGB 888, при трёхплоскостном разделении, аналогично расположению данных YCbCr 4:4:4, при трёхплоскостном разделении.

При хранении в памяти данных формата ARGB 8888 данные располагаются в одной области в виде последовательности компонент синего B, зелёного G, красного R цветов и компоненты прозрачности. На рисунке 1.83 изображена структура памяти при расположении данных формата ARGB 8888.

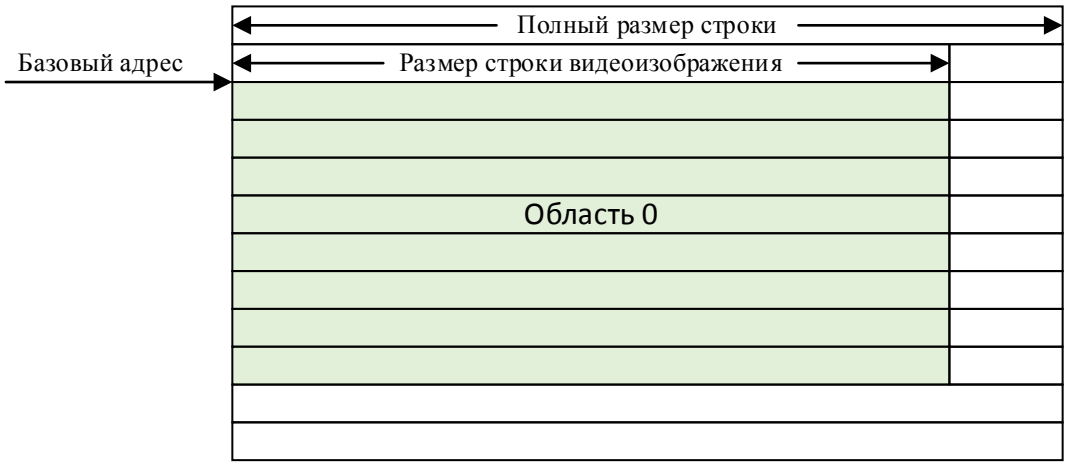


Рисунок 1.83 - Расположение данных формата ARGB 8888 во внутренней памяти системы

Порядок данных внутри плоскости

Устройство захвата видеоизображения поддерживает только линейный формат расположения данных. Изображение внутри плоскости компоненты расположено следующим образом:

- строки последовательно расположены сверху вниз;
- внутри каждой строки отсчеты расположены последовательно слева направо.

Данные в памяти лежат в виде 64-х разрядных слов. 64х разрядное слово памяти может содержать максимально до восьми 8ми разрядных отсчетов (компонент цвета) для плоскости компоненты. Внутри слова памяти первый отсчет содержится в байте с младшим адресом, т.е. в младшем байте LSB для формата little endian. Байты по следующим адресам содержат отсчеты в соответствии с правилами чтения изображения в плоскости.

На рисунке 1.84, на примере плоскости яркостной компоненты Y, показана структура 64-х разрядного слова, лежащего в памяти. Стоит отметить, что структура 64-х разрядного слова, лежащего в памяти, для цветоразностных компонент Cb и Cr при их раздельном расположении будет выглядеть аналогично (режим тройного разделения).

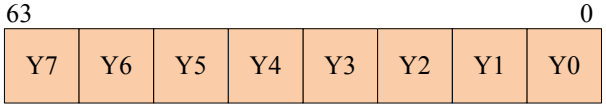


Рисунок 1.84 - Структура 64-х разрядного слова на примере плоскости яркостной компоненты Y (порядок данных little endian)

На рисунке 1.85 изображена структура плоскости памяти при расположении в ней данных формата YCbCr (порядок данных little endian). На рисунке изображена структура плоскости памяти на примере яркостной компоненты Y, стоит отметить, что для цветоразностных составляющих Cb и Cr структура плоскости памяти будет выглядеть аналогично, в случае, если они будут храниться в памяти раздельно (режим тройного разделения).

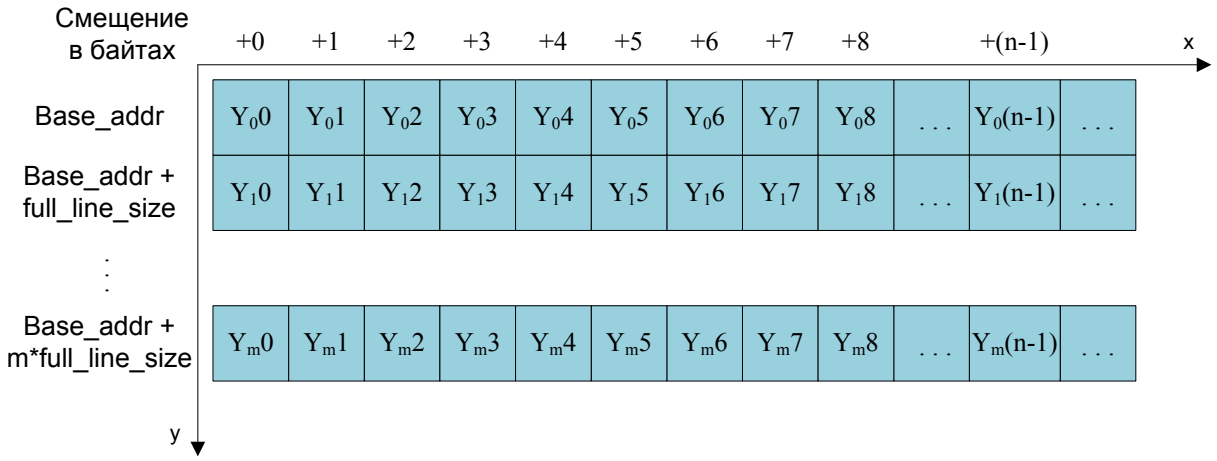


Рисунок 1.85 - Структура памяти на примере плоскости яркостных компонент Y (порядок данных little endian)

При расположении цветоразностных компонент Cb и Cr в одной плоскости (режим двойного разделения), структура 64-х разрядного слова, содержащего эти компоненты, будет выглядеть, как показано на рисунке Рисунок 1.86.

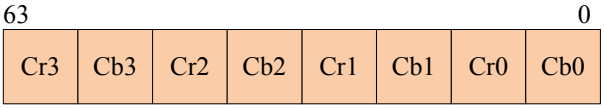


Рисунок 1.86 - Структура 64-х разрядного слова на примере плоскости цветоразностных компонент Cb и Cr (режим двойного разделения, порядок данных little endian)

На рисунке 1.87 изображена структура плоскости памяти при расположении в ней цветоразностных компонент Cb и Cr в одной плоскости (режим двойного разделения, порядок данных little endian).

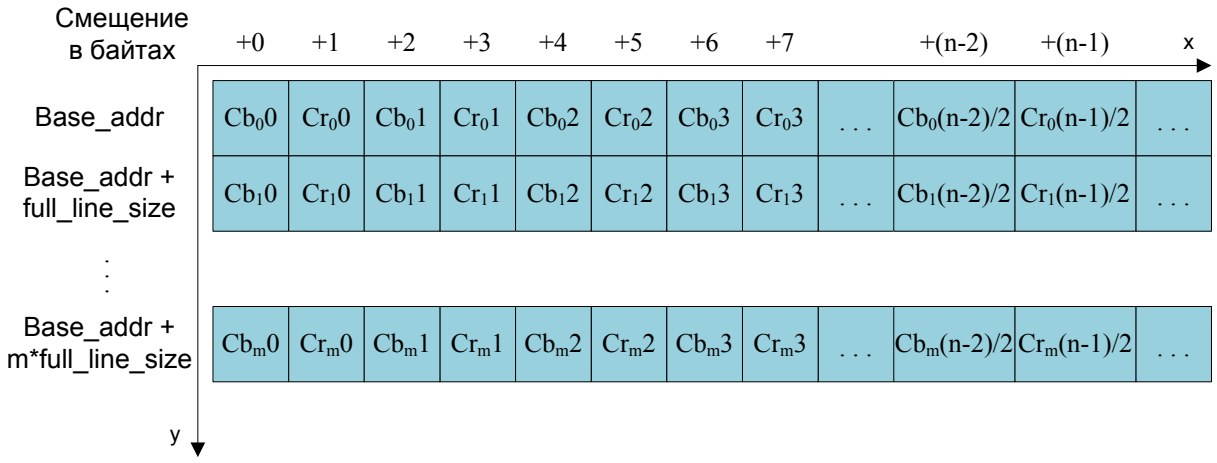


Рисунок 1.87 - Структура плоскости памяти при расположении в ней цветоразностных компонент Cb и Cr в одной плоскости (режим двойного разделения, порядок данных little endian)

На рисунке 1.88 изображена структура 64-х разрядного слова при расположении в памяти данных формата ARGB 8888 (порядок данных little endian).

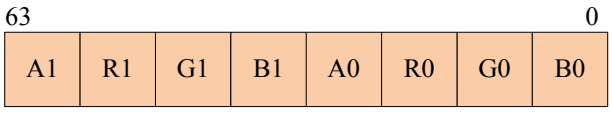


Рисунок 1.88 - Структура 64-х разрядного слова при расположении в памяти данных формата ARGB 8888 (порядок данных little endian)

На рисунке 1.89 изображена структура плоскости памяти при расположении в ней данных формата ARGB 8888 (порядок данных little endian).

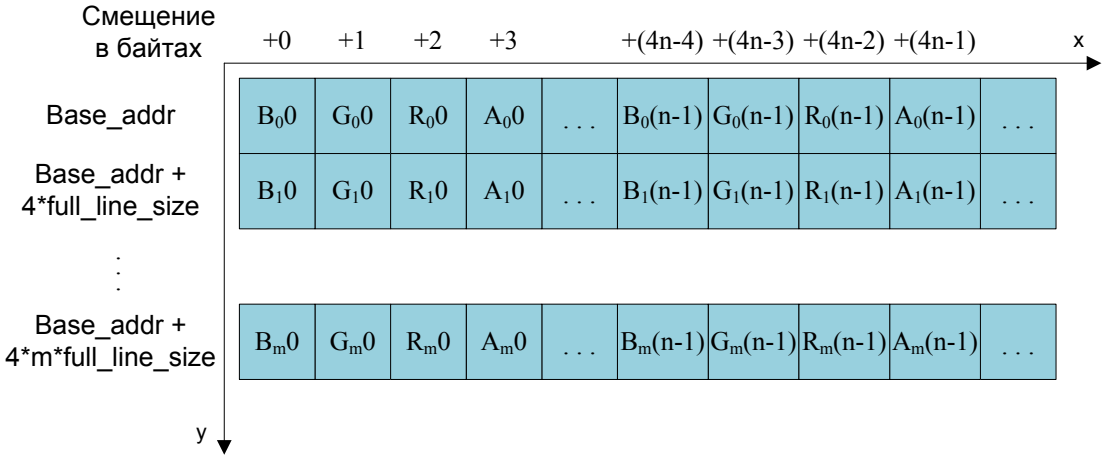


Рисунок 1.89 - Структура плоскости памяти при расположении в ней данных формата ARGB 8888 (порядок данных little endian)

1.3.5.2.1.5 Структура кадра входного видеоизображения

Устройство захвата видеоизображения захватывает активную часть видеоизображения телевизионной развёртки поддерживаемого формата. Передача видеоизображения сопровождается сигналами синхронизации, которые могут передаваться как по внешним линиям, так и встраиваться в поток видеоданных.

Кадр видеоизображения состоит из межстрочных и межкадровых интервалов (blank) и активного видеоизображения.

Существует два основных типа развертки: прогрессивная – при которой все строки активного изображения передаются последовательно друг за другом, и чересстрочная – при которой вначале передаются все четные строки активного видеоизображения, а затем – нечетные, таким образом, кадр состоит из двух полукадров.

Структура прогрессивного кадра представлена на рисунке 1.90.

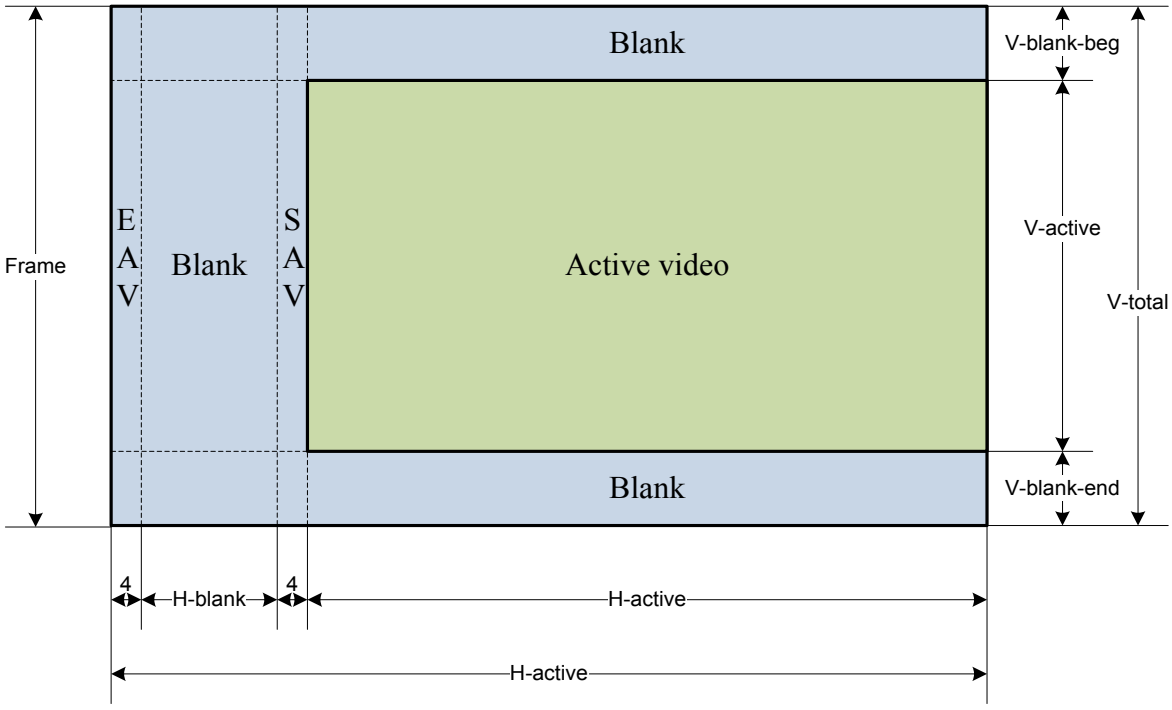


Рисунок 1.90. Структура кадра при прогрессивной развертке

Структура чересстрочного кадра представлена на рисунке 1.91

.

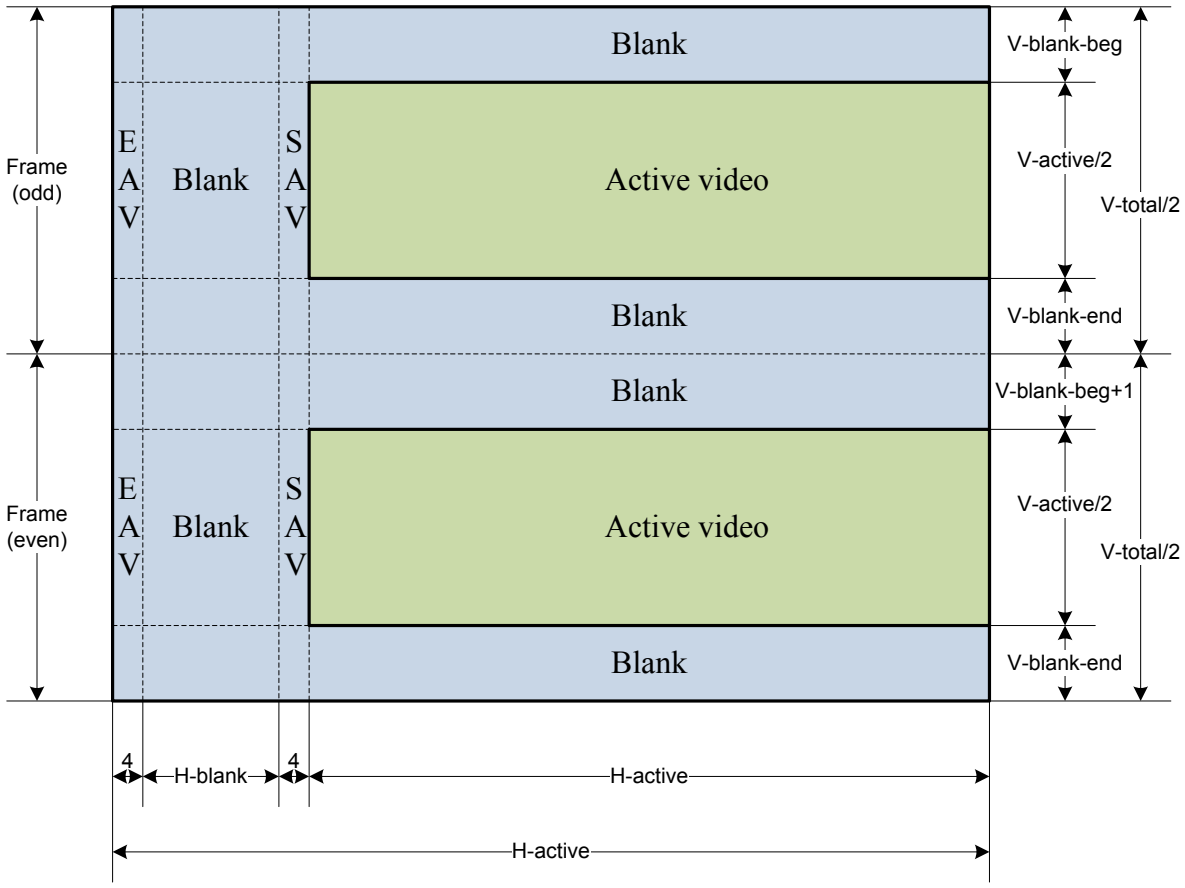


Рисунок 1.91 - Структура кадра при чересстрочной развертке

1.3.5.2.1.6 Входной видео интерфейс

Входной видео интерфейс состоит из 24-разрядной шины данных (video data) и сигналов синхронизации кадров.

В состав 24-х разрядной шины данных входят 3 линии данных (dv0, dv1 и dv2). Распределение линий dv0, dv1 и dv2 по шине данных video data представлено в таблице 1.104.

Таблица 1.132 - Распределение линий dv0, dv1 и dv2 по шине данных video data

Линия	Разряды шины данных video data
dv2	[23:16]
dv0	[15: 8]
dv1	[7: 0]

Устройство захвата видеоизображения обеспечивает захват сигналов структурирования кадра (развертки) в соответствии со стандартом CEA-861-E. К сигналам развертки (синхронизации) относятся:

Data Enable – сигнал, указывающий на данные активной части видеоизображения;

HSYNC – сигнал горизонтальной синхронизации;

VSYNC – сигнал вертикальной синхронизации;

FIELD – идентификатор выводимого полукадра (‘0’ – четный, ‘1’ – нечетный).

На рисунке 1.92 изображена прогрессивная развёртка в сопровождении сигналов синхронизации.

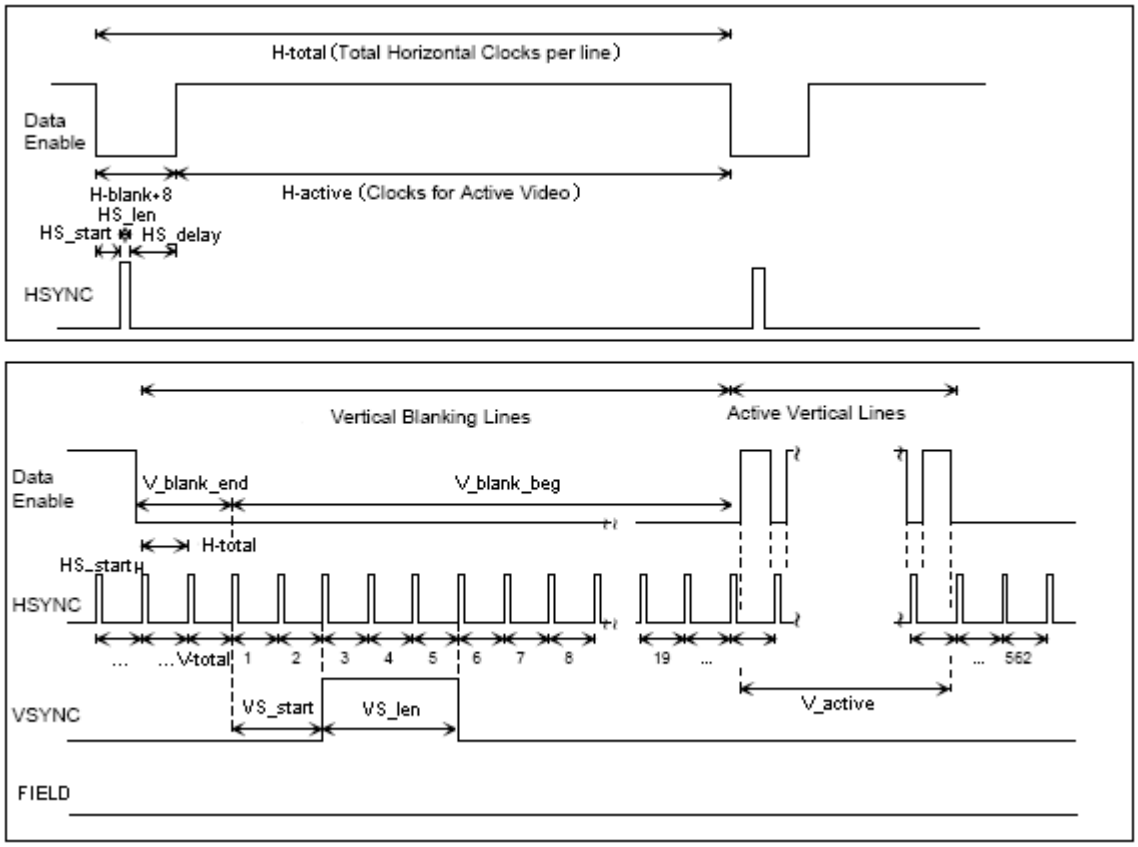


Рисунок 1.92 - Прогрессивная развёртка в сопровождении сигналов синхронизации

Компоненты цвета точек активной части видеоизображения передаются по шине данных видео интерфейса (video interface), в это время сигнала DataEnable находится в активном состоянии (подсвечивает данные).

Структурирующая информация (синхронизирующие сигналы) помимо передачи по отдельным линиям может передаваться и в потоке данных посредством кодов начала и конца активных данных видеоизображения (SAV, EAV).

Коды SAV, EAV – это 4-х байтовые посылки формата: FF, 00, 00, XY.

XY = 1, F, V, H, P3, P2, P1, P0 – где:

F – бит четности полукадра;

V – бит вертикальной развертки (вертикальной или кадровой синхронизации);

H – бит горизонтальной развертки (строчной синхронизации);

P3 = (V + H);

P2 = (F + H);

P1 = (F + V);

P0 = (F + V + H),

где – P0-P3 – биты четности, + - операция «исключающее или».

Биты F, V, H являются аналогами внешних сигналов FIELD, VSYNC, HSYNC.

На рисунке 1.93 изображена чересстрочная развёртка в сопровождении сигналов синхронизации.

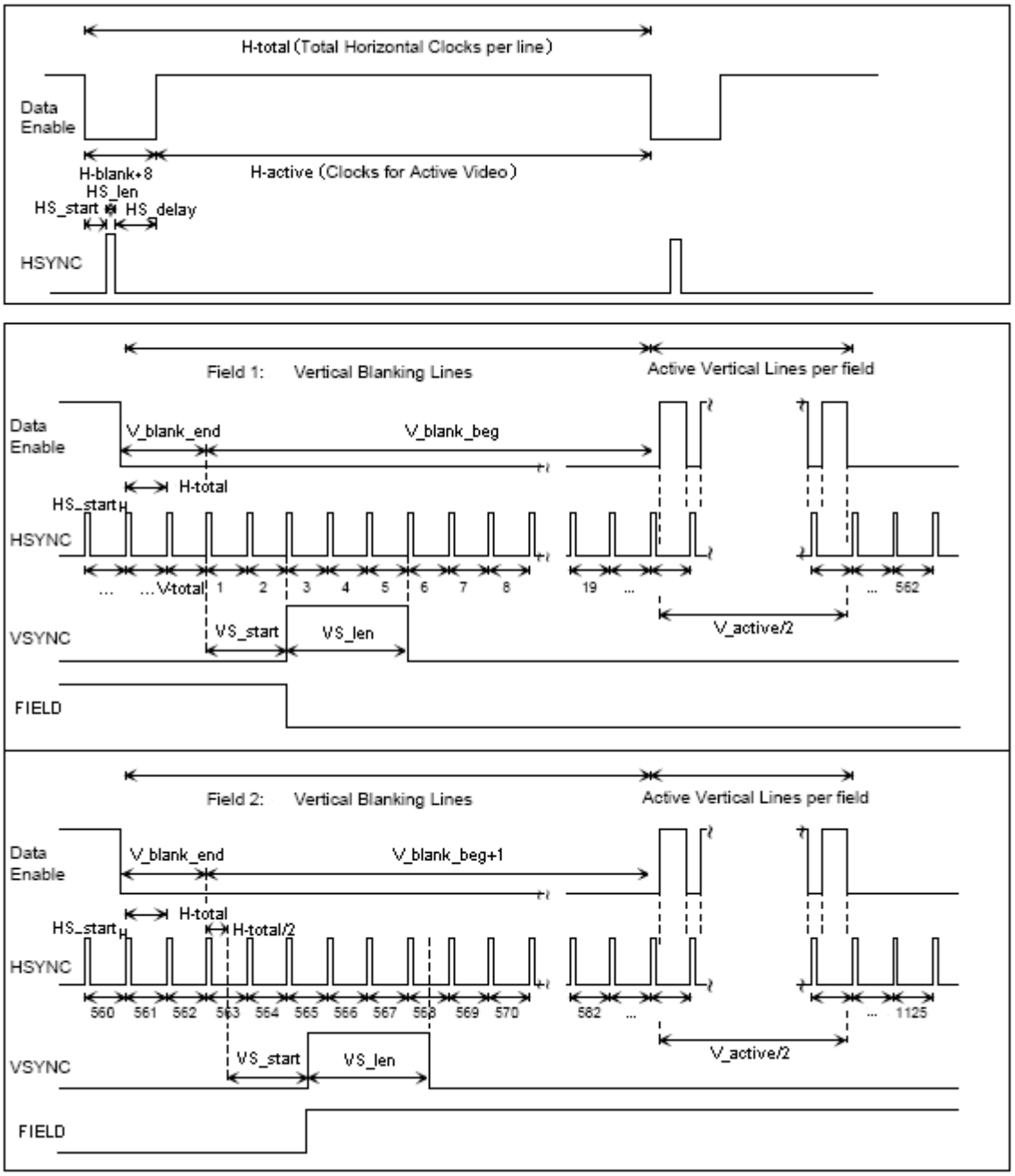


Рисунок 1.93 -Чересстрочная развёртка в сопровождении сигналов синхронизации

На рисунке 1.94 изображён поток данных видеоизображения формата SDTV по одной линии.

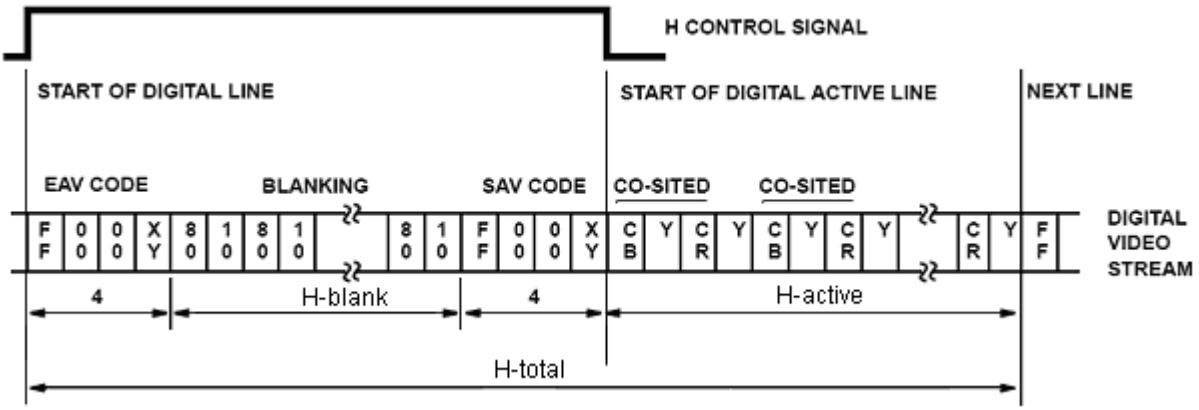


Рисунок 1.94 - Поток данных видеоизображения формата SDTV

На рисунке 1.95 изображён поток данных видеоизображения формата HDTV по двум линиям.

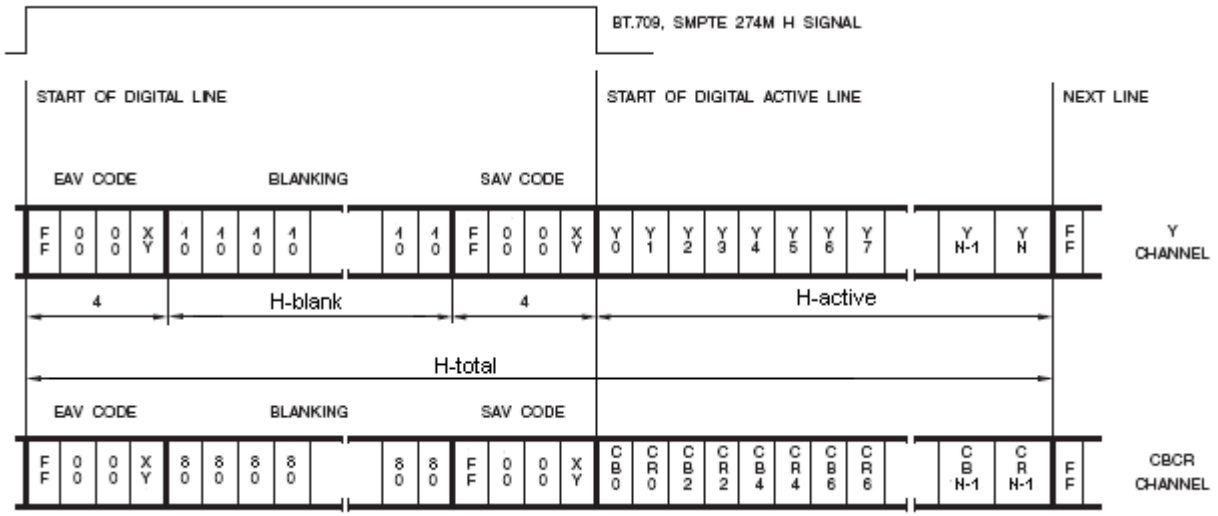


Рисунок 1.95 - Поток данных видеоизображения формата HDTV

1.3.5.2.1.7 Описание входных форматов видео интерфейса

Устройство захвата видеоизображения захватывает данные форматов YCbCr 4:2:2, YCbCr 4:4:4 и RGB 888. Разрядность яркостных и цветоразностных компонент точек – 8 бит.

Данные SDTV форматов могут передаваться двумя способами:

Последовательно по 8-ми разрядной шине (YCbCr 4:2:2). В этом случае яркостная (Y) и цветоразностные (Cb, Cr) составляющие точек передаются по одной шине в следующем порядке Cb0, Y0, Cr0, Y1, Cb2, Y2, Cr2, Y3 … и так далее. Значения передаются на частоте 27 МГц.

Параллельно по 24-х разрядной шине (YCbCr 4:2:2, YCbCr 4:4:4, RGB 888). В данном случае Y, Cb, Cr или R, G, B составляющие передаются по отдельным 8-разрядным шинам. В данном режиме производится репликация точек (каждая точка передается дважды), таким образом, данные передаются с частотой 13,5 МГц.

Данные EDTV, HDTV форматов могут передаваться двумя способами:

По 16-и разрядной шине (YCbCr 4:2:2). Яркостная составляющая (Y) и цветоразностная пара (Cb, Cr) передаются по собственным 8-разрядным шинам.

По 24-х разрядной шине (YCbCr 4:4:4, RGB 888). В данном случае Y, Cb, Cr или R, G, B составляющие передаются по отдельным 8-разрядным шинам. Репликация точек видеоизображения не производится.

В таблице 1.105 представлено распределение составляющих цвета точек на шине данных для различных YCbCr форматов видеоизображения.

Таблица 1.133 - Распределение составляющих цвета точек на шине данных для различных YCbCr форматов видеоизображения

VideoData [23:0]	SDTV		HDTV/EDTV	
	4:2:2	4:4:4	4:2:2	4:4:4
dv0[7:0]	Cb0, Y0, Cr0, Y1…	Y0, Y0, Y1, Y1…	Y0, Y1…	Y0, Y1…
dv1[7:0]	-	Cr0, Cr0, Cr1, Cr1…	Cb0, Cr0…	Cb0, Cb1…
dv2[7:0]	-	Cb0, Cb0, Cb1, Cr1…	-	Cr0, Cr1…

В таблице 1.106 представлено распределение составляющих цвета точек на шине данных для RGB формата видеоизображения.

Таблица 1.134. Распределение составляющих цвета точек на шине данных для RGB формата видеоизображения

VideoData [23:0]	SDTV	HDTV/EDTV
	888	888
dv0[7:0]	G0, G0, G1, G1…	G0, G1…
dv1[7:0]	R0, R0, R1, R1…	R0, R1…
dv2[7:0]	B0, B0, B1, B1…	B0, B1…

1.3.5.2.1.8 Программный (регистровый) интерфейс

Задание конфигурационных параметров и отображение текущего состояния устройства захвата видеоизображения ведется через регистры. Доступ к регистрам осуществляется посредством системного интерфейса управления APB. Каждое обращение к регистрам имеет выравнивание до 32-х разрядов. При обращении к регистрам, имеющим размер меньше ширины шины системного интерфейса управления (32 разряда), данные прижимаются (выравниваются) по правому краю (младшим разрядам).

При обращении к регистрам устройства захвата видеоизображения по зарезервированным адресам никаких ошибок не возникает, все пересылки завершаются нормально. При этом операции записи игнорируются, операции чтения возвращают все нули. При попытке записи в регистры, предназначенные только для чтения через системный интерфейс управления, операция записи завершается успешно, никакой записи не производится, состояние регистров не изменяется.

Регистры устройства захвата разделены на группы по функциональному признаку. Каждая группа имеет свой базовый адрес. Базовые адреса групп и список групп содержатся в таблице 1.107.

Таблица 1.135 - Список групп регистров и базовые адреса групп устройства захвата видеоизображения

Имя группы	Название	Базовые адреса групп
CONTROL	Управляющие регистры	0x0000_0000
STATUS	Статусные регистры	0x0000_0100
INTERRUPT	Регистры прерываний	0x0000_0200
COLOR_CONVERSION	Регистры цветовой коррекции	0x0000_0300
DMA_WR	Регистры интерфейса с памятью	0x0000_0400
GAMMA_TABLE	Таблицы функции гамма-коррекции	0x0000_0d00 0x0000_0e00 0x0000_0f00

1.3.5.2.1.8.1 CONTROL – Управляющие регистры

Таблица 1.136 - Управляющие регистры (CONTROL)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
ID_REG	Идентификационный номер устройства захвата видеоизображения.	0x000	r	32	0xec176627
ENABLE	Разрешение захвата видеоизображения.	0x004	rw	32	0x0
PR_RESET	Программный сброс устройства захвата видеоизображения.	0x008	rw	32	0x0
REC_ENABLE	Разрешение автоматического определения развёртки.	0x00c	rw	32	0x0

1.3.5.2.1.8.1.1 Регистр идентификационного номера устройства захвата видеоизображения (ID_REG)

Регистр содержит идентификационный номер устройства захвата видеоизображения.
Адрес: 0x000.
Разрядность: 32 разряда.

Таблица 1.137 - Разряды регистра идентификационного номера устройства захвата видеоизображения (ID_REG)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:0	ID_REG	Идентификационный номер устройства захвата видеоизображения.	r	0xec176627

1.3.5.2.1.8.1.2 Регистр разрешения захвата видеоизображения (ENABLE)

Регистр содержит управляющую информацию о разрешении/запрете захвата видеоизображения:
0 – захват видеоизображения запрещён;
1 – захват видеоизображения разрешён.
Адрес: 0x004.
Разрядность: 32 разряда.

Таблица 1.138 - Разряды регистра разрешения захвата видеоизображения (ENABLE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:1	RESERVED			
0	ENABLE	Разрешение захвата видеоизображения: 0 – захват видеоизображения запрещён; 1 – – захват видеоизображения разрешён.	rw	0x0

1.3.5.2.1.8.1.3 Регистр программного сброса устройства захвата видеоизображения (PR_RESET)

Регистр содержит управляющую информацию о программном сбросе устройства захвата видеоизображения.

Для выполнения программного сброса устройства захвата видеоизображения необходимо записать в регистр 1. После сброса регистр примет значение 0, что будет свидетельствовать о том, что программный сброс был выполнен.

Адрес: 0x008.
Разрядность: 32 разряда.

Таблица 1.139 - Разряды регистра программного сброса устройства захвата видеоизображения (PR_RESET)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:1	RESERVED			
0	PR_RESET	Программный сброс устройства захвата видеоизображения.	rw	0x0

1.3.5.2.1.8.2 Регистр разрешения автоматического определения развёртки (REC_ENABLE)

Данный регистр разрешает или запрещает устройству автоматически распознавать развёртку:

0 – автоматическое распознание развёртки запрещено;

1 – автоматическое распознание развёртки разрешено.

При включении данной функции устройство анализирует два целых кадра для прогрессивного или два полукадра для чересстрочного типа развёртки. После этого генерируется сигнал прерывания в соответствующем бите регистра состояния прерывания INT_STATUS.

Адрес: 0x00c.

Разрядность: 32 разряда.

Таблица 1.140 - Разряды регистра разрешения автоматического определения развёртки (REC_ENABLE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:1	RESERVED			
0	REC_ENABLE	Разрешение распознавания телевизионной развёртки: 0 – автоматическое распознание развёртки запрещено; 1 – автоматическое распознание развёртки разрешено.	rw	0x0

1.3.5.2.1.8.3 STATUS – Статусные регистры

Таблица 1.141 - Статусные регистры (STATUS)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
GRB_STATUS	Регистр статуса устройства видеозахвата	0x100	r	32	0x1501
FRAME_SIZE	Размеры изображения распознанной развёртки.	0x104	r	32	0x0
FRAME_PARAM	Параметры распознанной развёртки.	0x108	r	32	0x0 или 0x4
DMA0_STATUS	Регистр состояния записи нулевого канала	0x10c	r	32	0x0
DMA1_STATUS	Регистр состояния записи первого канала	0x110	r	32	0x0
DMA2_STATUS	Регистр состояния записи второго канала	0x114	r	32	0x0
DMA_ERROR	Ответ с AXI шины bresp для нулевого, первого и второго каналов	0x118	r	32	0x0

1.3.5.2.1.8.3.1 Регистр статуса (GRB_STATUS)

Данный регистр содержит статусную информацию об устройстве захвата видеоизображения.

Адрес: 0x100.

Разрядность: 32 разряда.

Таблица 1.142 - Разряды регистра статуса (GRB_STATUS)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:14	RESERVED			
13	CH2_BUF_FULL	Буфер второго канала полный	r	0x0
12	CH2_BUF_EMPTY	Буфер второго канала пустой	r	0x1
11	CH1_BUF_FULL	Буфер первого канала полный	r	0x0
10	CH1_BUF_EMPTY	Буфер первого канала пустой	r	0x1
9	CH0_BUF_FULL	Буфер нулевого канала полный	r	0x0
8	CH0_BUF_EMPTY	Буфер нулевого канала пустой	r	0x1
7:5	RESERVED			
4	WORKING	Данный сигнал статуса содержит информацию о захвате изображения: 0 – захват изображения не проводится, 1 – проводится захват изображения.	r	0x0
3:1	RESERVED			
0	ACTIVE_FRAME	Данный регистр содержит номер активной страницы видеопамяти (по которой в данный момент происходит запись данных): 0 – активна страница с индексом 0; 1 – активна страница с индексом 1.	r	0x1

1.3.5.2.1.8.3.2 Регистр размера изображения распознанной развёртки (FRAME_SIZE)

Данный регистр содержит в себе информацию о размере изображения распознанной развёртки. При поступлении на вход устройства развёртки стандарта SDTV значение горизонтального размера изображения HOR_SIZE будет в два раза больше реального горизонтального размера изображения. При чересстрочном типе развёртки значение вертикального размера изображения VER_SIZE будет в два раза меньше реального вертикального размера кадра.

Адрес: 0x104.

Разрядность: 32 разряда.

Таблица 1.143 - Разряды регистра размера изображения распознанной развёртки (FRAME_SIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:28	RESERVED			
27:16	VER_SIZE	Вертикальный размер изображения	r	0x0
15:12	RESERVED			
11:0	HOR_SIZE	Горизонтальный размер изображения	r	0x0

1.3.5.2.1.8.3.3 Регистр параметров распознанной развёртки (FRAME_PARAM)

Данный регистр содержит в себе информацию о параметрах развёртки.
Адрес: 0x108.
Разрядность: 32 разряда.

Таблица 1.144 - Разряды регистра параметров распознанной развёртки (FRAME_PARAM)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:11	RESERVED			
10	DV2_ACTIVE	Сигнал активности данных на линии 0 (dv0, устанавливается в 1 при поступлении на вход ненулевых данных)	r	0x0
9	DV1_ACTIVE	Сигнал активности данных на линии 1 (dv1, устанавливается в 1 при поступлении на вход ненулевых данных)	r	0x0
8	DV0_ACTIVE	Сигнал активности данных на линии 2 (dv2, устанавливается в 1 при поступлении на вход ненулевых данных)	r	0x0
7	RESERVED			
6	HSYNC_POL	Полярность сигнала горизонтальной синхронизации	r	0x0
5	VSYNC_POL	Полярность сигнала вертикальной синхронизации	r	0x0
4	INTERLACE	Прогрессивный или чересстрочный тип развёртки	r	0x0
3	RESERVED			
2	VI_I_ENABLE	Сигнал активности входного цифрового видео интерфейса (устанавливается по фронту сигнала синхронизации pix_clk_i). Исходное значение равно: 0 – в случае, если на вход не поступает сигнал синхронизации pix_clk_i; 1 – в случае, если на вход поступает сигнал синхронизации pix_clk_i.	r	0x0 или 0x1
1:0	ACTIVE_BUS	Количество активных шин, по которым передаётся изображение: 1 – при передаче по линии dv0 2 – при передаче данных по линиям dv0 и dv1 3 – при передаче данных по линиям dv0, dv1 и dv2	r	0x0

1.3.5.2.1.8.3.4 Регистры состояния записи нулевого, первого и второго каналов (DMA0_STATUS, DMA1_STATUS и DMA2_STATUS)

Данный регистр содержит информацию о записанных в память точек и линий.
Адреса: 0x10c, 0x110, и 0x114.
Разрядность: 32 разряда.

Таблица 1.145- Разряды регистров состояния записи (DMA0_STATUS, DMA1_STATUS и DMA2_STATUS)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:28	RESERVED			
27:16	COUNT_Y	Счётчик записанных линий в память. Счётчик показывает сколько строк осталось записать.	r	0x0
15:12	RESERVED			
11:0	COUNT_X	Счётчик записанных точек в память.	r	0x0

1.3.5.2.1.8.3.5 Регистр ответов с AXI шины bresp для нулевого, первого и второго каналов (DMA_ERROR)

Данный регистр содержит в себе информацию об ответах с AXI шины bresp для нулевого, первого и второго каналов.

Адрес: 0x118.

Разрядность: 32 разряда.

Таблица 1.146 - Разряды регистра ответов с AXI шины записи ответов bresp для нулевого, первого и второго каналов (DMA_ERROR)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:9	RESERVED			
8:7	DMA0_BRESP	Ответ bresp от ведомого устройства на шине AXI для нулевого канала	r	0x0
6	DMA0_FLAG	Флаг ответа ошибки с AXI шины bresp для нулевого канала: 1 – получен ответ «ошибка»; 0 – ответ «ошибка» получен не был.	r	0x0
5:4	DMA1_BRESP	Ответ bresp от ведомого устройства на шине AXI для первого канала	r	0x0
3	DMA1_FLAG	Флаг ответа ошибки с AXI шины bresp для первого канала: 1 – получен ответ «ошибка»; 0 – ответ «ошибка» получен не был.	r	0x0
2:1	DMA2_BRESP	Ответ bresp от ведомого устройства на шине AXI для первого канала	r	0x0
0	DMA2_FLAG	Флаг ответа ошибки с AXI шины bresp для второго канала: 1 – получен ответ «ошибка»; 0 – ответ «ошибка» получен не был.	r	0x0

1.3.5.2.1.8.4 INTERRUPT – Регистры прерываний

Таблица 1.147 - Регистры прерываний (INTERRUPT)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. Знач.
INT_STATUS	Регистр состояния прерывания	0x200	rw	32	0x0
INT_MASK	Регистр разрешения прерывания	0x204	rw	32	0x0
TEST_INT	Регистр тестового прерывания	0x208	rw	32	0x0
INTERRUPTION	Регистр состояния прерывания	0x20c	r	32	0x0

Группа регистров прерывания содержит два основных регистра – регистр разрешения прерываний INT_MASK и регистр состояний прерываний INT_STATUS. Эти регистры имеют одинаковую структуру. При возникновении прерывания в аппаратуре устройства захвата видеоизображения оно фиксируется в регистре состояния прерываний вне зависимости от того установлен ли соответствующий бит в регистре разрешения прерываний. Если в разряде регистра разрешения прерываний установлена единица и в регистре состояния зафиксировано возникновение соответствующего прерывания, то устройством захвата видеоизображения генерируется сигнал о возникновении прерывания, который выдается в систему.

Для снятия прерывания в соответствующий разряд регистра статуса прерываний (INT_STATUS) должна быть записана единица. При записи нулевого значения в соответствующий разряд регистра прерываний его значение не изменяется.

1.3.5.2.1.8.4.1 Регистр состояния прерывания (INT_STATUS)

Регистр содержит информацию, отображающую состояние прерываний устройства захвата видеоизображения. При возникновении прерывания аппаратура выставляет в соответствующем разряде регистра INT_STATUS единичное значение, свидетельствующее о возникновении прерывания. В дальнейшем анализируется состояние регистра разрешения прерываний и при единичном значении в соответствующем разряде регистра INT_MASK на основании сборки сигналов данного регистра вырабатывается единый сигнал о запросе прерывания.

Для снятия сигнала прерывания в соответствующий разряд регистра INT_STATUS должно быть записано единичное значение.

Адреса: 0x200.

Разрядность: 32 разряда.

Таблица 1.148 - Разряды регистра состояния прерывания (INT_STATUS)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31	CH2_OVERFLOW	Запрос прерывания по событию: запись в полный буфер второго канала	rw	0x0
30:29	RESERVED			
28	CH1_OVERFLOW	Запрос прерывания по событию: запись в полный буфер первого канала	rw	0x0
27:26	RESERVED			
25	CH0_OVERFLOW	Запрос прерывания по событию: запись в полный буфер нулевого канала	rw	0x0
24:19	RESERVED			
18	RECEIVE_END	Запрос прерывания по событию: окончание захвата кадра	rw	0x0
17	RESERVED			
16	INT_VSYNC	Запрос прерывания по событию: фронт сигнала развёртки вертикальной синхронизации	rw	0x0
15	INT_HSYNC	Запрос прерывания по событию: фронт сигнала развёртки горизонтальной синхронизации	rw	0x0

Продолжение таблицы 1.149

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
14	INT_ST_ERROR	Запрос прерывания по событию: несовпадение количества шин, по которым передаются видеоданные, в соседних кадрах	rw	0x0
13	INT_VS_ERROR	Запрос прерывания по событию: несовпадение количества линий в соседних кадрах	rw	0x0
12	INT_HS_ERROR	Запрос прерывания по событию: несовпадение количество точек в соседних линиях	rw	0x0
11	INT_DONE	Запрос прерывания по событию: окончание распознавания развёртки	rw	0x0
10:9	RESERVED			
8	GRB_OFF	Запрос прерывания по событию: состояние активности устройства захвата видеоизображения	rw	0x0
7	END_WRITE	Запрос прерывания по событию: окончание записи кадра в память	rw	0x0
6	INT_UNC_ERR	Запрос прерывания по событию: неисправимая ошибка синхрокода EAV и SAV	rw	0x0
5	INT_REPL	Запрос прерывания по событию: исправление ошибки синхрокода EAV и SAV	rw	0x0
4	INT_ERR_CODE	Запрос прерывания по событию: ошибка синхрокода EAV и SAV	rw	0x0
3	DMA2_ERROR	Запрос прерывания по событию: при записи в память второго канала ответ с AXI «ошибка»	rw	0x0
2	DMA1_ERROR	Запрос прерывания по событию: при записи в память первого канала ответ с AXI «ошибка»	rw	0x0
1	DMA0_ERROR	Запрос прерывания по событию: при записи в память нулевого канала ответ с AXI «ошибка»	rw	0x0
0	TEST_INT	Запрос прерывания по событию: тестовый запрос прерывания в активном состоянии	rw	0x0

1.3.5.2.1.8.4.2 Регистр разрешения прерывания (INT_MASK)

Регистр содержит информацию о разрешении генерации сигнала прерывания:
0 – прерывание запрещено;
1 – прерывание разрешено.
Адреса: 0x204.
Разрядность: 32 разряда.

Таблица 1.150. Разряды регистра разрешения прерывания (INT_MASK)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31	CH2_OVERFLOW	Разрешение генерации сигнала прерывания по событию: запись в полный буфер второго канала	rw	0x0
30:29	RESERVED			
28	CH1_OVERFLOW	Разрешение генерации сигнала прерывания по событию: запись в полный буфер первого канала	rw	0x0
27:26	RESERVED			
25	CH0_OVERFLOW	Разрешение генерации сигнала прерывания по событию: запись в полный буфер нулевого канала	rw	0x0
24:19	RESERVED			
18	RECEIVE_END	Разрешение генерации сигнала прерывания по событию: окончание захвата кадра	rw	0x0

Продолжение таблицы 1.151

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
17	RESERVED			
16	INT_VSYNC	Разрешение генерации сигнала прерывания по событию: фронт сигнала развёртки вертикальной синхронизации	rw	0x0
15	INT_HSYNC	Разрешение генерации сигнала прерывания по событию: фронт сигнала развёртки горизонтальной синхронизации	rw	0x0
14	INT_ST_ERROR	Разрешение генерации сигнала прерывания по событию: несовпадение количества шин, по которым передаются видеоданные, в соседних кадрах	rw	0x0
13	INT_VS_ERROR	Разрешение генерации сигнала прерывания по событию: несовпадение количества линий в соседних кадрах	rw	0x0
12	INT_HS_ERROR	Разрешение генерации сигнала прерывания по событию: несовпадение количество точек в соседних линиях	rw	0x0
11	INT_DONE	Разрешение генерации сигнала прерывания по событию: окончание распознавания развёртки	rw	0x0
10:9	RESERVED			
8	GRB_OFF	Разрешение генерации сигнала прерывания по событию: состояние активности устройства захвата видеоизображения	rw	0x0
7	END_WRITE	Разрешение генерации сигнала прерывания по событию: окончание записи кадра в память	rw	0x0
6	INT_UNC_ERR	Разрешение генерации сигнала прерывания по событию: неисправимая ошибка синхрокода EAV и SAV	rw	0x0
5	INT_REPL	Разрешение генерации сигнала прерывания по событию: исправление ошибки синхрокода EAV и SAV	rw	0x0
4	INT_ERR_CODE	Разрешение генерации сигнала прерывания по событию: ошибка синхрокода EAV и SAV	rw	0x0
3	DMA2_ERROR	Разрешение генерации сигнала прерывания по событию: при записи в память второго канала ответ с AXI «ошибка»	rw	0x0
2	DMA1_ERROR	Разрешение генерации сигнала прерывания по событию: при записи в память первого канала ответ с AXI «ошибка»	rw	0x0
1	DMA0_ERROR	Разрешение генерации сигнала прерывания по событию: при записи в память нулевого канала ответ с AXI «ошибка»	rw	0x0
0	TEST_INT	Разрешение генерации сигнала прерывания по событию: тестовый запрос прерывания в активном состоянии	rw	0x0

1.3.5.2.1.8.4.3 Регистр тестового прерывания (TEST_INT)

Регистр состояния тестового прерывания:
0 – событие прерывания в неактивном состоянии;
1 – событие прерывания в активном состоянии.
Адреса: 0x208.
Разрядность: 32 разряда.

Таблица 1.152 - Разряды регистра тестового прерывания (TEST_INT)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:1	RESERVED			
0	TEST_INT	Регистр состояния сигнала тестового запроса прерывания: 0 – сигнал тестового запроса прерывания в неактивном состоянии; 1 – сигнал тестового запроса прерывания в активном состоянии.	rw	0x0

1.3.5.2.1.8.4.4 Регистр состояния прерывания (INTERRUPTION)

Регистр содержит информацию о статусе запроса прерывания:
0 – неактивный уровень сигнала прерывания;
1 – активный уровень сигнала прерывания.
Адреса: 0x20c.
Разрядность: 32 разряда.

Таблица 1.153 - Разряды регистра состояния прерывания (INTERRUPTION)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:1	RESERVED			
0	INTERRUPT	Регистр содержит информацию о статусе запроса прерывания: 0 – неактивный уровень сигнала прерывания; 1 – активный уровень сигнала прерывания.	r	0x0

1.3.5.2.1.8.5 COLOR_CONVERSION – Регистры цветовой коррекции

Таблица 1.154 - Регистры цветовой коррекции (COLOR_SPACE)

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
GAM_ENABLE	Разрешение гамма коррекции	0x300	rw	32	0x0
CONV_ENABLE	Разрешение преобразования цветовых пространств	0x304	rw	32	0x0
C_0_0	Коэффициент с индексом 0 в нулевой строке формулы преобразования цветовых пространств.	0x308	rw	32	0x0
C_0_1	Коэффициент с индексом 1 в нулевой строке формулы преобразования цветовых пространств.	0x30c	rw	32	0x0
C_0_2	Коэффициент с индексом 2 в нулевой строке формулы преобразования цветовых пространств.	0x310	rw	32	0x0
C_0_3	Коэффициент с индексом 3 в нулевой строке формулы преобразования цветовых пространств.	0x314	rw	32	0x0
C_1_0	Коэффициент с индексом 0 в первой строке формулы преобразования цветовых пространств.	0x318	rw	32	0x0
C_1_1	Коэффициент с индексом 1 в первой строке формулы преобразования цветовых пространств.	0x31c	rw	32	0x0
C_1_2	Коэффициент с индексом 2 в первой строке формулы преобразования цветовых пространств.	0x320	rw	32	0x0
C_1_3	Коэффициент с индексом 3 в первой строке формулы преобразования цветовых пространств.	0x324	rw	32	0x0
C_2_0	Коэффициент с индексом 0 во второй строке формулы преобразования цветовых пространств.	0x328	rw	32	0x0
C_2_1	Коэффициент с индексом 1 во второй строке формулы преобразования цветовых пространств.	0x32c	rw	32	0x0
C_2_2	Коэффициент с индексом 2 во второй строке формулы преобразования цветовых пространств.	0x330	rw	32	0x0
C_2_3	Коэффициент с индексом 3 во второй строке формулы преобразования цветовых пространств.	0x334	rw	32	0x0
CH0_RANGE	Нижняя и верхняя граница выходных данных для яркостной составляющей Y	0x338	rw	32	0x0
CH1_RANGE	Нижняя и верхняя граница выходных данных для цветоразностной составляющей Cr	0x33c	rw	32	0x0
CH2_RANGE	Нижняя и верхняя граница выходных данных для цветоразностной составляющей Cr	0x340	rw	32	0x0

1.3.5.2.1.8.5.1 Регистр разрешения гамма коррекции (GAM_ENABLE)

Регистр разрешения гамма-коррекции изображения;
0 – гамма-коррекция изображения запрещена;
1– гамма-коррекция изображения разрешена.
Адрес: 0x300.
Разрядность: 32 разряда.

Таблица 1.155 - Разряды регистра разрешения гамма коррекции (GAM_ENABLE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:1	RESERVED			
0	GAM_ENABLE	Регистр разрешения гамма-коррекции изображения: 0 – гамма-коррекция запрещена; 1 – гамма-коррекция разрешена.	rw	0x0

1.3.5.2.1.8.5.2 Регистр разрешения преобразования цветового пространств (CONV_ENABLE)

Регистр разрешения преобразования цветовой модели представления цвета:
0 – преобразование запрещено;
1 – преобразование разрешено.
Адрес: 0x304.
Разрядность: 32 разряда.

Таблица 1.156 - Разряды регистра разрешения преобразования цветового пространств (CONV_ENABLE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:1	RESERVED			
0	CONV_ENA	Регистр разрешения преобразования цветовой модели представления цвета: 0 – преобразование запрещено; 1 – преобразование разрешено.	rw	0x0

1.3.5.2.1.8.5.3 Регистры коэффициентов формулы преобразования цветовых пространств (C_0_1, C_0_2, C_0_3, C_1_1, C_1_2, C_1_3, C_2_1, C_2_2 и C_2_3)

Данный регистр содержит значения коэффициентов и знак, входящих в формулу для преобразования цветовых пространств.
Адреса: 0x30c, 0x310, 0x314, 0x31c, 0x320, 0x324, 0x32c, 0x330 и 0x334.
Разрядность: 32 разряда.

Таблица 1.157 - Разряды регистров коэффициентов формулы преобразования цветовых пространств (C_0_1, C_0_2, C_0_3, C_1_1, C_1_2, C_1_3, C_2_1, C_2_2 и C_2_3)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:11	RESERVED			
10	SIGN	Знак коэффициента, входящего в формулу: 0 – плюс; 1 – минус.	rw	0x0
9:0	COEFFICIENT	Значение коэффициента, умноженное на 256 и округлённое до целого значения.	rw	0x0

1.3.5.2.1.8.5.4 Регистры коэффициентов формулы преобразования цветовых пространств (C_0_0, C_1_0 и C_2_0)

Данный регистр содержит значения коэффициентов и знак, входящих в формулу для преобразования цветовых пространств.
Адреса: 0x308, 0x318 и 0x328.
Разрядность: 32 разряда.

Таблица 1.158 - Разряды регистров коэффициентов формулы преобразования цветовых пространств (C_0_0, C_1_0 и C_2_0)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:19	RESERVED			
18	SIGN	Знак коэффициента, входящего в формулу: 0 – плюс; 1 – минус.	rw	0x0
17:0	COEFFICIENT	Значение коэффициента, умноженное на 256 и округлённое до целого значения.	rw	0x0

1.3.5.2.1.8.5.5 Регистры нижней и верхней границы выходных данных первого, второго и третьего каналов (CH0_RANGE, CH1_RANGE и CH2_RANGE)

Данные регистры содержат значение нижней и верхней границы диапазона для выходных значений, преобразованных видеоданных.

Адреса: 0x338, 0x33c, 0x340.

Разрядность: 32 разряда.

Таблица 1.159 - Разряды регистров нижней и верхней границы выходных данных первого, второго и третьего каналов (CH0_RANGE, CH1_RANGE и CH2_RANGE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:24	RESERVED			
23:16	MAX_VALUE	Верхняя граница выходных данных.	rw	0x0
15:8	RESERVED			
7:0	MIN_VALUE	Нижняя граница выходных данных.	rw	0x0

1.3.5.2.1.8.6 DMA_WR – Регистры интерфейса с памятью

Таблица 1.160 - Регистры интерфейса с памятью DMA_WR

Имя	Название (описание)	Адрес	Реж	Разр	Исх. знач.
BASE_SW_ENA	Разрешение переключения базовых адресов	0x400	rw	32	0x0
DMA0_ADDR_0	Базовый адрес для яркостной составляющей для нечётных кадров	0x404	rw	32	0x0
DMA0_ADDR_1	Базовый адрес для яркостной составляющей для чётных кадров	0x408	rw	32	0x0
DMA1_ADDR_0	Базовый адрес для цветоразностной составляющей для нечётных кадров	0x40c	rw	32	0x0
DMA1_ADDR_1	Базовый адрес для цветоразностной составляющей для чётных кадров	0x410	rw	32	0x0
DMA2_ADDR_0	Базовый адрес для цветоразностной составляющей для нечётных кадров	0x414	rw	32	0x0
DMA2_ADDR_1	Базовый адрес для цветоразностной составляющей для чётных кадров	0x418	rw	32	0x0
Y_SIZE	Размер кадра изображения для яркостной компоненты	0x41c	rw	32	0x0
C_SIZE	Размер кадра изображения для цветоразностной компоненты	0x420	rw	32	0x0
FULL_LINE_SIZE	Полный размер строки изображения в памяти	0x424	rw	32	0x0
MODE	Формат входного потока данных	0x428	rw	32	0x0
LOC_DATA	Формат расположения данных в памяти	0x42c	rw	32	0x0
TRANSPARENCY	Регистр коэффициента прозрачности	0x430	rw	32	0x0
BASE_POINT	Базовая точка захвата видеоизображения	0x434	rw	32	0x0
DMA_ID	Идентификационные номера DMA	0x438	rw	32	0xfa5
AXI_PARAM	Параметры контроллера ведущего на шине AXI	0x43c	rw	32	0xf31_0000

1.3.5.2.1.8.6.1 Регистр разрешения переключения базовых адресов (BASE_SW_ENA)

Регистр содержит информацию о разрешении переключения страниц видеопамяти (базовых адресов):

0 – переключение запрещено;

1 – переключение разрешено.

Когда переключение страниц разрешено, устройство захвата видеоизображения переключается между двумя страницами видеопамяти, адреса которых расположены в регистрах «базовый адрес для нечётных кадров» и «базовый адрес для чётных кадров». При запрете переключения между этими адресами, последующий кадр записываться в память не будет, пока снова не будет разрешено переключение.

Адрес: 0x400.

Разрядность: 32 разряда.

Таблица 1.161 - Разряды регистра разрешения переключения базовых адресов (BASE_SW_ENA)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:1	RESERVED			
0	BASE_SW_ENA	Регистр содержит информацию о разрешении переключения страниц видеопамяти (базовых адресов): 0 – переключение запрещено; 1 – переключение разрешено.	rw	0x0

1.3.5.2.1.8.6.2 Регистры базовых адресов доступной области памяти для компонент (DMA0_ADDR_0, DMA0_ADDR_1, DMA1_ADDR_0, DMA1_ADDR_1, DMA2_ADDR_0 и DMA2_ADDR_1)

Данные регистры содержат базовые адреса доступных областей памяти для различных компонент видеоизображения.

Базовый адрес страницы памяти должен быть выровнен на границу слова памяти, то есть должен быть кратен 8.

Существует два набора данных регистров с индексом «0» и «1». При этом, в текущий момент времени один из наборов, например «0», содержит значение текущего базового адреса – того по которому в данный момент времени ведется запись захваченных видеоданных, при этом второй набор, в данном случае «1» содержит базовый адрес положения в памяти следующего кадра.

После окончания записи видеостраницы по данному базовому адресу статус наборов регистра переключается и «0» станет следующим, а «1» – текущим (при разрешенном переключении страниц).

Адреса: 0x404, 0x408, 0x40c, 0x410, 0x414 и 0x418.

Разрядность: 32 разряда.

Таблица 1.162 - Разряды регистра базовых адресов доступной области памяти для компонент (DMA0_ADDR_0, DMA0_ADDR_1, DMA1_ADDR_0, DMA1_ADDR_1, DMA2_ADDR_0 и DMA2_ADDR_1)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:0	BASE_ADDR	Базовый адрес страницы видеопамяти	rw	0x0

1.3.5.2.1.8.6.3 Размер кадра изображения для яркостной компоненты (Y_SIZE)

Регистр содержит информацию о вертикальном (количество строк) и горизонтальном (количество пикселов в строке) размерах изображения для яркостной компоненты.

Адрес: 0x41c.

Разрядность: 32 разряда.

Таблица 1.163 - Разряды регистра размера кадра изображения для яркостной компоненты (Y_SIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:28	RESERVED			
27:16	Y_VER_SIZE	Вертикальный размер изображения для яркостной компоненты	rw	0x0
15:12	RESERVED			
11:0	Y_HOR_SIZE	Горизонтальный размер изображения для яркостной компоненты	rw	0x0

1.3.5.2.1.8.6.4 Размер кадра изображения для цветоразностной компоненты (C_SIZE)

Регистр содержит информацию о вертикальном (количество строк) и горизонтальном (количество точек в строке) размерах изображения для цветоразностной компоненты.

Адрес: 0x420.

Разрядность: 32 разряда.

Таблица 1.164 - Разряды регистра кадра изображения для цветоразностной компоненты (C_SIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:28	RESERVED			
27:16	C_VER_SIZE	Вертикальный размер изображения для цветоразностной компоненты	rw	0x0
15:12	RESERVED			
11:0	C_HOR_SIZE	Горизонтальный размер изображения для цветоразностной компоненты	rw	0x0

1.3.5.2.1.8.6.5 Полный размер строки изображения в памяти (FULL_LINE_SIZE)

Данный регистр содержит информацию о полном размере строки изображения в памяти.
Адрес: 0x424.
Разрядность: 32 разряда.

Таблица 1.165 - Разряды регистра полного размера строки изображения в памяти (FULL_LINE_SIZE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:28	RESERVED			
27:16	C_FULL_LINE_SIZE	Полный размер строки изображения в памяти для цветоразностной компоненты	rw	0x0
15:12	RESERVED			
11:0	Y_FULL_LINE_SIZE	Полный размер строки изображения в памяти для яркостной компоненты	rw	0x0

1.3.5.2.1.8.6.6 Входной формат данных (MODE)

Данный регистр содержит в себе информацию о стандарте телевизионной развёртки и входном формате данных.
Адрес: 0x428.
Разрядность: 32 разряда.

Таблица 1.166 - Разряды регистра входного формата данных (MODE)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:5	RESERVED			
4	DEC_ENABLE	Регистр настраивает работу устройства по синхронизирующей информации: 0 – устройство работает по сигналам внешней синхронизации (hsync, vsync, field и data_enable); 1– по сигналам внутренней синхронизации (синхрокодам EAV и SAV).	rw	0x0
3	REPLICATION	Данный регистр содержит информацию о дублировании данных на видео шине: 0 – дублирования нет; 1 – дублирование (SDTV).	rw	0x0
2:1	ACTIVE_BUS	Количество активных шин, по которым передаётся изображение: 1 – при передаче по линии dv0; 2 – при передаче данных по линиям dv0 и dv1; 3 – при передаче данных по линиям dv0, dv1 и dv2.	rw	0x0
0	FORMAT_DATA	Данный регистр содержит информацию о формате данных, поступающих на вход УЗВИ: 0 – YCbCr; 1 – RGB.	rw	0x0

1.3.5.2.1.8.6.7 Выходной формат данных (LOC_DATA)

Данный регистр содержит информацию о расположении данных в памяти.
Адрес: 0x42c.
Разрядность: 32 разряда.

Таблица 1.167 - Разряды регистра выходного формата данных (LOC_DATA)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:4	RESERVED			
3	FORMAT_DATA	Данный регистр содержит информацию о формате данных (только для YCbCr): 0 – YCbCr 4:2:2; 1 – YCbCr 4:4:4.	rw	0x0
2:1	QUANTITY_AREA	Данный регистр содержит информацию о количестве областей памяти, использующихся для записи: 1 – для формата ARGB 8888; 2 – для формата YCbCr 4:2:2; 3 – для формата YCbCr 4:4:4, YCbCr 4:2:2 и RGB 888.	rw	0x0
0	COLOR_MODEL	Данный регистр содержит информацию о цветовой модели записываемых данных: 0 – YCbCr; 1 – RGB.	rw	0x0

1.3.5.2.1.8.6.8 Регистр коэффициента прозрачности (TRANSPARENCY)

Данный регистр содержит значения коэффициента прозрачности для записи данных в память формата ARGB 8888.
Адрес: 0x430.
Разрядность: 32 разряда.

Таблица 1.168 - Разряды регистра коэффициента прозрачности (TRANSPARENCY)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:8	RESERVED			
7:0	TRANSPARENCY	Данный регистр содержит значения коэффициента прозрачности для записи данных в память формата ARGB 8888.	rw	0x0

1.3.5.2.1.8.6.9 Начальная точка захвата видеоизображения (BASE_POINT)

Данный регистр содержит координаты начальной точки захвата видеоизображения относительно начала активной части кадра видеоизображения.
Адрес: 0x434.
Разрядность: 32 разряда.

Таблица 1.169 - Разряды регистра начальной точки захвата видеоизображения относительно начала активной части кадра видеоизображения (BASE_POINT)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:28	RESERVED			
27:16	BASE_POINT_Y	Вертикальная координата точки захвата видеоизображения	rw	0x0
15:12	RESERVED			
11:0	BASE_POINT_X	Горизонтальная координата точки захвата видеоизображения	rw	0x0

1.3.5.2.1.8.6.10 Идентификационные номера DMA (DMA_ID)

Данный регистр содержит идентификационные номера для трёх каналов DMA.
Адрес: 0x438.
Разрядность: 32 разряда.

Таблица 1.170 - Разряды регистра идентификационных номеров DMA (DMA_ID)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:12	RESERVED			
11:10	DMA2_WID	Идентификационный номер Dma второго канала записи данных	rw	0x3
9: 8	DMA2_AWID	Идентификационный номер Dma второго канала записи адреса	rw	0x3
7: 6	DMA1_WID	Идентификационный номер Dma первого канала записи данных	rw	0x2
5: 4	DMA1_AWID	Идентификационный номер Dma первого канала записи адреса	rw	0x2
3: 2	DMA0_WID	Идентификационный номер Dma нулевого канала записи данных	rw	0x1
1: 0	DMA0_AWID	Идентификационный номер Dma нулевого канала записи адреса	rw	0x1

1.3.5.2.1.8.6.11 Регистр параметров контроллера ведущего на шине AXI (AXI_PARAM)

Регистр содержит конфигурационные параметры, определяющие настройку интерфейса AXI для DMA канала записи данных. Подробно назначение этих параметров описывается в спецификации на соответствующий интерфейс.

Адрес: 0x43c.
Разрядность: 32 разряда.

Таблица 1.171 - Разрядность регистра параметров контроллера ведущего на шине AXI (AXI_PARAM)

№ Разр	Имя	Название (описание)	Реж	Исх. знач.
31:28	RESERVED			
27:24	AWLEN	Размер пакета (определяет количество посылок в одном пакете)	rw	0xf
23	RESERVED			
22:20	AWSIZE	Определяет размер одной посылки (количество байт в одной посылке имеет постоянное значение - 3’b011, т.е. 64 разряда)	r	0x3
19:18	RESERVED			
17:16	AWBURST	Тип обращения. Доступен только режим обращения incr, поэтому сигнал постоянно имеет значение 2’b01	r	0x1
15:10	RESERVED			
9:8	AWLOCK	Задание режима эксклюзивного доступа (не поддерживается, всегда обнулен)	r	0x0
7:4	AWCACHE	Задание режимы работы с кеш (не поддерживается, всегда обнулен)	rw	0x0
3	RESERVED			
2:0	AWPROT	Задание режима защищенного доступа (не поддерживается, всегда обнулен)	r	0x0

1.3.5.2.1.9 GAMMA_TABLE – таблицы функции гамма-коррекции.

Запись и чтение из таблиц гамма-коррекции доступны для пользователя как регистры. В таблице Таблица 1.172 представлен список таблиц, используемых для гамма-коррекции, соответствие таблицы с компонентами цвета, подвергаемые коррекции и адресное пространство таблицы.

Таблица 1.172 - Список таблиц гамма-коррекции

Линия	Цветовая модель		Начальный адрес таблицы	Конечный адрес таблицы
	YCbCr	RGB		
dv0	Y	G	0xd00	0xdff
dv1	Cr	R	0xe00	0xeff
dv2	Cb	B	0xf00	0xfff

Более подробно реализация функции гамма-коррекции расписана в пункте 1.3.5.2.2.1.6.

1.3.5.2.2 Функциональное описание.

1.3.5.2.2.1 Подготовка устройства захвата видеоизображения к использованию

Для приведения устройства захвата видеоизображения в рабочее состояние необходимо после включения произвести его сброс и начальную инициализацию.

1.3.5.2.2.1.1 Сброс устройства захвата видеоизображения

Сброс устройства захвата видеоизображения может быть либо программным, либо аппаратным. Аппаратный сброс осуществляется посредством подачи низкого уровня сигнала на порт sys_reset_i. Программный сброс осуществляется путем установки в единицу регистра PR_RESET. После прохождения операции сброса значение регистра программного сброса PR_RESET автоматически восстановится в ноль.

Следует отметить, что, как при аппаратном, так и при программном сбросах значения некоторых конфигурационных регистров устройства захвата видеоизображения не будут обнулены, а будут установлены в начальное положение.

1.3.5.2.2.1.2 Начальная инициализация

Инициализация: после включения питания и прохождения операции сброса, все блоки устройства захвата видеоизображения будут находиться в исходном рабочем состоянии. Следующий шаг - конфигурирование устройства на требуемый режим работы, для этого требуется записать соответствующие значения во все регистры устройства, необходимые для заданного режима работы.

Перед включением устройства необходимо задать значения базовых адресов плоскостей памяти, в которые будет вестись запись видеоданных. В зависимости от расположения данных в памяти (количества плоскостей), может использоваться от одной до трёх пар базовых адресов (для «текущего» и «следующего» кадров). Инициализация регистров DMA0_ADDR_0 и DMA0_ADDR_1, содержащих базовые адреса нулевой области, проводится для любого типа расположения данных в памяти. Инициализация регистров DMA1_ADDR_0 и DMA1_ADDR_1, содержащих базовые адреса первой области, проводится в случае расположения данных формата YCbCr (для любого разделения данных в памяти). Инициализация регистров DMA2_ADDR_0 и DMA2_ADDR_1, содержащих базовые адреса второй плоскости, проводится только в случае расположения в памяти данных формата YCbCr с трёхплоскостным распределением данных. Все базовые адреса должны быть выровнены по границе 8 байт.

Для записи данных в память необходимо задать размеры плоскостей памяти, отведённых для записи данных. Для этого необходимо записать в регистры Y_SIZE и C_SIZE размеры изображения. Регистр размера кадра изображения для яркостной компоненты Y_SIZE содержит поле горизонтального размера кадра Y_HOR_SIZE и поле вертикального размера кадра Y_VER_SIZE. Регистр размера кадра изображения для цветоразностных компоненты C_SIZE содержит поле горизонтального размера кадра C_HOR_SIZE и поле вертикального размера кадра C_VER_SIZE. Значения, которые необходимо записать в эти регистры, зависят от физического размера изображения и содержатся в таблицах Таблица 1.173 и Таблица 1.174.

Горизонтальный размер изображения hor_size и вертикальный размер изображения ver_size в таблице ниже – реальные размеры изображения в точках.

Таблица 1.173. Размеры плоскости памяти для расположения данных формата YCbCr

Тип i/p	Формат 4:2:2/ 4:4:4	Кол-во пл-ей 2/3	Поле Y_HOR_SIZE	Поле Y_VER_SIZE	Поле C_HOR_SIZE	Поле C_VER_SIZE
i	4:2:2	2	hor_size	ver_size/2	hor_size	ver_size/2
		3	hor_size	ver_size/2	hor_size/2	ver_size/2
	4:4:4	3	hor_size	ver_size/2	hor_size	ver_size/2
p	4:2:2	2	hor_size	ver_size	hor_size	ver_size
		3	hor_size	ver_size	hor_size/2	ver_size
	4:4:4	3	hor_size	ver_size	hor_size	ver_size

Таблица 1.174. Размеры плоскости памяти для расположения данных формата RGB

Тип i/p	Формат	Кол-во пл-ей 1/3	Поле Y_HOR_SIZE	Поле Y_VER_SIZE	Поле C_HOR_SIZE	Поле C_VER_SIZE
i	ARGB 8888	1	hor_size	ver_size/2	-	-
	RGB 888	3	hor_size	ver_size/2	hor_size	ver_size/2
p	ARGB 8888	1	hor_size	ver_size	-	-
	RGB 888	3	hor_size	ver_size	hor_size	ver_size

Необходимо задать полный размер строки в памяти в регистре FULL_LINE_SIZE (должен быть кратен 8). Регистр полного размера строки изображения в памяти FULL_LINE_SIZE содержит поле полного размера строки изображения в памяти для яркостной компоненты Y_FULL_LINE_SIZE и поле полного размера строки изображения в памяти для цветоразностных компонент C_FULL_LINE_SIZE.

Примечание! При записи в память данных формата ARGB 8888 реальные размены изображения Y_HOR_SIZE и строки памяти Y_FULL_LINE_SIZE будут в 4 раза больше.

Для настройки устройства на захват телевизионной развёртки определённого формата необходимо задать формат входного потока данных в регистре MODE. Регистр MODE содержит поля, в которых задаётся количество шин, по которым ведётся приём видеоизображения, формат данных и тип синхронизации.

Для задания порядка расположения данных в памяти необходимо сконфигурировать регистр LOC_DATA.

Для разрешения записи данных в память по заданным базовым адресам необходимо разрешить переключение записи данных между страницами памяти, записав единицу в регистр BASE_SW_ENA.

1.3.5.2.2.1.3 Распознавание параметров телевизионной развёртки

Устройство захвата видеоизображения имеет функцию автоматического распознавания параметров телевизионной развёртки. Для её включения необходимо записать единицу в регистр REC_ENABLE. По прошествии двух полных кадров устройство сгенерирует запрос прерывания по событию INT_DONE. После этого параметры развёртки можно считать из регистров FRAME_SIZE и FRAME_PARAM. На основании полученных данных можно сконфигурировать устройство.

Примечание! При передаче видеоизображения формата SDTV значение горизонтального размера изображения в поле HOR_SIZE регистра FRAME_SIZE будет больше в 2 раза реального горизонтального размера изображения.

1.3.5.2.2.1.4 Обработка видеоизображения

Устройство видеозахвата поддерживает функции гамма-коррекции и преобразование цвета из одной цветовой модели в другую. На рисунке 1.96 изображена схема алгоритма обработки видеоизображения.

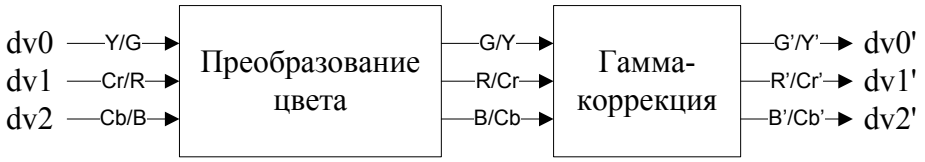


Рисунок 1.96 - Схема алгоритма обработки видеоизображения

Внутренним форматом данных в устройстве захвата видеоизображения является формат YCbCr 4:4:4 или RGB 888. Данные передаются по трём линиям. В таблице 1.145 содержится распределение компонент цвета по внутренним линиям устройства захвата видеоизображения.

Таблица 1.175 - Распределение компонент цвета по внутренним линиям устройства захвата видеоизображения

Линия	Цветовая модель	
	YCbCr	RGB
dv0	Y	G
dv1	Cr	R
dv2	Cb	B

В пункте 1.3.5.2.2.1.5 подробно описана реализация функции преобразования цвета из одной цветовой модели в другую. В пункте 1.3.5.2.2.1.6 подробно описана реализация функции гамма-коррекции изображения.

1.3.5.2.2.1.5 Преобразование цветовой модели

Устройство захвата видеоизображения производит математические преобразования по формулам:

$$dv0' = \frac{C_0^0 + C_1^0 * dv0 + C_2^0 * dv1 + C_2^0 * dv2}{256}$$

$$dv1' = \frac{C_0^1 + C_1^1 * dv0 + C_2^1 * dv1 + C_2^1 * dv2}{256}$$

$$dv2' = \frac{C_0^2 + C_1^2 * dv0 + C_2^2 * dv1 + C_2^2 * dv2}{256}$$

Где dv0, dv1 и dv2 – линии, по которым передаются компоненты цвета до преобразования, а dv0', dv1' и dv2' – линии, по которым передаются компоненты цвета после преобразования.

Для осуществления преобразования представления цвета из цветовой модели YCbCr в RGB и наоборот для стандартов SDTV и HDTV и осуществления преобразования представления цвета из SDTV в HDTV и наоборот для формата данных YCbCr необходимо записать в регистры устройства видеозахвата значения, указанные в таблице 1.146.

Таблица 1.176 - Коэффициенты для осуществления преобразования цвета

Коэф	Регистр	Преобразования					
		SDTV		HDTV		YCbCr	
		Из YCbCr в RGB	Из RGB в YCbCr	Из YCbCr в RGB	Из RGB в YCbCr	Из SDTV в HDTV	Из HDTV в SDTV
C_0^0	C_0_0	+34656	+16*256	+19680	+16*256	+10624	-9472
C_1^0	C_0_1	+298	+129	+298	+157	+256	+256
C_2^0	C_0_2	-208	+66	-136	+47	-53	+49
C_3^0	C_0_3	-100	+25	-55	+16	-30	+25
C_0^1	C_1_0	-57120	+128*256	-63520	+128*256	-3328	+2944
C_1^1	C_1_1	+298	-94	+298	-102	0	0
C_2^1	C_1_2	+409	+112	+459	+112	+263	+252
C_3^1	C_1_3	0	-18	0	-10	+19	-19
C_0^2	C_2_0	-70944	+128*256	-74016	+128*256	-4352	+3968
C_1^2	C_2_1	+298	-74	+298	-87	0	0
C_2^2	C_2_2	0	-38	0	-26	+29	-28
C_3^2	C_2_3	+517	+112	+541	+112	+261	+253

Компоненты R, G и B принимают значения в диапазоне от 0 до 255. Компонента Y принимает значения в диапазоне от 16 до 235. Компоненты Cb и Cr принимают значения от 16 до 240.

Для ограничения выходных значений необходимо записать верхние и нижние границы диапазона в соответствующие регистры. Регистр CH0_RANGE ограничивает данные на шине dv0, регистр CH1_RANGE ограничивает данные на шине dv1, регистр CH2_RANGE ограничивает данные на шине dv2.

Для разрешения преобразования необходимо записать единицу в регистр CONV_ENABLE.

1.3.5.2.2.1.6 Гамма-коррекция изображения

Гамма-коррекция – нелинейная функция предискажения яркости. Функция гамма-коррекции описывается формулами ниже. Скорректированное значение обозначается знаком «’».

$R' = R1/\gamma$

$G' = G1/\gamma$

$B' = B1/\gamma$

Где R, G и B – компоненты цвета до преобразования, нормированные в диапазоне [0…1], а R’, G’ и B’ – компоненты цвета после преобразования, нормированные в диапазоне [0…1].

Для данных формата YCbCr гамма-коррекции подвергается только яркостная компонента Y, цветоразностные компоненты Cb’ и Cr’ остаются без изменения и равны Cb и Cr соответственно.

Устройство захвата видеоизображения поддерживает функцию гамма-коррекции изображения табличным способом. Таблица имеет 256 ячеек. Каждая ячейка в таблице имеет свой адрес от 0x00 до 0xff. Адрес ячейки таблицы является аргументом функции, а значение, записанное в эту ячейку – значением функции. Для проведения гамма-коррекции изображения необходимо записать значения функции в соответствующие ячейки таблицы, значение регистра GAM_ENABLE в этот момент должно быть равно нулю.

Список таблиц гамма коррекции, соответствующие им компоненты цвета и базовые адреса таблиц содержаться в таблице 1.147.

Таблица 1.177 - Список таблиц гамма коррекции, соответствующие им компоненты цвета и базовые адреса таблиц.

Линия	Цветовая модель		Базовый адрес таблицы
	YCbCr	RGB	
dv0	Y	G	0xd00
dv1	Cr	R	0xe00
dv2	Cb	B	0xf00

Для включения гамма-коррекции изображения необходимо записать единицу в регистр GAM_ENABLE.

1.3.5.2.2.2 Управление устройством захвата видеоизображения

1.3.5.2.2.2.1 Переключение между двумя страницами памяти

При работе устройство захвата видеоизображения переключается между двумя страницами памяти («текущая» и «следующая» плоскости). При включении устройства «текущей» областью считается область, чьи базовые адреса страниц памяти задаются регистрами с индексом «0»: DMA0_ADDR_0, DMA1_ADDR_0 и DMA2_ADDR_0.

Возникновение прерывания по событию END_WRITE сигнализирует об окончании записи данных в «текущую» плоскость и переключение на «следующую». Запись следующего кадра будет проводится в плоскость с базовым адресом, имеющим индекс «1», данная плоскость становится «текущей». Область с базовым адресом, имеющем индекс «0» становится «следующей». При записи данных в «текущую» область можно менять базовый адрес «следующей» области.

Для запрета записи данных в память по заданным базовым адресам необходимо запретить переключение записи данных между страницами памяти, записав ноль в регистр BASE_SW_ENA.

1.3.5.2.2.2.2 Включение

Для запуска работы устройства, после начальной инициализации устройства, нужно установить значение регистра ENABLE в единицу. Устройство начнёт захватывать и записывать видеоизображение с начала кадра (для чересстрочного режима с начала чётного кадра).

1.3.5.2.2.2.3 Выключение

Выключение устройства проводится путём программного сброса в любой момент времени. Программный сброс осуществляется путем установки в единицу значения регистра PR_RESET. После прохождения операции сброса значение регистра программного сброса PR_RESET автоматически восстановится в ноль. В результате сброса устройство приходит в начальное неактивное состояние.

1.3.5.2.2.2.4 Настройка генератора запроса прерывания

Для нормальной работы устройства захвата видеоизображения надо разрешить прерывания, необходимые для функционирования устройства в заданном режиме, для этого надо записать единицу в соответствующие биты регистра INT_MASK.

1.3.5.2.2.3 Обработка прерываний

Возникновение прерывания по событиям CH0_OVERFLOW, CH1_OVERFLOW и CH2_OVERFLOW сигнализирует о переполнении буфера, т.е. запрос записи в полный буфер с потерей данных. Необходимо осуществить программный сброс устройства для последующего перезапуска.

Возникновение прерывания по событию INT_ST_ERROR сигнализирует об изменении количества активных шин, по которым ведётся передача данных. Необходимо осуществить программный сброс устройства для последующего перезапуска.

Возникновение прерывания по событию INT_VS_ERROR сигнализирует об изменении количества линий изображения во входном потоке данных. Необходимо осуществить программный сброс устройства для последующего перезапуска.

Возникновение прерывания по событию INT_HS_ERROR сигнализирует об изменении количества точек в соседних линиях изображения. Необходимо осуществить программный сброс устройства для последующего перезапуска.

Возникновение прерывания по событию INT_UNC_ERR сигнализирует о детектировании непоправимой ошибки в синхронизирующей посылке (SAV или EAV). Необходимо осуществить программный сброс устройства для последующего перезапуска.

Возникновение прерывания по событиям DMA0_ERROR, DMA1_ERROR или DMA2_ERROR сигнализирует о получении ответа от ведомого устройства на шине AXI «ошибка».

1.3.5.2.2.4 Тестирование генератора прерывания

В устройстве захвата видеоизображения поддерживается функция тестирования генератора запроса прерывания. Для генерации сигнала запроса прерывания необходимо записать единицу в регистр TEST_INT. При разрешении прерывания по событию TEST_INT сигнал запроса прерывания перейдёт в активное состояние.

1.3.6 Подсистема кодирования и декодирования

Подсистема кодирования и декодирования предназначена для обеспечения кодирования, декодирования видеосигналов в режиме реального времени или же проводиться одновременное кодирование и декодирование нескольких потоков видеоизображений, имеющих разные форматы, с разным разрешением.

На рисунке ниже приведена структурная схема подсистемы. Подсистема состоит из двух типов устройств:

- VIDEO CODER/DECODER (кодер видеопотока);
- JPEG CODER (кодер JPEG).

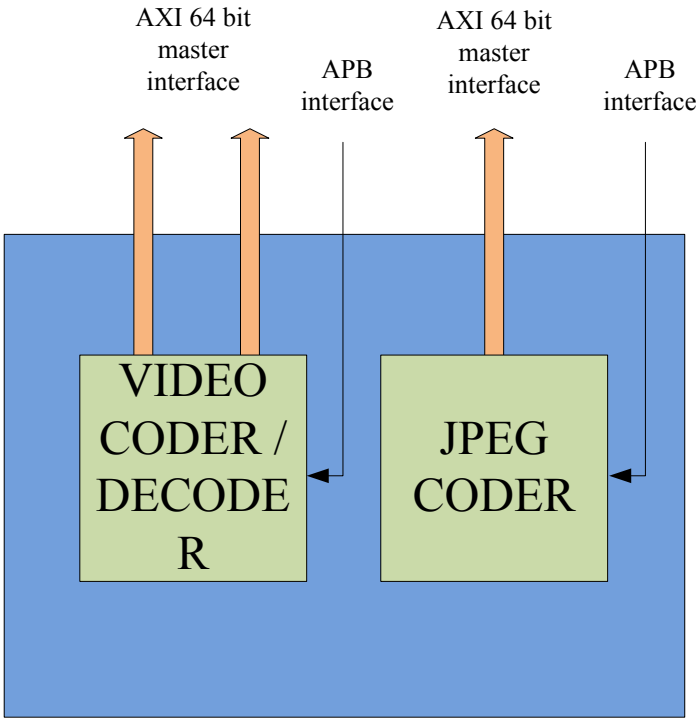


Рисунок 1.97 – Структурная схема подсистемы кодирования и декодирования

1.3.6.1 Кодер видеопотока (описание)

1.3.6.1.1 Функциональное описание

1.3.6.1.1.1 Функциональные возможности

Кодек CODA980 (блок обработки видеосигналов (далее VPU)) компании “Chips&Media” представляет собой мультистандартное решение IP-видео формата “full HD” (сверхвысокой четкости), устанавливаемое в мультимедийных устройствах потребительского уровня: телевизорах, телевизионных приставках и DVD-плеерах, поддерживающих формат HD. Он позволяет декодировать сжатое динамическое изображение в форматах H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-1/2, MPEG-4 SP/ASP, H.263P3, DivX/XviD, AVS (вкл.

AVS+), RV-8/9/10, VP8, MVC и Theora до разрешения уровня HD (4096x2208). Данный кодек может кодировать видеосигналы формата H.264, MPEG-4 и H.263 до уровня высокой четкости (далее HD) - 4096x2208. В блоке VPU может выполняться кодирование, декодирование видеосигналов в режиме реального времени или же проводиться одновременное кодирование и декодирование нескольких потоков видеоизображений, имеющих разные форматы, с разным разрешением.

В блок VPU входит 16-разрядный цифровой сигнальный процессор (далее DSP), являющийся собственностью компании “Chips&Media”, который называется BIT-процессором. BIT-процессор взаимодействует с центральным процессором хост-компьютера через главный интерфейс и управляет другими субблоками VPU. Для обеспечения работы центрального процессора хост-компьютера не требуется большого ресурса (менее 1 миллиона команд в секунду), потому что все основные функции, такие как разбор битового потока, управление субблоками видеоустройства и обеспечение устойчивости к ошибкам, реализуются в BIT-процессоре. Кроме того, он оптимально рассчитан на работу сразу с несколькими субблоками, используемыми для обработки видеосигналов, что обеспечивает существенную экономию мощности и использование меньшего количества транзисторных переключателей.

Блок VPU соединяется с центральным процессором хост-компьютера через 32-разрядную расширенную шину подключения периферийных устройств (APB) AMBA3 для системного управления, а через 64-разрядную шину дополнительного расширяемого интерфейса (далее AXI) AMBA3 для передачи данных. Существует две 64-разрядные шины AXI: основная и вспомогательная. Для достижения высокого уровня производительности вспомогательную шину можно подключить к системе памяти на кристалле.

1.3.6.1.2 Интерфейс системного уровня

1.3.6.1.2.1 Формат входных и выходных данных

1.3.6.1.2.1.1 Битовый поток

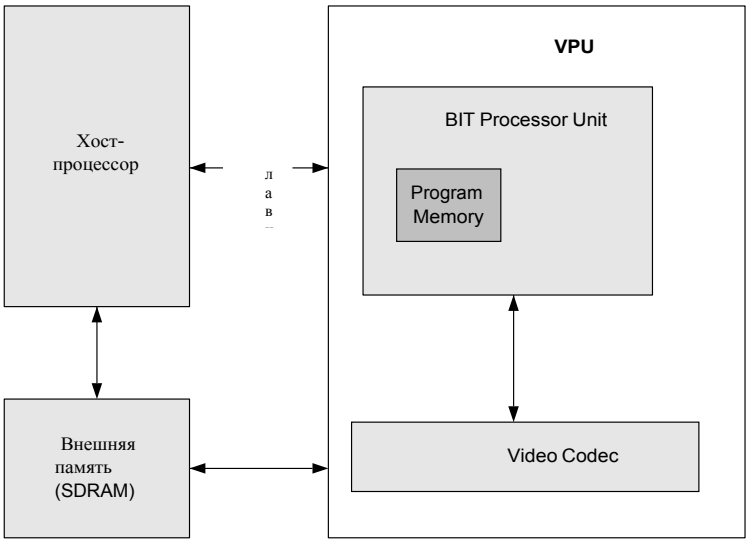
1.3.6.1.2.1.1.1 Форматы буфера битовых потоков

В буфере(ах) битовых потоков хранятся потоки битов, записанные процессором внешнего хост-компьютера или аппаратными модулями. Если пользователь конфигурирует блок VPU для декодирования нескольких битовых потоков, то каждому из этих потоков должен быть назначен отдельный буфер, а в VPU должны поступить данные о базовом адресе и размере каждого из этих буферов битовых потоков.

Внешний буфер битовых потоков находится в синхронной динамической памяти (далее SDRAM) и представляет собой кольцевой буфер. Исходный адрес кольцевого буфера и его размер должны регистрироваться в хост-компьютере. Текущий адрес чтения или записи кольцевого буфера автоматически зацикливается. При декодировании хост-компьютер сначала записывает подлежащий декодированию битовый поток, а затем BIT-процессор считывает этот битовый поток. При этом может возникнуть ситуация переполнения или недозагрузки буфера. И в этом случае декодирование станет невозможным. Во избежание подобной ситуации хост-компьютер и BIT-процессор должны обмениваться данными о чтении/записи текущих битовых потоков.

В BIT-процессоре также есть внутренний буфер битовых потоков. Он используется как кэш-память между ядром BIT-процессора и внешним буфером битовых потоков в памяти SDRAM. При декодировании BIT-процессор считывает данные битового потока из внешнего буфера в памяти SDRAM и помещает их во внутренний буфер битовых потоков. Для большей эффективности во внутреннем буфере (строго) выполняется операция разбора. В случае кодирования закодированные данные битового потока помещаются на хранение сначала во внутренний буфер, а позже отдельными блоками записываются во внешний буфер.

1.3.6.1.2.1.1.2 Схема взаимодействия между хост-компьютером и блоком VPU



VPU – Блок VPU
BIT Processor Unit – BIT-процессор
Program Memory – Программная память
Video Codec – Видео-кодек

Рисунок 1.98- Программная память BIT-процессора

Согласно рисунку 1.98 - “Программная память BIT-процессора” в BIT-процессоре находится программная память, необходимая для выполнения BIT-программы. На представленной схеме показан процесс загрузки BIT-программы в программную память.

Хост-процессор загружает микропрограммное обеспечение BIT-процессора во внешнюю память (SDRAM).

Затем хост-процессор передает некоторые из микропрограмм BIT-процессора, используемых для разных стандартов, в программную память.

На этапе инициализации в блоке VPU BIT-процессор определяет то, какое стандартное приложение необходимо использовать.

По окончании инициализации BIT-процессор выгружает коды соответствующих стандартов из внешней памяти (SDRAM) и загружает их в программную память.

1.3.6.1.2.1.2 Буфер кадра

1.3.6.1.2.1.2.1 Карты распределения памяти

Существует несколько типов карт распределения макроблоков буфера кадра:

Тип 0: Линейная карта распределения кадра

Тип 1: Тайловая карта распределения кадра в порядке вертикального сканирования

Тип 2: Тайловая карта распределения кадра в порядке горизонтального сканирования

Тип 3: Тайловая карта распределения полукадров в порядке вертикального сканирования

Тип 4: Тайловая карта распределения кадра/полукадра в порядке вертикального сканирования

Тип 5: Тайловая карта распределения макроблоков кадра в порядке растрового сканирования

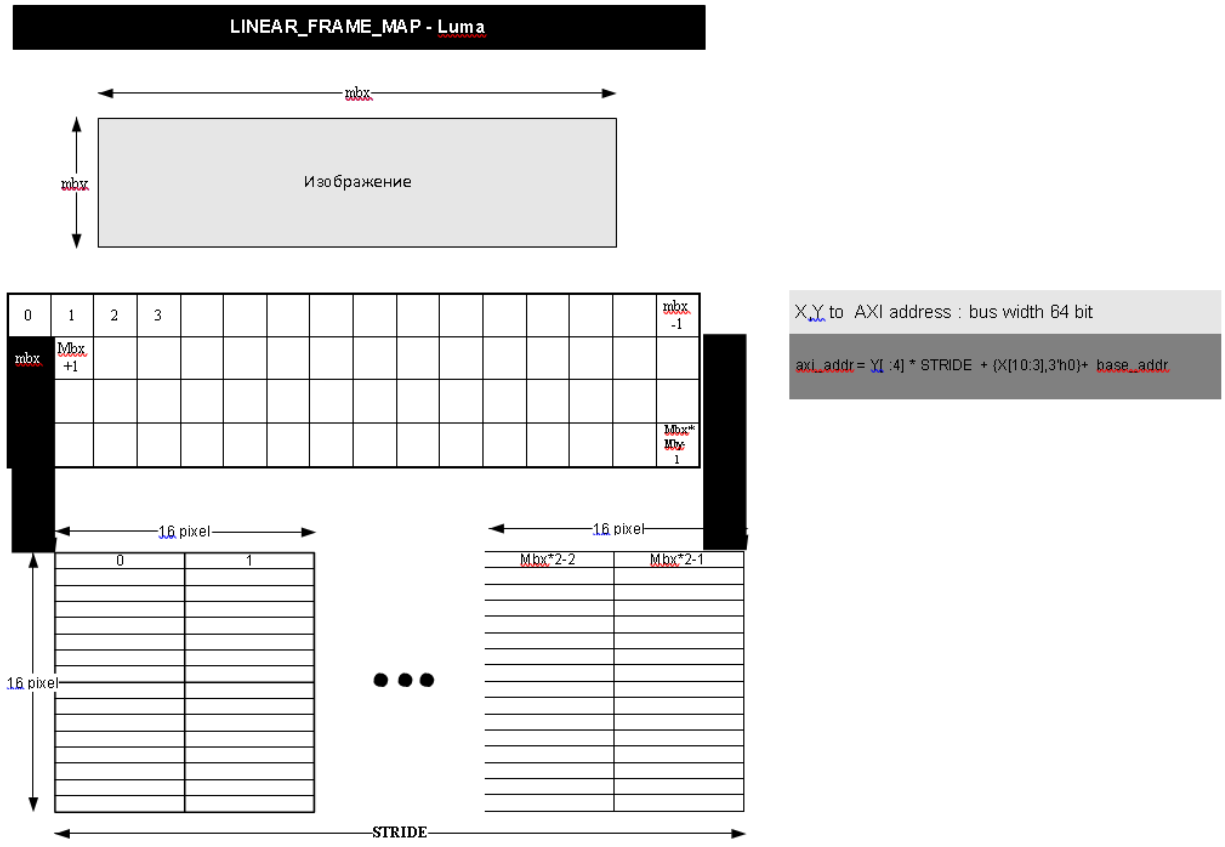
Тип 6: Тайловая карта распределения полукадров в порядке растрового сканирования

В случае линейной карты распределения каждый пиксель считывается или записывается в буфер в порядке растрового сканирования кадра. Однако блоку VPU требуются пиксели в виде макроблоков (далее MB), а такой адресный доступ не может обеспечиваться в физической памяти на постоянной основе.

В то же время на тайловой карте распределения вся область памяти логически разбивается на определенное число сегментов, имеющих форму изразца (тайла). Тайлом может быть модуль доступа, в котором все адреса выстроены в виде последовательности и обладают эффективной для доступа структурой. Размер тайла можно конфигурировать в соответствии с приложением. Это может быть один или несколько макроблоков, адрес(а) которых принадлежат к одному тому же ряду и банку адресов, и которые являются последовательными адресами столбцов в памяти SDRAM.

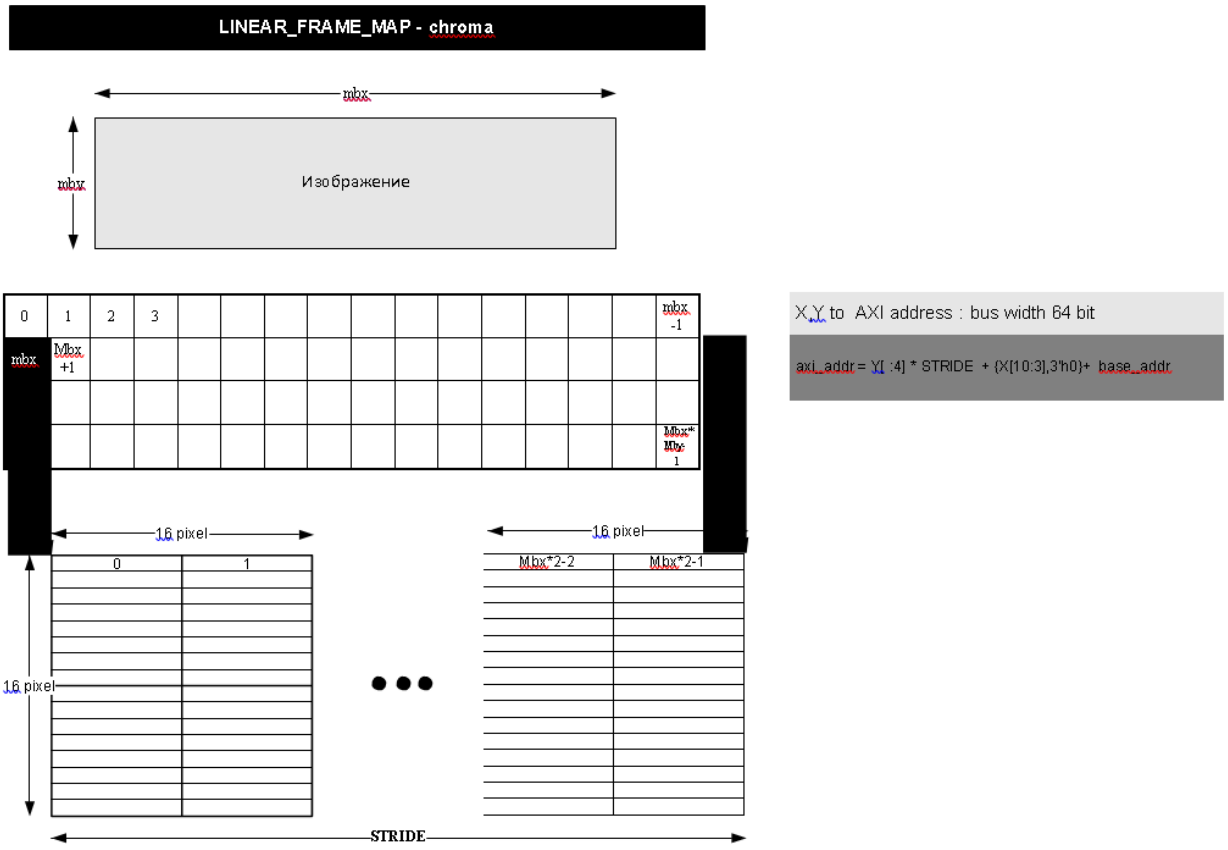
Типы карт распределения буфера кадра проиллюстрированы на следующих рисунках:

Тип 0: Линейная карта распределения кадра (режим нормальной адресации)



16 pixel – 16 пикселей
16 pixel – 16 пикселей
16 pixel – 16 пикселей
STRIDE – ШАГ ПО ИНДЕКСУ
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита

Рисунок 1.98 - Линейная карта распределения кадра (по компоненте яркости)



16 pixel – 16 пикселей
16 pixel – 16 пикселей
16 pixel – 16 пикселей
STRIDE – ШАГ ПО ИНДЕКСУ
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита

Рисунок 1.99 - Линейная карта распределения кадра (по компоненте цветности)

Тип 1: Тайловая карта распределения кадра в порядке вертикального сканирования

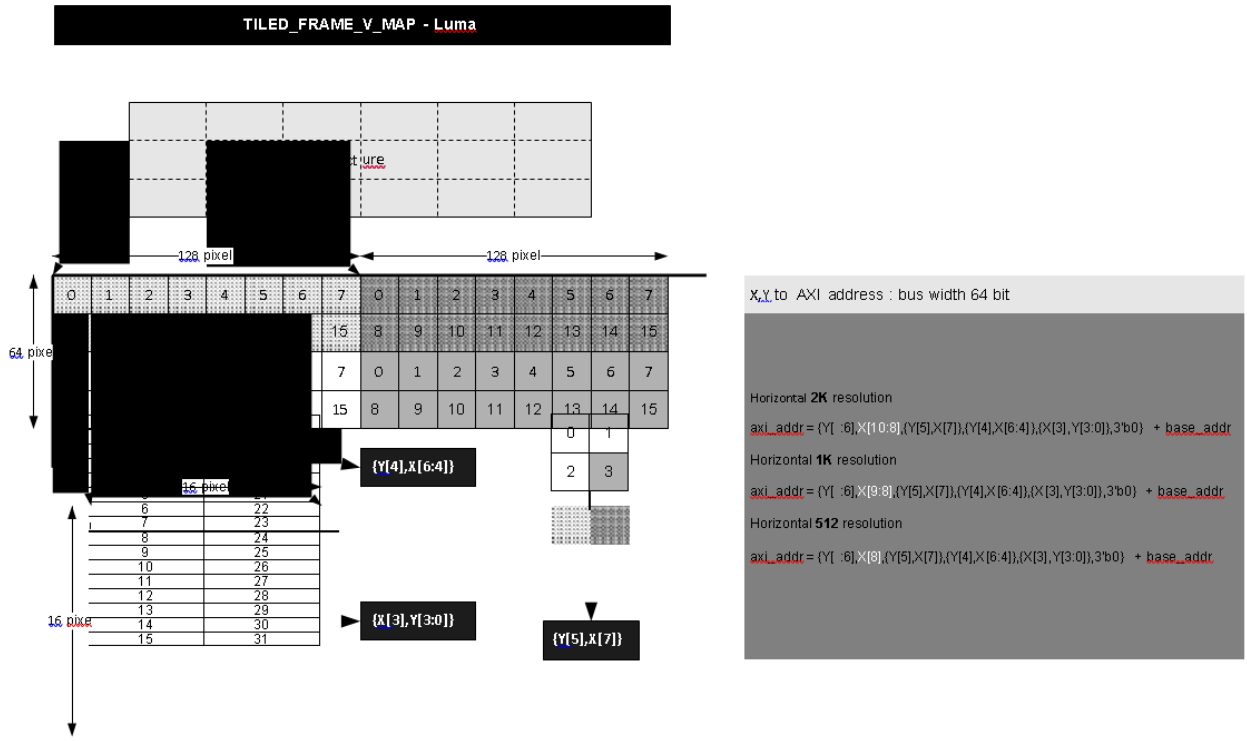


Рисунок 1.100 - Тайловая карта распределения кадра в порядке вертикального сканирования (по компоненте яркости)

Picture – Изображение
128 pixel – 128 пикселей
128 pixel – 128 пикселей
64 pixel – 64 пикселя
16 pixel – 16 пикселей
16 pixel – 16 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

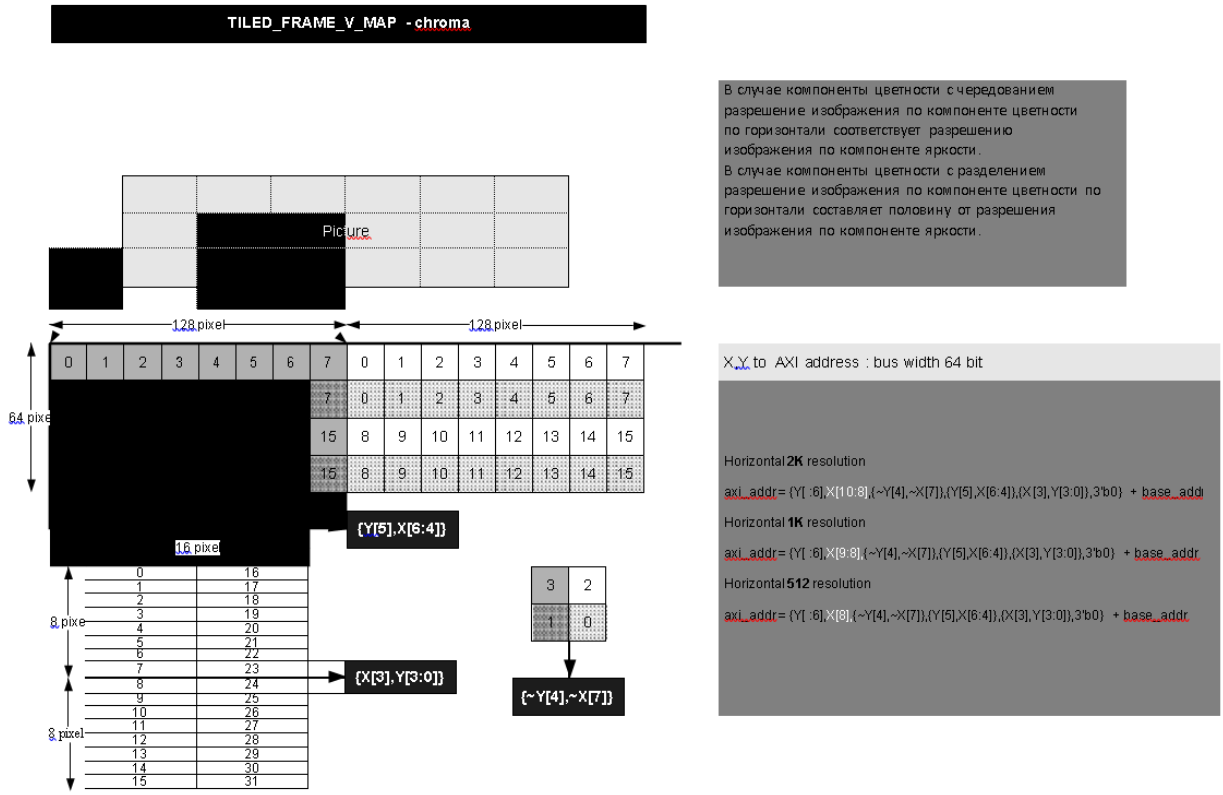


Рисунок 1.101 - Тайловая карта распределения кадра в порядке вертикального сканирования (по компоненте цветности)

Picture – Изображение
128 pixel – 128 пикселей
128 pixel – 128 пикселей
64 pixel – 64 пикселя
16 pixel – 16 пикселей
8 pixel – 8 пикселей
8 pixel – 8 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

Тип 2: Тайловая карта распределения кадра в порядке горизонтального сканирования

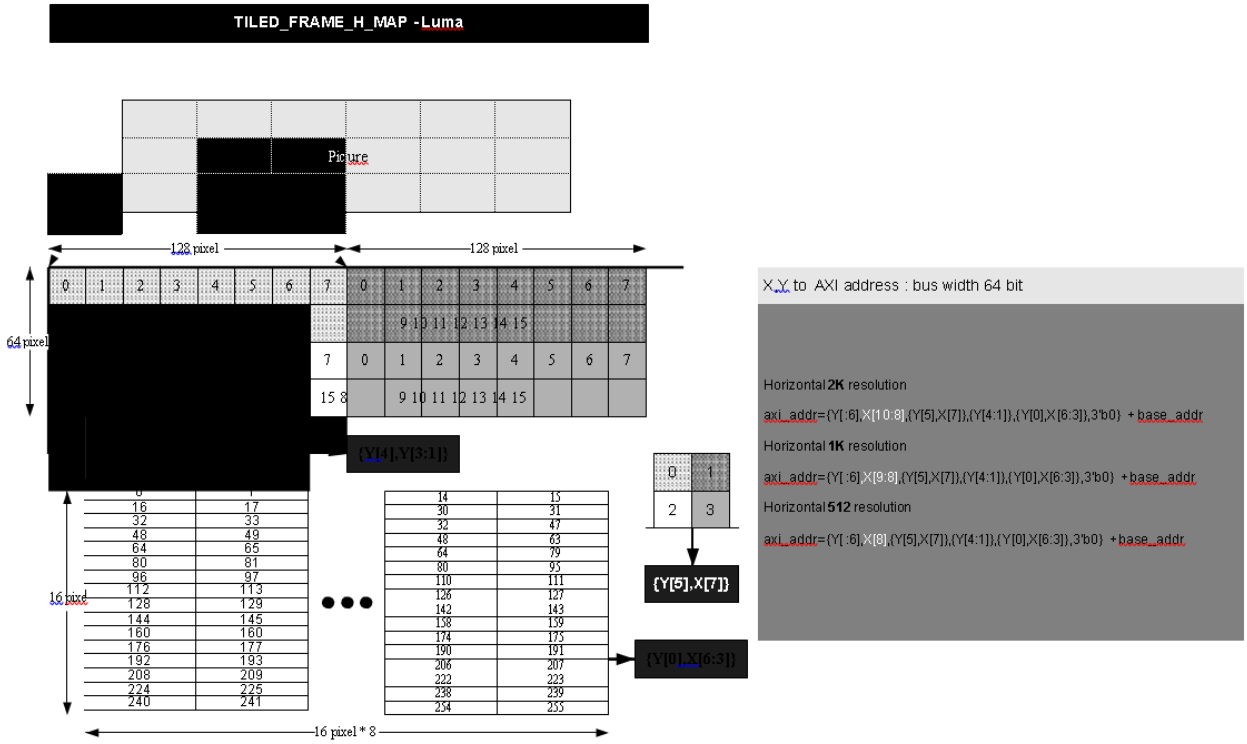


Рисунок 1.102 - Тайловая карта распределения кадра в порядке горизонтального сканирования (по компоненте яркости)

Picture – Изображение
128 pixel – 128 пикселей
128 pixel – 128 пикселей
64 pixel – 64 пикселя
16 pixel – 16 пикселей
16 pixel*8 – 16*8 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

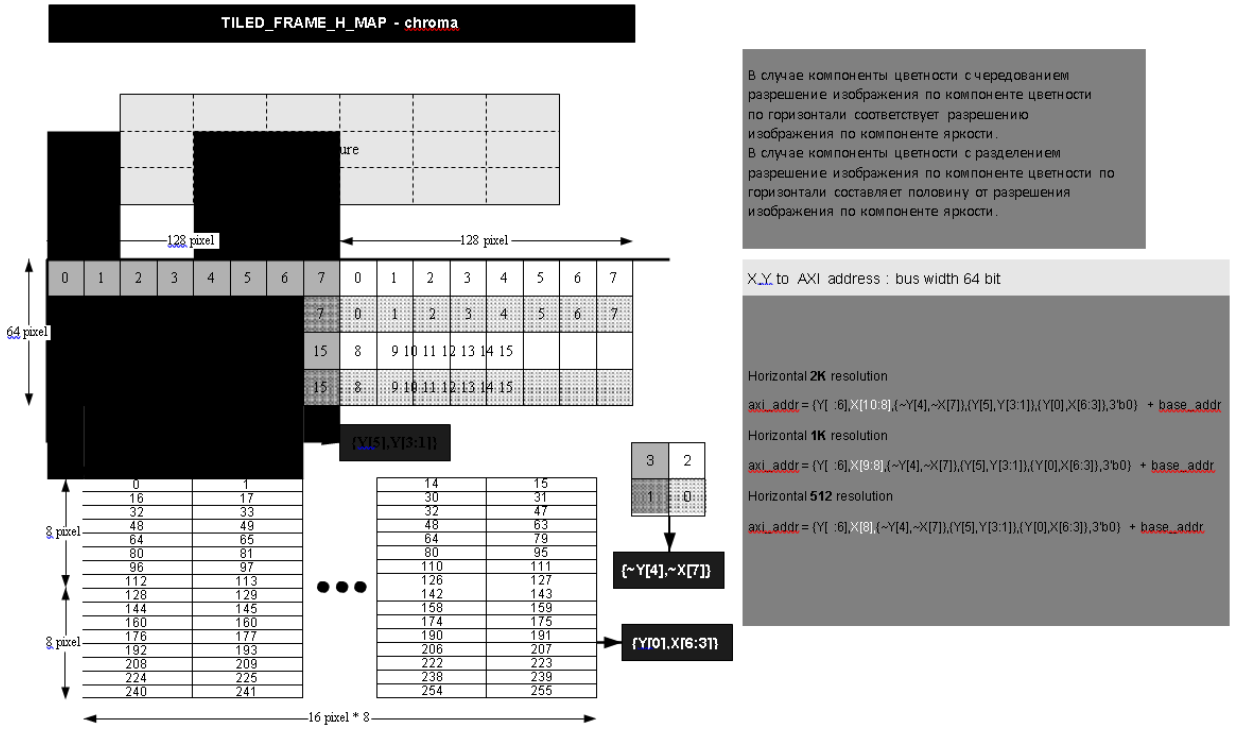


Рисунок 1.103 - Тайловая карта распределения кадра в порядке горизонтального сканирования (по компоненте цветности)

Picture – Изображение
128 pixel – 128 пикселей
128 pixel –128 пикселей
64 pixel – 64 пикселя
16 pixel – 16 пикселей
8 pixel – 8 пикселей
8 pixel – 8 пикселей
16 pixel*8 – 16*8 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

Тип 3: Тайловая карта распределения полукадров в порядке вертикального сканирования

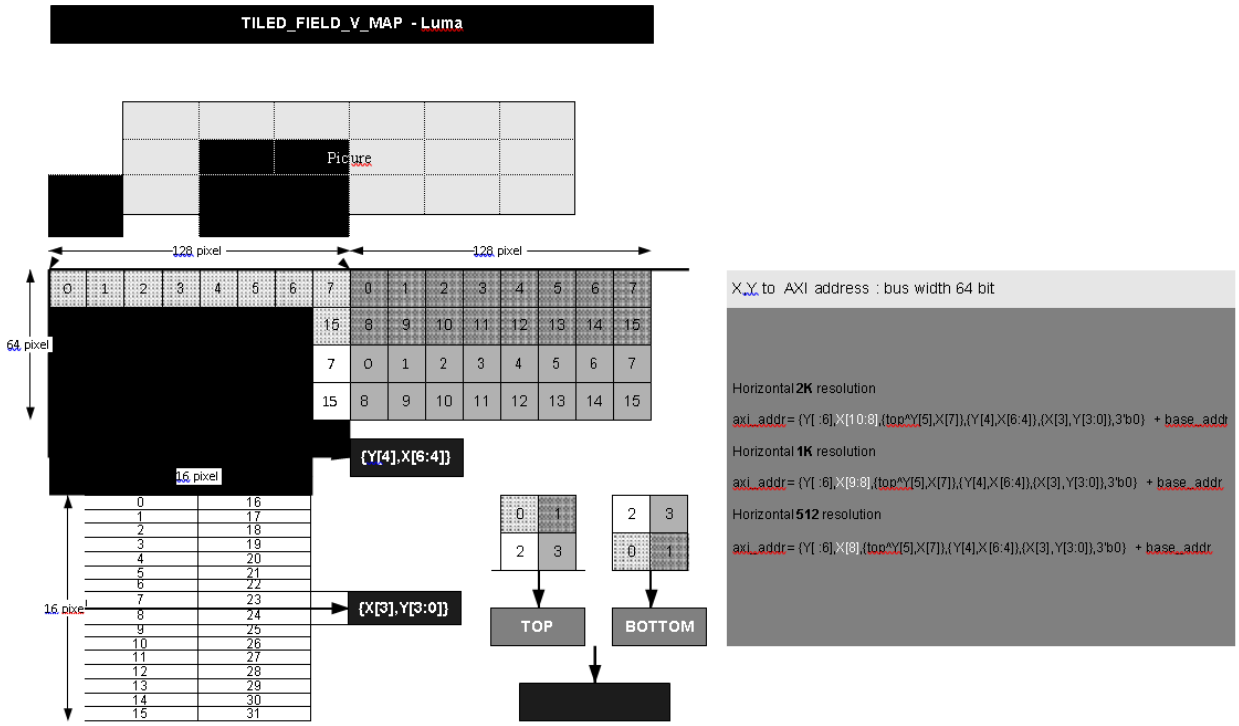


Рисунок 1.104 – Тайловая карта распределения макроблоков полукадра в порядке вертикального сканирования (по компоненте яркости)

Picture – Изображение
128 pixel – 128 пикселей
128 pixel – 128 пикселей
64 pixel – 64 пикселя
16 pixel – 16 пикселей
16 pixel – 16 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

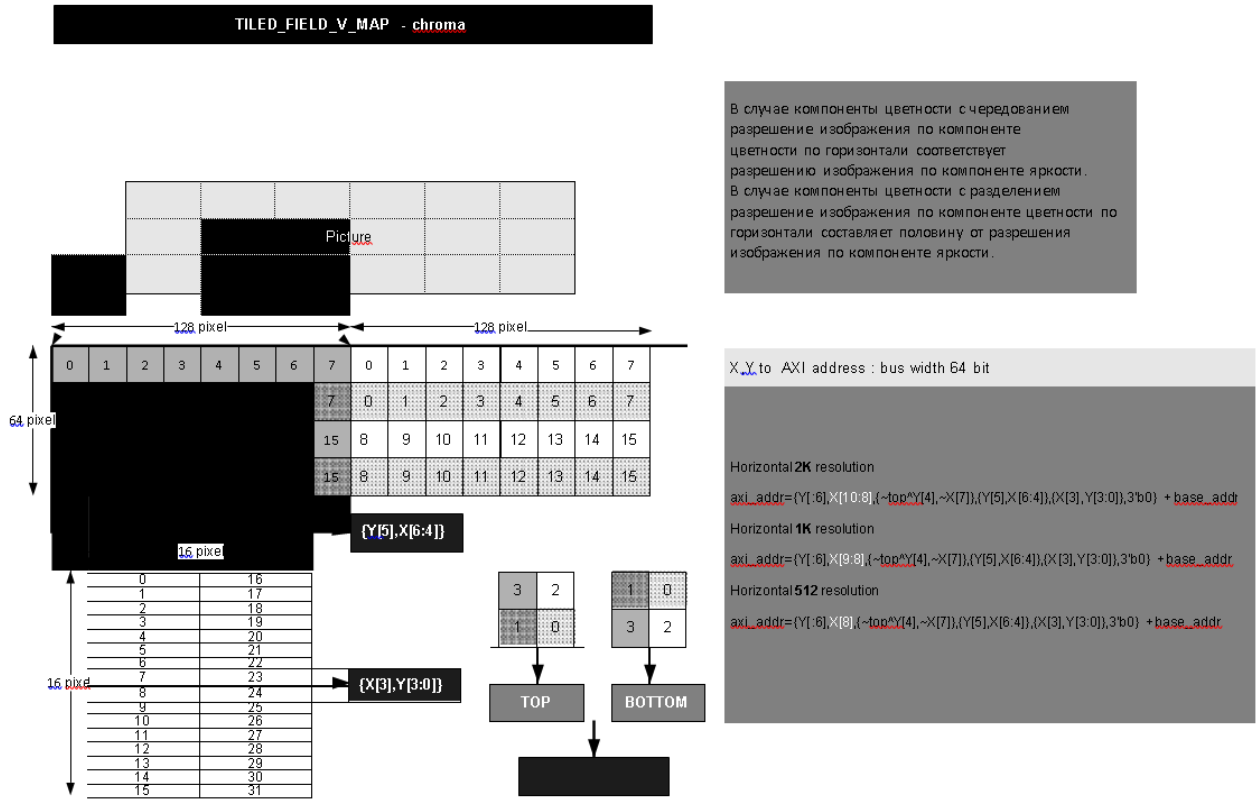


Рисунок 1.105 - Тайловая карта распределения полукадров в порядке вертикального сканирования (по компоненте цветности)

Picture – Изображение
128 pixel – 128 пикселей
128 pixel – 128 пикселей
64 pixel – 64 пикселя
16 pixel – 16 пикселей
16 pixel –16 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

Тип 4: Тайловая карта распределения кадра/полукадра в порядке вертикального сканирования

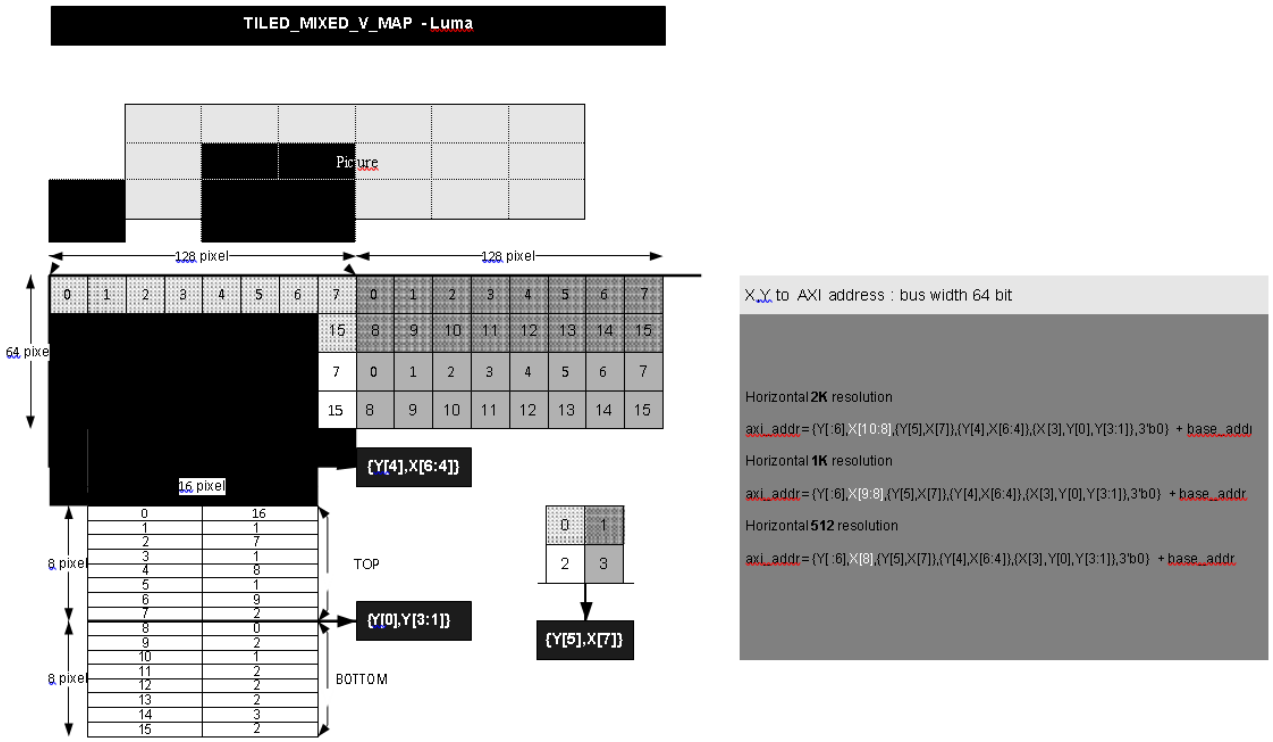
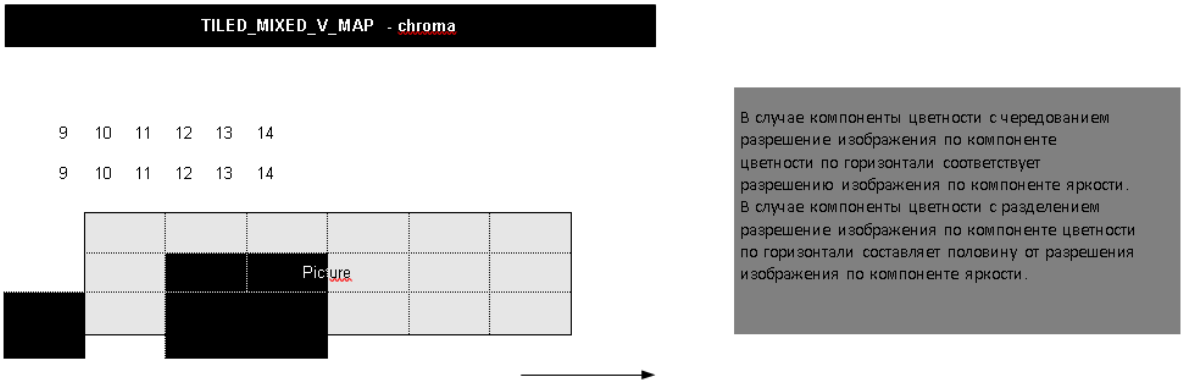


Рисунок 1.106 – Тайловая карта распределения кадра/полукадра в порядке вертикального сканирования (по компоненте яркости)

Picture – Изображение
128 pixel – 128 пикселей
128 pixel –128 пикселей
64 pixel – 64 пикселя
16 pixel – 16 пикселей
8 pixel – 8 пикселей
8 pixel – 8 пикселей
TOP – ВЕРХ
BOTTOM - НИЗ
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали



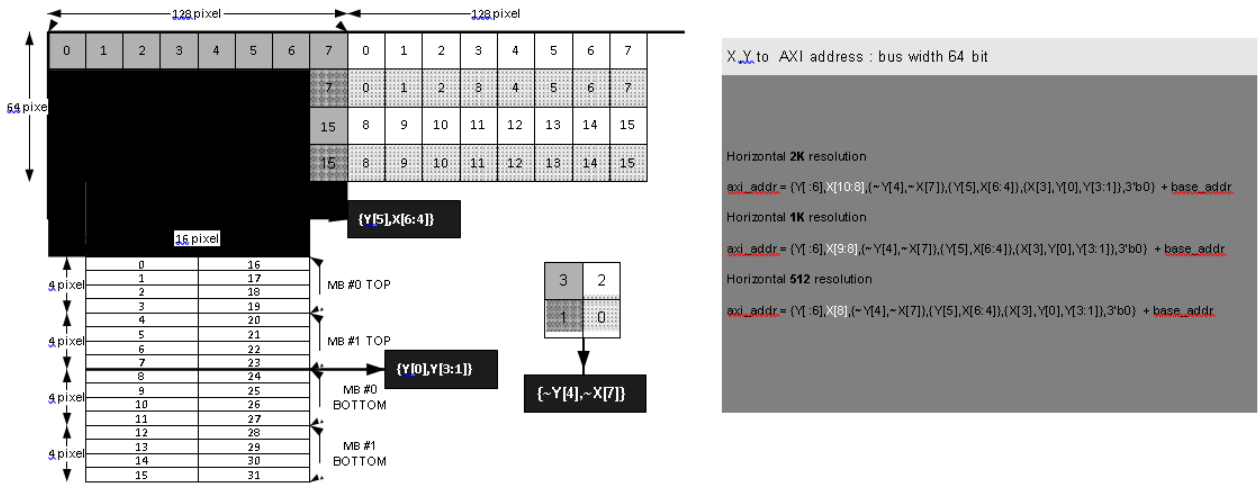


Рисунок 1.107 – Тайловая карта распределения кадра/полукадра в порядке вертикального сканирования (по компоненте цветности)

Picture – Изображение
128 pixel – 128 пикселей
128 pixel –128 пикселей
64 pixel – 64 пикселя
16 pixel – 16 пикселей
4 pixel – 4 пикселя
4 pixel – 4 пикселя
TOP – ВЕРХ
TOP – ВЕРХ
BOTTOM – НИЗ
BOTTOM – НИЗ
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

Тип 5: Тайловая карта распределения макроблоков кадра в порядке растрового сканирования

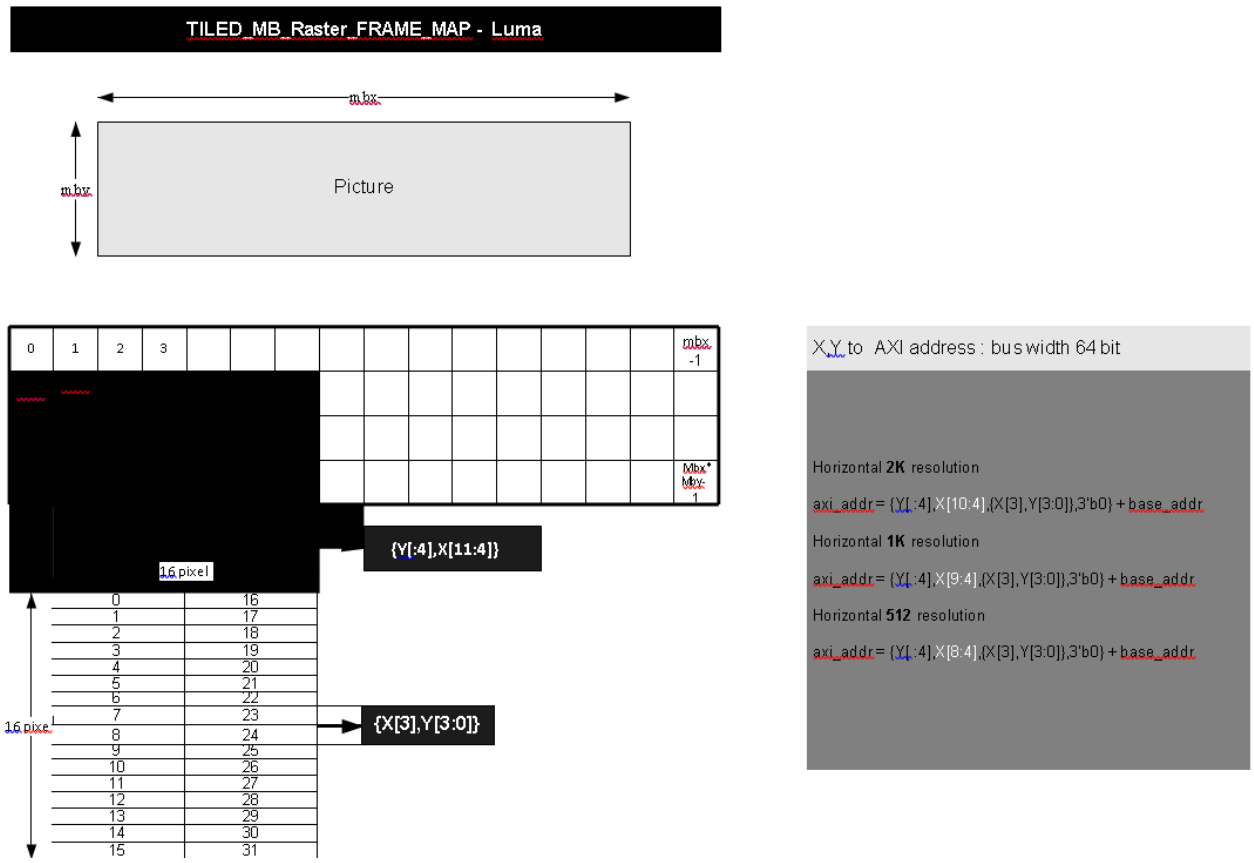


Рисунок 1.108 – Тайловая карта распределения макроблоков кадра в порядке растрового сканирования (по компоненте яркости)

Picture – Изображение
16 pixel – 16 пикселей
16 pixel – 16 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

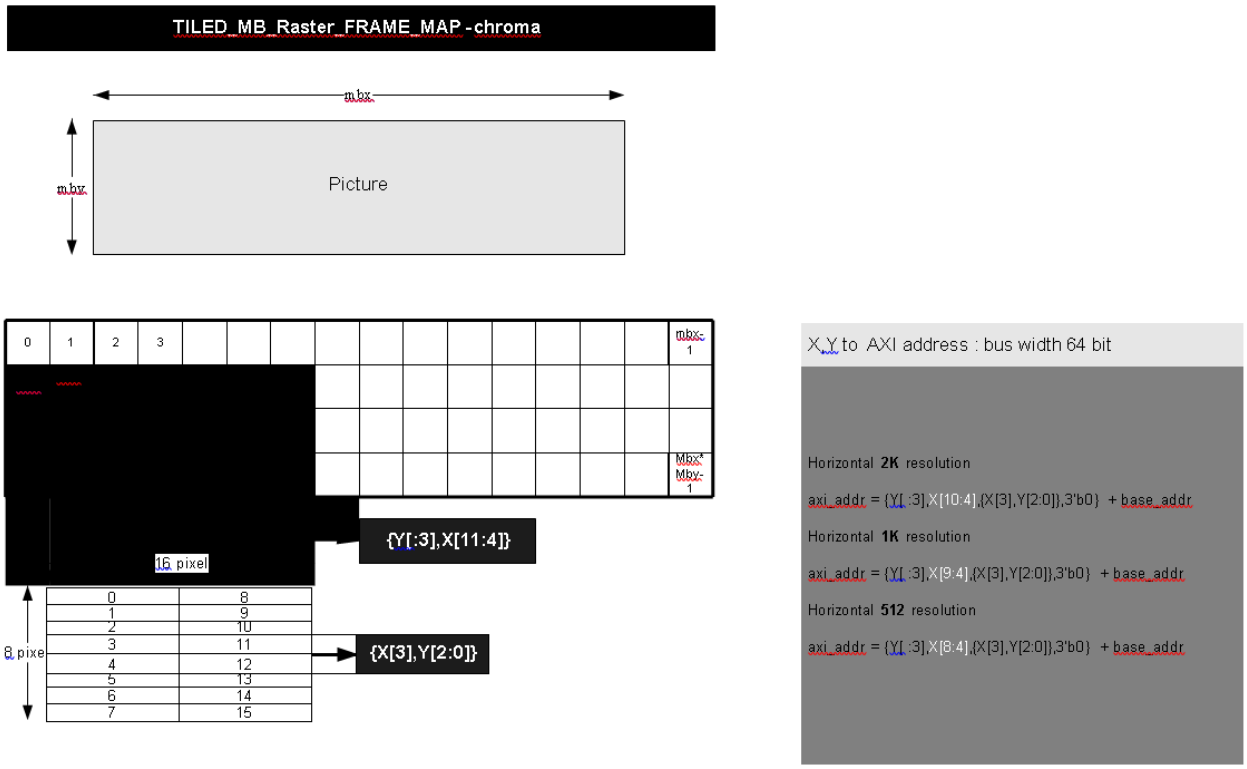


Рисунок 1.109 – Тайловая карта распределения макроблоков кадра в порядке растрового сканирования (по компоненте цветности)

Picture – Изображение
16 pixel – 16 пикселей
8 pixel – 8 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

Тип 6: Тайловая карта распределения полукадров в порядке растрового сканирования

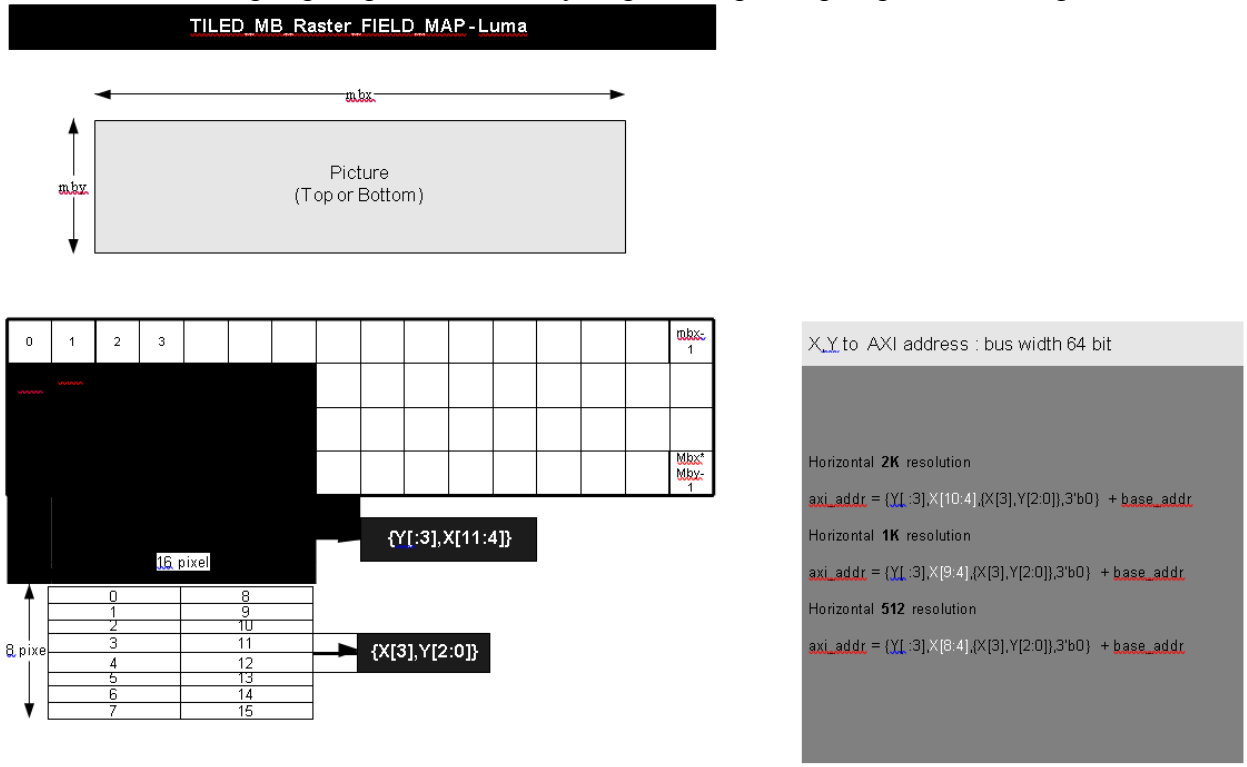


Рисунок 1.110 - Тайловая карта распределения полукадров в порядке растрового сканирования (по компоненте яркости)

Picture (Top or Bottom) – Изображение (верхний или нижний край)
16 pixel – 16 пикселей
8 pixel – 8 пикселей
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

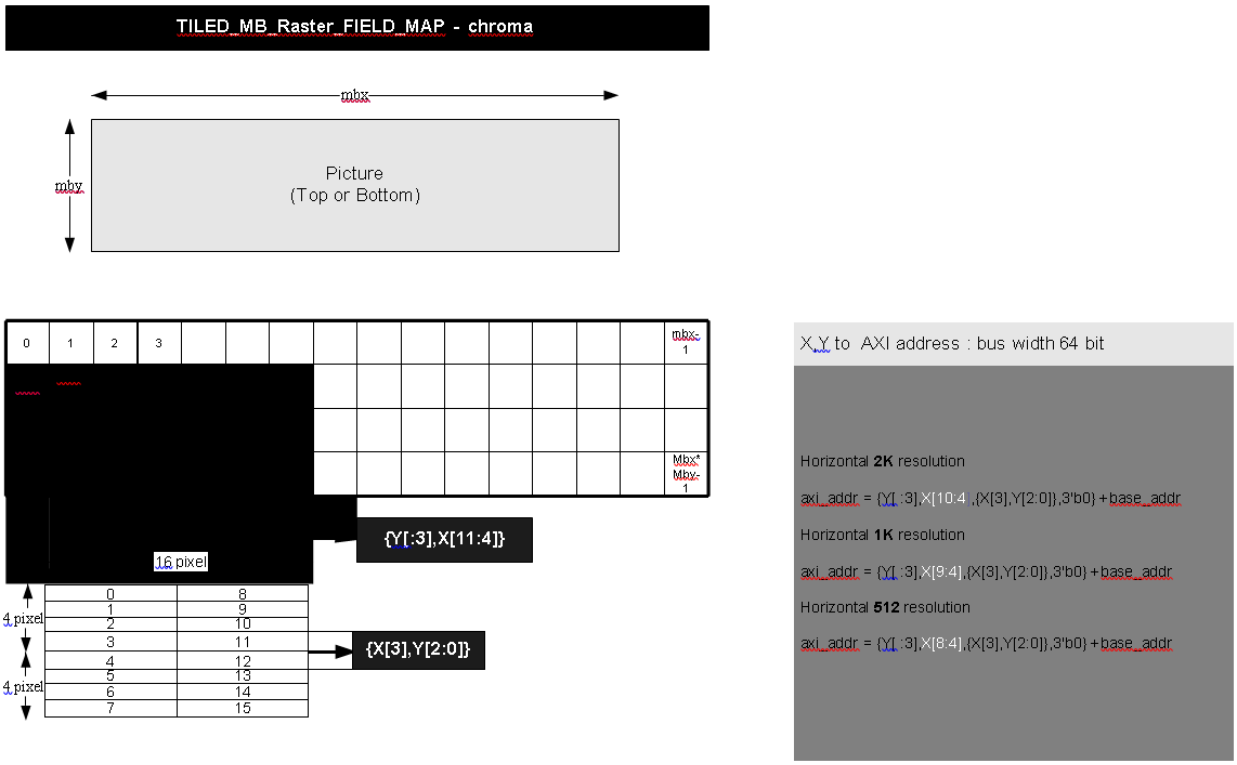


Рисунок 1.111 - Тайловая карта распределения полукадров в порядке растрового сканирования (по компоненте цветности)

Picture (Top or Bottom) – Изображение (верхний или нижний край)
16 pixel – 16 пикселей
4 pixel – 4 пикселя
4 pixel – 4 пикселя
X, Y to AXI address: bus width 64 bit – X, Y для адреса AXI: ширина шины – 64 бита
Horizontal 2K resolution – Разрешение 2000 пикселей по горизонтали
Horizontal 1K resolution – Разрешение 1000 пикселей по горизонтали
Horizontal 512 resolution - Разрешение 512 пикселей по горизонтали

1.3.6.1.2.1.2.2 Форматы буфера кадра

Исходные или декодированные изображения (кадры или полукадры) состоят из трех массивов компонент (сэмплов) – одного массива компонент яркости и двух массивов компонент цветности. В данном документе буфер кадра принято считать множеством сэмплов, находящихся во внешней памяти. Буфер кадра обладает следующими особенностями:

Для всех стандартов видеосигналов используется единый формат сэмплирования 4:2:0

Буфер кадра конфигурируется в соответствии со следующими параметрами:

Базовый адрес буфера кадра по компоненте яркости равен 8 байтам

Базовый адрес буфера кадра по компоненте Cb равен 8 байтам

Базовый адрес буфера кадра по компоненте Cr равен 8 байтам: В формате с чередованием сэмплов для компонент Cb/Cr на карте распределения базовый адрес компоненты Cr теряет свое значение, вследствие того что для хранения или загрузки сэмплов компонент Cb/Cr в формате с чередованием используется базовый адрес компоненты Cb.

Шаг по индексу в ширине буфера кадра по компоненте яркости: значение шага по индексу должно быть равно (или больше) значению ширины изображения, умноженному на 8. Это значит, что, по меньшей мере, 3 значимых бита из 13-битовой последовательности шагов по индексу должны быть всегда равны “0”. Шаг по индексу буферов кадра по компоненте цветности составляет половину шага по индексу по компоненте яркости. Следует убедиться в том, что на карте распределения компонент Cb/Cr без чередования (с разделением компонент Cb/Cr), значение шага по индексу буферов кадра по компоненте яркости кратно 16, а значение шага по индексу буферов кадра по компоненте цветности может быть кратным 8.

Порядок байтов в буферах кадра

Порядок байтов в буферах кадра можно сконфигурирован по методу от младшего к старшему (прямой порядок) или по методу от старшего к младшему (обратный порядок).

Ширина внутренних регистров в размере буфера кадра и изображения составляет 12 бит.

Шаг по индексу для определения ширины буфера кадра в блоке пикселей

Размер изображения в блоке пикселей, как в горизонтальном, так и в вертикальном направлениях

Поддерживаются обе карты распределения компонент Cb/Cr – как в формате с чередованием, так и в формате с разделением компонент.

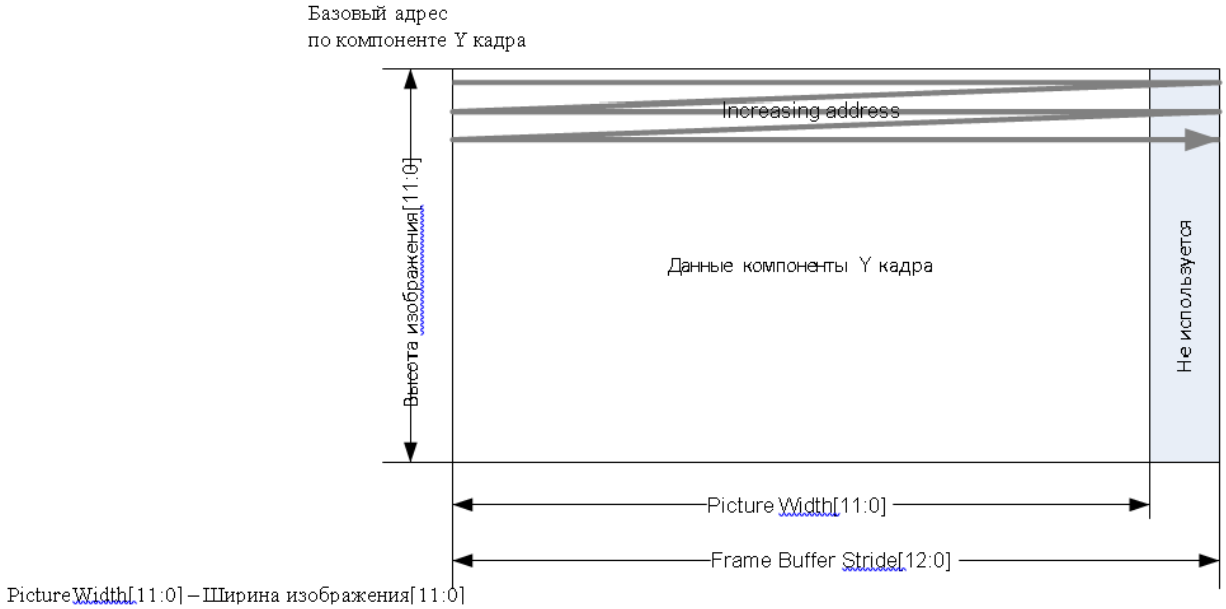
Пара полукадров изображений (верхний и нижний) хранится в общем буфере кадра.

1.3.6.1.2.1.2.3 Составление карты распределения для буферов кадра

На рисунке 1.112 - “Типы конфигурирования буфера кадра” приведен пример того, как можно сконфигурировать буфер кадра на линейной карте распределения. Различают два типа конфигурирования.

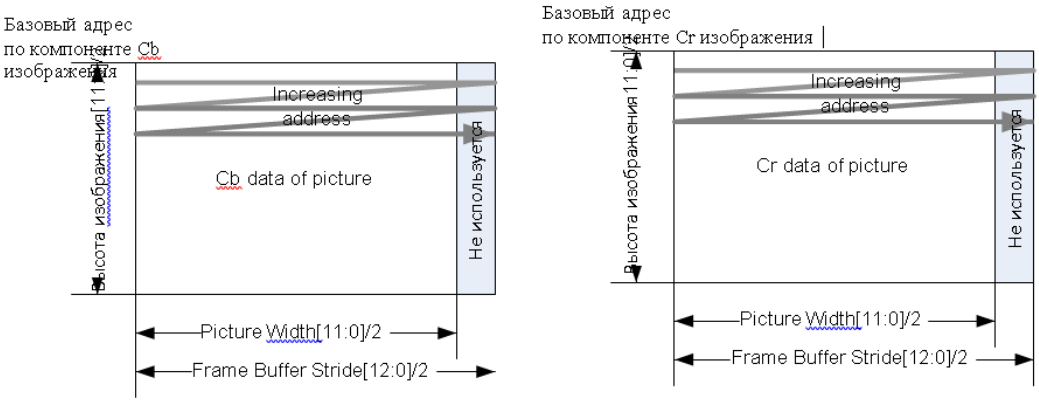
Первый тип конфигурирования рассмотрен на схемах (а) и (б). В данном случае каждой компоненте цветности выделяется собственный буфер. Поэтому, у компонент Cb и Cr есть два буфера.

Второй тип конфигурирования рассмотрен на схемах (а) и (в). Данный тип конфигурирования предназначен для снижения степени загрузки шины за счет использования более длительного импульса, чем при конфигурировании по первому типу. Из схемы (в) видно, что сэмплы компонент Cb и Cr хранятся в формате с чередованием.



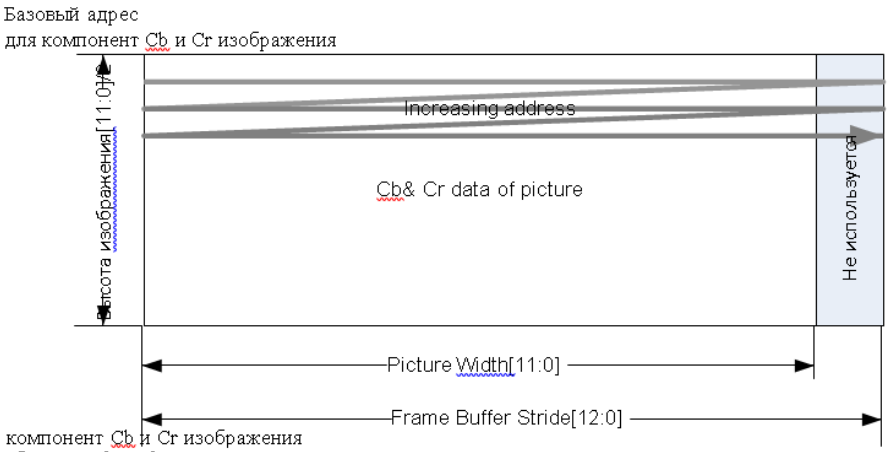
PictureWidth[11:0] – Ширина изображения[11:0]
Frame Buffer Stride[12:0] – Шаг по индексу буфера кадра[12:0]
Increasing address – В сторону увеличения адреса

a. Буфер кадра по компоненте яркости



Cb data of picture – Данные компоненты Cb изображения
Picture Width[11:0]/2 – Ширина изображения[11:0]/2
Frame Buffer Stride[12:0]/2 – Шаг по индексу буфера кадра[12:0]/2
Increasing address – В сторону увеличения адреса
Cr data of picture – Данные компоненты Cr изображения
Picture Width[11:0]/2 – Ширина изображения[11:0]/2
Frame Buffer Stride[12:0]/2 – Шаг по индексу буфера кадра[12:0]/2
Increasing address – В сторону увеличения адреса

б. Буфер кадра для компонент Cb и Cr в формате без чередования



Cb&Cr data of picture – Данные компонент Cb и Cr изображения
Picture Width[11:0] – Ширина изображения[11:0]
Frame Buffer Stride[12:0] – Шаг по индексу буфера кадра[12:0]
Increasing address – В сторону увеличения адреса

в. Буфер кадра для компонент Cb и Cr в формате с чередованием

Рисунок 1.112 - Типы конфигурирования буфера кадра

Более детально типы конфигурирования (а именно, порядок байтов) будут рассматриваться на примерах изображений в формате QCIF(176x144) в последующих разделах.

1.3.6.1.2.1.2.3.1 Буферы кадра по компоненте яркости

На рисунке – 1.113 “Расположение пикселей компоненты яркости в буфере кадра” показан пример того, как сэмплы компоненты яркости отображаются во внешней памяти на линейной карте распределения. На схеме (а) показано распределение пикселей в режиме «от младшего к старшему», а на схеме (б) – распределение пикселей в режиме «от старшего к младшему». За базовый адрес принят базовый адрес буфера кадра по компоненте яркости.



а Расположение пикселей @ ENDIAN_MODE = 2’b00

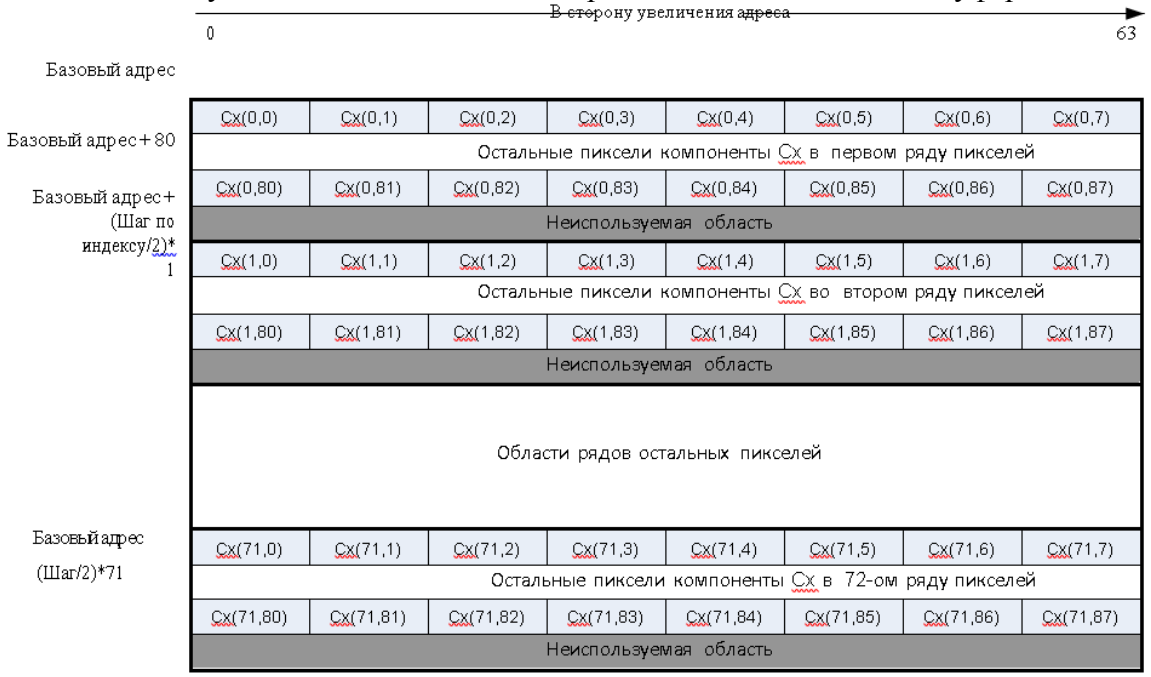


б Расположение пикселей @ ENDIAN_MODE = 2’b01

Рисунок 1.113 - Расположение пикселей компоненты яркости в буфере кадра

1.3.6.1.2.1.2.3.2 Буферы кадра для компонент цветности

На рисунке – 1.114 “Расположение пикселей компоненты цветности в буферах кадра компонент CbCr в формате с разделением” приводится пример того, как сэмплы компоненты цветности отображаются во внешней памяти на линейной карте распределения в буфере кадра компонент Cb/Cr в формате с разделением. На схеме (а) показано распределение пикселей в режиме «от младшего к старшему», а на схеме (б) – распределение пикселей в режиме «от старшего к младшему». У каждой компоненты яркости есть собственный буфер.



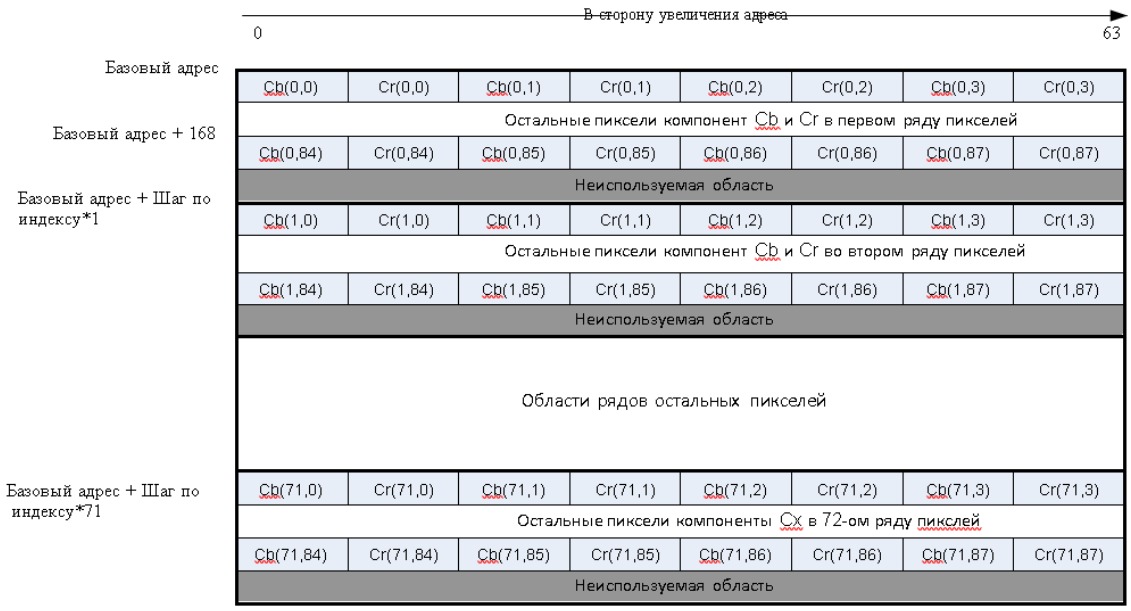
(a) Расположение пикселей @ ENDIAN_MODE = 2’b00 для буфера кадра для компонент Cb/Cr в режиме с разделением



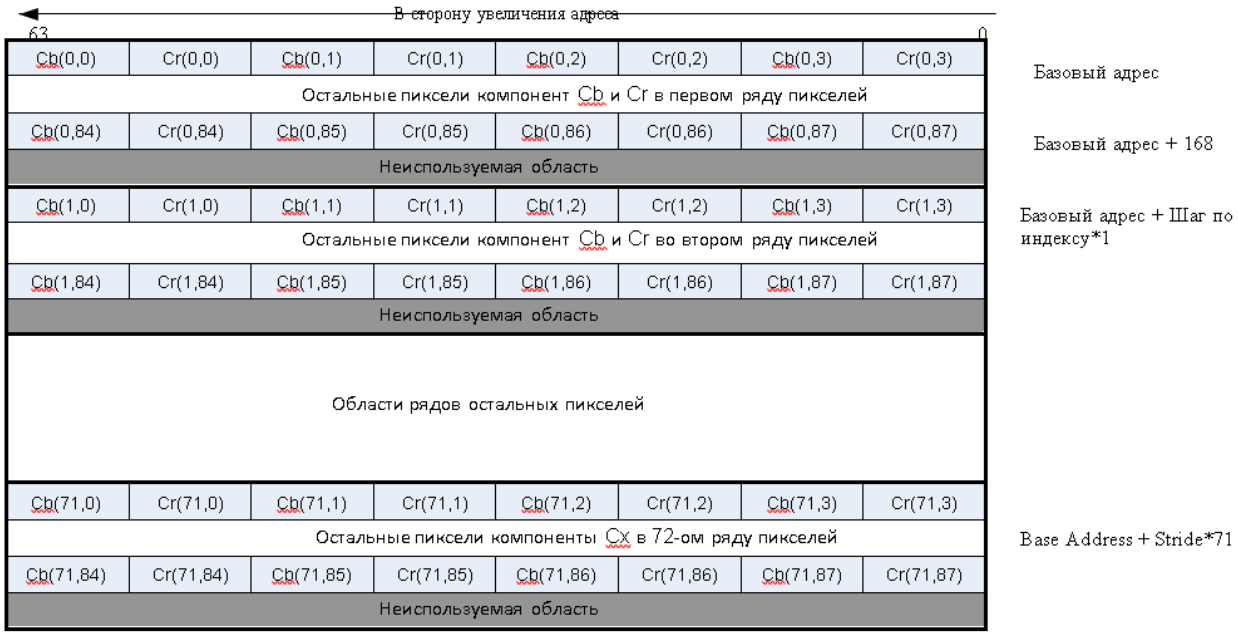
(b) Расположение пикселей @ ENDIAN_MODE = 2’b01 для буфера кадра компонент Cb/Cr в режиме с разделением

Рисунок 1.114 - Расположение пикселей компоненты цветности в буферах кадра компонент CbCr в формате с разделением

На рисунке - 1.115 “Расположение пикселей компоненты цветности в буфере кадра компонент CbCr в формате с чередованием” рассматривается пример того, как сэмплы компоненты цветности отображаются во внешней памяти на линейной карте распределения в буфере кадра компонент Cb/Cr в формате с чередованием. На схеме (а) показано распределение пикселей в режиме «от младшего к старшему», а на схеме (б) – распределение пикселей в режиме «от старшего к младшему». На карте распределения компонент Cb/Cr в формате с чередованием компонентам Cb и Cr изображения выделяется всего один буфер. Значение шага по индексу соответствует значению шага по индексу в режиме яркости.



(а) Расположение пикселей @ ENDIAN_MODE = 2’b00 для буфера кадра компонент Cb/Cr в режиме с чередованием



(б) Расположение пикселей @ ENDIAN_MODE = 2’b01 для буфера компонент Cb/Cr в режиме с чередованием

Рисунок 1.115 - Расположение пикселей компоненты цветности в буфере кадра компонент CbCr в формате с чередованием

1.3.6.1.2.1.3 Рабочий буфер блока VPU

1.3.6.1.2.1.3.1 Управление буфером параметров

Буфер параметров используется для хранения входных аргументов командной строки или выходных данных, получаемых в ответ на определенные команды. Данный буфер параметров находится в памяти SDRAM. Адрес стартового байта должен задаваться в регистре ParaBufAddr.

Адрес буфера кадра

Входные аргументы командной строки для команды SET_FRAME_BUF. Более подробное описание этого адреса приводится в описании команды SET_FRAME_BUF.

Число бит закодированного макроблока

Выходные данные, получаемые в ответ на команду ENC_PIC_RUN

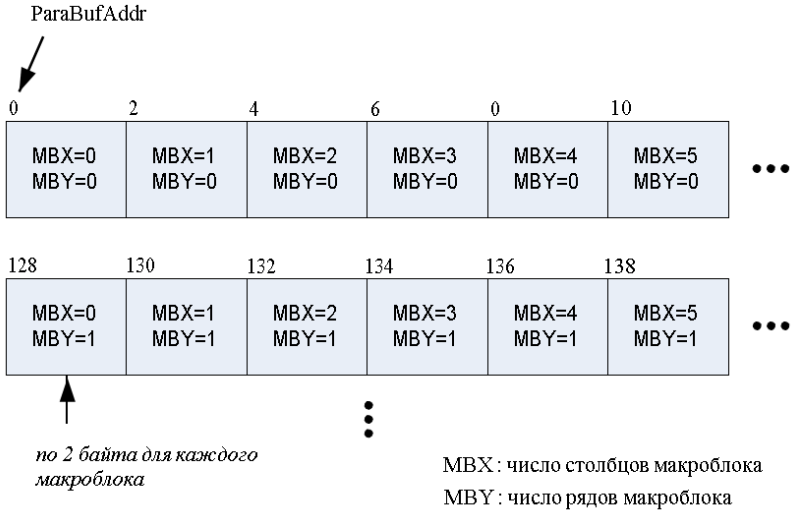


Рисунок 1.116 - Структура количества бит закодированного макроблока

Если в регистре CMD_ENC_SEQ_OPTION флаг MbBitReport имеет значение 1, то по завершении выполнения команды ENC_PIC_RUN BIT-процессор заносит в буфер параметров позицию стартовых бит каждого макроблока в направлении от начала изображения. Позиция стартовых бит каждого макроблока записывается в формате 16-разрядных беззнаковых данных. Сдвиг шага по индексу рядов макроблока составляет 128 байт, поэтому последние 128-720/16 * 2 = 38 байт из совокупности рядов макроблока никагда не используются в формате D1. Байтовым адресом позиции макроблока <MbX, MbY> является {ParaBufAddr + MbY * 128 + MbX * 2}. Максимальная высота закодированного исходного изображения в формате D1 составляет 576 пикселей. Поэтому максимальный объем буфера числа бит макроблока - 128 * 576/16 = 4608 байт (4,5 килобайт). Более подробное описание данного формата приводится на рисунке - 1.116 “Структура количества бит закодированного макроблока”.

Данные о закодированных слайсах

Выходные данные, передаваемые в ответ на команду ENC_PIC_RUN

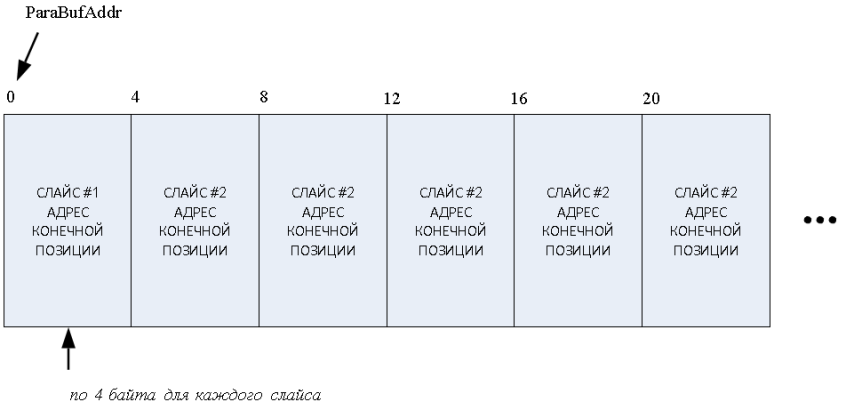


Рисунок 1.117 - Структура данных о закодированных слайсах

На рисунке 1.117 “Структура данных о закодированных слайсах” представлен формат построения данных о закодированных слайсах. Если в регистре CMD_ENC_SEQ_OPTION флаг SliceInfoReport имеет значение 1, то по завершении выполнения команды ENC_PIC_RUN BIT-процессор заносит в буфер параметров конечный адрес по каждому из закодированных слайсов из внешнего буфера битовых потоков в памяти SDRAM. Значение общего количества сгенерированных слайсов записывается в регистр RET_ENC_PIC_SLICE_NUM, а адрес конечной позиции каждого слайса в памяти SDRAM заносится в буфер параметров. Хост-компьютер может рассчитать точное число байт для каждого слайса, начиная с конечной позиции каждого слайса. Стартовым адресом буфера позиций слайсов, который располагается непосредственно под буфером номеров бит макроблока, является адрес {ParaBufAddr

+ 4608(0x1200)}. Адрес конечной позиции слайса в SDRAM записывается в формате 32-разрядных беззнаковых данных по каждому слайсу.

Данные RBSP о наборе параметров изображения/последовательности

Входной аргумент для команды DEC_PARA_SET или выходной поток закодированных первичных байтовых данных (далее RBSP) для ENC_PARA_SET.

1.3.6.1.2.1.3.2 Область управления рабочим буфером

Данная область предназначена для внутреннего буфера памяти SDRAM при выполнении операций кодирования и декодирования. Например, для буфера восстановленных рядов пикселей при прогнозировании коэффициентов AC/DC в стандарте MPEG4 или прогнозировании в моде intra в стандарте H.264, для буфера контекстного хранения при выполнении нескольких процессов, для буфера переупорядочения битового потока при разделении данных в стандарте MPEG4 или для обеспечения гибкого порядка макроблоков/произвольного порядка слайсов (далее FMO/ASO) в стандарте H.264 и т.д.

Более подробное описание организации рабочего буфера представлено в таблице 1.148.

Таблица 1.148 - Организация рабочего буфера

Тип	Описание	Объем (Кбайт)
PIC_TEMP	Данная область используется совместно всеми экземплярами	210 кбайт
STATIC	47 кбайт хранилища статических данных для каждого экземпляра	47 кбайт * число экземпляров

Кроме рабочего буфера декодеру стандарта кодирования аудио и видеоданных (далее AVC) требуется дополнительный рабочий буфер, который задается в двух регистрах главного интерфейса - CMD_SET_FRAME_SLICE_BB_START и CMD_SET_FRAME_SLICE_BB_SIZE.

Согласно рекомендациям, объем CMD_SET_FRAME_SLICE_BB_SIZE должен превосходить объем буфера для одного кадра, например, для формата HD это 1920x1088 пикселей.

1.3.6.1.2.1.4 Требования к объему внешней памяти

В памяти SDRAM каждому стандарту выделяется отдельный буфер кадра. В таблице 1.149 приведены объемы памяти (SDRAM) по каждому из стандартов, поддерживающих формат HD (1920x1088).

Таблица 1.149 - Объем буфера кадра

H.264		VC-1	AVS/RV	MPEG-4	MPEG-2	VP8	Theora
Минимальное количество буферов кадра	6	6	4	4	4	4	4
Буфер кадра	3 060	3 060	3 060	3 060	3 060	3 060	3 060
Совмещенный вектор движения MV	510	510	510	510	-	-	-
Общий буфер кадра	21 420	21 420	14 280	14 280	12 240	12 240	12 240
BIT-код	132	132	132	132	132	132	132
Рабочий буфер битовых потоков	681	257,5	257, 5	257,5	257,5	767	257,5
Общий рабочий буфер (кбайт)	22 233	21 809,5	14 669,5	14 669,5	12 629,5	13 139	12 629,5

Общий буфер кадра =

Для AVC_DEC: количество буферов кадра * (буфер кадра + совмещенный MV)

Прочие декодеры: (количество буферов кадра * буфер кадра) + совмещенный MV

Общая память SDRAM = общий буфер кадра + BIT-код + рабочий буфер битовых потоков

Примечание Более подробную информацию по данной теме можно найти в файле SDRAM_requirement.xls. Для Базового профиля (далее BP) стандарта H.264 требуется дополнительная память (обычно 512 кбайт~1 Мбайт). Для стандарта Theora также требуется дополнительная память (обычно 1 Мбайт, максимально 2759 кбайт).

1.3.6.1.2.2 Программная модель

1.3.6.1.2.2.1 Регистры главного интерфейса

Блок VPU связывается с хост-процессором через шинный интерфейс APB. В таблице 1.150 представлена карта распределения адресов области, доступ к которой может обеспечиваться по интерфейсу APB.

Таблица 1.150 - Карта распределения адресов внутреннего регистра

PAD-DR[13:9]	Модуль Описание	
5'b0000x	Главный интерфейс BIT-процессора	
5'b00010	Контроллер макроблоков (секвенсер)	
5'b00011	Оценка параметров движения (далее ME)	В режиме нормального функционирования доступа с хост-процессора к данным регистрам нет
5'b00100	Фильтр деблокирования	
5'b00101	Не определен	
5'b0011x	Модуль Inter-прогнозирования	
5'b01XXX	Универсальный интерфейс прямого доступа к памяти (далее GDI) (ШИНА)	
5'b10XXX	GCU (внутренний модуль DMA)	

Регистры BIT-процессора подразделяются на 2 категории: аппаратные регистры (далее HW) и программные регистры (далее SW). Адресное пространство 0x000 ~ 0x0FC (адресное пространство для 64 регистров) выделено под аппаратные регистры. Значения и функция сброса этих регистров являются фиксированными (не конфигурируются).

Адресное пространство 0x100 ~ 0x1FC (64 регистра) предназначено для главных аппаратных регистров. Они конфигурируются приложением в BIT-процессоре и после сброса не имеют значений. Данные регистры используются для обеспечения интерфейса между хост-компьютером и BIT-процессором. Регистры выше 32-го (адресное пространство 0x100 ~ 0x17C) используются в качестве статических параметров. Значения и функции этих регистров являются неизменными для любых команд запуска (SEQ_INIT, SEQ_END, PICTURE_RUN и т.д.) и используются для любых команд и процессов. Регистры ниже 32-го (адресное пространство 0x180 ~ 0x1FC) используются в качестве временных аргументов командной строки. Значения или функции этих регистров могут быть разными для каждой из команд запуска.

Например, для всех процессов используется один регистр BitStreamCtrl. Это означает, что он влияет на все операции с видеоданными. А регистр CMD_SET_FRAME_BUF_STRIDE используется только в процессе запуска при выполнении команды SET_FRAME_BUF. Поэтому для других команд данный регистр (с адресом 0x184) будет выполнять другие функции. Значение [LineStride] используется только для текущего процесса. В остальных процессах сдвиг шага по индексу в строке может быть другим, потому что в ходе выполнения команды SET_FRAME_BUF BIT-процессор считывает регистр CMD_SET_FRAME_BUF_STRIDE и хранит значение [LineStride] во внутренней памяти.

Примечание Более подробную информацию о регистрах можно найти в «Руководстве к блоку VPU для программистов».

1.3.6.1.2.2.2 Инициализация блока VPU

При первом включении блока VPU для последующего его использования, то есть, проведении, так называемой, инициализации, хост-процессор должен выполнить ряд операций (см. список ниже). При вызове одной функции прикладного программного интерфейса (далее API) - VPU_Init() можно легко выполнить следующие операции.

- Отключить BIT-процессор, задав CodeRun (BASE + 0x000) = 0.

- Записать микрокод BIT-процессора в память SDRAM. Доступ к ней с блока VPU обеспечивается на протяжении всего периода работы.
- Загрузить первый микрокод N KB (загрузочный код) непосредственно в кодовую память BIT-процессора.
- Задать указатели буферов BIT-процессора: рабочего буфера, буфера параметров и буфера хранения кодов.
- Настроить опции управления буфером потока данных и установить режим упорядочивания байтов буфера кадра.
- Регистры разрешения прерывания и сброса.
- Включить BIT-процессор, задав CodeRun = 1.
- Подождать когда сбросится регистр BusyFlag.

Примечание Если регистр CodeRun установлен в 1, то через 13 циклов регистр BusyFlag установится в 1.

1.3.6.1.2.2.3 Кодирование

1.3.6.1.2.2.3.1 Создание экземпляра кодера

По завершении инициализации блока VPU на первом этапе работы кодера необходимо создать экземпляр кодера и указатель для идентификации экземпляра этого кодера.

При создании нового экземпляра кодера в прикладной структуре EncOpenParam должны быть определены внутренние характеристики данного экземпляра кодера. В данную структуру входят следующие данные о новом экземпляре кодера:

- адрес и объем буфера битового потока;
- стандарт кодека;
- размер изображения;
- заданная частота кадров и заданная битовая скорость кадра с параметрами модели VBV, initialDelay;
- интервал между ключевыми кадрами;
- включение/отключение слайсов, режим размера слайса и размер слайса;
- итоговая отчетность: sliceReport, mbReport, qpReport и т.д.;
- прочие опции: enableAutoSkip и intraRefresh;
- включение режима кольцевого буфера;
- разрешение размещения динамического буфера;
- шаг квантизации в моде Intra;
- параметры, характерные для данного стандарта видеосигналов.

1.3.6.1.2.2.3.2 Конфигурирование блока VPU для экземпляра кодека

Инициализация последовательности

После записи всех необходимых данных по новому экземпляру кодера хост-приложение должно сконфигурировать блок VPU таким образом, чтобы он смог поддерживать новый экземпляр кодера. Данная процедура выполняется посредством настройки всех параметров кодера в регистрах главного интерфейса VPU и передачи в блок VPU команды ENC_SEQ_INIT для инициализации операции конфигурирования в блоке VPU.

Регистрация буферов кадра

Процесс конфигурирования завершается после регистрации буферов кадра в блоке VPU для последующего кодирования изображений.

Генерирование синтаксисов заголовков верхнего уровня

Существует два способа генерирования синтаксисов заголовков – путем передачи команды PARA_BUF и с использованием буфера потоков данных. Чтобы получить синтаксисы заголовков, рекомендуется воспользоваться командой ENC_PUT_AVC/MP4 _HEADER через буфер потоков данных. Если в приложении используется данный набор команд, то сформированные синтаксисы заголовков будут храниться в буфере битовых потоков в соответствии с данными настройками порядка байтов.

1.3.6.1.2.2.3.3 Работа кодера изображений в блоке VPU

Входные данные YUV

Перед началом работы кодера изображений приложение хост-компьютера должно обеспечить, чтобы исходное изображение YUV в вертикальном формате 4:2:0 или 4:2:2 соответствовало предварительно заданному размеру, установленному в стандартах H.263, MPEG-4 и H.264.

Начало кодирования изображения

При запуске операции кодирования изображения приложение должно передать в блок VPU следующие данные:

- адрес кадра источника;
- шаг квантизации;
- опции принудительного пропуска кадров и принудительного использования I-изображений;
- формат источника.

После настройки всех этих данных в блоке VPU хост-процессор, передав в блок VPU команду ENC_PIC_RUN, может приступить к кодированию изображения.

Завершение кодирования изображения

Процесс кодирования изображения может занять некоторое количество времени, но пока ожидается завершение кодирования изображения, приложение может продолжать выполнять другие задачи, например, пакетизацию закодированного потока данных для последующей его передачи. Приложением используются два способа определить, завершена ли операция кодирования изображения: опрос регистра статуса и передача сигнала о прерывании.

Обработка потока данных кодирования

Если буфер закодированных потоков данных имеет достаточно большой объем и рассчитан на хранение потока изображений любого объема, то в процессе кодирования изображения битовый поток кодеру не потребуется. По окончании кодирования хост-приложение считывает закодированный битовый поток в соответствии с требованиями пакетизации.

Однако если буфер закодированных потоков данных не рассчитан на хранение всего потока изображений, может наступить момент, когда буфер кодера будет переполнен, и пока эта проблема не будет устранена, кодер в блоке VPU не сможет выполнять никакие задачи и будет находиться в «зависшем» состоянии. Поэтому чтобы избежать подобного нежелательного эффекта «зависания» кодера, приложение в процессе кодирования изображений должно продолжать считывать закодированный битовый поток из буфера потоков.

Извлечение результатов кодирования

Как только кодирование изображения будет завешено, хост-приложение сможет получить результаты этого процесса: тип закодированного изображения, количество слайсов и т.д. И согласно настройкам входных параметров закодированного изображения, из блока VPU можно также извлечь данные о разделителе слайсов и разделителе макроблоков.

1.3.6.1.2.2.3.4 Закрытие экземпляра кодера

Каждый раз, когда приложение собирается завершить работу кодера и закрыть экземпляр кодера, оно просто отключает указатель этого экземпляра и посредством передачи команды SEQ_END в блок VPU, позволяет блоку VPU «узнать» о предстоящем закрытии этого экземпляра.

1.3.6.1.2.2.3.5 Команды динамического конфигурирования

Иногда в ходе процесса кодирования изображений приложению необходимо передать в блок VPU некоторые специальные команды, например, команду о необходимости развернуть исходные изображения перед началом кодирования, команду вставки синтаксисов заголовка верхнего уровня и т.д. Специальные запросы такого типа поддерживаются интерфейсом API блока VPU “C&M” с помощью набора команд, передаваемых из хост-приложения:

- повернуть и отразить исходный кадр перед кодированием;
- извлечь синтаксисы заголовков верхнего уровня: заголовок VOS, VO и VOL в формате MPEG-4, и SPS и PPS в формате H.264 для использования вне канала;
- вставить синтаксисы заголовков верхнего уровня, например, заголовок VOS, VO и VOL в формате MPEG-4, и SPS и PPS в формате H.264;
- изменить в динамическом режиме параметры кодера между операциями кодирования изображения, например, битовую скорость, частоту кадров, интервал между ключевыми кадрами, режим слайсов и т.д.

Для обеспечения широкого спектра функций, которые могут понадобиться пользователям, предоставляется ряд узкоспециализированных команд.

1.3.6.1.2.2.4 Декодирование

1.3.6.1.2.2.4.1 Создание экземпляра декодера

По окончании инициализации блока VPU, на первом этапе работы декодера необходимо создать экземпляр декодера и указатель для идентификации экземпляра этого декодера.

При создании нового экземпляра декодера следует определить внутренние характеристики данного экземпляра декодера в структуре DecOpenParam приложения. В данную структуру входят следующие данные о новом экземпляре декодера:

- адрес и объем буфера битового потока;
- стандарт кодека;
- reorderEnable;
- данные о размере изображения;
- размер и адрес буфера для хранения последовательности RBSP с SPS/PPS.

В случае декодирования наибольший объем данных извлекается из входного потока данных, поэтому, для создания экземпляра декодера потребуется как можно больше параметров.

1.3.6.1.2.2.4.2 Конфигурирование блока VPU для экземпляра декодера

Подача битового потока в буфер потоков данных

В декодере при инициализации последовательности для считывания конфигураций декодера выполняется разбор синтаксисов заголовков верхнего уровня, например, VOS/VO/VOL в формате MPEG-4 и SPS/PPS в формате H.264. Поэтому для того, чтобы запустить процесс инициализации последовательности, приложение должно заполнить буферы потоков данных декодирования соответствующим количеством битовых данных. В некоторых системах хост-приложения не могут гарантировать расположение синтаксисов заголовков таких типов в начале битового потока. Поэтому пока в блок VPU будут поступать из входного потока все

необходимые данные, приложение не должно прерывать подачу потока входных данных в буфер битового потока декодера.

Инициализация последовательности

После создания нового экземпляра и подачи битового потока входных данных в буфер потоков приложение может передать в блок VPU команду DEC_SEQ_INIT, чтобы извлечь из битового потока данные о конфигурации декодера. После разбора синтаксисов заголовка декодер передает в ответ важные данные о конфигурации декодера:

- размер изображения,
- данные о прямоугольной обрезке изображения,
- частота кадров,
- минимальное количество буферов кадра,
- данные о формате MPEG-4,
- величина задержки в буфере кадра на время переупорядочивания отображаемых данных,
- индикация опции деблокирования (Приложение J),
- номер ближайшего индекса декодированного изображения, выданного после декодирования одного кадра,
- расчетные объемы буферов хранения слайсов.

Размер изображения, данные о котором были получены из битового потока, не обязательно должен быть кратным 16x16. Однако для правильной работы декодера объем буфера кадра должен быть кратным 16x16. Поэтому объем данных, полученных после работы в максимальном режиме, будет изменен на величину кратную 16x16. Пользуясь данными о размере изображения и минимальном количестве буферов кадра приложение перед началом декодирования изображения должно зарезервировать буферы кадра и предоставить их блоку VPU.

Регистрация буферов кадра

Процесс конфигурирования завершается регистрацией буферов кадра в блоке VPU для их последующего использования при декодировании изображений.

Конфигурирование интерфейса AXI по вспомогательной шине

В целях повышения качества декодирования за счет уменьшения полосы пропускания системы, в которой задействован интерфейс AXI по вспомогательной шине, обеспечивается доступ с блока VPU к внутренней памяти SRAM. В VPU для интерфейса AXI по вспомогательной шине обеспечивается шесть опций: BIT_USE, IP_USE, DBKY_USE, DBKC_USE, OVL_USE и BTP_USE. Соответственно, хост-компьютер, учитывая данные о системных ресурсах, может сам выбрать подходящие опции и задать базовый адрес каждой области, используемой в каждом конкретном случае.

- Если подключена опция BIT_USE, то в блоке VPU сохраняются такие промежуточные данные, как, например, данные прогнозирования макроблока. Размер опции BIT_USE вычисляется как 128 * ширину макроблока в байтах. ПРИМЕЧАНИЕ: ширина макроблока в байтах = размеру изображения по горизонтали / ширину макроблока. Например, если размер изображения равен 1920 * 1088, то ширина макроблока в байтах составляет 1920/16 = 120.]
- Если подключена опция IP_USE, то в блоке VPU сохраняются промежуточные данные для прогнозирования в моде intra/ACDC. Для прогнозирования в моде intra размер опции IP_USE вычисляется как 64 * ширину макроблока в байтах. Для прогнозирования коэффициентов ACDC размер опции IP_USE равен 128 * ширину макроблока в байтах.
- Если подключена опция DBKY_USE, то в VPU сохраняются промежуточные данные для фильтра деблокирования.

- Если подключена опция DBKC_USE, то в блоке VPU сохраняются промежуточные данные цветности для фильтра деблокирования. Совокупный размер опций DBKY_USE и DBKC_USE для VC-1 вычисляется как 512 * ширину макроблока в байтах. Для формата H.264 размер этих опций составляет 256 * ширину макроблока. В случае деблокирования выходных данных в форматах MPEG-4 и MPEG-2 размер опций составляет 128 * ширину макроблока в байтах.
- Если подключена опция OVL_USE, то в VPU сохраняются промежуточные данные для фильтра перекрывания. Данный буфер используется только в случае использования VC-1. Размер опции OVL_USE определяется как 80 * ширину макроблока в байтах.
- Если подключена опция BTP_USE, то в VPU сохраняются промежуточные данные битовой плоскости, полученные от BIT-процессора. Данный буфер используется только в случае VC-1. Размер опции OVL_USE вычисляется как 160 * ширину макроблока в байтах.

1.3.6.1.2.2.4.3 Работа декодера изображений в блоке VPU

Начало декодирования изображения

При запуске операции декодирования изображения приложение должно передать в блок VPU следующие параметры:

- опция поиска I-кадров;
- режим пропуска кадров;
- dispOrderBuf;
- picStreamBufferAddr и picStartByteOffset;
- chunkSize.

После предоставления в блок VPU указанных параметров приложение, передав команду DEC_PIC_RUN, может приступить к декодированию изображения.

Если в декодере H.264 включена опция переупорядочивания отображаемых данных, то вывод первых декодированных выходных данных начнется только после декодирования нескольких кадров. Чтобы избежать подобной ситуации, на декодер H.264, который декодирует изображение в первый раз, накладывается ограничение, согласно которому необходимо заполнить все буферы переупорядоченных видеоданных. Это означает, что если величина задержки в буфере кадра, определенная на основе заголовка потока данных, равна 5, то при первом декодировании (операция DEC_PIC_RUN) декодер h.264 должен декодировать сразу 6 кадров. В дальнейшем при декодировании полного изображения на экран всегда будут выводиться выходные данные этого изображения. Для обеспечения возможности переупорядочивания отображаемых данных в формате H.264 также может использоваться особый параметр, позволяющий извлечь сохраненные результирующие данные декодера из буфера переупорядочивания отображаемых данных без необходимости декодировать изображение. Данная возможность предназначена специально для извлечения декодированного, но еще не выведенного на экран изображения, в конце процесса декодирования последовательности видеоданных. Если опция переупорядочивания отображаемых данных включена, и в буфере кадров переупорядоченных данных находятся 5 декодированных изображений, то первые результирующие видеоданные появятся на экране только после декодирования 6-го кадра. Поэтому по окончании декодирования потока данных 5 декодированных изображений не появятся на экране, даже несмотря на то, что другого битового потока, подлежащего декодированию, уже нет. В этом случае приложение может проигнорировать не выведенные на экран изображения или может отобразить их, задав параметр dispReorderBuf равным 1 и передавая команду DEC_PIC_RUN в постоянном режиме до тех пор, пока из блока VPU не будет передан индекс декодированного изображения равный -1.

Завершение декодирования изображения

Процесс декодирования изображения может занять некоторое количество времени, но пока ожидается завершение операции декодирования, приложение может продолжать выполнять другие задачи, например, обработку предварительно декодированных выходных данных. Приложением используются два способа определить, завершена ли операция декодирования изображения: опрос регистра статуса и передача сигнала о прерывании.

Извлечение результатов декодирования

Как только декодирование изображения будет завершено, хост-приложение сможет получить выходные декодированные данные по этому процессу: индекс выводимого кадра, индекс декодированного кадра, тип изображения декодированного кадра, количество макроблоков со скрытыми ошибками и т.д.

Управление отображением буферов, закодированных данных

В блоке VPU существуют флаги, указывающие, отображается или не отображается буфер кадра. Флаг устанавливается после того, как из блока VPU в автоматическом режиме будет передан индекс отображаемого кадра. В блоке VPU никогда не будет использоваться буфер с установленным флагом отображения. Прежде чем начать декодирование в блоке VPU проводится проверка того, если ли в наличии доступный буфер кадра, и в случае, если доступного для записи декодированного изображения буфера кадра нет, сразу же передается текущий индекс декодированного изображения равный -1.

Как избежать «зависания» декодера

Из-за ошибок в потоке данных или недостатка доступного потока по окончании декодирования последовательности данных приложение может столкнуться с проблемой «зависания» кодера. В середине процесса декодирования изображения в приложение через сигнал прерывания по опустошению буфера декодера (если данная опция включена) поступает сигнал о «зависании» декодера. Подобной проблемы можно избежать, если помещать в буфер потока больше данных битового потока.

1.3.6.1.2.2.4.4 Закрытие экземпляра декодера

Чтобы закрыть экземпляр декодера приложению требуется выполнение следующих условий:

- указать конец потока и последнего изображения в буфере потока данных;
- закрыть текущий экземпляр.

1.3.6.1.2.2.4.5 Команды динамического конфигурирования

Иногда в ходе процесса декодирования изображений приложению необходимо передать в блок VPU некоторые специальные команды. Интерфейс API блока VPU C&M, поддерживает специальные запросы такого типа с помощью набора команд, передаваемых из хост-приложения:

- повернуть и отразить результирующий кадр перед кодированием;
- применить набор параметров последовательности данных (SPS) и набор параметров изображения внешнего протокола внеканальной передачи.

1.3.6.1.3 ТЕХНИЧЕСКОЕ ОПИСАНИЕ

1.3.6.1.3.1 ПОРЯДОК БАЙТОВ

Для передачи данных через интерфейс с блока VPU обеспечивается доступ с выравниванием до 8 байтов через шину интерфейса AXI, а также поддерживается 4 режима порядка следования байтов. Хост-процессор может сконфигурировать режим порядка следования байтов с помощью функции интерфейса API “C&M”.

1.3.6.1.3.1.1 Операция записи

Пример ADDR = 0x7766554433221100, где ADDR[2:0] всегда соответствует 3'b000.

Если в блок VPU заносится 0x7766554433221100 по адресу A, то данные в awdata[63:0] собираются в следующем порядке:

ENDIAN_MODE = 2'b00

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
77	66	55	44	33	22	11	00

ENDIAN_MODE = 2'b01

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
00	11	22	33	44	55	66	77

ENDIAN_MODE = 2'b10

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
33	22	11	00	77	66	55	44

ENDIAN_MODE = 2'b11

63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
44	55	66	77	00	11	22	33

Current MB – Текущий макроблок

1.3.6.1.3.1.2 Обмен данными с хост-процессором

Доступ с блока VPU к внешней памяти обеспечивается для выполнения последующих функций. Порядок следования байтов в буфере можно сконфигурировать отдельно.

Доступ к буферу кадра

Порядок байтов буфера данных можно сконфигурировать отдельно и для других типов доступа к блоку внешней памяти. Более подробно данная тема рассматривается в описании взаимодействия между порядком следования байтов и расположением пикселей для разных форматов буфера кадров.

Доступ к буферу битового потока

Порядок байтов битового потока можно сконфигурировать отдельно и для других типов доступа к блоку внешней памяти.

Доступ к буферу кодов

Порядок байтов буфера кодов можно сконфигурировать отдельно и для других буферов. Перед передачей команды INIT_VPU адрес буфера кодов должен быть установлен и выровнен по разделителю до 4 кбайт. Более подробную информацию по данной теме можно найти в разделе «Руководства для программистов», посвященном последовательности инициализации блока VPU.

Доступ к буферу стека, рабочему буферу и временному буферу

Буфер стека, рабочий буфер и временной буфер указывает области внешней памяти, выделенные для VPU. Ни с хост-процессора, ни с других периферийных устройств доступа к данной области памяти нет. При доступе к этим буферам, в блоке VPU, независимо от конфигурации других буферов, используется свой собственный порядок байтов.

Внешние данные для доступа к буферу хост-процессора

Порядок байтов внешних данных для хост-процессора, например, данных из буфера пользователя и отчетной информации, можно сконфигурировать отдельно.

1.3.6.1.3.2 МАСКИРОВАНИЕ ОШИБОК

1.3.6.1.3.2.1 Методы обнаружения ошибок

Если в ходе обработки макроблока происходит ошибка декодирования с переменной длинной слова (далее VLD), то в блоке VPU “C&M” допускается, что во всех предыдущих макроблоках ошибок не было. Считается, что ошибками поражены только следующие до очередного маркера синхронизации макроблоки. Так же как в случае с ошибками VLD, декодер VPU может обнаружить ошибки в синтаксисах заголовков в ходе проверки достоверности каждого из значений. Зная предварительно определенный диапазон каждого из значений, биты маркера и стартовые коды, декодер может обнаружить ошибки также в битовом потоке. В случае обнаружения ошибок в синтаксисах заголовков, анализатор событий декодера в зависимости от уровня синтаксисов заголовков и подключенных опций может использовать различные методы обработки.

Если обнаруживаются ошибки в синтаксисах заголовков на уровне последовательностей (VOL в формате MPEG-4, SPS в формате AVC, SEQ_META в формате VC-1 и т.д.), то декодер не будет обрабатывать последующие потоки изображений. В данном случае декодер VPU может действовать двумя способами.

- Декодер VPU может попытаться найти синтаксисы заголовков на уровне очередной последовательности, продолжая обращаться к функции. Такой способ может быть полезен для приложений цифрового телевизионного вещания и видеосвязи.
- Декодер VPU может прекратить обработку. Данный способ может пригодиться в случае использования локального медиа-плейера, в котором синтаксисы верхнего уровня располагаются только в начале потока данных.

При обнаружении ошибок в синтаксисах заголовка нижнего уровня, например, в синтаксисах заголовков на уровне изображения, заголовках слайсов и т.д., декодер будет игнорировать текущий поток изображений и попытается декодировать следующий кадр. В данном случае декодер VPU может действовать двумя способами.

- Декодер VPU может попытаться найти заголовок следующего изображения, продолжая обращаться к текущей функции.
- Декодер VPU может сразу же возвратиться в исходное состояние с кодом ошибки.

1.3.6.1.3.2.2 Вложенные схемы маскирования ошибок

Схема маскирования ошибок в блоке VPU основана на концепции пропуска макроблоков. Это одна из самых простых, но, вместе с тем, и самых эффективных методик. Пропущенные макроблоки обладают разными значениями, что полностью отвечает стандартам видеокодеков, но подход к ним абсолютно одинаковый, независимо от стандартов видеокодеков. Пропускаемые макроблоки с точки зрения статистической информации являются представителями наиболее вероятного типа макроблоков. Поэтому когда происходят ошибки и декодер не может восстановить определенный макроблок, этот макроблок будет прогнозироваться по наиболее вероятному макроблоку из ссылочного кадра. Если схема маскирования ошибок применяется к ошибкам в битах, то возможны два варианта:

- замена всех макроблоков в пораженном ошибками секторе данных VLC (слайсе или кадре);
- замена всех макроблоков, начиная с позиции, в которой были обнаружены ошибки.

При обнаружении скрытых ошибок в блоке VPU для любых видеодекодеров можно выбрать второй вариант. Поэтому замещаться пропущенными макроблоками будут только те макроблоки, которые на этапе обнаружения ошибок можно причислить к макроблокам, пораженным ошибками. То есть если блоком VLD во время разбора входного потока данных

определенного макроблока обнаруживается ошибка, то все макроблоки, начиная с текущего и заканчивая последним перед очередным указателем ресинхронизации (маркер ресинхронизации, граница слайса, стартовый код и т.д.) будут замещаться пропущенными макроблоками.

- В случае использования P-кадров пропущенный макроблок можно скопировать из переднего ссылочного кадра с вектором движения 0 (для форматов PEG-2, MPEG-4, H.263) или с разностью векторов движения равной 0 (для форматов H.264, VC-1).
- В случае использования B-кадров пропущенный макроблок можно скопировать из начального ссылочного и завершающего ссылочного кадров. Обе копии будут интерполированы для формирования заключительных макроблоков предиктора.
- В случае использования I-кадра для некоторых стандартов видео, поддерживающих предсказание коэффициентов AC-DC, вместо временного прогнозирования можно использовать пространственное прогнозирование. Макроблоки с прогнозированием коэффициента DC будут использоваться в качестве предикторов макроблоков, пораженных ошибками. Если прогнозирование коэффициентов AC/DC не поддерживается, то из ссылочного кадра будут копироваться совмещенные макроблоки.

Операция декодирования запускается, начиная с первого после очередного указателя ресинхронизации макроблока. Поэтому если слайсов или пакетов данных с маркером ресинхронизации много, процесс распознавания ошибки будет значительно более эффективным. Это значит, что только небольшая часть изображения будет замаскированную ошибку. Большую часть изображения можно будет успешно восстановить. Но на такую процедуру «уйдет» некоторое количество бит, а использование чрезмерного количества слайсов или пакетов данных приведет к ухудшению качества изображения в кодере. Приняв во внимание частоту появления ошибок, размер изображения, характеристики заданного приложения и т.д., приложению следует подобрать соответствующее значение размера слайса для данного изображения.

1.3.6.1.3.2.3 Поддержка внешней схемы маскирования ошибок

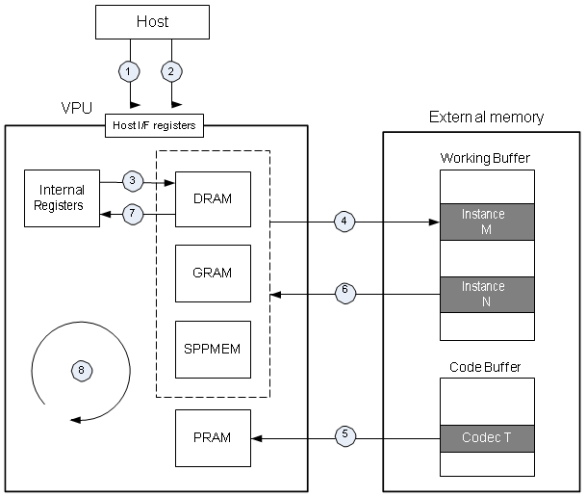
В зависимости от среды функционирования приложения, может возникнуть необходимость в применении большего числа методик маскирования ошибок в разных приложениях. Чтобы обеспечить поддержку внешних методик маскирования ошибок пользовательских приложений из блока VPU в хост-процессор передается некоторая внутренняя информация, которая может потребоваться в случае использования внешней схемы. Например, из блока VPU в хост-процессор может передаваться отчетная информация о количестве макроблоков, пораженных ошибками. На основе данной информации хост-процессор может заменить участки декодированных изображений другими участками с замаскированными ошибками.

Для более эффективного маскирования в хост-процессор может передаваться информация о векторе движения, значениях коэффициента DC в моде intra, карте распределения ошибок в макроблоке, по каждому из изображений. Этот список данных не является окончательным по умолчанию. В зависимости от потребностей, он может быть расширен. Например, если в случае разделения данных MPEG-4хост-приложение собирается использовать внешние пользовательские схемы маскирования ошибок, то из блока VPU в хост-процессор могут передаваться отчетные данные обо всех параметрах, не имеющих ошибок, например, вектор движения, коэффициент DC в моде intra и т.д. Это можно сделать с хост-процессора с помощью карты распределения ошибок, информации о слайсах битового потока и упомянутых внутренних параметров.

1.3.6.1.3.3 КОНФИГУРИРОВАНИЕ ПРИ ИСПОЛЬЗОВАНИИ НЕСКОЛЬКИХ ЭКЗЕМПЛЯРОВ

Блок VPU может работать с несколькими экземплярами. Данная особенность может использоваться при многоканальном декодировании. Чтобы обеспечить работу системы в режиме с несколькими экземплярами в блоке VPU предусмотрена опция переключения экземпляров по команде. Если используемый экземпляр отличается от экземпляра, который выполнялся в последний раз, то перед обработкой каждой команды происходит переключение экземпляра. Каждый экземпляр идентифицируется регистром главного интерфейса BIT_RUN_INDEX в диапазоне от 0 до 65534.

При переключении экземпляра контексты последнего из выполненных экземпляров сохраняются во внешней памяти, контекст текущего экземпляра загружается из внешней памяти. Контекст включает в себя внутренние регистры и внутреннюю память, доступ к которой обеспечивается по команде. Область внешней памяти, начиная с регистра главного интерфейса BIT_WORK_BUF_ADDR, сегментируется и присваивается каждому из экземпляров. Если стандарт кодека, определенный регистром главного интерфейса BIT_RUN_COD_STD и BIT_RUN_AUX_STD, поменялся, то блоком VPU перед загрузкой контекста текущего экземпляра в память PRAM должна быть загружена программа, находящаяся во внешней памяти.



Host – Хост-компьютер
VPU – Блок VPU
Host registers – регистры I/F хоста
Internal Registers – Внутренние регистры
External memory – Внешняя память
Working Buffer – Рабочий буфер
Instance M – Экземпляр M
Instance M – Экземпляр M
Code Buffer – Буфер кодов
Codec T – Кодек T

Рисунок E.1 - Конфигурирование при использовании нескольких экземпляров

- Хост по команде заносит параметры в регистры главного интерфейса.
- Хост заносит команду в регистр главного интерфейса.
- VPU размещает в памяти DRAM внутренние регистры.
- VPU размещает в рабочем буфере модули внутренней памяти.
- Если стандарт кодека изменился, то VPU восстанавливает PRAM из буфера кодов.

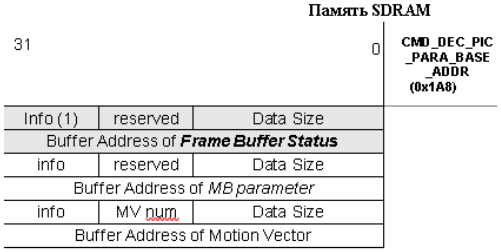
- VPU восстанавливает из рабочего буфера модули внутренней памяти.
- VPU восстанавливает из памяти DRAM внутренние регистры.
- VPU выполняет команду, определенную на этапе 1.

1.3.6.1.3.4 ПРЕДОСТАВЛЕНИЕ ВНЕШНИХ ДАННЫХ

Данная глава посвящена описанию опции предоставления промежуточных данных декодера. В процессе декодирования последовательности битовых потоков в блок VPU должен поступить некоторый промежуточный параметр, который служит для управления оборудованием VPU. Для предоставления этих промежуточных данных в VPU предусмотрена специальная функция. VPU записывает промежуточные данные в память хоста. После этого хост сможет «видеть», что происходит внутри VPU при декодировании/кодировании последовательности битовых потоков. Существует три категории данных, записываемых VPU при декодировании: статус буфера кадров, информация о макроблоке и данные пользователя. На стороне хоста с помощью установки и сброса флагов регистров VPU можно включать и отключать опцию предоставления промежуточных данных. Более подробную информацию об интерфейсе приложения можно найти в «Руководстве для программистов» и в «Справочном руководстве к API».

1.3.6.1.3.4.1 Способы использования данной опции

Для каждого из этапов установки команд декодирования изображений хост с помощью особого установочного регистра в VPU, может сконфигурировать область отчета: CMD_DEC_PIC_PARA_BASE_ADDR (см. главу «Регистры главного интерфейса» «Руководства для программистов»). Из структуры данных внешней памяти в блок VPU поступает информация о том, где располагается область буфера параметров.



Info (1) – Инфо (1)
Reserved – зарезервированы
Data Size – Объем данных
Buffer Address of Frame Buffer Status – Адрес буфера для статуса буфера кадра
Info – Инфо
Reserved – зарезервированы
Data Size – Объем данных
Buffer Address of MB parameter – Адрес буфера параметра макроблока
Info – Инфо
MV num – Число векторов движения
Data Size – Объем данных
Buffer Address of Motion Vector – Адрес буфера вектора движения

Рисунок 1.118 - Конфигурация опции предоставления отчетных данных

Чтобы ВКЛЮЧИТЬ/ОТКЛЮЧИТЬ (ON/OFF) опцию предоставления отчетных данных на стороне хоста нужно установить/сбросить флаги в специальном регистре VPU: CMD_DEC_PIC_OPTION[8:5] (см. «Руководство для программистов»)

У каждого поля «Info» есть свой уникальный идентификационный номер (тип данных – данные параметра):

PARA_TYPE_FRM_BUF_STATUS = 1
PARA_TYPE_MB_PARA = 2
PARA_TYPE_MV = 4

Если в поле «Инфо» стоит 0, то это означает, что для данного сектора данных отчетной информации нет.

1.3.6.1.3.4.2 Отчет о статусе буфера кадров

В результирующих данных, предоставляемых по завершении декодирования каждого кадра, содержится информация о статусах всех буферов кадра. VPU может обеспечивать управление индексами буферов кадра более чем в 32 слотах. В ходе декодирования каждого кадра VPU обновляет статус буфера для индекса каждого кадра. Статус буфера отображается с помощью 4 флагов согласно приведенному ниже описанию.

Формат отчетности

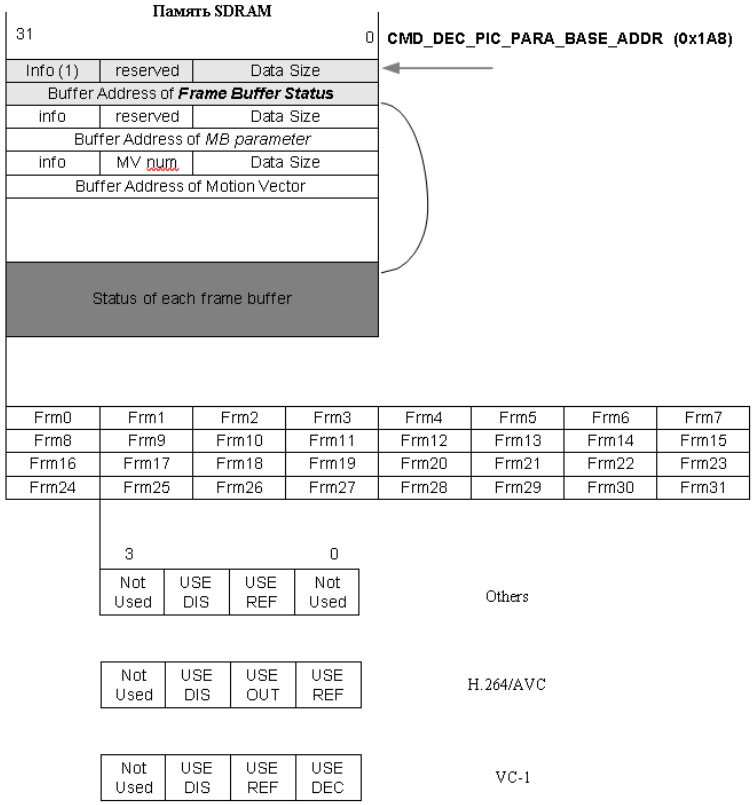
Каждый буфер кадра может использоваться для отображения, в качестве ссылочного или не использоваться совсем. Декодер предоставляет данные о статусе буфера каждого кадра с помощью следующих 4 бит. [3:0] =

- [3] не используется
- [2] USE_DIS
- [1] USE_REF
- [0] не используется

Например, если значение статуса буфера кадра равно 6, то буфер кадра используется в качестве ссылочного и для отображения. При значении статуса 4 буфер кадра используется для отображения и не используется в качестве ссылочного. В формате H.264 битовое поле определяется следующим образом. [3:0] =

- [3] не используется
- [2] USE_DIS
- [1] USE_OUT
- [0] USE_REF

Если USE_OUT равно 1, то это означает, что данный кадр находится в буфере DPB.



Info (1) – Инфо (1)
Reserved – зарезервированы
Data Size – Объем данных
Buffer Address of Frame Buffer Status – Адрес буфера для статуса буфера кадра
Info – Инфо
Reserved – зарезервированы
Data Size – Объем данных
Buffer Address of MB parameter – Адрес буфера параметра макроблока
Info – Инфо
MV num – Число векторов движения
Data Size – Объем данных
Buffer Address of Motion Vector – Адрес буфера вектора движения
Status of each frame buffer – Статус каждого буфера кадра
Not Used – Не используется
Not Used – Не используется
Not Used – Не используется
Not Used – Не используется
Others - Прочие

Рисунок 1.119 - Матрица представления флага статуса буфера кадра

1.3.6.1.3.5 ПАРАМЕТРЫ КОДИРОВАНИЯ

VPU предоставляет файл конфигурирования (*.cfg) в эталонной С-модели, позволяющей пользователю задавать параметры и логику кодирования блока VPU.

1.3.6.1.3.5.1 Стандарт H.264/AVC

Таблица 1.151 – Параметры кодирования в стандартах H.264/AVC

Название	Описание
Исходные данные	
YUV_SRC_IMG PICTURE_WIDTH	Источник изображения, подлежащего кодированию Ширина изображения Мин.: 96 Макс.: 4096 То есть, ширина изображения даже после его разворота преротатором, не должна быть меньше 96. Например, если ширина развернутого изображения равна 64, то кодирование может оказаться невозможным.
PICTURE_HEIGHT	Высота изображения Мин.: 16 Макс.: 1088
INTERLACED_PIC	0: чересстрочный режим отключен (режим кадра) 1: чересстрочный режим включен
CHROMA_FORMAT_400	0: 4:2:0 1: 4:0:0 (режим MONO, доступный только в HP)
Управление кодером	
FRAME_NUMBER_ENCODED	Количество кадров, подлежащих кодированию
GOP_PIC_NUMBER	Количество групп изображений (далее GOP) 0: только первый - I 1: все - I 3: I, P, P, I, …
FRAME_RATE	Частота кадров
PIC_QP_Y	Значение параметра квантизации (далее QP) (0 ~ 51)
CHROMA_QP_OFFSET	chroma_qp_index_offset (p81, ITU-T H.264) с -12 ~ 12
Управление прогнозированием	
TRANSFORM_8X8_MODE	Включение опции преобразования 8x8 0: включено 1: отключено
SEARCH_RANGE_X	Диапазон поиска ME 0: -64 ~ 63
	1: -48 ~ 47
	2: -32 ~ 31
	3: -16 ~ 15
	H.263 (короткий заголовок: всегда 0)
SEARCH_RANGE_Y	Диапазон поиска ME
	0 : -48 ~ 47
	1 : -32 ~ 31
	2 : -16 ~ 15
	H.263 (короткий заголовок: всегда 0)
CONSTRAINED_INTRA	constrained_intra_pred_flag (p82, ITU-T H.264) с 0 или 1
ME_USE_ZERO_PMV	0: использовать PMV 1: не использовать PMV
Управление энтропийным кодированием	

Продолжение таблицы 1.151

ENTROPY_CODING_MODE	Режим энтропийного кодирования 0: CAVLC 1: CABAC
CABAC_INIT_IDC	Исходное значение idc CABAC (0 ~ 2)
Фильтр деблокирования	
DISABLE_DEBLK	Отключение фильтра деблокирования 0: включено 1: отключено 2: отключено на разделителе слайсов
DEBLK_ALPHA	slice_alpha_c0_offset_div2 (p90, ITU-T H.264) с -6 ~ 6
DEBLK_BETA	slice_beta_offset_div2 (p91, ITU-T H.264) с -6 ~ 6
Структура слайса	
SLICE_MODE	Режим слайса 0: один слайс 1: несколько слайсов
SLICE_SIZE_MODE	Режим размера слайса 0: количество битов слайса 1: количество макроблоков слайса
SLICE_SIZE_NUMBER	Количество битов или макроблоков в слайсе Если SLICE_SIZE_MOE установлен в 0, то это значение показывает количество битов в одном слайсе. Если SLICE_SIZE_MOE установлен в 1, то это значение отражает количество макроблоков в одном слайсе.
Контроль скорости передачи	
RATE_CONTOL_ENABLE	Флаг функции контроля скорости передачи 0: отключено 1: включено
BIT_RATE_KBPS	Битовая скорость, измеренная в килобитах в секунду (если функция контроля скорость отключена, то данное значение игнорируется)
DELAY_IN_MS	*Величина задержки в буфере декодера (0: игнорируется)*
VBV_BUFFER_SIZE	*Размер буфера для эталонного декодера, измеренный в битах (0: игнорируется)*
Устройчивость к ошибкам	
INTRA_MB_REFERESH	*0: отключено* *N: принудительная работа каждого N макроблока в моде intra*

1.3.6.1.3.5.2 Стандарт MPEG-4

Таблица 1.152 – Параметры кодера MPEG-4

Название	Описание
Общие параметры	
FRAME_NUMBER_ENCODED	Количество кадров, подлежащих кодированию
PICTURE_WIDTH	Ширина изображения Мин.: 96 Макс.: 1920

Продолжение таблицы 1.152

Название	Описание
PICTURE_HEIGHT	Высота изображения Мин.: 16 Макс.: 1088
FRAME_RATE	Частота кадров
DATA_PART_ENABLE	включение/отключение опции разделения данных 0: отключено 1: включено
REV_VLC_ENABLE	включение/отключение реверсивного кодирования VLC 0: отключено 1: включено
INTRA_DC_VLC_THRES	intra_dc_vlc_thr 0: применяется кодирование VLC коэффициента DC в моде intra для всего VOP 1 ~ 6: применяется кодирование VLC коэффициента AC 7: применяется кодирование VLC коэффициента AV в моде intra для всего VOP
Параметры стандарта H.263	
SHORT_VIDEO	Поддерживаются ли короткие заголовки видеоданных 0: отключено 1: включено
ANNEX_I_ENABLE	Поддерживается ли ПРИЛОЖЕНИЕ I 0: отключено 1: включено
ANNEX_J_ENABLE	Поддерживается ли ПРИЛОЖЕНИЕ J 0: отключено 1: включено
ANNEX_K_ENABLE	Поддерживается ли ПРИЛОЖЕНИЕ K 0: отключено 1: включено
ANNEL_T_ENABLE	Поддерживается ли ПРИЛОЖЕНИЕ T 0: отключено 1: включено
Плоскость видео-объекта (далее VOP)	
VOP_QUANT_SCALE	Режим слайсов
GOP_PIC_NUMBER	Количество GOP 0: запрещено 1: все I-изображения 3: изображения I,P,P,I
Структура слайса	
SLICE_MODE	0: один слайс 1: несколько слайсов
SLICE_SIZE_MODE	Режим размера слайса 0: количество битов слайса 1: количество макроблоков слайса

Продолжение таблицы 1.152

Название	Описание
SLICE_SIZE_NUMBER	Количество битов или макроблоков в слайсе Если SLICE_SIZE_MOE установлен в 0, то это значение показывает количество битов в одном слайсе. Если SLICE_SIZE_MOE установлен в 1, то это значение отражает количество макроблоков в одном слайсе.
Контроль скорости передачи	
RATE_CONTOL_ENABLE	Флаг функции контроля скорости передачи 0: отключено 1: включено
BIT_RATE_KBPS	Битовая скорость, измеренная в килобитах в секунду (если функция контроля скорости отключена, то данное значение игнорируется)
DELAY_IN_MS	Величина задержки в буфере декодера (0: игнорируется)
VBV_BUFFER_SIZE	Размер буфера для эталонного декодера, измеренный в битах (0: игнорируется)
Устойчивость к ошибкам	
INTRA_MB_REFERESH	Регенерация макроблоков в режиме Intra 0: ни одного 1 ~: количество макроблоков - 1
Оценка параметров движения	
SEARCH_RANGE_X	Диапазон поиска для выполнения ME 0: -64 ~ 63 1: -48 ~ 47 2: -32 ~ 31 3: -16 ~ 15
	Стандарт H.263 (короткий заголовок: всегда 0)
SEARCH_RANGE_Y	Диапазон поиска для выполнения ME 0: -48 ~ 47 1: -32 ~ 31 2: -16 ~ 15 Стандарт H.263 (короткий заголовок: всегда 0)
ME_USE_ZERO_PMV	0: использовать PMV 1: не использовать PMV
WEIGHT_INTRA_COST	Задает значение веса для выбора режима Inter
Дополнительные параметры	
CONST_INTRA_QP_EN	Включение/отключение постоянного параметра квантизации кадра в моде intra 0: отключено 1: включено
CONST_INTRA_QP HEC_ENABLE	0: использовать PMV 1: не использовать PMV Включение/отключение опции увеличения заголовка пакета

1.3.6.1.3.6 Приложение Н. КОНТРОЛЬ СКОРОСТИ ПЕРЕДАЧИ

Функция контроля скорости передачи применяется для стабилизации объема выходного битового потока для достижения CBR (постоянной битовой скорости). В процессе кодирования генерируются видеосигналы, которые передаются по сети в декодер. При этом объем битов, передаваемых за секунду (битовая скорость), должен быть постоянным, поскольку пропускная способность сети, как правило, ограничена. С другой стороны, мощность сигналов YUV (энтропия) в видеокодере на входе может быть чрезвычайно непостоянной. Если все кадры квантизованы с одинаковым параметром квантизации (значением QP), то для каждого кадра кодер производит битовые потоки разного объема. Поэтому перед началом обработки декодер в течение еще некоторого времени вынужден продолжать буферизацию, а это со временем может привести к сильной задержке в отображении видеопотока.

Чтобы справиться с подобной проблемой, с которой сталкиваются многие кодеры, предлагается воспользоваться алгоритмом контроля скорости передачи. Алгоритм контроля скорости передачи обеспечивает возможность при соответствующей настройке параметра QP (или с помощью других методик) генерировать битовые потоки постоянного объема на выходе кодера, что позволит сократить разрыв в мощности сигналов между сжатыми кадрами. Однако качество изображения одного кадра может отличаться от качества изображения другого. В заключении стоит отметить, что функция контроля скорости предназначена не только для поддержания приемлемой величины задержки в передаче, а также для контроля даже минимальных изменений качества изображения.

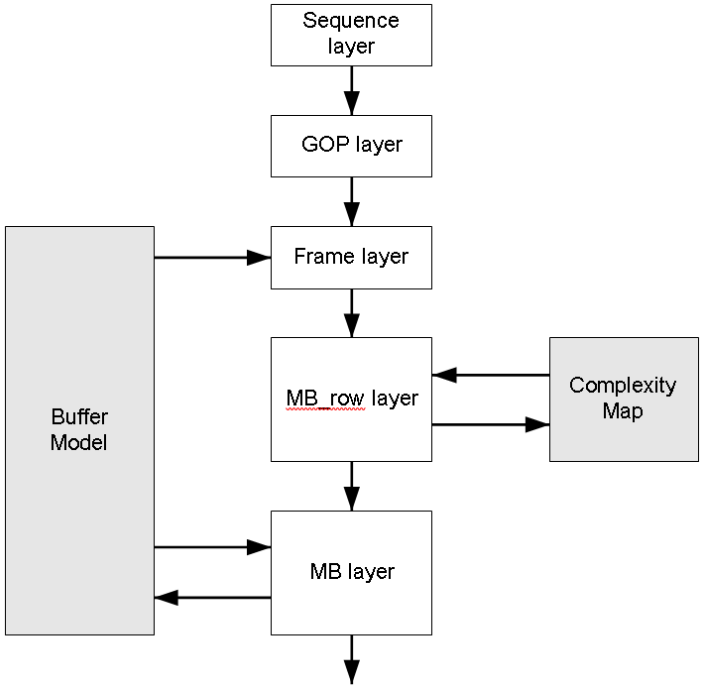
Алгоритм контроля передачи, применяемый в блоке VPU, построен на модели HRD (гипотетически эталонного декодера). На основании данных модели HRD кодер «узнает», поступил ли битовый поток из кодера в декодер в момент декодирования соответствующего изображения, и сколько данных осталось в буфере битового потока декодера. Если модель HRD показывает недозаполнение, это значит, что в декодере больше нет потоков, и вследствие того, что декодер не может продолжать декодирование и вынужден находить с режиме ожидания следующего битового потока, может возникнуть задержка или ошибка.

1.3.6.1.3.6.1 Схема контроля скорости передачи

Система контроля скорости передачи в VPU образует 5-уровневую иерархическую структуру:

- уровень последовательности,
- уровень GOP,
- уровень кадра,
- уровень строки макроблока,
- уровень макроблока.

На верхнем уровне данной структуры выбирается скорость передачи данных для последовательности, GOP, кадра и т.д., а на нижнем уроне происходит стабилизация скорости передачи данных в соответствии с заданной скоростью передачи данных с верхнего уровня.



Sequence layer - Уровень последовательности
GOP Layer - Уровень GOP
Frame Layer - Уровень кадра
MB_row Layer - Уровень ряда макроблока
MB - Уровень макроблока
Buffer Model – Модель буфера
Complexity Map – Карта уровней сложности

Рисунок 1.120 - Схема контроля скорости передачи

- Самым верхним уровнем является уровень последовательности. На данном уровне задаются некоторые параметры модели HRD: битовая скорость для всей последовательности, величина задержки и т.д.
- Вторым сверху уровнем является уровень GOP (группы изображений). На уровне GOP определяется целевой объем битов для текущей GOP. До того момента пока стабильное изображение в моде intra не начнет отображаться через постоянные интервалы времени, уровень GOP будет пропускаться.
- Средним уровнем является уровень кадра. На уровне кадра определяется целевой объем текущего кадра. Битовая скорость может распределяться из заданного объема битов GOP и с другим весом, в зависимости от типа кадра, потому что обычно I-кадр состоит из большего числа бит, чем P-кадр, а качество I- и P-изображений одинаковое (их QP почти одинаковый).
- При распределении бит на уровне кадра используются преимущества модели буфера HRD. Если один кадр забирает из объема битов, предназначенных для GOP, слишком большое количество бит, то работа буфера может быть нарушена. Поэтому на данном уровне распределение битового потока кадров должно выполняться очень гибко и с учетом типа и буфера кадра. В случае распределения потока полукадров на данном уровне также определяется целевая битовая скорость для изображения полукадра.
- Вторым снизу уровнем является уровень ряда макроблоков. Ряд макроблоков обозначает группу макроблоков в одной строке. На данном уровне рядов макроблоков на основании вычислений по карте уровней сложности из заданного объема битов кадра выделяется объем битов. Карта уровней сложности – это

массив коэффициентов сложности предыдущего кадра в блоке рядов, предназначенная для установления различий между областью повышенной сложности и областью особой сложности в рамках одного кадра. Но в случае изменения сцены использование такой карты уровней сложности может оказаться нецелесообразным, поскольку текущий кадр с предыдущим кадром никак не связан.

- Самым нижним уровнем является уровень макроблока. На уровне макроблока распределения бит не происходит. На этом уровне проводится проверка скорости передачи сгенерированных бит и настройка посредством изменения значения параметра QP общей скорости передачи бит в одном ряду макроблоков в соответствии с заданным для ряда макроблоков объемом бит.

1.3.6.1.3.6.2 Конфигурация

Такие параметры как величина задержки, длина GOP и использование опции CIR (циклическая регенерация макроблоков Intra) могут оказывать влияние на эффективность контроля скорости передачи. В представленной ниже таблице приведены сравнительные данные по готовности, эквивалентности качества в GOP и эквивалентности качества изображения при изменении трех параметров.

Таблица 1.153 – Сравнительный анализ эффективности контроля скорости передачи данных

		GOP == 0		GOP == N	
		Без CIR	CIR	Без CIR	CIR
delay < 1 frame time	Availability	O	O	X	O
	Quality equivalance in GOP	X	X	X	X
	Quality equivalance in picture	X	X	X	X
1 GOP time > delay & delay ≥ 1 frame time	Availability	O	O	O	O
	Quality equivalance in GOP	X	X	X	X
	Quality equivalance in picture	O	O	O	O
delay ≥ 1 GOP time	Availability	O	O	O	O
	Quality equivalance in GOP	O	O	O	O
	Quality equivalance in picture	O	O	O	O

Delay < 1 frame time – Величина задержки < времени передачи 1 кадра
Availability – Готовность
Quality equivalance in GOP – Эквивалентность качества в GOP
Quality equivalance in picture – Эквивалентность качества изображения
1 GOP time > delay & delay ≥ 1 frame time – Время передачи 1 GOP > величины задержки, а величина задержки ≥ времени передачи 1 кадра
Availability – Готовность
Quality equivalance in GOP – Эквивалентность качества в GOP
Quality equivalance in picture – Эквивалентность качества изображения
delay ≥ 1 GOP time – Величина задержки ≥ времени передачи 1 GOP
Availability – Готовность
Quality equivalance in GOP – Эквивалентность качества в GOP
Quality equivalance in picture – Эквивалентность качества изображения

Из этого следует, что с уменьшением времени задержки эквивалентность качества изображения ухудшается. Обычно при увеличении объема буфера декодера величина задержки увеличивается, поэтому, нельзя получить сжатое видео стабильно хорошего качества при небольшом буфере декодера.

Рекомендуется устанавливать величину минимальной задержки, равную периоду передачи GOP. Например, если интервал между I-кадрами (GOP) равен 30 при скорости передачи 30

кадров в секунду, то рекомендуется установить величину задержки 1000 миллисекунд. Но если задать величину задержки меньше периода передачи GOP, то нельзя гарантировать, что кодер VPU сможет сгенерировать битовый поток при CBR.

В таблице 1.154 приведены рекомендованные минимальные значения скорости передачи бит для каждого размера изображения.

Таблица 1.154 – Минимальная рекомендованная битовая скорость

Размер изображения	Ширина	Высота	Количество макроблоков	Минимальная рекомендованная битовая скорость
QCIF	176	144	99	64
QVGA	320	240	300	200
CIF	352	288	396	200
VGA	640	480	1200	400
D1, SD	720	480	1350	400
HD	1280	720	3600	600
1K	960	540	2025	500
FHD, 2K	1920	1080	8100	1000

Можно установить значение скорости передачи бит ниже рекомендованного, но в этом случае не гарантируется, что функция контроля скорости передачи будет правильно работать. Даже несмотря на то, что ограничений по величине максимальной скорости передачи битов не существует, не следует устанавливать слишком большое значение битовой скорости, потому что в этом случае кодер может просто не достичь этого значения.

Также линейное преобразование. В среднем значение количества байт на один макроблок составляет 771 байт.

1.3.6.1.3.7 ТРЕБОВАНИЯ К ПРОПУСКНОЙ СПОСОБНОСТИ ПРИ КОДИРОВАНИИ H.264

Данная глава посвящена описанию требований к пропускной способности и моделей доступа, необходимых для кодирования H.264 1080p 30 к/с. Приняты следующие условия:

Параметры, влияющие на модель доступа: формат буфера кадра, режим линейного буфера и диапазон поиска компоненты Y

- Чтобы уменьшить пропускную способность памяти аппаратного модуля компенсации движения линейный буфер объемом 67.5 кбайт подключается к блоку VPU.
- Диапазон поиска при вертикальном сканировании составляет +/-16. Диапазон поиска при горизонтальном сканировании не связан с пропускной способностью памяти.
- Иходные кадры, подлежащие кодированию, хранятся в линейном формате с чередованием компонент Cb/Cr.
- Восстановленные и ссылочные кадры хранятся в буфере кадров виде тайлов с перемежающимися компонентами Cb/Cr.
- Дополнительная память SRAM, применяемая для уменьшения пропускной способности памяти, подключается через вспомогательный интерфейс AXI. Более подробную информацию по данной теме можно найти в разделе J.2 “Дополнительная память SRAM, подключаемая по вспомогательной шине AXI”.

На рисунке - 1.121 “Поток данных кодирования H.264” представлен поток данных, идущий в направлении от/к внешней памяти. Шина AXI на данном рисунке – это основная шина AXI. Вспомогательная шина AXI не указана.

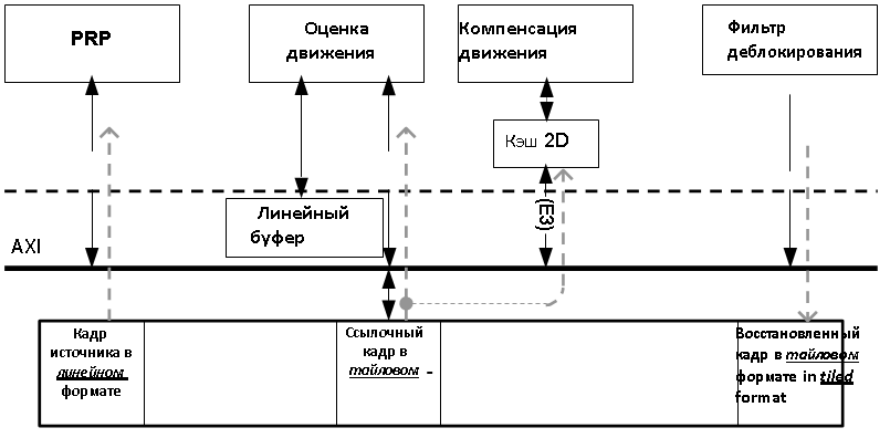


Рисунок 1.121 - Поток данных кодирования H.264

1.3.6.1.3.7.1 Дополнительная память SRAM для вспомогательного AXI

Чтобы уменьшить пропускную способность, к блоку VPU по вспомогательной шине AXI можно подключить дополнительную память. В расположенной ниже таблице представлены значения объемов памяти, необходимые для каждого типа данных, передаваемых по вспомогательной шине AXI. Чтобы уменьшить TBD, выраженную в Мбайт/с, при кодировании H.264 1080p 30 к/с требуется TBD, выраженную в кбайтах.

Таблица 1.155 - Дополнительная SRAM для кодирования H.264 1080p 30 к/с

Данные	Требуемый объем памяти (в байтах)	Экономия полосы пропускания (в Мбайт/с)

Данные макроблоков от/к BPU	15 360	62,67
Пиксели, восстановленные в результате прогнозирования в моде intra	7 680	31,33
Промежуточные данные аппаратного модуля фильтра деблокирования	15 360	62,67
Итого	38 400	156,67

Примечание Объем необходимой памяти не зависит от высоты изображения и скорости передачи кадров. Следовательно, для кодирования H.264 1080p 60 к/с требуется такой же объем памяти, как и для кодирования H.264 1080p 30 к/с.

1.3.6.1.3.7.2 Заключение

В таблице 1.156 представлены общие требования к пропускной способности, необходимой для кодирования 1080 30 к/с, при условии соблюдения следующих условий:

- Объем линейного буфера должен быть достаточно большим для размещения всех ссылочных пикселей в сконфигурированном диапазоне поиска (кроме значения 67,5 кбайт в диапазоне (+/32H,+/-16V).
- Использование кэша 2D поможет обеспечить 30-процентную экономию исходной пропускной способности благодаря компенсации движения при интерполяции пикселей цветности.

Таблица 1.156 – Общие требования в пропускной способности для кодирования 1080p 30 к/с

Данные	Требования к пропускной способности (в Мбайт/с)	Примечание
Для считывания исходного кадра с помощью PRP	90	
Для считывания ссылочных пикселей яркости посредством оценки параметров движения	60	
Для считывания ссылочных пикселей цветности посредством компенсации движения	38	См. G.3.Case 1. Sun-flower 1080p 40 Мбайт/с 30 к/с
Для считывания отфильтрованных пикселей с помощью фильтра деблокирования	90	
Итого	278	

Благодаря использованию дополнительной памяти объемом 38400 байт, другие модели доступа при 156,67 Мбайт/с, отличные от приведенных в данной таблице, можно исключить из требований к пропускной способности внешней памяти. Если пользователь не желает использовать дополнительную память, то к итоговому значению таблицы необходимо добавить еще 156,67 Мбайт/с.

Примечание Требования к пропускной способности при записи закодированного битового потока в данной главе не рассматриваются. То есть, если битовая скорость равна 40 Мбит/с, то пропускная способность будет равна 5 Мбайт/с

1.3.6.1.3.8 ТРЕБОВАНИЯ К ПРОПУСКНОЙ СПОСОБНОСТИ ДЛЯ ДЕКОДИРОВАНИЯ H.264

Данная глава посвящена описанию требований к пропускной способности и моделей доступа, необходимой для декодирования H.264 1080p Высокого профиля. Приняты следующие условия:

Восстановленные и ссылочные кадры хранятся в буфере кадров виде тайлов.

Дополнительная память SRAM объемом 52,5 кбайт, применяемая для уменьшения пропускной способности памяти, подключается через вспомогательный интерфейс AXI.

На рисунке 1.122 - “Поток данных декодирования” представлен поток данных, идущий в направлении от/к внешней памяти.

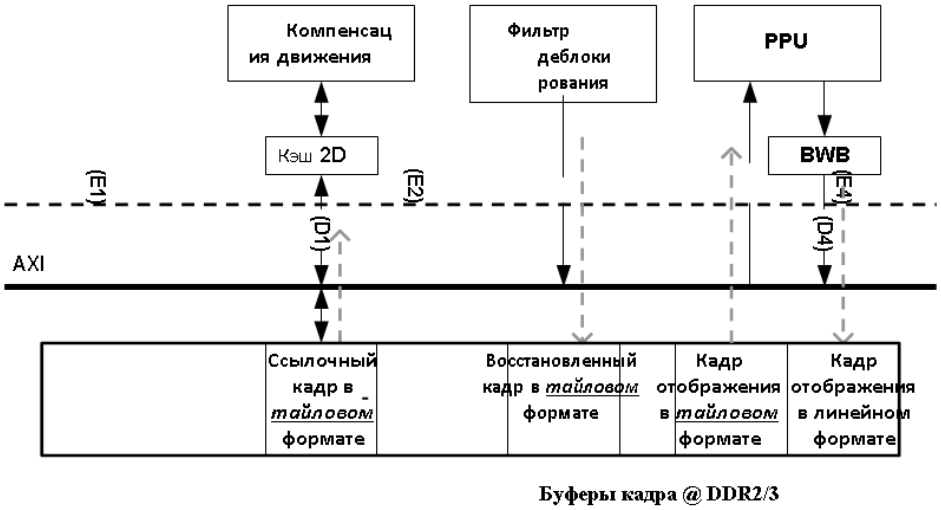


Рисунок 1.122 - Поток данных декодирования

1.3.6.1.3.8.1 Дополнительная память SRAM для вспомогательного AXI

Чтобы уменьшить пропускную способность, через дополнительную вспомогательную шину AXI блок VPU можно подключить к внутренней памяти. Если система сможет предоставить внутреннюю память SRAM для временного хранения данных фильтра деблокирования, то общая пропускная способность снизится. В зависимости от конфигурации доступ к этой памяти можно организовать или по главной, или по вспомогательной шине интерфейса AXI.

В таблице 1.157 указаны требуемые значения для объема памяти (SRAM) и величина экономии полосы пропускания для каждого стандарта, поддерживающего формат HD (1920x1088).

Таблица 1.157 – Использование интерфейса AXI по вспомогательной шине для хранения данных аппаратного модуля

стандарт	профиль/уровень		Объем буфера	Тип временного буфера
H.264	BP	Объем SRAM (в байтах)	38 400	Данные макроблока +прогноз в моде Intra + деблокирование
стандарт	профиль/уровень		Объем буфера	Тип временного буфера
	MP/HP	Экономия полосы пропускания (Мбит/с) Объем SRAM (в байтах)	149,41 53 760	Данные макроблока +прогноз в моде Intra + деблокирование Данные макроблока +прогноз в моде Intra + деблокирования
		Экономия полосы пропускания (Мбит/с)	149,41	Данные макроблока +прогноз в моде Intra + деблокирования
MPEG-4		Объем SRAM (в байтах)	30 720	Данные макроблока +прогноз коэффициентов AC/DC (+внеканальное деблокирование)
		Экономия полосы пропускания (Мбит/с)	119,53	Данные макроблока +прогноз коэффициентов AC/DC (+внеканальное деблокирование)
H.263P3	Объем SRAM (в байтах) Экономия полосы пропускания (Мбит/с)		46 080 179,30	Данные макроблока +прогноз коэффициентов AC/DC + деблокирование Данные макроблока +прогноз коэффициентов AC/DC + деблокирование
VC-1	MP (с перекрыванием)	Объем SRAM (в байтах)	67 200	Данные макроблока +прогноз коэффициентов AC/DC + перекрывание +деблокирование
		Экономия полосы пропускания (Мбит/с)	252,51	Данные макроблока +прогноз коэффициентов AC/DC + перекрывание +деблокирование
MPEG-2	AP (с перекрыванием) Объем SRAM (в байтах)		97 920	Данные макроблока +прогноз коэффициентов AC/DC + перекрывание +деблокирование
		Экономия полосы пропускания (Мбит/с) Объем SRAM (в байтах)	372,04 15 360	Данные макроблока +прогноз коэффициентов AC/DC + перекрывание +деблокирование (+внеканальное деблокирование)
		Экономия полосы пропускания (Мбит/с)	59,77	(+внеканальное деблокирование)
AVS		Объем SRAM (в байтах)	38 400	Данные макроблока +прогноз в моде Intra + деблокирование
		Экономия полосы пропускания (Мбит/с)	149,41	Данные макроблока +прогноз в моде Intra + деблокирование
RV	Объем SRAM (в байтах) Экономия полосы пропускания (Мбит/с)		53 760 209,18	Данные макроблока +прогноз в моде Intra + деблокирование Данные макроблока +прогноз в моде Intra + деблокирование
VP8		Объем SRAM (в байтах)	38 400	прогноз в моде Intra + деблокирование
		Экономия полосы пропускания (Мбит/с) МАКС. Объем SRAM (в байтах)	149,41 97 920	прогноз в моде Intra + деблокирование

Примечание Более подробную информацию, касающуюся пропускной способности, можно найти в файле Optional_SRAM_BW_Saving.xls.

В таблице 1.158, являющейся дополнением к таблице 1.157, приводится объем памяти, необходимый для декодирования H.264 1080p 30 к/с. Каждое из значений этой таблицы можно передать по вспомогательной шине AXI. Чтобы обеспечить экономию полосы пропускания 156,89 Мбайт/с при декодировании H.264 1080p 30 к/с, требуется 40 320 байт.

Таблица 1.158 – Дополнительная SRAM для декодирования H.264 1080p 30 к/с

Данные	Требуемый объем памяти (в байтах)	Экономия полосы пропускания (в Мбайт/с)
Данные макроблоков от/к BPU	17 280 7 680	67,24 29,88
Пиксели, восстановленные в результате прогнозирования в моде intra	15 360	59,77
Промежуточные данные аппаратного модуля фильтра деблокинга	40 320	156,89

Причиной того, почему аппаратному модулю фильтра деблокирования для кодирования требуется в два раза больший объем памяти, является то, что MBAFF для кодирования требуется в два раза больше промежуточных данных, подлежащих хранению в дополнительной памяти.

Примечание Объем необходимой памяти не зависит от высоты изображения и скорости передачи кадров. Следовательно, для кодирования H.264 1080p 60 к/с требуется такой же объем памяти, как и для кодирования H.264 1080p 30 к/с.

1.3.6.1.3.8.2 Заключение

В таблице 1.159 представлены общие требования к пропускной способности, необходимой для декодирования 1080p 30 к/с.

Таблица 1.159 – Общие требования в пропускной способности для декодирования 1080p 30 к/с

Данные	Требования к пропускной способности (в Мбайт/с)	Примечание
Для считывания ссылочных пикселей посредством компенсации движения	240	Sunflower_4B_2ref_WP_ 40Mbps.264 с размером строки кэша = 8x8 для длины пакета 64 байта /128 байт
Для считывания отфильтрованных пикселей с помощью фильтра деблокирования	90	
Для считывания отформатированного по тайлам кадра с помощью PPU	90	Для перехода тайл-строка
Для считывания линейно отформатированного кадра с помощью BWB	90	Для перехода тайл-строка
Итого	330+180	Для перехода тайл-строка

Благодаря использованию дополнительной памяти объемом 53 760 байт, другие модели доступа при 219,34 Мбайт/с, отличные от приведенных в данной таблице, можно исключить из требований к пропускной способности внешней памяти. Если пользователь не желает использовать дополнительную память, то к итоговому значению таблицы необходимо добавить еще 219,34 Мбайт/с.

Примечание Требования к пропускной способности при записи закодированного битового потока в данной главе не рассматриваются. То есть, если битовая скорость равна 40 Мбит/с, то пропускная способность будет равна 5 Мбайт/с.

1.3.6.1.3.9 ПРОФИЛИРОВАНИЕ ЭФФЕКТИВНОСТИ

В данной главе описывается процесс профилирования эффективности в ходе кодирования и декодирования H.264.

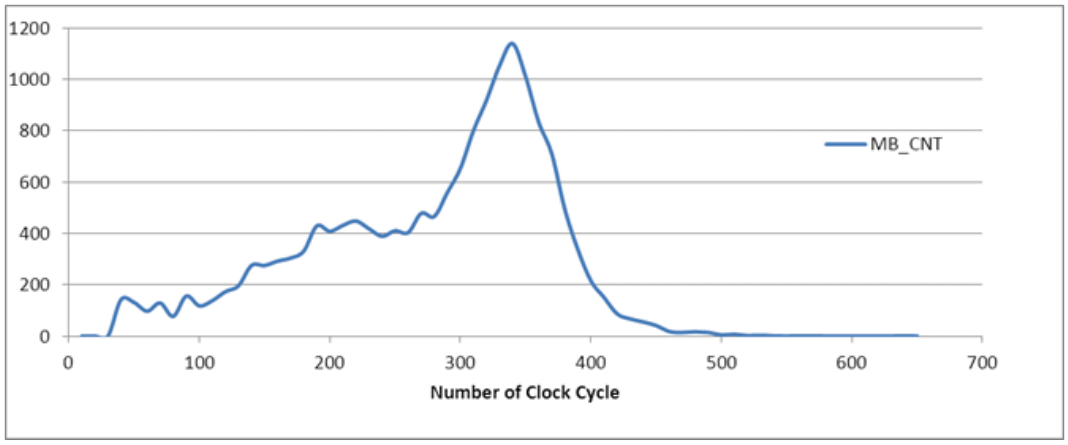
1.3.6.1.3.9.1 Кодирование H.264

В таблице 1.160 перечислены процессы, выполняемые в ходе кодирования H.264, а также критерии для кодирования H.264 1080p 60 к/с на частоте 266 МГц, происходящие в реальном масштабе времени. Все описанные ниже процессы происходят одновременно. Блоком обработки данных является один макроблок, за исключением случаев использования моделей доступа к внешней памяти. Каждый из процессов, с точки зрения CCLK, должен завершаться, как правило, в течение 500 циклов. Допустим, что соотношение CCLK:ACLK равно 1:f.

Таблица 1.160 – Результаты кодирования H.264 HP 1080p по эффективности

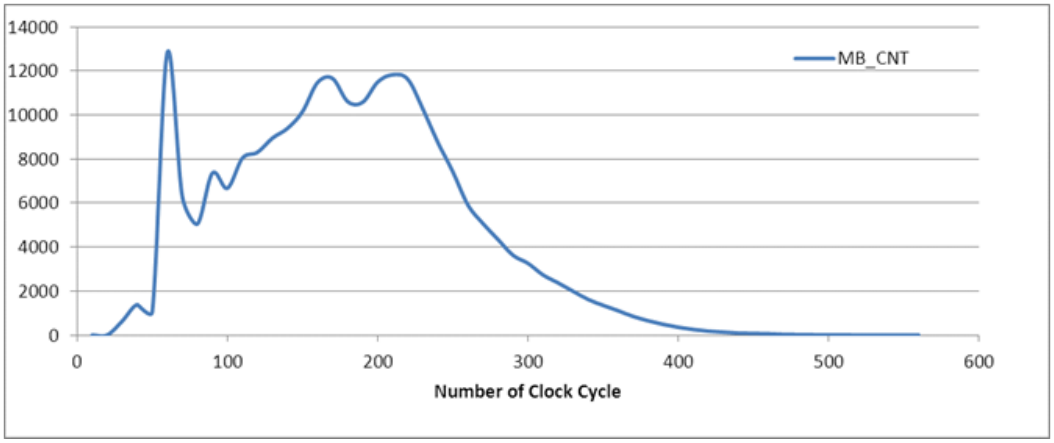
Процесс	Циклы
Считывание макроблоков, подлежащих кодированию, из внешней памяти	Передача 384 байтов, в среднем, должна завершаться за 500*f циклов. 384 байта занимают 48 циклов в 64-разрядном шинном интерфейсе.
Считывание ссылочных пикселей компоненты яркости для оценки параметров движения	Передача 384 байтов, в среднем, должна завершаться за 500*f циклов. 384 байта занимают 48 циклов в 64-разрядном шинном интерфейсе.
Integer-pel оценка параметров движения	Цикл фиксированной обработки < 500 циклов
Fractional-pel оценка параметров движения	Цикл фиксированной обработки < 500 циклов
Распознавание в режиме Intra-прогнозирования	Цикл фиксированной обработки < 500 циклов
Считывание ссылочных пикселей компоненты цветности для компенсации движения	Передача 162 байтов, в среднем, должна завершаться за 500*f циклов. 162 байта занимают 21 цикл в 64- разрядном шинном интерфейсе.
Компенсация движения для пикселей компоненты цветности	Цикл обработки @ наихудший сценарий ≅ 500 циклов
Intra-прогнозирование	Цикл обработки @ наихудший сценарий ≅ 500 циклов
Преобразование/квантизация	Цикл обработки @ наихудший сценарий ≅ 500 циклов
Фильтр деблокирования	Цикл фиксированной обработки < 500 циклов
Запись отфильтрованных пикселей обратно во внешнюю память	Передача 384 байтов, в среднем, должна завершаться за 500*f циклов обращения к памяти. 384 байта занимают 48 циклов в 64-разрядном шинном интерфейсе.
Энтропийное кодирование	Количество циклов обработки в случае CAVLC зависит от битовой скорости, а именно, от количества элементов синтаксиса, подлежащих кодированию. Но наиболее важным является значение количества коэффициентов. См. рисунок M.1 - “Распределение циклов обработки для макроблоков в I-слайсе @ 50 Мбит/сек. CAVLC” и рисунок M.2 - “Распределение циклов обработки макроблоков в P-слайсе @ 50 Мбит/с CAVLC”. В случае кодирования CABAC, количество циклов, главным образом, зависит от количества накопителей.

По оси абсцисс отложено количество циклов обработки. По оси ординат отложено количество макроблоков, цикл обработки которых соответствует точке на оси абсцисс.



Number of Clock Cycle – Количество циклов синхронизации

Рисунок 1.123 - Распределение циклов обработки для макроблоков в I-слайсе @ 50 Мбит/с CAVLC



Number of Clock Cycle – Количество циклов синхронизации

Рисунок 1.124 - Распределение циклов обработки для макроблоков в P-слайсе @ 50Mbps CAVLC

На рисунке 1.123 - “Распределение циклов обработки для макроблоков в I-слайсе @ 50 Мбит/с CAVLC” и рисунок 1.124 - “Распределение циклов обработки для макроблоков в P-слайсе @ 50 Мбит/с CAVLC” показано что энтропийное кодирование с 50 Мбит/с завершается, как правило, менее чем за 500 циклов.

1.3.6.1.3.9.2 Декодирование H.264

В приведенной ниже таблице приведены данные по эффективности декодирования H.264 HP 1080p при условии соблюдения следующих условий:

- дополнительная память, доступ к которой обеспечивается по вспомогательной шине AXI, составляет 53 760 байтов;
- 64-разрядная SDRAM (32-разрядная SDRAM x 2) - для внешней памяти;
- ACLK:CCLK = 1:1 - соотношение частоты синхронизации;
- Используется контроллер SDRAM C&M.

Таблица 1.161 - Результаты декодирования H.264 HP 1080p по эффективности

Стандарт	Усредн. частота (в МГц)	Описание потока
H.264 HP	106	Ripped Blu-ray 1080p24fps
H.264 HP	154	Allegro_BdWidth_CABAC_01_L41 1080p30fps
H.264 HP	250	Allegro_BdWidth_CABAC_01_L42 1080p60fps

1.3.6.2 Кодер видеопотока (программная модель)

Chips&Media JPU (JPEG Processing Unit – блок обработки изображений в формате JPEG) – это высокопроизводительный кодек JPEG для IP-камер, способный осуществлять кодирование и декодирование изображений в базовом (Baseline) и движущемся (Motion) форматах JPEG. Блок JPU поддерживает скорость кодирования цветного изображения формата 4:2:2 до 155 мегапикселей/сек и скорость декодирования цветного изображения формата 4:4:4 до 100 мегапикселей/сек. Блок-схема модуля JPU представлена на рисунке 1.162.

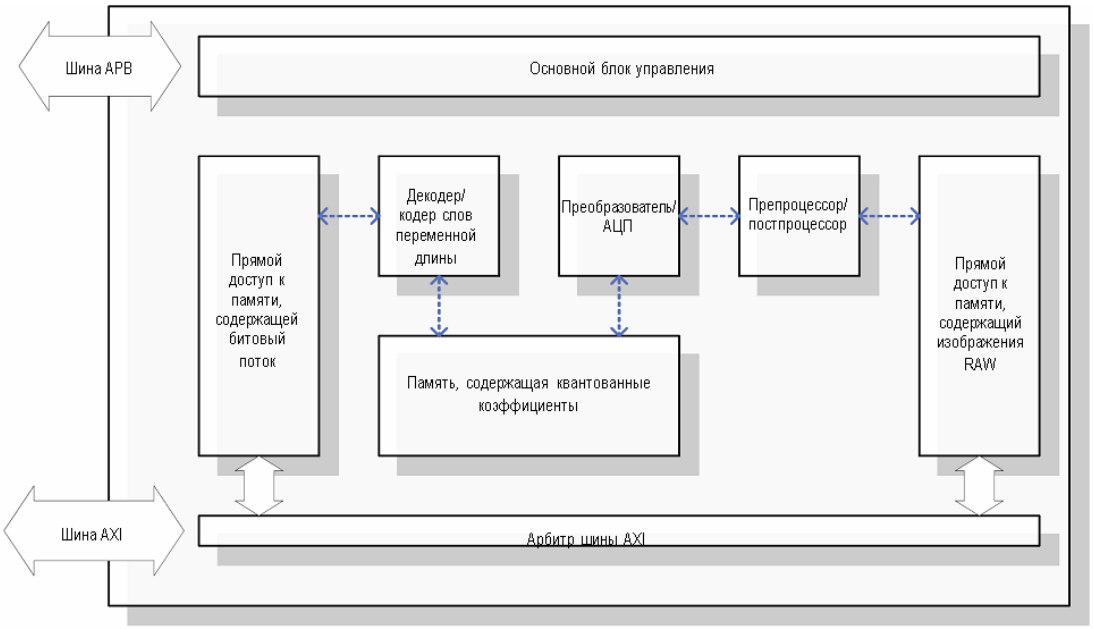


Рисунок 1.162 - Блок-схема работы модуля JPU

Блок JPU соединяется с системой по 32-битной шине AMBA 2 APB для управления системой и по 64-битной шине AMBA 3 AXI для обеспечения пропускной способности канала передачи данных, а наличие внутрикристальной памяти обеспечивает высокую производительность.

Блок JPU встраивается в определяемую заказчиком систему на кристалле (SoC) и управляется главным процессором как высокопроизводительной подсистемой обработки изображений. Для обеспечения простого и эффективного управления блоком JPU главным процессором блок JPU оснащен серией регистров. Как правило, большая часть команд и откликов, которыми обмениваются главный процессор и блок JPU, передается через данные регистры. Главный процессор и блок JPU могут осуществлять прямой доступ к потоковым данным и данным изображений.

1.3.6.2.1 Основные характеристики

Блок JPU поддерживает все разрешения вплоть до 32768x32768 пикселей для операций кодирования и декодирования в соответствии со стандартом базового JPEG. Блок JPU легко интегрируется в систему за счет возможности адаптации шинной архитектуры AXI/APB, что обеспечивает простое функционирование блока в системе на кристалле.

Основные функции блока JPU приведены ниже.

Кодер и декодер MJPEG в реализации Baseline

- Соответствие стандарту JPEG ISO/IEC 10918-1
- Поддержка от 1 до 3 цветовых компонентов

- 3 компонента в скане (только чересстрочный скан)
- 8-битные выборки для каждого компонента
- Поддержка форматов цветного изображения 4:2:0, 4:2:2, 4:4:0, 4:4:4 и 4:0:0 (максимум 8x8 блоков в одном минимальном кодированном блоке (MCU))
- Минимальный размер кодируемых данных – 16x16 пикселей
- Поддержка зоны ROI (зона интереса – только декодер)

Примечание - Для формата цветного изображения 4:4:4 поддерживается упакованный десятичный код. Более подробная информация приведена в Приложении Е «УПАКОВАННЫЙ ДЕСЯТИЧНЫЙ ФОРМАТ».

1.3.6.2.2 ГЛАВНЫЙ ИНТЕРФЕЙС

1.3.6.2.2.1 Схема управления

Блок JPU обеспечивает серию команд для управления операциями с изображениями JPEG с использованием режима Picture by Picture (картинка рядом с картинкой), а также соответствующие отклики. Большая часть команд и откликов, которыми обмениваются главный процессор и блок JPU, передается через данные регистры.

1.3.6.2.2.2 Регистры

Блок JPU поддерживает связь с главным процессором через интерфейс шины APB. Далее приводится перечень и подробное описание регистров блока JPU.

1.3.6.2.2.2.1 Общие сведения о регистрах

Таблица 1.163 содержит перечень регистров блока JPU, осуществляющих связь с главным процессором.

Таблица 1.163 – Регистры интерфейса главного процессора

Адрес	Тип	Имя	Описание
Регистры главного устройства управления			
BASE+0x000	Запись/чтение	mjpeg_pic_start_reg	Регистр начала/завершения кодирования/декодирования изображения
BASE+0x004	Запись/чтение	mjpeg_pic_status_reg	Регистры состояния (прерывания) кодека
BASE+0x008	Чтение	mjpeg_pic_errmb_reg	Информация об ошибочном MCU
BASE+0x00C	Запись	mjpeg_pic_setmb_reg	Регистр задания положения MCU
BASE+0x010	Запись/чтение	mjpeg_pic_ctrl_reg	Регистр управления изображением
BASE+0x014	Запись/чтение	mjpeg_pic_size_reg	Регистр формата изображения
BASE+0x018	Запись/чтение	mjpeg_mcu_info_reg	Информация о MCU
BASE+0x01C	Запись/чтение	mjpeg_rot_info_reg	Регистр режима поворота/зеркального отображения
Base+0x020	Запись/чтение	mjpeg_scl_info_reg	Регистр режима субдискретизации
Base+0x024	Запись/чтение	mjpeg_if_info_reg	Регистр внешнего интерфейса
Base+0x02C	Запись/чтение	mjpeg_op_info_reg	Информационный регистр частичного режима
Base+0x030	Запись/чтение	mjpeg_dpb_config_reg	Конфигурация буфера декодированных изображений (DPB)
Base+0x034	Запись/чтение	mjpeg_dpb_base00_reg	База DPB 0 для компонента ID 0
Base+0x038	Запись/чтение	mjpeg_dpb_base01_reg	База DPB 0 для компонента ID 1
Base+0x03C	Запись/чтение	mjpeg_dbp_base02_reg	База DPB 0 для компонента ID 2
Base+0x040	Запись/чтение	mjpeg_dpb_base10_reg	База DPB 1 для компонента ID 0
Base+0x044	Запись/чтение	mjpeg_dpb_base11_reg	База DPB 1 для компонента ID 1
Base+0x048	Запись/чтение	mjpeg_dbp_base12_reg	База DPB 1 для компонента ID 2
Base+0x04C	Запись/чтение	mjpeg_dpb_base20_reg	База DPB 2 для компонента ID 0
Base+0x050	Запись/чтение	mjpeg_dpb_base21_reg	База DPB 2 для компонента ID 1
Base+0x054	Запись/чтение	mjpeg_dbp_base22_reg	База DPB 2 для компонента ID 2
Base+0x058	Запись/чтение	mjpeg_dpb_base30_reg	База DPB 3 для компонента ID 0
Base+0x05C	Запись/чтение	mjpeg_dpb_base31_reg	База DPB 3 для компонента ID 1
Base+0x060	Запись/чтение	mjpeg_dbp_base32_reg	База DPB 3 для компонента ID 2
Base+0x064	Запись/чтение	mjpeg_dpb_ystride_reg	Шаг по индексу компонента яркости DPB
Base+0x068	Запись/чтение	mjpeg_dpb_cstride_reg	Шаг по индексу компонента цветности DPB

Продолжение таблицы 1.163

Адрес	Тип	Имя	Описание
Base+0x06C	Чтение	mjpeg_wresp_check_reg	Регистр проверки канала отклика операций записи шины AXI
Base+0x080	Запись/чтение	mjpeg_huff_ctrl_reg	Регистр управления таблицей Хаффмана
Base+0x084	Запись/чтение	mjpeg_huff_addr_reg	Регистр адреса таблицы Хаффмана
Base+0x088	Запись/чтение	mjpeg_huff_data_reg	Регистр данных таблицы Хаффмана
Base+0x090	Запись/чтение	mjpeg_qmat_ctrl_reg	Регистр управления матрицей квантования
Base+0x094	Запись/чтение	mjpeg_qmat_addr_reg	Регистр адреса матрицы квантования
Base+0x098	Запись/чтение	mjpeg_qmat_data_reg	Регистр данных матрицы квантования
Base+0x0A0	Запись/чтение	mjpeg_coef_ctrl_reg	Регистр управления памятью, содержащей коэффициенты
Base+0x0A4	Запись/чтение	mjpeg_coef_addr_reg	Регистр адреса памяти, содержащей коэффициенты
Base+0x0A8	Запись/чтение	mjpeg_coef_data_reg	Регистр данных памяти, содержащей коэффициенты
Base+0x0B0	Запись/чтение	mjpeg_rst_intval_reg	Регистр интервала перезапуска
Base+0x0B4	Запись/чтение	mjpeg_rst_index_reg	Регистр маркера перезапуска
Base+0x0B8	Запись/чтение	mjpeg_rst_count_reg	Регистр счетчика перезапусков
Base+0x0C0	Запись/чтение	mjpeg_intr_mask_reg	Маски прерываний
Base+0x0C8	Чтение	mjpeg_cycle_info_reg	Данные цикла для кодирования/декодирования JPEG
Base+0x0F0	Запись/чтение	mjpeg_dpcm_diff_y_reg	Коэффициент DC для компонента 0
Base+0x0F4	Запись/чтение	mjpeg_dpcm_diff_cb_reg	Коэффициент DC для компонента 1
Base+0x0F8	Запись/чтение	mjpeg_dpcm_diff_cr_reg	Коэффициент DC для компонента 2
Регистры управления блоком GBU			
Base+0x100	Запись/чтение	mjpeg_gbu_ctrl_reg	Регистр управления GBU
Base+0x104	Чтение	mjpeg_gbu_pbit_busy_reg	Регистр состояния занятости при записи битов
Base+0x110	Запись/чтение	mjpeg_gbu_bt_ptr_reg	Регистр-указатель текущего бита буфера GRAM
Base+0x114	Запись/чтение	mjpeg_gbu_wd_ptr_reg	Регистр-указатель текущего слова буфера GRAM
Base+0x118	Запись/чтение	mjpeg_gbu_tt_cnt_reg	Регистр общего размера накопленных получаемых битов
Base+0x11C	Запись/чтение	mjpeg_gbu_tt_cnt_reg	Регистр общего размера накопленных получаемых битов
Base+0x120	Запись	mjpeg_gbu_pbit_08_reg	Регистр записи 8 битов
Base+0x124	Запись	mjpeg_gbu_pbit_16_reg	Регистр записи 16 битов
Base+0x128	Запись	mjpeg_gbu_pbit_24_reg	Регистр записи 24 битов
Base+0x12C	Запись	mjpeg_gbu_pbit_32_reg	Регистр записи 32 битов
Base+0x140	Запись/чтение	mjpeg_gbu_bbsr_reg	Регистр-указатель начала потока данных буфера GRAM
Base+0x144	Запись/чтение	mjpeg_gbu_bber_reg	Регистр-указатель конца потока данных буфера GRAM
Base+0x148	Запись/чтение	mjpeg_gbu_bbir_reg	Регистр-указатель прерываний буфера GRAM

Продолжение таблицы 1.163

Адрес	Тип	Имя	Описание
Base+0x14C	Запись/чтение	mjpeg_gbu_bbhr_reg	Регистр остановки потока данных во внутренний буфер битовых потоков
Base+0x158	Запись/чтение	mjpeg_gbu_bcnt_reg	Регистр размера накопленных получаемых битов
Base+0x160	Запись/чтение	mjpeg_gbu_ff_rptr_reg	Регистр-указатель чтения значений эмулятора FF
Base+0x164	Запись/чтение	mjpeg_gbu_ff_wptr_reg	Регистр-указатель записи значений эмулятора FF
Регистры управления блоком BBC			
Base+0x208	Запись/чтение	mjpeg_bbc_endaddr_reg	Регистр адреса команды прерывания потока данных внешнего буфера битовых потоков
Адрес	Тип	Имя	Описание
Base+0x20C	Запись/чтение	mjpeg_bbc_wr_ptr_reg	Регистр-указатель записи потока данных внешнего буфера битовых потоков
Base+0x210	Запись/чтение	mjpeg_bbc_rd_ptr_reg	Регистр-указатель чтения потока данных внешнего буфера битовых потоков
Base+0x214	Запись/чтение	mjpeg_bbc_ext_addr_reg	Регистр адреса начала потока данных внешнего буфера битовых потоков
Base+0x218	Запись/чтение	mjpeg_bbc_int_addr_reg	Регистр адреса начала потока данных внутреннего буфера битовых потоков
Base+0x21C	Запись/чтение	mjpeg_bbc_data_cnt_reg	Регистр размера порции передачи данных
Base+0x220	Запись/чтение	mjpeg_bbc_command_reg	Регистр команд обращения к памяти
Base+0x224	Чтение	mjpeg_bbc_busy_reg	Регистр-индикатор занятости блока BBC
Base+0x228	Запись/чтение	mjpeg_bbc_ctrl_reg	Регистр управления блоком BBC
Base+0x22C	Запись/чтение	mjpeg_bbc_cur_pos_reg	Текущее положение блока BBC
Base+0x230	Запись/чтение	mjpeg_bbc_bas_addr_reg	Базовый адрес битовых потоков
Base+0x234	Запись/чтение	mjpeg_bbc_strm_ctrl_reg	Данные управления битовым потоком (только декодер)
Base+0x238	Запись/чтение	mjpeg_bbc_flush_cmd_reg	Регистр команды завершения очистки для внутреннего буфера битовых потоков (только кодер)

Данные регистра могут быть подразделены на следующие три категории:

Регистры главного устройства управления

Регистры данной категории используются для инициализации информационных изображений и данных буферов изображения.

Регистры управления блоком GBU

Регистры данной категории используются для задания параметров блока GBU (Get Bit Unit – Блок получения битов).

Регистры управления блоком BBC

Регистры данной категории используются для задания параметров блока BBC (Bit-stream Buffer Control – Блок управления буфером битовых потоков).

1.3.6.2.2.2.2 Подробное описание регистров

В данном подразделе приведено подробное описание полей регистров.

1.3.6.2.2.2.2.1 Регистры главного устройства управления

Таблица 1.164 – Регистр MJPEG_PIC_START_REG (0x000)

Бит	Наименование	Тип	Описание	Значение параметра сброса
3:0	mjpeg_pic_start_reg	Запись/чтение	Регистр начала/завершения кодирования/декодирования изображения [3] – начало процесса частичного кодирования [2] – завершение процесса частичного кодирования/декодирования [1] – инициализация состояния кодера/декодера [0] – начало процесса кодирования/декодирования Если бит [0] установлен в «1», блок JPU запускает процесс кодирования/декодирования. Если бит [1] установлен в «1», блок JPU инициализирует регистры внутреннего состояния и управления. Если бит [2] установлен в «1» в ходе выполнения процесса кодирования/декодирования, блок JPU начинает работать в режиме конечного автомата (режим FSM - Finite State Machine), завершает операцию шины и создает прерывание. Бит [3] используется, когда кодер блока JPU работает в частичном режиме. Главный компьютер запускает кодер блока JPU значением бита [0], при этом бит [3] должен быть установлен в «1» при готовности буфера частичного режима. Главному компьютеру необходимо только задать регистр в случае готовности буфера независимо от завершения операции частичного кодирования кодером блока JPU, так как число для подготовленного буфера частичного режима уже подсчитано кодером JPEG. Если значение регистра установлено в «1», оно автоматически сбрасывается после формирования импульса в кодере JPEG, и остальные биты [0], [1] и [2] переходят в исходное состояние. Более подробная информация о регистрах, связанных с частичным режимом, приведена в описании регистров mjpeg_pic_ctrl_reg и mjpeg_op_info_reg.	0

Таблица 1.165 – Регистр MJPEG_PIC_STATUS_REG (0x004)

Бит	Наименование	Тип	Описание	Значение параметра сброса
8:0	mjpeg_pic_status_reg	Запись/чтение	Регистры состояния (прерывания) кодека [8] – остановка обработки JPEG изображений [7] – состояние буфера 3 частичного режима кодера/декодера JPEG [6] – состояние буфера 2 частичного режима кодера/декодера JPEG [5] – состояние буфера 1 частичного режима кодера/декодера JPEG [4] – состояние буфера 0 частичного режима кодера/декодера JPEG [3] – переполнение буфера при обработке JPEG изображений: Данное поле доступно только в частичном режиме. [2] – прерывание ВВС [1] – ошибка обработки JPEG изображений [0] – обработка JPEG изображений завершена В случае приема главным компьютером запроса на прерывание данный регистр должен произвести проверку на предмет типа прерывания. Более подробная информация приведена ниже. • JPEG STOP – Остановка обработки JPEG изображений Если бит [2] регистра MJPEG_PIC_START_REG установлен в «1», блок JPU установит соответствующий регистр в «1» и сформирует прерывание после перехода всех функций конечного автомата в состояние незанятости. Состояние буферов 3/2/1/0 частичного режима кодера/декодера JPEG: Данное поле отображает состояние буфера частичного режима кодера или декодера JPEG. В случае кодера JPEG если главный компьютер устанавливает флаг готовности буфера частичного режима кодера JPEG в «1» по завершении загрузки изображения первого буфера частичного режима (буфер частичного режима 0), блок JPU устанавливает флаг состояния буфера частичного режима кодера/декодера JPEG в «1» и формирует прерывание после считывания исходного изображения в буфере частичного режима 0. • JPEG_DONE – обработка JPEG изображений завершена Если в данном поле задано значение «1», будет произведено кодирование/декодирование одного изображения. • JPEG_ERROR – ошибка обработки JPEG изображений Если в данном поле задано значение «1», блок JPU детектирует ошибку в ходе декодирования битового	0

Продолжение таблицы 1.165

Бит	Наименование	Тип	Описание	Значение параметра сброса
			потока. (В случае кодирования значение данного поля не может быть установлено в «1», так как в данном случае оно будет доступно только для декодирования) • BBC_INTERRUPT – прерывание BBC Если в данном поле задано значение «1», главный компьютер должен считать или записать закодированный/декодированный битовый поток и обновить указатели чтения/записи. Подробная информация приведена в описании регистра прямого доступа к памяти буфера битовых потоков. • JPEG_OVERFLOW NOTE – переполнение буфера при обработке JPEG изображений. Данное поле доступно только в частичном режиме. В случае переполнения или неполного заполнения буфера при обмене данными между блоком JPU и внешним блоком, например, сенсорным интерфейсом или интерфейсом скейлера, значение данного поля устанавливается в «1». Даже в случае детектирования переполнения или неполного заполнения блок JPU не активизирует прерывание. Он только устанавливает значение данного поля в «1» и продолжает осуществлять кодирование или декодирование. Главный компьютер может признать переполнение или неполное заполнение после завершения кодирования/декодирования одного изображения. После чтения данного регистра главный компьютер должен произвести его сброс, так как в противном случае деактивация прерываний невозможна. Для сброса данного регистра главный компьютер должен произвести в него обратную запись. Например, если главный компьютер считывает «0x0000_0004» из данного регистра, он должен записать в регистр «0x0000_0004» для обеспечения возможности сброса прерывания.	

Таблица 1.66 – Регистр MJPEG_PIC_ERRMB_REG (0x008)

Бит	Наименование	Тип	Описание	Значение параметра сброса
27:0	mjpeg_pic_errmb_reg	Чтение	Информация об ошибочном MCU [27:24] – индекс перезапуска по ошибке JPEG [23:12] – положение по оси X ошибочного MCU изображения JPEG [11:0] – положение по оси Y ошибочного MCU изображения JPEG Регистр MJPEG_PIC_ERRMB_REG отображает положение минимального кодированного блока, содержащего ошибки. Когда главный компьютер читает регистр MJPEG_PIC_STATUS_REG (Base + 0x004) и обнаруживает, что бит [1] (JPEG_ERROR) установлен в «1», главный компьютер может определить, какой минимальный кодированный блок содержит ошибку из JPEG_ERROR_MCU_POSITION_X и JPEG_ERROR_MCU_POSITION_Y. Главный компьютер также может найти индекс последнего перезапуска до возникновения ошибки в минимальном кодированном блоке в поле JPEG_ERROR_RESTART_INDEX. Данное поле обновляется каждый раз, когда блок JPU детектирует индекс нового перезапуска.	0

Таблица 1.167 – Регистр MJPEG_PIC_SETMB_REG (0x00C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
27:0	mjpeg_pic_setmb_reg	Запись	Регистр задания положения MCU [27:16] – положение MCU по оси X [11:0] – положение MCU по оси Y При возникновении ошибки главный компьютер может отыскать код следующего перезапуска или код запуска изображения. Затем главный компьютер определяет положение минимального кодированного блока, который становится первым минимальным кодированным блоком в коде следующего перезапуска. Главный компьютер должен занести положение минимального кодированного блока в данный регистр, чтобы блок JPU мог декодировать битовый поток из данного положения. Примечание Для использования данного регистра главный компьютер должен найти код следующего перезапуска и обеспечить надлежащее управление битовым потоком.	0

Таблица 1.168 – Регистр MJPEG_PIC_CTRL_REG (0x010)

Бит	Наименование	Тип	Описание	Значение параметра сброса
12:0	mjpeg_pic_ctrl_reg	Запись/ чтение	Регистр управления изображением [12:10] – индекс таблицы Хаффмана для коэффициентов AC [9:7] - индекс таблицы Хаффмана для коэффициентов DC [6] – таблица Хаффмана разрешена [4] – направление TC, 1 - кодер, 0 - декодер [3] – режим кодера, по умолчанию режим декодера [2] – отклик на запись изображения RAW при прямом доступе к памяти разрешен [1:0] – рабочий режим Данный регистр используется как для кодера, так и для декодера. Описание полей приведено ниже. Бит [1:0]: Рабочий режим. Когда значение данного поля установлено в «0», блок JPU работает в нормальном режиме. Он лишь осуществляет декодирование или кодирование одного изображения. Когда значение данного поля установлено в «1», блок JPU работает частичном режиме. Остальные значения зарезервированы. Бит [2]: Проверка канала отклика операций записи разрешена. Когда значение данного поля установлено в «1», блок JPU проверяет сигнал BVALID в протоколе AXI для обеспечения непротиворечивости данных с каждым последним минимальным кодированным блоком изображения в нормальном режиме или с каждым последним минимальным кодированным блоком буфера деления изображения в частичном режиме. Когда значение данного поля установлено в «0», блок JPU не производит проверку сигнала BVALID в протоколе AXI. Бит [3]: Флаг кодера или декодера. Для запуска декодера главный компьютер должен установить значение данного поля в «0». Для запуска кодера главный компьютер должен установить значение данного поля в «1». Бит [4]: Направление коэффициента преобразования. В режиме кодера значение данного поля должно быть установлено в «1». В режиме декодера значение данного поля должно быть установлено в «0». Бит [5]: Резерв Бит [6]: Пользовательская таблица Хаффмана разрешена. Если будет использоваться пользовательская таблица Хаффмана, значение данного поля должно быть установлено в «1», в противном случае – в «0». Бит [9:7]: Индекс таблицы Хаффмана для коэффициентов DC Бит [12:10]: Индекс таблицы Хаффмана для коэффициентов AC	0

Таблица 1.169 – Регистр MJPEG_PIC_SIZE_REG (0x014)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_pic_size_reg	Запись/чтение	Регистр формата изображения [31:16] – размер изображения по горизонтали [15:0] – размер изображения по вертикали Это регистр формата изображения. Главный компьютер учитывает нижеприведенные ограничения по формату изображения. Во-первых, размер изображения должен быть кратным размеру минимального кодированного блока, т.е. при формате 4:2:2 размер по горизонтали должен быть кратным 16, а размер по вертикали – кратным 8. Во-вторых, минимальный размер кодируемого изображения составляет 16 x 16, а минимальный размер декодируемого изображения – один размер минимального кодированного блока. В-третьих, максимальный размер кодируемого и декодируемого изображения должен составлять 8192 x 8192. В случае кодирования размер изображения должен быть равен размеру изображения на входе независимо от настроек поворота/зеркального отображения. Например, размер изображения, передаваемого от CCD-датчика, составляет 720x480, и регистр должен быть установлен как 720x480. Даже если разрешен поворот на 90 градусов, и размер закодированного изображения составляет 480x740, регистр формата изображения должен быть установлен как 720x480. В случае декодирования размер изображения в данном регистре должен быть единым в заголовке битового потока JPEG независимо от настроек поворота/зеркального отображения.	0

Таблица 1.170 – Регистр MJPEG_MCU_INFO_REG (0x018)

Бит	Наименование	Тип	Описание	Значение параметра сброса
19:0	mjpeg_mcu_info_reg	Запись/чтение	Информация о MCU [19:16]: общее количество блоков [14:12] : количество компонентов [11:8] : информация о компоненте 0: - [11:10] коэффициент дискретизации по горизонтали - [9:8] коэффициент дискретизации по вертикали [7:4] : информация о компоненте 1: - [7:6] коэффициент дискретизации по вертикали - [5:4] коэффициент дискретизации по вертикали [3:0] : информация о компоненте 2: - [3:2] коэффициент дискретизации по вертикали - [1:0] коэффициент дискретизации по вертикали Данный регистр содержит информацию о минимальном кодированном блоке, определяющую формат YUV. Блок JPU может поддерживать 3 компонента в минимальном кодированном блоке, и каждый компонент может иметь формат 1x1, 1x2, 2x1 и 2x2.	

Продолжение таблицы 1.170

Бит	Наименование	Тип	Описание	Значение параметра сброса
19:0	mjpeg_mcu_info_reg	Запись/чтение	Бит [11:0] полностью посвящен размеру 3 компонентов. Бит [14:12] представляет количество компонентов в минимальном кодированном блоке. В случае 4:0:0 значение данного поля должно быть установлено в «1», так как присутствует только один компонент. В случае 4:2:0, 4:2:2, 2:2:4 и 4:4:4 значение поля [14:12] должно быть «3». Бит [19:16] представляет общее количество блоков в минимальном кодированном блоке. Блок может определяться как «8 пикселей на 8 пикселей» (8x8). Поэтому в случае если формат 4:2:0 содержит в целом 6 блоков в минимальном кодированном блоке (яркость составляет 16x16, что означает четыре блока 8x8, и цветность – один блок 8x8 для составляющей Cb и один блок 8x8 – для составляющей Cr). Главный компьютер должен задать значение данного регистра для каждого из нижеприведенных форматов YUV: • формат 4:2:0 - 0x0006_3A55 BIT[19:16] : 0x6 BIT[14:12] : 0x3 Comp0[11:8] : 0xA Comp1[7:4] : 0x5 Comp2[3:0] : 0x5 • формат 4:2:2 - 0x0004_3955 BIT[19:16] : 0x4 BIT[14:12] : 0x3 Comp0[11:8] : 0x9 Comp1[7:4] : 0x5 Comp2[3:0] : 0x5 • формат 2:2:4 - 0x0004_3655 BIT[19:16] : 0x4 BIT[14:12] : 0x3 Comp0[11:8] : 0x6 Comp1[7:4] : 0x5 Comp2[3:0] : 0x5 • двумерный формат 4:4:4 - 0x0003_3555 BIT[19:16] : 0x3 BIT[14:12] : 0x3 Comp0[11:8] : 0x5 Comp1[7:4] : 0x5 Comp2[3:0] : 0x5 • упакованный десятичный формат 4:4:4 - 0x0013_3555	0

Продолжение таблицы 1.170

Бит	Наименование	Тип	Описание	Значение параметра сброса
19:0	mjpeg_mcu_info_reg	Запись/чтение	BIT[19:16] : 0x3 BIT[14:12] : 0x3 Comp0[11:8] : 0x5 Comp1[7:4] : 0x5 Comp2[3:0] : 0x5 • формат 4:0:0 - 0x0001_1500 BIT[19:16] : 0x1 BIT[14:12] : 0x1 Comp0[11:8] : 0x5 Comp1[7:4] : 0x0 Comp2[3:0] : 0x0	

Таблица 1.171 – Регистр MJPEG_ROT_INFO_REG (0x01C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
4:0	mjpeg_rot_info_reg	Запись/чтение	Регистр режима поворота/зеркального отображения [4] – поворот-зеркальное отображение разрешены [3] – режим поворота-зеркального отображения Примечание Подробное описание приведено в техническом описании главы «Блок препроцессора/постпроцессора». Когда главный компьютер собирается использовать режим поворота/зеркального отображения, значение бита [4] регистра MJPEG_ROT_INFO_REG должно быть установлено в «1». BIT[1:0] представляет поворот влево на 90 x n градусов (n – 0, 1, 2 или 3) BIT[2] – зеркальное отображение по вертикали BIT[3] – зеркальное отображение по горизонтали	0

Таблица 1.172 – Регистр MJPEG_SCL_INFO_REG (0x020)

Бит	Наименование	Тип	Описание	Значение параметра сброса
4:0	mjpeg_scl_info_reg	Запись/чтение	Регистр режима субдискретизации [4] – масштаб разрешен [3:2] коэффициент субдискретизации по горизонтали [1:0] коэффициент субдискретизации по вертикали Если значение бита [3] установлено в «1», декодер блока JPU уменьшает масштаб декодированного изображения. Масштабирование доступно только для режима декодера и является взаимоисключающим с режимом поворота/зеркального отображения, что означает, что обе функции не могут работать одновременно. Бит [1:0] – это коэффициент субдискретизации по вертикали. - 0: Масштабирование отсутствует - 1: Уменьшение масштаба в 1/2 раза с использованием билинейного фильтра - 2: Уменьшение масштаба в 1/4 раза с использованием 4-tap фильтра - 3: Уменьшение масштаба в 1/8 раза с использованием 4-tap фильтра BIT [3:2] – это коэффициент субдискретизации по горизонтали. - Каждое значение равно значению коэффициента субдискретизации по вертикали. BIT[4]: Значение «1» разрешает субдискретизацию. Значение «0» запрещает субдискретизацию.	0

Таблица 1.173 – Регистр MJPEG_IF_INFO_REG (0x024)

Бит	Наименование	Тип	Описание	Значение параметра сброса
1:0	mjpeg_if_info_reg	Запись/чтение	Регистр внешнего интерфейса [1] – сброс сенсорного интерфейса [0] – сброс интерфейса отображения Данный регистр нужен только при активированном частичном режиме и детектировании переполнения/неполного заполнения. Подсчет запускается, когда блоку JPU необходимо детектировать переполнение или неполное заполнение буфера частичного режима. При возникновении события переполнения или неполного заполнения происходит повреждение закодированного битового потока или декодированного изображения (состояние ошибки). В случае возникновения данной ошибки главный компьютер устанавливает значение данного регистра как «1» (для каждого бита) для сброса счетчика буфера кадров. Примечание После этого главный компьютер возвращает значение данного регистра в «0».	0

Таблица 1.174 – Регистр MJPEG_CLP_INFO_REG (0x028)

Бит	Наименование	Тип	Описание	Значение параметра сброса
1:0	mjpeg_clp_info_reg	Запись/чтение	Данный регистр активирует режим вырезания (зона ROI) для декодера. [1] – критерий сигнала прерывания по завершению декодирования [0] – режим вырезания (зона ROI) для декодера разрешен Бит [1] указывает, будет ли продолжено декодирование после задания зоны ROI и будет ли значение поля по умолчанию «0». Если бит [1] = 0, блок JPU завершает декодирование до конца изображения. Но если бит [1] = 1, блок JPU декодирует последний минимальный кодированный блок в зоне ROI и возникает сигнал прерывания по завершению кодирования для завершения процесса декодирования.	0

Таблица 1.175 – Регистр MJPEG_OP_INFO_REG (0x02C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
2:0	mjpeg_op_info_reg	Запись/чтение	Данный регистр содержит служебную информацию блока JPU. Бит [2:0] представляет количество запросов минимального кодированного блока. Блок JPU поддерживает множество запросов в блоке прямого доступа к памяти, то эффективность использования полосы частот значительно возрастает. Для обеспечения максимальной эффективности использования полосы частот значение поля [2:0] должно быть задано следующим образом: 4:2:0 - 0x2 4:2:2 или 2:2:4 - 0x3 4:4:4 или 4:0:0 - 0x4 Значение поля [2:0] должно находиться в диапазоне от 1 до максимального вышеуказанного значения. Остальные значения запрещены.	1
5:3			Бит [5:3] представляет количество буферов частичного режима при активированном частичном режиме. В частичном режиме блок JPU может кодировать или декодировать часть за частью с использованием буферов кадров, заданных в данном поле, для низкой эффективности использования полосы частот. Количество буферов частичного режима должно быть не менее двух и не более четырех. Главный компьютер должен определить количество используемых буферов частичного режима и задать определенное значение в поле [5:3]. Количество буферов частичного режима должно быть от 1 до 4.	0
6:15	-		Резерв	-
31:16			Бит [31:16] указывает количество линий в одном буфере частичного режима. Он должен иметь фиксированное значение, 16 для формата 420 или 440(224) и 8 для других форматов, 422, 444 и 400. Например, 4 буфера частичного режима с высотой 8 должны быть заданы в регистре как «4» в поле [5:3] и «8» в поле [31:16], как показано на рисунке 1.125.	'd16

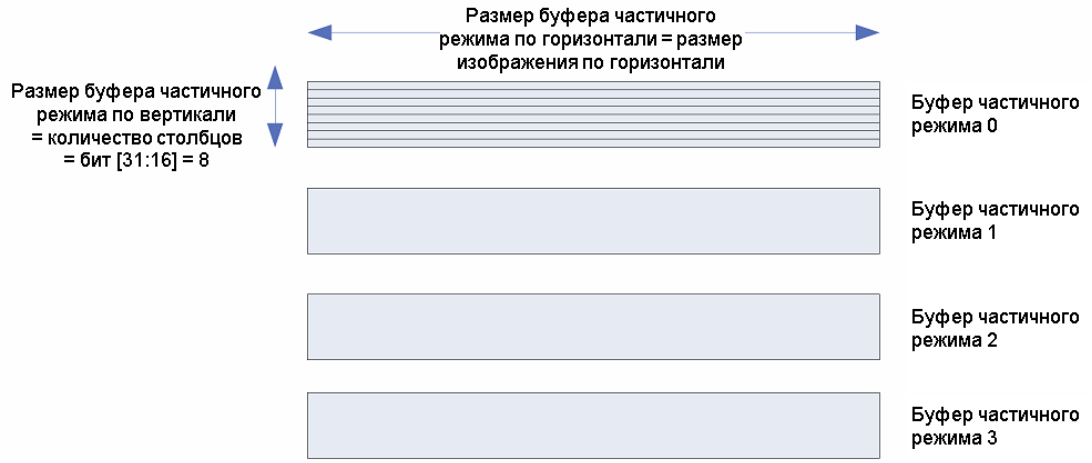


Рисунок 1.125 - Размер буфера частичного режима

Таблица 1.176 – Регистр MJPEG_DPB_CONFIG_REG (0x030)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_dpb_config_reg	Запись/чтение	Конфигурация буфера декодированных изображений (DPB) [7:6] – порядок следования байтов буфера DPB [5:2] – упакованный десятичный код - 4'b0000: упакованный десятичный код запрещен - 4'b1000: упакованный десятичный код YUV444 - 4'b0100: YUYV - 4'b0101: UYVY - 4'b0110: YVYU - 4'b0111: VYUY Другие значения не допускаются. [1:0] – режим чередования uv буфера DPB - 2'b00: нормальный - 2'b10: чередование CbCr (например, NV12 в 4:2:0 или NV16 в 4:2:2) - 2'b11: чередование CrCb (например, NV21 в 4:2:0) Установка значения битов режима упакованного десятичного кода [5:2] и режима чередования uv [1:0] является взаимоисключающей. Когда главный компьютер задает режим чередования uv, режим упакованного десятичного кода должен быть выключен. Таким образом, поле [5:0] будет иметь одно из следующих значений: • 0000 00 : двумерный • 0000 10 : полудвумерный cbcr • 0000 11 : полудвумерный crcb • 1000 00 : 444 упакованный • 0100 00 : 422 упакованный YUYV • 0101 00 : 422 упакованный UYUV • 0110 00 : 422 упакованный YVYU • 0111 00 : 422 упакованный VYVU Примечание Описание режима упакованного десятичного кода приведено в Приложении E «УПАКОВАННЫЙ ДЕСЯТИЧНЫЙ ФОРМАТ»	0

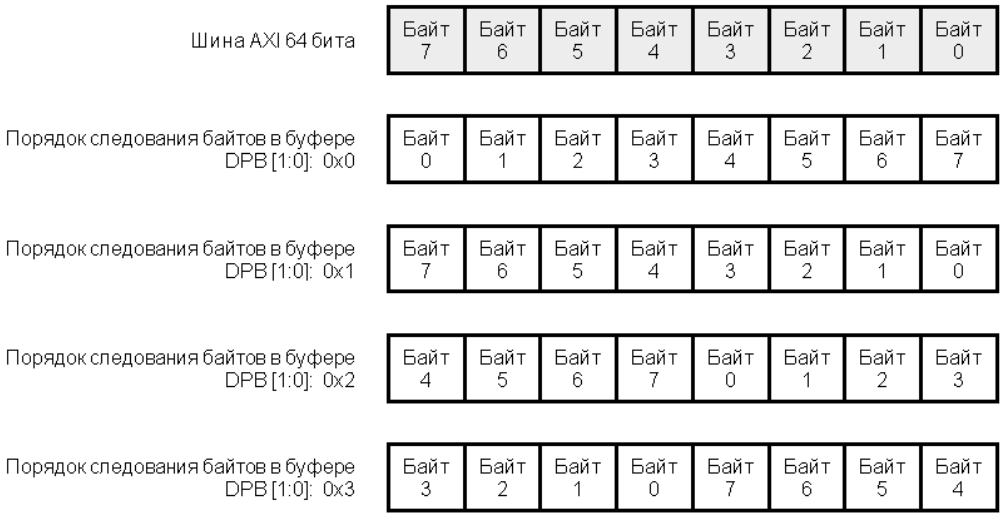


Рисунок 1.126 - Порядок следования байтов в буфере кадров

Таблица 1.177 – Регистр MJPEG_DPB_BASE00_REG (0x034)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base00_reg	Запись/чтение	База DPB 0 для компонента ID 0 Адреса с «Base+0x034» до «Base+0x060» - это адреса баз буфера кадров или буфера частичного режима. Существуют 4 адреса, и каждый адрес включает адреса 3 компонентов Y, Cb и Cr. В нормальном режиме работы используются только адреса «Base + 0x034», «Base + 0x038» и «Base + 0x03C». В частичном режиме доступные адреса определяются битом [5:3] регистра MJPEG_OP_INFO_REG (Base + 02C). Например, если значение бита [5:3] в регистре MJPEG_OP_INFO_REG составляет «4», используются все адреса, а если значение составляет «2», для кодера или декодера блока JPU используются два адреса (от «Base + 0x034» до «Base + 0x048»).	0

Таблица 1.178 – Регистр MJPEG_DPB_BASE01_REG (0x038)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base01_reg	Запись/чтение	База DPB 0 для компонента ID 1	0

Таблица 1.179 – Регистр MJPEG_DPB_BASE02_REG (0x03C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base02_reg	Запись/чтение	База DPB 0 для компонента ID 2	0

Таблица 1.180 – Регистр MJPEG_DPB_BASE10_REG (0x040)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base10_reg	Запись/чтение	База DPB 1 для компонента ID 0	0

Таблица 1.181 – Регистр MJPEG_DPB_BASE11_REG (0x044)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base11_reg	Запись/чтение	База DPB 1 для компонента ID 1	0

Таблица 1.182 – Регистр MJPEG_DPB_BASE12_REG (0x048)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base12_reg	Запись/чтение	База DPB 1 для компонента ID 2	0

Таблица 1.183 – Регистр MJPEG_DPB_BASE20_REG (0x04C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base20_reg	Запись/чтение	База DPB 2 для компонента ID 0	0

Таблица 1.184 – Регистр MJPEG_DPB_BASE21_REG (0x050)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base21_reg	Запись/чтение	База DPB 2 для компонента ID 1	0

Таблица 1.185 – Регистр MJPEG_DPB_BASE22_REG (0x054)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base22_reg	Запись/чтение	База DPB 2 для компонента ID 2	0

Таблица 1.86 – Регистр MJPEG_DPB_BASE30_REG (0x058)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base30_reg	Запись/чтение	База DPB 3 для компонента ID 0	0

Таблица 1.187 – Регистр MJPEG_DPB_BASE31_REG (0x05c)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base31_reg	Запись/чтение	База DPB 3 для компонента ID 1	0

Таблица 1.88 – Регистр MJPEG_DPB_BASE32_REG (0x060)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_dpb_base32_reg	Запись/чтение	База DPB 3 для компонента ID 2	0

Таблица 1.89 – Регистр MJPEG_DPB_YSTRIDE_REG (0x064)

Бит	Наименование	Тип	Описание	Значение параметра сброса
15:0	mjpeg_dpb_ystride_reg	Запись/чтение	Шаг по индексу компонента яркости DPB Данный регистр указывает шаг по индексу компонента яркости в буфере кадров. Значение шага по индексу должно быть кратно 8, так как блок JPU 64-битную (8-байтную) шину AXI.	0

Таблица 1.190 – Регистр MJPEG_DPB_CSTRIDE_REG (0x068)

Бит	Наименование	Тип	Описание	Значение параметра сброса
15:0	mjpeg_dpb_cstride_reg	Запись/чтение	Шаг по индексу компонента цветности DPB Данный регистр указывает шаг по индексу компонента цветности в буфере кадров. Значение шага по индексу должно быть кратно 8, так как блок JPU 64-битную (8-байтную) шину AXI.	0

Таблица 1.191 – Регистр MJPEG_WRESP_CHECK_REG (0x06C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
0	mjpeg_wresp_check_ reg	Чтение	Регистр проверки канала отклика операций записи шины AXI 0 : Блок обработки видеоинформации получил сигнал отклика на операцию записи. Произведена операция записи. 1 : Блок обработки видеоинформации еще не получил сигнал отклика на операцию записи.	0

Таблица 1.192 – Регистр MJPEG_CLP_BASE_REG (0x070)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_clp_base_reg	Запись/чтение	Начальная позиция для зоны ROI в пикселях Бит [31:16] : зона ROI по оси x Бит [15:0] : зона ROI по оси y Это настройка в пикселях, но зона ROI измеряется в минимальных кодированных блоках. Например, если положения по оси x и y составляют 15 каждое в формате 4:2:0, то положение по данным осям принадлежит первому минимальному кодированному блоку. Таким образом, блок JPU начинает работу с зоной ROI с координат (0, 0).	0

Таблица 1.193 – Регистр MJPEG_CLP_SIZE_REG (0x074)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_clp_size_reg	Запись/чтение	Координаты по осям X и Y для зоны ROI в пикселях. Бит [31:16]: координаты зоны ROI по оси x Бит [15:0] : координаты зоны ROI по оси y Это настройка в пикселях, но зона ROI находится в минимальном кодированном блоке. Например, если положения по оси x и y составляют 16 каждое в формате 4:2:0, то общее количество минимальных кодированных блоков для данной зоны ROI будет составлять «один».	0

Таблица 1.194 – Регистр MJPEG_HUFF_CTRL_REG (0x080)

Бит	Наименование	Тип	Описание	Значение параметра сброса
11:0	mjpeg_huff_ctrl_reg	Запись/чтение	Регистр управления таблицей Хаффмана [11:10] – адреса назначения таблицы Хаффмана [1] – автоматический режим [0] - режим	0

Таблица 1.195 – Регистр MJPEG_HUFF_ADDR_REG (0x084)

Бит	Наименование	Тип	Описание	Значение параметра сброса
11:0	mjpeg_huff_ctrl_reg	Запись/чтение	Регистр адреса таблицы Хаффмана [11:10] – адреса назначения таблицы Хаффмана [9:0] – базовый адрес таблицы Хаффмана	0

Таблица 1.196 – Регистр MJPEG_HUFF_DATA_REG (0x088)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_huff_data_reg	Запись/чтение	Регистр данных таблицы Хаффмана Подробная информация о загрузке таблицы Хаффмана для режима кодера приведена в описании задачи vtJpgEncHuffTabSetUp тестового модуля RTL и функции JpgEncLoadHuffTab исходного программного обеспечения. Подробная информация о декодере приведена в описании задачи vtJpgDecHuffTabSetUp тестового модуля RTL и функции JpgDecHuffTabSetUp исходного программного обеспечения.	0

Таблица 1.197 – Регистр MJPEG_QMAT_CTRL_REG (0x090)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_qmat_ctrl_reg	Запись/чтение	Регистр управления матрицей квантования [7:6] - адреса назначения матрицы квантования [1] - автоматический режим [0] - режим	0

Таблица 1.198 – Регистр MJPEG_QMAT_ADDR_REG (0x094)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_qmat_addr_reg	Запись/чтение	Регистр адреса матрицы квантования [7:6] - адреса назначения матрицы квантования [5:0] – базовый адрес матрицы квантования	0

Таблица 1.199 – Регистр MJPEG_QMAT_DATA_REG (0x098)

Бит	Наименование	Тип	Описание	Значение параметра сброса
19:0	mjpeg_qmat_data_reg	Запись/чтение	Регистр данных матрицы квантования Для хранения матрицы квантования блок JPU использует память шириной 40 бит (20 бит+ 20 бит). Главный компьютер производит запись данных в регистр [19:0] через 32-битную шину APB, после чего осуществляет копирование данных в память. [39:20]. Способ сохранения данных матрицы квантования в 40-битную память. Подробная информация о загрузке таблицы квантования для режима кодера приведена в описании задачи vtJpgEncQMatTabSetUp тестового модуля RTL и функции JpgEncLoadQMatTab исходного программного обеспечения. Подробная информация о декодере приведена в описании задачи vtJpgDecQMatTabSetUp тестового модуля RTL и функции JpgDecQMatTabSetUp исходного программного обеспечения.	

Таблица 1.200 – Регистр MJPEG_COEF_CTRL_REG (0x0A0)

Бит	Наименование	Тип	Описание	Значение параметра сброса
8:0	mjpeg_coef_ctrl_reg	Запись/чтение	Регистр управления памятью, содержащей коэффициенты [8:6] - адреса назначения памяти, содержащей коэффициенты [1] – автоматический режим [0] - режим	0

Таблица 1.201 – Регистр MJPEG_COEF_ADDR_REG (0x0A4)

Бит	Наименование	Тип	Описание	Значение параметра сброса
8:0	mjpeg_coef_addr_reg	Запись/чтение	Регистр адреса памяти, содержащей коэффициенты [8:6] - адреса назначения памяти, содержащей коэффициенты [5:0] – основной адрес памяти, содержащей коэффициенты	0

Таблица 1.202 – Регистр MJPEG_COEF_DATA_REG (0x0A8)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_coef_data_reg	Запись/чтение	Регистр данных памяти, содержащей коэффициенты	0

Таблица 1.203 – Регистр MJPEG_RST_INTERVAL_REG (0x0B0)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_rst_intval_reg	Запись/чтение	Регистр интервала перезапуска	0

Таблица 1.204 – Регистр MJPEG_RST_INDEX_REG (0x0B4)

Бит	Наименование	Тип	Описание	Значение параметра сброса
25:0	mjpeg_rst_index_reg	Запись/чтение	Регистр маркера перезапуска	0

Таблица 1.205 – Регистр MJPEG_RST_COUNT_REG (0x0B8)

Бит	Наименование	Тип	Описание	Значение параметра сброса
25:0	mjpeg_rst_count_reg	Запись/чтение	Регистр счетчика перезапусков	0

Таблица 1.206 – Регистр MJPEG_INTR_MASK_REG (0x0C0)

Бит	Наименование	Тип	Описание	Значение параметра сброса
8:0	mjpeg_intr_mask_reg	Запись/чтение	Маски прерываний [8] – остановка обработки JPEG изображений [7] – буфер частичного режима 3 [6] – буфер частичного режима 2 [5] – буфер частичного режима 1 [4] – буфер частичного режима 0 [3] – переполнение буфера при обработке JPEG изображений [2] – прерывание ВВС [1] – ошибка обработки JPEG изображений [0] – обработка JPEG изображений завершена	0

Таблица 1.207 – Регистр MJPEG_CYCLE_INFO_REG (0x0C8)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_cycle_info_reg	Чтение	Отчет о полном цикле Начинает формироваться при запуске кодирования/декодирования изображений в формате JPEG и полностью генерируется по завершении операции кодирования/декодирования изображений в формате JPEG.	0

Таблица 1.208 – Регистр MJPEG_DPCM_DIFF_Y_REG (0x0F0)

Бит	Наименование	Тип	Описание	Значение параметра сброса
11:0	mjpeg_dpcm_diff_y _reg	Запись/чтение	Коэффициент DC для компонента 0	0

Таблица 1.209 – Регистр MJPEG_DPCM_DIFF_CB_REG (0x0F4)

Бит	Наименование	Тип	Описание	Значение параметра сброса
11:0	mjpeg_dpcm_diff_cb _reg	Запись/чтение	Коэффициент DC для компонента 1	0

Таблица 1.210 – Регистр MJPEG_DPCM_DIFF_CR_REG(0x0F8)

Бит	Наименование	Тип	Описание	Значение параметра сброса
11:0	mjpeg_dpcm_diff_cr _reg	Запись/чтение	Коэффициент DC для компонента 2	0

Регистры управления блоком GBU

Блок GBU (Get Bit Unit – блок получения битов) имеет внутренний буфер битовых потоков под названием GRAM, и размер буфера GRAM составляет 512 байт.

Таблица 1.211 – Регистр MJPEG_GBU_CTRL_REG (BASE+0x100)

Бит	Наименование	Тип	Описание	Значение параметра сброса
3:0	mjpeg_gbu_ctrl_reg	Запись/чтение	Регистр управления GBU [3] – задание значения бита для очистки GBU для кодера [2] – инициализация блока GBU для декодера [1] - инициализация блока GBU для модуля эмуляции декодера [0] - резерв Бит [3] означает, что в конце битового потока JPEG вставляется 1-битное значение для выравнивания по границе слова. Поэтому если значение данного поля установлено в «1», другие биты после сигнала EOI приобретают значение «1». Если значение данного поля установлено в «0», другие биты после сигнала EOI приобретают значение «0»	0

Таблица 1.212 – Регистр MJPEG_GBU_PBIT_BUSY_REG (BASE+0x104)

Бит	Наименование	Тип	Описание	Значение параметра сброса
0	mjpeg_gbu_pbit_busy_reg	Чтение	Флаг занятости при записи битов блоком GBU Данный флаг индицирует состояние занятости блока GBU в момент приема данным блоком битов, при этом команда на данную операцию отправляется главным компьютером. Следовательно, главный компьютер должен произвести считывание с соответствующего регистра для проверки состояния блока GBU до ввода в действие регистров MJPEG_GBU_PBIT_08_REG, MJPEG_GBU_PBIT_16_REG, MJPEG_GBU_PBIT_24_REG и MJPEG_GBU_PBIT_32_REG	0

Таблица 1.213 – Регистр MJPEG_GBU_BT_PTR_REG (BASE+0x110)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_gbu_bt_ptr_reg	Запись/чтение	Регистр-указатель текущего бита буфера GRAM	0

Таблица 1.214 – Регистр MJPEG_GBU_WD_PTR_REG (BASE+0x114)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_gbu_wd_ptr_reg	Запись/чтение	Регистр-указатель текущего слова буфера GRAM	0

Таблица 1.215 – Регистр MJPEG_GBU_TT_CNT_REG (BASE+0x118)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_gbu_tt_cnt_reg	Запись/чтение	Общий подсчет битов Для кодера это накопленное общее количество битов, сгенерированных блоком JPU в качестве записываемых битов. Для декодера это накопленный общий размер битов, используемых блоком JPU в качестве принимаемых битов. Данные значения продолжают увеличиваться с момента перезагрузки системы, если главный компьютер не производит принудительный сброс счетчика битов.	0

Таблица 1.216 – Регистр MJPEG_GBU_TT_CNT_REG (BASE+0x11C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
63:32	mjpeg_gbu_tt_cnt_reg	Запись/чтение	Общий подсчет битов Для кодера это накопленное общее количество битов, сгенерированных блоком JPU в качестве записываемых битов. Для декодера это накопленный общий размер битов, используемых блоком JPU в качестве принимаемых битов. Данные значения продолжают увеличиваться с момента перезагрузки системы, если главный компьютер не производит принудительный сброс счетчика битов.	0

Таблица 1.217 – Регистр MJPEG_GBU_PBIT_08_REG (BASE+0x120)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_gbu_pbit_08_ reg	Запись	8-битные данные, передаваемые в буфер битовых потоков кодера JPEG Значение, записываемое соответствующим регистром в кодер JPEG, будет временно сохранено в регистре блока GBU, после чего перемещено во внешний буфер битовых потоков.	0

Таблица 1.218 – Регистр MJPEG_GBU_PBIT_16_REG (BASE+0x124)

Бит	Наименование	Тип	Описание	Значение параметра сброса
15:0	mjpeg_gbu_pbit_16_reg	Запись	16-битные данные, передаваемые в буфер битовых потоков кодера JPEG Значение, записываемое соответствующим регистром в кодер JPEG, будет временно сохранено в регистре блока GBU, после чего перемещено во внешний буфер битовых потоков.	0

Таблица 1.219 – Регистр MJPEG_GBU_PBIT_24_REG (BASE+0x128)

Бит	Наименование	Тип	Описание	Значение параметра сброса
23:0	mjpeg_gbu_pbit_24_reg	Запись	24-битные данные, передаваемые в буфер битовых потоков кодера JPEG Значение, записываемое соответствующим регистром в кодер JPEG, будет временно сохранено в регистре блока GBU, после чего перемещено во внешний буфер битовых потоков.	0

Таблица 1.220 – Регистр MJPEG_GBU_PBIT_32_REG (BASE+0x12C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_gbu_pbit_32_reg	Запись	32-битные данные, передаваемые в буфер битовых потоков кодера JPEG Значение, записываемое соответствующим регистром в кодер JPEG, будет временно сохранено в регистре блока GBU, после чего перемещено во внешний буфер битовых потоков.	0

Таблица 1.221 – Регистр MJPEG_GBU_BBSR_REG (BASE+0x140)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_gbu_bbsr_reg	Запись/чтение	Регистр-указатель начала потока данных буфера GRAM	0

Таблица 1.222 – Регистр MJPEG_GBU_BBER_REG (BASE+0x144)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_gbu_bber_reg	Запись/чтение	Регистр-указатель конца потока данных буфера GRAM	0

Таблица 1.223 – Регистр MJPEG_GBU_BBIR_REG (BASE+0x148)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_gbu_bbir_reg	Запись/чтение	Регистр-указатель прерываний буфера GRAM	0

Таблица 1.224 – Регистр MJPEG_GBU_BBHR_REG (BASE+0x14C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_gbu_bbhr_reg	Запись/чтение	Это регистр остановки Если блок GBU пуст, возникает прерывание, и главный компьютер производит установку данного регистра. Коэффициент преобразования ТС декодера/кодера JPEG переходит в режим ожидания до тех пор, пока значение данного регистра не будет установлено в «0».	0

Таблица 1.225 – Регистр MJPEG_GBU_BCNT_REG (BASE+0x158)

Бит	Наименование	Тип	Описание	Значение параметра сброса
15:0	mjpeg_gbu_bcnt_reg	Запись/чтение	Общий подсчет битов Для кодера это накопленное общее количество битов, сгенерированных блоком JPU в качестве записываемых битов. Для декодера это накопленный общий размер битов, используемых блоком JPU в качестве принимаемых битов. При каждой команде запуска кодирования/декодирования данный регистр производит сброс счетчика.	0

Таблица 1.226 – Регистр MJPEG_GBU_FF_RPTR_REG (BASE+0x160)

Бит	Наименование	Тип	Описание	Значение параметра сброса
6:0	mjpeg_gbu_ff_rptr_reg	Запись/чтение	Текущий регистр-указатель чтения буфера GRAM для эмулятора FF	0

Таблица 1.227 – Регистр MJPEG_GBU_FF_WPTR_REG (BASE+0x164)

Бит	Наименование	Тип	Описание	Значение параметра сброса
6:0	mjpeg_gbu_ff_wptr_reg	Запись/чтение	Текущий регистр-указатель записи в буфер GRAM для эмулятора FF	0

1.3.6.2.2.2.2.2 Регистры управления блоком BBC

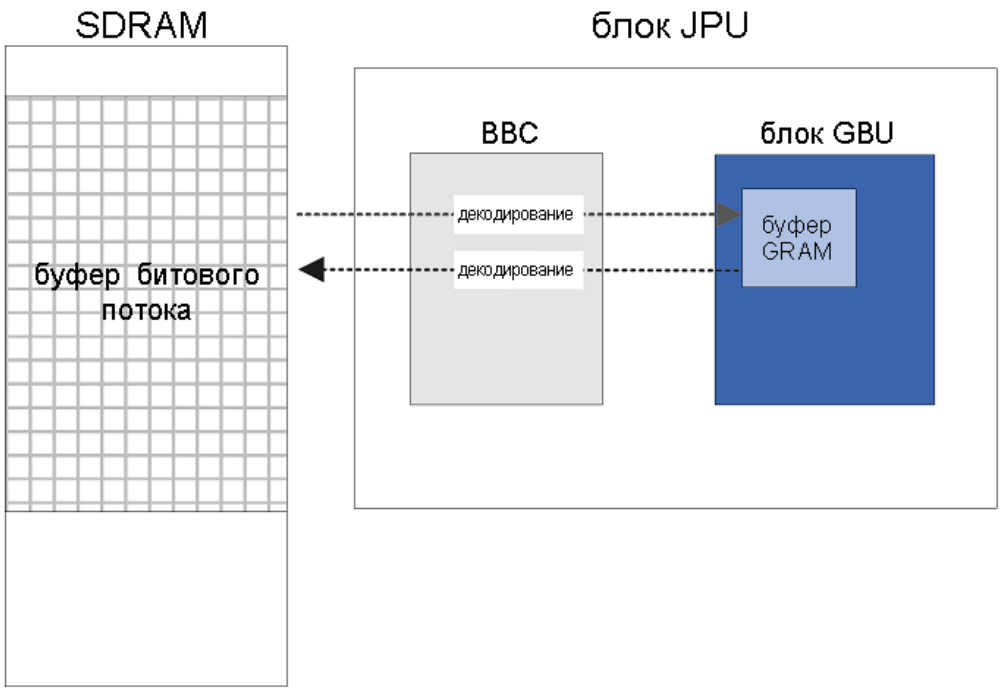


Рисунок 1.126 - Прохождение битового потока через блок управления буфером битовых потоков

Блок BBC (Bitstream Buffer Control - блок управления буфером битовых потоков) осуществляет копирование битовых потоков из внешнего буфера битовых потоков на SDRAM во внутренний буфер битовых потоков (GRAM) или перемещает битовый поток из буфера GRAM во внешний буфер битовых потоков.

Примечание - Информация по управлению буфером битовых потоков в данном исходном программном обеспечении приведена в пояснении Приложения D «УПРАВЛЕНИЕ БУФЕРОМ БИТОВЫХ ПОТОКОВ В ИСХОДНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ».

Таблица 1.228 – Регистр MJPEG_BBC_ENDADDR_REG (0x208)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_bbc_endaddr_ reg	Запись/ чтение	Регистр адреса команды прерывания потока данных внешнего буфера битовых потоков Когда значение данного регистра достигает mjpeg_bbc_wr_ptr_reg/ mjpeg_bbc_rd_ptr_reg, главный компьютер должен произвести обработку прерывания. В кодере главный компьютер производит считывание битового потока из внешней памяти. В декодере главный компьютер производит запись битового потока во внешнюю память. Если команда блока BBC установлена в «1», блок BBC осуществляет перенос данных из внутренней/внешней памяти во внешнюю/внутреннюю память до тех пор, пока не будет достигнуто значение mjpeg_bbc_int_addr_reg в случае кодера/декодера.	0

Таблица 1.229 – Регистр MJPEG_BBC_WR_PTR_REG (0x20C)

Бит	Наименование	Тип	Описание	Значение

				параметра сброса
31:0	mjpeg_bbc_wr_ptr_reg	Запись/чтение	Регистр-указатель записи потока данных внешнего буфера битовых потоков	0

Таблица 1.230 – Регистр MJPEG_BBC_RD_PTR_REG (0x210)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_bbc_rd_ptr_reg	Запись/чтение	Регистр-указатель чтения потока данных внешнего буфера битовых потоков	0

Таблица 1.231 – Регистр MJPEG_BBC_EXT_ADDR_REG (0x214)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_bbc_ext_addr_ reg	Запись/чтение	Регистр адреса начала потока данных внешнего буфера битовых потоков	0

Таблица 1.232 – Регистр MJPEG_BBC_INT_ADDR_REG (0x218)

Бит	Наименование	Тип	Описание	Значение параметра сброса
6:0	mjpeg_bbc_int_addr_ reg	Запись/чтение	Регистр адреса начала потока данных внутреннего буфера битовых потоков в 4-байтном блоке	0

Таблица 1.233 – Регистр MJPEG_BBC_DATA_CNT_REG (0x21C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
7:0	mjpeg_bbc_data_cnt_ reg	Запись/чтение	Размер передаваемых данных в 4-байтном блоке	0

Таблица 1.234 – Регистр MJPEG_BBC_COMMAND_REG (0x220)

Бит	Наименование	Тип	Описание	Значение параметра сброса
2:0	mjpeg_bbc_command_ reg	Запись/чтение	Регистр команд обращения к памяти [2:1] – регистр порядка следования байтов [0] – команда загрузки/сохранения Данный регистр может использоваться для передачи данных из внутренней памяти во внешнюю память или из внешней памяти во внутреннюю. Если пользователь задает значение данного регистра «1», аппаратное обеспечение производит перенос данных из внутренней во внешнюю память. В остальных случаях аппаратное обеспечение работает противоположным образом. Команда обращения к интерфейсу памяти зависит от системы. Второй бит означает режим прямого/обратного порядка следования байтов. И третий бит означает ширину данных 64 бита/32 бита. Если машина пользователя работает в режиме с прямым порядком следования байтов 64 бита, необходимо задать значение «2'b01».	0

Таблица 1.235 – Регистр MJPEG_BBC_BUSY_REG (0x224)

Бит	Наименование	Тип	Описание	Значение параметра

				сброса
0	mjpeg_bbc_busy_reg	Чтение	Регистр-индикатор занятости блока BBC Главный компьютер через данный регистр получает информацию о том, занято аппаратное обеспечение или нет. Если определенный блок, использующий данную команду, находится в состоянии занятости, возможно, аппаратное обеспечение не функционирует надлежащим образом.	0

Таблица 1.236 – Регистр MJPEG_BBC_CTRL_REG (0x228)

Бит	Наименование	Тип	Описание	Значение параметра сброса
2:0	mjpeg_bbc_ctrl_reg	Запись/ чтение	Регистр управления блоком BBC [2:1] – регистр порядка следования байтов [0] – команда автоматического запуска блока BBC После заполнения буфера GRAM на этапе инициализации операции декодирования, главный компьютер может записать данный регистр как «1», и блок BBC автоматически поместит битовый поток в буфер GRAM.	0

Таблица 1.237 – Регистр MJPEG_BBC_CUR_POS_REG (0x22C)

Бит	Наименование	Тип	Описание	Значение параметра сброса
23:0	mjpeg_bbc_cur_pos_ reg	Запись/ чтение	Текущее положение внешнего буфера битовых потоков в блоке страницы	0

Таблица 1.238 – Регистр MJPEG_BBC_BAS_ADDR_REG (0x230)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_bbc_bas_addr_ reg	Запись/ чтение	Базовый адрес внешнего буфера битовых потоков	0

Таблица 1.239 – Регистр MJPEG_BBC_STRM_CTRL_REG (0x234)

Бит	Наименование	Тип	Описание	Значение параметра сброса
31:0	mjpeg_bbc_strm_ctrl_ reg	Запись/ чтение	Данные управления битовым потоком (только декодер) Когда главный компьютер загружает битовые потоки во внешний буфер битовых потоков, может случиться так, что текущий битовый поток окажется меньше буфера битовых потоков. Затем через данный регистр главный компьютер может задать EOF (End of File – конец файла) для файла битового потока и счетчик битовых потоков, чтобы блок BBC по достижению заданного значения остановил закачку битовых потоков. [31] – конец файла битового потока [23:0] – счетчик битовых потоков	0

Таблица 1.240 – Регистр MJPEG_BBC_FLUSH_CMD_REG(0x238)

Бит	Наименование	Тип	Описание	Значение параметра сброса
0	mjpeg_bbc_flush_	Запись/	Регистр команды завершения очистки для внутреннего	0

	cmd_reg	чтение	буфера битовых потоков (только кодер) Для успешного кодирования главному компьютеру необходимо удалить оставшийся битовый поток из внутреннего буфер битовых потоков и сбросить данный регистр на 0 для инициализации конечного автомата блока ВВС, который обеспечивает операцию освобождения буфера.	

1.3.6.2.3 УПРАВЛЕНИЕ БЛОКОМ JPU

Блок JPU (JPEG Processing Unit – блок обработки изображений в формате JPEG) – это аппаратный блок кодека JPEG фирмы Chips&Media, осуществляющий кодирование изображений в формате YUV в битовый поток JPEG и декодирование битового потока JPEG. Схема управления блоком JPU отличается от существующей схемы управления блоком VPU (Video Processing Unit – блок обработки видеоинформации). Главный компьютер должен следить за выполнением задач уплотнения/интерпретации заголовков JPEG файлов и задавать необходимые параметры в соответствующих регистрах JPU, чтобы блок JPU мог осуществлять кодирование или декодирование в соответствии с параметрами регистров, сконфигурированными главным компьютером. Главному компьютеру необходимо понимать синтаксическую структуру заголовков JPEG файлов, а также как информация, полученная после синтаксического разбора заголовка, будет влиять на результат декодирования сегмента ECS.

Рисунок 3.1 «Синтаксическая структура заголовка JPEG» отображает базовую синтаксическую структуру JPEG, указанную в Спецификации JPEG, ISO/IEC 10918-1.

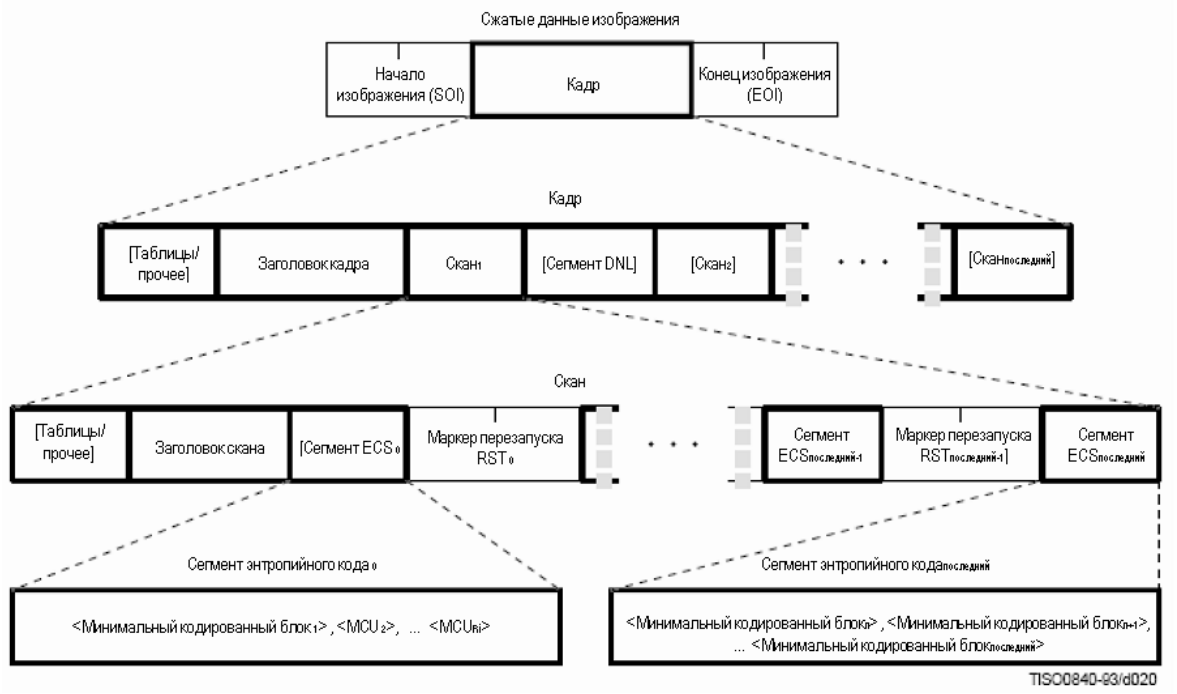


Рисунок 1.127 - Синтаксическая структура заголовка файла JPEG

Более подробная информация о синтаксической структуре заголовка JPEG приведена в стандарте ISO/IEC 10918-1 или CCITT Rec. T.81 Annex B Compressed data formats (Приложение B «Форматы сжатых данных»).

1.3.6.2.3.1 Управление кодером

Кодирование JPEG – это процесс преобразования изображений в формате YUV в битовый поток JPEG. Главный компьютер помогает блоку JPU осуществлять операции кодирования, предоставляя изображения YUV, задавая соответствующие значения в регистрах JPU, запуская блок JPU, осуществляя уплотнение синтаксической структуры заголовка, получая битовый поток (ECS + RST), генерируемый блоком JPU. Перед тем как продолжить главный компьютер должен определить следующие моменты:

- Информация об изображении, такая как ширина и высота изображения, а также формат цветного изображения: данная информация впоследствии используется для

формирования синтаксической структуры заголовка кадра (SOF) в процессе уплотнения синтаксической структуры заголовка.

- Информация о таблице, такая как таблицы квантования, таблицы Хаффмана и табличный индекс каждого цветового компонента: в процессе уплотнения синтаксиса заголовка таблица квантования приобретает синтаксическую структуру таблицы дискретизации (DQT), а и таблица Хаффмана - синтаксическую структуру таблицы Хаффмана (DHT), соответственно.
- Информация о перезапуске для обеспечения устойчивости к ошибкам: данная информация в ходе уплотнения заголовка приобретает синтаксическую структуру интервала перезапуска (DRI).

После определения данных факторов кодирования главный компьютер может задать значения регистров блока JPU в соответствии с полученной информацией, и блок JPU может приступить к работе.

1.3.6.2.3.1.1 Инициализация последовательности

Сначала главный компьютер должен задать следующую информацию в регистрах блока JPU, чтобы кодер JPU смог начать операцию кодирования.

Информация об изображении и минимальном кодированном блоке

Главный компьютер должен задать параметры изображения, таблицы и перезапуска в регистрах блока JPU. Необходимые регистры можно найти в Разделе 2.2.2.1 «Регистры главного устройства управления» настоящего документа и функции EncSeqInit() исходного предоставляемого программного обеспечения.

Внутренний буфер битовых потоков

Главный компьютер должен выделить буфер битовых потоков для кодированного битового потока. Существуют два буфера битовых потоков, необходимые для работы блока JPU: внутренний и внешний буферы битовых потоков. Блок JPU содержит небольшой блок, называемый GBU (Get Bit Unit – блок получения битов), и внутренний буфер битовых потоков принадлежит блоку GBU, который сохраняет кодированный битовый поток во внутренний буфер битовых потоков или автоматически перемещает его во внешний буфер битовых потоков в случае, если внутренний буфер заполнен настройками. Размер внутреннего буфера битовых потоков составляет 512 байтов, и блок GBU работает в едином 512-байтовом буфере или в двух 256-байтовых буферах (двойном буфере), в зависимости от настроек.

Подробная информация о соответствующих регистрах приведена в Разделе 2.2.2.2 «Регистры управления блоком GBU» настоящего документа и функции EncSeqInit() исходного предоставляемого программного обеспечения. Для упрощения работы пользователя исходное программное обеспечение использует схему двойного буфера для внутреннего буфера битовых потоков для предотвращения снижения производительности в результате невозможности быстрого освобождения буфера.

Внешний буфер битовых потоков

Другой вышеупомянутый буфер является внешним буфером битовых потоков. Буфер находится в памяти системы (внешняя память DRAM или SDRAM), и доступ к нему может осуществляться как блоком JPU, так и главным компьютером. Блок JPU перемещает кодированный битовый поток из внутреннего буфера битовых потоков во внешний буфер битовых потоков через блок BBC (Bitstream Buffer Control – блок управления буфером битовых потоков) и имеет прямой доступ к памяти для целей копирования буферных потоков. Главный компьютер впоследствии может запросить битовый поток из внешнего буфера битовых потоков. Если внешний буфер битовых потоков заполнен, блок JPU выдает на главный компьютер сигнал о прерывании и переходит в режим ожидания до опустошения буфера.

Главный компьютер должен задать адрес начала, адрес конца, указатель чтения, указатель записи и т.д. внешнего буфера битовых потоков в соответствующих регистрах блока BBC. Подробная информация о соответствующих регистрах приведена в Разделе 2.2.2.2 «Регистры

управления блоком GBU» настоящего документа и функции EncSeqInit() исходного предоставляемого программного обеспечения.

1.3.6.2.3.1.2 Функции буфера кадров

Блоку JPU, собственно, для кодирования необходимо исходное YUV изображение. Главный компьютер должен выделить буфер кадров для исходного изображения и задать соответствующую информацию относительно буфера кадров.

Обычно блок JPU использует один буфер кадров для кодирования одного статического изображения. Тем не менее, в приложениях реального времени блок JPU должен поддерживать кодирование ряда изображений и поэтому для обеспечения максимальной производительности рекомендуется использовать два или более буферов кадров. Перед запуском блока JPU необходимо загрузить в буфер кадров не менее одного исходного изображения. При использовании двух или более буферов кадров загрузка изображения и запуск блока JPU могут производиться одновременно.

1.3.6.2.3.1.3 Работа блока JPU в режиме кодера

1.3.6.2.3.1.3.1 Загрузка YUV изображения

Перед запуском операции кодирования изображения главный компьютер должен загрузить формат цветного YUV изображения в буфер кадров.

1.3.6.2.3.1.3.2 Запуск кодера

Главный компьютер должен задать параметры буфера кадров, в который загружено YUV изображение, в регистрах блока JPU. Обычно выделение буфера кадров и конфигурирование параметров регистров могут производиться одновременно. При работе нескольких буферов кадров исходное программное обеспечение отображает настройки регистров буферов кадров непосредственно перед запуском блока JPU.

Подробная информация о задании значений в регистрах приведена в описании регистров MJPEG_DPB_* в Разделе 2.2.2.1 «Регистры главного устройства управления» и функции EncPicRun() исходного предоставляемого программного обеспечения. После задания настроек буфера кадров главный компьютер может запустить операцию кодирования через регистр MJPEG_PIC_START_REG.

1.3.6.2.3.1.3.3 Освобождение буфера битовых потоков

В процессе кодирования блок JPU продолжает генерировать закодированную строку. Если размер буфера битовых потоков достаточный, блок JPU может беспроблемно продолжать кодирование. Тем не менее, если буфер битовых потоков быстро заполняется вследствие недостаточного объема буфера, блок JPU выдает сигнал прерывания и переходит в режим ожидания до освобождения буфера. Главный компьютер проверяет сигнал прерывания от блока JPU и производит очистку внешнего буфера битовых потока.

Управление внешним буфером битовых потоков может осуществляться несколькими способами настройками, заданными главным компьютером, как и внутренним буфером битовых потоков. Исходное программное обеспечение, предоставляемое нами, использует несколько страниц для внешнего буфера битовых потоков, и объем каждой страницы составляет 256 байтов, как и для внутреннего буфера битовых потоков. Более подробная информация приведена в Приложении D «УПРАВЛЕНИЕ БУФЕРОМ БИТОВЫХ ПОТОКОВ ПОСРЕДСТВОМ ИСХОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ» настоящего документа. Главный компьютер может определить степень заполнения внешнего буфера битовых потоков через регистр MJPEG_PIC_STATUS_REG.

1.3.6.2.3.1.3.4 Прекращение кодирования

Операция кодирования изображения занимает определенное количество времени, и главный компьютер в процессе ожидания завершения операции кодирования изображения может перейти к выполнению других задач. Главный компьютер может объявить о завершении кодирования изображения посредством опроса регистра состояния (MJPEG_PIC_STATUS_REG) или генерацией сигнала прерывания.

Регистр состояния отображает состояние внешнего буфера битовых потоков, а также состояние завершения кодирования. Проинформированный сигналом прерывания, главный компьютер должен проверить регистр MJPEG_PIC_STATUS_REG для определения причины прерывания.

После завершения операции кодирования главный компьютер производит очистку буфера битовых потоков от оставшихся данных и сбрасывает сигнал прерывания по завершению кодирования через регистр MJPEG_PIC_STATUS_REG.

1.3.6.2.3.1.3.5 Уплотнение заголовка

Блок может создавать только комбинацию «сегмент ECS + маркер перезапуска RST» из исходного YUV изображения, и главному компьютеру необходимо прикрепить синтаксическую структуру заголовка к сгенерированному битовому потоку для получения завершенного битового потока JPEG. К счастью, кодер JPU может вставлять синтаксис заголовка скана (SOS) и маркер EOI для кодирования комбинации «сегмент ECS + маркер перезапуска RST» для упрощения работы главного компьютера, после чего компьютер должен добавить следующую информацию:

- маркер SOI
- синтаксическую структуру таблицы дискретизации (DQT)
- синтаксическую структуру таблицы Хаффмана (DHT)
- синтаксическую структуру интервала перезапуска (DRI)
- синтаксическую структуру заголовка кадра (SOF)
- синтаксическую структуру специфических данных программы (APP и COM) - опционально

Примечание - Более подробная информация о синтаксической структуре каждого заголовка приведена в стандарте ISO/IEC 10918-1 или CCITT Rec. T.81 Annex B Compressed data formats (Приложение B «Форматы сжатых данных»).

На Рисунке 3.2 «Уплотнение заголовка» продемонстрированы несколько синтаксических структур заголовка, сгенерированных главным компьютером и блоком JPU, соответственно.

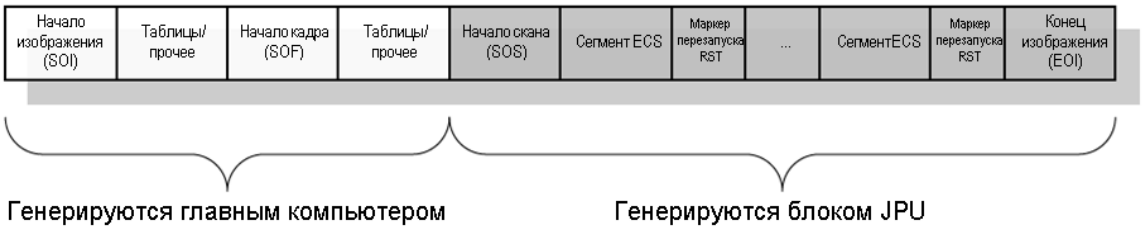


Рисунок 1.128 - Уплотнение заголовка

Полная последовательность операций кодера JPU представлена в Разделе С.1 «Кодер блока JPU».

1.3.6.2.3.2 Управление декодером

Декодирование JPEG – операция, обратная операции кодирования, представляющая собой процесс преобразования битового потока JPEG в изображения в формате YUV. Главный компьютер помогает блоку JPU осуществлять операции декодирования, производя поставку и

синтаксический анализ битового потока, задавая соответствующие значения в регистрах JPU, запуская блок JPU и извлекая YUV изображения, декодированные блоком JPU. Перед тем как продолжить главный компьютер должен извлечь из битового потока следующую информацию:

- Информацию об изображении, такую как ширина и высота изображения, а также формат цветного изображения
- Информацию о таблице, такую как таблицы квантования, таблицы Хаффмана и табличный индекс каждого цветового компонента
- Информацию о перезапуске для обеспечения устойчивости к ошибкам:
- Информацию о буфере битовых потоков, такую как указатель страницы, указатель слова и указатель считывания внешнего буфера битовых потоков, а также указатель считывания внутреннего буфера битовых потоков

После синтаксического анализа главный компьютер может задать данную информацию в регистрах блока JPU, и блок JPU может приступить к работе.

Примеры интерпретации заголовка, выявление указателей внешнего и внутреннего буферов битовых потоков приведена в исходной программе модели С предоставляемого пакета программного обеспечения и Приложении В «ГЕНЕРИРОВАНИЕ УКАЗАТЕЛЕЙ ЧТЕНИЯ БЛОКОВ BBC И GBU» настоящего документа.

1.3.6.2.3.2.1 Загрузка битового потока

Для декодирования битовых потоков блок JPU использует два уровня буферов битовых потоков: внешний буфер битовых потоков и внутренний буфер битовых потоков. Так как блок JPU не имеет доступа к информации об объеме и адресе загрузки битовых потоков, главный компьютер осуществляет загрузку определенного объема битовых потоков во внешний буфер битовых потоков и перемещает данный объем во внутренний буфер битовых потоков.

Данная операция осуществляется посредством задания регистров блока управления BBC перед началом декодирования. Далее каждый раз при опустошении внутреннего буфера битовых потоков производится автоматическое копирование битового потока из внешнего буфера между буферами BBC и GBU. Информация о соответствующих регистрах приведена в Таблице 2.66 «Регистр MJPEG_BBC_ENDADDR_REG (0x208)» настоящего документа и функции StreamCopy2Gram() исходного предоставляемого программного обеспечения.

1.3.6.2.3.2.2 Инициализация последовательности

Для осуществления данной операции главный компьютер должен задать следующую информацию для блока JPU.

Информация об изображении и минимальном кодированном блоке

Главный компьютер должен задать извлеченные параметры изображения, таблицы и перезапуска в регистрах блока JPU. Блок JPU содержит таблицу Хаффмана в качестве таблицы по умолчанию, рекомендуемую стандартом ISO/ IEC 10918-1. Если таблица Хаффмана, полученная в ходе интерпретации заголовка, совпадает с таблицей, содержащейся в блоке JPU, главный компьютер может пропустить процедуру загрузки таблицы. Блок JPU, использующий таблицу по умолчанию, может в два раза повысить производительность декодирования переменной длины по сравнению с пользовательской таблицей Хаффмана.

Информация по таблице Хаффмана приведена в Приложении А «ТАБЛИЦЫ ХАФФМАНА БЛОКА JPU» настоящего документа и в стандарте ISO/IEC 10918-1 или CCITT Rec. T.81. K.3 Typical Huffman tables for 8-bit precision luminance and chrominance («Типовые таблицы Хаффмана для сигналов яркости и цветности с точностью 8 бит»). Также можно обратиться к Таблице 2.2 «Регистр MJPEG_PIC_START_REG (0x000)» и функции DecSeqInit() исходного программного обеспечения для получения более подробной информации о настройках данного регистра.

Внешний и внутренний буферы битовых потоков

Далее главный компьютер должен назначить буфер битовых потоков для сохранения декодированных битовых потоков. Главный компьютер должен загрузить страницу, содержащую начало сегмента ECS (сегмент энтропийного кода), во внутренний буфер битовых потоков, даже если внешний буфер битовых потоков заполнен целым битовым потоком, включающим заголовок.

Соответствующие регистры перечислены в Разделе 2.2.2.2 «Регистры управления блоком GBU», Разделе 2.2.2.3 «Регистры управления блоком BBC» и функции DecSeqInit() исходного программного обеспечения. Информация по управлению буфером битовых потоков в данном исходном программном обеспечении приведена в Приложении D «УПРАВЛЕНИЕ БУФЕРОМ БИТОВЫХ ПОТОКОВ В ИСХОДНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ».

1.3.6.2.3.2.2.1 Функции буфера кадров

Главный компьютер, как правило, назначает буфер кадров для хранения декодированных YUV изображений.

Блок JPU может использовать один буфер кадров для сохранения одного декодированного JPEG изображения. Тем не менее, в приложениях реального времени блок JPU должен поддерживать декодирование ряда изображений и поэтому для обеспечения максимальной производительности рекомендуется использовать два или более буферов кадров. Перед запуском блока JPU необходимо загрузить в буфер кадров не менее одного исходного изображения. При использовании двух или более буферов кадров операция декодирования (запуск JPU) будет более эффективно производиться с одновременным использованием функции сохранения YUV изображений или функцией отображения YUV изображений.

1.3.6.2.3.2.3 Работа блока JPU в режиме декодера

1.3.6.2.3.2.3.1 Запуск декодера

Перед запуском декодера главный компьютер должен задать базовый адрес буфера кадров каждого цветового компонента и шаг по индексу в соответствующих регистрах. После задания данной информации в регистрах главный компьютер может запустить декодер через регистр MJPEG_PIC_START_REG.

1.3.6.2.3.2.3.2 Заполнение буфера битовых потоков

В ходе процесса декодирования блок JPU предотвращает перемещение битового потока из внутреннего буфера битовых потоков. Если внутренний буфер битовых потоков пуст, блок GBU отправляет блоку BBC сигнал прерывания, и блок BBC загружает страницу битового потока из внешнего буфера битовых потоков.

Тем не менее, все битовые потоки, ранее загруженные главным компьютером во внешний буфер битовых потоков, полностью используются. Блок JPU выдает сигнал прерывания и переходит в режим ожидания до заполнения внешнего буфера битовых потоков. Затем главный компьютер осуществляет проверку сигнала прерывания и заполняет буфер.

1.3.6.2.3.2.3.3 Прекращение декодирования

Операция декодирования изображения занимает определенное количество времени, и главный компьютер в процессе ожидания завершения операции декодирования изображения может перейти к выполнению других задач.

После завершения процесса декодирования блок JPU может информировать главный компьютер о завершении декодирования сигналом прерывания или через регистр MJPEG_PIC_STATUS_REG. Затем главный компьютер может сбросить сигнал прерывания по завершении декодирования через регистр MJPEG_PIC_STATUS_REG.

1.3.6.2.3.2.4 Сохранение или отображение YUV изображения

Декодированное YUV изображение будет сохранено или отображено по запросу главного компьютера.

Рисунок 1.129 «Схема последовательности операций декодера» демонстрирует последовательность операций декодера RTL-модели.

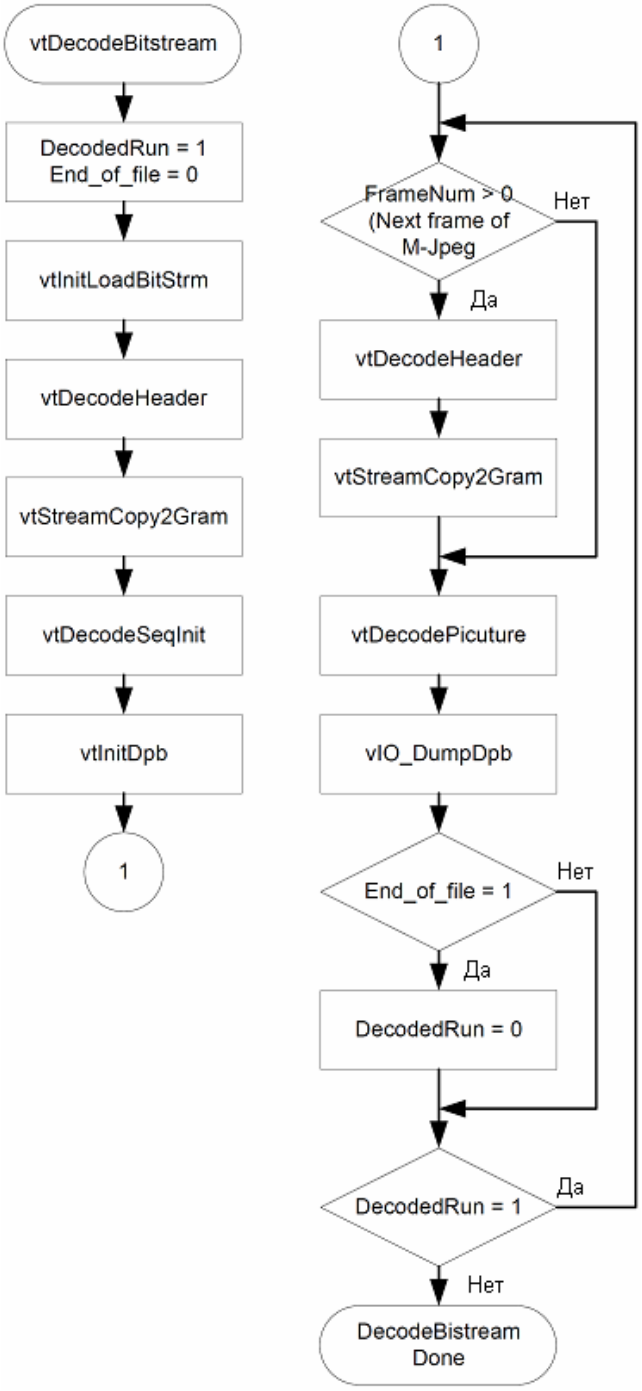


Рисунок 1.129 - Схема последовательности операций декодера

Примечание - Схема последовательности операций декодирования исходного программного обеспечения также приведена в Разделе С.2 «Декодер блока JPU».

1.3.6.2.4 ТАБЛИЦЫ ХАФФМАНА БЛОКА JPU

На этапе кодирования/декодирования слов переменной длины (VLC/VLD) таблица Хаффмана используется для кодирования квантованных коэффициентов или декодирования битовых потоков. Блок JPU поддерживает кодированное с переменной длиной слова Хаффмана по умолчанию или пользователя. При кодировании блоком JPU коэффициентов с использованием пользовательской таблицы Хаффмана она загружается в табличную память. И при кодировании с использованием таблицы Хаффмана по умолчанию данная таблица также должна быть размещена в табличной памяти. Нижеприведенная таблица демонстрирует таблицу Хаффмана по умолчанию.

Таблица 1.241 - Таблицы Хаффмана по умолчанию

Таблица Хаффмана по умолчанию							
Luma DC BitLength							
0x00	0x01	0x05	0x01	0x01	0x01	0x01	0x01
0x01	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Luma DC HuffValue							
0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x08	0x09	0x0A	0x0B				
Luma AC BitLength							
0x00	0x02	0x01	0x03	0x03	0x02	0x04	0x03
0x05	0x05	0x04	0x04	0x00	0x00	0x01	0x7D
Luma AC HuffValue							
0x01	0x02	0x03	0x00	0x04	0x11	0x05	0x12
0x21	0x31	0x41	0x06	0x13	0x51	0x61	0x07
0x22	0x71	0x14	0x32	0x81	0x91	0xA1	0x08
0x23	0x42	0xB1	0xC1	0x15	0x52	0xD1	0xF0
0x24	0x33	0x62	0x72	0x82	0x09	0x0A	0x16
0x17	0x18	0x19	0x1A	0x25	0x26	0x27	0x28
0x29	0x2A	0x34	0x35	0x36	0x37	0x38	0x39
0x3A	0x43	0x44	0x45	0x46	0x47	0x48	0x49
0x4A	0x53	0x54	0x55	0x56	0x57	0x58	0x59
0x5A	0x63	0x64	0x65	0x66	0x67	0x68	0x69
0x6A	0x73	0x74	0x75	0x76	0x77	0x78	0x79
0x7A	0x83	0x84	0x85	0x86	0x87	0x88	0x89
0x8A	0x92	0x93	0x94	0x95	0x96	0x97	0x98
0x99	0x9A	0xA2	0xA3	0xA4	0xA5	0xA6	0xA7
0xA8	0xA9	0xAA	0xB2	0xB3	0xB4	0xB5	0xB6
0xB7	0xB8	0xB9	0xBA	0xC2	0xC3	0xC4	0xC5
0xC6	0xC7	0xC8	0xC9	0xCA	0xD2	0xD3	0xD4
0xD5	0xD6	0xD7	0xD8	0xD9	0xDA	0xE1	0xE2
0xE3	0xE4	0xE5	0xE6	0xE7	0xE8	0xE9	0xEA
0xF1	0xF2	0xF3	0xF4	0xF5	0xF6	0xF7	0xF8
0xF9	0xFA						
Chroma DC BitLength							
0x00	0x03	0x01	0x01	0x01	0x01	0x01	0x01
0x01	0x01	0x01	0x00	0x00	0x00	0x00	0x00
Chroma DC HuffValue							
0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07

Продолжение таблицы 1.241

Таблица Хаффмана по умолчанию							
0x08	0x09	0x0A	0x0B				
Chroma AC BitLength							
0x00	0x02	0x01	0x02	0x04	0x04	0x03	0x04
0x07	0x05	0x04	0x04	0x00	0x01	0x02	0x77
Chroma AC HuffValue							
0x00	0x01	0x02	0x03	0x11	0x04	0x05	0x21
0x31	0x06	0x12	0x41	0x51	0x07	0x61	0x71
0x13	0x22	0x32	0x81	0x08	0x14	0x42	0x91
0xA1	0xB1	0xC1	0x09	0x23	0x33	0x52	0xF0
0x15	0x62	0x72	0xD1	0x0A	0x16	0x24	0x34
0xE1	0x25	0xF1	0x17	0x18	0x19	0x1A	0x26
0x27	0x28	0x29	0x2A	0x35	0x36	0x37	0x38
0x39	0x3A	0x43	0x44	0x45	0x46	0x47	0x48
0x49	0x4A	0x53	0x54	0x55	0x56	0x57	0x58
0x59	0x5A	0x63	0x64	0x65	0x66	0x67	0x68
0x69	0x6A	0x73	0x74	0x75	0x76	0x77	0x78
0x79	0x7A	0x82	0x83	0x84	0x85	0x86	0x87
0x88	0x89	0x8A	0x92	0x93	0x94	0x95	0x96
0x97	0x98	0x99	0x9A	0xA2	0xA3	0xA4	0xA5
0xA6	0xA7	0xA8	0xA9	0xAA	0xB2	0xB3	0xB4
0xB5	0xB6	0xB7	0xB8	0xB9	0xBA	0xC2	0xC3
0xC4	0xC5	0xC6	0xC7	0xC8	0xC9	0xCA	0xD2
0xD3	0xD4	0xD5	0xD6	0xD7	0xD8	0xD9	0xDA
0xE2	0xE3	0xE4	0xE5	0xE6	0xE7	0xE8	0xE9
0xEA	0xF2	0xF3	0xF4	0xF5	0xF6	0xF7	0xF8
0xF9	0xFA						

Таблица Хаффмана состоит из таблиц яркости и цветности. Таблица яркости включает таблицы коэффициентов DC и AC. Таблица коэффициентов DC содержит 16 значений длины в битах (BitLength) и 12 значений Хаффмана (HuffValue). Таблица коэффициентов AC содержит 16 значений длины в битах и 162 значения Хаффмана.

В режиме декодера, если поток данных JPEG кодируется с использованием пользовательской таблицы Хаффмана, главный процессор должен загрузить таблицы во внутреннюю память для декодирования с использованием дерева Хаффмана в заданном формате перед началом процесса декодирования. Рекомендуется обратиться к Рисунку A.1 «Таблица Хаффмана для декодера JPEG» и Разделу A.1 «Генерирование таблиц Хаффмана для декодера».

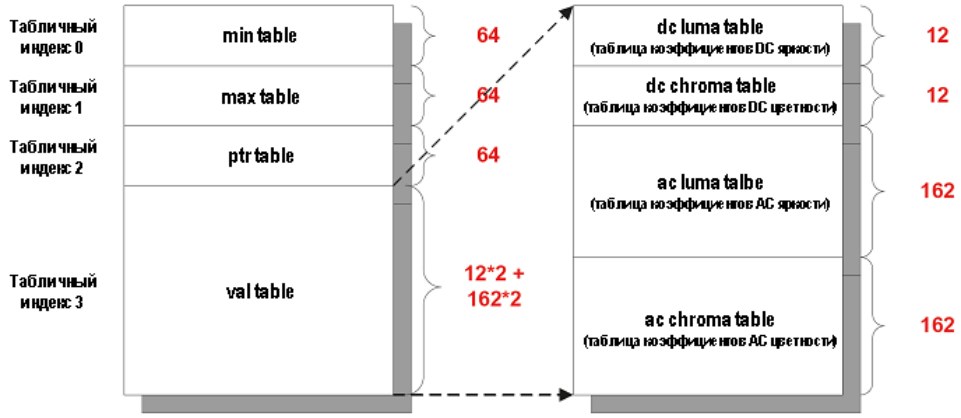


Рисунок 1.130 - Таблица Хаффмана для декодера JPEG

Если при декодировании используется таблица Хаффмана по умолчанию, декодер блока JPU использует встроенную таблицу Хаффмана.

В режиме кодера главный процессор генерирует код Хаффмана и таблицы размеров и загружает таблицы во внутреннюю память для кодирования по алгоритму Хаффмана перед запуском процесса кодирования. Образец кода приведен в Разделе A.2 «Генерирование таблиц Хаффмана для кодера».

1.3.6.2.4.1 Генерирование таблиц Хаффмана для декодера

Далее приведен пример генерирования таблицы Хаффмана для декодера блока JPU с использованием таблиц BitLength и HuffValue.

```
// To generate Huffman tables.

for (i=0; i<16; i++) {
if (huffBits[i]) {              // if there is bit cnt value
huffPtr[i] = ptrCnt;
ptrCnt += huffBits[i];
huffMin[i] = huffCode;
huffMax[i] = huffCode + (huffBits[i] - 1);
dataFlag = 1;
zeroFlag = 0;
} else {
huffPtr[i] = 0xFF;
huffMin[i] = 0xFFFF;
huffMax[i] = 0xFFFF;
zeroFlag = 1;
}

if (dataFlag == 1) {
if (zeroFlag == 1)
huffCode <<= 1;
else
huffCode = (huffMax[i] + 1) << 1;
}
}
```

1.3.6.2.4.2 Генерирование таблиц Хаффмана для кодера

Далее приведен пример генерирования таблицы Хаффмана для кодера блока JPU.

```
// To generate Huffman tables.
maxsymbol = tabNum & 1 ? 256 : 16;
/* Figure C.1: make table of Huffman code length for each symbol */
p = 0;
for (l=1; l<=16; l++) {
i = bits[l-1];
if (i < 0 || p + i > maxsymbol)
return 0;
while (i--)
huffsize[p++] = l;
}

lastp = p;

/* Figure C.2: generate the codes themselves */
/* We also validate that the counts represent a legal Huffman code tree. */
code = 0;
si = huffsize[0];
p = 0;
while (p < lastp) {
while (huffsize[p] == si) {
huffcode[p++] = code;
code++;
}
if (code >= (1 << si))
return 0;
code <<= 1;
si++;
}

/* Figure C.3: generate encoding tables */
/* These are code and size indexed by symbol value */
memset(ehufsi, 0, sizeof(int) * 256);
memset(ehufco, 0, sizeof(int) * 256);

for (p=0; p<lastp; p++) {
i = huffval[p];
if (i < 0 || i >= maxsymbol || ehufsi[i])
return 0;
ehufco[i] = huffcode[p];
ehufsi[i] = huffsize[p];
}
```

1.3.6.2.5 ГЕНЕРИРОВАНИЕ УКАЗАТЕЛЕЙ ЧТЕНИЯ БЛОКОВ BBC И GBU

Далее приведен пример генерирования точных указателей чтения блоков BBC и GBU в блоке JPU.

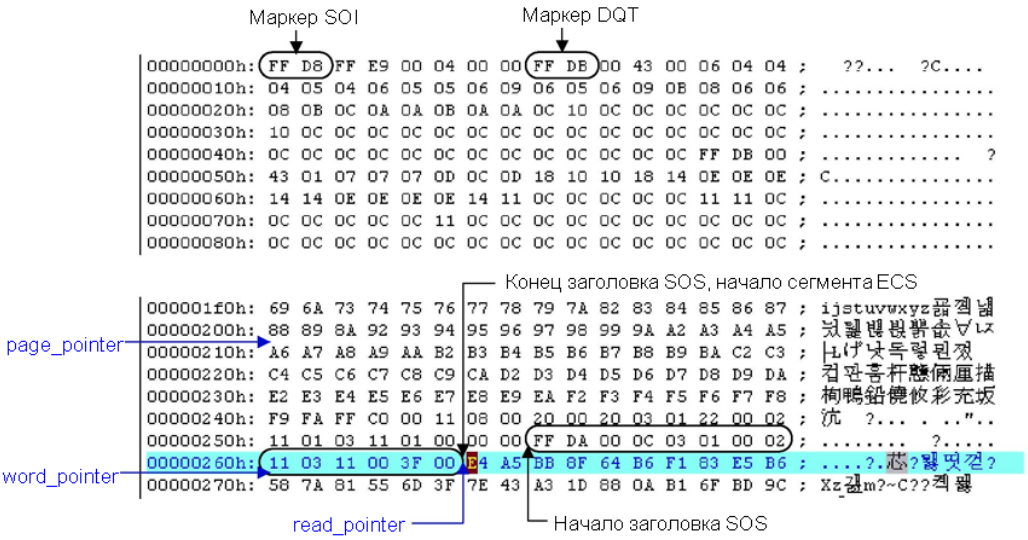
```
// ptr = Total read byte count during header parsing before ECS (Entropy-Coded Segment)

// 265 page pointer for BBC
page_ptr = ptr / 256;       // page unit (256 byte)

// word pointer for GBU
word_ptr = (ptr % 256) / 4;         // word unit (4 byte)
if (page_ptr & 1)
word_ptr += 64;                     // offset of second bitstream buffer of GBU
if (word_ptr & 1)
word_ptr -= 1;              // to make even.

// read pointer for GBU
read_ptr = (ptr % 4) * 8;           // bit unit
if (((ptr % 256) / 4) & 1)
read_ptr += 32;
```

Нижеприведенный рисунок демонстрирует указатели страницы, слова и бита примерного битового потока.




```
ptr = 0x266;     //Start of ECS (entropy-coded segment (начало сегмента ECS (сегмент энтропийного кода)
page_ptr = 0x266 / 0x100 = 2;
word_ptr = (0x66 / 4) – 1 = 0x18;
read_ptr = 0x6 * 8 = 0x30;
```

Рисунок 1.131 - Пример указателей

1.3.6.2.6 ПСЕВДОКОД ИСХОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.3.6.2.6.1 Кодер блока JPU

```
EncStreaminit {
open stream
}
EncSeqInit {
set BBC registers          // init stream cnt, bbc pointer, bbc interrupt address
set GBU registers          // init gram write pointer and interrupt address
set MAIN registers         // init picture info, mcu info, op info
JpgEncLoadHuffTab {
generate Huff table
set MAIN (huff) registers          // upload huffman table
}
JpgEncLoadQMatTab {
generate Qmat table
set MAIN (qmat) registers          // upload quantization table
}
}
FramebufferInit {
allocate frame buffer
}
LoadimageYuvBurst {
write to SDRAM             // load image will be encoding to the frame buffer
}
EncPicRun {
set MAIN registers         // init frame buffer address, run JPU
wait interrupt
saveBitBuffer {// if stream buffer full
read BBC registers         // check stream buffer write pointer
read from SDRAM
EncodeHeader  // put huffman table, quantization table
        // put picture info, mcu info,
write to stream
set BBC registers          // update stream buffer read pointer
}
set BBC registers          // after saveBitBuffer, init bbc pointer
        // init bbc interrupt address
}
EncSeqEnd {
saveBitBuffer {// flush the remained stream
read BBC registers         // check stream buffer write pointer
read from SDRAM
EncodeHeader  // put huffman table, quantization table
        // put picture info, mcu info,
write to stream
set BBC registers          // update stream buffer read pointer
}
set BBC registers          // clear stream buffer flush command
close stream
}
```

1.3.6.2.6.2 Декодер блока JPU

```
DecStreamInit {
open stream
set BBC registers          // init stream buffer read/write pointer
loadBitBuffer {// fill the stream buffer
read from stream
write to SDRAM
set BBC registers          // update stream buffer write pointer
}
}

DecodeHeader {
parsing(read) Header info          // get huffman table, quantization table
        // get picture info, mcu info, gbu ff read pointer
}
StreamCopy2Gram {
set BBC registers          // set stream cnt, gram address and run bbc
set BBC registers          // set stream cnt, gram address and run bbc
}

DecSeqInit {
set GBU registers          // init gram write pointer and interrupt address
set BBC registers          // init bbc interrupt address
set MAIN registers         // init picture info, mcu info, op info, frame buffer info
JpgDecHuffTabSetup {
generate(read) Huff table
set MAIN (huff) registers          // upload huffman table
}
JpgDecQMatTabSetup {
generate(read) Qmat table
set MAIN (qmat) registers          // upload quantization table
}
}

FrameBufferInit {
allocate frame buffer      // generate frame buffer address
}

DecPicRun {
set MAIN registers         // init dpcm register, set gbu ff read pointer
        // init frame buffer address, run JPU
wait interrupt
loadBitBuffer {// if stream buffer empty
read from stream
write to SDRAM
set BBC registers          // update stream buffer write pointer
}
}

SaveImageYuvBurst {
read from SDRAM        // save decoded image from the frame buffer
}

DecSeqEnd {
close stream
}
```

1.3.6.2.7 УПРАВЛЕНИЕ БУФЕРОМ БИТОВЫХ ПОТОКОВ В ИСХОДНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

Рисунок 1.132 отображает схему управления буфером битовых потоков в режиме декодирования.

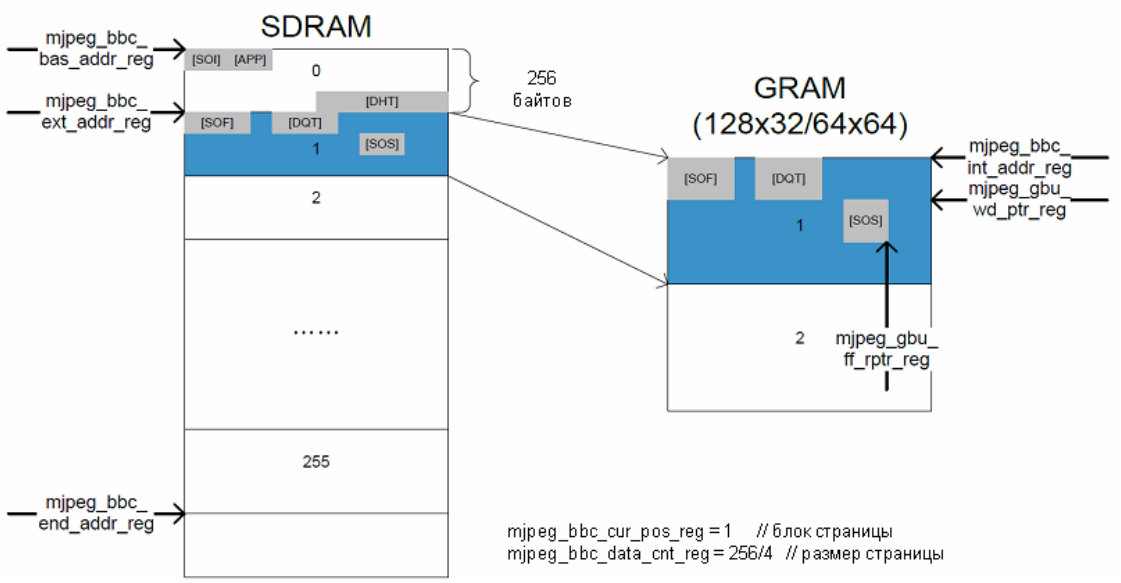


Рисунок 1.132 - Примерная схема управления буфером битовых потоков

Блок BBC (Bitstream Buffer Control – блок управления буфером битовых потоков) осуществляет передачу битовых потоков между внешним буфером битовых потоков на SDRAM и внутренним буфером битовых потоков (буфером GRAM).

В данной модели тестирования буфер битовых потоков находится в памяти SDRAM (от 0x0000_0000 до 0x0001_0000). Буфер GRAM (dpsram128x32/64x64) является встроенной памятью блока GBU, обеспечивающей хранение битовых потоков, размер которой составляет 512 байтов. Тестовая модель использует двойную буферизацию, т.е. в каждый буфер поступает одновременно по 256 байтов. Это необходимо для предотвращения «зависания» аппаратного обеспечения блока JPU в момент перемещения битового потока блоком BBC (блок управления буфером битовых потоков) в буфер GRAM по возникновению события опустошения буфера.

Страница может быть определена как страница размером 256 байтов (64 x 4 байта) потока в качестве единицы порции обмена битовыми потоками между блоком BBC и буфером GRAM. В соответствии с определением данной страницы, размер буфера битовых потоков 0x0001_0000 позволяет содержать 0x100 (256 байтов) страниц. На этапе инициализации операции декодирования главный компьютер может осуществлять копирование 256 байтов потока в буфер GRAM посредством функции StreamCopy2Gram два раза подряд. Поле помещения в буфер GRAM первых 256 байтов потока блоки GBU и BBC могут автоматически отправить сигналы на заполнение буфера GRAM.

При этом блок BBC осуществляет доставку битового потока коэффициентов, начиная с сегмента ECS (сегмент энтропийного кода) не из заголовка битового потока. Так как аппаратное обеспечение JPEG не поддерживает интерпретацию заголовка, данная функция возлагается на главный компьютер. Таким образом, главный компьютер должен передавать в блок BBC фактический адрес начала. Главный компьютер сначала должен задать адрес начала буфера битового потока в регистрах mjpeg_bbc_bas_addr_reg и mjpeg_bbc_end_addr_reg. Также главный компьютер задает адрес начала страницы, содержащей сегмент ECS, в регистре mjpeg_bbc_ext_addr_reg, после чего блок BBC может копировать страницу из данного адреса в буфер GRAM.

Но в большинстве случаев передача битового потока в буфер GRAM не может начинаться с сегмента ECS, и программному обеспечению необходимо записать точное местоположение начала сегмента ECS в регистре-указателе слова/чтения MJPEG_GBU_WD_PTR_REG/ MJPEG_GBU_FF_RPTR_REG.

По завершении задания начальных установок декодер JPEG может считывать каждые 16 битов битового потока из буфера GRAM и осуществлять декодирование, сопоставляя данные с таблицей Хаффмана.

Примечание - Подробное описание упомянутых регистров приведено в Разделе 2.2.2, «Подробное описание регистров».

1.3.6.2.8 УПАКОВАННЫЙ ДЕСЯТИЧНЫЙ ФОРМАТ

Блок JPU поддерживает упакованный десятичный формат для форматов цветного изображения 444 и 422. Рисунок 1.133 «Упакованный десятичный формат 444» демонстрирует считывание или запись каждого цветового компонента в двумерном и упакованном десятичном форматах.

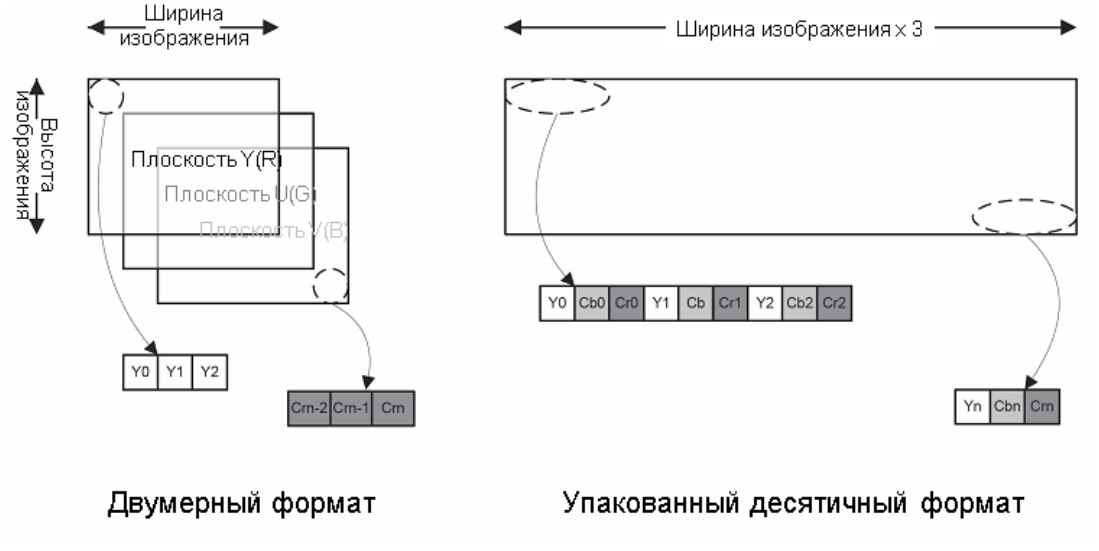


Рисунок 1.133 - Упакованный десятичный формат 444

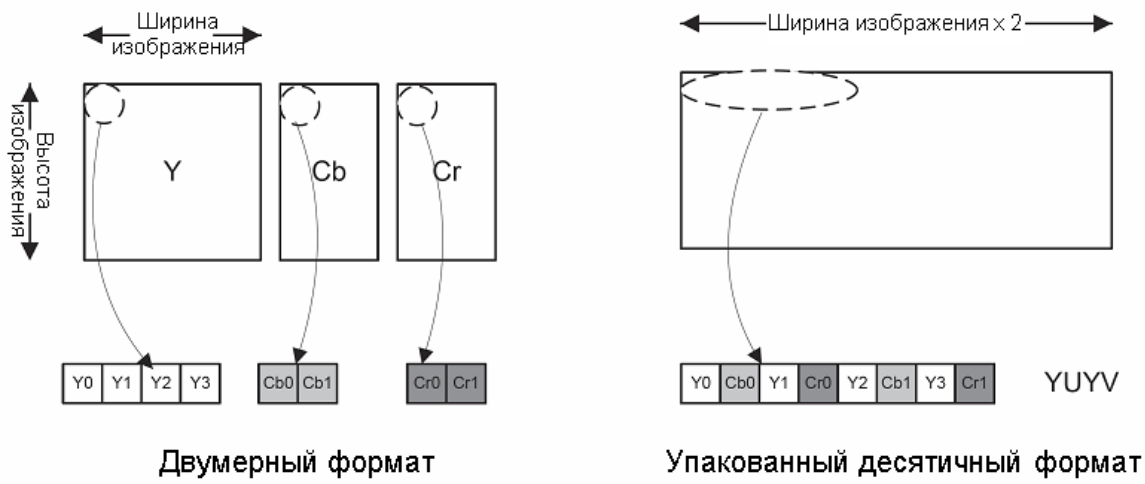


Рисунок 1.134 - Упакованный десятичный формат 422

Упакованный десятичный код использует одну плоскость, поддерживающую пару данных Y/Cb/Cr (8 битов каждая, всего 24 бита), в то время как двумерный формат использует три различные плоскости для каждого из цветовых компонентов Y, Cb и Cr. Главный компьютер получает данные о данных форматах – входной формат исходного изображения для работы кодека и выходной формат декодированного изображения для отображения.

1.3.6.2.9 КОЭФФИЦИЕНТЫ ДИСКРЕТИЗАЦИИ СТАНДАРТА JPEG

Данная глава приводит описание форматов цветных изображений для кодирования и декодирования JPEG, поддерживаемых блоком JPU (JPEG Processing Unit – блок обработки изображений в формате JPEG) фирмы Chips&Media. Также она кратко описывает некоторые спецификации JPEG относительно коэффициентов дискретизации, синтаксических структур заголовка кадра и скана, что позволит пользователю лучше понять работу системы.

Спецификация JPEG, ISO/IEC 10918-1 (CCITT Rec.T.81), содержит подробную информацию о коэффициентах дискретизации и количестве компонентов. Следующие подразделы содержат выдержки из данной Спецификации.

1.3.6.2.9.1 Математические определения

Исходное изображение содержит компоненты Nf. Каждый компонент с уникальным идентификатором Ci содержит прямоугольную сетку дискретных значений, состоящую из столбцов xi и строк yi. Размеры компонентов определяются двумя параметрами, X и Y, где X – это максимум значений xi, а Y – максимум значений yi для всех компонентов в кадре. Для каждого компонента определены коэффициенты дискретизации Hi и Vi, определяющие отношения между размерами xi и yi и максимальными размерами X и Y, в соответствии с формулой:

$$x_i = \left[X \times \frac{H_i}{H_{\max}} \right] u \; y_i \left[Y \times \frac{V_i}{V_{\max}} \right],$$

где Hmax и Vmax – максимальные коэффициенты дискретизации для всех компонентов в кадре, а e u – функция CEILING.

Примечание - Параметры X, Y, Hi и Vi содержатся в заголовке кадра сжатого изображения (см. Раздел F.2 «Синтаксическая структура заголовка кадра»), в то время как размеры отдельных компонентов xi и yi выводятся декодером. Исходные изображения с размерами xi и yi, не удовлетворяющими вышеприведенным выражениям, не могут быть надлежащим образом восстановлены.

Совет - Более подробная информация приведена в разделе A.1.1 «Размеры и коэффициенты дискретизации» Спецификации JPEG.

1.3.6.2.9.2 Синтаксическая структура заголовка кадра

Спецификация JPEG определяет параметры компонентов как синтаксические структуры заголовков кадров.

Nf (Число компонентов изображения в кадре)

Определяет количество компонентов исходного изображения в кадре. Значение Nf равно количеству наборов параметров компонентов кадра (Ci, Hi, Vi и Tqi), содержащихся в заголовке кадра.

Примечание - В реализации Baseline значение Nf составляет 1-255.

Ci (Идентификатор компонента)

Присваивает уникальную метку i-тому компоненту в последовательности параметров компонентов кадра. Данные значения используются в заголовках сканов для идентификации компонентов в скане. Значение Ci отличается от значений с C1 по Ci - 1.

Hi (Коэффициент дискретизации по горизонтали)

Определяет связь между размером компонента по горизонтали и максимальным размером X изображения (см. A.1.1), а также количество единиц данных по горизонтали компонента Ci в каждом минимальном кодированном блоке, когда скан содержит более одного кодированного компонента.

Vi (Коэффициент дискретизации по вертикали)

Определяет связь между размером компонента по вертикали и максимальным размером Y изображения (см. A.1.1), а также количество единиц данных по вертикали компонента Ci в

каждом минимальном кодированном блоке, когда скан содержит более одного кодированного компонента.

Примечание - Более подробная информация приведена в разделе B.2.2 «Синтаксическая структура заголовка кадра» Спецификации JPEG.

1.3.6.2.9.3 Синтаксическая структура заголовка скана

Спецификация JPEG приводит толкование ограничений по количеству компонентов и коэффициенту дискретизации синтаксической структуры заголовка скана в скане.

Ns (Число компонентов изображения в скане)

Определяет количество компонентов исходного изображения в скане. Значение Ns равно количеству наборов параметров компонентов скана (Csj, Tdj и Taj), содержащихся в заголовке скана.

Примечание - Значение Ns составляет 1-4 в любой реализации.

Csj (Селектор компонентов скана)

Выбирает, какие из компонентов изображения Nf, указанные в параметрах кадра, будут i-тыми компонентами в скане. Каждое значение Csj совпадает с одним из значений Ci, указанных в заголовке кадра, и порядок их расположения в заголовке скана соответствует порядку расположения в заголовке кадра. Если Ns > 1, порядок чередующихся компонентов в минимальном кодированном блоке будет следующим: сначала Cs1, затем Cs2 и т.д. Если Ns > 1, компоненты изображения, содержащиеся в скане, должны соответствовать неравенству:

$$\sum_{j=1}^{N_s} H_j \times V_j \leq 10,$$

где Hj и Vj – коэффициенты дискретизации по горизонтали и вертикали для компонента скана j. Данные коэффициенты дискретизации определены в заголовке кадра для компонента i, где i – индекс спецификации компонента, для которого идентификатор компонентов кадра Ci соответствует селектору компонентов скана Csj.

1.3.6.2.9.4 Компоненты и коэффициенты дискретизации блока JPU

Блок JPU (JPEG Processing Unit – блок обработки изображений в формате JPEG) поддерживает только несколько более широко применимых комбинаций коэффициентов дискретизации, так как на самом деле существует множество комбинаций количества компонентов и коэффициентов дискретизации по горизонтали и вертикали.

Изначально блок JPU был разработан на базе формата цветного изображения 4:2:0, так как блок VPU поддерживает большинство видеостандартов, таких как MPEG-2, MPEG-4 и H.264. Формат 4:2:0 содержит в общем количестве шесть блоков 8x8 - четыре блока для компонента Y, один блок для компонента Cb и один блок для компонента Cr. Позже возможности блока JPU были расширены до возможности поддержки других форматов с большим количеством компонентов и коэффициентами дискретизации в пределах шести блоков 8x8, что обеспечивает поддержку собственных функций.

Данные форматы цветных изображений описаны тремя компонентами (Y, Cb и Cr), и каждый из компонентов имеет два компонента дискретизации – по горизонтали и по вертикали.

1.3.6.2.9.4.1 Формат 4:2:0

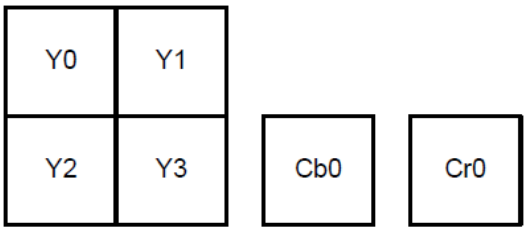


Рисунок 1.135 - Формат 4:2:0

Таблица 1.242 – Формат 4:2:0

	компонент 1 (Y)	компонент 2 (Cb)	компонент 3 (Cr)
Коэффициент дискретизации по горизонтали	2	1	1
Коэффициент дискретизации по вертикали	2	1	1

1.3.6.2.9.4.2 Формат 4:2:2 (4:2:2h)

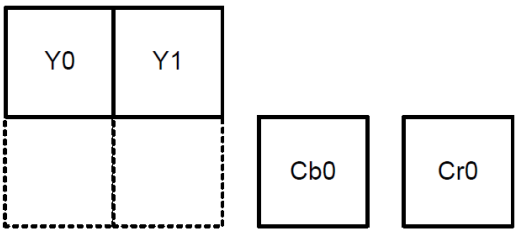


Рисунок 1.136 - Формат 4:2:2h

Таблица 1.243 – Формат 4:2:2

	компонент 1 (Y)	компонент 2 (Cb)	компонент 3 (Cr)
Коэффициент дискретизации по горизонтали	2	1	1
Коэффициент дискретизации по вертикали	1	1	1

1.3.6.2.9.4.3 F.4.3. Формат 2:2:4 (4:2:2v)

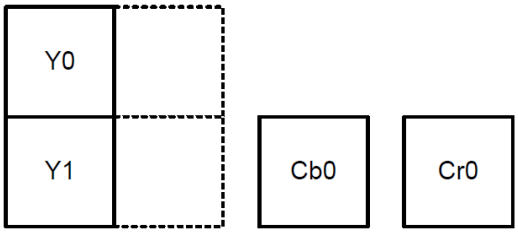


Рисунок 1.137 - Формат 4:2:2v

Таблица 1.244 – Формат 2:2:4

	компонент 1 (Y)	компонент 2 (Cb)	компонент 3 (Cr)
Коэффициент дискретизации по горизонтали	1	1	1
Коэффициент дискретизации по вертикали	2	1	1

1.3.6.2.9.4.4 Формат 4:4:4

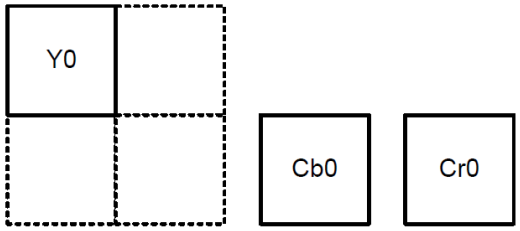


Рисунок 1.138 - Формат 4:4:4

Таблица 1.245 – Формат 4:4:4

	компонент 1 (Y)	компонент 2 (Cb)	компонент 3 (Cr)
Коэффициент дискретизации по горизонтали	1	1	1
Коэффициент дискретизации по вертикали	1	1	1

1.3.6.2.9.4.5 Формат 4:0:0

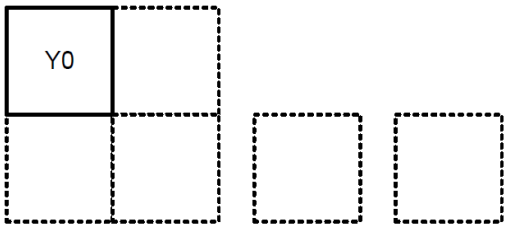


Рисунок 1.139 - Формат 4:0:0

Таблица 1.246 – Формат 4:0:0

	компонент 1 (Y)	компонент 2 (Cb)	компонент 3 (Cr)
Коэффициент дискретизации по горизонтали	1	0	0
Коэффициент дискретизации по вертикали	1	0	0

1.3.6.3 Кодер JPEG (описание)

1.3.6.3.1 Общий обзор

JPU (JPEG Processing Unit – блок обработки изображений в формате JPEG) – это высокопроизводительный кодек JPEG для IP-камер, способный осуществлять кодирование и декодирование изображений в базовом (Baseline) и движущемся (Motion) форматах JPEG. Блок JPU поддерживает скорость кодирования цветного изображения формата 4:2:2 до 155 мегапикселей/сек и скорость декодирования цветного изображения формата 4:4:4 до 100 мегапикселей/сек.

Блок JPU соединяется с системой по 32-битной шине AMBA 3 APB для управления системой и по 64-битной шине AMBA 3 AXI для обеспечения пропускной способности канала передачи данных, а наличие внутрикристальной памяти обеспечивает высокую производительность.

1.3.6.3.1.1 Характеристики

Основные характеристики блока JPU находятся в соответствии с основными характеристиками стандарта JPEG Baseline. В режиме кодирования и декодирования поддерживается обработка изображений размером до 32768x32768. Блок JPU легко встраивается в систему благодаря комбинации стандартного и простого интерфейсов AXI/APB.

Таблица 1.247 приводится перечень характеристик блока JPU.

Таблица 1.247 – Характеристики кодека MJPEG

Стандарт	Профиль	Разрешение
Кодер JPEG	Baseline (базовый)	32768x32768
Декодер JPEG	Baseline (базовый)	32768x32768

Кодер и декодер MJPEG в реализации Baseline

- Соответствие стандарту JPEG ISO/IEC 10918-1
- Поддержка от 1 до 3 цветовых компонентов
- 3 компонента в скане (только чересстрочный скан)
- 8-битные выборки для каждого компонента
- Поддержка форматов цветного изображения 4:2:0, 4:2:2, 4:4:0, 4:4:4 и 4:0:0 (максимум 8x8 блоков в одном минимальном кодированном блоке (MCU))
- Минимальный размер кодируемых данных – 16x16 пикселей
- Поддержка зоны ROI (зона интереса – только декодер)
- Поддержка упакованного десятичного кода для форматов цветного изображения 4:4:4 и 4:2:0

Дополнительные положительные характеристики

- Режим частичного кодирования
- Поворот/зеркальное отображение в режиме кодера
- Режим частичного кодирования
- Поворот/зеркальное отображение в режиме декодера
- Субдискретизация в режиме декодера

Эксплуатационные характеристики и скорость передачи данных

В таблице 1.248 приводятся эксплуатационные характеристики кодека JPEG, обеспечивающий размер битового потока до 1920x1088 в программной среде Chips&Media.

Таблица 1.248 – Эксплуатационные характеристики кодера и декодера блока JPU

Кодек блока JPU	4:0:0	4:2:0	4:2:2	4:4:4	Тактовая частота
Кодер	170 мегапикселей/сек	220 мегапикселей/сек	155 мегапикселей/сек	100 мегапикселей/сек	200 МГц
Декодер	180 мегапикселей/сек	230 мегапикселей/сек	160 мегапикселей/сек	100 мегапикселей/сек	

Простое интегрирование

- 32-битный интерфейс AMBA APB (с функцией PREADY) для связи с главным процессором
- 64-битный интерфейс AMBA AXI для доступа к внешней памяти

Прерывание

- Генерация и подача сигналов прерывания на главный процессор или контроллер прерываний

Область применения

- Пропускная способность логических вентилей около 164 КБ для частоты ядра процессора 240 МГц, разработанного согласно технологии TSMC 65 нм
- Встроенная память около 6,1 КБ

1.3.6.3.1.2 Блок-схема

Блок JPU включает главное устройство управления и основной модуль кодека JPEG. Главное устройство управления обеспечивает связь с кодеком JPEG. Управление главным компьютером обеспечивается через шину AMBA APB, а загрузка данных битового потока и YUV изображения осуществляется через шину AMBA AXI.

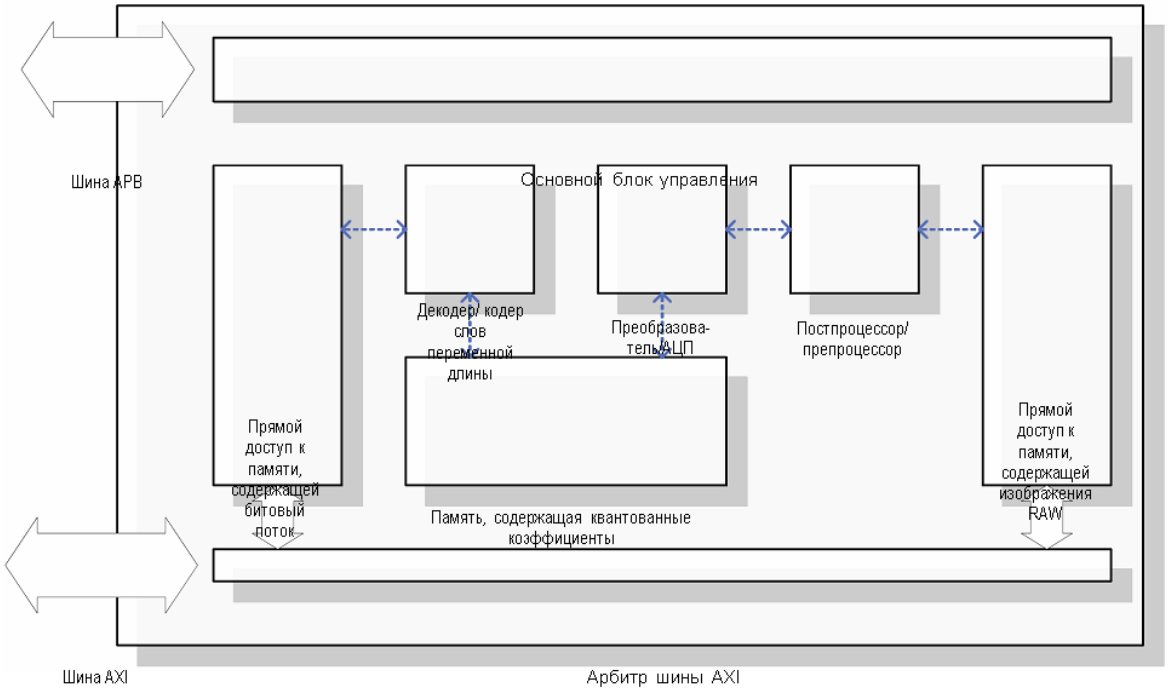


Рисунок 1.140 - Блок-схема работы модуля JPU

Двусторонними стрелками представлены основные интерфейсы между блоком JPU и внешней системой. Интерфейс шины APB предназначен для управления, интерфейс шины AXI предназначен для доступа к SDRAM.

Внутренняя шина AXI сходна с внешней шиной AXI, за исключением того, что блок JPU предусматривает два контроллера шины AXI. Таким образом, блок JPU оснащен встроенным модулем арбитра, обеспечивающим обработку запросов от внутренних контроллеров прямого доступа к памяти, и только один канал записи и чтения шины AXI представлен в перечне портов блока JPU для обеспечения простой интеграции блока JPU в пользовательскую систему на кристалле.

Главное устройство управления распределяет обязанности функциональных блоков JPU. Подробное описание характеристик, структуры и функций подблоков приведено в следующих разделах.

1.3.6.3.2 ПОРЯДОК СЛЕДОВАНИЯ БАЙТОВ

Блок JPU использует 8-байтовый доступ через интерфейс шины AXI и поддерживает 4 порядка следования байтов при передаче данных через интерфейс. Главный процессор может сконфигурировать порядок следования байтов с помощью функции API, предоставляемой компанией C&M.

1.3.6.3.2.1 Операция записи/считывания

Например, когда блок JPU записывает «0x7766554433221100» по адресу «A», расположение данных с данным порядком следования байтов будет следующим:

ENDIAN_MODE = 2'b00							
63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
00	11	22	33	44	55	66	77
ENDIAN_MODE = 2'b01							
63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
77	66	55	44	33	22	11	00
ENDIAN_MODE = 2'b10							
63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
44	55	66	77	00	11	22	33
ENDIAN_MODE = 2'b11							
63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
33	22	11	00	77	66	55	44

1.3.6.3.2.2 Обмен данными с главным процессором

Блок JPU обращается к внешней памяти для выполнения следующих операций:

Доступ к буферу кадров

Порядок следования байтов для буфера кадров может быть сконфигурирован отдельно. Подробная информация приведена в описании связи между порядком следования байтов и расположением пикселей для различных форматов буферов кадров.

Доступ к буферу битовых потоков

Порядок следования байтов для буфера кадров может быть сконфигурирован отдельно.

1.3.6.3.3 ТРЕБОВАНИЯ К ИНТЕГРАЦИИ

Данная глава содержит некоторую необходимую информацию по интеграции блока JPU в систему пользователя. Существует множество параметров памяти, таких как буферы кадров, их размеры и разметка изображений в буферах кадров.

1.3.6.3.3.1 Требования к полосе частот

Ширина полосы частот, необходимая для кодирования и декодирования JPEG, зависит от размера изображения и скорости передачи битов. Например, блоку JPU требуется только один буфер кадров для декодирования статического изображения, поэтому ширина полосы частот для воспроизводимого изображения будет той же, что и размер буфера кадров. Что касается буфера битовых потоков, коэффициент сжатия, как правило, составляет около 20% изображения JPEG. Поэтому требуемая ширина полосы частот для битового потока может составить около 20% ширины полосы частот воспроизводимого изображения. Требования к ширине полосы частот одинаковы для операций кодирования и декодирования.

1.3.6.3.3.2 Буфер кадров

1.3.6.3.3.2.1 Карта распределения памяти

Исходные и декодированные изображения (содержащие кадры или поля) состоят из трех сеток дискретных данных, одной сетки дискретных значений яркости и двух сеток дискретных значений цветности. В настоящем документе под буфером кадров подразумевается сетка дискретных значений во внешней памяти. Далее приводятся характеристики буфера кадров:

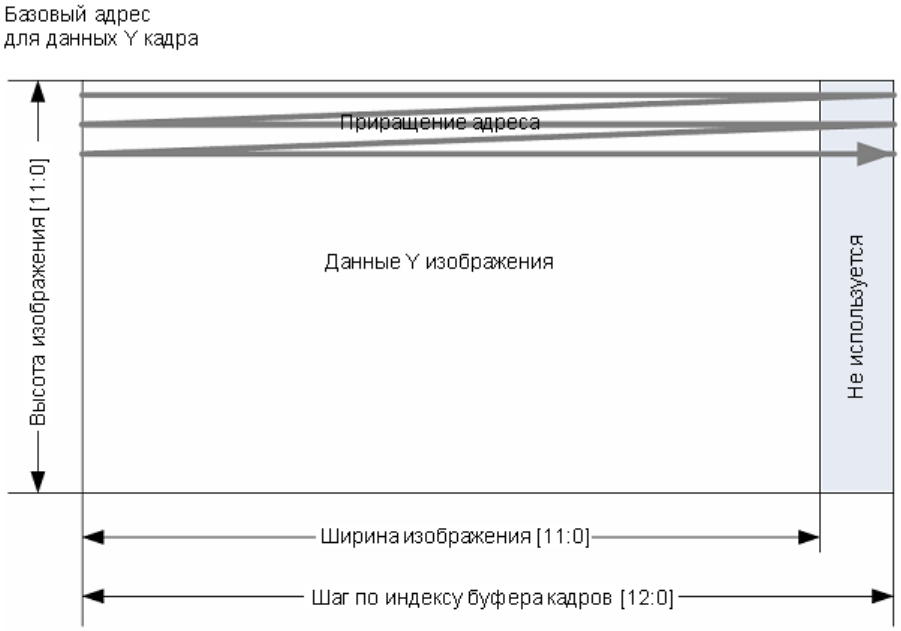
- поддерживаемые форматы: 4:0:0, 4:2:0, 4:2:2 и 4:4:4
- Буфер кадров сконфигурирован со следующими параметрами:
  - Базовый адрес буфера кадров компонентов яркости содержится в восьмом байте
  - Базовый адрес буфера кадров компонентов Cb содержится в восьмом байте
  - Базовый адрес буфера кадров компонентов Cr содержится в восьмом байте: в случае карты памяти с чередованием компонентов Cb/Cr базовый адрес для сетки компонентов Cr не имеет смысла, так как для хранения или загрузки чередующихся дискретных значений компонентов Cb/Cr используется базовый адрес для Cb.
  - Шаг по индексу для ширины буфера кадров яркости: Шаг по индексу должен быть равным или превышать ширину изображения и быть кратным 8. Это значит, что младший значащий бит 3 шага в 13 бит должен всегда составлять 0. Шаг по индексу буферов кадров компонентов цветности составляет половину шага по индексу буферов кадров компонентов яркости. Но необходимо убедиться, что в карте распределения памяти без чередования компонентов Cb/Cr шаг по индексу для буферов кадров компонентов яркости является кратным 16, чтобы шаг по индексу для буферов кадров компонентов цветности мог быть кратным 8.
- Порядок следования байтов буферов кадров
  - Порядок следования байтов буферов кадров конфигурируется как прямой и обратный порядок следования байтов.
- Ширина внутренних регистров для задания буфера кадров и размера изображения составляет 15 битов.
  - Шаг по индексу для задания ширины буфера кадров в пикселях
  - Размер изображения в пикселях для горизонтального и вертикального направлений

- Поддерживаются карты распределения памяти с чередованием компонентов Cb/Cr и без чередования компонентов Cb/Cr.
- Для изображений, составленных из полей, пара полей (верхнее и нижнее) хранится в одном буфере кадров.
-

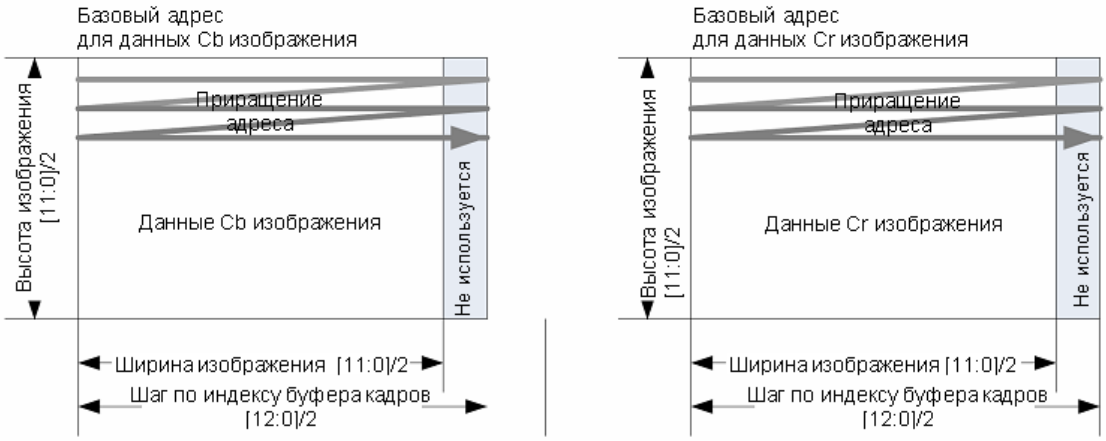
1.3.6.3.3.2.2 Разметка изображений в буферах кадров

Рисунок 1.141 «Конфигурация буфера кадров» отображает конфигурацию буфера кадров. Сконфигурировать буфер кадров можно двумя способами. Первая конфигурация включает (a) и (b).

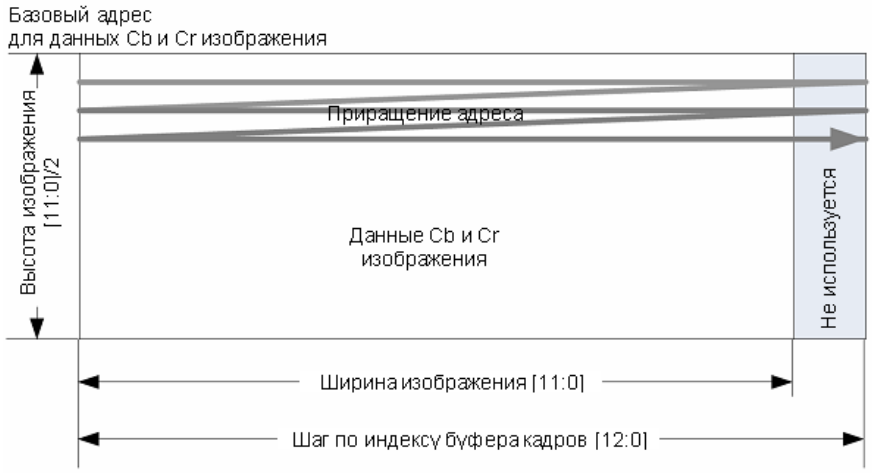
В данной конфигурации каждый компонент цветности имеет собственный буфер. Поэтому компоненты Cb и Cr используют два буфера. Вторая конфигурация включает (a) и (c). Данная конфигурация используется для снижения нагрузки на шину, использующую более длинные пакеты импульсов, чем в первой конфигурации. Как показано в (c), дискретные значения Cb и Cr чередуются.



(a) Буфер кадров для компонента яркости



(b) Буфер кадров для не чередующихся компонентов Cb и Cr



(c) Буфер кадров для чередующихся компонентов Cb и Cr

Рисунок 1.141 - Конфигурация 1 буфера кадров

В нижеприведенных разделах содержится подробное описание данных конфигураций, касающееся порядка следования байтов, на примерах для изображений QCIF (176x144).

1.3.6.3.3.2.3 Буферы кадров для компонента яркости

На рисунке 1.142 «Расположение пикселей по яркости в буфере кадров» показано расположение дискретных значений яркости во внешней памяти. На (a) представлен режим с обратным порядком следования байтов, на (b) – режим с прямым порядком следования байтов. Базовый адрес на рисунке указан для буфера кадров яркости.

Приращение адреса

0 63

Базовый адрес	Y(0,0)	Y(0,1)	Y(0,2)	Y(0,3)	Y(0,4)	Y(0,5)	Y(0,6)	Y(0,7)
	Другие пиксели Y в первом ряду пикселей							
Базовый адрес + 168	Y(0,168)	Y(0,169)	Y(0,170)	Y(0,171)	Y(0,172)	Y(0,173)	Y(0,174)	Y(0,175)
	Неиспользуемая область							
Базовый адрес + шаг по индексу*1	Y(1,0)	Y(1,1)	Y(1,2)	Y(1,3)	Y(1,4)	Y(1,5)	Y(1,6)	Y(1,7)
	Другие пиксели Y во втором ряду пикселей							
	Y(1,168)	Y(1,169)	Y(1,170)	Y(1,171)	Y(1,172)	Y(1,173)	Y(1,174)	Y(1,175)
	Неиспользуемая область							
	Области для других рядов пикселей							
Базовый адрес + шаг по индексу*143	Y(143,0)	Y(143,1)	Y(143,2)	Y(143,3)	Y(143,4)	Y(143,5)	Y(143,6)	Y(143,7)
	Другие пиксели Y в 144-ом ряду пикселей							
	Y(143,168)	Y(143,169)	Y(143,170)	Y(143,171)	Y(143,172)	Y(143,173)	Y(143,174)	Y(143,175)
	Неиспользуемая область							

Расположение пикселей при ENDIAN_MODE = 2’ b00

Приращение адреса

63 … 0

Y(0,0)	Y(0,1)	Y(0,2)	Y(0,3)	Y(0,4)	Y(0,5)	Y(0,6)	Y(0,7)	Базовый адрес + 0
Другие пиксели Y в первом ряду пикселей								
Y(0,168)	Y(0,169)	Y(0,170)	Y(0,171)	Y(0,172)	Y(0,173)	Y(0,174)	Y(0,175)	Базовый адрес + 168
Неиспользуемая область								
Y(1,0)	Y(1,1)	Y(1,2)	Y(1,3)	Y(1,4)	Y(1,5)	Y(1,6)	Y(1,7)	Базовый адрес + шаг по индексу*1
Другие пиксели Y во втором ряду пикселей								
Y(1,168)	Y(1,169)	Y(1,170)	Y(1,171)	Y(1,172)	Y(1,173)	Y(1,174)	Y(1,175)	
Неиспользуемая область								
Области для других рядов пикселей								
Y(143,0)	Y(143,1)	Y(143,2)	Y(143,3)	Y(143,4)	Y(143,5)	Y(143,6)	Y(143,7)	Базовый адрес + шаг по индексу*143
Другие пиксели Y в 144-ом ряду пикселей								
Y(143,168)	Y(143,169)	Y(143,170)	Y(143,171)	Y(143,172)	Y(143,173)	Y(143,174)	Y(143,175)	
Неиспользуемая область								

Расположение пикселей при ENDIAN_MODE = 2’ b01

Рисунок 1.142 - Расположение пикселей по яркости в буфере кадров

1.3.6.3.3.2.4 Буферы кадров для компонентов цветности

На рисунке 1.143 «Расположение пикселей по цветности в буфере кадров без чередования CbCr» показано расположение дискретных значений цветности в буфере без чередования Cb/Cr внешней памяти. На (a) представлен режим с обратным порядком следования байтов, на (b) – режим с прямым порядком следования байтов. Каждый компонент цветности имеет собственный буфер.

Приращение адреса

0 … 63

Базовый адрес	Cx(0,0)	Cx(0,1)	Cx(0,2)	Cx(0,3)	Cx(0,4)	Cx(0,5)	Cx(0,6)	Cx(0,7)
Другие пиксели Cx в первом ряду пикселей								
Базовый адрес + 80	Cx(0,80)	Cx(0,81)	Cx(0,82)	Cx(0,83)	Cx(0,84)	Cx(0,85)	Cx(0,86)	Cx(0,87)
Неиспользуемая область								
Базовый адрес + (шаг по индексу/2)*1	Cx(1,0)	Cx(1,1)	Cx(1,2)	Cx(1,3)	Cx(1,4)	Cx(1,5)	Cx(1,6)	Cx(1,7)
Другие пиксели Cx во втором ряду пикселей								
	Cx(1,80)	Cx(1,81)	Cx(1,82)	Cx(1,83)	Cx(1,84)	Cx(1,85)	Cx(1,86)	Cx(1,87)
Неиспользуемая область								
Области для других рядов пикселей								
Базовый адрес + (шаг по индексу/2)*71	Cx(71,0)	Cx(71,1)	Cx(71,2)	Cx(71,3)	Cx(71,4)	Cx(71,5)	Cx(71,6)	Cx(71,7)
Другие пиксели Cx в 72-ом ряду пикселей								
	Cx(71,80)	Cx(71,81)	Cx(71,82)	Cx(71,83)	Cx(71,84)	Cx(71,85)	Cx(71,86)	Cx(71,87)
Неиспользуемая область								

Расположение пикселей при ENDIAN_MODE = 2’ b00 для буфера без чередования Cb/Cr

Приращение адреса

63 … 0

Cx(0,0)	Cx(0,1)	Cx(0,2)	Cx(0,3)	Cx(0,4)	Cx(0,5)	Cx(0,6)	Cx(0,7)	Базовый адрес
Другие пиксели Cx в первом ряду пикселей								
Cx(0,80)	Cx(0,81)	Cx(0,82)	Cx(0,83)	Cx(0,84)	Cx(0,85)	Cx(0,86)	Cx(0,87)	Базовый адрес + 80
Неиспользуемая область								
Cx(1,0)	Cx(1,1)	Cx(1,2)	Cx(1,3)	Cx(1,4)	Cx(1,5)	Cx(1,6)	Cx(1,7)	Базовый адрес + (шаг по индексу/2)*1
Другие пиксели Cx во втором ряду пикселей								
Cx(1,80)	Cx(1,81)	Cx(1,82)	Cx(1,83)	Cx(1,84)	Cx(1,85)	Cx(1,86)	Cx(1,87)	
Неиспользуемая область								
Области для других рядов пикселей								
Cx(71,0)	Cx(71,1)	Cx(71,2)	Cx(71,3)	Cx(71,4)	Cx(71,5)	Cx(71,6)	Cx(71,7)	Базовый адрес + (шаг по индексу/2)*71
Другие пиксели Cx в 72-ом ряду пикселей								
Cx(71,80)	Cx(71,81)	Cx(71,82)	Cx(71,83)	Cx(71,84)	Cx(71,85)	Cx(71,86)	Cx(71,87)	
Неиспользуемая область								

Расположение пикселей при ENDIAN_MODE = 2’ b01 для буфера без чередования Cb/Cr

Рисунок 1.143 - Расположение пикселей по цветности в буфере без чередования CbCr

На рисунке 1.144 «Расположение пикселей по цветности в буфере кадров с чередованием CbCr» показано расположение дискретных значений цветности в буфере с чередованием Cb/Cr внешней памяти. На (a) представлен режим с обратным порядком следования байтов, на (b) – режим с прямым порядком следования байтов.

На карте с чередованием Cb/Cr для компонентов Cb и Cr изображения выделяется только один буфер. Шаг по индексу буфера тот же, что и у буфера яркости.

0 — Приращение адреса → 63

Базовый адрес	Cb(0,0)	Cr(0,0)	Cb(0,1)	Cr(0,1)	Cb(0,2)	Cr(0,2)	Cb(0,3)	Cr(0,3)
	Другие пиксели Cb и Cr в первом ряду пикселей							
Базовый адрес + 168	Cb(0,84)	Cr(0,84)	Cb(0,85)	Cr(0,85)	Cb(0,86)	Cr(0,86)	Cb(0,87)	Cr(0,87)
	Неиспользуемая область							
Базовый адрес + шаг по индексу*1	Cb(1,0)	Cr(1,0)	Cb(1,1)	Cr(1,1)	Cb(1,2)	Cr(1,2)	Cb(1,3)	Cr(1,3)
	Другие пиксели Cb и Cr во втором ряду пикселей							
	Cb(1,84)	Cr(1,84)	Cb(1,85)	Cr(1,85)	Cb(1,86)	Cr(1,86)	Cb(1,87)	Cr(1,87)
	Неиспользуемая область							
	Области для других рядов пикселей							
Базовый адрес + шаг по индексу*71	Cb(71,0)	Cr(71,0)	Cb(71,1)	Cr(71,1)	Cb(71,2)	Cr(71,2)	Cb(71,3)	Cr(71,3)
	Другие пиксели Cx в 72-ом ряду пикселей							
	Cb(71,84)	Cr(71,84)	Cb(71,85)	Cr(71,85)	Cb(71,86)	Cr(71,86)	Cb(71,87)	Cr(71,87)
	Неиспользуемая область							

Расположение пикселей при ENDIAN_MODE = 2’ b00 для буфера с чередованием Cb/Cr

Приращение адреса

63 … 0

Cb(0,0)	Cr(0,0)	Cb(0,1)	Cr(0,1)	Cb(0,2)	Cr(0,2)	Cb(0,3)	Cr(0,3)	Базовый адрес
Другие пиксели Cb и Cr в первом ряду пикселей								
Cb(0,84)	Cr(0,84)	Cb(0,85)	Cr(0,85)	Cb(0,86)	Cr(0,86)	Cb(0,87)	Cr(0,87)	Базовый адрес + 168
Неиспользуемая область								
Cb(1,0)	Cr(1,0)	Cb(1,1)	Cr(1,1)	Cb(1,2)	Cr(1,2)	Cb(1,3)	Cr(1,3)	Базовый адрес + шаг по индексу*1
Другие Cb и Cr пиксели во 2-ом ряду пикселей								
Cb(1,84)	Cr(1,84)	Cb(1,85)	Cr(1,85)	Cb(1,86)	Cr(1,86)	Cb(1,87)	Cr(1,87)	
Неиспользуемая область								
Области для других рядов пикселей								
Cb(71,0)	Cr(71,0)	Cb(71,1)	Cr(71,1)	Cb(71,2)	Cr(71,2)	Cb(71,3)	Cr(71,3)	Базовый адрес + шаг по индексу*71
Другие пиксели Cx в 72-ом ряду пикселей								
Cb(71,84)	Cr(71,84)	Cb(71,85)	Cr(71,85)	Cb(71,86)	Cr(71,86)	Cb(71,87)	Cr(71,87)	
Неиспользуемая область								

Расположение пикселей при ENDIAN_MODE = 2’ b01 для буфера с чередованием Cb/Cr

Рисунок 1.144 - Расположение пикселей по цветности в буфере с чередованием CbCr

1.3.6.3.3.2.5 Форматы буферов кадров JPEG

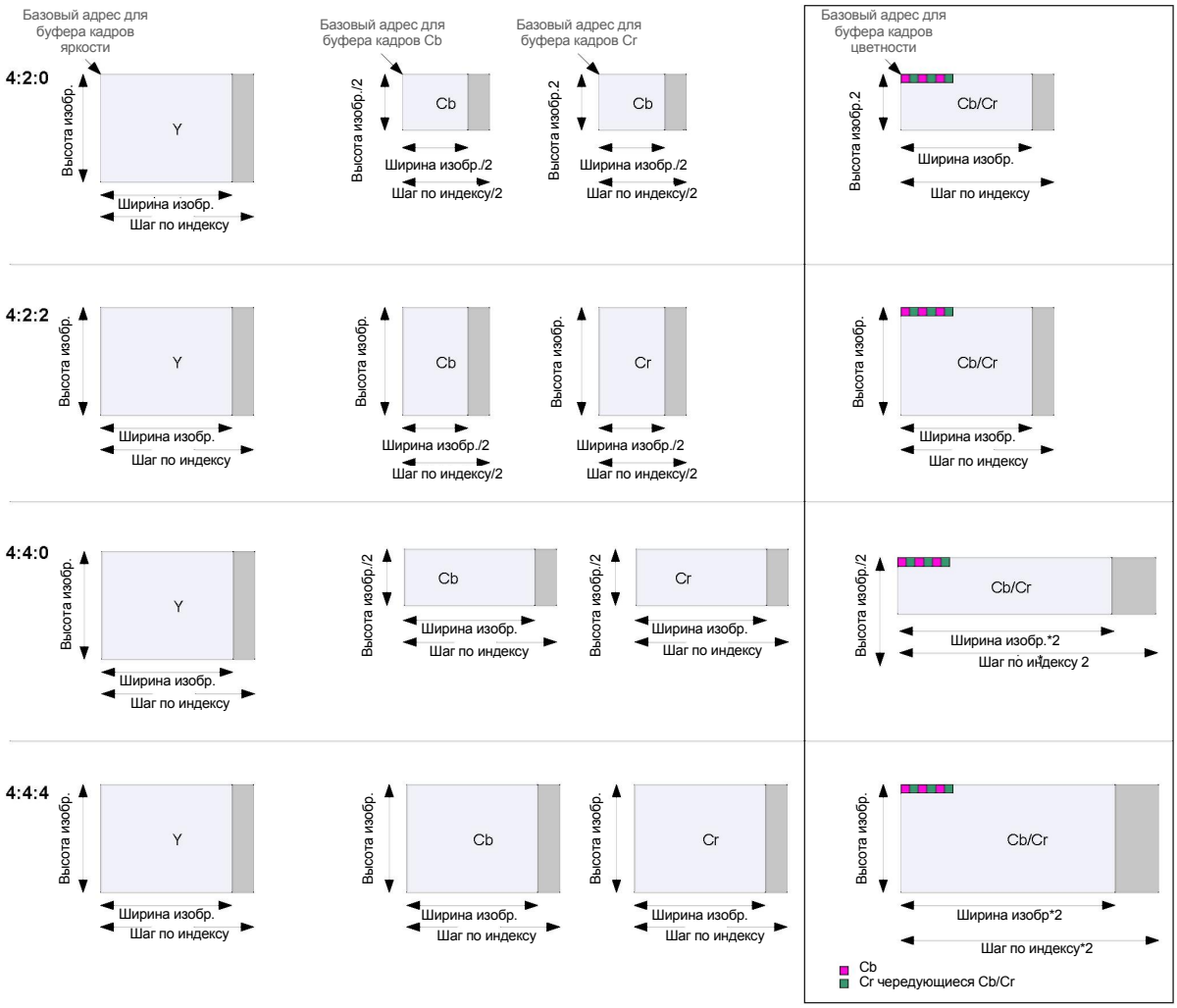


Рисунок 1.145 - Форматы буферов кадров JPEG

1.3.6.3.3.2.5.1 Формат буфера кадров в режиме упакованного десятичного кода 444/422

Блок JPU поддерживает формат упакованного десятичного кода для форматов цветного изображения 444 и 422. На рисунке 1.146 «Упакованный десятичный формат 444» показаны схемы считывания или записи каждого цветового компонента в двумерном и упакованном форматах.

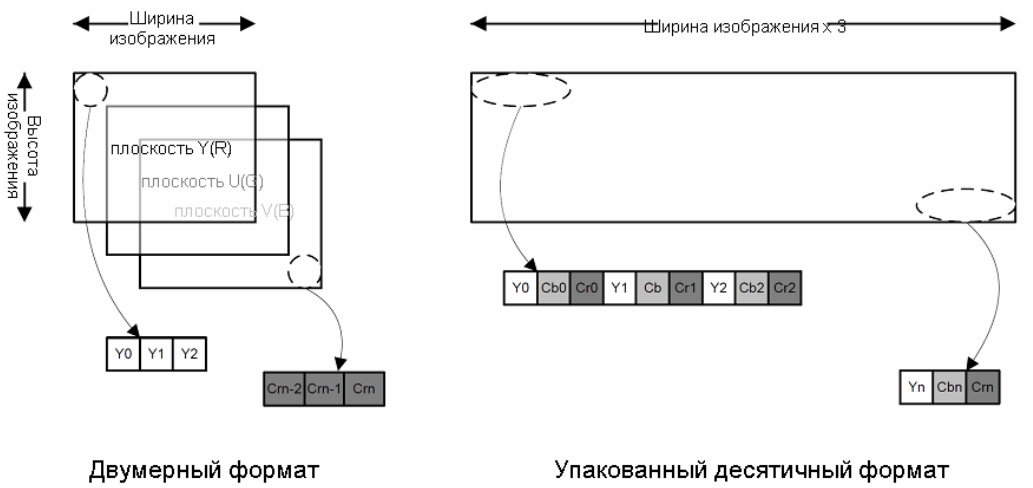


Рисунок 1.146 - Упакованный десятичный формат 444

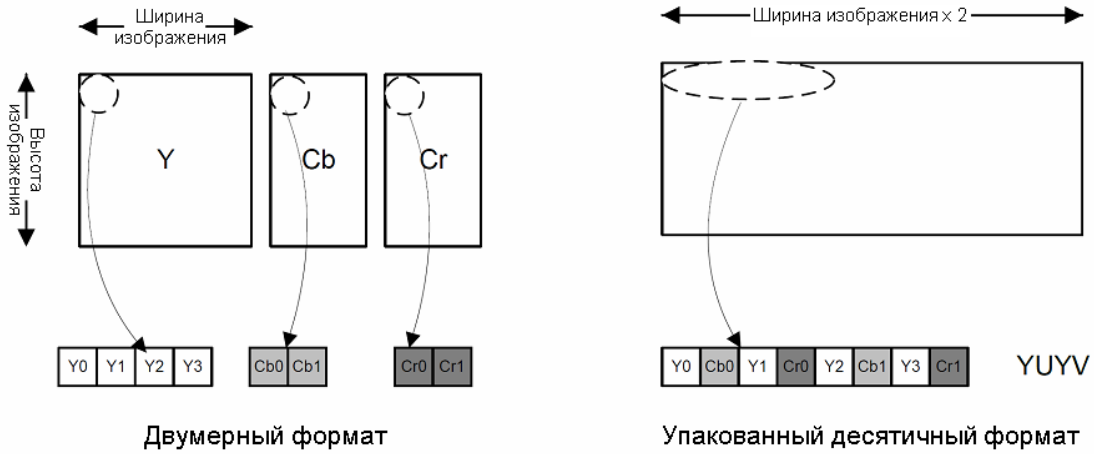


Рисунок 1.147 - Упакованный десятичный формат 422

Упакованный десятичный формат использует одну плоскость, в которой размещается пара данных Y/Cb/Cr каждый 8 бит (всего 24 бита), в то время как двумерный формат использует три различные плоскости для каждого компонента Y, Cb и Cr. Главный компьютер получает данные о данных форматах – входной формат исходного изображения для работы кодека и выходной формат декодированного изображения для отображения.

1.3.6.3.3.2.6 Размер буфера в памяти SDRAM

Буферы кадров в SDRAM отличаются для каждого размера изображения и цветового формата. Таблица ниже приводит необходимый объем памяти (SDRAM) для размеров изображения и цветовых форматов.

Таблица 1.249 - Требуемый цикл для кодирования изображений

Размер изображения	Формат 4:0:0	Формат 4:2:0	Формат 4:2:2	Формат 4:4:4
176x144	25 344 Байт	38 016 Байт	50 688 Байт	76 032 Байт
352x288	101 376 Байт	152 064 Байт	202 752 Байт	304 128 Байт
640x480	307 200 Байт	460 800 Байт	614 400 Байт	921 600 Байт
1280x720	921 600 Байт	1 382 400 Байт	1 843 200 Байт	2 764 800 Байт
8192x8192	67 108 864 Байт	100 663 296 Байт	134 217 728 Байт	201 326 592 Байт

1.3.6.3.3.3 Буферы битовых потоков

Буфер битовых потоков хранит битовый поток, записанный внешним главным процессором. Если пользователь конфигурирует Nieuport для декодирования битовых потоков, должен быть назначен буфер битовых потоков, базовый адрес которого и размер должны быть переданы на Nieuport. Ограничения по размеру буфера битовых потоков отсутствуют. Но пользователю необходимо задать регистры управления блоком BBC, особенно регистр mjpeg_bbc_int_addr_reg. Данный регистр представляет конец адреса битового потока. Если указатель кодера/декодера достигает конца адреса, главный процессор может получить сигнал прерывания от кодека JPEG. Данное прерывание означает, что буфер битовых потоков пуст, и главный процессор должен загрузить битовый поток во внешнюю память или изменить базовый адрес битового потока.

1.3.6.3.4 АППАРАТНЫЕ БЛОКИ JPEG

1.3.6.3.4.1 Описание работы

1.3.6.3.4.1.1 Работа кодера

1.3.6.3.4.1.1.1 Поток данных

На рисунке 1.148 показан поток данных процесса кодирования, осуществляемого блоком JPU.

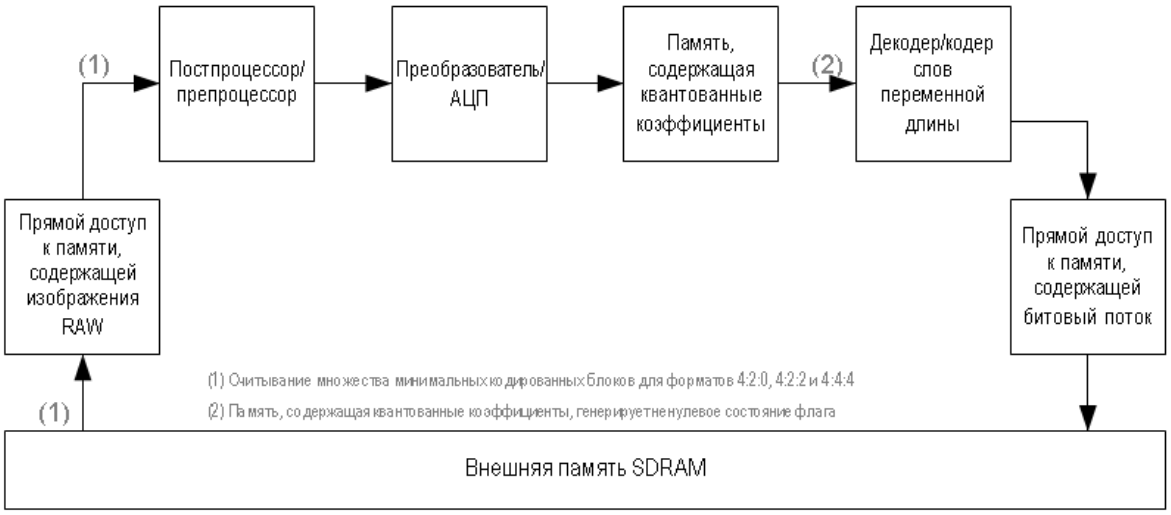


Рисунок 1.148 - Поток данных кодера JPEG

Когда начинается операция кодирования, контроллер прямого доступа к памяти, содержащей элементы изображения в формате RAW, производит считывание данных яркости и цветности из SDRAM и записывает эти данные во внутренний буфер для предварительной обработки. В случае, когда активирован режим поворота/зеркального отображения, препроцессор поворачивает или зеркально отражает данные и сохраняет их в локальный буфер.

Данные элементы изображения после предварительной обработки направляются в преобразователь и АЦП. Перед началом работы преобразователь осуществляет сдвиг уровня. И квантованные остаточные данные записываются в буфер памяти QCM.

Затем кодер слов переменной длины (VLC) считывает остаточные данные с ненулевым флагом, полученные из QCM, и осуществляет кодирование по алгоритму Хаффмана. Блок VLC также осуществляет предварительную оценку коэффициентов DC.

Наконец, когда локальный буфер битовых потоков полон, контроллер прямого доступа к памяти, содержащей битовый поток, производит запись кодированного битового потока в память SDRAM.

1.3.6.3.4.1.1.2 Информационный канал

На рисунке 1.149 представлены четыре ступени информационного канала, по которому осуществляется кодирование JPEG.

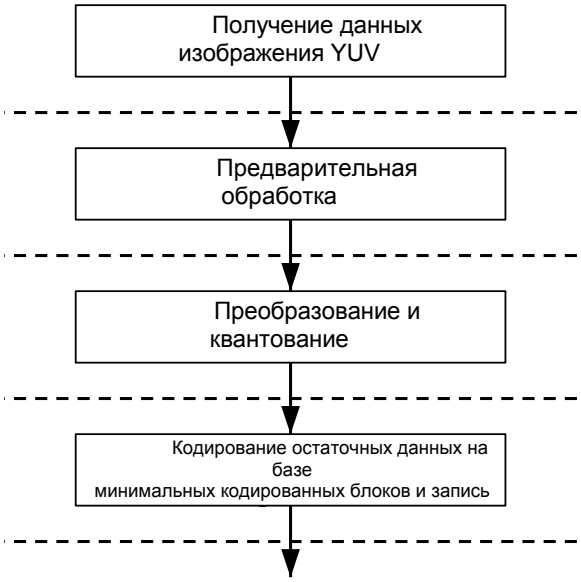


Рисунок 1.149 - Информационный канал кодера JPEG

Получение данных YUV

Контроллер прямого доступа к памяти считывает изображение YUV для множества минимальных кодированных блоков (MCU – Mininum Coded Unit) из SDRAM. Изображение для минимальных кодированных блоков записывается во внутренний буфер.

Предварительная обработка

Если активирован режим поворота/зеркального отображения, аппаратный блок предварительной обработки считывает данные изображения YUV из внутреннего буфера и осуществляет поворот или зеркальное отображение изображения.

Преобразование и квантование

Остаточные данные из блока предварительной обработки считываются и преобразуются аппаратным блоком. Выходные данные отправляются в блок квантования (аналого-цифрового преобразования). Результаты аналого-цифрового преобразования, произведенные аппаратным блоком квантования, записываются в буфер коэффициентов.

Остаточные данные кодируются на базе минимальных кодированных блоков и записываются.

Блок VLC считывает данные буфера коэффициентов и генерирует битовый поток. И контроллер прямого доступа к памяти, содержащей битовый поток, записывает кодированный битовый поток в память SDRAM.

1.3.6.3.4.1.2 Работа декодера

1.3.6.3.4.1.2.1 Поток данных

Операция декодирования данных кажется абсолютно симметричной относительно операции кодирования данных, за исключением этапа последующей обработки. Последующая обработка, производимая декодером, поддерживает режим субдискретизации.

1.3.6.3.4.1.2.2 Информационный канал

На рисунке 1.150 «Информационный канал декодера JPEG» представлена структура информационного канала, по которому осуществляется декодирование JPEG.

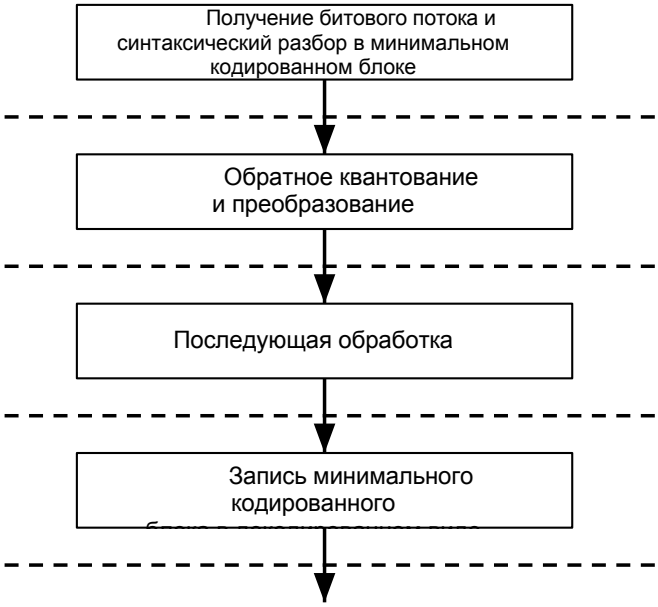


Рисунок 1.150 - Информационный канал декодера JPEG

Получение битового потока и синтаксический разбор в минимальном кодированном блоке

Контроллер прямого доступа к памяти, содержащей битовый поток, получает данные битового потока, и блок VLD производит синтаксический разбор коэффициентов одного минимального кодированного блока. Декодированные коэффициенты записываются в буфер коэффициентов, из которого считываются аппаратными блоками обратного преобразования и обратного квантования.

Обратное квантование и преобразование

Аппаратное обеспечение считывает коэффициенты в буфере и производит их обратное квантование. Результаты не сохраняются в буфере, а передаются на аппаратный блок обратного преобразования. Выходные данные аппаратного блока обратного преобразования, остаточные данные минимальных кодированных блоков, записываются во внутренний буфер.

Последующая обработка

На данном этапе аппаратный блок последующей обработки считывает остаточные данные и производит дополнительную обработку, такую как поворот/зеркальное отображение или субдискретизация.

Запись минимального кодированного блока в декодированном виде

Восстановленные минимальные кодированные блоки записываются во внешнюю память через контроллер прямого доступа к памяти.

1.3.6.3.4.2 Аппаратные блоки

1.3.6.3.4.2.1 Блок MB

Блок MB (блок прямого доступа к памяти, содержащей минимальный кодированный блок), контроллер прямого доступа к памяти, содержащей изображение RAW (Рисунок 8.1, «Поток данных кодера JPEG»), производит считывание данных изображения из SDRAM в режиме кодирования и записывает декодированное изображение в SDRAM в режиме декодирования.

Данный модуль поддерживает функцию чтения/записи множества минимальных кодированных блоков, что способствует повышению эффективности использования полосы частот. Когда данная функция активирована посредством задания соответствующего регистра, модуль прямого доступа к памяти производит считывание или запись множества минимальных кодированных блоков из/в SDRAM на этапе передачи информации, что способствует увеличению длины пакетов импульсов, передаваемых по шине AXI. Например, длина пакета импульсов для минимального кодированного блока с яркостью 422 составляет только 2. Тем не менее, при передаче 3 минимальных кодированных блоков на этапе передаче информации длина пакета импульсов может достигать 6. Данный тип увеличения длины пакета импульсов способен повысить эффективность использования полосы частот. Нижеприведенная таблица отображает максимальное количество минимальных кодированных блоков для каждого формата.

Таблица 1.250 - Характеристики кодека MJPEG

Формат	Максимальное количество минимальных кодированных блоков на этапе передачи информации
4:2:0	2
4:2:2 или 4:4:0	3
4:4:4	4
4:0:0	4

Примечание - В случае если задана зона ROI в режиме декодера или кодер/декодер работает в режиме использования формата уплотненного десятичного кода, считывание/запись множества минимальных кодированных блоков не поддерживается.

1.3.6.3.4.2.2 Блок PP

Блок PP (Препроцессор/постпроцессор) поддерживает функцию поворота/зеркального отображения, как для режима кодирования, так и декодирования, а также функцию масштабирования для режима декодирования.

Модуль PP поддерживает поворот на 90 x n градусов (n=0, 1, 2, 3) и зеркальное отображение. Когда опции поворота и зеркального отображения активированы одновременно, блок PP сначала производит зеркальное отображение изображения, после чего поворачивает его согласно заданным настройкам.

Ниже приведены поддерживаемые режимы поворота/зеркального отображения, а также все возможные комбинации поворота и зеркального отображения.

Таблица 1.251 - Характеристики кодека MJPEG

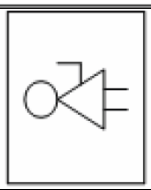
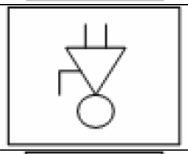
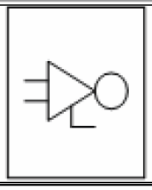
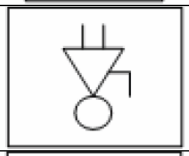
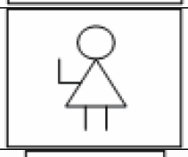
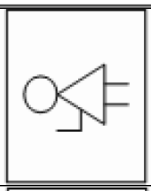
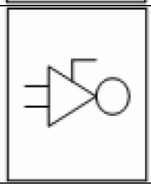
Режим	Изображение	Описание
NONE_ROTATE 4'b0000, 4'b1110		Исходное изображение (без обработки) Примерный размер изображения: 720x480
ROT LEFT 90 4'b0001, 4'b1111		Поворот влево на 90 градусов (поворот вправо на 270 градусов) Размер изображения: 480x720
ROT_LEFT_130 4'b0010, 4'b1100		Поворот влево на 180 градусов Размер изображения: 720x480
ROT_LEFT_270 4'b0011, 4'b1101		Поворот влево на 270 градусов (поворот вправо на 90 градусов) Размер изображения: 480x720
MIR_VERT 4'b0100, 4'b1010		Зеркальное отображение по вертикали Размер изображения: 720x480
MIR_HORIZ 4'b1000, 4'b0110		Зеркальное отображение по горизонтали Размер изображения: 720x480
MIR_VERT_ROT_LEFT_270 4'b0111, 4'b1001		Сначала зеркальное отображение по вертикали, затем поворот влево на 270 градусов Размер изображения: 480x720
MIR_VERT_ROT_LEFT_90 4'b0101,4'b1011		Сначала зеркальное отображение по вертикали, затем поворот влево на 90 градусов Размер изображения: 480x720

Таблица 1.251 отображает режимы поворота/зеркального отображения. В приведенных выше примерах входные данные для всех режимов поворота/зеркального отображения аналогичны выходным данным изображения, обработанного в режиме NONE_ROTATE. Главный процессор может выбрать один из следующих режимов посредством задания соответствующего регистра.

Поле изображения в режимах до и после поворота является 4-битовым. Первый бит (старший значащий бит) отвечает за зеркальное отображение по горизонтали, следующий бит – за зеркальное отображение по вертикали и два последние бита (два младших значащих бита) – за вращение на определенное количество градусов против часовой стрелки. Каждое поле функции поворота/зеркального отображения (по горизонтали, по вертикали, поворот) используются независимо.

Блок PP также поддерживает режим масштабирования (билинейный или бикубический с четырьмя опорными точками интерполяции (параметр TAPS)), который является действительным только в режиме декодирования. Коэффициент масштабирования составляет 1/2, 1/4 и 1/8 как для горизонтального, так и для вертикального направления. Если размер декодированного изображения составляет 2048x1600, коэффициент масштабирования по горизонтали будет составлять 1/2, а по вертикали – 1/4, исходное изображение будет уменьшено блоком PP до размера 1024x400.

Примечание - При активном режиме масштабирования режим поворота/зеркального отображения должен быть деактивирован, так как два данных режима взаимно исключают друг друга.

1.3.6.3.4.2.3 Блок TQ

В блоке TQ (Блок преобразования/квантования) за дискретное косинусное преобразование (DCT) и обратное дискретное косинусное преобразование отвечает один аппаратный блок. Оба блока используют один информационный канал, но разные матрицы коэффициентов. Блок DCT сначала считывает данные коэффициентов по горизонтали, после чего записывает их по вертикали. И наоборот, блок обратного дискретного косинусного преобразования считывает данные по вертикали и записывает их по горизонтали.

Блок дискретного косинусного преобразования и блок обратного дискретного косинусного преобразования имеют двухканальную структуру. Данные двух коэффициентов считываются и записываются за один такт. Поэтому блок QCM IF два порта ввода/вывода для данных коэффициентов. Блок квантования производит масштабирование данных коэффициентов с использованием данной таблицы QMatrix. Данная таблица берется из заголовка изображения JPEG или определяется пользователем. Блок выполняет данную функцию немедленно.

1.3.6.3.4.2.4 Блок QCM IF

Блок QCM IF (Интерфейс памяти, содержащей квантованные коэффициенты) осуществляет зигзагообразное сканирование. В режиме кодирования блок TQ производит обратную запись коэффициентов в блок QCM IF, который автоматически генерирует ненулевые флаги, после чего следующий блок, TC, считывает только действительные данные посредством проверки ненулевых флагов. Перед тем как блок TQ начнет обратную запись блок QCM IF производит сброс ненулевых флагов для осуществления обратной записи следующих коэффициентов минимального кодированного блока.

1.3.6.3.4.2.5 Кодер/декодер слов переменной длины

Аппаратные блоки VLC (Кодирование слов переменной длины) и VLD (Декодирование слов переменной длины) включают подблоки TC и GBU.

1.3.6.3.4.2.5.1 Блок коэффициента преобразования кодера

Блок TC (Коэффициент преобразования) кодера производит обработку данных после кодирования по алгоритму Хаффмана. Он может кодировать коэффициенты с использованием пользовательской таблицы Хаффмана или таблицы Хаффмана по умолчанию и отправлять один тактовый цикл по завершении кодирования одного значения коэффициента. Блок TC также создает маркер перезапуска, после чего главный компьютер может задать интервал перезапуска.

Упаковка данных или эмуляция осуществляется в блоке GBU. Блок TC кодера отправляет сгенерированные данные блоку GBU по тактовому сигналу для сжатия битовых потоков.

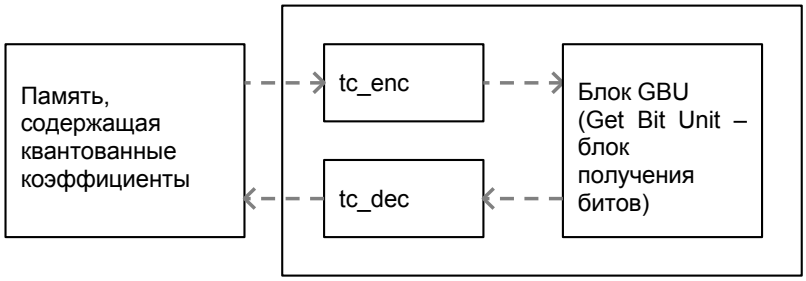


Рисунок 1.252 - Блок-схема работы модуля VLD/VLC

Примечание - Подробная информация о таблице Хаффмана и ее формате приведена в руководстве для программиста.

1.3.6.3.4.2.5.2 Блок коэффициента преобразования декодера

Блок TC (Коэффициент преобразования) декодера производит декодирование передаваемых данных коэффициентов. Аппаратное обеспечение блока TC декодера поддерживает декодирование по алгоритму Хаффмана и декодирование с использованием таблицы Хаффмана по умолчанию. Когда данный блок производит декодирование с использованием пользовательского дерева Хаффмана, перед декодированием коэффициентов он должен сохранить данные таблицы в памяти. Тем не менее, таблица Хаффмана по умолчанию хранится в аппаратном обеспечении для обеспечения удобства пользователей и избавления их от необходимости сохранять таблицу Хаффмана по умолчанию.

Один цикл посвящается декодированию с использованием дерева Хаффмана по умолчанию. С другой стороны, декодирование с использованием пользовательского дерева Хаффмана занимает два цикла.

Аппаратный блок TC декодера получает кодированные битовые потоки от аппаратного блока GBU. Блок GBU оправляет блоку TC декодера запрос на отображение битового потока или получения битового потока, ShowBit() или GetBits(), соответственно. В конечном итоге он отправляет декодированные данные коэффициентов блоку QCM IF.

1.3.6.3.4.2.5.3 Блок GBU

Блок GBU (Get Bit Unit – Блок получения битов) осуществляет уплотнение и распаковку битовых потоков по специализированной ускоренной логической схеме. Блок GBU проверяет маркер FF в режиме декодирования или вставляет код маркера FF в режиме кодирования. Он может за один цикл передавать не более 32 битов данных, за исключением маркера. Он передает один маркер байта за один такт, и данные маркера выравниваются по границе байта.

1.3.6.3.4.2.6 Блок BBC

Блок BBC (Bitstream Buffer control – блок управления буфером битовых потоков) осуществляет транспортировку битовых потоков между буфером битовых потоков во внешней памяти и локальными буферами блока GBU.

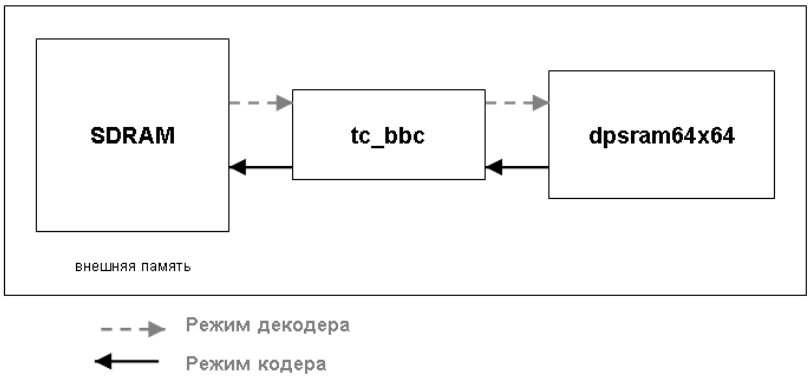


Рисунок 1.253 - Блок-схема блока BBC

Главный компьютер может контролировать блок BBC и отправлять данные битового потока по любому адресу в блок GBU. Размер транспортируемых данных также является достаточно гибким. Тем не менее, данный размер ограничивается размером буфера блока GBU.

Аппаратное обеспечение блока BBC за несколько раз осуществляет отправку или прием битовых потоков. Когда транспортировка данных блоком BBC завершена, формируется сигнал прерывания, отправляемый главному компьютеру. Необходимость проверки данного сигнала определяется пользователем. Блок BBC работает в двух режимах:

- автоматическая работа в режиме декодирования/обработки данных
- работа в ручном режиме в режиме заголовка (режим заголовка только для режима декодирования).

Метод двойной буферизации требует от кодека JPEG минимизации времени задержки, при этом размер каждого буфера ограничивается до 256 байтов; таким образом, общий размер буфера составляет 512 байтов.

1.3.6.4 Кодер JPEG (программная модель)

1.3.6.4.1 Общий обзор

Блок VPU (блок обработки видеоданных), разработанный компанией “Chips&Media” - это высокопроизводительное мультистандартное решение IP-видео, способное выполнять декодирование в форматах H.264/AVC BP, MP и HP, VC-1 SP, MP и AP, MPEG-1 и 2, MPEG4 SP/ASP, DivX и XviD, AVS, RV-8,9 и 10, VP8 и Theora, а также кодирование в форматах H.264/AVC, MPEG-4, H.263 в едином блоке IP-видео.

Блок VPU обеспечивает кодирование и декодирование до уровня HD с разрешением 4096x2208. На рисунке 1.254 представлена функциональная схема работы блока VPU.

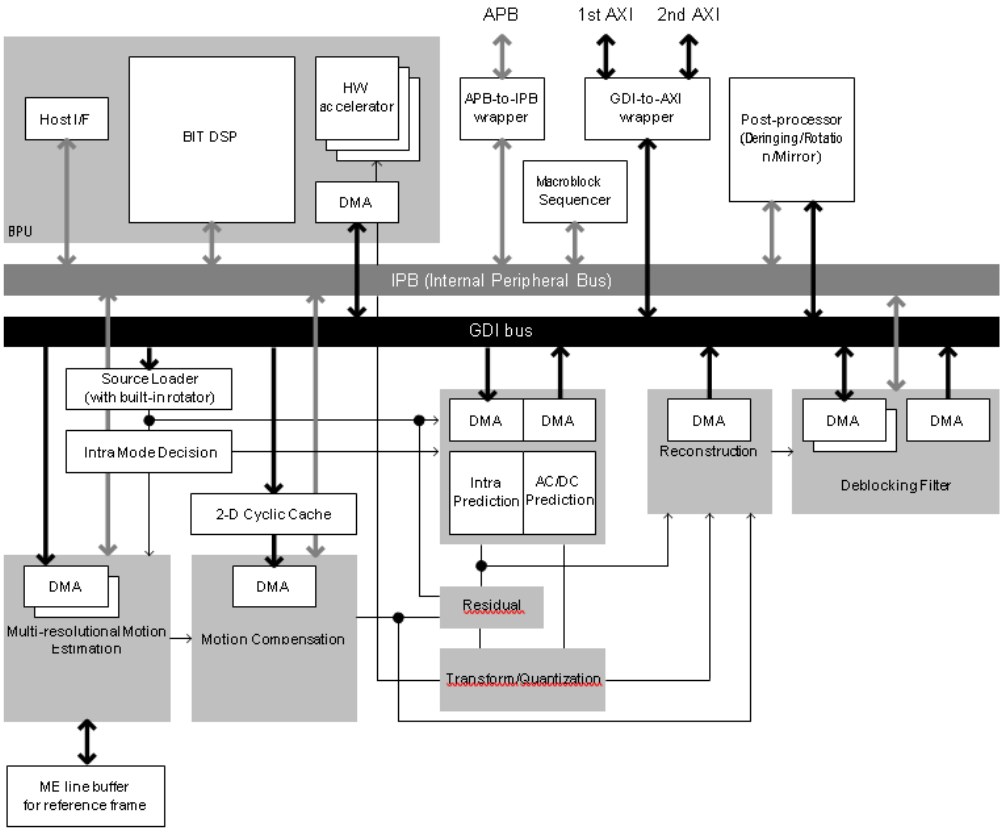


Рисунок 1.254 - Функциональная схема работы VPU

Host I/F – Главный интерфейс

HW accelerator – Аппаратный ускоритель

APB-to-IPB wrapper – Надстройка APB-IPB

Macroblock sequencer – Указатель следования макроблоков

GDI-to-AXI wrapper – Надстройка GDI- AXI

Post-processor (Deringing/Rotation/Mirror) – Постпроцессор (шумоподавление/вращение/отражение)

1st AXI – AXI по основной шине

2nd AXI – AXI по вспомогательной шине

IPB (Internal Peripheral Bus) – IPB (шина встроенных периферийных устройств)

GDI bus – Шина GDI

Source Loader (with built-in rotator) – Загрузчик исходных данных (со встроенным ротатором)

Intra Mode Decision – Выбор режима Intra

2-D Cyclic Cache – Циклический кэш 2-D

Intra Prediction – Intra-прогнозирование
AC/DC Prediction – Прогнозирование коэффициентов AC и DC
Reconstruction – Восстановление
Deblocking Filter – Фильтр деблокирования
Multi-Resolutional Motion Estimation – Оценка параметров движения для разных разрешений
Motion Compensation – Компенсация движения
Residual – Остаточные данные
Transform/Quantization – Преобразование/квантизация
ME Line Buffer for reference frame – Линейный буфер ME для ссылочного кадра

Блок VPU соединяется с системой через 32-разрядную расширенную шину подключения периферийных устройств (далее APB) AMBA3 для системного управления, а через 64-разрядную шину дополнительного расширяемого интерфейса (далее AXI) AMBA3 для передачи данных, а для достижения высокого уровня эффективности использует модули памяти на кристалле.

Блок VPU включает в себя 16-разрядный цифровой сигнальный процессор (далее DSP), являющийся собственностью компании “Chips&Media”, который называется BIT-процессором. BIT-процессор взаимодействует с хост-процессором через главный интерфейс и управляет другими субблоками VPU. Для обеспечения работы VPU не требуется большого ресурса CPU хоста (менее 1 миллиона команд в секунду), потому что все основные функции, такие как разбор битового потока, управление блоками видеоустройства и обеспечение устойчивости к ошибкам, реализуются в BIT-процессоре. Большая часть блока видеодекодера в VPU, за исключением BIT-процессора, оптимально распределена, что позволяет добиться сверхнизкого уровня потребления мощности и использовать меньшее количество транзисторных переключателей.

Как показано на приведенном ранее рисунке, блок VPU включает в себя 16-разрядное ядро DSP - BIT-процессор Chips&Media, управляющий внутренними операциями декодирования видеоданных в VPU. Кроме того, BIT-процессор через регистры главного интерфейса контролирует все взаимодействия между хост-процессором VPU.

Блок VPU встраивается в пользовательскую систему на кристалле (далее SOC) и управляется хост-процессором в SOC по модели высокопроизводительного модуля обработки изображений. Для обеспечения простого, но эффективного управления VPU хост-процессором, существует набор регистров, называемых регистрами главного интерфейса. Большинство команд и откликов, передаваемых между хост-процессором и VPU, как правило, проходят через регистры главного интерфейса. С хост-процессора и VPU можно получить прямой доступ к данным потока и некоторым выходным данным изображения. Для более комплексного управления кодеком IP-видео предусмотрен целый набор функций API, которые полностью охватывают весь спектр необходимых операций, осуществляемых хост-процессором.

1.3.6.4.1.1 Функциональные особенности

Блоком VPU поддерживаются все форматы разрешений: вплоть до формата сверхвысокой четкости - 1920 x 1088 или 2048 x 1024 пикселей для режима декодирования и формата высокой четкости - 1920 x 1088 пикселей для режима кодирования, характерных большинству стандартов видеоданных.

VPU обладает рядом инструментов, обеспечивающих устойчивость к ошибкам, а также позволяет осуществлять многократное кодирование или декодирование, или одновременное кодирование-декодирование вместе с сеансом конференцсвязи в полнодуплексном режиме.

VPU легко интегрируется в систему, поскольку для более упрощенного и универсального использования в SoC он привязан к шинной архитектуре AXI и APB. Это относится также к микропрограммному обеспечению блока VPU, запускаемому на программируемом BIT-процессоре. Все функции по управлению обработкой видеоданных и главным интерфейсом

реализуются в данном микропрограммном обеспечении, а это дает удобство программирования, гибкость и легкость при обновлении.

Все функциональные возможности блока VPU далее будут рассмотрены более подробно.

1.3.6.4.1.2 Особенности работы с видеоданными для разных стандартов

<ИНСТРУМЕНТАРИЙ КОДИРОВАНИЯ>

Кодер BP/MP/HP H.264/AVC

Данный кодер отвечает требованиям к стандарту H.264 рекомендаций ITU-T.

Поддерживается Высокий профиль стереоскопического изображения MVC с опросным прогнозированием только для анкерного изображения.

Для оценки параметров движения кодеру необходим только один ссылочный кадр.

Оценка параметров движения проводится с точностью до четвертьпикселя и полупикселя с программируемым поиском в диапазоне до [+/-64, +/-48].

Диапазон поиска перенастраивается в соответствии со следующими 4 горизонтальными и 3 вертикальными диапазонами:

- горизонтальный (-64 ~ 63), горизонтальный (-48 ~ 47), горизонтальный (-32 ~ 31), горизонтальный (-16 ~ 15);
- вертикальный (-48 ~ 47), вертикальный (-32 ~ 31), вертикальный (-16 ~ 15).

Поддерживаются следующие размеры блоков 16 x 16, 16 x 8, 8 x 16 и 8 x 8.

Размеры блоков можно перенастраивать.

Intra-прогнозирование:

- режим яркости I4x4: 9 режимов,
- режим яркости I8x8: 9 режимов,
- режим яркости I16x16: 3 режима (с вертикальным, горизонтальным сканированием, DC),
- режим цветности: 3 режима (с вертикальным, горизонтальным сканированием, DC).

Минимальный размер закодированного изображения равен 96 пикселям по горизонтали и 16 - по вертикали.

Кодером поддерживаются следующие инструменты обеспечения устойчивости к ошибкам: пакетирование видеоданных (фиксированное количество бит и фиксированное количество макроблоков), CIR (циклическая регенерация в моде Intra), а также поддержка многослойной структуры.

Инструменты FMO и ASO H.264 не поддерживаются.

Уровень кадра и уровень макроблока в ходе контроля скорости передачи данных.

Кодирование полукадров может проводиться без PAFF, MBAFF.

Поддерживается CQM (кастомизированная Q-матрица).

Кодер MPEG4-SP

Данный кодер отвечает требованиям к стандарту 14496-2 ISO/IEC.

MV с неограниченным вектором движения.

Поддерживается прогнозирование AC и DC.

Оценка параметров движения проводится с полупиксельной точностью с диапазоном поиска до [+/-64, +/-48].

Диапазон поиска перенастраивается в соответствии со следующими 4 горизонтальными и 3 вертикальными диапазонами:

- горизонтальный (-64 ~ 63), вертикальный (-48 ~ 47),
- горизонтальный (-48 ~ 47), вертикальный (-32 ~ 31),
- горизонтальный (-32 ~ 31), вертикальный (-16 ~ 15),
- горизонтальный (-16 ~ 15).

Кодером поддерживаются следующие инструменты обеспечения устойчивости к ошибкам: маркер ресинхронизации, разбиение данных с реверсивным VLC.

Кодер H.263 P0/P3 (Профиль интерактивной и беспроводной потоковой передачи)

MV с режимом неограниченного вектора движения, удовлетворяющий требованиям Приложения D.

Диапазон поиска соответствует -16 ~ 15 по горизонтали и -16 ~ 15 по вертикали.

Базовый профиль H.263 + Приложение J, K (RS=0 и ASO=0) и T.

<ИНСТРУМЕНТАРИЙ ДЕКОДИРОВАНИЯ>

Декодер H.264/AVC

Данный декодер полностью удовлетворяет требованиям к стандарту H.264 рекомендаций ITU-T в отношении BP, MP и HP.

Поддерживается Высокий профиль стереоскопического изображения MVC.

Обеспечивается поддержка CABAC и CAVLC.

Поддерживаются следующие размеры блоков 16 x 16, 16 x 8, 8 x 16 и 8 x 8, 8 x 4, 4 x 8 и 4 x 4.

Инструменты обнаружения ошибок, маскирования ошибок, а также обеспечения устойчивости к ошибкам с поддержкой FMO и ASO.

Декодер VC-1/WMV-9

Данный декодер поддерживает все функциональные особенности профиля VC-1 - стандарт SMPTE для телевидения: формат сжатия битового видеопотока VC-1 и процесс декодирования.

Поддерживаются Простой, Главный и Расширенный профили.

Поддерживается несколько форматов разрешений (динамического типа), что позволяет достичь нужного значения без проведения масштабирования.

Декодер MPEG-4

Данный декодер полностью отвечает требованиям к стандарту 14496-2 ISO и IEC в отношении SP и ASP, за исключением GMC (общей компенсации движения).

Обеспечивается полная совместимость с форматом XviD.

Поддерживаются короткие заголовки видеоданных.

Декодер DivX

Полностью сертифицированный декодер DivX для профилей Карманное устройство/Портативное устройство/Домашний кинотеатр/Высокое разрешение, используемый как для прогрессивного, так и для чересстрочного декодирования.

Декодер Sorenson Spark

Данный декодер полностью соответствует требованиям к стандарту декодера Sorenson Spark.

Декодер H.263 V2 (Профиль интерактивной и беспроводной потоковой передачи, профиль 3)

Базовый профиль H.263 + Приложение I, J, K (за исключением RS/ASO) и T.

MPEG-1 и MPEG-2

Данный декодер полностью отвечает требованиям к стандарту MPEG2 13818-2 ISO и IEC в отношении Главного профиля.

Поддерживаются I-, P- и B-кадры.

Поддерживается (чересстрочно) закодированное изображение полукадра и закодированное изображение кадра.

Декодер AVS

Поддерживается профиль Jizhun уровня 6.2 (кроме случая 422).

Поддерживается профиль Guangdian уровня 6.2 (кроме случая 422).

Декодер Real Video 10

Данный декодер полностью соответствует стандарту RV-8, 9 и 10, за исключением функции повторной выборки.

Минимальный размер декодированных данных составляет 32x32 пикселей.

Декодер VP8

Данный декодер полностью соответствует требованиям к стандарту декодера VP8.

Поддерживается функция упрощенного и нормального канального деблокирования.

Декодер Theora

Данный декодер полностью соответствует требованиям к стандарту декодера Theora.

1.3.6.4.1.2.1 Функциональные особенности, не связанные с кодеками

Дополнительные функции:

Шумоподавление MPEG-2 и -4;

Вложенный фильтр деблокирования для форматов MPEG-2, -4 и DivX;

Пре- и пост-ротатор/зеркальное отображение.

Возможность программирования

В блок VPU встроен 16-разрядный процессор DSP Chips&Media, предназначенный для обработки битового потока и управления видеоустройством.

Регистры общего назначения и сигнал прерывания предназначены для обеспечения взаимодействия между хост-процессором и кодеком IP-видео.

Оптимальный доступ к внешней памяти

Форматы буферов кадра конфигурируются (линейная карта или тайловая карта распределения) для поддержки более длинных пакетов данных.

Кэш 2D, используемый при компенсации движения, предназначен для сокращения числа обращений к внешней памяти.

Линейный буфер предназначен для размещения ссылочных пикселей оценки параметров движения.

Порт вспомогательной шины AXI для подключения к памяти на кристалле в целях повышения эффективности.

Сигнал, связанный с электропитанием

В качестве дополнительной функции обеспечивается поддержка VPU_IDLE.

1.3.6.4.1.3 Возможность программирования

В блоке VPU находится внутренний процессор DSP, который называется BIT-процессором. BIT-процессор управляет всеми внутренними аппаратными модулями для выполнения определенных операций с видеоданными. Работа BIT-процессора описывается с помощью специального микрокода, называемого BIT-программой. Компания “C&M” уже разработала полный набор BIT-программ, а также полный набор управляющих функций VPU, называемых интерфейсом API VPU. Поэтому разработчикам приложений не нужно тратить время на решение различных проблем, связанных с кодеком на стороне хост-процессора.

Внося изменения в микрокод BIT-процессора, VPU может обеспечивать эффективную поддержку функций, предназначенных для заказчиков или определенных приложений. Многие из прикладных функций C&M уже используются в BIT-программах, но, исходя из потребностей заказчиков, в BIT-программах также могут быть легко реализованы некоторые дополнительные функции.

1.3.6.4.1.3.1 Покадровая обработка

BIT-процессор работает с видеоданными в покадровом режиме, что позволяет снизить уровень зависимости операций VPU в хост-процессоре. В ходе работы с кадром не нужно, чтобы между хост-процессором и VPU осуществлялось взаимодействие. Это ключевой фактор для снижения нагрузки на хост-процессор в процессе работы с видеоданными.

После запуска процесса обработки изображения хост-приложение может выполнять свои собственные операции до того момента, когда оно не будет готово к обработке следующего изображения или пока VPU не передаст сигнал прерывания, информирующий о завершении обработки данного изображения.

1.3.6.4.1.3.2 Управление программной памятью

В блоке VPU есть собственная программная память, предназначенная для загрузки BIT-программ, обеспечивающих поддержку прикладных операций. Чтобы использовать ресурс внутренней памяти наиболее эффективно, в BIT-программу вложена схема динамической перезагрузки, позволяющая сократить размер программной памяти внутри VPU.

Например, когда в VPU идет операция декодирования VC-1, в программную память VPU закачивается микро-ПО декодера VC-1. Однако если запускается операция декодирования H.264/AVC, то BIT-процессор выгружает программу декодирования H.264/AVC из памяти SDRAM и загружает ее в программную память. Ввиду того, что в VPU выполняется покадровая обработка данных, то за одну секунду операцию динамической перезагрузки можно выполнить около 30 раз (максимально) при условии использования одного экземпляра декодера. Поскольку для размещения BIT-программ одного стандарта декодирования требуется менее 16 кбайт памяти, то ни на производительность VPU, ни на полосу пропускания памяти это никак не влияет.

Чтобы сократить частоту динамических перезагрузок, требуется программная память большего объема. Но такое изменение увеличит общую стоимость целевой SOC. Выбирая компромиссное решение в отношении стоимости системы и ее производительности, можно сказать, что данная схема динамического перезапуска может быть очень полезной для заказчиков VPU.

1.3.6.4.1.3.3 Работа с несколькими экземплярами

Блок VPU может работать с несколькими экземплярами, что может быть полезным для приложений многоканального декодирования. Чтобы обеспечить работу системы в режиме с несколькими экземплярами, для каждого экземпляра декодера BIT-процессор использует свой набор внутренних контекстных параметров. Набор данных контекстных параметров создается и обновляется автоматически внутри VPU при создании нового экземпляра и при запуске обработки изображения. Благодаря такой схеме внутреннего контекстного управления, задачи по декодированию, выполняемые процессором, могут управлять работой VPU, независимо от количества собственных экземпляров.

При создании нового экземпляра прикладная задача получает новый указатель, который определяет экземпляр (если в VPU есть доступ к новому указателю). С помощью данного позадачного указателя все последующие операции для данной прикладной задачи смогут выполняться в VPU по-отдельности. При записи драйвера VPU данный указатель можно считать идентификатором устройства или идентификатором порта блока VPU для каждой из задач. Ввиду того, что блок VPU в отдельно взятый момент времени может выполнять только одну задачу по обработке изображения, перед началом каждой новой операции с изображением в каждой прикладной задаче должна проводиться проверка готовности блока VPU.

Завершение одной из прикладных задач по обработке видеоданных в VPU возможно при вызове функции закрытия определенного экземпляра.

1.3.6.4.2 ГЛАВНЫЙ ИНТЕРФЕЙС

1.3.6.4.2.1 Методика управления блоком VPU

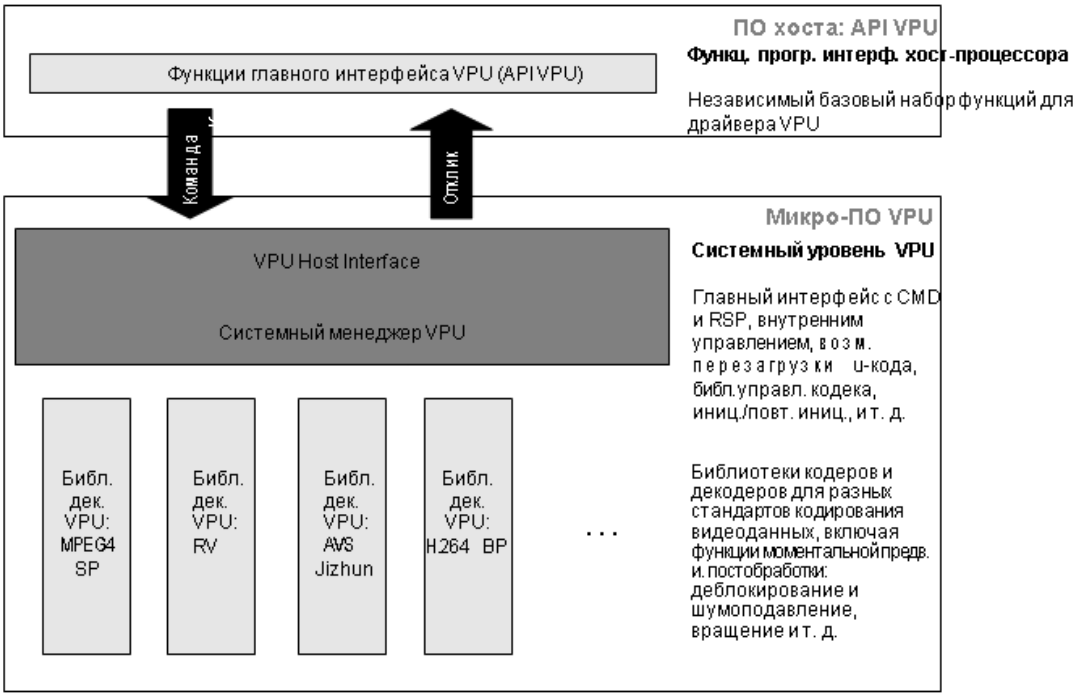
В данной главе приводится общее описание главных интерфейсов для хост-процессора, обеспечивающих управление блоком VPU.

1.3.6.4.2.1.1 Модели взаимодействия

Для обмена данными и/или сообщениями между хост-процессором и VPU в блоке VPU должен быть выделен отдельный путь. В блоке VPU для хранения данных обмена между хост-процессором и VPU используется общий модуль памяти. Доступ к данному общему модулю памяти обеспечивается по основной шине ABMA. Обмен данными битового потока и кадров осуществляется с использованием общего пространства указанного модуля общей памяти.

Обмен сообщениями между хост-процессором и VPU осуществляется по отдельному пути, отличному от пути обмена данными, с помощью набора регистров VPU, называемых регистрами главного интерфейса. Все команды и отклики передаются между хост-процессором и VPU с помощью данных регистров.

На рисунке 1.255 между представлена подробная схема процесса обмена данными и сообщениями между хост-процессором и блоком VPU.



VPU Host Interface – Главный интерфейс VPU
VPU Codec Library – Библиотека кодека VPU

Рисунок 1.255 – Обмен данными и сообщениями между хост-процессором и блоком VPU

Доступ ко всем данным битового потока и изображения осуществляется напрямую с хост-процессора и блока VPU. Обмен информацией по передаче этих данных осуществляется через главный интерфейс посредством команд и откликов. Для этого в главном интерфейсе VPU резервируется набор регистров, доступ к которым осуществляется с хост-процессора. Некоторые из главных регистров используются для обмена актуальными командами и откликами. А остальные регистры используются для передачи к хост-процессору информации о внутреннем состоянии блока VPU.

Компания “C&M” предоставляет набор преопределенных команд и соответствующих откликов на них. Микропрограммное обеспечение, на котором работает BIT-процессор, оптимизировано для данного набора команд и откликов. Хотя с помощью данного набора команд можно работать со всеми основными командами и откликами, при изменении регистров интерфейса, а также частично в код BIT-процессора, можно создавать пользовательские команды и отклики.

1.3.6.4.2.1.2 Обработка данных

Все транзакции, связанные с обработкой выходных данных о пикселях или потоковых данных, осуществляются непосредственно с процессора или из VPU через общее пространство во внешней памяти SDRAM. Чтобы гарантировать безопасность транзакций между хост-процессором и VPU, вся важная информация помещается в регистры главного интерфейса. Обратиться к ним можно в любое время с хост-процессора или блока VPU. Все подобные транзакции, как правило, являются однонаправленными. Это означает то, что только одна из них предназначена для записи данных, а остальные – только для считывания записанных данные в едином буфере данных (например, буфере потоков). Поэтому, пользуясь всего лишь одним указателем считывания и одним указателем записи, можно легко и безопасно управлять всеми этими транзакциями.

Для обработки видеоданных, так же как для размещения общих данных в общем модуле памяти, BIT-процессору требуется определенный объем памяти, который называется рабочим буфером. Доступ к рабочему буферу BIT-процессора осуществляется с блока VPU. Управление буферами кадров, используемыми при декодировании изображений, также как и рабочим буфером, осуществляется только с VPU, что позволяет предотвратить аварийный доступ из хост-приложений и гарантировать безопасность декодирования в блоке VPU.

Для корректной потоковой передачи можно запросить доступ к свободному пространству в буфере потока данных декодера с помощью указателя чтения из буфера, указателя записи в буфер и зная объем буфера. Для данной цели компания “C&M” предлагает набор интерфейсов API, к которым можно обратиться со стороны приложения в любое время.

1.3.6.4.2.2 Регистры главного интерфейса

1.3.6.4.2.2.1 Обзор регистров главного интерфейса

Компанией “C&M” предоставляется набор команд по управлению операциями с видеоданными в покадровом режиме, а также соответствующие отклики на них. В приведенной ниже таблице представлены регистры главного интерфейса, обеспечиваемые BIT-процессором для организации взаимодействия с хост-процессорами.

Таблица 1.256 – Список регистров главного интерфейса

Адрес	Тип	Значение после сброса	Имя	Описание
BASE+0x000	W	0	BIT_CODE_RUN	Запуск BIT-процессора
BASE+0x004 BASE+0x008	W W	0 0	BIT_CODE_DOWN BIT_INT_REQ	Регистр Загружаемые данные для кодирования Запрос в BIT-процессор о прерывании роботы хоста
BASE +0x00C	W	0	BIT_INT_CLEAR	Сброс сигнала прерывания в BIT-процессоре
BASE+0x010	R	0	BIT_INT_STS	Статус прерывания в BIT-процессоре
BASE+0x014 BASE+0x018	W R	0	BIT_CODE_RESET BIT_CUR_PC	Переход программного счетчика BIT-процессора на сброс вектор Текущий PC BIT-процессора
BASE+0x020	R/W	0	BIT_CODEC_BUSY	Кодек BIT-процессора занят
BASE+0x024	W	0	BIT_SW_RESET	Сброс предположения

Продолжение таблицы 1.256

Адрес	Тип	Значение после сброса	Имя	Описание
BASE+0x034	R	0	BIT_SW_RESET_STATUS	Статус сброса
Зарезервирован				
Адрес	Тип	Знач. после сброса	Имя	Описание
BASE+0x100 BASE+0x104	R/W R/W	NA NA	BIT_CODE_BUF_ADDR BIT_WORK_BUF_ADDR	Адрес ячейки памяти SDRAM в таблице КОДОВ Адрес ячейки памяти SDRAM рабочего буфера
BASE+0x108	R/W	NA	BIT_PARA_BUF_ADDR	Адрес ячейки памяти SDRAM буфера аргументов/параметров возвращаемых данных
BASE +0x10C BASE+0x110	R/W R/W	NA NA	BIT_BIT_STREAM_CTRL BIT_FRAME_MEM_CTRL	Управление буфером битового потока Управление модулем памяти кадров
BASE+0x114	R/W	NA	BIT_BIT_STREAM_PARAM	Управление работой декодера
BASE+0x118	R/W	NA	BIT_TEMP_BUF_ADDR	Адрес ячейки памяти SDRAM временного буфера
Зарезервирован				
BASE+0x120 BASE+0x124	R/W R/W	NA NA	BIT_RD_PTR BIT_WR_PTR	Адрес считывания из буфера битового потока Адрес записи в буфер битового потока
BASE+0x128	R/W	0	BIT_RESET_CTRL	Регистр, используемый для выполнения внутренних операций
Зарезервирован				
BASE+0x140	R/W	NA	BIT_AXI_SRAM_USE	Внутренняя конфигурация рабочей области памяти RAM
BASE+0x144 BASE+0x148	R/W R/W	NA NA	BIT_BYTE_POS_FRAME _START BIT_BYTE_POS_FRAME _END	Адрес байта, с которого начинается декодирование изображения Адрес байта, в котором завершается декодирование одного изображения
Зарезервировано				
BASE+0x150 BASE+0x160	R/W R/W	NA NA	BIT_FRM_DIS_FLG BIT_BUSY_FLAG	Флаг, указывающий буфер кадра для текущего экземпляра Флаг, указывающий на занятость процессора
BASE+0x164 BASE+0x168	R/W R/W	NA NA	BIT_RUN_COMMAND BIT_RUN_INDEX	Команда запуска Индекс процесса запуска
BASE +0x16C BASE+0x170	R/W R/W	NA NA	BIT_RUN_COD_STD BIT_INT_ENABLE	Стандарт запущенного кодера Включение прерывания
BASE+0x174 BASE+0x178	R/W R/W	NA NA	BIT_INT_REASON BIT_RUN_AUX_STD	Причина прерывания Регистр индекса дополнительного стандарта кодека
Зарезервирован				
BASE+0x180 ~BASE +0x1DC	R/W	NA	CMD I/O	Регистры ввода и вывода команд

Данные регистра главного интерфейса можно подразделить на три категории.

Регистры управления BIT-процессором

Регистры главного интерфейса данной категории используются для обновления или отображения статуса BIT-процессора для хост-процессоров. Большинство регистров этой категории используются для инициализации BIT-процессора при первоначальном запуске системы.

Глобальные регистры BIT-процессора

Регистры главного интерфейса данной категории используются для хранения всех глобальных переменных, которые никогда не меняются, за исключением случаев, когда меняется действующий экземпляр.

Регистры ввода и вывода команд BIT-процессора

Регистры главного интерфейса данной категории перезаписываются или обновляются каждый раз, когда из хост-процессора поступает новая команда. С помощью этих регистров обрабатываются все команды с входными аргументами и соответствующие отклики с возвращаемыми значениями.

1.3.6.4.2.2.2 Регистры ввода и вывода команд

Среди регистров главного интерфейса регистры ввода и вывода команд (предварительно определенные) применяются для каждой команды по управлению VPU. Выборка регистров ввода и вывода команд представлена в нижеследующей таблице.

Таблица 1.178 – Регистры параметра ENC_SEQ_INIT

ENC_SEQ_INIT				
	Адрес	Тип	Имя	Описание
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_ENC_SEQ_BB_START	Адрес буфера битового потока
	BASE +0x184	R/W	CMD_ENC_SEQ_BB_SIZE	Объем буфера битового потока
	BASE +0x188	R/W	CMD_ENC_SEQ_OPTION	Опция кодирования последовательности a
	BASE +0x18C	R/W	CMD_ENC_SEQ_COD_STD	Стандарт кодирования последовательности
	BASE +0x190	R/W	CMD_ENC_SEQ_SRC_SIZE	Размер исходного кадра, предназначенного для кодирования
	BASE +0x194	R/W	CMD_ENC_SEQ_SRC_F_RATE	Частота исходных кадров последовательности кодирования
	BASE +0x198	R/W	CMD_ENC_SEQ_MP4_PARA	Параметр b формата кодирования MPEG4
	BASE +0x19C	R/W	CMD_ENC_SEQ_263_PARA	Параметр формата кодирования H.263
	BASE +0x1A0	R/W	CMD_ENC_SEQ_264_PARA	Параметр формата кодирования H.264/AVC
	BASE +0x1A4	R/W	CMD_ENC_SEQ_SLICE_MODE	Режим слайсов последовательности кодирования
	BASE +0x1A8	R/W	CMD_ENC_SEQ_GOP_NUM	Количества GOP последовательности кодирования
	BASE +0x1AC	R/W	CMD_ENC_SEQ_RC_PARA	Параметр контроля скорости последовательности кодирования
	BASE +0x1B0	R/W	CMD ENC SEQ RC BUF SIZE	Объем буфера декодера ссылочного кадра последовательности кодирования
	BASE +0x1B4	R/W	CMD_ENC_SEQ_INTRA_REFRESH	Количество регенерированных макроблоков Intra последовательности кодирования
	BASE +0x1C4	R/W	CMD_ENC_SEQ_INTRA_QP	Величина параметра квантизации I-кадра. Данный параметр действует, когда задан 5-й бит регистра CMD_ENC_SEQ_OPTION.
	BASE +0x1C8	R/W	CMD_ENC_SEQ_RC_QP_MAX	Максимальное пороговое значение параметра квантизации для контроля скорости

Продолжение таблицы 1.256

	Адрес	Тип	Имя	Описание
ENC_SEQ_INIT				
	BASE +0x1CC	R/W	CMD_ENC_SEQ_RC_GAMMA	Гамма для получения целевой контрольной суммы для контроля скорости
	BASE +0x1D0	R/W	CMD ENC SEQ RC INTERVAL MODE	Интервальный режим контроля скорости
	BASE +0x1D4	R/W	CMD_ENC_SEQ_INTRA_WEIGHT	Расходно-весовой анализ Intra для уменьшения плотности расположения макроблоков intra
	BASE +0x1D8	R/W	CMD_ENC_SEQ_ME_OPTION	Параметры оценки движения
	BASE +0x1DC	R/W	CMD ENC SEQ RC PARA 2	Параметры контроля скорости
	BASE +0x1E0	R/W	CMD_ENC_SEQ_QP_RANGE_SET	Параметры для задания диапазона параметра квантизации для контроля скорости
	BASE +0x1F0	R/W	CMD ENC SEQ RC MAX INTRA _SIZE	Максимальный размер бита intra-кадра для контроля скорости
	BIT_RUN_CMD(Index, CodecStd, 1)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_ENC_SEQ_SUCCESS	Статус результатов выполнения команд

[a] - В данный регистр добавляется бит ConstIntraQpEn.
[b] - В данный регистр добавляются биты включения HEC и включения Version ID 1.

Таблица 1.257 – Регистры параметра ENC_SEQ_END

ENC_SEQ_END				
	Адрес	Тип	Имя	Описание
ВХОДНОЙ АРГУМЕНТ	BIT_RUN_CMD (Index, CodecStd, 2)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_ENC_SEQ_END_SUCCESS	Результат выполнения команды ENC_SEQ_END

Таблица 1.258 – Регистры параметра DEC_SEQ_INIT

DEC_SEQ_INIT				
	Адрес	Тип	Имя	Описание
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_DEC_SEQ_BB_START	Адрес буфера битового потока
	BASE +0x184	R/W	CMD_DEC_SEQ_BB_SIZE	Объем буфера битового потока
	BASE +0x188	R/W	CMD_DEC_SEQ_OPTION	Опция кодирования последовательности
	BASE +0x194	R/W	CMD DEC SEQ USER DATA OPTION	Опция выдачи данных пользователя, относительно заголовка последовательности (только для MPEG2)
	BASE	R/W	CMD DEC SEQ MP4 ASP CLASS	Класс MP4 - ASP
	+0x19C	R/W	CMD DEC SEQ X264 MV EN	H.264/AVC – Включение вектора движения более высокого уровня в X.264
		R/W	CMD DEC SEQ VC1 STREAM _FMT	VC1 - Формат потока
	BASE +0x1AC	R/W	CMD DEC SEQ USER DATA BASE_ADDR	Адрес пользовательских данных, по которому из VPU передаются данные пользователя
	BASE +0x1B0	R/W	CMD DEC SEQ USER DATA BUF_SIZE	Объем буфера пользовательских данных

Продолжение таблицы 1.258

DEC_SEQ_INIT				
	Адрес	Тип	Имя	Описание
	BIT_RUN_CMD(Index, DecoderStd, 1)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x180	R	RET DEC SEQ MP2 BAR LEFT RIGHT	Информация о пользовательских данных ATSC MPEG-2
	BASE +0x184	R	RET DEC SEQ MP2 BAR TOP BOTTOM	Информация о пользовательских данных ATSC MPEG-2
DEC_SEQ_INIT				
	BASE +0x18C	R	RET_DEC_SEQ_VUI_INFO	Информация из VUI H.264/AVC
	BASE +0x190	R	RET DEC SEQ VUI PIC STRUCT	Информация pic_struct из VUI H.264/AVC
	BASE +0x1B4	R	RET_DEC_SEQ_BIT_RATE	Битовая скорость закодированных исходных данных
	BASE +0x1B8	R	RET_DEC_SEQ_EXT_INFO	Расширенные данные о последовательности MPEG-2
	BASE +0x1C0	R	RET_DEC_SEQ_SUCCESS	Статус результата выполнения команды
	BASE +0x1C4	R	RET_DEC_SEQ_SRC_SIZE	Размер декодированного изображения. Действительный размер декодированных изображений.
	BASE +0x1C8	R	RET_DEC_SEQ_ASPECT	Зрительное отношение декодированных исходных данных
	BASE +0x1CC	R	RET_DEC_SEQ_FRAME_NEED	Буфер декодированного кадра минимально требуемого размера
	BASE +0x1D0	R	RET_DEC_SEQ_FRAME_DELAY	Максимальная длительность задержки в буфере отображаемого кадра
	BASE +0x1D4	R	RET_DEC_SEQ_INFO	MPEG4 – информация о декодированной последовательности
			RET_DEC_SEQ_VP8_SCALE_INFO	Информация о масштабе VP8 для процесса увеличения разрешения вне канала декодирования
	BASE +0x1D8	R	RET DEC SEQ CROP LEFT RIGHT	Информация AVC об обрезке изображения слева и справа
	BASE +0x1DC	R	RET DEC SEQ CROP TOP BOTTOM	Информация AVC об обрезке изображения сверху и снизу
	BASE +0x1E0	R	RET_DEC_SEQ_SEQ_ERR_REASON	Указание причины возникновения ошибки при инициализации последовательности
	BASE +0x1E4	R	RET_DEC_SEQ_FRATE_NR	Числитель дроби в выражении частоты закодированного кадра
	BASE +0x1E8	R	RET_DEC_SEQ_FRATE_DR	Знаменатель дроби в выражении частоты закодированного кадра
	BASE +0x1EC	R	RET_DEC_SEQ_HEADER_REPORT	Информация о заголовке последовательности, уровне профиля, чередовании и т. д.

Таблица 1.259 – Регистры параметров DEC_SEQ_END

DEC_SEQ_END				
	Адрес	Тип	Имя	Описание
ВХОДНОЙ АРГУМЕНТ	BIT_RUN_CMD(Index, CodecStd, 2)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_DEC_SEQ_END_SUCCESS	Результат выполнения команды DEC_SEQ_END

Таблица 1.260 – Регистры параметра ENC_PIC_RUN

ENC_PIC_RUN				
	Адрес	Тип	Имя	Описание
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_ENC_PIC_SRC_INDEX	Индекс буфера исходного кадра
	BASE +0x184	R/W	CMD_ENC_PIC_SRC_STRIDE	Шаг по индексу буфера исходного кадра
	BASE +0x18C	R/W	CMD_ENC_PIC_QS	Шаг квантизации при кодировании изображения
	BASE +0x190	R/W	CMD_ENC_PIC_SRC_MODE	Режим загрузки исходных кадров
	BASE +0x194	R/W	CMD_ENC_PIC_OPTION	Опция кодирования изображения
	BASE +0x198	R/W	CMD_ENC_PIC_BB_START	Стартовый адрес буфера потока изображений
	BASE +0x19C	R/W	CMD_ENC_PIC_BB_SIZE	Объем буфера потока изображений
	BASE +0x1A0	R/W	CMD ENC PIC PARA BASE ADDR	Базовый адрес ячейки внешней памяти буфера параметров
	BASE +0x1A8	R/W	CMD_ENC_PIC_SRC_ADDR_Y	Адрес буфера исходного кадра для компоненты яркости
	BASE +0x1AC	R/W	CMD_ENC_PIC_SRC_ADDR_CB	Адрес буфера исходного кадра для компоненты Cb
	BASE +0x1B0	R/W	CMD_ENC_PIC_SRC_ADDR_CR	Адрес буфера исходного кадра для компоненты Cr
	BASE +0x1E8	R/W	CMD_ENC_PIC_SRC_BOTTOM_Y	Адрес буфера нижнего исходного полукадра для компоненты Y.
	BASE +0x1EC	R/W	CMD_ENC_PIC_SRC_BOTTOM_CB	Адрес буфера нижнего исходного полукадра для компоненты Cb.
	BASE +0x1F0	R/W	CMD_ENC_PIC_SRC_BOTTOM_CR	Адрес буфера нижнего исходного полукадра для компоненты Cr.
	BIT_RUN_CMD(Index, CodecStd, 3)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_ENC_PIC_FRAME_NUM	Количество закодированных кадров
	BASE +0x1C4	R	RET_ENC_PIC_TYPE	Тип закодированного изображения
	BASE +0x1C8	R	RET_ENC_PIC_FRAME_IDX	Индекс восстановленного кадра
	BASE +0x1CC	R	RET_ENC_PIC_SLICE_NUM	Количество закодированных слайсов изображения
	BASE +0x1D0	R	RET_ENC_PIC_FLAG	Проверки BitstreamBufferSize достаточно для того, чтобы сохранить данные одного кадра при сокращении диапазона в режиме BufPicReset
	BASE +0x1D8	R	RET_ENC_PIC_SUCCESS	Статус результатов выполнения команд

Таблица 1.261 – Регистры параметра DEC_PIC_RUN

DEC_PIC_RUN					
	Адрес	Тип	Имя	Описание	
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_DEC_PIC_ROT_MODE	Опции шумоподавления и вращения	
				Примечание	Опция шумоподавления поддерживается только для MPEG-2 и -4
	BASE +0x184	R/W	CMD_DEC_PIC_ROT_INDEX	Индекс кадра после вращения	
	BASE +0x188	R/W	CMD_DEC_PIC_ROT_ADDR_Y	Адрес кадра повернутого изображения для компоненты яркости	
	BASE +0x18C	R/W	CMD_DEC_PIC_ROT_ADDR_CB	Адрес кадра повернутого изображения для компоненты Cb	
	BASE +0x190	R/W	CMD_DEC_PIC_ROT_ADDR_CR	Адрес кадра повернутого изображения для компоненты Cr	
	BASE +0x194	R/W	CMD_DEC_PIC_OPTION	Опция декодера	
	BASE +0x198	R/W	CMD_DEC_FRAME_SKIP_NUM	Количество пропущенных кадров	

Продолжение таблицы 1.261

DEC_PIC_RUN				
	Адрес	Тип	Имя	Описание
	BASE +0x1AC	R/W	CMD DEC PIC USER DATA BASE_ADDR	Адрес пользовательских данных, по которому из VPU передаются данные пользователя
	BASE +0x1B0	R/W	CMD DEC PIC USER DATA BUF_SIZE	Объем буфера пользовательских данных
	BASE +0x1B4	R/W	CMD_DEC_PIC_NUM_ROWS	Количество рядов макроблоков
	BASE +0x1B8	R/W	CMD_DEC_PIC_ROT_STRIDE	Шаг по индексу кадра повернутого изображения
	BASE +0x1E8	R/W	CMD_DEC_PIC_ROT_BOTTOM_Y	Адрес буфера нижнего полукадра для повернутого выходного изображения и/или выходного изображения с шумоподавлением по компоненте Y.
	BASE +0x1EC	R/W	CMD_DEC_PIC_ROT_BOTTOM_CB	Адрес буфера нижнего полукадра для повернутого выходного изображения и/или выходного изображения с шумоподавлением по компоненте Cb.
	BASE +0x1F0	R/W	CMD_DEC_PIC_ROT_BOTTOM_CR	Адрес буфера нижнего полукадра для повернутого выходного изображения и/или выходного изображения с шумоподавлением по компоненте Cr.
	BIT_RUN_CMD(Index, DecoderStd, 3)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x180	R	RET DEC PIC MP2 BAR LEFT RIGHT	Информация о пользовательских данных ATSC MPEG-2
	BASE +0x184	R	RET DEC PIC MP2 BAR TOP BOTTOM	Информация о пользовательских данных ATSC MPEG-2
	BASE +0x19C	R	RET DEC PIC AVC FPA SEI0	Информация SEI о расположении упакованных кадров AVC
			RET_DEC_PIC_MP2_OFFSET1	MPEG-2 frame_centre_offset
	BASE +0x1A0	R	RET DEC PIC AVC FPA SEI1	Информация SEI о расположении упакованных кадров AVC
			RET_DEC_PIC_MP2_OFFSET2	MPEG-2 frame_centre_offset
	BASE +0x1A4	R	RET DEC PIC AVC FPA SEI2	Информация SEI о расположении упакованных кадров AVC
			RET_DEC_PIC_MP2_OFFSET3	MPEG-2 frame_centre_offset
	BASE +0x1A8	R	RET_DEC_PIC_MP2_OFFSET_NUM	Количество frame_centre_offset для frame_centre_offset для MPEG-2
	BASE +0x1B4	R	RET_DEC_NUM_MB_ROWS	Отчет о рядах закодированных макроблоков текущего изображения
	BASE +0x1B8	R	RET_DEC_PIC_HRD_INFO	Отчет с информацией о HRD H.264/AVC
	BASE +0x1BC	R	RET_DEC_PIC_SIZE	Разрешение декодированного изображения
	BASE +0x1C0	R	RET_DEC_PIC_FRAME_NUM	Количество декодированных кадров
	BASE +0x1C4	R	RET_DEC_PIC_DISPLAY_IDX	Индекс отображаемого кадра
	BASE +0x1C8	R	RET_DEC_PIC_ERR_MB	Количество макроблоков с ошибками в декодированном изображении
	BASE +0x1CC	R	RET_DEC_PIC_TYPE	Тип декодированного изображения
	BASE +0x1D0	R	RET_DEC_PIC_POST	Информация о мультиразрешении и сокращении диапазона
			RET DEC PIC SEQ EXT INFO	Информация о расширении последовательности MPEG-2
	BASE +0x1D8	R	RET_DEC_PIC_SUCCESS	Статус результата выполнения команды а
	BASE +0x1DC	R	RET_DEC_PIC_DECODED_IDX	Индекс декодированного кадра
	BASE +0x1E0	R	RET_DEC_PIC_CROP_LEFT_RIGHT	Информация об обрезке кадра (слева и справа) H.264/AVC
		R	RET DEC PIC MODULO TIME BASE	modulo_time_base в заголовке VOP

Продолжение таблицы 1.261

DEC_PIC_RUN				
	Адрес	Тип	Имя	Описание
	BASE +0x1E4	R	RET DEC PIC CROP TOP BOTTOM	Информация об обрезке кадра (сверху и снизу) H.264/AVC
			RET DEC PIC VOP TIME INCREMENT	vop_time_increment в заголовке VOP
	BASE +0x1E8	R	RET DEC PIC VP8 PIC REPORT	Справочная информация и информация об изображении VP8
			RET DEC PIC ATSC USER _DATA_INFO	active_format в пользовательских данных AFD для H.264, AVC и MPEG-2
	BASE +1x1EC	R	RET_DEC_PIC_VUI_INFO	Информация VUI H.264/AVC
	BASE +0x1F0	R	RET_DEC_PIC_ASPECT	Зрительное отношение декодированного изображения
	BASE +0x1F0	R	RET_DEC_PIC_VP8_SCALE_INFO	Информация о масштабе VP8 для увеличения разрешения вне канала декодирования
	BASE +0x1F4	R	RET_DEC_PIC_FRATE_NR	Числитель дроби в выражении частоты декодированных кадров
	BASE +0x1F8	R	RET_DEC_PIC_FRATE_DR	Знаменатель дроби в выражении частоты декодированных кадров
	BASE +0x1D0	R	RET_DEC_PIC_MVC_REPORT	Информация об изображении MVC

[a] - В данный регистр добавляются биты для передачи данных о недостатке места в буфере хранения СЛОЕВ/PPS/SPS.

Таблица 1.262 – Регистр параметра SET_FRAME_BUFFER

SET_FRAME_BUF				
	Адрес	Тип	Имя	Описание
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_SET_FRAME_BUFNUM	Количество буферов кадра, используемых декодером
	BASE +0x184	R/W	CMD_SET_FRAME_BUF STRIDE	Шаг по индексу строк буфера кадра
	BASE +0x188	R/W	CMD SET FRAME SLICE BB START	Стартовый адрес буфера байтов для хранения SLICE RBSP.
			CMD_SET_FRAME_SUBSAMP_A	Базовый адрес буфера A для хранения изображений с увеличенным разрешением (только для кодеров)
	BASE +0x18C	R/W	CMD SET FRAME SLICE BB SIZE	Объем буфера для хранения слайсов
			CMD_SET_FRAME_SUBSAMP_B	Базовый адрес буфера B для хранения изображений с увеличенным разрешением (только для кодеров)
	BASE +0x190	R/W	CMD SET FRAME AXI BIT ADDR	Базовый адрес данных BIT-процессора по вспомогательному AXI
	BASE +0x194	R/W	CMD SET FRAME AXI IPACDC _ADDR	Базовый адрес доступа к AcDc и IP по вспомогательному AXI
	BASE +0x198	R/W	CMD SET FRAME AXI DBKY _ADDR	Базовый адрес доступа к опции деблокирования в режиме яркости по вспомогательному AXI
	BASE +0x19C	R/W	CMD SET FRAME AXI DBKC _ADDR	Базовый адрес доступа к опции деблокирования в режиме цветности по вспомогательному AXI
	BASE +0x1A0	R/W	CMD SET FRAME AXI OVL _ADDR	Базовый адрес доступа к опции перекрывания (только для VC-1) по вспомогательному AXI

Продолжение таблицы 1.262

SET_FRAME_BUF				
	Адрес	Тип	Имя	Описание
	BASE +0x1A4	R/W	CMD SET FRAME AXI BTP ADDR	Базовый адрес доступа к опции битовой карты (только для VC-1) по вспомогательному AXI
	BASE +0x1AC	R/W	CMD SET FRAME CACHE CONFIG	Конфигурация кэш-памяти
	BASE +0x1B0	R/W	CMD SET FRAME MB BUF _BASE	Буфер для хранения первой части декодированной информации VP8
		R/W	CMD SET FRAME SUBSAMP _MVC_A	Базовый адрес буфера A для хранения изображений с увеличенным разрешением (только для кодеров MVC)
	BASE +0x1B4	R/W	CMD SET FRAME SUBSAMP _MVC_B	Базовый адрес буфера B для хранения изображений с увеличенным разрешением (только для кодеров MVC)
	BASE +0x1B8	R/W	CMD SET FRAME MAX DEC _SIZE	Максимальный размер декодированного изображения
	BASE +0x1BC	R/W	CMD_SET_FRAME_DELAY	Величина задержки отображения (только для H.264/AVC)
	BIT_RUN_CMD(Index, DecoderStd, 4)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_SET_FRAME_SUCCESS	Статус результатов выполнения команды

Таблица 1.263 – Регистр параметра ENC_HEADER

ENC_HEADER				
	Адрес	Тип	Имя	Описание
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_ENC_HEADER_CODE	Код заголовка, подлежащего кодированию
	BASE +0x184	R/W	CMD_ENC_HEADER_BB_START	Стартовый адрес буфера потока синтаксисов заголовка
	BASE +0x188	R/W	CMD_ENC_HEADER_BB_SIZE	Объем буфера потока синтаксисов заголовка
	BASE +0x18C	R/W	CMD ENC HEADER FRAME CROP_H	Количество сэмплов области обрезки в строке
	BASE +0x190	R/W	CMD ENC HEADER FRAME CROP_V	Количество сэмплов области обрезки в столбце изображения
	BASE +0x194	R/W	CMD ENC HEADER CABAC _MODE	Метод энтропийного кодирования
	BASE +0x198	R/W	CMD ENC HEADER CABAC _INIT_IDC	Индекс, указывающий таблицу инициализации, используемую в процессе инициализации для CABAC
	BASE +0x19C	R/W	CMD ENC HEADER TRANSFORM _8X8	Intra-прогнозирование 8x8
	BASE +0x1A0	R/W	CMD ENC HEADER CHROMA _FORMAT	Формат цветности
	BASE +0x1A4	R/W	CMD ENC HEADER FIELD _FLAG	Прогрессивное или чересстрочное кодирование
	BASE +0x1A8	R/W	CMD_ENC_HEADER_PROFILE	Параметр profile_idc
	BASE +0x1AC	R/W	CMD ENC HEADER SCALING _LIST	Флаг наличия списка/матрицы масштабирования
	BASE +0x1B0	R/W	CMD ENC HEADER QMAT _BASE	База параметров списка масштабирования
	BIT_RUN_CMD(Index, CodecStd, 5)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_ENC_HEADER_SUCCESS	Статус результата выполнения команды

Таблица 1.264 – Регистр параметра ENC_PARA_SET

ENC_PARA _SET				
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_ENC_PARA_SET_TYPE	Тип набора параметров изображения и последовательности
	BASE +0x194	R/W	CMD ENC PARA SET CABAC _MODE	Метод энтропийного кодирования
	BASE +0x198	R/W	CMD ENC PARA SET CABAC _INIT_IDC	Индекс, указывающий таблицу инициализации, используемую в процессе инициализации для CABAC
	BASE +0x19C	R/W	CMD ENC PARA SET TRANSFORM_8X8	Intra-прогнозирование 8x8
	BASE +0x1A0	R/W	CMD ENC PARA SET CHROMA _FORMAT	Формат цветности
	BASE +0x1A4	R/W	CMD ENC PARA SET FIELD _FLAG	Прогрессивное или чересстрочное кодирование
	BASE +0x1A8	R/W	CMD_ENC_PARA_SET_ PROFILE	Параметр profile_idc
	BASE +0x1AC	R/W	CMD ENC PARA SET SCALING_LIST	Флаг наличия списка/матрицы масштабирования
	BASE +0x1B0	R/W	CMD ENC PARA SET QMAT _BASE	База параметров списка масштабирования
	BIT_RUN_CMD(Index, CodecStd, 6)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_ENC_PARA_SET_SIZE	Длина байта RBSP набора параметров закодированного изображения и последовательности
	BASE +0x1C4	R	RET_ENC_PARA_SET_SUCCESS	Статус результата выполнения команды

Таблица 1.265 – Регистр параметра DEC_PARA_SET

DEC_PARA_SET				
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_DEC_PARA_SET_TYPE	Тип набора параметров изображения и последовательности
	BASE +0x184	R/W	CMD_DEC_PARA_SET_SIZE	Длина байта RBSP в наборе параметров закодированного изображения и последовательности
	BIT_RUN_CMD(Index, DecoderStd, 7)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_DEC_PARA_SET_SUCCESS	Статус результата выполнения команды

Таблица 1.266 – Регистра параметра DEC_BUF_FLUSH

DEC_BUF_FLUSH				
ВХОДНОЙ АРГУМЕНТ	OTCYTCB.			
	BIT_RUN_CMD(Index, DecoderStd, 8)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_DEC_BUF_FLUSH_SUCCES S	Статус результата выполнения команды

Таблица 1.267 – Регистр параметра ENC_PARAM_CHANGE

ENC_PARAM_CHANGE				
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD ENC PARAM CHANGE ENABLE	Регистр разрешения изменения параметров кодера
	BASE +0x184	R/W	CMD ENC PARAM CHANGE GOP_NUM	Количество новых GOP кодера
	BASE +0x188	R/W	CMD ENC PARAM CHANGE INTRA_QP	Параметр квантования новых данных кодера в моде Intra
	BASE +0x18C	R/W	CMD ENC PARAM CHANGE BI-TRATE	Скорость нового битового потока кодера
	BASE +0x190	R/W	CMD ENC PARAM CHANGE F_RATE	Частота нового кадра кодера
	BASE +0x194	R/W	CMD ENC PARAM CHANGE IN-TRA_ REFRESH	Количество новых восстановленных макроблоков кодера в моде Intra
	BASE +0x19C	R/W	CMD ENC PARAM CHANGE _HEC_MODE	Режим разрешения нового Hec декодера
	BIT_RUN_CMD(Index, CodecStd, 9)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET ENC PARAM CHANGE _SUCCESS	Статус результата выполнения команды

Таблица 1.268 – Регистр параметра ENC_ROI_INIT

ENC_ROI_INIT				
ВХОДНОЙ АРГУМЕНТ	BASE +0x180	R/W	CMD_ENC_ROI_MODE	Включение режима Roi
	BASE +0x184	R/W	CMD_ENC_ROI_NUM	Общее количество ROI
	BASE +0x188	R/W	CMD_ENC_ROI_POS_0	Позиция ROI_0
	BASE +0x18C	R/W	CMD_ENC_ROI_QP_0	Параметр квантизации ROI_0
ENC_ROI_INIT				
	BASE +0x190	R/W	CMD_ENC_ROI_POS_1	Позиция ROI_1
	BASE +0x194	R/W	CMD_ENC_ROI_QP_1	Параметр квантизации ROI_1
	BASE +0x198	R/W	CMD_ENC_ROI_POS_2	Позиция ROI_2
	BASE +0x19C	R/W	CMD_ENC_ROI_QP_2	Параметр квантизации ROI_2
	BASE +0x1A0	R/W	CMD_ENC_ROI_POS_3	Позиция ROI_3
	BASE +0x1A4	R/W	CMD_ENC_ROI_QP_3	Параметр квантизации ROI_3
	BASE +0x1A8	R/W	CMD_ENC_ROI_POS_4	Позиция ROI_4
	BASE +0x1AC	R/W	CMD_ENC_ROI_QP_4	Параметр квантизации ROI_4
	BASE +0x1B0	R/W	CMD_ENC_ROI_POS_5	Позиция ROI_5
	BASE +0x1B4	R/W	CMD_ENC_ROI_QP_5	Параметр квантизации ROI_5
	BASE +0x1B8	R/W	CMD_ENC_ROI_POS_6	Позиция ROI_6
	BASE +0x1BC	R/W	CMD_ENC_ROI_QP_6	Параметр квантизации ROI_6
	BASE +0x1C0	R/W	CMD_ENC_ROI_POS_7	Позиция ROI_7
	BASE +0x1C4	R/W	CMD_ENC_ROI_QP_7	Параметр квантизации ROI_7
	BASE +0x1C8	R/W	CMD_ENC_ROI_POS_8	Позиция ROI_8
	BASE +0x1CC	R/W	CMD_ENC_ROI_QP_8	Параметр квантизации ROI_8
	BASE +0x1D0	R/W	CMD_ENC_ROI_POS_9	Позиция ROI_9
	BASE +0x1D4	R/W	CMD_ENC_ROI_QP_9	Параметр квантизации ROI_9
	BIT_RUN_CMD(Index, CodecStd, 12)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1D8	R	RET_ENC_ROI_SUCCESS	Статус результата выполнения команды

Таблица 1.269 – Регистр параметра GET_FW_VERSION

GET_FW_VERSION				
ВХОДНОЙ АРГУМЕНТ	ОТСУТСТВ.			
	BIT_RUN_CMD(Index, DecoderStd, 15)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_FW_VER_NUM	Версия микро-ПО с идентификатором продукта и идентификатором версии
	BASE +0x1C4	R	RET_FW_CODE_REV	Количество переработанных версий микро-ПО

Таблица 1.270 – Регистр параметра SLEEP/WAKE

GET_FW_VERSION				
АРГУМЕНТ ВОДА	ОТСУТСТВ.			
	BIT_RUN_CMD(Index, CodecStd, 10/11)			
ВЫХОДНОЙ РЕЗУЛЬТАТ	BASE +0x1C0	R	RET_SLEEP_WAKE_SUCCESS	Статус результата выполнения команды

1.3.6.4.2.3 Подробное описание регистров

В данном подразделе приводится подробное описание регистров главного интерфейса. В нем более детально рассматривается значение каждого битового полукадра регистров главного интерфейса.

1.3.6.4.2.3.1 Регистры управления

Таблица 1.271 – BIT_CODE_RUN (0x000)

Бит	Имя	Тип	Функция	Значение после сброса
0	BIT_CODE_RUN	W	0 – Выполнение останова BIT-процессора 1 - Выполнение запуска BIT-процессора	0

Таблица 1.272 – BIT_CODE_DOWN (0x004)

Бит	Имя	Тип	Функция	Значение после сброса
15:0	CodeData	W	Данные о загрузке 16-битного BIT-кода	0
28:16	CodeAddr	W	Адрес загрузки 13-битного BIT-кода, адрес слова BIT-кода (16-битный адрес) – В данной архитектуре предусматривается объем памяти для хранения кодовых слов, равный 6 K (12 кбайт). Поэтому CodeAddr[12:0] не должен превышать 6144.	

Таблица 1.273 – BIT_INT_REQ (0x008)

Бит	Имя	Тип Функция	Значение после сброса
0	IntReq	W Запрос прерывания передается в BIT-процессор. Для передачи запроса на прерывание в BIT-процессор со стороны хоста в данный регистр записывается 1. *В текущей версии микро-ПО возможность передачи сигнала прерывания со стороны хоста в BIT-процессор не поддерживается. Поэтому данный бит не используется.	0

Таблица 1.274 – BIT_INT_CLEAR (0x00C)

Бит	Имя	Тип Функция	Значение после сброса
0	IntClear	W Для сброса сигнала прерывания BIT-процессора на стороне хоста в данный регистр заносится 1. *Примечание - Если прерывание сбрасывается с использованием обоих регистров BIT_INT_REASON и BIT_INT_CLEAR, то для корректной работы после установления регистра BIT_INT_REASON в 0 регистр.*	0

Таблица 1.275 - BIT_INT_STS (0x010)

Бит	Имя	Тип Функция	Значение после сброса
0	IntSts	R 1 обозначает то, что с BIT-процессора на хост поступает предположение о прерывании. Данный бит сбрасывается, когда на стороне хоста в регистр BitIntClear будет записана 1.	0

Таблица 1.276 - BIT_CODE_RESET (0x014)

Бит	Имя	Тип Функция	Значение после сброса
0	CodeReset	W Счетчик программы BIT-процессора переходит к вектору сброса без какой-либо перенастройки аппаратной логики. После этого перехода данный регистр автоматически Предупреждение - При использовании данного регистра будьте очень внимательны.	0

Таблица 1.277 - BIT_CUR_PC (0x018)

Бит	Имя	Тип Функция	Значение после сброса
13:0	CurPc	R Текущий счетчик программы BIT-процессора. Данный регистр может использоваться только для отладки	0

Таблица 1.278 - BIT_CODEC_BUSY (0x020)

Бит	Имя	Тип Функция	Значение после сброса
0	CodecBusy	R/W Флаг занятости текущего кодека. На BIT-процессоре при запуске кодека в данный регистр записывается 1. 0 обозначает то, что BIT-процессор ожидает получение команды. Данное значение связано с o_vpu_idle.	0

Таблица 1.279 - BIT_SW_RESET (0x024)

Бит	Имя	Тип	Функция	Значение после сброса
2:0	Зарезервировано	-	НЕ УСТАНАВЛИВАЙТЕ В 1.	
3	RESET_BPU_CORE	W	Предположение о сбросе в BPU в области синхронизации ядра.	
4	RESET_BPU_BUS	W	Предположение о сбросе в BPU в области синхронизации шины.	
5	RESET_VCE_CORE	W	Предположение о сбросе в VCE в области синхронизации ядра.	
6	RESET_VCE_BUS	W	Предположение о сбросе в VCE в области синхронизации шины.	
7	RESET_GDI_CORE	W	Предположение о сбросе в GDI в области синхронизации ядра.	
8	RESET_GDI_BUS	W	Предположение о сбросе в VCE в области синхронизации шины.	
31:9	Зарезервировано	-	- Предупреждение - При использовании данного регистра будьте очень внимательны. Подходящий код можно найти в драйвере API VPU.	

Таблица 1.280 - BIT_SW_RESET_STATUS (0x034)

Бит	Имя	Тип	Функция	Значение после сброса
0	RST_STATE_BPU_CORE	R	Статус режима сброса, заявляемый в RESET_BPU_CORE	0
1	RST_STATE_BPU_BUS	R	Статус режима сброса, заявляемый в RESET_BPU_BUS	
2	RST_STATE_VCE_CORE	R	Статус режима сброса, заявляемый в RESET_VCE_CORE	
3	RST_STATE_VCE_BUS	R	Статус режима сброса, заявляемый в RESET_VCE_BUS	
4	RST_STATE_GDI_CORE	R	Статус режима сброса, заявляемый в RESET_GDI_CORE	
5	RST_STATE_GDI_BIS	R	Статус режима сброса, заявляемый в RESET_GDI_BUS Флаги данного регистра, устанавливаются сразу после того, как будет установлен регистр RESET_XXX. При сбросе данное значение остается равным 1. Перед входом в нормальный режим на хост-процессоре должна быть проведена проверка значения данного регистра. Предупреждение - При использовании данного регистра будьте очень внимательны.	

1.3.6.4.2.3.2 Глобальные регистры

Таблица 1.281 - BIT_CODE_BUF_ADDR (0x100)

Бит	Имя	Тип Функция	Значение после сброса
31:0	CodeBufAddr	R/W Адрес стартового байта закодированного изображения в микро-ПО BIT-процессора, который располагается в памяти SDRAM. Перед запуском BIT-процессора на стороне хоста в данном регистре необходимо задать адрес стартового байта SDRAM закодированного изображения в BIT-процессоре. Уточнить размер закодированного изображения в BIT-процессоре можно в предоставляемом справочном ПО к API.	N/A

Таблица 1.282 - BIT_WORK_BUF_ADDR (0x104)

Бит	Имя	Тип Функция	Значение после сброса
31:0	WorkBufAddr	R/W Адрес бита памяти SDRAM рабочего буфера BIT-процессора. На стороне хоста необходимо зарезервировать рабочий буфер для кодирования и декодирования данных в BIT-процессоре. Данное значение должно быть выровнено до 1024. Объем рабочего буфера в текущей архитектуре для каждого экземпляра можно узнать из справочного ПО к API.	N/A

Таблица 1.283 – BIT_PARA_BUF_ADDR (0x108)

Бит	Имя	Тип Функция	Значение после сброса
31:0	ParaBufAddr	R/W Адрес байта памяти SDRAM буфера параметров в BIT-процессоре. На стороне хоста необходимо зарезервировать буфер параметра в SDRAM для аргумента выполнения команд в BIT-процессоре и возвращаемых данных. Внутри буфера параметров хранятся следующие типы адресов буферов с указанием их диапазона: 0 - 383: адреса буферов кадра 382 - 511: адреса совмещенных буферов mv 512 - 895: адреса нижних ячеек буферов кадров (при использовании тайловой карты распределения) 896 - 1279: адрес верхней ячейки буфера кадра WTL (если включена функция WTL) 1280: адрес нижней ячейки буфера кадра WTL (если включена функция WTL)	N/A

Таблица 1.284 – BIT_BIT_STREAM_CTRL (0x10C)

Бит	Имя	Тип	Функция	Значение после сброса
0	SelBigEndian	R/W	0 – буфер битового потока имеет прямой порядок байтов 1 - буфер битового потока имеет обратный порядок байтов	N/A
1	Sel64BitsEndian	R/W	0 – буфер битового потока имеет 64-битный порядок следования 1 - буфер битового потока имеет 32-битный порядок следования	
2	BufStsCheckDis	R/W	ВЫШЛО ИЗ УПОТРЕБЛЕНИЯ: пользоваться данным флагом не рекомендуется 0 – включена проверка переполнения и недозагрузки буфера битового потока 1 – отключена проверка переполнения и недозагрузки буфера битового потока Если буфер битового потока в режиме декодирования недозагружен, то в BIT-процессоре останавливается загрузка битового потока. Если данный флаг установлен в 1, то проверка статуса переполнения и недозагрузки буфера битового потока в BIT-процессоре не проводится.	
4	EncBufPicReset	R/W	В режиме декодирования данный флаг игнорируется. Значение 1 показывает то, что данный буфер битового потока сбрасывается при каждой команде кодирования изображения. В режиме кодирования после кодирования одного изображения, буфер битового потока сбрасывается во внешнюю память SDRAM, и закодированные данные следующего изображения перезаписываются на начало буфера битового потока. Соответственно, по окончании кодирования каждого изображения в хост должны поступать закодированные данные.	
5	EncDynBufAllocEn	R/W	Разрешение на размещение буфера потоков динамических изображений в кодере. Чтобы воспользоваться данной опцией, необходимо также разрешить EncBufPicReset. В противном случае данное значение игнорируется. Если опция включена, то при кодировании каждого изображения в кодере в динамическом режиме размещается буфер потока. Данная опция используется в случае, если необходимо повысить эффективность буферизации потока закодированных изображений.	
6	EncLineBufIntEn	R/W	Разрешение прерывания в режиме линейного буфера. 0 – прерывание запрещено 1 – прерывание разрешено	

Таблица 1.285 – BIT_FRAME_MEM_CTRL (0x110)

Бит	Имя	Тип	Функция	Значение после сброса
0	SelBigEndian	R/W	0 - память кадров имеет прямой порядок байтов 1 – память кадров имеет обратный порядок байтов	N/A
1	Sel64BitsEndian	R/W	0 – буфер кадра имеет 64-битный порядок следования байтов 1 – буфер кадра имеет 32-битный порядок следования байтов	
2	SelCInterleave	R/W	0 – данные компонент CbCr записываются в отдельные модули памяти кадров 1 – данные компонент CbCr чередуются в памяти компоненты цветности. – Благодаря возможности работы с несколькими экземплярами и для идентификации каждого из экземпляров, независимо от того сколько экземпляров создается с помощью RunIndexes, на стороне хоста следует вести собственный список CrCrInterleave.	
3	SelNV21	R/W	Определение NV21. Действителен только, если разрешено SelCInterleave. 0 – данные компоненты цветности чередуются в порядке CbCr.(NV12) 1 - данные компоненты цветности чередуются в порядке CrCb.(NV21)	
12:9	TiledMapEn	R/W	Конфигурация тайловых карт распределения для буфера каждого закодированного изображения 0 - LINEAR_FRAME_MAP - TILED_FRAME_V_MAP - TILED_FRAME_H_MAP - TILED_FIELD_V_MAP - TILED_MIXED_V_MAP - TILED_FRAME_MB_RASTER_MAP - TILED_FIELD_MB_RASTER_MAP - TILED_FRAME_NO_BANK_MAP - TILED_FIELD_NO_BANK_MAP - LINEAR_FIELD_MAP	
14:13	TiledToLinearEn	R/W	Преобразование тайловой карты распределения в линейную в целях отображения данных В режиме кодирования данное поле обозначает преобразование линейной карты распределения в тайловую карту. 0 – запрещено преобразование тайловой карты в линейную 1 – тайловая карта - LINEAR_FRAME_MAP 2 – тайловая карта - LINEAR_FIELD_MAP	
15	Burstwritebacken	R/W	0 – запрещена отложенная запись пакета 1 – разрешена отложенная запись пакета. Запись выходных данных с 8 пакетов в режиме линейной карты распределения.	
16	RotBufGdiDirect	R/W	Примечание Данная возможность поддерживается только в декодере. 0: регистры GDI для буфера данных вращения задаются программными методами. Микро-ПО позволяет считывать регистры CMD_DEC_PIC_ROT_INDEX, CMD_DEC_PIC_ROT_ADDR_Y, CMD_DEC_PIC_ROT_ADDR_CB, CMD_DEC_PIC_ROT_ADDR_CR, CMD_DEC_PIC_ROT_STRIDE и обновлять их до соответствующих регистров GDI.	
18:17	TiledToLinearOnTheFlyEn	R/W	Преобразование тайловой карты распределения в линейную для каждого используемого в данный момент буфера восстановленных данных Если данное поле установлено в ноль, то для хранения данных кадра линейной карты распределения на каждый буфер закодированного изображения главной системе потребуется на один буфер кадра больше 0 – запрещено преобразование тайловой карты в линейную в процессе работы 1 – тайловая карта -LINEAR_FRAME_MAP 2 – тайловая карта - LINEAR_FIELD_MAP	

Таблица 1.286 – BIT_BIT_STREAM_PARAM (0x114)

Бит	Имя	Тип	Функция	Значение после сброса
0	SeqInitEscape	R/W	ВЫШЛО ИЗ УПОТРЕБЛЕНИЯ: пользоваться данным флагом не рекомендуется 0 – Функция принудительного ухода не активирована. 1 - Функция принудительного ухода активирована.	N/A
1	-	-	Зарезервировано	
2	StreamEnd	R/W	0 – Битовый поток больше, чем требуется декодеру. 1 – Весь битовый поток необходимо декодеру. В состоянии PIC_RUN BIT-процессор, проверив этот бит, «понимает», что битовый поток закончился. Данный флаг устанавливается на стороне хоста после записи всего битового потока, что позволяет получить последнее изображение битового потока. Кроме того, пока BIT-процессор ожидает оставшуюся часть битового потока, соответствующую одному изображению, и получает одно изображение посредством установки данного флага, на стороне хоста состояние занятости может быть сброшено. Данный флаг можно установить или сбросить на любом этапе процесса декодирования на стороне хоста после инициализации BIT-процессора. Как только этот бит будет установлен, декодер перестанет больше принимать битовый поток, соответственно, перед запуском декодирования этот флаг нужно сбросить на стороне хоста в 0, и установить его в 1 по окончании записи всего битового потока. Благодаря возможности использования нескольких экземпляров и для идентификации каждого из экземпляров, на стороне хоста осуществляется управление собственным списком потоков, независимо от того сколько экземпляров было создано с помощью RunIndexes.	
4:3	StreamMode	R/W	Режим обработки потоковых данных Когда указатель считывания доходит до указателя записи в середине процесса декодирования одного изображения, 0: из VPU на хост передается сигнал прерывания и на декодирование ожидается еще один битовый поток. (режим прерывания) : Из VPU непосредственно перед командой PIC_RUN возвращается информация о статусе (режим возврата) : В VPU выполняется поиск скрытых ошибок в оставшихся макроблоках. (режим picend) Описание режима обработки потока можно найти в разделе 4.1.5 - “Управление буфером битового потока” .	

Таблица 1.287 – BIT_TEMP_BUF_ADDR (0x118)

Бит	Имя	Тип Функция	Знач. после возврата
31:0	TempBufAddr	R/W Адрес байта памяти SDRAM во временном буфере BIT-процессора № Перед запуском блока VPU для успешного декодирования или кодирования изображения на стороне хоста в памяти SDRAM необходимо зарезервировать временной буфер. Данное значение должно быть выровнено до 1024. Узнать объем временного буфера в текущей архитектуре можно из справочного ПО к API.	N/A

Таблица 1.288 – BIT_RD_PTR (0x120)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	StreamRdPtr	R/W	Благодаря возможности использования нескольких экземпляров и для идентификации каждого из экземпляров, на стороне хоста осуществляется управление собственным списком RdPtr, независимо от того сколько экземпляров создано с помощью RunIndexes. На BIT-процессоре в регистр BIT_RD_PTR заносится адрес считывания битового потока во внешней памяти SDRAM с текущим RunIndex, таким образом, на стороне хоста распознается, какого объема битовый поток был считан и декодирован для определенного экземпляра. BIT-процессор обновляет данный регистр при каждой загрузке данных битового потока и автоматически зацикливает его. В текущей архитектуре при каждой передаче во внутренний буфер можно загрузить 512 байт. Поэтому по окончании загрузки данных указатель считывания битового потока увеличивается на 512	N/A

Таблица 1.289 – BIT_WR_PTR (0x124)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	StreamWrPtr	R/W	Благодаря возможности использования нескольких экземпляров и для идентификации каждого из экземпляров, на стороне хоста осуществляется управление собственным списком WrPtr независимо от того сколько экземпляров создано с помощью RunIndexes. При записи необходимой для декодирования битовой последовательности на стороне хоста в регистр BIT_WR_PTR заносится адрес текущей записи битового потока во внешней памяти SDRAM с RunIndex. Таким образом, на BIT-процессоре выполняется этот экземпляр.	N/A

Таблица 1.290 – BIT_RESET_CTRL (0x128)

Бит	Имя	Тип	Функция	Значение после сброса
0	ResetCtrl	R/W	Данный регистр используется для внутренних операций. Однако для безопасной работы следует убедиться в том, что на этапе инициализации VPU на стороне хоста данный регистр установлен в значение 0.	N/A

Таблица 1.291 – BIT_AXI_SRAM_USE (0x140)

Бит	Имя	Тип	Функция	Значение после сброса
0	BitAxiSecEn	R/W	1 – В BIT-процессоре разрешается использовать вспомогательную шину AXI для размещения буфера прогнозирования. 0 – Буфер прогнозирования размещается во внешней памяти SDRAM. Чтобы задействовать вторичную память RAM на стороне хоста перед выполнением команды SET_FRAME_BUF следует установить ее базовый адрес с помощью регистра CMD_SET_FRAME_AXI_BIT_ADDR (0x190).	N/A
1	IpAxiSecEn	R/W	1 - При intra-прогнозировании/AC-DC для прогнозирования буфера пикселей строк разрешается использовать вспомогательный AXI. 0 – Буфер пикселей строк Intra-прогнозирования/AC-DC размещается во внешней памяти SDRAM. Чтобы задействовать вторичную память RAM на стороне хоста перед выполнением команды SET_FRAME_BUF следует установить ее базовый адрес с помощью регистра CMD_SET_FRAME_AXI_IPACDC_ADDR (0x194).	
2	DbkYAxiSecEn	R/W	1 - Для временного буфера хранения данных компоненты яркости разрешается в фильтре деблокирования использовать вспомогательный AXI. 0 – Буфер фильтра деблокирования размещается во внешней памяти SDRAM. Чтобы задействовать вторичную память RAM на стороне хоста перед выполнением команды SET_FRAME_BUF следует установить ее базовый адрес с помощью регистра CMD_SET_FRAME_AXI_DBKY_ADDR (0x198).	
3	DbkCAxiSecEn	R/W	1 - Для временного буфера хранения данных компоненты цветности разрешается использовать в фильтре деблокирования вспомогательную шину AXI. 0 - Буфер фильтра деблокирования размещается во внешней памяти SDRAM. Чтобы задействовать вторичную память RAM на стороне хоста перед выполнением команды SET_FRAME_BUF следует установить базовый адрес с помощью регистра CMD_SET_FRAME_AXI_DBKC_ADDR (0x19C). 1 - Разрешается использовать вспомогательный AXI в фильтре перекрывания для размещения ее временного буфера. (только для VC1) 0 – Буфер фильтра перекрывания размещается во внешней памяти SDRAM.	
4	OvlAxiSecEn	R/W		

Продолжение таблицы 1.291

Бит	Имя	Тип	Функция	Значение после сброса
	OvlAxiSecEn	R/W	Чтобы задействовать вторичную память RAM на стороне хоста перед выполнением команды SET_FRAME_BUF следует установить ее базовый адрес с помощью регистра CMD_SET_FRAME_AXI_OVL_ADDR (0x1A0).	
5	BtpAxiSecEn	R/W	1 – Разрешается на BIT-процессоре использовать вспомогательную шину AXI для размещения временного буфера данных битового слоя (только для VC1) 0 - Буфер данных битового слоя размещается во внешней памяти SDRAM. Чтобы задействовать вторичную память RAM на стороне хоста перед выполнением команды SET_FRAME_BUF следует установить ее базовый адрес с помощью регистра CMD_SET_FRAME_AXI_BTP_ADDR (0x1A4).	

Таблица 1.292 – BIT_BYTE_POS_FRAME_START (0x144)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	BytePosFrameStart	R/W	Уведомление о позиции стартового байта в текущем кадре после декодирования этого кадра при синхронизации аудио и видеоданных. Декодер H.264/AVC ищет только 3-байтовый стартовый код (0x000001), а остальные декодеры ищут 4-байтовый стартовый код (0x00000001).	N/A

Таблица 1.293 – BIT_BYTE_POS_FRAME_END (0x148)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	BytePosFrameEnd	R/W	Уведомление о позиции конечного байта в текущем кадре по окончании декодирования этого кадра при синхронизации аудио и видеоданных.	N/A

Таблица 1.294 – BIT_FRM_DIS_FLG (0x150)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	Display Use	R/W	Номер бита n обозначает то, что n-ый индекс буфера кадра от индекса запуска 0 еще не отображался, но будет отображен. Поэтому данный буфер не используется для декодированного кадра. Если хост может управлять отображением данных буфера при сбросе n-ного бита, то они будут отображаться после индекса n. Благодаря возможности использования нескольких экземпляров и для идентификации каждого из экземпляров, на стороне хоста осуществляется управление собственным списком DisplayFlag независимо от того сколько экземпляров создано с помощью RunIndexes.	N/A

Таблица 1.295 – BIT_BUSY_FLAG (0x160)

Бит	Имя	Тип	Функция	Значение после сброса
0	BusyFlag	R/W	Значение 0 показывает, что BIT-процессор готов к приему команды со стороны хоста. Значение 1 показывает, что BIT-процессор выполняет и еще не завершил выполнение команды со стороны хоста. Перед записью регистра RunCommand на стороне хоста необходимо провести проверку данного бита. Если бит установлен в 1, то хост будет находиться в режиме ожидания, пока не установится значение 0.	N/A

Таблица 1.296 – BIT_RUN_COMMAND (0x164)

Бит	Имя	Тип	Функция	Значение после сброса
3:0	RunCommand	R/W	На стороне хоста в данный регистр записывается командный код. Командный код 4'b0001: SEQ_INIT Инициализация кодирования и декодирования последовательности. В режиме кодирования в BIT-процессоре анализируется параметр кодирования и заголовка кодовой последовательности. В режиме декодирования в BIT-процессоре декодируется заголовок последовательности (заголовок VOL для MPEG-4, SPS для H,264 и AVC) и передается отчетная информация о заголовке последовательности. • 4'b0010: SEQ_END Завершение последовательности кодирования и декодирования. В режиме кодирования BIT-процессор сбрасывает внутренний буфер битового потока во внешний буфер битового потока. В режиме декодирования BIT-процессор завершает процесс. * 4'b0011: PICURE_RUN Кодирование и декодирование одного изображения. В BIT-процессоре выполняется кодирование и декодирование одного изображения. Добавление SPS и PPS в буфер набора параметров BIT-процессора. Для H.264/AVC допустимо использовать несколько SPS и PPS, и с хост-процессора в BIT-процессор может быть передана информация об использовании в процессе декодирования одного из наборов параметров. 4'b1000: DEC BUF FLUSH Сброс всех данных относительно контекста буфера кадров. После завершения данной команды XOCT-процессор может приступить к декодированию, не отображая некоторые из задерживающихся буферов кадра. • 4'b1010: SLEEP Сохранение контекста (всех переменных внутренних параметров, которые необходимы для восстановления) в рабочем буфере. • 4'b1011: WAKE Загрузка из рабочего буфера контекста, сохраненного после декодирования предыдущего изображения 4'b0101: ENCODE HEADER Кодирование заголовка. Например, для H.264 и AVC во время данной команды между границами изображения можно вставить SPS (набор параметров последовательности), PPS (набор параметров изображения).	N/A

Таблица 1.297 – BIT_RUN_INDEX (0x168)

Бит	Имя	Тип	Функция	Значение после сброса
15:0	RunIndex	R/W	На стороне хоста перед каждой командой запуска в данный регистр заносится индекс процесса кодирования и декодирования. Процессы кодирования и декодирования в BIT-процессоре могут выполняться одновременно. Если запущено более одного процесса, то каждому процессу должен назначаться индекс другого процесса, указанного в данном регистре. Например, если одновременно запущен 1 процесс декодирования MPEG-2, 1 процесс декодирования AVC, 1 процесс декодирования AVC и 1 процесс декодирования VC1, то декодеру MPEG-2 назначается индекс процесса 0, декодеру AVC – индекс процесса 1, декодеру AVC – индекс процесса 2 и декодеру VC1 – индекс процесса 3. RunIndex должен входить в диапазон от 0 до 65535, включительно.	N/A

Таблица 1.298 – BIT_RUN_COD_STD (0x16C)

Бит	Имя	Тип	Функция	Значение после сброса
3:0	CodStd	R/W	CodStd[3:0] – стандарт кодека, который рассматривается далее. Перед каждой командой начала записи на стороне хоста в данный регистр заносится код индекса стандарта кодека. 4'b0000: ДЕКОДЕР H.264/AVC (включая MVC) 4'b0001: ДЕКОДЕР VC-1 4'b0010: ДЕКОДЕР MPEG-2 4'b0011: ДЕКОДЕР MPEG-4 И DivX-3 4'b0100: ДЕКОДЕР RVX 4'b0101: ДЕКОДЕР AVS 4'b0110: Зарезервировано 4'b0111: ДЕКОДЕР VPX 4'b1000: КОДЕР H.264/AVC 4'b1011: КОДЕР MPEG-4	N/A

Таблица 1.299 – BIT_INT_ENABLE (0x170)

Бит	Имя	Тип	Функция	Значение после сброса
15:0	IntEnable	R/W	Регистр флага разрешения прерывания	N/A
			Каждый бит данного регистра представляет собой флаг разрешения прерывания, предназначенный для разных режимов включения прерывания. 1 означает, что прерывание разрешено и что BIT-процессор генерирует прерывание, а 0 означает, что прерывание запрещено. 0-ой бит (LSB): инициализация выполнена. Такое прерывание генерируется только один раз после запуска BIT-процессора [0]: После установления BIT_CODE_RUN в 1, VPU инициализируется. [1]: После установления BIT_RUN_COMMAND в 1, выполнение команды DEC_SEQ_INIT/ENC_SEQ_INIT завершается. [2]: После установления BIT_RUN_COMMAND в 2, выполнение команды DEC_SEQ_END/ENC_SEQ_END завершается . [3]: Поле установления BIT_RUN_COMMAND в 3, выполнение команды DEC_PIC_RUN/ENC_PIC_RUN завершается. [4]: После установления BIT_RUN_COMMAND в 4, выполнение команды SET_FRAME_BUF завершается. [5]: После установления BIT_RUN_COMMAND в 5, выполнение команды ENC_HEADER_SET завершается. [6]: После установления BIT_RUN_COMMAND в 6, выполнение командыENC_PARA_SET завершается. [7]: После установления BIT_RUN_COMMAND в 7, выполнение команды DEC_PARA_SET завершается. [8]: После установления BIT_RUN_COMMAND в 8, выполнение команды DEC_BUF_FLUSH завершается. [9]: Работа буфера пользовательских данных полностью прервана. Такой сигнал выдается, когда объема userDataBufSize не достаточно для хранения пользовательских данных в процессе декодирования кадра. Чтобы справиться с данной проблемой, на CMD_DEC_SEQ_USER_DATA_OPTION[5] передаются данные пользователя и на CMD_DEC_SEQ_USER_DATA_OPTION[10] устанавливается режим отчета о прерывании [10]: Декодирование одного полукадра изображения из файла завершено. [13]: Декодирование NumMbRows выполнено. Данные о данной настройке можно получить в CMD_DEC_PIC_OPTION[18] и CMD_DEC_PIC_NUM_ROWS. [14]: В ходе декодирования внешний буфер битового потока остается пустым. [15]: В ходе кодирования внешний буфер битового потока заполнен. Если BufPicReset установлен в 1, то данный флаг устанавливается в ноль. В результате этого наблюдается переизбыток блоков, и указатель записи в WR_PTR вынужден перейти на стартовую позицию данного буфера блоков без какого-либо подтверждения. Если и BufPicReset, и данный флаг установлены в 1 и наблюдается переизбыток блоков, то после получения сигнала прерывания буфера вследствие его заполненности блок VPU прекращает кодирование и переходит в режим ожидания отклика со стороны хоста. В таком случае, чтобы продолжить процесс кодирования следует сбросить на стороне хоста сигнал прерывания.	

Таблица 1.300 – BIT_INT_REASON (0x174)

Бит	Имя	Тип	Функция	Значение после сброса
15:0	IntReason	R/W	Регистр флага указания причины прерывания Каждый бит данного регистра обозначает то же самое, что и соответствующее битовое поле регистра BIT_INT_ENABLE. В BIT-процессоре в соответствующее битовое поле, которое отвечает за генерирование запроса прерывания, может быть записана 1, а затем на стороне хоста данный регистр может быть считан при обслуживании этого прерывания и выяснится, откуда этот сигнал прерывания пришел. Хост отвечает за установку данного регистра в 0 для приема следующего сигнала прерывания. Примечание - Если прерывание сбрасывается с помощью обоих регистров BIT_INT_REASON и BIT_INT_CLEAR, то для корректной работы после установления регистра BIT_INT_REASON в 0 следует установить в 1 регистр BIT_INT_CLEAR.	N/A

Таблица 1.301 – BIT_RUN_AUX_STD (0x178)

Бит	Имя	Тип	Функция	Значение после сброса
1:0	RunAuxStd	R/W	Вспомогательный регистр индекса стандарта кодека. На стороне хоста в данный регистр перед выполнением каждой команды запуска записывается код индекса кодека. Значение данного регистра может меняться в зависимости от значения регистра RunCodStd. Если RunCodStd установлен в 4'b00000 (ДЕКОДЕР H.264/AVC), 0 - ДЕКОДЕР H.264/AVC 1 - ДЕКОДЕР MVC Если RunCodStd установлен в 4'b00000 (ДЕКОДЕР MPEG-4/DivX-3) 0 - ДЕКОДЕР MPEG-4 (включая H.263, Sorenson Spark, DivX and XviD) 1 - ДЕКОДЕР DivX3.11 Если RunCodStd установлен в 4'b0111(ДЕКОДЕР VPX), 0 - ДЕКОДЕР Theora 1 - ДЕКОДЕР VP6 (вышел из употребления) 2 - ДЕКОДЕР VP8	N/A

1.3.6.4.2.3.3 Регистры ввода и вывода команд

Таблица 1.302 – CMD_DEC_SEQ_INIT_ESCAPE (0x114)

Бит	Имя	Тип	Функция	Значение после сброса
0	SeqInitEscape	R/W	0 – Функция принудительного ухода не активирована. 1 - Функция принудительного ухода не активирована.	N/A

Таблица 1.303 – CMD_DEC_SEQ_BB_START (0x180)

Бит	Имя	Тип	Функция	Команда
31:0	BitBufAddr	R/W	Адрес байта в памяти SDRAM буфера битового потока Необходимо выполнить запись этого регистра на стороне хоста перед выполнением команды DEC_SEQ_INIT. Буфер битового потока должен согласовываться с шириной шины памяти. Например, если на стороне хоста используется 64-разрядная шина, то буфер битового потока должен соответствовать по ширине 8 байтам. Более подробную информацию можно найти в разделе 4.1.5 - “Управление буфером битового потока”	DEC_SEQ_INIT

Таблица 1.304 – CMD_DEC_SEQ_BB_SIZE (0x184)

Бит	Имя	Тип	Функция	Команда
13:0	BitBufSize	R/W	Объем буфера битового потока, выраженный в килобайтах Необходимо выполнить запись этого регистра на стороне хоста перед выполнением команды DEC_SEQ_INIT. Объем буфера битового потока может изменяться в пределах от 2 кбайт до 256 Мбайт. Для обеспечения управления несколькими экземплярами в VPU рекомендуется использовать буфер битового потока объемом свыше 10 кбайт. Более подробную информацию можно найти в разделе 4.1.5 - “Управление буфером битового потока”	DEC_SEQ_INIT

Таблица 1.305 – CMD_DEC_SEQ_OPTION (0x188)

Бит	Имя	Тип	Функция	Команда
0	OutLoopDbkEn	R/W	Возможность внеканального деблокирования MP2 и 4. Если данный флаг включен, то для обеспечения работы фильтра деблокирования количество буферов декодированных кадров минимального объема в регистре RET_DEC_SEQ_FRAME_NEED устанавливается в +1. Декодированные данные и данные с устраненной блочностью заносятся в буфер кадра.	DEC_SEQ_INIT
1	ReorderEn	R/W	Возможность переупорядочивания в буфере изображения при декодировании H.264, VC-1 и AVS. В случае H.264/AVC выходное декодированное изображение можно переупорядочить, если pic_order_cnt_type установлен в 0 или 1. В этом случае кодер должен задержать отображение выходных данных для их переупорядочения, но для некоторых приложений (кроме видеотелефонии) такая задержка отображения может оказаться нежелательной. Чтобы запретить функцию переупорядочения в буфере отображения выходных данных на хосте можно установить данный флаг в 0. BIT-процессор не переупорядочивает данные в буфере выходных данных, если pic_order_cnt_type установлен в 0 или 1. Если данный флаг установлен в 1, то в BIT-процессоре выполняется переупорядочение выходных декодированных изображений, и отображение входных данных задерживается на величину, указанную в регистре [RET_DEC_SEQ_FRAME_DELAY].	
3:2	-	-	Зарезервировано	
4	-	-	Зарезервировано для отладки (VC-1)	
5	-	-	Зарезервировано	
6	-	-	Зарезервировано	

Таблица 1.306 – CMD_DEC_SEQ_USER_DATA_OPTION (0x194)

Бит	Имя	Тип	Функция	Команда
4:0	-	-	Зарезервировано	DEC_SEQ_INIT
5	SeqUserDataEn- able	R/W	Возможность выдачи пользовательских данных Если данная опция включена, то пользовательские данные записываются в буфер пользовательских данных, который указан в UserDataBase. Данное поле доступно только для декодера MPEG-2.	
9:6	-	-	Зарезервировано	

Продолжение таблицы 1.306

Бит	Имя	Тип	Функция	Команда
10	SeqUserDataRe- port Mode	R/W	Устанавливается режим выдачи пользовательских данных 0 – режим прерывания 1 – режим отключения прерывания Если данная опция установлена в 0, то по заполнению буфера пользовательских данных декодером выдается сигнал прерывания. Чтобы продолжить декодирование на стороне хоста необходимо сбросить бит прерывания. Если данная опция установлена в 1, то декодер выдает такой объем пользовательские данные, какой способен вместить буфер пользовательских данных. Если буфер полный, то декодер игнорирует оставшиеся пользовательские данные. На хост приходит информация о переполнении буфера пользовательских данных с помощью переменой userDataBufFull в формате отчета о пользовательских данных. Данное поле доступно только для декодера MPEG-2.	

Таблица 1.307 – CMD_DEC_SEQ_MP4_ASP_CLASS (0x19C)

Бит	Имя	Тип	Функция	Команда
2:0	Mp4AspClass	R/W	Указывается формат стандарта MPEG-4. • 3'h0 - MPEG-4 3'h1 - DivX 5.0 или версии выше 3'h2 - Xvid • 3'h5 - DivX 4.0 3'h6 – устаревшая версия Xvid (ниже 0.9.x или с номером сборки <= 12) Каждый бит показывает инструменты, используемые для декодирования MPEG-4 [0]: divx_flag с поддержкой формулы Divx MV_chroma и Divx idct. [1]: xvid_flag и поддержкой формулы xvid MV_chroma [2]: невыровненное заполнение	DEC_SEQ_INIT
3	DisableGMC	R/W	Данный флаг действует только при декодировании MPEG-4. Отключается флаг декодирования GMC (глобальная компенсация движения). 1 – запрещается поток декодирования с GMC. 0 – разрешается поток декодирования с GMC. Данный флаг действует только при декодировании MPEG-4.	
8	SorensonStream	R/W	Указывается то, в каком формате поток – в Sorenson Spark или другом. 1 – поток sorenson spark 0 – поток не sorenson spark	

Таблица 1.308 – CMD_DEC_SEQ_X264_MV_EN (0x19C)

Бит	Имя	Тип	Функция	Команда
0	EnX264Mv	R/W	Возможность обработки векторов движения вне нормального диапазона для X264. Некоторые из потоков X264 могут иметь векторы движения вне нормального диапазона. В таком случае компенсатор движения может выдать иной результат. Данный флаг заставляет VPU обрабатывать векторы движения не в стандартно поддерживаемом диапазоне, а в удвоенном. Эффективность такой обработки может быть ниже, чем эффективность обработки в обычном диапазоне.	DEC_SEQ_INIT

Таблица 1.309 – CMD_DEC_SEQ_VC1_STREAM_FMT (0x19C)

Бит	Имя	Тип	Функция	Команда
1:0	Vc1StreamFmt	R/W	Указывается формат потока VC-1. Данные биты доступны только в VC-1 0 - V1 RCV ПОТОКА - V2RCV ПОТОКА – ПОТОК Элементарный	DEC_SEQ_INIT
2	AutoDetecDis	R/W	Запрещается автообнаружение потока VC-1	

Таблица 1.310 – CMD_DEC_SEQ_USER_DATA_BASE_ADDR (0x1AC)

Бит	Имя	Тип	Функция	Команда
31:0	UseDataBase	R/W	Базовый адрес пользовательских данных Более подробную информацию можно найти в описании структуры DecParam. Данное поле доступно только для декодера H.264/AVC и MPEG-2.	DEC_SEQ_INIT

Таблица 1.311 – CMD_DEC_SEQ_USER_DATA_BUF_SIZE (0x1B0)

Бит	Имя	Тип	Функция	Команда
31:0	UserDataBufSize	R/W	Объем буфера пользовательских данных Более подробную информацию можно найти в описании структуры DecParam. Данное поле доступно только для декодера H.264/AVC и MPEG-2.	DEC_SEQ_INIT

Таблица 1.312 – RET_DEC_SEQ_MP2_BAR_LEFT_RIGHT (0x180)

Бит	Имя	Тип	Функция	Команда
15:0	PixelNumberStart OfRightBar	R	Значение синтаксиса pixel_number_start_of_right_bar информации из столбца в пользовательских данных ATSC Данное поле доступно только для декодера MPEG-2.	DEC_SEQ_INIT
31:16	PixelNumberEnd- Of LeftBar	R	Значение синтаксиса pixel_number_end_of_left_bar информации из столбца в пользовательских данных ATSC Данное поле доступно только для декодера MPEG-2.	

Таблица 1.313 – RET_DEC_SEQ_MP2_BAR_TOP_BOTTOM (0x184)

Бит	Имя	Тип	Функция	Команда
15:0	LineNumber- StartOf Bot- tomBar	R	Значение синтаксиса line_number_start_of_bottom_bar информации из столбца в пользовательских данных ATSC Данное поле доступно только для декодера MPEG-2	DEC_SEQ_INIT
31:16	LineNumberEnd- Of TopBar	R	Значение синтаксиса line_number_end_of_top_bar информации из столбца в пользовательских данных ATSC Данное поле доступно только для декодера MPEG-2.	

Таблица 1.314 – RET_DEC_SEQ_VUI_INFO (0x18C)

Бит	Имя	Тип Функция	Команда
0	FixedFrameRate-Flag	R1: указывает на то, что временной интервал между моментами времени вывода декодером двух любых последовательных изображений ограничивается fixed_frame_rate_flag в синтаксисе VUI H.264/AVC. 0: указывает на то, что никаких ограничений в отношении временного интервала между моментами времени вывода декодером двух любых последовательных изображений нет. Примечание - Подробную информацию о данных флагах можно найти в описании синтаксиса VUI H.264/AVC.	DEC_SEQ_INIT
1	TimingInfoPre-sent	R Синтаксис VUI timing_info_present_flag in H.264/AVC 1: FixedFrameRateFlag действительный. 0: FixedFrameRateFlag не действительный.	
4:2 7:5	ChromaSample-LocType Bottom-Field ChromaSample-LocType Top-Field	R chroma_sample_loc_type_bottom_field в синтаксисе VUI H.264/ AVC. Он показывает расположение сэмплов компоненты цветности для нижнего полукадра. R chroma_sample_loc_type_top_field в синтаксисе VUI H.264/ AVC. Он показывает расположение сэмплов компоненты яркости для верхнего полкукадра.	
8	ChromaLocInfoP-resent	R chroma_loc_info_present_flag в синтаксисе VUI H.264/AVC 1: ChromaSampleLocTypeTopField и Chro-maSampleLoc TypeTopField действительные. 0: ChromaSampleLocTypeTopField и Chro-maSampleLocTypeTopField не действительные.	
15:9	-	- Зарезервировано	
23:16	ColorPrimaries	R Синтаксис colour_primaries syntax в параметре VUI в H.264/AVC	
24	ColorDescPresent	R colour_description_present_flag в параметре VUI в H.264/AVC	
25	isExtSAR	R Данный флаг указывает регистр. RET_DEC_PIC_ASPECT представляет 8-битный aspect_ratio_idc или 32-битный extended_SAR. Если aspect_ratio_idc == extended_SAR mode, флаг возвращается в 1. Данный флаг действует только в случае H.264/ AVC.	

Продолжение таблицы 1.314

Бит	Имя	Тип Функция	Команда
26	VidFullRange	R video_full_range в параметре VUI в H.264/ AVC	
29:27	VidFormat	R video_forma в параметре VUI в H.264/AVC	
30	VidSigTypePre-sent	R video_signal_type_present_flag в параметре VUI в H.264/AVC	
31	VuiParamPresent	R vui_parameters_present_flag в параметре VUI в H.264/AVC	

Таблица 1.315 – RET_DEC_SEQ_VUI_PIC_STRUCT (0x190)

Бит	Имя	Тип	Функция	Команда
0	PicStrPresent	R	pic_struct_present_flag блока VUI в H.264/AVC. Данное поле действует только в случае декодирования H.264/AVC	DEC_SEQ_INIT
4:1	AvcPicStruct	R	pic_struct при выдаче данных VUI H.264/AVC	

Таблица 1.316 – RET_DEC_SEQ_BIT_RATE (0x1B4)

Бит	Имя	Тип	Функция	Команда
31:0	BitRate	R	Если возвращается в -1 (0xFFFF_FFFF), то это означает то, что в заголовке последовательности нет информации о битовой скорости. MPEG-2 bit_rate (30-битное целочисленное значение), которое связано с bit_rate_extension[11:0] и bit_rate_value[17:0] MPEG-4 bit_rate (30-битное целочисленное значение), которое связано с first_half_bit_rate[14:0] и latter_half_bit_rate[14:0] AVS bit_rate (30-битное целочисленное значение), которое связано с bit_rate_upper[11:0] и bit_rate_lower[17:0] VC-1 В случае Простого и Главного профилей, HRD_RATE[31:0] Расширенном профиле устанавливается в -1 AVC/DivX3.11/RV -1	DEC_SEQ_INIT

Таблица 1.317 – RET_DEC_SEQ_EXT_INFO (0x1B8)

Бит	Имя	Тип	Функция	Команда
0	LowDelay	R	Синтаксис low_delay расширенного представления последовательности спецификации MPEG-2	DEC_SEQ_INIT
14:1	DisplayVertical-Size	R	Синтаксис display_vertical_size расширенного представления последовательности спецификации MPEG-2	
28:15	DisplayHorizontal Size	R	Синтаксис display_horizontal_size расширенного представления последовательности спецификации MPEG-2	

Таблица 1.318 – RET_DEC_SEQ_SUCCESS (0x1C0)

Бит	Имя	Тип	Функция	Команда
0	RetStatus	R	0 – команда DEC_SEQ_INIT выполнена с ошибкой 1 – команда DEC_SEQ_INIT выполнена без ошибок	DEC_SEQ_INIT
3:1	-	-	Зарезервировано	
4	BitEmptyRollback	R	Данный флаг действует только когда RetStatus (нулевой бит) установлен в 0, в StreamMode в регистре BIT_BIT_STREAM_PARAM установлен как в режиме повторного прогона. 0 – команда DEC_SEQ_INIT, которая не была выполнена из-за другой ошибки в синтаксисте. 1 – команда DEC_SEQ_INIT не была выполнена из-за недостаточного битового потока для одного изображения в режиме повторного прогона.	
30:5	-	-	Зарезервировано	
31	MemoryAccess Violation	R	0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.319 – RET_DEC_SEQ_SRC_SIZE (0x1C4)

Бит	Имя	Тип	Функция	Команда
15:0	PictureHeight	R	Высота декодированного изображения в пикселях	DEC_SEQ_INIT
31:16	PictureWidth	R	Ширина декодированного изображения в пикселях	

Таблица 1.320 - RET_DEC_SEQ_ASPECT (0x1C8)

Бит	Имя	Тип Функция	Команда
31:0	AspectRatio	R • H.264/AVC aspect_ratio_idc[7:0] когда isExtSAR флаг на регистре RET_DEC_SEQ_HEADER_REPORT или RET_DEC_PIC_VUI_INFO установлен в 0. Если aspect_ratio_info_present_flag = 0, то данный регистр возвращается в -1 (0xffffffff). VC-1 Зависимость между регистрами ASPEC_HORZ_SIZE и ASPEC_VERT_SIZE в отношении длины и ширины изображения описывается как [15:8] против [7:0] по специф. • MPEG-4/H.263: [3:0] ParCode 0 = запрещено 1 = 1:1 2 = 12:11 3 = 10:11 4 = 16:11 5 = 40:33 15(0xF) = расширенное PAR Если ParCode равен 15 (0xF: расширенное PAR), то на стороне хоста для передачи информации aspect_ratio следует использовать с 4-го по 19-й биты. [19:4] расширенное PAR [19:12] = высота в PAR [11:4] = ширина в PAR (двоичное представление соотношения PAR по ширине и высоте) RV: aspect_ratio_info[3:0] MPEG-2 Данное значение - aspect_ratio_information[3:0]. Оно используется как индекс таблицы 6-3 в ISO/IEC 13818-2. Данное значение определяется как половина объема памяти, выделенного для однорядного изображения YUV, выраженное в килобайтах. AVS Данное значение - aspect_ratio_info[3:0]. Оно используется как индекс таблицы 7-5 AVS Part2.	DEC_SEQ_INIT

Таблица 1.321 – RET_DEC_SEQ_FRAME_NEED (0x1CC)

Бит	Имя	Тип	Функция	Команда
4:0	FrameBufNeed	R	Количество буферов кадра минимального объема, необходимое для успешного декодирования потока, а также его хранения. Для H.264/MVC данное значение можно увеличить более чем в 2 раза, а максимальным значением может быть 19 (16 – для справочного, 2 – для текущего и 1 – для отображения для MVC). На стороне хоста буферы кадра должны резервироваться с минимальным значением. Например, с BIT-процессора в данный регистр возвращается 7 и на стороне хоста подготавливается 7 буферов кадра и по команде SET_FRAME_BUF приходит информация об адресе буфера кадра. На BIT-процессоре размещается 7 буферов кадра и на область, предназначенную для хранения данных закодированного изображения, выделяются соответствующие адреса буфера кадра. Если операция MMCO (операция управления распределением памяти) и переупорядочение выходных данных не выполняются, то BIT-процессор назначает буферу декодированного кадра номера в порядке 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, …, Это позволит избежать ситуации, когда данные справочного кадра вынуждены будут записываться поверх других.	DEC_SEQ_INIT

Таблица 1.322 – RET_DEC_SEQ_FRAME_DELAY (0x1D0)

Бит	Имя	Тип	Функция	Команда
4:0	FrameBufDelay	R	Максимальное время задержки в буфере кадра изображения при буферизации переупорядоченных декодированных изображений. В случае переупорядочения изображений в BIT-процессоре может происходить задержка отображения декодированного изображения. Например, BIT-процессор может возвращать в данный регистр 5, когда по команде DEC_PIC_RUN декодируются 5 первых кадров, потому что отсутствует изображение, которое следует вывести на экран. Максимальным значением данного регистра может быть 16. Значение равно 0, если флаг [ReorderEn] имеет значение 0 для H.264/AVC. В случае декодирования VC-1 данное значение равно 0, если отсутствует B-изображение, и равно 1, если, несмотря на флаг [ReorderEn] есть B-изображение.	DEC_SEQ_INIT

Таблица 1.323 – RET_DEC_SEQ_INFO (0x1D4)

Бит	Имя	Тип	Функция	Команда
0	DataPartEn	R	0 – запрещено разбиение данных 1 – разрешено разбиение данных (только для MPEG-4) После выполнения команды DEC_SEQ_INIT, в BIT-процессоре записывается флаг разрешения разбиения данных, указанный в информации заголовка декодированной последовательности MPEG-4, в данном регистре. В процессе кодирования данный регистр не используется.	DEC_SEQ_INIT
1	RevVlcEn	R	0 – используется таблица нормального VLC 1 - используется таблица реверсивного VLC Если DataPartEn равно 0, данный бит не используется.	
2	ShortVideoHeader	R	0 – нормальный поток MPEG-4 1 - видеопоток с коротким заголовком	
3	AnnexJ	R	1 - поддерживается приложение J для MPEG-4	
14:9	SpsId	R	Для H.264/AVC данное поле возвращает значение seq_parameter_set_id, если DEC_SEQ_INIT успешно выполнена. В этом случае значение 0x3f не действует.	

Таблица 1.324 – RET_DEC_SEQ_VP8_SCALE_INFO (0x1D4)

Бит	Имя	Тип	Функция	Команда
31:30	HorScale	R	Фактор передискретизации VP8 по горизонтали 0: без масштабирования 1: увеличивается в масштабе 5/4 2: увеличивается в масштабе 5/3 3: увеличивается в масштабе в 2 раза	DEC_SEQ_INIT
29:28	VerScale	R	Фактор передискретизации VP8 по вертикали 0 : без масштабирования: увеличивается в масштабе 5/4 : увеличивается в масштабе 5/3 3: увеличивается в масштабе в 2 раза	
27:14	PicWidth	R	Ширина кадра Vp8 в канале декодирования	
13:0	PicWidth	R	Высота кадра Vp8 в канале декодирования	

Таблица 1.325 – RET_DEC_SEQ_CROP_LEFT_RIGHT (0x1D8)

Бит	Имя	Тип	Функция	Команда
15:0	CropRightOffset	R	Информация о сдвиге вправо при обрезке в декодере Данное поле доступно только для декодера H.264/AVC.	DEC_SEQ_INIT
31:16	CropLeftOffset	R	Информация о сдвиге влево при обрезке в декодере Данное поле доступно только для декодера H.264/AVC.	

Таблица 1.326 – RET_DEC_SEQ_CROP_TOP_BOTTOM (0x1DC)

Бит	Имя	Тип	Функция	Команда
15:0	CropBottomOffset	R	Информация о сдвиге с нижнего края при обрезке в декодере Данное поле доступно только для декодера H.264/AVC.	DEC_SEQ_INIT
31:16	CropTopOffset	R	Информация о сдвиге с верхнего края при обрезке в декодере Данное поле доступно только для декодера H.264/AVC.	

Таблица 1.327 – RET_DEC_SEQ_SEQ_ERR_REASON (0x1E0)

Бит	Имя	Тип	Функция	Команда
31:0	SeqErrReason	R	Уведомление о причине возникновения ошибки в последовательности. Более подробную информацию об ошибках, возникающих при декодировании заголовков, можно найти в приложении G – ПРИЧИНЫ ВОЗНИКНОВЕНИЯ ОШИБОК ПРИ ДЕКОДИРОВАНИИЗАГОЛОВКОВ ПОСЛЕДОВАТЕЛЬНОСТЕЙ.	DEC_SEQ_INIT

Таблица 1.328 – RET_DEC_SEQ_FRATE_NR (0x1E4)

Бит	Имя	Тип	Функция	Команда
31:0	FRateNr	R	Числитель в выражении частоты кадров Требования к числителю для каждого стандарта видеокодирования рассмотрены в приложении D – ЧИСЛИТЕЛИ В ВЫРАЖЕНИИ ЧАСТОТЫ КАДРОВ настоящего документа.	DEC_SEQ_INIT

Таблица 1.329 – RET_DEC_SEQ_FRATE_DR (0x1E8)

Бит	Имя	Тип	Функция	Команда
31:0	FRateDr	R	Знаменатель в выражении частоты кадров Если в потоке нет информации о частоте кадров, то VPU передает в данный регистр -1. Требования к знаменателю для каждого стандарта видеокодирования рассмотрены в приложении C – ЗНАМЕНАТЕЛИ В ВЫРАЖЕНИИ ЧАСТОТЫ КАДРОВ настоящего документа.	DEC_SEQ_INIT

Таблица 1.330 – RET_DEC_SEQ_HEADER_REPORT (0x1EC)

Бит	Имя	Тип	Функция	Команда
7:0	Profile	R	Профиль декодированного потока. Данное значение совпадает со значением, установленным в требованиях по каждому стандарту видеокодирования. Для кодеков VP8 данное поле возвращает номер версии декодированных данных (0~3), рассмотренный в пункте 9.1 спецификации VP8.	DEC_SEQ_INIT
15:8	Level	R	Уровень потока декодированных данных	

Продолжение таблицы 1.330

Бит	Имя	Тип	Функция	Команда
16	Progressive	R	Этот единичный бит означает, что поток декодированных данных включает в себя только прогрессивные кадры. MPEG-4: обратное значение прореженного бита в заголовке VOL H.264/AVC: frame_mbs_only_flag в SPS AVS: progressive_sequence в заголовке последовательности MPEG-2: progressive_sequence в расширенном заголовке последовательности VC-1: обратное значение прореженного бита в заголовке последовательности	
17	Direct_8x8 _ inference_flag	R	Элемент синтаксиса SPS H.264/AVC, используемый в B-изображении. Для кодеков VC1 это родственный бит для PullDown, который указывает на то, присутствуют ли в заголовках изображения элементы синтаксиса RPT FRM или TFF и RFF.	
18	Vc1_psf	R	Это единичный бит обозначает прогрессивно сегментированный кадр в VC-1.	
19	Constraint_set0 _flag	R	Элемент синтаксиса в H.264/AVC, используемый для формирования уровня в форматах H.264/AVC. В других стандартах данный бит должен игнорироваться.	
20	Constraint_set1 _ flag	R	Элемент синтаксиса в H.264/AVC, используемый для формирования уровня в форматах H.264/AVC. В других стандартах данный бит должен игнорироваться.	
21	Constraint_set2 _ flag	R	Элемент синтаксиса в H.264/AVC, используемый для формирования уровня в форматах H.264/AVC. В других стандартах данный бит должен игнорироваться.	
22	Constraint_set3 _ flag	R	Элемент синтаксиса в H.264/AVC, используемый для формирования уровня в форматах H.264/AVC. В других стандартах данный бит должен игнорироваться.	
24:23	chromaFormat	R	Элемент синтаксиса в видеостандартах, который ссылается на формат цветности. • H.264/AVC, – 0 – черно-белый – 1 - 4:2:0 – 2 - 4:2:2 – 3 - 4:4:4 AVS, – 1 - 4:2:0 – 2 - 4:2:2 Прочие стандарты, – 1 - 4:2:0	
25 26	isExtSAR -	R -	Данный флаг указывает регистр RET_DEC_PIC_ASPECT представляет 8-битный aspect_ratio_idc или 32-битный extended_SAR. Если aspect_ratio_idc == режим extended_SAR, данный флаг возвращает 1. Флаг действует только в случае H.264/ AVC. Зарезервировано	
30:27	MaxNumRef-Frames	R	Это max_num_ref_frames, один из элементов синтаксиса SPS в H.264/AVC.	
31	MaxNumRef-FramesFlag	R	Один из элементов синтаксиса SPS в H.264/ AVC. 0: max_num_ref_frames равно 0. 1: max_num_ref_frames равно 0.	

Таблица 1.331 – RET_DEC_SEQ_END_SUCCESS (0x1C0)

Бит	Имя	Тип	Функция	Команда
0	RetStatus	R	0 – команда DEC_SEQ_END выполнена с ошибкой 1 – команда DEC_SEQ_END выполнена без ошибок	DEC_SEQ_END
31	MemoryAccess Violation	R	0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.332 – CMD_DEC_PIC_ROT_MODE (0x180)

Бит	Имя	Тип	Функция	Команда
3:0	PostRotMode	R/W	PostRotMode[1:0]: повернуть на 01 - 90 градусов по часовой стрелке - повернуть на 180 градусов по часовой стрелке - повернуть на 270 градусов по часовой стрелке PostRotMode[2]: включить/отключить отражение по вертикали	DEC_PIC_RUN
4	PostRotEn	R/W	PostRotMode[3]: включить/отключить отражение по горизонтали	
			Если PostRotMode равен 0000, то возможность вращения после обработки отключена. 0: отключение фильтра вращения.1: включение фильтра вращения	
5	PostDeringEn	R/W	0: отключение фильтра шумоподавления 1: включение фильтра шумоподавления Повернутые изображения и/или изображения, обработанные фильтром шумоподавления, хранятся по адресам RotBufAddrY, RotBufAddrCb, RotBu- fAddrCr, которые задаются на стороне ХОСТа. Примечание - Функция шумоподавления поддерживается только в формате MPEG-2/4.	
6	PostBufConfigEn	R/W	Если данное значение равно 1, то у буфера постобработки есть собственная конфигурация, отличная от конфигурации буфера восстановленных блоков.	
SelBigEndian		R/W	Данное поле действительно только, если PostBufConfigEn равно 1. Порядок следования байтов в буфере исходящих кадров постобработки 0: постобработка памяти исходящих кадров происходит в прямом порядке. 1: постобработка памяти исходящих кадров	
Sel32bit		R/W	происходит в обратном порядке. Данное поле действительно только, если PostBufConfigEn равно 1. Формат слов в буфере исходящих кадров постобработки 0: постобработка памяти исходящих кадров происходит в формате 64-битных слов. 1: постобработка памяти исходящих кадров происходит в формате 32-битных слов.	

Продолжение таблицы 1.332

Бит	Имя	Тип	Функция	Команда
18	CbCrInterleave	R/W	Данное поле действительно только, если PostBufConfigEn равно 1. Формат с прореживанием по компоненте цветности для буфера исходящих кадров постобработки 0: Данные компонент Cb и Cr разделяются в памяти по каждой компоненте цветности буфера исходящих кадров постобработки. 1: Данные компонент Cb и Cr прореживаются в памяти по компоненте цветности буфера исходящих кадров постобработки.	

Таблица 1.333 – CMD_DEC_PIC_ROT_INDEX (0x184)

Бит	Имя	Тип	Функция	Команда
7:0	RotDstIndex	R/W	Индекс буфера кадра для повернутых изображений и/или выходных изображений, обработанных фильтром шумоподавления JPEG: индекс буфера кадра по компоненте яркости для декодированных выходных изображений	DEC_PIC_RUN
			Примечание: Функция шумоподавления поддерживается только для стандарта MPEG-2/4.	
15:8	RotCbIndex	R/W	JPEG: индекс буфера прореженных кадров по компонентам Cb/Cr или отдельных кадров по компоненте Cb для декодированных выходных изображений (по умолчанию: для буфера кадров по компоненте яркости значение то же)	
23:16	RotCrIndex	R/W	JPEG: буфер отдельных кадров по компоненте Cr (по умолчанию: для буфера кадров по компоненте яркости значение то же)	

Таблица 1.334 – CMD_DEC_PIC_ROT_ADDR_Y (0x188)

Бит	Имя	Тип	Функция	Команда
0	RotBufAddrY	R/W	Адрес буфера кадра для повернутых изображений и/или выходных изображений, обработанных фильтром шумоподавления, по компоненте Y	DEC_PIC_RUN
			Примечание: Функция шумоподавления поддерживается только для стандарта MPEG-2/4.	

Таблица 1.335 – CMD_DEC_PIC_ROT_ADDR_CB (0x18C)

Бит	Имя	Тип	Функция	Команда
31:0	RotBufAddrCb	R/W	Адрес буфера кадра для повернутых изображений и/или выходных изображений, обработанных фильтром шумоподавления, по компоненте Cb	DEC_PIC_RUN
			Примечание: Функция шумоподавления поддерживается только для стандарта MPEG-2/4.	

Таблица 1.336 – CMD_DEC_PIC_ROT_ADDR_CR (0x190)

Бит	Имя	Тип	Функция		Команда
31:0	RotBufAddrCr	R/W	Адрес буфера кадра для повернутых изображений и/или выходных изображений, обработанных фильтром шумоподавления, по компоненте Cr		DEC_PIC_RUN
			Примечание	Функция шумоподавления поддерживается только для стандарта MPEG-2/4.	

Таблица 1.337 – CMD_DEC_PIC_OPTION (0x194)

Бит	Имя	Тип Функция	Команда
1:0	-	- Зарезервировано	
2	IframeSearchEn	R/W Возможность поиска I-кадра. 0 - запрещено 1 - разрешено	
4:3	SkipFrameMode	Если данный бит установлен в 1, то формат [4:3] игнорируется. R/W Режим пропуска кадра 0 – пропуск запрещен – пропускаются изображения, кроме I(IDR)-изображений. - пропускаются B-изображения (если в формате H.264/AVC nal_ref_idc == 0, то пропускаются).	
5	picUserDataEn-able	R/W Возможность выдачи пользовательских данных. Если данная опция включена, то пользовательские данные записываются в буфер пользовательских данных, который определен в UserDataBase. Если функция IframeSearchEn включена или включена функция SkipFrameMode, то данный флаг игнорируется, и пользовательские данные не выдаются.	
8:6	-	- Зарезервировано	
9	-	- Зарезервировано	
10	UserDataReport-Mode	R/W Устанавливается режим выдачи пользовательских данных. 0 – режим прерывания 1 – режим отключения прерывания Если данная опция установлена в 0, то по заполнению буфера пользовательских данных декодер выдает сигнал прерывания. Чтобы продолжить декодирование на стороне хоста следует сбросить бит прерывания. Если данная опция установлена в 1, то декодер выдает такой объем пользовательских данные, какой способен разместить буфер пользовательских данных. Если буфер полный, то декодер игнорирует оставшиеся пользовательские данные. На хост приходит информация о переполнении буфера пользовательских данных с помощью переменой userDataBufFull в формате отчета о пользовательских данных.	
11	SearchIDR	R/W Поиск кадра IDR вместо I-кадра. Данный флаг действителен только, если включен флаг IframeSearchEn flag ([2] бит данного регистра). 0: поиск I-кадра 1: поиск только кадра IDR	

Продолжение таблицы 1.337

Бит	Имя	Тип Функция	Команда
14:12	-	- Зарезервировано	
15	Mpeg2PicFlush	R/W Принудительный сброс индекса отображения буфера кадра, который задерживается без декодирования текущего изображения. 0: отключена 1: включена опция сброса	
17:16	RvDbkMode	R/W Установление режима фильтра деблокирования для потоков RV. 0: фильтр деблокирования включен для всех изображений. 1: фильтр деблокирования отключен для всех изображений. : фильтр деблокирования отключен для P- и B-изображений. : фильтр деблокирования отключен только для B-изображений	
18	DecRowEnable	R/W Данное поле используется в декодере H.264/AVC для декодирования с малой задержкой. 0: декодирование с малой задержкой отключено. 1: декодирование с малой задержкой включено.	

Таблица 1.338 – CMD_DEC_PIC_SKIP_NUM (0x198)

Бит	Имя	Тип Функция	Команда
15:0	DecSkipFra-meNum	R/W (вышел из употребления) Пользоваться данным значением НЕ СЛЕДУЕТ.	DEC_PIC_RUN

Таблица 1.339 – CMD_DEC_PIC_USER_DATA_BASE_ADDR (0x1AC)

Бит	Имя	Тип Функция	Команда
31:0	UseDataBase	R/W Адрес базы данных пользователя Подробности смотрите в описании структуры DecParam.	DEC_PIC_ RUN

Таблица 1.340 - CMD_DEC_PIC_USER_DATA_BUF_SIZE (0x1B0)

Бит	Имя	Тип Функция	Команда
31:0	UserDataBufSize	R/W Объем буфера пользовательских данных Подробности смотрите в описании структуры DecParam.	DEC_PIC_ RUN

Таблица 1.341 - CMD_DEC_PIC_NUM_ROWS (0x1B4)

Бит	Имя	Тип Функция	Команда
31:8	-	- Зарезервировано	DEC_PIC_ RUN
7:0	NumMbRows	R/W Количество рядов макроблоков (блока макроблоков), требуемых для генерирования первого сигнала прерывания с малой задержкой	

Таблица 1.342 – CMD_DEC_PIC_ROT_STRIDE (0x1B8)

Бит	Имя	Тип	Функция	Команда
0	RotBufStride	R/W	Шаг по индексу кадра повернутого после обработки	DEC_PIC_RUN

Таблица 1.343 – CMD_DEC_PIC_ROT_BOTTOM_Y (0x1E8)

Бит	Имя	Тип	Функция	Команда
31:0	RotBufBottomAddrY	R/W	Адрес буфера нижнего полукадра для повернутых изображений и/или выходных изображений, обработанных фильтром шумоподавления, по компоненте Y. Данное поле доступно только, если Tiled2LinearEn равно 2 (LINEAR_FIELD_MAP). Примечание – Функция шумоподавления поддерживается только для формата MPEG-2/4.	DEC_PIC_RUN

Таблица 1.344 – CMD_DEC_PIC_ROT_BOTTOM_CB (0x1EC)

Бит	Имя	Тип	Функция	Команда
31:0	RotBufBottomAddrCb	R/W	Адрес буфера нижнего полукадра для повернутых изображений и/или выходных изображений, обработанных фильтром шумоподавления, по компоненте Cb. Данное поле доступно только, если Tiled2LinearEn равно 2 (LINEAR_FIELD_MAP). Примечание – Функция шумоподавления поддерживается только для формата MPEG-2/4.	DEC_PIC_RUN

Таблица 1.345 – CMD_DEC_PIC_ROT_BOTTOM_CR (0x1F0)

Бит	Имя	Тип	Функция	Команда
31:0	RotBufBottomAddrCr	R/W	Адрес буфера нижнего полукадра для повернутых изображений и/или выходных изображений, обработанных фильтром шумоподавления, по компоненте Cr. Данное поле доступно только, если Tiled2LinearEn равно 2 (LINEAR_FIELD_MAP). Примечание - Функция шумоподавления поддерживается только для формата MPEG-2/4.	DEC_PIC_RUN

Таблица 1.346 – RET_DEC_PIC_MP2_BAR_LEFT_RIGHT (0x180)

Бит	Имя	Тип	Функция	Команда
15:0	PixelNumberStartOfRightBar	R	Значение синтаксиса pixel_number_start_of_right_bar информации из столбца в пользовательских данных ATSC Данное поле доступно только для декодера MPEG-2. This field is available only for MPEG-2 decoder.	DEC_PIC_RUN
31:16	PixelNumberEndOf LeftBar	R	Значение синтаксиса pixel_number_end_of_left_bar информации из столбца в пользовательских данных ATSC Данное поле доступно только для декодера MPEG-2.	

Таблица 1.347 – RET_DEC_PIC_MP2_BAR_TOP_BOTTOM (0x184)

Бит	Имя	Тип	Функция	Команда
15:0	LineNumber-StartOf Bot-tomBar	R	Значение синтаксиса line_number_start_of_bottom_bar информации из столбца в пользовательских данных ATSC Данное поле доступно только для декодера MPEG-2.	DEC_PIC_RUN
31:16	LineNumberEnd-Of TopBar	R	Значение синтаксиса line_number_end_of_top_bar информации из столбца в пользовательских данных ATSC Данное поле доступно только для декодера MPEG-2.	

Таблица 1.348 – RET_DEC_PIC_AVC_FPA_SEI0 (0x19C)

Бит	Имя	Тип	Функция	Команда
31:0	frame_packing_ arrangement_id	R	-1: Отсутствует информация SEI о расположении упакованных кадров 0 ~ 2^32-1: Идентификационный номер, который может использоваться для идентификации сообщения SEI о расположении упакованных кадров. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	DEC_PIC_RUN

Таблица 1.349 – RET_DEC_PIC_MP2_OFFSET1 (0x19C)

Бит	Имя	Тип	Функция	Команда
15:0	frame_centre_ vertical_offset	R	Сдвиг прямоугольника по горизонтали в блоках сэмпла 1/16. Более подробную информацию можно найти в спецификации к стандарту MPEG-2.	DEC_PIC_RUN
31:16	frame_centre_ horizontal_offset	R	Сдвиг прямоугольника по вертикали в блоках сэмпла 1/16. Более подробную информацию можно найти в спецификации к стандарту MPEG-2.	

Таблица 1.350 – RET_DEC_PIC_MP2_OFFSET2 (0x1A0)

Бит	Имя	Тип	Функция	Команда
15:0	frame_centre_ vertical_offset	R	Сдвиг прямоугольника по горизонтали в блоках сэмпла 1/16. Более подробную информацию можно найти в спецификации к стандарту MPEG-2.	DEC_PIC_RUN
31:16	frame_centre_ horizontal_offset	R	Сдвиг прямоугольника по вертикали в блоках сэмпла 1/16. Более подробную информацию можно найти в спецификации к стандарту MPEG-2.	

Таблица 1.351 – RET_DEC_PIC_MP2_OFFSET3 (0x1A4)

Бит	Имя	Тип	Функция	Команда
15:0	frame_centre_ vertical_offset	R	Сдвиг прямоугольника по горизонтали в блоках сэмпла 1/16. Более подробную информацию можно найти в спецификации к стандарту MPEG-2.	DEC_PIC_RUN
31:16	frame_centre_ horizontal_offset	R	Сдвиг прямоугольника по вертикали в блоках сэмпла 1/16. Более подробную информацию можно найти в спецификации к стандарту MPEG-2.	

Таблица 1.352 – RET_DEC_PIC_MP2_OFFSET_NUM (0x1A8)

Бит	Имя	Тип Функция	Команда
1:0	frame_centre_ offset_num	R Количество frame_centre_offset. Значения данного регистра лежат в диапазоне от 0 до 3, включительно	DEC_PIC_RUN

Таблица 1.353 – RET_DEC_PIC_AVC_FPA_SEI1 (0x1A0)

Бит	Имя	Тип	Функция	Команда
5:0	content_inter pretation_type	R	Показывает предполагаемую интерпретацию составляющих кадров. См. пункт D.2.25 ISO/IEC 14496-10	DEC_PI C_ RUN
12:6	frame_packing_arrange ment_type	R	Тип расположения пакетов кадров в соответствии в таблицей D-8 в пункте D.2.25ISO/IEC 14496-10.	
13	frame_packing_arrange ment_extension_flag	R	0 показывает то, что в сообщении SEI о расположении пакетов кадров нет никаких дополнительных данных. Более подробную информацию можно найти в пункте D.2.25ISO/IEC 14496-10.	
14	frame1_self_contained _flag	R	Указывает, ссылаются ли операции inter-прогнозирования при декодировании сэмплов составляющего кадра 1 в последовательности закодированных видеоданных на другой составляющий кадр 0. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
15	frame0_self_contained _flag	R	Указывает, ссылаются ли операции inter-прогнозирования при декодировании сэмплов составляющего кадра 0 в последовательности закодированных видеоданных на другой составляющий кадр 1. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
16	current_frame_is_frame0 _flag	R	Указывает текущий и последующий закодированные кадры при выводе данных. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
17	field_views_flag	R	1 показывает, что все изображения в текущей последовательности кодированных видеоданных закодированы как пары дополняющих друг друга полукадров. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
18	frame0_flipped_flag	R	Показывает, на каком из двух составляющих кадров происходит переключение. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
19	spatial_flipping_flag	R	Показывает то, на каком из двух составляющих кадров происходит переключение в пространстве. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
20	quincunx_sampling_flag	R	Показывает, выбирается ли компонентная матрица каждого цвета в каждом составляющем кадре в шахматном порядке. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	

Продолжение таблицы 1.353

Бит	Имя	Тип	Функция	Команда
21	frame_packing_arrangement_cancel_flag	R	Показывает, отменяет ли сообщение SEI о размещении пакетов кадров присутствие других предыдущих сообщений о размещении пакетов кадров SEI при выводе данных. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10	

Таблица 1.354 – RET_DEC_PIC_AVC_FPA_SEI2 (0x1A4)

Бит	Имя	Тип	Функция	Команда
14:0	frame_packing_arrangement_repetition_period	R	Показывает присутствие сообщения SEI о размещении пакетов кадров. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	DEC_PIC_ RUN
15	-	-	Зарезервировано	
19:16	frame1_grid_position_y	R	Определяется расположение левого верхнего сэмпла составляющего кадра 1 по вертикали ниже уровня пространственного ориентира.	
			Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
23:20	frame1_grid_position_x	R	Определяется расположение левого верхнего сэмпла составляющего кадра 1 по горизонтали справа от пространственного ориентира. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
27:24	frame0_grid_position_y	R	Определяется расположение левого верхнего сэмпла составляющего кадра 0 по вертикали ниже уровня пространственного ориентира. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10.	
31:28	frame0_grid_position_x	R	Определяется расположение левого верхнего сэмпла составляющего кадра 0 по горизонтали справа от пространственного ориентира. Более подробную информацию можно найти в пункте D.2.25 ISO/IEC 14496-10..	

Таблица 1.355 – RET_DEC_NUM_MB_ROWS (0x1B4)

Бит	Имя	Тип Функция	Команда
31:16	-	- Зарезервировано	DEC_PIC_ RUN
15:0	CurMbRows	R Выдача данных о рядах макроблоков текущих закодированных данных.	

Таблица 1.356 – RET_DEC_PIC_HRD_INFO (0x1B8)

Бит	Имя	Тип Функция	Команда
31:2	CpbMinus1	R cpb_cnt_minus1	DEC_PIC_ RUN
1	VclHrdParamFlag	R vcl_hrd_parameters_present_flag	
0	NalHrdParamFlag	R nal_hrd_parameters_present_flag Примечание: Данный регистр действует только для стандарта H.264/AVC.	

Таблица 1.357 - RET_DEC_PIC_SIZE (0x1BC)

Бит	Имя	Тип	Функция	Команда
31:16	Horizontal Size in Pixel	R	Размер декодированных данных по горизонтали, выраженный в пикселях	DEC_PIC_ RUN
15:0	Vertical Size in Pixel	R	Размер декодированных данных по вертикали, выраженный в пикселях. Обновление данного регистра происходит тогда, когда меняется информация о последовательности (разрешении).	

Таблица 1.358 – RET_DEC_PIC_FRAME_NUM (0x1C0)

Бит	Имя	Тип	Функция	Команда
15:0	DecFrameNum	R	Количество декодированных кадров. После кодирования одного кадра BIT-процессор увеличивает номер кадра и заносит количество декодированных кадров в данный регистр.	DEC_PIC_ RUN

Таблица 1.359 – RET_DEC_PIC_DISPLAY_IDX (0x1C4)

Бит	Имя	Тип	Функция	Команда
15:0	DecPicIdx	R	Индекс кадра изображения После декодирования одного кадра BIT-процессор возвращает в данный регистр индекс кадра изображения. Индекс кадра – это индекс массива адресов буферов кадра, который сообщается на стороне хоста по команде SET_FRAME_BUF. -1 (0xFFFF): в VPU больше нет изображений, которые нужно декодировать и отображать. -2 (0xFFFE): Произошел переход кадра, поэтому VPU не может отобразить его или в VPU после выполнения текущей команды PIC_RUN больше нет изображений для вывода на экран. -3 (0xFFFD): Произошла задержка отображения вследствие упорядочивания изображений	DEC_PIC_ RUN

Таблица 1.360 – RET_DEC_PIC_ERR_MB (0x1C8)

Бит	Имя	Тип	Функция	Команда
15:0	ErrMbNum	R	Номер макроблока, содержащего ошибку, в текущем декодированном изображении Если BIT-процессор обнаруживает ошибку в потоке данных, то он выполняет маскирование ошибок по макроблокам и возвращает номер замаскированного макроблока во всем изображении. Если данное значение равно 0, то это означает, что кадр был декодирован без ошибок.	DEC_PIC_RUN

Таблица 1.361 – RET_DEC_PIC_TYPE (0X1CC)

Бит	Имя	Тип	Функция	Команда
5:0	DecPicType	R	Вид закодированного в данный момент изображения 0 – I-изображение 1 – P-изображение 2 – B-изображение – D-изображение в видеокодеке MPEG-2, которое состоит только из коэффициентов DC, и S-изображение в видеокодеке MPEG4, которое является сокращением от слова Sprite, и используется для GMC. - некодированное P-изображение в режиме уплотненного десятичного кода MPEG4 [5:3] - DecTopPicType [2:0] - DecBotPicType или FramePicType Биты [5:3] показывают тип изображения в верхнем полукадре (для чересстрочного режима), а биты [2:0] показывают тип изображения в нижнем полукадре (для чересстрочного режима) или изображения кадра (для прогрессивного режима)	DEC_PIC_RUN
6	IDRFlag	R	Например, регистр может вывести 0 > 1 > 1 для определенных наборов изображений кадра. Это объясняется тем, что актуальными битами являются биты 000_000 > 000_001 > 000_001, и регистр выдает I-, P-, P-изображения в качестве выходных данных декодирования. Для потока кодека VC-1 данный регистр показывает тип закодированного изображения со значениями: 0 – I-изображение 1 – P-изображение 2 – BI-изображение 3 – B-изображение 4 – изображение P_SKIP Когда все биты DecPicType выравниваются до 1, то это означает, что из-за некоторого количества ошибок VPU не может декодировать равноценное изображение. Например, VPU выдает 7 в трех верхних битах для изображения верхнего полукадра и 0 – в трех нижних битах для изображения нижнего полукадра, если изображение первого полукадра было пропущено по ошибке, а следующим изображением полукадра является I-изображение. Это случай NPF (несдвоенного полукадра). Флаг IDR (H.264/AVC) [6] – Флаг IDR показывающий, является ли текущий кадр изображением IDR, или (в случае кадра полукадра), является ли первый полукадр изображением IDR. В случае кодирования MPEG-4, [6] – Закодированный VOP MPEG-4	
14:7	-	-	Зарезервировано	
15	DecFrameInfo	R	Информация о закодированном кадре 0 – декодированным кадром является NPF. 1 – декодированный кадр сдваивается с полукадром или кадром. Данный флаг действует только в случае декодирования H.264/AVC.	

Продолжение таблицы 1.361

Бит	Имя	Тип	Функция	Команда
17:16	NpfFieldInfo	R	Информация о полукадре индекса кадра изображения, выдаваемого по команде RET_DEC_PIC_DISPLAY_IDX(0x1C4) 0 – сдвоенный полукадр 1 - низ (пропущен верхний полукадр) 2 - верх (пропущен нижний полукадр) 3 - отсутствует (пропущен верхний и нижний полукадры) Данное поле действует только в случае декодирования H.264/AVC.	
18	InterlacedFrame	R	0 – прогрессивный кадр, который состоит из одного изображения кадра 1 – чересстрочный кадр, который состоит из двух изображений полукадра	
20:19	PictureStructure	R	MPEG-2 – структура изображения в расширении кода изображения MPEG-4 – чересстрочный режим на уровне видео-объекта H.264/AVC – информация MBAFF (флаг режима адаптивного полукадра/кадра макроблока), которая представлена только на бите [19] (на бите [20] – всегда 0). VC-1 – режим FCM (режим покадрового кодирования) в заголовке изображения	
21	TopFieldFirst	R	Данное поле показывает то, какой полукадр indicates which field is decoded first. It является действительным, только если флаг interlaceFrame установлен в 1. 0 – первым декодируется нижний полукадр 1 – первым декодируется верхний полукадр Действительно для VC1, AVS, MPEG-4 и MPEG-2.	
22	RepeatFirstField	R	Повторяется для отображения первого полукадра. Данный флаг действует только для VC-1, AVS и MPEG-2.	
24:23	ProgressiveFrame	R	MPEG-2 - progressive_frame в расширении кода изображения VC-1 – RPTFRM (число повторяющихся кадров) в заголовке изображения	
27:25	FieldSequence	R	field_sequence в расширении кода изображения кодека MPEG-2. Данное поле действует только в случае декодирования MPEG-2.	
27	PicStrPresent	R	Флаг pic_struct_present_flag в информации SEI pic_timing для H.264/AVC. Данное поле действует только в случае декодирования MPEG-2.	
28	FrameDCT	R	frame_pred_frame_dct в расширении кода изображения MPEG-2. Данное поле действует только в случае декодирования MPEG-2.	
29	ProgressSeq	R	progressive_sequence в расширении последовательности MPEG-2. Данное поле действует только в случае декодирования MPEG-2.	

Продолжение таблицы 1.361

Бит	Имя	Тип	Функция	Команда
31:28	AvcPicStruct	R	pic_struct в выдаваемой информации SEI pic_timing H.264/AVC. (Таблица D-1 спецификации H.264/AVC.) Если SEI pic_timing отсутствует, то pic_struct получается на основе части pic_struct пункта D.2.1. спецификации H.264/ AVC. Данное поле действует только в случае декодирования MPEG-2.	

Таблица 1.362 – RET_DEC_PIC_POST (0x1D0)

Бит	Имя	Тип Функция		Команда
0	RangeRed	R 0 – сокращение диапазона для данного кадра не проводится 1 - для данного кадра выполняется сокращение диапазона		DEC_PIC_RUN
		Примечание	Более подробно тема сокращения диапазона описывается на рис. А.1 - “Преобразование индексов для кадров VC-1”.	
2:1	MultiRes	R 2'b00 – полный размер по горизонтали, полный размер по вертикали. 2'b01 – половинный масштаб по горизонтали, полный размер по вертикали. 2'b10 – полный размер по горизонтали, половинный масштаб по вертикали. 2'b11 – половинный масштаб по горизонтали, половинный масштаб по вертикали.		

Таблица 1.363 – RET_DEC_PIC_SEQ_EXT_INFO (0x1D0)

Бит	Имя	Тип Функция	Команда
14:1	DisplayVertical-Size	Синтаксис display_vertical_size расширения отображения последовательности в спецификации к MPEG-2	DEC_PIC_RUN
28:15	DisplayHorizontal Size	Синтаксис display_horizontal_size syntax расширения отображения последовательности в спецификации к MPEG-2	

Таблица 1.364 – RET_DEC_PIC_SUCCESS (0x1D8)

Бит	Имя	Тип Функция	Команда
0	RetStatus	R 0 – команда DEC_PIC_RUN выполнена с ошибкой в заголовке при кодировании 1 – команда DEC_PIC_RUN выполнена без ошибок	DEC_PIC_RUN
1	-	- Зарезервировано	
2	LackOfSliceBuf	R 0 – буфер слайсов достаточного объема. 1 – буфер слайсов недостаточного объема. Данный флаг действует только для VC-1, AVS и MPEG-2.	
3	LackOfPsBuf	R 0 – буфер Ps достаточного объема. 1 – буфер Ps недостаточного объема.	

Продолжение таблицы 1.364

Бит	Имя	Тип	Функция	Команда
4	BitEmptyRoll-back	R	Данный флаг действует, только если RetStatus (нулевой бит данного регистра) равен 0, а StreamMode в BIT_BIT_STREAM_PARAM[4:3] настроен на режим повторного прогона. 0 – команда DEC_PIC_RUN не была выполнена из-за другой ошибки в синтаксисе. 1 – команда DEC_PIC_RUN не была выполнена из-за недостаточного размера битового потока для одного изображения в режиме повторного прогона.	
15:5	-	-	Зарезервировано	
16	ChunkReuse	R	H.264/AVC: Данный флаг действует только, когда выполняется декодирование AVC, и StreamMode в регистре BIT_BIT_STREAM_PARAM[4:3] настроен на режим picend.	
			0 – декодирование происходит в обычном порядке 1 – поток входных данных, предназначенных для декодирования, благодаря функции несдвоенного полукадра может использоваться повторно. MPEG4: Данный флаг действует только при декодировании MPEG-4, и, когда StreamMode в регистре BIT_BIT_STREAM_PARAM[4:3] установлен в режим picend. 0 - декодирование происходит в обычном порядке 1 - поток входных данных, предназначенных для декодирования, благодаря функции обнаружения упакованного кадра может использоваться повторно.	
17	SyntaxError	R	H.264/AVC, MPEG-2, MPEG-4: Данный флаг показывает имеет ли место ошибка синтаксиса в заголовке PPS (H.264/AVC), заголовке изображения (MPEG-2) или заголовке VOP (MPEG-4). 0 – ошибки нет. 1 – произошла ошибка синтаксиса.	
18	NoHeader	R	H.264/AVC, MPEG-2, MPEG-4: Данный флаг показывает, присутствует ли заголовок PPS (H.264/AVC), заголовок изображения (MPEG-2) или заголовок VOP (MPEG-4). 0 – заголовок есть. 1 – заголовка нет.	
19	FrameNumWrap	R	H.264/AVC: Данный флаг показывает, не произошел ли по неизвестной причине (ошибке) неожиданный возврат номера текущего кадра к меньшему значению. И если это произойдет, на стороне хоста нужно будет выполнить поиск IDR. 0 – возврата номера текущего кадра не произошло. 1 – произошел возврат номера текущего кадра.	

Продолжение таблицы 1.364

Бит	Имя	Тип	Функция	Команда
20	SequenceChange	R	H.264/AVC, MPEG-2, VC1, AVS, MPEG-4: Данный флаг указывает на то, что произошло изменение последовательности. Если произошло изменение последовательности, то обновляются буферные ресурсы для новой последовательности. Если для новой последовательности требуется больший объем буфера или количество буферов кадра, то на стороне хоста следует перераспределить буфер кадра и выполнить команду SEQ_INIT с заголовком новой последовательности (повторно задействовать битовый поток). Однако если на стороне хоста уже установлен буфер достаточного объема, VPU продолжит декодирование, на стороне хоста не придется предпринимать никаких действий для данной новой последовательности. Декодер H.264/AVC обнаруживает изменение размера изображения, флаг последовательности чересстрочных изображений и минимально требуемое количество буферов кадра. Декодеры VC1, AVS, MPEG-4 обнаруживают изменение размера изображения и флаг последовательности чересстрочных изображений.	
		0 – изменений в последовательности нет. 1 – имеет место изменение последовательности.		
21	MissingFrame	R	H.264/AVC: Данный флаг указывает, пропущен ли ссылочный кадр. В случае если из-за потери кадра отсутствует ссылочный кадр, то восстановление кадра может быть сорвано даже без каких-либо других ошибок синтаксиса. 0 – ссылочный кадр не пропущен 1 – ссылочный кадр пропущен	
31	MemoryAccess Violation	R	0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.365 – RET_DEC_PIC_DECODED_IDX (0x1DC)

Бит	Имя	Тип	Функция	Команда
15:0	DecodedPicIdx	R	Индекс декодированного кадра После декодирования одного кадра BIT-процессор возвращает в данный регистр индекс кадра изображения. Индекс кадра – это индекс массива адресов буферов кадра, который сообщается на стороне хоста по команде SET_FRAME_BUF. BIT-процессор возвращает -1 (0xFFFF), если по команде запуска обработки изображения это изображение не декодируется вследствие того, что для продолжения процесса декодирования нет достаточного количества буферов кадра. В случае VC-1 для декодирования одного кадра требуется 2 буфера кадра ввиду необходимости постобработки. BIT-процессор возвращает -2 (0xFFFE), если по команде обработки изображения это изображение не декодируется. Это может произойти по одной из причин: в заголовке изображения была некоторая ошибка, изображение не содержит данных для декодирования, или данное изображение пропускается, как изображение P_SKIP потока данных VC-1, или же включена опция пропуска. Примечание Когда регистр RET_DEC_PIC_TYPE (0X1CC) возвращает 4, а затем данный регистр возвращает -2, то на стороне хоста выяснится, что данное изображение является потоком VC-1, тип изображения которого - P_SKIP.	DEC_PIC_RUN

Таблица 1.366 – RET_DEC_PIC_CROP_LEFT_RIGHT (0x1E0)

Бит	Имя	Тип	Функция	Команда
15:0	RightOffset	R	Информация о сдвиге вправо при обрезке в декодере H.264/ AVC	DEC_PIC_RUN
31:16	LeftOffset	R	Информация о сдвиге влево при обрезке в декодере H.264/ AVC Примечание - Данный регистр действует только при декодировании H.264/AVC.	

Таблица 1.367 – RET_DEC_PIC_MODULO_TIME_BASE (0x1E0)

Бит	Имя	Тип	Функция	Команда
0	ModTimeBase	R	modulo_time_base в заголовке VOP. Данный регистр действует только при декодировании MPEG-4.	DEC_PIC_RUN

Таблица 1.368 – RET_DEC_PIC_CROP_TOP_BOTTOM (0x1E4)

Бит	Имя	Тип	Функция	Команда
15:0	BottomOffset	R	Информация о сдвиге снизу при обрезке в декодере H.264/AVC	DEC_PIC_RUN
31:16	TopOffset	R	Информация о сдвиге сверху при обрезке в декодере H.264/ AVC	
		Примечание	Данный регистр действует только при декодировании H.264/AVC.	

Таблица 1.369 – RET_DEC_PIC_VOP_TIME_INCREMENT (0x1E4)

Бит	Имя	Тип Функция	Команда
15:0	VopTimeIncr	R vop_time_increment в заголовке VOP. Данный регистр действует только при декодировании MPEG-4.	DEC_PIC_RUN

Таблица 1.370 – RET_DEC_PIC_ATSC_USER_DATA_INFO (0x1E8)

Бит	Имя	Тип Функция	Команда
3:0	ActiveFormat	R active_format (4-битное значение синтаксиса) в пользовательских данных AFD. По умолчанию задано значение 0000b. Данный регистр распространяется только на потоки H.264/AVC и MPEG-2.	DEC_PIC_RUN

Таблица 1.371 – RET_DEC_PIC_ASPECT (0x1F0)

Бит	Имя	Тип Функция	Команда
31:0	AspectRatio	R Данный регистр возвращает величину зрительного отношения последовательности, в которую входит декодируемое на текущий момент изображение. При изменении информации последовательности относительно форматного соотношения, данный регистр обновляется. Каждое возвращаемое значение можно найти в описании регистра RET_DEC_SEQ_ASPECT.	DEC_PIC_RUN

Таблица 1.372 – RET_DEC_PIC_FRATE_NR (0x1F4)

Бит	Имя	Тип Функция	Команда
31:0	FrameRateRes	R Для кодера AVC бит FrameRateRes обозначает time_scale в синтаксисе VUI H.264/AVC. В остальных случаях бит FrameRateRes обозначает остаточную частоту кадров декодированного изображения; число тактов генератора, функционирующего с частотой [FrameRateDiv] Гц. Например, Если [FrameRateDiv] = 1001 и [FrameRateRes] = 30000, то частота видеокадров = 30000 / 1001= 29.97 Гц. Если [FrameRateDiv] = 1 и [FrameRateRes] = 15, то частота видеокадров = 15 / 1 = 15 Гц.	DEC_PIC_RUN

Таблица 1.373 – RET_DEC_PIC_FRATE_DR (0x1F8)

Бит	Имя	Тип Функция	Команда
31:0	FrameRateDiv	R Для AVC бит FrameRateDiv обозначает num_units_in_tick в синтаксисе VUI H.264/AVC. В остальных случаях бит FrameRateDiv обозначает количество вариантов частоты кадров декодированного изображения, выраженной в герцах.	DEC_PIC_RUN

Таблица 1.374 - RET_DEC_PIC_VP8_PIC_REPORT (0x1E8)

Бит	Имя	Тип	Функция	Команда
31:24	GoldFrmIdx	R	Индекс «золотого» кадра VP8 для последующего декодирования	DEC_PIC_RUN
23:16	AltRefFrmIdx	R	Альтернативный индекс ссылочного кадра VP8 для последующего декодирования	
15:8	LastFrmIdx	R	Индекс последнего кадра VP8 для последующего декодирования	
2:1	VersionNumber	R	Номер версии VP8, 0 - 3 четыре профиля с разными уровнями сложности декодирования	
0	ShowFrame	R	0: если текущий кадр не предназначен для вывода на экран 1: если текущий кадр предназначен для вывода на экран	

Таблица 1.375 – RET_DEC_PIC_VUI_INFO (0x1EC)

Бит	Имя	Тип	Функция	Команда
0	FixedFrameRate-Flag	R	1: показывает то, что временной интервал между моментами времени вывода декодированных данных двух любых последовательных изображений в (команде вывода) ограничивается флагом fixed_frame_rate_flag в синтаксисе VUI H.264/AVC. 0: показывает то, что никаких ограничений, налагаемых на временной интервал между моментами времени вывода декодированных данных двух любых последовательных изображений в (команде вывода), нет. Примечание: Более подробно ознакомиться с данными флагами можно в пункте, в котором рассматривается синтаксис VUI	DEC_PIC_RUN
1	TimingInfoPre-sent	R	timing_info_present_flag в синтаксисе VUI H.264/AVC 1 : FixedFrameRateFlag действует. 0 : FixedFrameRateFlag не действует.	
4:2	ChromaSample-LocType Bottom-Field	R	chroma_sample_loc_type_bottom_field в синтаксисе VUI H.264/ AVC. Данный бит указывает расположение сэмплов компоненты цветности для нижнего полукадра.	
7:5	ChromaSample-LocType Top-Field	R	chroma_sample_loc_type_top_field в синтаксисе VUI H.264/ AVC. Данный бит указывает расположение сэмплов компоненты цветности для верхнего полукадра.	
8	ChromaLocInfoP-resent	R	chroma_loc_info_present_flag в синтаксисе VUI H.264/AVC 1: Биты ChromaSampleLocTypeTopField и ChromaSampleLocTypeTopField действуют. 0: Биты ChromaSampleLocTypeTopField и ChromaSampleLocTypeTopField не действуют.	

Продолжение таблицы 1.375

Бит	Имя	Тип	Функция	Команда
15:9	-	-	Зарезервировано	
23:16	ColorPrimaries	R	Синтаксис colour_primaries в параметре VUI в H.264/AVC	
24	ColorDescPresent	R	colour_description_present_flag в параметре VUI в H.264/AVC	
25	isExtSAR	R	Данный флаг указывает регистр Регистр RET_DEC_PIC_ASPECT представляет 8-битный aspect_ratio_idc или 32-битный extended_SAR. Если aspect_ratio_idc == extended_SAR mode, то данный флаг возвращает 1. Данный флаг действует только для H.264/ AVC.	
26	VidFullRange	R	video_full_range в параметре VUI в H.264/ AVC	
29:27	VidFormat	R	video_forma в параметре VUI в H.264/AVC	
30	VidSigTypePre-sent	R	video_signal_type_present_flag в параметре VUI в H.264/AVC	
31	VuiParamPresent	R	vui_parameters_present_flag в параметре VUI в H.264/AVC	

Таблица 1.376 – RET_DEC_PIC_VP8_SCALE_INFO (0x1F0)

Бит	Имя	Тип	Функция	Команда
31:30	HorScale	R	Фактор увеличения разрешения VP8 по горизонтали 0: Разрешение не увеличивается 1: увеличение до 5/4 2: увеличение до 5/3 3: увеличение в 2 раза	DEC_PIC_RUN
29:28	VerScale	R	Фактор увеличения разрешения VP8 по вертикали 0: Разрешение не увеличивается 1: увеличение до 5/4 2: увеличение до 5/3 3: увеличение в 2 раза	
27:14	PicWidth	R	Ширина кадра Vp8 в канале кодирования	
13:0	PicWidth	R	Высота кадра Vp8 в канале кодирования	

Таблица 1.377 – RET_DEC_PIC_MVC_REPORT (0x1D0)

Бит	Имя	Тип	Функция	Команда
0	ViewIdxDisplay	R	0 – Буфер кадра, соответствующий регистру RET_DEC_PIC_DISPLAY_IDX, является базовым представлением. 1 – Буфер кадра, соответствующий регистру RET_DEC_PIC_DISPLAY_IDX, не является базовым представлением.	DEC_PIC_RUN
1	ViewIdxDecoded	R	0 - Буфер кадра, соответствующий регистру RET_DEC_PIC_DECODED_IDX, является базовым представлением. 1 - Буфер кадра, соответствующий регистру RET_DEC_PIC_DECODED_IDX, не является базовым представлением.	

Таблица 1.378 – CMD_DEC_PARA_SET_TYPE (0x180)

Бит	Имя	Тип Функция	Команда
0	DecParaSetType	R/W Тип набора параметров 0 – набор параметров последовательности 1 – набор параметров изображения	DEC_PARA_SET

Таблица 1.379 – CMD_DEC_PARA_SET_SIZE (0x184)

Бит	Имя	Тип Функция	Команда
8:0	DecParaSetSize	R/W Размер байта RBSP параметров последовательности/изображения. Максимальный размер RBSP составляет 511 байт.	DEC_PARA_SET

Таблица 1.380 – RET_DEC_PARA_SET_SUCCESS (0x1C0)

Бит	Имя	Тип	Функция	Команда
0	RetStatus	R	0 – команда DEC_PARA_SET выполнена с ошибкой 1 – команда DEC_PARA_SET выполнена без ошибок	DEC_PARA_SET
31	MemoryAccess Violation	R	0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.381 – RET_DEC_BUF_FLUSH_SUCCESS (0x1C0)

Бит	Имя	Тип	Функция	Команда
0	RetStatus	R	0 – команда DEC_BUF_FLUSH выполнена с ошибкой 1 – успешно выполненная команда DEC_BUF_FLUSH	DEC_BUF_FLUSH
31	MemoryAccess Violation	R	0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.382 – CMD_ENC_SEQ_BB_START (0x180)

Бит	Имя	Тип	Функция	Команда
31:0	BitBufAddr	R/W	Адрес байта SDRAM буфера битового потока Объем буфера битового потока должен быть кратным 8 байтам. На стороне хоста данный регистр должен быть записан перед выполнением команды ENC_SEQ_INIT.	ENC_SEQ_INIT

Таблица 1.383 – CMD_ENC_SEQ_BB_SIZE (0x184)

Бит	Имя	Тип	Функция	Команда
13:0	BitBufSize	R/W	Объем битового потока, выраженный в килобайтах На стороне хоста данный регистр должен быть записан перед выполнением команды ENC_SEQ_INIT. Максимальный объем буфера битового потока равен 2^14-1 = 16383 кбайт	ENC_SEQ_INIT

Таблица 1.384 – CMD_ENC_SEQ_OPTION (0x188)

Бит	Имя	Тип	Функция	Команда
1:0	-	-	Зарезервировано	ENC_SEQ_INIT
2	AUDEnable	R/W	Возможность кодирования RBSP доступа к блоку разграничителя H.264/AVC. Если данный флаг установлен в 1, то BIT-процессор кодирует RBSP доступа к блоку разграничителя в начале каждого изображения. RBSP доступа к блоку разграничителя AUD используется для того, чтобы упростить процедуру обнаружения the границ изображения. В случае кодирования в MPEG-4/H.263 данный флаг игнорируется. В случае MVC перед небазовым представлением также добавляется блок NAL разграничителя представления и зависимости, описанный в ISO/IEC 13818-1.	
3	-	-	Зарезервировано	
4	InterviewEn	R/W	Если данное поле равно 1, то в кодере MVC для кодирования анкерного изображения в небазовом представлении используется опросное прогнозирование.	
5	RcConstIn-traQpEnable	R/W	Возможность настройки значений параметров квантизации I-кадров. См. информацию о поле IntraQp.	
6	RcQpMaxSetEn- able	R/W	Возможность использования фактора RcQpMax для контроля скорости. См. информацию о поле RcQpMax.	
7	RcGammaSetEn- able	R/W	Возможность использования фактора RcGamma для контроля скорости передачи. См. информацию о поле RcGamma.	
8	ParasetRefreshEn	R/W	Если данное поле равно 1, то перед каждым анкерным изображением добавляется SPS/PPS.	

Продолжение таблицы 1.384

Бит	Имя	Тип	Функция	Команда
9	PrefixNalEn	R/W	Если данное поле равно 1, то перед каждым блоком NAL слайса в базовом представлении кодер MVC добавляет блок NAL префикса.	
10	FieldFlag	R/W	Выбирается тип кодирования – прогрессивный или чересстрочный.	
			0 – прогрессивное кодирование кадров 1 – чересстрочное кодирование кадров	
11	FieldReference-Mode	R/W	Ссылочный режим для чересстрочного кодирования полукадров. Если бит FieldFlag равен 0, то данный флаг игнорируется. 0 – ссылка на полукадр той же четности 1 – ссылка на полукадр противоположной четности	

Таблица 1.385 – CMD_ENC_SEQ_COD_STD (0x18C)

Бит	Имя	Тип	Функция	Команда
3:0	EncCodStd	R/W	Кодирование стандарта кодирования. 4'h0 - H.264/AVC 4'h3 – Простой профиль MPEG-4 4'hB – короткий заголовок видеоданных MPEG-4 / H.263+ Данный регистр должен быть записан на стороне хоста перед выполнением команды SEQ_INIT.	ENC_SEQ_INIT

Таблица 1.386 – CMD_ENC_SEQ_SRC_SIZE (0x190)

Бит	Имя	Тип	Функция	Команда
15:0	PictureHeight	R/W	Кодирование высоты исходного изображения, выраженной в пикселях. Высота исходного изображения должна быть кратной 16, меньше или равна 1088. В случае H.263 (без приложения) размер изображения вычисляется следующим образом: (ширина, высота) = (128, 96), (176, 144), (352, 288), (704, 576) Когда хост записывает в данный регистр невыровненный размер (не кратный 16), VPU преобразует этот размер в выровненный по макроблокам для того, чтобы использовать его при конфигурировании буфера восстановленных кадров и прочих целей. При кодировании MPEG-4 невыровненное хостом значение кодируется непосредственно как video_object_layer_height в синтаксисе VOL	ENC_SEQ_INIT

Продолжение таблицы 1.386

Бит	Имя	Тип	Функция	Команда
31:16	PictureWidth	R/W	Кодирование ширины исходного изображения, выраженной в пикселях. Ширина исходного изображения должна быть кратной 16, меньше или равна 1920. Когда хост записывает в данный регистр невыровненный размер (не кратный16), VPU преобразует этот размер в выровненный по макроблокам для того, чтобы использовать его при конфигурировании буфера восстановленных кадров и прочих целей. При кодировании MPEG-4 невыровненное хостом значение кодируется напрямую как video_object_layer_width в синтаксисе VOL.	

Таблица 1.387 – CMD_ENC_SEQ_SRC_F_RATE (0x194)

Бит	Имя	Тип	Функция	Команда
15:0	FrameRateRes	R/W	Данное значение на 16 бит ниже частоты исходных кадров, что означает, что частота кадров & 0xFFFF.	ENC_SEQ_INIT
31:16	FrameRateDiv Minus1	R/W	Данное значение на 16 бит выше частоты исходных кадров, что означает, что частота кадров >> 16. 16 LSB [15:0] - это числитель, а 16 MSB [31:16] - это знаменатель при вычислении частоты кадров. Числитель означает число тактов в секунду, а знаменатель – число тактов между кадрами минус 1. Следовательно, частоту кадров можно найти по формуле: (числитель/(знаменатель+1)) равно (frameRateInfo & 0xffff)/((frameRate-Info >> 16)+1). Для представления частоты кадров можно задавать любое значение, даже значение с плавающей запятой. Например, если на стороне хоста устанавливается частота кадров со значением значение 30, [15:0] может иметь 30, а затем [31:16] становится 0. Чтобы задать частоту кадров равную 29.97 кадров/сек., данный регистр может выглядеть как 0x3e87530.	

Таблица 1.388 – CMD_ENC_SEQ_MP4_PARA (0x198)

Бит	Имя	Тип	Функция	Команда
0	DataPartEn	R/W	0 – запрещено разделение данных 1 – разрешено разделение данных	ENC_SEQ_INIT
1	RevVlcEn	R/W	0 – используются нормальные таблицы VLC 1 – используются обратимые таблицы VLC. Если бит DataPartEn равен 0, то данный бит игнорируется	
4:2	IntraDcVlcThr	R/W	Код пороговых величин VLC с коэффициентом DC Intra MPEG-4 Разрешен следующий диапазон значений: [0 ~ 7]	

Продолжение таблицы 1.388

Бит	Имя	Тип	Функция	Команда
3	mp4HecEnable	R/W	Кодирование с кодом расширения заголовка (HEC): such as modulo_time_base, vop_time_increment и vop_coding_type 0 – запрещается кодирование с HEC. 1 – разрешено кодирование с HEC.	
	mp4Verid	R/W	Версия уровня видео-объекта 0 – кодирование версии 2 с информацией заголовка quarter_sample, newpred_enable и reduced_resolution_vop_enable (по умолчанию) 1 - кодирование версии 1 без подобной информации заголовка	

Таблица 1.389 – CMD_ENC_SEQ_263_PARA (0x19C)

Бит	Имя	Тип	Функция	Команда
0	Annex T	R/W	0 – отключено приложение T 1 – включено приложение T	ENC_SEQ_INIT
1	Annex K	R/W	0 - отключено Приложение K 1 - Приложение K включено	
2	Annex J	R/W	0 - отключено Приложение J 1 – отключено Приложение J off	
3	Annex I	R/W	0 - отключено Приложение I 1 - отключено Приложение I * В текущей архитектуре приложение I в режиме кодирования не поддерживается. Поэтому данный флаг следует установить в 0.	

Таблица 1.390 – CMD_ENC_SEQ_264_PARA (0x1A0)

Бит	Имя	Тип	Функция	Команда
4:0	ChromaQpOffset	R/W	chroma_qp_index_offset в наборе параметров изображения В диапазоне от -12 до +12. Назначаются 5 бит дополнительного кода 1_0100: -12 1_0101: -11 … 0_1100: +12	ENC_SEQ_INIT
5	ConstIntraFlag	R/W	constrained_intra_pred_flag в наборе параметров изображения 0 – в intra-прогнозировании используются данные макроблока в моде inter. 1 - в intra-прогнозировании не используются данные макроблока в моде inter.	
7:6	DisableDeblk	R/W	disable_deblocking_filter_idc в заголовке слайса 0 – включается фильтр деблокирования – отключается фильтр деблокирования – включается фильтр деблокирования везде кроме краев слайса	
11:8	DeblkAlphaOffset	R/W	slice_alpha_c0_offset_div2 в заголовке слайса в диапазоне от -6 до +6 Назначаются 4 бита дополнительного кода	
15:12	DeblkBetaOffset	R/W	slice_beta_offset_div2 в заголовке слайса в диапазоне от -6 до +6 Назначаются 4 бита дополнительного кода	

Таблица 1.391 – CMD_ENC_SEQ_SLICE_MODE (0x1A4)

Бит	Имя	Тип	Функция	Команда
1:0	SliceMode	R/W	0 – один слайс на изображение – режим 0 для кодирования нескольких слайсов; слайс изменяется по числу бит закодированного слайса. – режим 1 для кодирования нескольких слайсов; слайс изменяется по числу закодированных макроблоков. – режим 2 для кодирования нескольких слайсов (только для H.264/ AVC); слайс изменяется по таблице последовательного кодирования макроблоков. MPEG-4 mode, a re-sync marker and a packet Header is inserted between slice boundaries. Режим H.263 с Приложением K = 0, заголовок GOB добавляется в начале каждого слоя GOB. Режим H.263 с Приложением K = 1, генерируется несколько слайсов. Режим H.264/AVC, генерируется RBSP на уровне нескольких слайсов. В режиме H.264/AVC, если бит SliceMode равен 3, то на стороне хоста через внешнюю память должен выдаться некоторый параметр (таблица run_length_minus1), определенный в BIT_PARA_BUF_ADDR.	ENC_SEQ_INIT
31:2	SliceSizeNum	R/W	Если бит SliceMode равен 1, то в данный регистр должна быть записана закодированная контрольная сумма одного слайса. Если бит SliceMode равен 2, то в данный регистр должно быть записано количество макроблоков одного слайса. Если бит SliceMode равен 3, то в данный регистр записывается количество групп слайсов - 1(диапазон действительный значений 1~255). (В спецификации H.264/AVC num_slice_groups_minus1) В режиме H.263 и MPEG-4, действительное битовое поле - [16:2]. Данный бит игнорируется, если SliceMode bit равен 0. В режиме H.263 с Приложением K = 0, данный бит игнорируется.	

Таблица 1.392 – CMD_ENC_SEQ_GOP_NUM (0x1A8)

Бит	Имя	Тип	Функция	Команда
5:0	EncGopNum	R/W	Задание значение GOP (группы изображений) для кодирования. I-изображение добавляется к каждому значению изображений GOP. Ограничение по GOP составляет 60. 0 - I, P, P, P, …(только первое изображение – I-изображение) 1 - I, I, I, …(не P-изображение) 2 - I, P, I, P, … 3 - I, P, P, I, P, P, I, …	ENC_SEQ_INIT

Продолжение таблицы 1.392

Бит	Имя	Тип	Функция	Команда
19:16	RcGopIQpOffset	R/W	Начальный сдвиг QP для I-изображения в GOP. Данная настройка может понизить значение QP I-изображение до значения, заданного пользователем в диапазоне от -4 до 4. Данное значение доступно для AvcEnc, а когда RcEnable равен 0 или RcGopIQpOffsetEn равен 0, оно игнорируется.	
20	RcGopIQpOff- setEn	R/W	Включенный флаг для начального сдвига QP для I-изображения в GOP. 0 - отключено (по умолчанию) 1 - отключено Данное значение доступно для AvcEnc и игнорируется, если RcEnable равен 0.	
28:21	IDRInterval	R/W	Интервальный режим изображений при IDR 0 – Только первое изображение - IDR. 1 - Все I-изображения - IDR. n – Каждое n-ое I-изображение - IDR.	

Таблица 1.393 – CMD_ENC_SEQ_RC_PARA (0x1AC)

Бит	Имя	Тип	Функция	Команда
0	RcEnable	R/W	Данный флаг включает контроль скорости передачи при кодировании MPEG-4. Если данный флаг установлен в 0, то значение PictureQs регистра используется в качестве шага квантизации во всей последовательности. Данный флаг игнорируется, если запущен кодер H.264/AVC.	ENC_SEQ_INIT
15:1	BitRate	R/W	Целевая битовая скорость, выраженная в килобитах в секунду, для кодера MPEG-4 Данное значение игнорируется, если RcEnable равно 0 или в текущий момент работает кодер H.264/AVC. Максимально допустимое значение равно 32767 (0x7FFF).	
30:16	InitDelay	R/W	Задержка при удалении начального буфера эталонного декодера, выраженная в миллисекундах (мс) Данное значение игнорируется, если бит RcEnable равен 0. Максимально допустимое значение равно 32767 (0x7FFF). 0 does not check Reference decoder buffer delay constraint	

П

родолжение таблицы 1.393

Бит	Имя	Тип	Функция	Команда
31	SkipDisable	R/W	Отключение автоматического пропуска изображения при контроле скорости. Если данный флаг равен 0, то BIT-процессор может пропустить одно изображение, если доступные биты не играют существенной роли в распределении бюджета битов. Если данный флаг равен 1, то BIT-процессор никогда не пропустит изображение, но закодированный битовый поток может быть переполнен сильнее, чем при скорости передачи целевых битов в последовательностях, которые сложно поддаются кодированию. Данный флаг игнорируется, если [RcEnable] равен 0.	

Таблица 1.394 – CMD_ENC_SEQ_RC_BUF_SIZE (0x1B0)

Бит	Имя	Тип	Функция	Команда
31:0	VbvBufSize	R/W	Объем буфера эталонного декодера, выраженный в битах Данное значение игнорируется, если бит RcEnable или InitDelay равны 0. Максимально допустимое значение равно 0x7FFF_FFFF. 0 – ограничение по объему буфера эталонного декодера не контролируется.	ENC_SEQ_INIT

Таблица 1.395 – CMD_ENC_SEQ_INTRA_REFRESH (0x1B4)

Бит	Имя	Тип	Функция	Команда
15:0	IntraMbRefresh Num	R/W	Число восстановленных макроблоков в режиме intra Данное значение должно быть меньше значения для закодированного потока (высота изображения * ширина изображения / 256) 0 – восстановление макроблоков в режиме intra не используется. N – N макроблоков, как минимум, для каждого изображения кодируются как в режиме intra.	ENC_SEQ_INIT

Продолжение таблицы 1.395

Бит	Имя	Тип	Функция	Команда
16	ConscIntraR-efreshEn	R/W	Режим восстановления последовательных макроблоков intra Данная опция действует только, когда бит IntraMbRefreshNum[15:0] не равен 0. 0 – опция ConscIntraRefreshEn отключена. Количество макроблоков IntraMbRefreshNum кодируются как макроблоки intra в рамках установленного интервала по размеру изображения. 1 – Количество IntraMbRefreshNum последовательных макроблоков кодируются как макроблоки intra.	
17	CountIntraMbEn	R/W	Разрешен контроль разрядов макроблоков Intra Если данная опция включена, то VPU не задействует макроблоки intra, исходя из выбранного режима при подсчете IntraM-bRefreshNum. Данная опция действует только, когда бит IntraMbRefreshNum[15:0] не равен 0. 0 – подсчет макроблоков intra запрещен. 1 - подсчет макроблоков intra разрешен.	
18	FieldSeqIntraR-efresh En	R/W	Разрешено восстановление последовательности полукадров в режиме intra Данная опция действует только, когда бит IntraMbRefresh- Num[15:0] не равен 0, и при чересстрочном кодировании. 0 – опция FieldSeqIntraRefreshEn отключена. 1 – происходит последовательное восстановление макроблоков Intra каждого полукадра.	

Таблица 1.396 – CMD_ENC_SEQ_INTRA_QP (0x1C4)

Бит	Имя	Тип	Функция	Команда
31:0	IntraQp	R/W	Параметр шага квантизации изображений кадра в режиме Intra для процесса кодирования. В режиме MPEG-4/H.263 диапазоном доступных значений является диапазон от 1 до 31. В режиме H.264/AVC диапазоном QP в режиме intra является диапазон от 0до 51. Если контроль скорости передачи отключен, то данное значение игнорируется.	ENC_SEQ_INIT

Таблица 1.397 – CMD_ENC_SEQ_RC_QP_MAX (0x1C8)

Бит	Имя	Тип	Функция	Команда
31:0	RcQpMax	R/W	Максимальное значение параметра шага квантизации для процесса контроля скорости передачи. В режиме MPEG-4/H.263 диапазоном доступных значений является диапазон от 3 до 31. В режиме H.264/AVC диапазоном доступных значений является диапазон от 13 до 51.	ENC_SEQ_INIT

Таблица 1.398 – CMD_ENC_SEQ_RC_GAMMA (0x1CC)

Бит	Имя	Тип	Функция	Команда
31:0	RcGamma	R/W	Гамма представляет собой коэффициент сглаживания в формуле оценки. Значение для гамма можно найти по формуле коэффициент * 32768, значение коэффициента выбирается из диапазона значений, лежащих в области 0 ≤ коэффициент ≤ 1. Если значение стремится к 0, то изменение Qp происходит постепенно. Если значение стремится к 1, то Qp изменяется резко. The default Gamma value is 0.75*32768.	ENC_SEQ_INIT

Таблица 1.399 – CMD_ENC_SEQ_RC_INTERVAL_MODE (0x1D0)

Бит	Имя	Тип	Функция	Команда
1:0	RcIntervalMode	R/W	Интервал обновления контроля скорости Параметр QP макроблока с помощью данной настройки может изменять каждый макроблок, кадр, слайс, или определенное число макроблоков для настройки уровня буфера битов. 0: нормальный режим MB 1/2: зарезервировано 3: режим MB-Num	ENC_SEQ_INIT
31:2	RcMbInterval	R/W	Значение интервала, когда RcIntervalMode равно 3.	

Таблица 1.400 – CMD_ENC_SEQ_INTRA_WEIGHT (0x1D4)

Бит	Имя	Тип	Функция	Команда
19:0	IntraCostWeight	R/W	Расходно-весовой коэффициент intra для распознавания типа Intra/Inter. По умолчанию он может быть нулевым. Если в данном регистре присутствует несколько значений W, а расход лучшего режима intra по Refine-Intra-Mode-Decision - ICOST, то FIC итоговый расход intra можно найти по следующей формуле: FIC = ICOST + W	ENC_SEQ_INIT

Таблица 1.400 – CMD_ENC_SEQ_ME_OPTION (0x1D8)

Бит	Имя	Тип	Функция	Команда
1:0	MeSearchRangeX	R/W	Режим диапазона поиска по горизонтали для оценки параметров движения 0: по горизонтали (-64 ~ 63) : по горизонтали (-48 ~ 47) : по горизонтали (-32 ~ 31) : по горизонтали (-16 ~ 15)	ENC_SEQ_INIT

Продолжение таблицы 1.400

Бит	Имя	Тип	Функция	Команда
3:2	MeSearchRangeY	R/W	Режим диапазона поиска по вертикали для оценки параметров движения 0: по вертикали (-48 ~ 47) 1: по вертикали (-32 ~ 31) 2: по вертикали (-16 ~ 15)	
4	MeUseZeroPmv	R/W	Опция PMV оценки параметров движения 0: в механизме оценки параметров движения используется PMV, который был выведен из соседнего вектора движения. 1: в механизме оценки параметров движения используется нулевой PMV. Если данное поле равно 1, то качество кодирования может быть хуже, чем при нулевом значении.	
8:5	MeBlkModeEn- able	R/W	Флаг включения режима блоков для оценки параметров движения. (Только для H.264/AVC). Но стороне хоста могут использоваться различные сочетания (поразрядное логическое «ИЛИ») каждого из приведенных ниже значений. 4'b0000 или 4'b1111: используется весь блок 4'b0001: разрешен блок 16x16 4'b0010: разрешен блок 16x8 4'b0100: разрешен блок 8x16 4'b1000: разрешен блок 8x8	

Таблица 1.401 – CMD_ENC_SEQ_RC_PARA_2 (0x1DC)

Бит	Имя	Тип	Функция	Команда
3:0	RateControlType	R/W	Задание типа контроля скорости передачи 0: Кодирование при постоянном Qp 1: Контроль скорости передачи осуществляется с CBR 2: Контроль скорости передачи осуществляется с ABR	ENC_SEQ_INIT
21:4	BitRateKBPS	R/W	Целевая скорость передачи битов, выраженная в кбит/с. Данное значение игнорируется, если регистр RATE_CONTROL_TYPE равен 0. Максимально допустимыми значениями являются 262 или 143.	
22	StrictCBR	R/W	Данный бит обеспечивает возможность использования данных заполнителя (вставка бит). Он служит для предотвращения недозаполнения буфера кодера (buf_level ниже 0). buf_level = buf_level - (bit_rate / frame_rate) сгенерированные биты Он работает только, когда бит RateControlType равен 1 (CBR)	

Таблица 1.402 – CMD_ENC_SEQ_QP_RANGE_SET (0x1E0)

Бит	Имя	Тип	Функция	Команда
5:0	MaxQp	R/W	Задание параметра MaxQp. Диапазон допустимых значений - от 0 до 51.	ENC_SEQ_INIT
6	MaxQpSetEnable	R/W	Разрешен параметр MaxQp.	
7	-	-	Зарезервировано	
13:8	MinQp	R/W	Задание параметра MinQp. Диапазон допустимых значений - от 0 до 51.	
14	MinQpSetEnable	R/W	Разрешен параметр MinQp.	
15	-	-	Зарезервировано	
21:16	MaxDeltaQp	R/W	Задание параметра MaxDeltaQp. Диапазон допустимых значений - от 0 до 51	
22	MaxDeltaQpSetEn-able	R/W	Разрешен параметр MaxDeltaQp.	
23	-	-	Зарезервировано	
29:24	MinDeltaQp	R/W	Задание параметра MinDeltaQp. Диапазон допустимых значений - от 0 до 51.	
30	Min-DeltaQpSetEnable	R/W	Разрешен параметр MinDeltaQp	
31	-	-	Зарезервировано	

Таблица 1.403 – CMD_ENC_SEQ_RC_MAX_INTRA_SIZE (0x1F0)

Бит	Имя	Тип	Функция	Команда
15:0	MaxIntraSize	R/W	Максимальное число бит кадра intra, выраженное в килобитах, предназначенное для контроля скорости передачи. Данное поле может ограничивать целевой бит для кадра intra в целях соответствия эффективности кодирования в реальном времени. Если возможность контроля скорости отключена, данный бит игнорируется.	ENC_SEQ_INIT
16	MaxIntraSizeEn	R/W	Включение флага для указания максимального размера кадра intra. 1: включено 0: отключено Данное значение игнорируется, если опция контроля скорости передачи отключена.	

Таблица 1.404 – RET_ENC_SEQ_SUCCESS (0x1C0)

Бит	Имя	Тип	Функция	Команда
0	RetStatus	R	0 – команда ENC_SEQ_INIT выполнена с ошибкой. 1 – команда ENC_SEQ_INIT выполнена без ошибок.	ENC_SEQ_ INIT
31	MemoryAccess Violation	R	0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.405 – RET_ENC_SEQ_END_SUCCESS (0x1C0)

Бит	Имя	Тип Функция	Команда
0	RetStatus	R 0 – команда ENC_SEQ_END выполнена с ошибкой 1 – команда ENC_SEQ_END выполнена без ошибок	ENC_SEQ_END
31	MemoryAccess Violation	R 0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.406 – CMD_ENC_PIC_SRC_INDEX (0x180)

Бит	Имя	Тип Функция	Команда
4:0	EncSrcFrameIn-dex	R/W Индекс буфера исходного кадра кодирования (максимальное значение - 31). Значение данного индекса должно быть выше, чем числа FrameBufNum, что позволит избежать использования зоны буфера восстановленных кадров.	ENC_PIC_ RUN

Таблица 1.407 – CMD_ENC_PIC_SRC_STRIDE (0x184)

	Имя	Тип Функция	Ком
1:0	EncSrcStride	R/W Размер шага по индексу для буфера исходных кадров. На стороне хоста должно быть записано значение размера шага по индексу самого буфера исходных кадров без учета вращения, выполненного в ходе предварительной обработки.	ENC_PIC_ RUN

Таблица 1.408 – CMD_ENC_PIC_QS (0x18C)

Бит	Имя	Тип Функция	Команда
31:0	PictureQs	R/W Параметр шага квантизации изображения (или параметр квантизации) в процессе кодирования В режиме MPEG-4/H.263 диапазоном допустимых значений является диапазон от 1 до 31. В режиме H.264/AVC диапазоном допустимых значений является диапазон от 0 до 51. Если опции контроля скорости передачи и ROI включены, и при этом RoiType равен 0, то данный регистр используется не для ROI. Если опция контроля скорости отключена, то BIT-процессор, используя данное значение, кодирует все макроблоки в текущем изображении. При настройке данного значения на поочередную обработку каждого изображения на стороне хоста может использоваться свой собственный алгоритм контроля скорости поочередно каждого изображения.	ENC_PIC_ RUN

Таблица 1.409 – CMD_ENC_PIC_ROT_MODE (0x190)

Бит	Имя	Тип Функция	Команда
3:0	PreRotMode	R/W Режим предварительного вращения PreRotMode[3:0] = {HorMir, VerMir, RotAng[1:0]} HorMir: отражение по горизонтали VerMir: отражение по вертикали RotAng[1:0] 0: 0 degree counterclockwise rotate 1: поворот на 90 градусов по часовой стрелке 2: поворот на 180 градусов по часовой стрелке 3: поворот на 270 градусов по часовой стрелке Если данное поле равно 4'b0000, то функция предварительного вращения отключается.	ENC_PIC_ RUN
4	LongBurstMode	R/W Возможность загрузки данных исходного кадра с более длинным пакетом данных в целях сокращения времени ожидания DMA. 0: режим с длинным пакетом данных отключен 1: режим с длинным пакетом данных включен	

Продолжение таблицы 1.409

Бит	Имя	Тип Функция	Команда
6:5	DmaMode	R/W Режим запроса с DMA для загрузки данных исходного кадра 0: запрос с DMA на загрузку 1 Мбайта за 1 такт 1: запрос с DMA на загрузку 2 Мбайтов за 1 такт 2: запрос с DMA на загрузку 8 Мбайтов за 1 событие запроса	
16	SelBigEndian	R/W Порядок байтов в буфере исходного кадра 0: байты исходного кадра располагаются в памяти в прямом порядке 1: байты исходного кадра располагаются в памяти в обратном порядке	
17	Sel32bit	R/W Формат слов в буфере исходного кадра 0: слова исходного кадра в памяти имеют 64-битный формат 1: слова исходного кадра в памяти имеют 32-битный формат	
18	CbCrInterleave	R/W Формат чересстрочной развертки в режиме цветности для буфера исходного кадра 0: Данные компонент Cb и Cr размещаются в памяти каждой компоненты цветности буфера исходного кадра по отдельности. 1: Данные компонент Cb и Cr размещаются в памяти каждой компоненты цветности буфера исходного кадра в чересстрочном порядке.	
20:19	-	- Зарезервировано	
21	SelNV21	R/W Определение NV21. Данный бит действует, только если разрешен бит CbCrInterleave. 0 – данные компоненты цветности чередуются в порядке CbCr. (NV12) 1 – данные компоненты цветности чередуются в порядке CrCb. (NV21)	

Таблица 1.410 – CMD_ENC_PIC_OPTION (0x194)

Бит	Имя	Тип Функция	Команда
0	PicSkipEn	R/W Флаг пропуска изображения Если данное поле установлено в 1, то биты EncSrcAddrY, EncSrcAddrCb и EncSrcAddrCr игнорируются, и кодируется одно пропущенное изображение. В этом случае, восстановленное на стороне декодера изображение представляет собой копию предыдущего изображения. Пропущенное изображение кодируется как P-изображение (Inter) несмотря на [EncGopNum]. Если следующий исходный кадр, подлежащий кодированию, недоступен, можно на стороне хоста установить данное поле в 1. Например, в случае кодирования кадра с частотой 5 Гц, если в выходных данных камеры отсутствует 4-ое изображение, то со стороны хоста необходимо 5 раз за секунду отправить команду ENC_PIC_RUN и установить флаг PicSkipEn - 0, 0, 0, 1, 0.	ENC_PIC_ RUN

Продолжение таблицы 1.410

Бит	Имя	Тип Функция	Команда
1	IdrPic	R/W Если данное поле равно 1, то исходное изображение несмотря на значение [EncGopNum] кодируется как изображение с IDR (мгновенное восстановление после декодирования) H.264/ AVC или как I(Intra)- изображение MPEG-4/H.263. Изображение с IDR – это I(Intra)-изображение с нулевым значением frame_num, все статусы декодирования которого (кроме входящих в список ссылочного изображения) сброшены. Первый кадр в битовом потоке - это автоматически закодированное изображение с IDR. После кодирования изображения с IDR все периодические расчеты I-изображения сбрасываются в исходное состояние. Например, если на стороне хоста на 18-ом кадре установлен флаг [IdrPic] и [EncGopNum] равен 15, то тип кодированного изображения определяется следующим образом: 1-й кадр: I (IDR - автоматически) 2-й кадр: P … 14-й кадр: P 15-й кадр: I 16-й кадр: P 17-й кадр: P 18-й кадр: I (IDR – установлено на стороне хоста) 19-й кадр: P … 32-й кадр: I 33-й кадр: P … В случае MPEG-4/H.263 I (Intra)-изображения достаточно для восстановления декодера. Чтобы ввести точку восстановления декодера в закодированный битовый поток на стороне хоста необходимо периодически устанавливать это поле в 1.	
5	UserDataEn	R/W Включение возможности выдачи пользовательских данных. Если данная опция включена, то пользовательские данные записываются в буфер пользовательских данных, заданный в UserDataBase.	
8	FieldRunMode	R/W Опция кодирования полукадра H.264/AVC; при кодировании H.264/AVC, если данный флаг установлен в 1 и FieldFlag (в регистре CMD_ENC_HEADER_FIELD_FLAG) установлен в 1 (кодирование полукадров), то VPU будет останавливаться (BusyFlag будет равным 0) по окончании кодирования изображения каждого полукадра (верхний/нижний).	

Таблица 1.411 – CMD_ENC_PIC_BB_START (0x198)

Бит	Имя	Тип Функция	Команда
15:0	EncPicBitBufStart	R/W Адрес байта буфера потока выходных изображений кодера Буфер битового потока должен быть выровнен по ширине шины памяти. Например, если на стороне хоста используется 64- битная по ширине шина, то буфер битового потока должен быть выровнен до 8 байтов. Данное значение действует только, если включена опция распределения динамического буфера кодера и опция сброса буфера потока. В данном случае данный бит показывает стартовый адрес данных битового потока.	ENC_PIC_ RUN

Таблица 1.412 – CMD_ENC_PIC_BB_SIZE (0x19C)

Бит	Имя	Тип Функция	Команда
15:0	EncPicBitBufSize	R/W Объем буфера битового потока для кодированного изображения, выраженный в килобайтах. Максимально допустимым объемом буфера битового потока является 256 Мбайт. Данное значение действует только, если включена опция распределения динамического буфера кодера и опция сброса буфера потока. В данном случае данный бит указывает окончание потока изображения.	ENC_PIC_ RUN

Таблица 1.413 – CMD_ENC_PIC_PARA_BASE_ADDR (0x1A0)

Бит	Имя	Тип Функция	Команда
31:0	EncPic-ParaBaseAddr	R/W Базовый адрес буфера параметра, расположенный во внешней памяти для выдачи информации кодирования в ходе кодирования изображения (пока недоступно.)	ENC_PIC_ RUN

Таблица 1.414 – CMD_ENC_PIC_SRC_ADDR_Y (0x1A8)

Бит	Имя	Тип Функция	Команда
31:0	SrcAddrY	R/W Адрес исходного кадра, подлежащего кодированию, в яркостном режиме На стороне хоста перед выполнением команды ENC_PIC_RUN следует записать данный регистр. Каждый раз перед началом кодирования изображения на стороне хоста в данный регистр должен устанавливаться стартовый адрес буфера кадра в SDRAM.	ENC_PIC_ RUN

Таблица 1.415 – CMD_ENC_PIC_SRC_ADDR_CB (0x1AC)

Бит	Имя	Тип Функция	Команда
31:0	SrcAddrCb	R/W Адрес исходного кадра, подлежащего кодированию, по компоненте Cb	ENC_PIC_ RUN

Таблица 1.416 – CMD_ENC_PIC_SRC_ADDR_CR (0x1B0)

Бит	Имя	Тип Функция	Команда
31:0	SrcAddrCr	R/W Адрес исходного кадра, подлежащего кодированию, по компоненте Cr	ENC_PIC_ RUN

Таблица 1.417 – CMD_ENC_PIC_SRC_BOTTOM_Y (0x1E8)

Бит	Имя	Тип Функция	Команда
31:0	SrcBottomAddrY	R/W Адрес буфера нижнего полукадра по компоненте Y исходных данных Данное поле доступно только, если Tiled2LinearEn равно 2 (LINEAR_FIELD_MAP).	ENC_PIC_ RUN

Таблица 1.418 – CMD_ENC_PIC_SRC_BOTTOM_CB (0x1EC)

Бит	Имя	Тип Функция	Команда
31:0	SrcBottomAddr-Cb	R/W Адрес буфера нижнего полукадра по компоненте Cb исходных данных Данное поле доступно только, если Tiled2LinearEn равно 2 (LINEAR_FIELD_MAP).	ENC_PIC_ RUN

Таблица 1.419 – CMD_ENC_PIC_SRC_BOTTOM_CR (0x1F0)

Бит	Имя	Тип Функция	Команда
31:0	SrcBottomAddrCr	R/W Адрес буфера нижнего полукадра по компоненте Cr исходных данных Данное поле доступно только, если Tiled2LinearEn равно 2 (LINEAR_FIELD_MAP).	ENC_PIC_ RUN

Таблица 1.420 – RET_ENC_PIC_FRAME_NUM (0x1C0)

Бит	Имя	Тип Функция	Команда
15:0	EncFrameNum	R Количество закодированных кадров После того, как в BIT-процессоре кодируется один кадр, количество кадров увеличивается, и это значение числа кадров заносится в данный регистр.	ENC_PIC_ RUN

Таблица 1.421 – RET_ENC_PIC_TYPE (0x1C4)

Бит	Имя	Тип Функция	Команда
0	EncPicType	R Тип только что закодированного изображения. 0 – I(Intra)-изображение 1 – P(Inter)-изображение	ENC_PIC_ RUN

Таблица 1.422 – RET_ENC_PIC_FRAME_IDX (0x1C8)

Бит	Имя	Тип Функция	Команда
15:0	RecPicIdx	R Индекс восстановленного кадра После кодирования одного кадра BIT-процессор возвращает в данный регистр индекс восстановленного кадра. Восстановленный кадр используется в качестве ссылочного для всех последующих кадров.	ENC_PIC_ RUN

Таблица 1.423 – RET_ENC_PIC_SLICE_NUM (0x1CC)

Бит	Имя	Тип Функция	Команда
14:0	EncSliceNum	R Количество слайсов кодируемого в текущий момент изображения	ENC_PIC_ RUN

Таблица 1.424 – RET_ENC_PIC_FLAG (0x1D0)

Бит	Имя	Тип Функция	Команда
0	EncPicFlag	R Данный флаг означает то, что в режиме сброса буфера объема буфера битового потока недостаточно для хранения данных одного кадра. Если данный флаг установлен, то кодирование текущего битового потока прерывается.	ENC_PIC_ RUN

Таблица 1.425 – RET_ENC_PIC_SUCCESS (0x01D8)

Бит	Имя	Тип Функция	Команда
0	RetStatus	R 0 – команда ENC_PIC_RUN выполнена с ошибкой 1 – команда ENC_PIC_RUN выполнена без ошибок	ENC_PIC_RUN
31	MemoryAccess Violation	R 0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.426 – CMD_ENC_HEADER_CODE (0x180)

Бит	Имя	Тип Функция	Команда
1:0	HeaderCode	R/W Кодирование кода заголовка Для MPEG-4 2'b00 – заголовок VOL 2'b01 – заголовок VOS 2'b10 – заголовок VO Для H.264/MVC 2'b00 - rbsp SPS 2'b01 - rbsp PPS 2'b10 - rbsp SPS поднабора для второго образа 2'b11 - rbsp PPS для второго образа Для H.263 команда ENC_HEADER игнорируется.	ENC_HEADER
2	FrameCrop-pingFlag	R/W frame_cropping_flag для rbsp SPS H.264/AVC; Данное поле действует только при кодировании rbsp SPS H.264/AVC. В синтаксисе SPS AVC это индекс флага (frame_cropping_flag), который показывает то, содержит ли набор закодированных параметров информацию об обрезке кадра. Если в ходе кодирования SPS H.264/AVC данный флаг установлен в 1, а бит HeaderCode установлен в 0, то VPU получает информацию об обрезке из регистров EncHdrFrameCropLeft/Right/Top/Bottom, а затем кодирует ее в синтаксис SPS. Если данный флаг установлен в ноль, то это означает, что в закодированной SPS отсутствует информация об обрезке кадра.	

Продолжение таблицы 1.426

Бит	Имя	Тип Функция	Команда
4	UserProfileLevel Enable	R/W Данный бит помогает определить, устанавливать ли profile_and_level_indication в заголовке VOS в соответствии с предварительно заданными значениями MPEG-4. Если данный бит равен 0, profile_and_level_indication кодируется с одним из нижеследующих значений: 8'b0000 0001: L1 (176x144 @15Hz) 8'b0000 0010: L2 (352x288 @15Hz) 8'b0000 0011: L3 (352x288 @30Hz) 8'b0000 0100: L4a (640x480 @30Hz) 8'b0000 0101: L5 (720x576 @25Hz) 8'b0000 0110: L6 (в прочих случаях) Если данный бит равен 1, то на стороне хоста можно установить профиль пользователя и его уровень с помощью следующих 8 битов - UserProfileLevelIndication Данный флаг выполняет в кодере H.264/AVC те же функции. Если он установлен в 1, то на стороне хоста можно установить level_idc с помощью следующих 8 бит - UserProfileLevelIndication. Если данный бит равен 0, то level_idc кодируется с помощью одного из этих значений. Если количество макроблоков в изображении (mbPicNum) ≤ 99 и mbPicNum × frameRate ≤ 1485, то level_idc равен 10. (QCIF) Если mbPicNum ≤ 396 и mbPicNum × frameRate ≤ 11880, то level_idc равен 20. (CIF) Если mbPicNum ≤ 1620 и mbPicNum × frameRate ≤ 40500, то level_idc равен 30. (SD) Если mbPicNum ≤ 3600 и mbPicNum × frameRate ≤ 108000, то level_idc равен 31. (HD)	
7:5	Зарезервировано	Если mbPicNum ≤ 8192 и mbPicNum × frameRate ≤ 245760, то level_idc равен 40. (Full HD) N/A Зарезервировано для последующего использования	
15:8	Levelldc	R/W level_idc для rbsp SPS H.264/AVC; Данное поле доступно для использования только в ходе кодирования rbsp SPS H.264/AVC.	
23:16	PpsId	R/W picture_parameter_set_id для rbsp PPS H.264/AVC; Данное поле доступно для использования только в ходе кодирования rbsp PPS H.264/AVC.	
28:24	SpsId	R/W sequence_parameter_set_id для rbsp SPS/PPS H.264/AVC; Данное поле доступно для использования только в ходе кодирования rbsp SPS/PPS H.264/AVC.	

Таблица 1.427 – CMD_ENC_HEADER_BB_START (0x184)

Бит	Имя	Тип Функция	Команда
15:0	EncHdrBitBufS-tart	R/W Адрес байта буфера выходного потока заголовков кодера Буфер битового потока должен быть выровнен до 8 байтов. Данное значение действует, только когда включены опция распределения динамического буфера кодера, а также опция сброса буфера потока данных. В этом случае он указывает стартовый адрес данных битового потока. Для H.263 команда ENC_HEADER игнорируется.	ENC_HEADER

Таблица 1.428 – CMD_ENC_HEADER_BB_SIZE (0x188)

Бит	Имя	Тип Функция	Команда
15:0	EncHdrBitBuf-Size	R/W Объем буфера битового потока для закодированного заголовка, выраженный в килобайтах. Максимальный объем буфера битового потока составляет 256 Мбайт. Данное значение действует, только когда включены опция распределения динамического буфера кодера, а также опция сброса буфера потока данных. В этом случае он указывает завершение потока изображений.	ENC_HEADER

Таблица 1.429 – CMD_ENC_HEADER_FRAME_CROP_H (0x18C)

Бит	Имя	Тип Функция	Команда
15:0	EncHdrFrame-CropRight	R/W Присутствует только в случае кодирования заголовка SPS H.264/AVC Данный параметр обозначает номер сэмпла в области обрезки в строке по правому краю. См. синтаксис frame_crop_right_offset в таблице SPS AVC. Данный параметр – это наименьший значащий бит (EncHdrFrameCropRight[0]), который всегда должен быть нулевым.	ENC_HEADER
31:16	EncHdrFrame-CropLeft	R/W Присутствует только в случае кодирования заголовка SPS H.264/AVC Данный параметр обозначает номер сэмпла в области обрезки в строке по левому краю. См. синтаксис frame_crop_left_offset в таблице SPS AVC. Данный параметр – это наименьший значащий бит (EncHdrFrameCropLeft[0]), который всегда должен быть нулевым.	

Таблица 1.430 – CMD_ENC_HEADER_FRAME_CROP_V (0x190)

Бит	Имя	Тип Функция	Команда
15:0	EncHdrFrame-Crop Bottom	R/W Присутствует только в случае кодирования заголовка SPS H.264/AVC Данный параметр обозначает номер сэмпла в области обрезки по нижнему краю в столбце изображения (выровненном по макроблокам). См. синтаксис frame_crop_bottom_offset в таблице SPS AVC. Данный параметр – это наименьший значащий бит (EncHdrFrameCropBottom[0]), который всегда должен быть нулевым.	ENC_HEADER
31:16	EncHdrFrame-CropTop	R/W Присутствует только в случае кодирования заголовка SPS H.264/AVC Данный параметр обозначает номер сэмпла в области обрезки по верхнему краю в столбце изображения (выровненном по макроблокам). См. синтаксис frame_crop_top_offset syntax в таблице SPS AVC. Данный параметр – это наименьший значащий бит (EncHdrFrameCropTop[0]), который всегда должен быть нулевым.	

Таблица 1.431 – CMD_ENC_HEADER_CABAC_MODE (0x194)

Бит	Имя	Тип Функция	Команда
0	CabacMode	R/W Выбор метода энтропийного кодирования, используемого в процессе кодирования. 0 - CAVLC 1 - CABAC	ENC_HEADER

Таблица 1.432 – CMD_ENC_HEADER_CABAC_INIT_IDC (0x198)

Бит	Имя	Тип Функция	Команда
1:0	CabacInitIdc	R/W Определяется индекс для определения таблицы инициализации, используемой в процессе инициализации для CABAC. Значение cabac_init_idc должно располагаться в диапазоне 0 ~ 2.	ENC_HEADER

Таблица 1.433 – CMD_ENC_HEADER_TRANSFORM_8X8 (0x19C)

Бит	Имя	Тип Функция	Команда
0	Transform8x8	R/W Устанавливается факт включения intra-прогнозирования 8x8 и преобразования 8x8. 0 - отключено intra-прогнозирование 8x8 и преобразование 8x8 1 - включено intra-прогнозирование 8x8 и преобразование 8x8	ENC_HEADER

Таблица 1.434 – CMD_ENC_HEADER_CHROMA_FORMAT (0x1A0)

Бит	Имя	Тип Функция	Команда
0	ChromaFormat	R/W Определяется тип формата цветности. 0 – формат YUV 4:2:0 1 – монохромный формат YUV 4:0:0	ENC_HEADER

Таблица 1.435 – CMD_ENC_HEADER_FIELD_FLAG (0x1A4)

Бит	Имя	Тип Функция	Команда
0	FieldFlag	R/W Выбирается тип кодирования – прогрессивный или чересстрочный. 0 – прогрессивное кодирование кадров. 1 – чересстрочное кодирование кадров.	ENC_HEADER
1	FieldReference-Mode	R/W Ссылочный режим для чересстрочного кодирования полукадров. Если бит FieldFlag равен 0, то данный флаг игнорируется. 0 – ссылка на полукадр той же четности 1 – ссылка на полукадр противоположной четности	

Таблица 1.436 – CMD_ENC_HEADER_PROFILE (0x1A8)

Бит	Имя	Тип Функция	Команда
1:0	ProfileIdc	R/W Параметр profile_idc H.264/AVC получается в результате использования каждого инструмента кодирования. Данное поле доступно только, если HeaderCode равен 0 . 0 – Базовый профиль – Главный профиль – Высокий профиль	ENC_HEADER

Таблица 1.437 – CMD_ENC_HEADER_SCALING_LIST (0x1AC)

Бит	Имя	Тип Функция	Команда
0	ScalingMatrixPre-sent	R/W seq/pic_scaling_matrix_present_flag для H.264/AVC 0 – списки вариантов масштабирования SPS/PPS отсутствуют. 1 – присутствует синтаксис масштабирующей матрицы SPS/PPS	ENC_HEADER
15:8	ScalingListPre-sent	R/W seq/pic_scaling_list_present_flag для H.264/AVC. В данном поле у каждой компоненты цветности есть 8-битовый флаг. Пользователь в своих целях может выполнять операцию «ИЛИ» с каждым из этих битов. 0 – списки масштабирования SPS/PPS отсутствуют. 1 – присутствует список масштабирования по яркостной компоненте блока intra 4x4 2 - присутствует список масштабирования по компоненте Cb блока intra 4x4 4 - присутствует список масштабирования по компоненте Cr блока intra 4x4 8 - присутствует список масштабирования по яркостной компоненте блока inter 4x4 16 - присутствует список масштабирования по компоненте Cb блока inter 4x4 32 - присутствует список масштабирования по компоненте Cr блока inter 4x4 64 - присутствует список масштабирования по яркостной компоненте блока intra 8x8 128 - присутствует список масштабирования по яркостной компоненте блока inter 8x8	

Продолжение таблицы 1.437

Бит	Имя	Тип Функция	Команда
23:16	ScalingListDe-fault	R/W Флаг UseDefaultScalingMatri x4x4/8x8 для H.264/AVC В данном поле у каждой компоненты цветности есть 8-битовый флаг. Пользователь в своих целях может выполнять операцию «ИЛИ» с каждым из этих битов. Более подробную информацию по списку масштабирования, заданному по умолчанию, можно найти в таблице 7-3/7-4 спецификации H.264/AVC. 0 – списки масштабирования SPS/PPS, заданные по умолчанию, не используются.	
		– для компоненты яркости блока intra 4x4 используется список масштабирования intra 4x4, заданный по умолчанию - для компоненты Cb блока intra 4x4 используется список масштабирования intra 4x4, заданный по умолчанию 4 - для компоненты Cr блока intra 4x4 используется список масштабирования intra 4x4, заданный по умолчанию 8 - для компоненты яркости блока inter 4x4 используется список масштабирования inter 4x4, заданный по умолчанию 16 - для компоненты Cb блока inter 4x4 используется список масштабирования inter 4x4, заданный по умолчанию 32 - для компоненты Cr блока inter 4x4 используется список масштабирования inter 4x4, заданный по умолчанию 64 - для компоненты яркости блока intra 8x8 используется список масштабирования intra 8x8, заданный по умолчанию 128 - для компоненты яркости блока inter 8x8 используется список масштабирования inter 8x8, заданный по умолчанию	

Таблица 1.438 – CMD_ENC_HEADER_QMAT_BASE (0x1B0)

Бит	Имя	Тип Функция	Команда
31:0	ScalingListParam-Base	R/W Используется только для кодирования заголовка H.264/AVC В данном поле указывается базовый адрес байтового массива таблицы списка масштабирования, который хранится в порядке растрового сканирования во внешней памяти. В байтовом массиве таблицы списка масштабирования должно содержаться 8 таблиц (4x4/8x8) для каждой цветовой компоненты: яркостная [16], Cb[16], Cr[16] intra 4x4 яркостная [16], Cb[16], Cr[16] inter 4x4 яркостная [64] inter 8x8, яркостная [64] intra 8x8	ENC_HEADER

Таблица 1.439 – RET_ENC_HEADER_SUCCESS (0x1C0)

Бит	Имя	Тип Функция	Команда
0	RetStatus	R 0 – команда ENC_HEADER выполнена с ошибкой 1 – команда ENC_HEADER выполнена без ошибок	ENC_HEADER
30:1	-	- Зарезервировано	
31	MemoryAccess Violation	R 0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.440 – CMD_ENC_PARA_SET_TYPE (0x180)

Бит	Имя	Тип Функция	Команда
1:0	EncParaSetType	R/W Код заголовка кодирования В формате MPEG-4: 2'b00 – заголовок VOL 2'b01 – заголовок VOS 2'b10 – заголовок VO В формате H.264/AVC: 2'b00 - rbsp SPS 2'b01 - rbsp PPS 2'b10 - rbsp SPS поднабора для второго образа 2'b11 - rbsp PPS для второго образа В формате H.263 команда ENC_HEADER игнорируется.	ENC_PARA_SET
2	FrameCrop-pingFlag	R/W Данный бит доступен только в случае кодирования заголовка SPS H.264/AVC. В синтаксисе SPS AVC это индекс флага (frame_cropping_flag), который указывает, содержится ли в наборе закодированных параметров информация об обрезке кадра. Если при кодировании SPS H.264/AVC данный флаг установлен в 1, и HeaderCode равен 0, блок VPU получает информацию об обрезке из регистров EncHdrFrameCropLeft/Right/Top/Bottom, и затем кодирует ее в синтаксис SPS. Если данный флаг установлен в ноль, то в закодированном SPS отсутствует информации об обрезке кадра.	
15:8	Levelldc	R/W level_idc формата H.264/AVC в SPS; данное поле доступно только при настройке параметров SPS H.264/AVC.	
23:16	PpsId	R/W picture_parameter_set_id формата H.264/AVC в PPS; данное поле доступно только при настройке параметров PPS H.264/ AVC.	
28:24	SpsId	R/W sequence_parameter_set_id формата H.264/AVC в seq_parameter_set_data; данное поле доступно только при настройке параметров SPS/PPS H.264/AVC.	

Таблица 1.441 – CMD_ENC_PARA_SET_CABAC_MODE (0x194)

Бит	Имя	Тип Функция	Команда
0	CabacMode	R/W Выбор метода энтропийного кодирования, используемого в процессе кодирования. 0 - CAVLC 1 - CABAC	ENC_PARA_SET

Таблица 1.442 – D_ENC_PARA_SET_CABAC_INIT_IDC(0x198)

Бит	Имя	Тип Функция	Команда
1:0	CabacInitIdc	R/W Устанавливается индекс для определения таблицы инициализации, используемой в процессе инициализации для CABAC. Значение функции cabac_init_idc должно укладываться в диапазон 0 ~ 2.	ENC_PARA_SET

Таблица 1.443 – CMD_ENC_PARA_SET_TRANSFORM_8X8 (0x19C)

Бит	Имя	Тип Функция	Команда
0	Transform8x8	R/W Определяется, разрешено ли intra-прогнозирование и трансформирование блоков 8x8. 0 – запрещена опция intra-прогнозирования и трансформирования блоков 8x8 1 – разрешена опция intra-прогнозирования и трансформирования блоков 8x8	ENC_PARA_SET

Таблица 1.444 – CMD_ENC_PARA_SET_CHROMA_FORMAT (0x1A0)

Бит	Имя	Тип Функция	Команда
0	ChromaFormat	R/W Определяется тип формата цветности. 0 – формат YUV 4:2:0 1 – монохромный формат YUV 4:0:0	ENC_PARA_SET

Таблица 1.445 – CMD_ENC_PARA_SET_FIELD_FLAG (0x1A4)

Бит	Имя	Тип Функция	Команда
0	FieldFlag	R/W Выбирается тип кодировании – прогрессивный или чересстрочный. 0 – прогрессивное кодирование кадра. 1 – чересстрочное кодирование кадра.	ENC_PARA_SET
1	FieldReference-Mode	R/W Ссылочный режим для чересстрочного кодирования полукадров. Если бит FieldFlag равен 0, то данный флаг игнорируется. 0 – ссылка на полукадр той же четности 1 – ссылка на полукадр противоположной четности	

Таблица 1.446 – CMD_ENC_PARA_SET_PROFILE (0x1A8)

Бит	Имя	Тип Функция	Команда
1:0	ProfileIdc	R/W Параметр profile_idc H.264/AVC получается при использовании каждого инструмента кодирования. Данное поле доступно, только если бит EncParaSetType равен 0. 0 – Базовый профиль – Главный профиль – Высокий профиль	ENC_PARA_SET

Таблица 1.447 – CMD_ENC_PARA_SET_SCALING_LIST (0x1AC)

Бит	Имя	Тип Функция	Команда
0	ScalingMatrixPresent	R/W seq/pic_scaling_matrix_present_flag формата H.264/AVC 0 – списки вариантов масштабирования SPS/PPS отсутствуют. 1 – присутствует синтаксис масштабирующей матрицы SPS/PPS	ENC_PARA_SET
15:8	ScalingListPresent	R/W seq/pic_scaling_list_present_flag формата H.264/AVC В данном поле у каждой компоненты цветности есть 8-битовый флаг. Пользователь в своих целях может выполнять операцию «ИЛИ» с каждым из этих битов. 0 – списки масштабирования SPS/PPS отсутствуют. 1 – присутствует список масштабирования по яркостной компоненте блока intra 4x4 2 - присутствует список масштабирования по компоненте Cb блока intra 4x4 4 - присутствует список масштабирования по компоненте Cr блока intra 4x4 8 - присутствует список масштабирования по яркостной компоненте блока inter 4x4 16 - присутствует список масштабирования по компоненте Cb блока inter 4x4 32 - присутствует список масштабирования по компоненте Cr блока inter 4x4 64 - присутствует список масштабирования по яркостной компоненте блока intra 8x8 128 - присутствует список масштабирования по яркостной компоненте блока inter 8x8	
23:16	ScalingListDefault	R/W Формат H.264/AVC UseDefaultScalingMatrix4x4/8x8Flag В данном поле у каждой цветовой компоненты есть 8-битовый флаг. Пользователь в своих целях может выполнять операцию «ИЛИ» с каждым из этих битов. Более подробную информацию по списку масштабирования, заданному по умолчанию, можно найти в таблице 7-3/7-4 спецификации H.264/AVC. 0 – списки масштабирования SPS/PPS, заданные по умолчанию, не используются. 1 - для компоненты яркости блока intra 4x4 используется список масштабирования intra 4x4, заданный по умолчанию 2 - для компоненты Cb блока intra 4x4 используется список масштабирования intra 4x4, заданный по умолчанию 4 - для компоненты Cr блока intra 4x4 используется список масштабирования intra 4x4, заданный по умолчанию 8 - для компоненты яркости блока inter 4x4 используется список масштабирования inter 4x4, заданный по умолчанию 16 - для компоненты Cb блока inter 4x4 используется список масштабирования inter 4x4, заданный по умолчанию 32 - для компоненты Cr блока inter 4x4 используется список масштабирования inter 4x4, заданный по умолчанию 64 - для компоненты яркости блока intra 8x8 используется список масштабирования intra 8x8, заданный по умолчанию 128 - для компоненты яркости блока inter 8x8 используется список масштабирования inter 8x8, заданный по умолчанию	

Таблица 1.448 – CMD_ENC_PARA_SET_QMAT_BASE (0x1B0)

Бит	Имя	Тип Функция	Команда
31:0	ScalingListParam-Base	R/W Используется только в целях настройки параметров H.264/AVC В данном поле указывается базовый адрес байтового массива таблицы списка масштабирования, который хранится в порядке растрового сканирования во внешней памяти. В байтовом массиве таблицы списка масштабирования должно содержаться 8 таблиц (4x4/8x8) для каждой цветовой компоненты: яркостная [16], Cb[16], Cr[16] блока intra 4x4 яркостная [16], Cb[16], Cr[16] блока inter 4x4 яркостная [64] блока inter 8x8, яркостная [64] блока intra 8x8	ENC_PARA_SET

Таблица 1.449 – RET_ENC_PARA_SET_SIZE (0x1C0)

Бит	Имя	Тип Функция	Команда
8:0	EncParaSetSize	R Размер байта RBSP набора параметров закодированной последовательности /изображения	ENC_PARA_SET

Таблица 1.450 – RET_ENC_PARA_SET_SUCCESS (0x1C4)

Бит	Имя	Тип Функция	Команда
0	RetStatus	R 0 – команда ENC_PARA_SET выполнена с ошибкой 1 – команда ENC_PARA_SET выполнена без ошибок	ENC_PARA_SET
31	MemoryAccess Violation	R 0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.451 – CMD_ENC_PARAM_CHANGE_ENABLE (0x180)

Бит	Имя	Тип Функция	Команда
0	GopNum-ChangeEn	R/W Возможность внесения изменений в количество GOP 0 - запрещено 1- разрешено	ENC_PARAM_ CHANGE
1	ConstIn-traQpChangeEn	R/W Возможность внесения изменений в QP в режиме intra с ограничением 0 - запрещено 1- разрешено Если RcConstIntraQpEnable в регистре CMD_ENC_SEQ_OPTION[5] равно 0, то данное значение игнорируется.	
2	BitRateChangeEn	R/W Возможность внесения изменений в битовую скорость 0 - запрещено 1- разрешено	
3	FrameRate-ChangeEn	R/W Возможность внесения изменений в частоту кадров 0 - запрещено 1- разрешено	
4	IntraRefreshNum ChangeEn	R/W Возможность внесения изменений в количество новых блоков intra 0 - запрещено 1- разрешено	
5	-	- Зарезервировано	
6	HECMod-eChangeEn	R/W Возможность включения режима HEC (только в кодере формата MPEG-4) 0 - запрещено 1- разрешено	

Таблица 1.452 – CMD_ENC_PARAM_CHANGE_GOP_NUM (0x184)

Бит	Имя	Тип Функция	Команда
15:0	GopNum	R/W Новое число GOP кодирования I-изображение добавляется к каждому значению изображений GOP 0 - I, P, P, P, …(только первое изображение – I-изображение) 1 - I, I, I, …(не P-изображение) 2 - I, P, I, P, … 3 - I, P, P, I, P, P, I, …	ENC_PARAM_ CHANGE

Таблица 1.453 – CMD_ENC_PARAM_CHANGE_INTRA_QP (0x188)

Бит	Имя	Тип Функция	Команда
31:0	IntraQp	R/W Параметр шага квантизации изображения кадра intra для процесса кодирования. В режиме MPEG-4/H.263 допускается диапазон значений от 1 до 31. В режиме H.264/AVC допускается диапазон значений от 0 до 51. Если функция управления скоростью передачи отключена, то данное значение игнорируется.	ENC_PARAM_ CHANGE

Таблица 1.454 – CMD_ENC_PARAM_CHANGE_BITRATE (0x18C)

Бит	Имя	Тип Функция	Команда
17:0	BitRate	R/W Скорость передачи целевого бита, выраженная в килобитах в секунду (кбит/с). Данное значение игнорируется, если бит RcEnable равен 0. Максимальное значение - 32767 (0x7FFF).	ENC_PARAM_ CHANGE

Таблица 1.455 – CMD_ENC_PARAM_CHANGE_F_RATE (0x190)

Бит	Имя	Тип Функция	Команда
15:0	FrameRateRes	R/W Остаточная частота исходных кадров при кодировании Это количество единиц времени генератора тактовых частот, работающего с частотой [FrameRateDiv] Гц Частота кадров = [FrameRateRes] / [FrameRateDiv] Если [EncCodStd] равен 11, и приложения I, J, K и T выключены (H.263 без PLUSPTYPE), то частота исходных кадров при кодировании должна составлять 29,97 ([FrameRateRes] = 30000, [FrameRateDiv] = 1001), вследствие того, что H.263 без PLUSPTYPE поддерживает только частоту исходных кадров, равную 29,97 Гц.	ENC_PARAM_ CHANGE
31:16	FrameRateDiv Minus1	R/W Число единиц частоты исходных кадров при кодировании, выраженной в герцах, минус 1 Значение [FrameRateDiv] получается при добавлении к этому значению единицы.	

Таблица 1.456 – CMD_ENC_PARAM_CHANGE_INTRA_REFRESH (0x194)

Бит	Имя	Тип Функция	Команда
15:0	IntraMbRefresh Num	R/W Число восстановленных макроблоков в режиме intra Данное значение должно быть меньше значения для закодированного потока (высота изображения * ширина изображения / 256) 0 – восстановление макроблоков в режиме intra не используется. N – N макроблоков, по меньшей мере, для каждого изображения кодируются как в режиме intra.	ENC_PARAM_ CHANGE

Таблица 1.457 – CMD_ENC_PARAM_CHANGE_HEC_MODE (0x19C)

Бит	Имя	Тип Функция	Команда
0	HECModeEnable	R/W Возможность кодирования HEC. (только в кодере MPEG-4) 0 – запрещено кодирование HEC. 1 – разрешено кодирование HEC.	ENC_PARAM_ CHANGE

Таблица 1.458 – RET_ENC_PAPAM_CHANGE_SUCCESS (0x1C0)

Бит	Имя	Тип Функция	Команда
0	RetStatus	R 0 – команда ENC_PARAM_CHANGE выполнена с ошибкой. 1 – команда ENC_PARAM_CHANGE выполнена без ошибки.	ENC_PARAM_ CHANGE
31	MemoryAccess Violation	R 0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.459 – CMD_ENC_ROI_MODE (0x180)

Бит	Имя	Тип Функция	Команда
0	RoiEn	R/W Флаг включения ROI. Данная команда доступна только, когда CodecStd - это AvcEnc, и когда включена функция контроля скорости. 0 - запрещено 1 - разрешено	ENC_ROI_INIT
1	RoiType	R/W Тип контроля скорости передачи ROI. 0 - QP хоста используется как для ROI, так и для не-ROI 1 - QP хоста используется для ROI	

Таблица 1.460 – CMD_ENC_ROI_NUM (0x184)

Бит	Имя	Тип Функция	Команда
3:0	RoiNum	R/W Общее количество ROI. Максимальный ROI равен 10.	ENC_ROI_INIT

Таблица 1.461 – CMD_ENC_ROI_POS_0 (0x188)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_0	R/W Позиция mb_x_start ROI_0	ENC_ROI_INIT
15:8	MbxEnd_0	R/W Позиция mb_x_end ROI_0	
23:16	MbyStart_0	R/W Позиция mb_y_start ROI_0. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_0	R/W Позиция mb_y_end ROI_0. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.462 – CMD_ENC_ROI_QP_0 (0x18C)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_0	R/W Параметр QP ROI_0	ENC_ROI_INIT

Таблица 1.463 – CMD_ENC_ROI_POS_1 (0x190)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_1	R/W Позиция mb_x_start ROI_1	ENC_ROI_INIT
15:8	MbxEnd_1	R/W Позиция mb_x_end ROI_1	
23:16	MbyStart_1	R/W Позиция mb_y_start ROI_1. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_1	R/W Позиция mb_y_end ROI_1. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.464 – CMD_ENC_ROI_QP_1 (0x194)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_1	R/W Параметр QP ROI_1	ENC_ROI_INIT

Таблица 1.465 – CMD_ENC_ROI_POS_2 (0x198)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_2	R/W Позиция mb_x_start ROI_2	ENC_ROI_INIT
15:8	MbxEnd_2	R/W Позиция mb_x_end ROI_2	
23:16	MbyStart_2	R/W Позиция mb_y_start ROI_2. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_2	R/W Позиция mb_y_end position ROI_2. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.466 – CMD_ENC_ROI_QP_2 (0x19C)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_2	R/W Параметр QP ROI_2	ENC_ROI_INIT

Таблица 1.467 – CMD_ENC_ROI_POS_3 (0x1A0)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_3	R/W Позиция mb_x_start ROI_3	ENC_ROI_INIT
15:8	MbxEnd_3	R/W Позиция mb_x_end ROI_3	
23:16	MbyStart_3	R/W Позиция mb_y_start ROI_3. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_3	R/W Позиция mb_y_end ROI_3. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.468 – CMD_ENC_ROI_QP_3 (0x1A4)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_3	R/W Параметр QP ROI_3	ENC_ROI_INIT

Таблица 1.469 – CMD_ENC_ROI_POS_4 (0x1A8)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_4	R/W Позиция mb_x_start ROI_4	ENC_ROI_INIT
15:8	MbxEnd_4	R/W Позиция mb_x_end ROI_4	
23:16	MbyStart_4	R/W Позиция mb_y_start ROI_4. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_4	R/W Позиция mb_y_end ROI_4. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.470 – CMD_ENC_ROI_QP_4 (0x1AC)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_4	R/W Параметр QP ROI_4	ENC_ROI_INIT

Таблица 1.471 – CMD_ENC_ROI_POS_5 (0x1B0)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_5	R/W Позиция mb_x_start ROI_5	ENC_ROI_INIT
15:8	MbxEnd_5	R/W Позиция mb_x_end ROI_5	
23:16	MbyStart_5	R/W Позиция mb_y_start ROI_5. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_5	R/W Позиция mb_y_end ROI_5. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.472 – CMD_ENC_ROI_QP_5 (0x1B4)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_5	R/W Параметр QP ROI_5	ENC_ROI_INIT

Таблица 1.473 – CMD_ENC_ROI_POS_6 (0x1B8)

Бит	*Имя*	*Тип Функция*	*Команда*
7:0	*MbxStart_6*	*R/W Позиция mb_x_start ROI_6*	*ENC_ROI_INIT*
15:8	*MbxEnd_6*	*R/W Позиция mb_x_end ROI_6*	
23:16	*MbyStart_6*	*R/W Позиция mb_y_start ROI_6. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.*	
31:24	*MbyEnd_6*	*R/W Позиция mb_y_end ROI_6. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.*	

Таблица 1.474 – CMD_ENC_ROI_QP_6 (0x1BC)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_6	R/W Параметр QP ROI_6	ENC_ROI_INIT

Таблица 1.475 – CMD_ENC_ROI_POS_7 (0x1C0)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_7	R/W Позиция mb_x_start ROI_7	ENC_ROI_INIT
15:8	MbxEnd_7	R/W Позиция mb_x_end ROI_7	
23:16	MbyStart_7	R/W Позиция mb_y_start ROI_7. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_7	R/W Позиция mb_y_end ROI_7. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.476 – CMD_ENC_ROI_QP_7 (0x1C4)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_7	R/W Параметр QP ROI_7	ENC_ROI_INIT

Таблица 1.477 – CMD_ENC_ROI_POS_8 (0x1C8)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_8	R/W Позиция mb_x_start ROI_8	ENC_ROI_INIT
15:8	MbxEnd_8	R/W Позиция mb_x_end ROI_8	
23:16	MbyStart_8	R/W Позиция mb_y_start ROI_8. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_8	R/W Позиция mb_y_end position ROI_8. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.478 – CMD_ENC_ROI_QP_8 (0x1CC)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_8	R/W Параметр QP ROI_8	ENC_ROI_INIT

Таблица 1.479 – CMD_ENC_ROI_POS_9 (0x1D0)

Бит	Имя	Тип Функция	Команда
7:0	MbxStart_9	R/W Позиция mb_x_start ROI_9	ENC_ROI_INIT
15:8	MbxEnd_9	R/W Позиция mb_x_end ROI_9	
23:16	MbyStart_9	R/W Позиция mb_y_start ROI_9. В случае чересстрочного кодирования mb_y_start должен быть установлен в mb_y_start_frame/2.	
31:24	MbyEnd_9	R/W Позиция mb_y_end ROI_9. В случае чересстрочного кодирования mb_y_end должен быть установлен в mb_y_end_frame/2.	

Таблица 1.480 – CMD_ENC_ROI_QP_9 (0x1D4)

Бит	Имя	Тип Функция	Команда
5:0	RoiQP_9	R/W Параметр QP ROI_9	ENC_ROI_INIT

Таблица 1.481 – RET_ENC_ROI_SUCCESS (0x1D8)

Бит	Имя	Тип Функция	Команда
0	RetStatus	R 0 – команда ENC_ROI_INIT выполнена с ошибкой. 1 – команда ENC_ROI_INIT выполнена без ошибок.	ENC_ROI_INIT

Таблица 1.482 – CMD_SET_FRAME_BUF_NUM (0x180)

Бит	Имя	Тип Функция	Команда
4:0	FrameBufNum	R/W Количество кадров, используемых для ссылочного или выходного переупорядочения. Данное значение при декодировании в регистре RET_DEC_SEQ_FRAME_NEED должно быть равно или превышать [FrameBufNeed]. На стороне хоста необходимо внести на адрес буфера памяти SDRAM [ParaBufAddr] соответствующий адрес SDRAM буфера кадров (компонент Y, Cb, Cr).	SET_FRAME_ BUF

Таблица 1.483 – CMD_SET_FRAME_BUF_STRIDE (0x184)

Бит	Имя	Тип Функция	Команда
10:0	LineStride	R/W Сдвиг шага по индексу строки в памяти кадров. Величина шага по индексу выражается в байтах.	SET_FRAME_ BUF

Таблица 1.484 – CMD_SET_FRAME_SLICE_BB_START (0x188)

Бит	Имя	Тип Функция	Команда
31:0	BitBufAddr	R/W Буфер для хранения адреса байта SLICE RBSP SDRAM. Данный буфер должен быть выровнен до 8 байтов. На стороне хоста перед выполнением команды SET_FRAME_BUF необходимо записать данный регистр	SET_FRAME_ BUF

Таблица 1.485 – CMD_SET_FRAME_SUBSAMP_A (0x188)

Бит	Имя	Тип Функция	Команда
31:0	SubSampBaseAd-drA	R/W Базовый адрес буфера для хранения изображения, разбитого на подсэмплы, с использованием аппаратного контурного фильтра или базовый адрес буфера для эталона CIME в аппаратной оценке ME (только для кодера). Объем буфера, как правило, равен 1/4 объема буфера восстановленных кадров. Биты SubSampBaseAddrA и SubSampBaseAddrB чередуются для того, чтобы сохранить выходные данные контурного фильтра и выходные данные эталона CIME. Информацию, касающуюся контурного фильтра, можно найти в техническом описании в разделе «Фильтр деблокирования/плавного перекрывания».	SET_FRAME_ BUF

Таблица 1.486 – CMD_SET_FRAME_SUBSAMP_B (0x18C)

Бит	Имя	Тип Функция	Команда
31:0	SubSampBaseAd-drB	R/W Базовый адрес буфера для хранения изображения, разбитого на подсэмплы, с использованием аппаратного контурного фильтра или базовый адрес буфера для эталона CIME в аппаратной оценке ME (только для кодера). Объем буфера, как правило, равен 1/4 объема буфера восстановленных кадров. Биты SubSampBaseAddrA и SubSampBaseAddrB чередуются для того, чтобы сохранить выходные данные контурного фильтра и выходные данные эталона CIME. Информацию, касающуюся контурного фильтра, можно найти в техническом описании в разделе «Фильтр деблокирования/плавного перекрывания».	SET_FRAME_ BUF

Таблица 1.487 – CMD_SET_FRAME_SLICE_BB_SIZE (0x18C)

Бит	Имя	Тип Функция	Команда
15:0	BitBufSize	R/W Объем буфера для хранения SLICE RBSP, выраженный в килобайтах. На стороне хоста, прежде чем выполнить команду SET_FRAME_BUF необходимо записать данный регистр. Максимальный объем буфера битового потока равен 2^16-1 = 65535 кбайт	SET_FRAME_ BUF

Таблица 1.488 – CMD_SET_FRAME_AXI_BIT_ADDR (0x190)

Бит	Имя	Тип Функция	Команда
31:0	AxiSecAddrBit	R/W Буфер для хранения предсказателя данных MVP (прогноз вектора движения)/ NB (соседнего блока) макроблоков верхнего уровня. На стороне хоста, прежде чем выполнять команду SET_FRAME_BUF, необходимо записать данный регистр. This can be set when use of secondary SRAM.	SET_FRAME_ BUF

Таблица 1.489 – CMD_SET_FRAME_AXI_IPACDC_ADDR (0x194)

Бит	Имя	Тип Функция	Команда
31:0	AxiSecAddrI-pAcDc	R/W Буфер для хранения данных intra-прогнозирования или коэффициентов AcDc макроблоков верхнего уровня. На стороне хоста, прежде чем выполнять команду SET_FRAME_BUF, необходимо записать данный регистр. Его можно установить при использовании вспомогательной SRAM.	SET_FRAME_ BUF

Таблица 1.490 – CMD_SET_FRAME_AXI_DBKY_ADDR (0x198)

Бит	Имя	Тип Функция	Команда
31:0	AxiSecAddrDbkY	R/W Буфер для хранения данных деблокирования верхних макроблоков в яркостном режиме. На стороне хоста, прежде чем выполнять команду SET_FRAME_BUF, необходимо записать данный регистр. Его можно установить при использовании вспомогательной SRAM.	SET_FRAME_ BUF

Таблица 1.491 – CMD_SET_FRAME_AXI_DBKC_ADDR (0x19C)

Бит	Имя	Тип Функция	Команда
31:0	AxiSecAddrDbkC	R/W Адрес буфера для хранения данных верхних макроблоков в цветовом режиме при использовании фильтра деблокирования. На стороне хоста, прежде чем выполнить команду SET_FRAME_BUF, необходимо записать данный регистр. Его можно установить при использовании вспомогательной SRAM.	SET_FRAME_ BUF

Таблица 1.492 – CMD_SET_FRAME_AXI_OVL_ADDR (0x1A0)

Бит	Имя	Тип Функция	Команда
31:0	AxiSecAddrOvl	R/W Адрес буфера для хранения данных перекрывания верхних макроблоков в VC-1. На стороне хоста, прежде чем выполнять команду SET_FRAME_BUF, необходимо записать данный регистр. Его можно установить при использовании вспомогательной SRAM.	SET_FRAME_ BUF

Таблица 1.493 – CMD_SET_FRAME_AXI_BTP_ADDR (0x1A4)

Бит	Имя	Тип Функция	Команда
31:0	AxiSecAddrBtp	R/W Адрес буфера в блоке на 256 байт для хранения данных битовой плоскости верхних макроблоков в VC-1. Если установлен бит SramUseBtpHost[13] регистра AXI_SRAM_USE (0x140), то прежде чем выполнять команду SET_FRAME_BUF на стороне хоста необходимо занести в данный регистр значение адреса. This can be set when use of secondary SRAM.	SET_FRAME_ BUF

Таблица 1.494 – CMD_SET_FRAME_CACHE_CONFIG (0x1AC)

Бит	Имя	Тип Функция	Команда
31:11	Зарезервировано	-	SET_FRAME_
10:9	CacheLineMerge	R/W Направление обработки в строке кэш-памяти и режим слияния 0: обработка по горизонтали без слияния 1: зарезервировано 2: обработка по вертикали без слияния 3: обработка по вертикали со слиянием	BUF
8:7	CacheWayLuma	R/W Объем кэш-памяти в режиме яркости 0, 1: 64x64 2, 3: 128x32	
6:5	CacheWayChro-ma	R/W Объем кэш-памяти в режиме цветности UV с чередованием 0, 1: 32x64 2: 64x32 3: 128x16 UV с разделением 0, 1: 32x32 2, 3: 64x16	
4 3	Зарезервировано CacheBurstMode	- R/W Пакетный режим работы кэш-памяти (размер строки) 0: пакет 4 1: пакет 8	

Продолжение таблицы 1.494

Бит	Имя	Тип Функция	Команда
2	CacheMapType	R/W Тип карты распределения кэш-памяти (форма строки) 0: линейная карта распределения 1: тайловая карта распределения Размер и форма строки (в байтах) Пакет 4, линейная карта распределения: 32x1 Пакет 4, тайловая карта распределения: 8x4 Пакет 8, линейная карта распределения: 64x1 Пакет 8, тайловая карта распределения: 8x8	
1	Зарезервировано	-	
0	CacheBypassMC	R/W Обход MC в кэш-памяти 0: в MC используется кэш. 1: в MC кэш не используется.	

Таблица 1.495 – CMD_SET_FRAME_MB_BUF_BASE (0x1B0)

Бит	Имя	Тип	Функция	Команда
31:0	MbDataBufBase	R/W	Буфер для декодированной информации первого блока VP8 Базовый адрес буфера для хранения информации первого декодированного блока VP8 (тип макроблока, данные MV). Данный бит упоминается при декодировании второго и прочих блоков коэффициентов. Объем буфера задается следующим образом: Объем буфера макроблока = 17 слов*4 байта (на слово)*общее число макроблоков Например, если во всех макроблоках находится 16 субмакроблоков в FHD (в худшем случае), то потребуется 554,88 кбайт (17*4*8160). Данный бит используется только при декодировании VP8.	SET_FRAME_ BUF

Таблица 1.496 – CMD_SET_FRAME_SUBSAMP_MVC_A (0x1B0)

Бит	Имя	Тип Функция	Команда
31:0	SubSampBase AddrMvcA	R/W Базовый адрес буфера для хранения изображения, разбитого на подсэмплы, с использованием аппаратного контурного фильтра или базовый адрес буфера для эталона CIME при аппаратной оценке ME (только для кодера). Объем буфера, как правило, равен 1/4 объема буфера восстановленных кадров. Биты SubSampBaseAddrMvcA и SubSampBaseAddrMvcB чередуются для того, чтобы сохранить выходные данные контурного фильтра и выходные данные эталона CIME. Информацию, касающуюся контурного фильтра, можно найти в техническом описании в разделе «Фильтр деблокирования/плавного перекрывания».	SET_FRAME_ BUF

Таблица 1.497 – CMD_SET_FRAME_SUBSAMP_MVC_B (0x1B4)

Бит	Имя	Тип Функция	Команда
31:0	SubSampBase AddrMvcB	R/W Базовый адрес буфера для хранения изображения, разбитого на подсэмплы, с использованием аппаратного контурного фильтра или базовый адрес буфера для эталона CIME при аппаратной оценке ME (только для кодера). Объем буфера, как правило, равен 1/4 объема буфера восстановленных кадров. Биты SubSampBaseAddrMvcA and SubSampBaseAddrMvcB чередуются для того, чтобы сохранить выходные данные контурного фильтра и выходные данные эталона CIME. Информацию, касающуюся контурного фильтра, можно найти в техническом описании в разделе «Фильтр деблокирования/плавного перекрывания».	SET_FRAME_ BUF

Таблица 1.498 – CMD_SET_FRAME_MAX_DEC_SIZE (0x1B8)

Бит	Имя	Тип Функция		Команда
31:16	MaxMBs	R/W Максимальное количество макроблоков в изображении		SET_FRAME_ BUF
15:8	MaxMBX	R/W Максимальное количество макроблоков по ширине изображения		
7:0	MaxMBY	R/W Максимальное количество макроблоков по высоте кадра Когда в VPU появляется последовательность, размер изображения которой больше, чем указано в настройке данного регистра, то блок VPU фиксирует изменение разрешения. Максимальный размер декодирования можно изменить на стороне хоста. Например, чтобы декодировать изображение размером до 1920x1088, в данном регистре нужно установить 0x1FE0_78_44. MaxMBs: 120 x 68 = 8160 (0x1FE0) MaxMBX: 1920 / 16 = 120 (0x78) MaxMBY: 1088 / 16 = 68 (0x44)		
		Примечание	Если на стороне хоста установлен 0 для всех полей, то VPU пропускает фиксацию изменения разрешения.	

Таблица 1.499 – CMD_SET_FRAME_DELAY (0x1BC)

Бит	Имя	Тип Функция	Команда
31:0	FrameDelay	R/W Номер кадра с задержкой отображения (только для формата H.264/AVC) С помощью данного регистра на стороне хоста можно установить величину задержки отображения Если на стороне хоста данный регистр установлен в 2, то VPU после декодирования 2 кадров возвращает индекс отображения (>=0). Необходимо учитывать то, что VPU не может гарантировать порядок отображения или другие функции в случае, если на стороне хоста данный регистр установлен в значение меньшее, чем значение регистра RET_DEC_SEQ_FRAME_DELAY. По умолчанию данное значение равно -1, что означает использование величины задержки отображения кадра, которая рассчитывается внутри VPU.	SET_FRAME_ BUF

Таблица 1.500 – RET_SET_FRAME_SUCCESS (0x1C0)

Бит	Имя	Тип	Функция	Команда
0	RetStatus	R	0 – команда SET_FRAME_BUF выполнена с ошибкой 1 – команда SET_FRAME_BUF выполнена без ошибок	SET_FRAME_ BUF
31	MemoryAccess Violation	R	0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

Таблица 1.501 – RET_FW_VER_NUM (0x1C0)

Бит	Имя	Тип	Функция	Команда
31:16	ProductID	R	Идентификатор продукта с 16-битным гексакодом.	GET_FW_ VER-SION
15:0	VersionID	R	Идентификатор продукта с 16-битным гексакодом.	

Таблица 1.502 – RET_FW_CODE_REV (0x1C4)

Бит	Имя	Тип	Функция	Команда
31:0	CodeRevision	R	Номер версии микро-ПО	GET_FW_ VER-SION

Таблица 1.503 – RET_SLEEP_WAKE_SUCCESS (0x1C0)

Бит	Имя	Тип	Функция	Команда
0	RetStatus	R	0 – команда SLEEP/WAKE выполнена с ошибкой 1 – команда SLEEP/WAKE выполнена без ошибок	SLEEP/WAKE
31	MemoryAccess Violation	R	0 – нарушения доступа к памяти не было 1 – произошло нарушение доступа к памяти	

1.3.6.4.2.4 Регистры GDI

1.3.6.4.2.4.1 Обзор регистров GDI

В блоке VPU есть свой собственный интерфейс DMA, который называется GDI (универсальный интерфейс прямого доступа к памяти), предназначенный для эффективного доступа к внешней памяти. При декодировании все регистры устанавливаются автоматически с помощью микро-ПО BIT-процессора, в зависимости от количества буферов кадра и справочных указателей. Например, если для декодирования потока требуется 5 буферов кадра, то в BIT-процессоре с помощью микро-ПО устанавливаются регистры GDI от GDI_INFO_xx до GDI_INFO_xx + 4 * 0x14 для пяти буферов кадра.

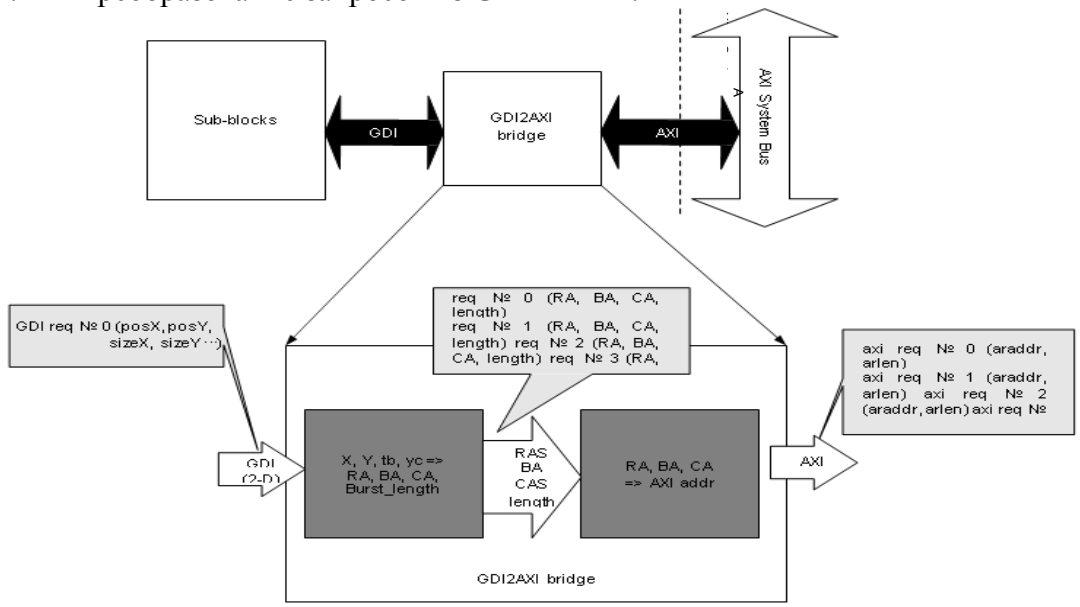
Если используется постпроцессор – вращение или шумоподавление (только для MPEG-2/4), то есть два способа настройки регистров для этого буфера кадра в соответствии со значением 16-го бита из таблицы 2.31 - “BIT_FRAME_MEM_CTRL (0x110)”.

Если 16-й бит из таблицы 2.31 - “BIT_FRAME_MEM_CTRL (0x110)” равен 1, то на стороне хоста следует вручную настроить регистры GDI_INFO_xx + 5 * 0x14 для буфера постпроцессора.

Если 16-й бит из таблицы 2.31 - “BIT_FRAME_MEM_CTRL (0x110)” равен 0, то микро-ПО настроит следующие регистры:

- CMD_DEC_PIC_ROT_MODE,
- CMD_DEC_PIC_ROT_INDEX,
- CMD_DEC_PIC_ROT_ADDR_Y,
- CMD_DEC_PIC_ROT_ADDR_CB,
- CMD_DEC_PIC_ROT_ADDR_CR,
- CMD_DEC_PIC_ROT_STRIDE.

Протокол интерфейса GDI представляет собой формат доступа к данным на основе 2D, который эффективен при необходимости доступа к данным в формате 2D, таким как видеоизображение. Например, параметрами запроса являются позиции X, Y и размер X, Y. Запросы GDI преобразуются в запросы AXI при их выдаче во внешнюю шинную систему (см. рисунок 2.2 - “Преобразование запросов из GDI в AXI”.



Sub-blocks – Субблоки
GDI2AXI bridge – Мост GDI-AXI
AXI System Bus – Системная шина AXI
RAS BA CAS length – Длина RAS BA CAS
GDI2AXI bridge – Мост GDI-AXI

Рисунок 1.151 – Преобразование запросов из GDI в AXI

Преобразование из GDI в AXI осуществляется в два этапа: адреса RAS, BA и CAS используются в качестве промежуточного формата при транслировании запросов от GDI к AXI. На первом этапе запрос GDI преобразуется в адрес и длину пакета RAS, BA, CAS. На втором этапе эти данные конвертируются в адрес и длину пакета AXI.

По этим адресам RAS, BA и CAS на стороне хоста можно легко понять, как сформирована карта распределения кадра и какой может быть эффективность.

Адреса RAS, BA, CAS генерируются из позиции X и Y с помощью битовой карты. При этом задействуются следующие регистры: GDI_XY2_CAS_n, GDI_XY2_BA_n, GDI_XY2_RAS_n и GDI_XY2_RBC_CONFIG. Логика построения битовой карты представлена на рисунке 2.3 и реализуется по следующим этапам:

- с помощью регистра XY_sel выбирается сигнал X или Y;
- с помощью bit_sel[3:0] в отобранном сигнале выбирается один бит;
- этот сигнал инвертируется с помощью сигнала tb;
- с помощью 1'b0 выбирается мультиплексор;
- инвертируются заключительные выходные данные.

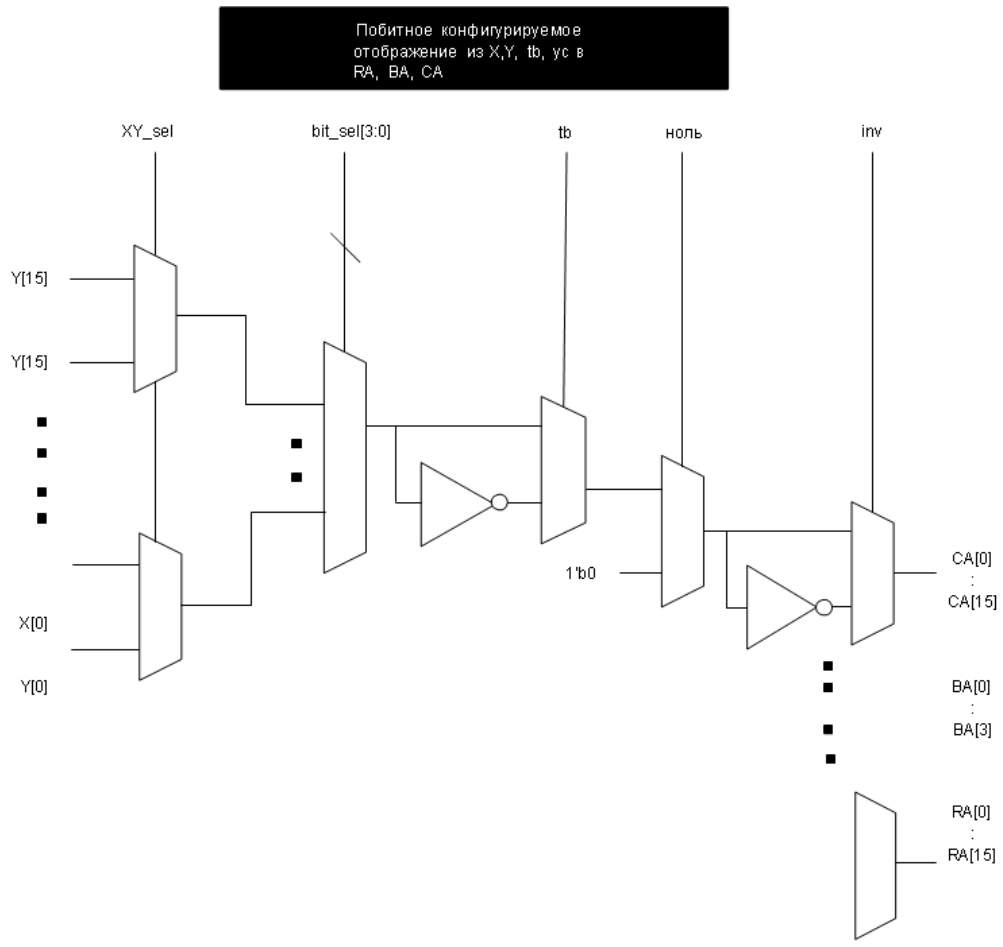


Рисунок 1.152 - Функция преобразования в RA, BA и CA

Адреса RAS, BA, CAS используются при генерировании адреса AXI. Адрес AXI также конфигурируется с помощью побитного отображения. Логика построения битовой карты представлена на рисунке xx. Данная процедура осуществляется в несколько этапов:

- Среди RAS, BA, CAS выбирается один сигнал.
- В отобранном сигнале выбирается один бит.

Данные логики обычно применяются для каждого бита адреса AXI с помощью регистров GDI_RBC2_AXI_n.

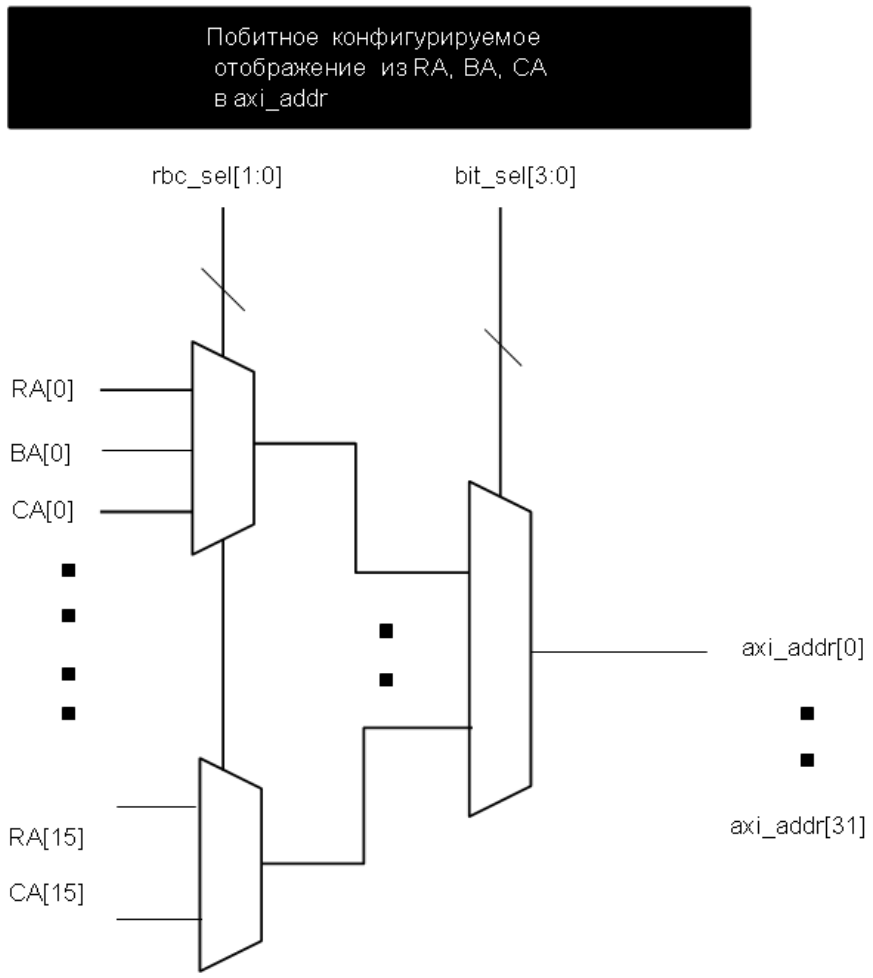


Рисунок 1.153 - Функция преобразования в адрес AXI

Таблица 1.503 – Сводные данные по регистру GDI

Адрес	Тип	Имя	Описание	MB/PIC
BASE+0x1000	RW	GDI_PRI_RD_PRIOR_L	Приоритет канала считывания [31:0]	PIC
BASE+0x1004	RW	GDI_PRI_RD_PRIOR_H	Приоритет канала считывания [63:32]	PIC
BASE+0x1008	RW	GDI_PRI_WR_PRIOR_L	Приоритет канала записи [31:0]	PIC
BASE+0x100C	RW	GDI_PRI_WR_PRIOR_H	Приоритет канала записи [63:32]	PIC
Зарезервировано				
BASE+0x1018	RW	GDI_SEC_RD_PRIOR_L	Приоритет канала считывания [31:0]	PIC
BASE+0x101C	RW	GDI_SEC_RD_PRIOR_H	Приоритет канала считывания [63:32]	PIC
BASE+0x1020	RW	GDI_SEC_WR_PRIOR_L	Приоритет канала записи [31:0]	PIC PIC
BASE+0x1024	RW	GDI_SEC_WR_PRIOR_H	Приоритет канала записи [63:32]	
Зарезервировано				
BASE+0x1030	RW	GDI_SEC_CLIENT_EN	Вспомогательная выборка	PIC
Зарезервировано				
BASE+0x1038	W	GDI_PIC_INIT_HOST	Генерирование на стороне хоста сигнала	PIC
BASE+0x1040	R	GDI_CONFIG_REPORT_0	инициализации изображения Название и версия продукта	PIC
BASE+0x1044	R	GDI_CONFIG_REPORT_1	Конфигурирование интерфейса, определение аппаратного обеспечения	PIC
BASE+0x1048	R	GDI_CONFIG_REPORT_2	Определение стандарта	PIC

Продолжение таблицы 1.503

Адрес	Тип	Имя	Описание	MB/PIC
BASE+0x104C	R	GDI_CONFIG_REPORT_3	Детальное определение стандарта	PIC
BASE+0x1050	R	GDI_CONFIG_REPORT_4	Определение в cnm_define	PIC
BASE+0x1054	R	GDI_CONFIG_REPORT_5	Дата конфигурирования отчета	PIC
BASE+0x1058	R	GDI_CONFIG_REPORT_6	Новая версия конфигурирования отчета	PIC
BASE+0x105C	R	GDI_CONFIG_REPORT_7	Тип конфигурирования отчета	PIC
Зарезервировано				
BASE+0x1060	W	GDI_PINFO_REQ	Косвенный доступ к pinfo: wr_flag, запрос	PIC
BASE+0x1064	R	GDI_PINFO_ACK	Косвенный доступ к pinfo: ack (данные подготовлены)	PIC
BASE+0x1068	RW	GDI_PINFO_ADDR	Косвенный доступ к pinfo: адрес доступа	PIC
BASE+0x106C	RW	GDI_PINFO_DATA	Косвенный доступ к pinfo: данные считывания	PIC
BASE+0x1070	RW	GDI_BWB_ENABLE	Обратная запись пакета: разрешено, frame_index	PIC
BASE+0x1074	RW	GDI_BWB_SIZE	Обратная запись пакета: размер изображения	PIC
BASE+0x1078	RW	GDI_BWB_STD_STRUCT	Обратная запись пакета: стандарт, прочие параметры	PIC
BASE+0x107C	R	GDI_BWB_STATUS	Статус модуля обратной записи пакета	PIC
Зарезервировано				
BASE+0x10A0	RW	GDI_WPROT_ERR_CLEAR	Сброс ошибки из-за защиты от записи	PIC
BASE+0x10A4	R	GDI_WPROT_ERR_RSN	Причина ошибки из-за защиты от записи	PIC
BASE+0x10A8	R	GDI_WPROT_ERR_ADR	Адрес ошибки из-за защиты от записи	PIC
BASE+0x10AC	RW	GDI_WPROT_RGN_EN	Область защиты от записи, выбор гл./вспом.	PIC
BASE+0x10B0	RW	GDI_WPROT_RGN0_STA	Стартовый адрес области № 0 защиты от записи	PIC
BASE+0x10B4	RW	GDI_WPROT_RGN0_END	Конечный адрес области № 0 защиты от записи	PIC
BASE+0x10B8	RW	GDI_WPROT_RGN1_STA	Стартовый адрес области № 1 защиты от записи	PIC
BASE+0x10BC	RW	GDI_WPROT_RGN1_END	Конечный адрес области № 1 защиты от записи	PIC
BASE+0x10C0	RW	GDI_WPROT_RGN2_STA	Стартовый адрес области № 2 защиты от записи	PIC
BASE+0x10C4	RW	GDI_WPROT_RGN2_END	Конечный адрес области № 2 защиты от записи	PIC
BASE+0x10C8	RW	GDI_WPROT_RGN3_STA	Стартовый адрес области № 3 защиты от записи	PIC
BASE+0x10CC	RW	GDI_WPROT_RGN3_END	Конечный адрес области № 3 защиты от записи	PIC
BASE+0x10D0	RW	GDI_WPROT_RGN4_STA	Стартовый адрес области № 4 защиты от записи	PIC
BASE+0x10D4	RW	GDI_WPROT_RGN4_END	Конечный адрес области № 4 защиты от записи	PIC
BASE+0x10D8	RW	GDI_WPROT_RGN5_STA	Стартовый адрес области № 5 защиты от записи	PIC PIC
BASE+0x10DC	RW	GDI_WPROT_RGN5_END	Конечный адрес области № 5 защиты от записи	
Зарезервировано				
BASE+0x10F0	RW	GDI_BUS_CTRL	Запросов больше нет	PIC
BASE+0x10F4	R	GDI_BUS_STATUS	Флаг пустой очереди запросов	PIC
Зарезервировано				
BASE+0x1400	RW	GDI_PIC_INFO_00_0	Карта распределения, цвет, чередование, шаг по индексу	PIC
BASE+0x1404	RW	GDI_PIC_INFO_00_1	Размер изображения по горизонтали, вертикали в режиме яркости	PIC
BASE+0x1408	RW	GDI_PIC_INFO_00_2	Базовый адрес изображения сверху в режиме яркости	PIC

Продолжение таблицы 1.503

Адрес	Тип	Имя	Описание	MB/PIC
BASE+0x140C	RW	GDI_PIC_INFO_00_3	Базовый адрес изображения сверху в режиме Cb	PIC
BASE+0x1410	RW	GDI_PIC_INFO_00_4	Базовый адрес изображения сверху в режиме Cr	PIC
BASE+0x1414	RW	GDI_PIC_INFO_00_5	Базовый адрес изображения снизу в режиме яркости	PIC
BASE+0x1418	RW	GDI_PIC_INFO_00_6	Базовый адрес изображения сверху в режиме	PIC
BASE+0x141C	RW	GDI_PIC_INFO_00_7	Cb Базовый адрес изображения сверху в режиме Cr	PIC
BASE+0x1420	RW	GDI_PIC_INFO_01_0	Карта распределения, цвет, чередование, шаг	PIC
BASE+0x1424	RW	GDI_PIC_INFO_01_1	по индексу Размер изображения по горизонтали, вертикали в режиме яркости	PIC
BASE+0x1428	RW	GDI_PIC_INFO_01_2	Базовый адрес изображения сверху в режиме яркости	PIC
BASE+0x142c	RW	GDI_PIC_INFO_01_3	Базовый адрес изображения сверху в режиме Cb	PIC
BASE+0x1430	RW	GDI_PIC_INFO_01_4	Базовый адрес изображения сверху в режиме Cr	PIC
…		(сокращено)	…	
BASE+0x17E0	RW	GDI_PIC_INFO_31_0	Карта распределения, цвет, чередование, шаг по индексу	PIC
BASE+0x17E4	RW	GDI_PIC_INFO_31_1	Размер изображения по горизонтали, вертикали в режиме яркости	PIC
BASE+0x17E8	RW	GDI_PIC_INFO_31_2	Базовый адрес изображения сверху в режиме яркости	PIC
BASE+0x17EC	RW	GDI_PIC_INFO_31_3	Базовый адрес изображения сверху в режиме Cb	PIC
BASE+0x17F0	RW	GDI_PIC_INFO_31_4	Базовый адрес изображения сверху в режиме	PIC
BASE+0x17F4	RW	GDI_PIC_INFO_31_5	Cr Базовый адрес изображения снизу в режиме яркости	PIC
BASE+0x17F8	RW	GDI_PIC_INFO_31_6	Базовый адрес изображения снизу в режиме Cb	PIC
BASE+0x17FC	RW	GDI_PIC_INFO_31_7	Базовый адрес изображения снизу в режиме Cr	PIC
BASE+0x1800	RW	GDI_XY2AXI_LUM_BIT00	Параметр для X,Y при преобразовании адреса AXI[00] в режиме яркости	PIC
BASE+0x1804	RW	GDI_XY2AXI_LUM_BIT01	Параметр для X,Y при преобразовании адреса AXI[01] в режиме яркости	PIC
BASE+0x1808	RW	GDI_XY2AXI_LUM_BIT02	Параметр для X,Y при преобразовании адреса AXI[02] в режиме яркости	PIC
BASE+0x180C	RW	GDI_XY2AXI_LUM_BIT03	Параметр для X,Y при преобразовании адреса AXI[03] в режиме яркости	PIC
BASE+0x1810	RW	GDI_XY2AXI_LUM_BIT04	Параметр для X,Y при преобразовании адреса AXI[04] в режиме яркости	PIC
BASE+0x1814	RW	GDI_XY2AXI_LUM_BIT05	Параметр для X,Y при преобразовании адреса AXI[05] в режиме яркости	PIC
BASE+0x1818	RW	GDI_XY2AXI_LUM_BIT06	Параметр для X,Y при преобразовании адреса AXI[06] в режиме яркости	PIC
BASE+0x181C	RW	GDI_XY2AXI_LUM_BIT07	Параметр для X,Y при преобразовании адреса AXI[07] в режиме яркости	PIC
BASE+0x1820	RW	GDI_XY2AXI_LUM_BIT08	Параметр для X,Y при преобразовании адреса AXI[08] в режиме яркости	PIC
BASE+0x1824	RW	GDI_XY2AXI_LUM_BIT09	Параметр для X,Y при преобразовании адреса AXI[09] в режиме яркости	PIC

Продолжение таблицы 1.503

Адрес	Тип	Имя	Описание	MB/PIC
BASE+0x1828	RW	GDI_XY2AXI_LUM_BIT0A	Параметр для X,Y при преобразовании адреса AXI[10] в режиме яркости	PIC
BASE+0x182C	RW	GDI_XY2AXI_LUM_BIT0B	Параметр для X,Y при преобразовании адреса AXI[11] в режиме яркости	PIC
BASE+0x1830	RW	GDI_XY2AXI_LUM_BIT0C	Параметр для X,Y при преобразовании адреса AXI[12] в режиме яркости	PIC
BASE+0x1834	RW	GDI_XY2AXI_LUM_BIT0D	Параметр для X,Y при преобразовании адреса AXI[13] в режиме яркости	PIC
BASE+0x1838	RW	GDI_XY2AXI_LUM_BIT0E	Параметр для X,Y при преобразовании адреса AXI[14] в режиме яркости	PIC
BASE+0x183C	RW	GDI_XY2AXI_LUM_BIT0F	Параметр для X,Y при преобразовании адреса AXI[15] в режиме яркости	PIC
BASE+0x1840	RW	GDI_XY2AXI_LUM_BIT10	Параметр для X,Y при преобразовании адреса AXI[16] в режиме яркости	PIC
BASE+0x1844	RW	GDI_XY2AXI_LUM_BIT11	Параметр для X,Y при преобразовании адреса AXI[17] в режиме яркости	PIC
BASE+0x1848	RW	GDI_XY2AXI_LUM_BIT12	Параметр для X,Y при преобразовании адреса AXI[18] в режиме яркости	PIC
BASE+0x184C	RW	GDI_XY2AXI_LUM_BIT13	Параметр для X,Y при преобразовании адреса AXI[19] в режиме яркости	PIC
BASE+0x1850	RW	GDI_XY2AXI_LUM_BIT14	Параметр для X,Y при преобразовании адреса AXI[20] в режиме яркости	PIC
BASE+0x1854	RW	GDI_XY2AXI_LUM_BIT15	Параметр для X,Y при преобразовании адреса AXI[21] в режиме яркости	PIC
BASE+0x1858	RW	GDI_XY2AXI_LUM_BIT16	Параметр для X,Y при преобразовании адреса AXI[22] в режиме яркости	PIC
BASE+0x185C	RW	GDI_XY2AXI_LUM_BIT17	Параметр для X,Y при преобразовании адреса AXI[23] в режиме яркости	PIC
BASE+0x1860	RW	GDI_XY2AXI_LUM_BIT18	Параметр для X,Y при преобразовании адреса AXI[24] в режиме яркости	PIC
BASE+0x1864	RW	GDI_XY2AXI_LUM_BIT19	Параметр для X,Y при преобразовании адреса AXI[25] в режиме яркости	PIC
BASE+0x1868	RW	GDI_XY2AXI_LUM_BIT1A	Параметр для X,Y при преобразовании адреса AXI[26] в режиме яркости	PIC
BASE+0x186C	RW	GDI_XY2AXI_LUM_BIT1B	Параметр для X,Y при преобразовании адреса AXI[27] в режиме яркости	PIC
BASE+0x1870	RW	GDI_XY2AXI_LUM_BIT1C	Параметр для X,Y при преобразовании адреса AXI[28] в режиме яркости	PIC
BASE+0x1874	RW	GDI_XY2AXI_LUM_BIT1D	Параметр для X,Y при преобразовании адреса AXI[29] в режиме яркости	PIC
BASE+0x1878	RW	GDI_XY2AXI_LUM_BIT1E	Параметр для X,Y при преобразовании адреса AXI[30] в режиме яркости	PIC
BASE+0x187C	RW	GDI_XY2AXI_LUM_BIT1F	Параметр для X,Y при преобразовании адреса AXI[31] в режиме яркости	PIC
BASE+0x1880	RW	GDI_XY2AXI_CHR_BIT00	Параметр для X,Y при преобразовании адреса AXI[00] в режиме цветности	PIC
BASE+0x1884	RW	GDI_XY2AXI_CHR_BIT01	Параметр для X,Y при преобразовании адреса AXI[01] в режиме цветности	PIC
BASE+0x1888	RW	GDI_XY2AXI_CHR_BIT02	Параметр для X,Y при преобразовании адреса AXI[02] в режиме цветности	PIC
BASE+0x188C	RW	GDI_XY2AXI_CHR_BIT03	Параметр для X,Y при преобразовании адреса AXI[03] в режиме цветности	PIC
BASE+0x1890	RW	GDI_XY2AXI_CHR_BIT04	Параметр для X,Y при преобразовании адреса AXI[04] в режиме цветности	PIC
BASE+0x1894	RW	GDI_XY2AXI_CHR_BIT05	Параметр для X,Y при преобразовании адреса AXI[05] в режиме цветности	PIC
BASE+0x1898	RW	GDI_XY2AXI_CHR_BIT06	Параметр для X,Y при преобразовании адреса AXI[06] в режиме цветности	PIC

Продолжение таблицы 1.503

Адрес	Тип	Имя	Описание	MB/PIC
BASE+0x189C	RW	GDI_XY2AXI_CHR_BIT07	Параметр для X,Y при преобразовании адреса AXI[07] в режиме цветности	PIC
BASE+0x18A0	RW	GDI_XY2AXI_CHR_BIT08	Параметр для X,Y при преобразовании адреса AXI[08] в режиме цветности	PIC
BASE+0x18A4	RW	GDI_XY2AXI_CHR_BIT09	Параметр для X,Y при преобразовании адреса AXI[09] в режиме цветности	PIC
BASE+0x18A8	RW	GDI_XY2AXI_CHR_BIT0A	Параметр для X,Y при преобразовании адреса AXI[10] в режиме цветности	PIC
BASE+0x18AC	RW	GDI_XY2AXI_CHR_BIT0B	Параметр для X,Y при преобразовании адреса AXI[11] в режиме цветности	PIC
BASE+0x18B0	RW	GDI_XY2AXI_CHR_BIT0C	Параметр для X,Y при преобразовании адреса AXI[12] в режиме цветности	PIC
BASE+0x18B4	RW	GDI_XY2AXI_CHR_BIT0D	Параметр для X,Y при преобразовании адреса AXI[13] в режиме цветности	PIC
BASE+0x18B8	RW	GDI_XY2AXI_CHR_BIT0E	Параметр для X,Y при преобразовании адреса AXI[14] в режиме цветности	PIC
BASE+0x18BC	RW	GDI_XY2AXI_CHR_BIT0F	Параметр для X,Y при преобразовании адреса AXI[15] в режиме цветности	PIC
BASE+0x18C0	RW	GDI_XY2AXI_CHR_BIT10	Параметр для X,Y при преобразовании адреса AXI[16] в режиме цветности	PIC
BASE+0x18C4	RW	GDI_XY2AXI_CHR_BIT11	Параметр для X,Y при преобразовании адреса AXI[17] в режиме цветности	PIC
BASE+0x18C8	RW	GDI_XY2AXI_CHR_BIT12	Параметр для X,Y при преобразовании адреса AXI[18] в режиме цветности	PIC
BASE+0x18CC	RW	GDI_XY2AXI_CHR_BIT13	Параметр для X,Y при преобразовании адреса AXI[19] в режиме цветности	PIC
BASE+0x18D0	RW	GDI_XY2AXI_CHR_BIT14	Параметр для X,Y при преобразовании адреса AXI[20] в режиме цветности	PIC
BASE+0x18D4	RW	GDI_XY2AXI_CHR_BIT15	Параметр для X,Y при преобразовании адреса AXI[21] в режиме цветности	PIC
BASE+0x18D8	RW	GDI_XY2AXI_CHR_BIT16	Параметр для X,Y при преобразовании адреса AXI[22] в режиме цветности	PIC
BASE+0x18DC	RW	GDI_XY2AXI_CHR_BIT17	Параметр для X,Y при преобразовании адреса AXI[23] в режиме цветности	PIC
BASE+0x18E0	RW	GDI_XY2AXI_CHR_BIT18	Параметр для X,Y при преобразовании адреса AXI[24] в режиме цветности	PIC
BASE+0x18E4	RW	GDI_XY2AXI_CHR_BIT19	Параметр для X,Y при преобразовании адреса AXI[25] в режиме цветности	PIC
BASE+0x18E8	RW	GDI_XY2AXI_CHR_BIT1A	Параметр для X,Y при преобразовании адреса AXI[26] в режиме цветности	PIC
BASE+0x18EC	RW	GDI_XY2AXI_CHR_BIT1B	Параметр для X,Y при преобразовании адреса AXI[27] в режиме цветности	PIC
BASE+0x18F0	RW	GDI_XY2AXI_CHR_BIT1C	Параметр для X,Y при преобразовании адреса AXI[28] в режиме цветности	PIC
BASE+0x18F4	RW	GDI_XY2AXI_CHR_BIT1D	Параметр для X,Y при преобразовании адреса AXI[29] в режиме цветности	PIC
BASE+0x18F8	RW	GDI_XY2AXI_CHR_BIT1E	Параметр для X,Y при преобразовании адреса AXI[30] в режиме цветности	PIC
BASE+0x18FC	RW	GDI_XY2AXI_CHR_BIT1F	Параметр для X,Y при преобразовании адреса AXI[31] в режиме цветности	PIC
BASE+0x1900	RW	GDI_XY2AXI_CFG	X,Y для общего конфигурирования преобразования AXI	PIC

1.3.6.4.2.4.2 Описание регистров GDI

Таблица 1.504 – GDI_PRI_RD_PRIOR_L (0x1000)

Бит	Имя	Тип	Функция	Значение после сброса
31:28	PRI_RD_PRI_7	RW	Приоритет эталона № 7 по считыванию	0
27:24	PRI_RD_PRI_6	RW	Приоритет эталона № 6 по считыванию	0
23:20	PRI_RD_PRI_5	RW	Приоритет эталона № 5 по считыванию	0
19:16	PRI_RD_PRI_4	RW	Приоритет эталона № 4 по считыванию	0
15:12	PRI_RD_PRI_3	RW	Приоритет эталона № 3 по считыванию	0
11: 8	PRI_RD_PRI_2	RW	Приоритет эталона № 2 по считыванию	0
7: 4	PRI_RD_PRI_1	RW	Приоритет эталона № 1 по считыванию	0
3: 0	PRI_RD_PRI_0	RW	Приоритет эталона № 0 по считыванию	0

Таблица 1.505 – GDI_PRI_RD_PRIOR_H (0x1004)

Бит	Имя	Тип	Функция	Значение после сброса
31:28	PRI_RD_PRI_15	RW	Приоритет эталона № 15 по считыванию	0
27:24	PRI_RD_PRI_14	RW	Приоритет эталона № 14 по считыванию	0
23:20	PRI_RD_PRI_13	RW	Приоритет master № 13 по считыванию	0
19:16	PRI_RD_PRI_12	RW	Приоритет master № 12 по считыванию	0
15:12	PRI_RD_PRI_11	RW	Приоритет master № 11 по считыванию	0
11: 8	PRI_RD_PRI_10	RW	Приоритет master № 10 по считыванию	0
7: 4	PRI_RD_PRI_9	RW	Приоритет master № 9 по считыванию	0
3: 0	PRI_RD_PRI_8	RW	Приоритет master № 8 по считыванию	0

Таблица 1.506 – GDI_PRI_WR_PRIOR_L (0x1008)

Бит	Имя	Тип	Функция	Значение после сброса
31:28	PRI_WR_PRI_7	RW	Приоритет эталона № 7 по записи	0
27:24	PRI_WR_PRI_6	RW	Приоритет эталона № 6 по записи	0
23:20	PRI_WR_PRI_5	RW	Приоритет эталона № 5 по записи	0
19:16	PRI_WR_PRI_4	RW	Приоритет эталона № 4 по записи	0
15:12	PRI_WR_PRI_3	RW	Приоритет эталона № 3 по записи	0
11: 8	PRI_WR_PRI_2	RW	Приоритет эталона № 2 по записи	0
7: 4	PRI_WR_PRI_1	RW	Приоритет эталона № 1 по записи	0
3: 0	PRI_WR_PRI_0	RW	Приоритет эталона № 0 по записи	0

Таблица 1.507 – GDI_PRI_WR_PRIOR_H (0x100C)

Бит	Имя	Тип	Функция	Значение после сброса
31:28	PRI_WR_PRI_15	RW	Приоритет эталона № 15 по записи	0
27:24	PRI_WR_PRI_14	RW	Приоритет эталона № 14 по записи	0
23:20	PRI_WR_PRI_13	RW	Приоритет эталона № 13 по записи	0
19:16	PRI_WR_PRI_12	RW	Приоритет эталона № 12 по записи	0
15:12	PRI_WR_PRI_11	RW	Приоритет эталона № 11 по записи	0
11: 8	PRI_WR_PRI_10	RW	Приоритет эталона № 10 по записи	0
7: 4	PRI_WR_PRI_9	RW	Приоритет эталона № 9 по записи	0
3: 0	PRI_WR_PRI_8	RW	Приоритет эталона № 8 по записи	0

Таблица 1.508 – GDI_SEC_RD_PRIOR_L (0x1018)

Бит	Имя	Тип	Функция	Значение после сброса
31:28	SEC_RD_PRI_7	RW	Приоритет эталона № 7 по считыванию	0
27:24	SEC_RD_PRI_6	RW	Приоритет эталона № 6 по считыванию	0
23:20	SEC_RD_PRI_5	RW	Приоритет эталона № 5 по считыванию	0
19:16	SEC_RD_PRI_4	RW	Приоритет эталона № 4 по считыванию	0
15:12	SEC_RD_PRI_3	RW	Приоритет эталона № 3 по считыванию	0
11: 8	SEC_RD_PRI_2	RW	Приоритет эталона № 2 по считыванию	0
7: 4	SEC_RD_PRI_1	RW	Приоритет эталона № 1 по считыванию	0
3: 0	SEC_RD_PRI_0	RW	Приоритет эталона № 0 по считыванию	0

Таблица 1.509 – GDI_SEC_RD_PRIOR_H (0x101C)

Бит	Имя	Тип	Функция	Значение после сброса
31:28	SEC_RD_PRI_15	RW	Приоритет эталона № 15 по считыванию	0
27:24	SEC_RD_PRI_14	RW	Приоритет эталона № 14 по считыванию	0
23:20	SEC_RD_PRI_13	RW	Приоритет эталона № 13 по считыванию	0
19:16	SEC_RD_PRI_12	RW	Приоритет эталона № 12 по считыванию	0
15:12	SEC_RD_PRI_11	RW	Приоритет эталона № 11 по считыванию	0
11: 8	SEC_RD_PRI_10	RW	Приоритет эталона № 10 по считыванию	0
7: 4	SEC_RD_PRI_9	RW	Приоритет эталона № 9 по считыванию	0
3: 0	SEC_RD_PRI_8	RW	Приоритет эталона № 8 по считыванию	0

Таблица 1.510 – GDI_SEC_WR_PRIOR_L (0x1020)

Бит	Имя	Тип	Функция	Значение после сброса
31:28	SEC_WR_PRI_7	RW	Приоритет эталона № 7 по записи	0
27:24	SEC_WR_PRI_6	RW	Приоритет эталона № 6 по записи	0
23:20	SEC_WR_PRI_5	RW	Приоритет эталона № 5 по записи	0
19:16	SEC_WR_PRI_4	RW	Приоритет эталона № 4 по записи	0
15:12	SEC_WR_PRI_3	RW	Приоритет эталона № 3 по записи	0
11: 8	SEC_WR_PRI_2	RW	Приоритет эталона № 2 по записи	0
7: 4	SEC_WR_PRI_1	RW	Приоритет эталона № 1 по записи	0
3: 0	SEC_WR_PRI_0	RW	Приоритет эталона № 0 по записи	0

Таблица 1.511 – GDI_SEC_WR_PRIOR_H (0x1024)

Бит	Имя	Тип	Функция	Значение после сброса
31:28	SEC_WR_PRI_15	RW	Приоритет эталона № 15 по записи	0
27:24	SEC_WR_PRI_14	RW	Приоритет эталона № 14 по записи	0
23:20	SEC_WR_PRI_13	RW	Приоритет эталона № 13 по записи	0
19:16	SEC_WR_PRI_12	RW	Приоритет эталона № 12 по записи	0
15:12	SEC_WR_PRI_11	RW	Приоритет эталона № 11 по записи	0
11: 8	SEC_WR_PRI_10	RW	Приоритет эталона № 10 по записи	0
7: 4	SEC_WR_PRI_9	RW	Приоритет эталона № 9 по записи	0
3: 0	SEC_WR_PRI_8	RW	Приоритет эталона № 8 по записи	0

Таблица 1.512 – GDI_SEC_CLIENT_EN (0x1030)

Бит	Имя	Тип	Функция	Значение после сброса
0	SecEn0	RW	Запись восстановленных данных для intra-прогнозирования	0
1	SecEn1	RW	Запись/считывание MIB BIT	0
2	SecEn2	RW	зарезервировано	-
3	SecEn3	RW	Пространственное прогнозирование (считывание/запись ACDC, считывание IP)	0
4	SecEn4	RW	Временной буфер фильтра перекрывания	0
5	SecEn5	RW	Временной буфер фильтра деблокирования в режиме яркости	0
6	SecEn6	RW	Временной буфер фильтра деблокирования в режиме цветности	0
7	SecEn7	RW	зарезервировано	-
8	SecEn8	RW	Разбиение на подсэмплы JPEG	0
9	SecEn9	RW	зарезервировано	-
10	SecEn10	RW	Временной буфер SPP BPU	0

Таблица 1.513 – GDI_PIC_INIT_HOST (0x1038)

Бит	Имя	Тип	Функция	Значение после сброса
0	GDI_PIC_INIT	W	Генерирование сигнала pic_init на стороне хоста	-

Таблица 1.514 – GDI_CONFIG_REPORT_0 (0x1040)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	GDI_CONFIG_ RE-PORT_0	R	Название и версия продукта	-

Таблица 1.515 – GDI_CONFIG_REPORT_1 (0x1044)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	GDI_CONFIG_ RE-PORT_1	R	Конфигурация интерфейса, определение аппаратного обеспечения	-

Таблица 1.516 – GDI_CONFIG_REPORT_2 (0x1048)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	GDI_CONFIG_ RE-PORT_2	R	Определение стандарта	-

Таблица 1.517 – GDI_CONFIG_REPORT_3 (0x104C)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	GDI_CONFIG_ RE-PORT_3	R	Детальное определение стандарта	-

Таблица 1.518 – GDI_CONFIG_REPORT_4 (0x1050)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	GDI_CONFIG_ RE-PORT_4	R	Определение в cnm_define	-

Таблица 1.519 – GDI_CONFIG_REPORT_5 (0x1054)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	GDI_CONFIG_ RE-PORT_5	R	Дата конфигурирования отчета	-

Таблица 1.520 – GDI_CONFIG_REPORT_6 (0x1058)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	GDI_CONFIG_ RE-PORT_6	R	Версия конфигурирования отчета	-

Таблица 1.521 – GDI_CONFIG_REPORT_7 (0x105C)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	GDI_CONFIG_ RE-PORT_7	R	Тип конфигурирования отчета	-

Таблица 1.522 – GDI_PINFO_REQ (0x1060)

Бит	Имя	Тип	Функция	Значение после сброса
0	PINFO_REQ	W	Запрос косвенного доступа к считыванию PINFO считывается не в области тактовой частоты ядра. При доступе к данным PINFO используется схема Request-Ack	-

Таблица 1.523 – GDI_PINFO_ACK (0x1064)

Бит	Имя	Тип	Функция	Значение после сброса
0	PINFO_ACK	R	Запрос косвенного доступа к считыванию PINFO считывается не в области тактовой частоты ядра. При доступе к данным PINFO используется схема Request-Ack	0

Таблица 1.524 – GDI_PINFO_ADDR (0x1068)

Бит	Имя	Тип	Функция	Значение после сброса
9:0	PINFO_ADDR	RW	Адрес считывания PINFO	0

Таблица 1.525 – GDI_PINFO_DATA (0x106C)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	PINFO_DATA	RW	Адрес считывания PINFO	0

Таблица 1.526 – GDI_BWB_ENABLE (0x1070)

Бит	Имя	Тип	Функция	Значение после сброса
9	PIC_INIT	RW	Сигнал pic_init для модуля BWB. Модуль BWB должен инициализироваться при записи в данный регистр 1 и 0.	0
8	BWB_EN	RW	Флаг включения модуля BWB. 1 - включен	0
7:0	FRM_IDX	RW	Индекс кадра, подлежащего буферизации с использованием модуля BWB.	0

Таблица 1.527 – GDI_BWB_SIZE (0x1074)

Бит	Имя	Тип	Функция	Значение после сброса
7:0	BWB_XSIZE	W	Размер изображения по горизонтали (единица измерения: байт)	0
15:0	BWB_YSIZE	W	Размер изображения по вертикали (единица измерения: байт)	0

Таблица 1.528 – GDI_BWB_STD_STRUCT (0x1078)

Бит	Имя	Тип	Функция	Значение после сброса
10	LOOPFILT_EN	RW	Флаг включения контурного фильтра (1: включен)	0
9	CBCR_INTLV	RW	Флаг чередования в режиме цветности (1: чередование)	0
8:5	CODEC_STD	RW	Значение codec_std. В модуле MBC аналогичное значение.	0
4:3	STD_AUX	RW	Значение std_aux value. В модуле MBC аналогичное значение.	0
2:0	PIC_STRUCT	RW	Значение pic_struct value. В модуле MBC аналогичное значение.	0

Таблица 1.529 – GDI_BWB_STATUS (0x107C)

Бит	Имя	Тип	Функция	Значение после сброса
4	Y_FLUSH	R	Сброс флага для компоненты Y	0
3	U_FLUSH	R	Сброс флага для компоненты Cb	0
2	V_FLUSH	R	Сброс флага для компоненты Cr	0
1	IN_RESP	R	Флаг занятости очереди запросов на входе	0
0	OUT_RESP	R	Флаг занятости очереди запросов на выходе	0

Таблица 1.530 – GDI_WPROT_ERR_CLEAR (0x10A0)

Бит	Имя	Тип	Функция	Значение после сброса
0	CLEAR	RW	Запись 1 в данный регистр очищает регистр причины возникновения ошибки WPROT. Данный регистр – это регистр с возвратом в ноль. Очистка выполняется тогда, когда после записи 1 данный регистр возвращается в ноль.	0

Таблица 1.531 – GDI_WPROT_ERR_RSN (0x10A4)

Бит	Имя	Тип	Функция	Значение после сброса
9	ERR_FLAG	R	Флаг ошибки из-за защиты от записи	0
8	ERR_PRI_SEC	R	Главный или вспомогательный флаг	0
7:4	ERR_AWID	R	Ошибка из-за защиты от записи AWID	0
3:0	ERR_AWLEN	R	Ошибка из-за защиты от записи AWLEN	0

Таблица 1.532 – GDI_WPROT_ERR_ADR (0x10A8)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	ERR_AWADDR	R	Ошибка из-за защиты от записи AWADDR	0

Таблица 1.533 – GDI_WPROT_RGN_EN (0x10AC)

Бит	Имя	Тип	Функция	Значение после сброса
11	RGN5_SEC	RW	Вспомогательный флаг области № 5, 1-вспомогательный	0
10	RGN4_SEC	RW	Вспомогательный флаг области № 4, 1-вспомогательный	0
9	RGN3_SEC	RW	Вспомогательный флаг области № 3, 1-вспомогательный	0
8	RGN2_SEC	RW	Вспомогательный флаг области № 2, 1-вспомогательный	0
7	RGN1_SEC	RW	Вспомогательный флаг области № 1, 1-вспомогательный	0
6	RGN0_SEC	RW	Вспомогательный флаг области № 0, 1-вспомогательный	0
5	RGN5_EN	RW	Флаг включения области № 5, 1-включена	0
4	RGN4_EN	RW	Флаг включения области № 4, 1-включена	0
3	RGN3_EN	RW	Флаг включения области № 3, 1-включена	0
2	RGN2_EN	RW	Флаг включения области № 2, 1-включена	0
1	RGN1_EN	RW	Флаг включения области № 1, 1-включена	0
0	RGN0_EN	RW	Флаг включения области № 0, 1-включена	0

Таблица 1.534 – GDI_WPROT_RGN0_STA (0x10B0)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_STA	RW	Стартовый адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.535 – GDI_WPROT_RGN0_END (0x10B4)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_END	RW	Конечный адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.536 – GDI_WPROT_RGN1_STA (0x10B8)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_STA	RW	Стартовый адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.537 – GDI_WPROT_RGN1_END (0x10BC)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_END	RW	Конечный адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.538 – GDI_WPROT_RGN2_STA(0x10C0)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_STA	RW	Стартовый адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.539 – GDI_WPROT_RGN2_END (0x10C4)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_END	RW	Конечный адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.540 – GDI_WPROT_RGN3_STA (0x10C8)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_STA	RW	Стартовый адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.541 – GDI_WPROT_RGN3_END (0x10CC)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_END	RW	Конечный адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.542 – GDI_WPROT_RGN4_STA (0x10D0)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_STA	RW	Стартовый адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.543 – GDI_WPROT_RGN4_END (0x10D4)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_END	RW	Конечный адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.544 – GDI_WPROT_RGN5_STA (0x10D8)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_STA	RW	Стартовый адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.545 – GDI_WPROT_RGN5_END (0x10DC)

Бит	Имя	Тип	Функция	Значение после сброса
19:0	RGN_END	RW	Конечный адрес области. Используются 20 верхних бит. 12 нижних бит всегда нулевые 0	0

Таблица 1.546 – GDI_BUS_CTRL (0x10F0)

Бит	Имя	Тип	Функция	Значение после сброса
4	NO_MORE_REQ _SEC	RW	1 – отсутствуют запросы для передачи во вспомогательный AXI	0
0	NO_MORE_REQ _PRI	RW	1 - отсутствуют запросы для передачи в главный AXI	0

Таблица 1.547 – GDI_BUS_STATUS (0x10F4)

Бит	Имя	Тип	Функция	Значение после сброса
6	WRESP_EMPTY _SEC	R	Пустая очередь wresp для вспомогательного AXI (1: обслуживаются все запросы)	1
5	WD_EMPTY_SEC	R	Пустая очередь на запись данных для вспомогательного AXI (1: обслуживаются все запросы)	1
4	RD_EMPTY_SEC	R	Пустая очередь на считывание данных для вспомогательного AXI (1: обслуживаются все запросы)	1
2	WRESP_EMPTY	R	Пустая очередь wresp для главного AXI (1: обслуживаются все запросы)	1
1	WD_EMPTY	R	Пустая очередь на запись данных для главного AXI (1: обслуживаются все запросы)	1
0	RD_EMPTY	R	Пустая очередь на считывание данных для главного AXI (1: обслуживаются все запросы)	1

Таблица 1.548 – GDI_PIC_INFO_nn_0 (0x1400 + nn*0x20 + 0x0)

Бит	Имя	Тип	Функция	Значение после сброса
31:24	Зарезервировано	-	-	-
23:20	FrameMap	RW	Формат адресации буфера кадров 0: LINEAR_FRAME_MAP 1: TILED_FRAME_V_MAP 2: TILED_FRAME_H_MAP 3: TILED_FIELD_V_MAP 4: TILED_MIXED_V_MAP 5: TILED_FRAME_MB_RASTER_MAP 6: TILED_FIELD_MB_RASTER_MAP 7: TILED_FRAME_NO_BANK_MAP 8: TILED_FIELD_NO_BANK_MAP 9: LINEAR_FIELD_MAP	0
19:17	ColorMode	RW	Цветовой формат кадра 0: 420 1: 422 2: 224 3: 444 4: 400	
16	CbCrInterleaved	RW	Чередование CbCr	
15:0	Stride	RW	Шаг по индексу компоненты яркости в линейной карте распределения	0

Таблица 1.549 – GDI_PIC_INFO_nn_1 (0x1400 + nn*0x20 + 0x4)

Бит	Имя	Тип	Функция	Значение после сброса
31:16	PicHorSizeLuma	RW	Размер изображения по вертикали (единица измерения: пиксель)	0
15:0	PicVerSizeLuma	RW	Размер изображения по горизонтали (единица измерения: пиксель)	0

Таблица 1.550 – GDI_PIC_INFO_nn_2 (0x1400 + nn*0x20 + 0x8)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	LumaTopBaseAddr	RW	Базовый адрес изображения сверху в режиме яркости	0

Таблица 1.551 – GDI_PIC_INFO_nn_3 (0x1400 + nn*0x20 + 0xC)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	CbTopBaseAddr	RW	Базовый адрес изображения сверху в режиме Cb	0

Таблица 1.552 – GDI_PIC_INFO_nn_4 (0x1400 + nn*0x20 + 0x10)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	CrTopBaseAddr	RW	Базовый адрес изображения сверху в режиме Cr	0

Таблица 1.553 – GDI_PIC_INFO_nn_5 (0x1400 + nn*0x20 + 0x14)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	LumaBotBaseAddr	RW	Базовый адрес изображения снизу в режиме яркости	0

Таблица 1.554 – GDI_PIC_INFO_nn_6 (0x1400 + nn*0x20 + 0x18)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	CbBotBaseAddr	RW	Базовый адрес изображения снизу в режиме Cb	0

Таблица 1.555 – GDI_PIC_INFO_nn_7 (0x1400 + nn*0x20 + 0x1C)

Бит	Имя	Тип	Функция	Значение после сброса
31:0	CrBotBaseAddr	RW	Базовый адрес изображения снизу в режиме Cr	0

Таблица 1.556 – GDI_XY2AXI_LUM_BIT00 ~ GDI_XY2AXI_LUM_BIT1F (0x1800~0x187C)

Бит	Имя	Тип	Функция	Значение после сброса
7	LUMA inv	RW	инвертирование выходных данных	0
6	LUMA zero	RW	присвоить ноль	0
5	LUMA tbxor	RW	xor с tb	0
4	LUMA xy_sel	RW	0: X 1: Y	0
3:0	LUMA bit_sel	RW	Выбор бита	0

Таблица 1.557 – GDI_XY2AXI_CHR_BIT00 ~ GDI_XY2AXI_CHR_BIT1F (0x1880~0x18FC)

Бит	Имя	Тип	Функция	Значение после сброса
7	CHROMA inv	RW	инвертирование выходных данных	0
6	CHROMA zero	RW	присвоить ноль	0
5	CHROMA tbxor	RW	xor с tb	0
4	CHROMA xy_sel	RW	0: X 1: Y	0
3:0	CHROMA bit_sel	RW	Выбор бита	0

Таблица 1.558 – GDI_XY2AXI_CFG (0x1900)

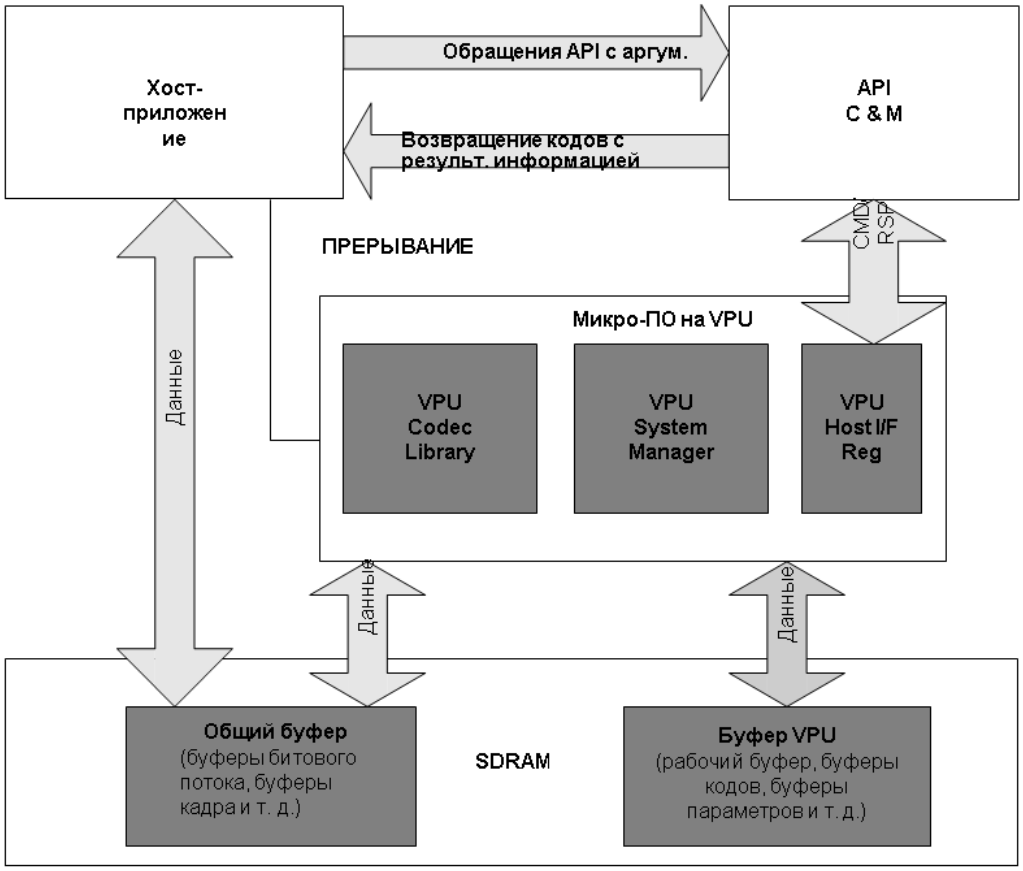
Бит	Имя	Тип	Функция	Значение после сброса
23:20	max_burst_1d	RW	Максимальная длина пакета для доступа 1-D	0
19	tb_seperate_map	RW	1: отдельная карта распределения	0
18:17	Зарезервировано	-	-	
16	burst_inc_dir	RW	0: по горизонтали 1: по вертикали	0
15:12	granule_frame_lum	RW	Зерно длины пакета для кадра в режиме яркости	0
11:8	granule_field_lum	RW	Зерно длины пакета для полукадра в режиме яркости	0
7:4	granule_frame_chr	RW	Зерно длины пакета для кадра в режиме цветности	0
3:0	granule_field_chr	RW	Зерно длины пакета для полукадра в режиме цветности	0

Примечание Регистры для загрузчика информации индекса кадра GDI – Внутренний буфер предназначен для хранения информации индексов максимально 32 кадров. Для каждого индекса кадра требуется 20 байт свободного пространства памяти. Адрес сдвига индекса первого кадра равен 0x1400, он занимает 0x14 (20 байтов). Поэтому адрес сдвига индекса второго кадра равен 0x1414.

1.3.6.4.3 ПРИКЛАДНОЙ ИНТЕРФЕЙС

1.3.6.4.3.1 Принцип управления на базе API блока VPU

Хост-приложение управляет блоком VPU, как правило, с помощью предварительно установленного интерфейса API посредством отправки команд и аргументов, а по окончании прерывания, указывающего на то, что требуемая операция выполнена, передается сигнал о том, что хост-приложение может получить результаты функций API (см. рисунок 1.153)..



VPU Codec Library – Библиотека кодека VPU
VPU System Manager – Менеджер системы VPU
VPU Host I/F Reg – Рег. I/F хоста VPU

Рисунок 1.153 – Программная модель управления VPU через хост-приложение

При определении каждого API задается необходимая команда, а также структура входных данных и структура выходных данных. Заданная команда из функции API всегда заносится в специальный регистр ввода-вывода, а структура входных данных и структура выходных данных передаются с помощью набора командных регистров ввода-вывода, в который входят входные аргументы и выходные результаты. Поэтому программистам приложений не придется терять время на решение сложных задач по определению и использованию регистров хоста.

1.3.6.4.3.2 Справочное ПО интерфейса API блока VPU

Всем заказчикам предоставляется справочное программное обеспечение API, которое позволяет реализовать приложение кодека с помощью функций API VPU. После переноса его на свою ОС или записи в память CPU, программисты приложений, опираясь на данное ПО, могут разрабатывать новые приложения для видео.

Существует пять иерархических уровней ПО API VPU. На рисунке 1.154 представлена полная схема архитектуры данного ПО.

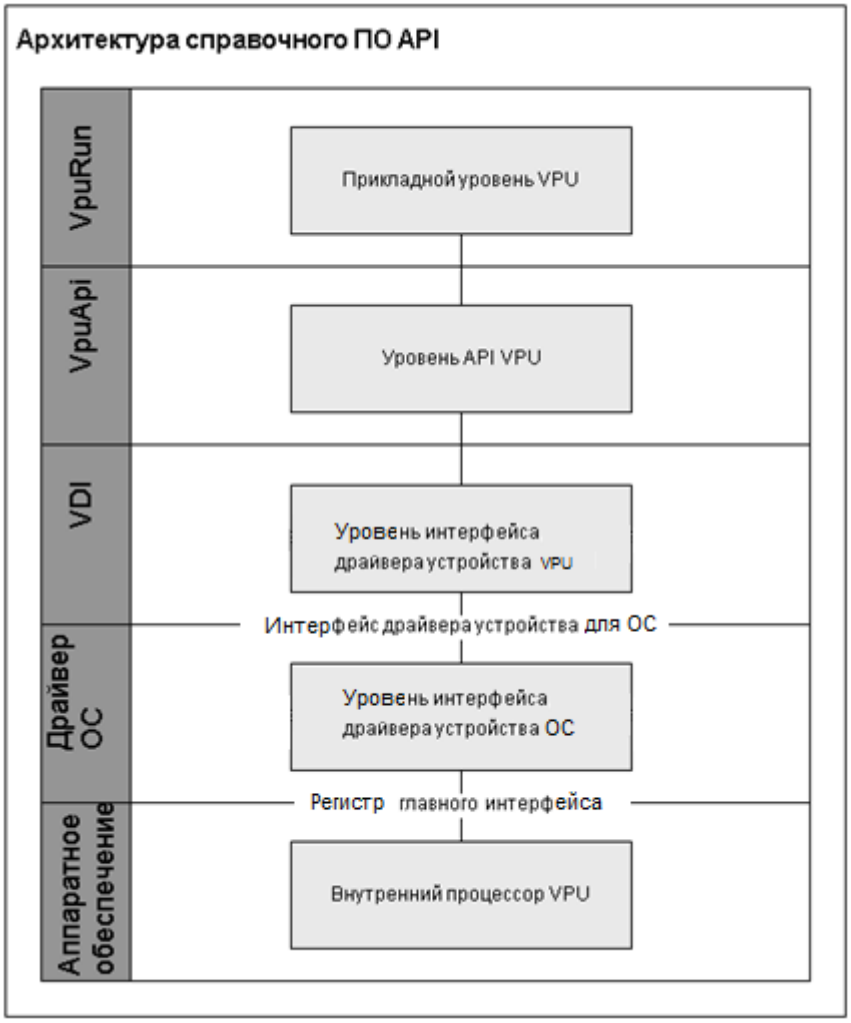


Рисунок 1.154 – Архитектура справочного ПО API

Прикладной уровень VPU – это вид содержимого простого приложения для декодирования битового потока (разбитого на пакеты в общих контейнерах) и кодирования данных изображения посредством обращения к интерфейсам API блока VPU. Уровень API VPU – это программный стек прикладного интерфейса для обеспечения связи с аппаратной архитектурой VPU. Он может функционировать на разных платформах ОС через интерфейс драйвера устройства VPU (VDI), который может получать функциональные вызовы ОС. В блоке VDI существует несколько кодов, зависимых от ОС, для каждой поддерживаемой ОС, и если в ОС есть режим ядра, то уровень VDI может связываться с уровнем драйвера устройства. А данный момент справочное ПО API обеспечивает реализацию трех различных типов блока VDI и его драйвера устройств - для Linux, Windows и NonOS. Также планируется расширить поддержку и других операционных систем.

1.3.6.4.3.2.1 Дерево исходного кода

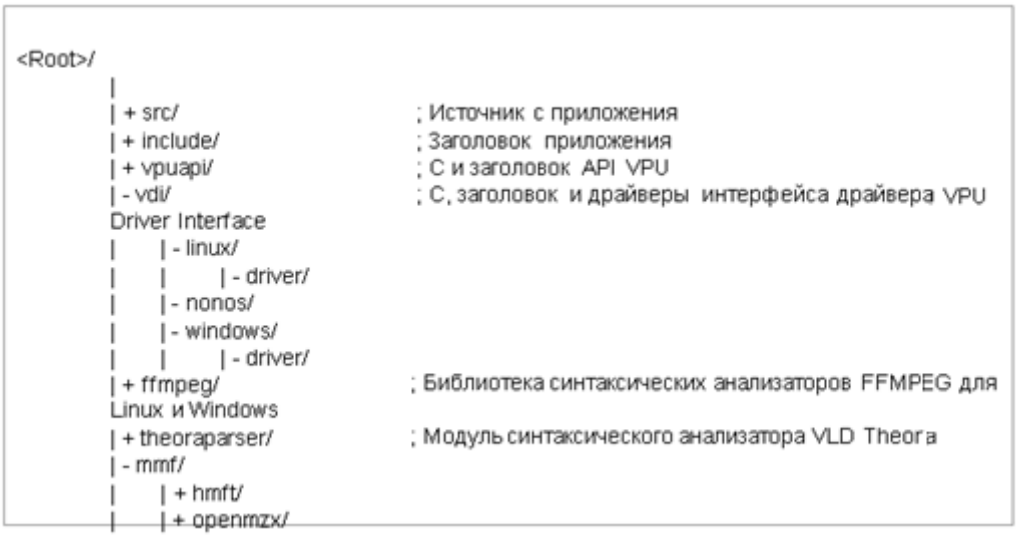


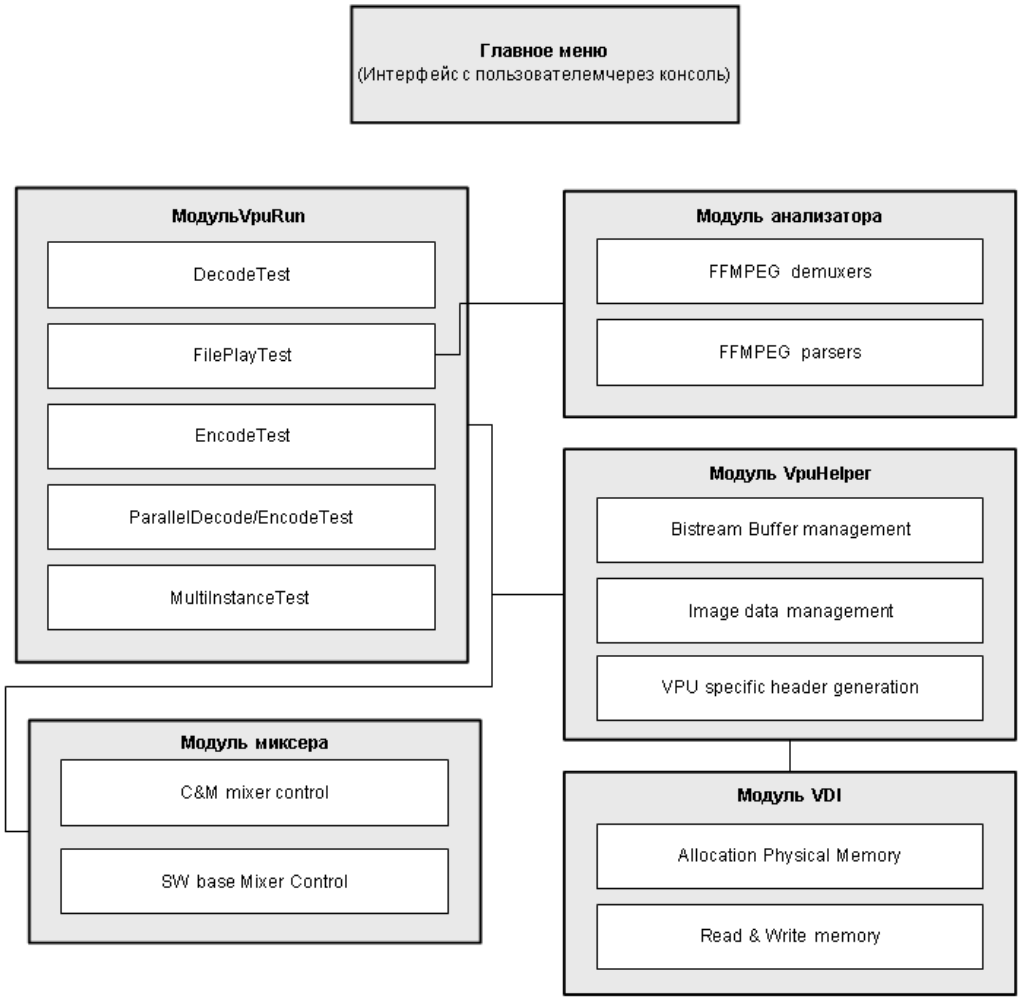
Рисунок 1.155 – Справочное ПО. Дерево исходного кода.

1.3.6.4.3.2.2 Архитектура

1.3.6.4.3.2.2.1 Прикладной уровень

На прикладном уровне блока VPU, как и на самом верхнем уровне справочного ПО, поддерживаются три функциональности. Первая состоит в поддержке последовательностей контроля видеоданных при декодировании и отображении с заданными аргументами. Вторая функциональность заключается в обеспечении вспомогательного модуля, считывающего и записывающего результирующие данные при декодировании и/или кодировании, а также интерфейсов с аппаратными ресурсами через модуль VDI. И, наконец, предоставляются функции пользовательского интерфейса для связи с пользователем приложения через меню на консоли.

Для обеспечения данных функциональностей в прикладной уровень VPU входит модуль VpuMain, модуль VpuRun, модуль семейства анализаторов и VPUHelper, а также модуль миксера. Модули прикладного уровня проиллюстрированы на рисунке 1.156 .



FFMPEG demuxers – Демультиплексоры FFMPEG
FFMPEG parsers – Анализаторы FFMPEG
Bitstream Buffer Management – Управление буфером битового потока
Image Data Management – Управление данными изображения
VPU specific header generation – Генерирование заголовков, характерных для VPU
C&M mixer control – Контроль миксера C&M
SW base Mixer Control – Контроль миксера на базе ПО
Allocation Physical Memory – Распределение физической памяти
Read & Write memory – Память для считывания и записи

Рисунок 1.156 - Прикладной уровень

Соответствующие коды находятся в SRC и включают в себя каталоги. В них находятся простые функции - инициализировать, открыть, SeqInit, декодировать/кодировать, закрыть, деинициализировать, предназначенные для управления блоком VPU через интерфейсы API VPU, а также коды, относящиеся к системе, которые обеспечивают распределение буферов кадра PP декодера, буферов исходного кадра кодера и буфер битового потока как в зависимой памяти пользователя. Но нем нет ни зависимого кода платформы, ни уровня портирования.

Ниже представлено краткое описание зон ответственности каждого из модулей и соответствующих файлов.

Главный модуль: main.c

- Точка входа
- Базовое меню консоли
- Вызов функций DecodeTest, FilePlayTest, EncoderTest, MultiInstanceTest, ParallelDecodeTest, ParallelEncodeTest в vpurun.c

Модуль VpuRun: vpurun.c

- Вызов интерфейсов API и VDI VPU для работы с VPU
- Вызов функции DecodeTest с элементарным потоком
- Функция FilePlayTest с форматом данных контейнера/надстройки и выделением элементарного потока из данного формата
- Функция EncodeTest с файлом конфигурирования опций кодера и опцией кодирования входных данных пользователя
- Функция MultiInstanceTest: Тестирование по нескольким экземплярам с несколькими задачами, имеющими другие стандарты кодека.

Модуль анализатора: библиотека с открытым исходным кодом ffmpeg

- Обеспечивает возможность справочному ПО API выделять элементарный поток из разных мультимедийных контейнеров.

Модуль VpuHelper: vpuhelper.c, vpuio.c

- Функции вспомогательного приложения для интерфейса видеокодека
- Сохранение или загрузка данных в буфере кадра физической памяти и буфере битового потока

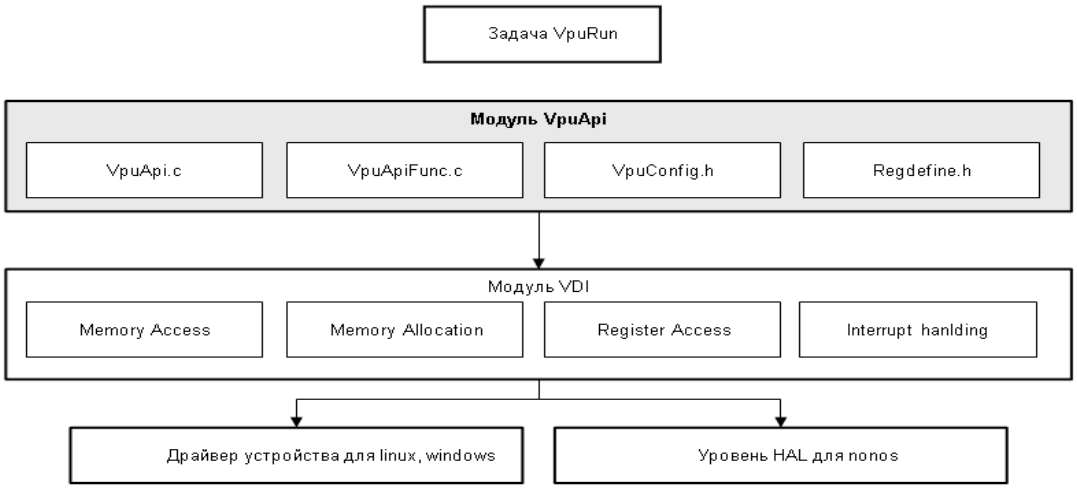
Модуль миксера: mixer.c

- Работа с модулем отображения компании “Chips&Media”
- Поддержка отображения на базе ПО, включая конвертер yuv2rgb для отображения буфера кадра ОС

1.3.6.4.3.2.2.2 Уровень API

Уровень API VPU отвечает за доступ к аппаратным регистрам VPU и непосредственное управление. Данный уровень состоит из двух модулей – главного модуля API и модуля API тайловой карты распределения, которые реализуются в vpuapi.c и vpugdi.c. Задача модуля VpuRun с верхнего прикладного уровня вызывает функции API VPU на данном уровне API для управления аппаратным блоком VPU. На нем нет ни кода платформы, ни уровня портирования.

На рисунке 1.157 представлены модули уровня API VPU, иерархия и схемы взаимодействия.



Memory Access – Доступ к памяти
Memory Allocation – Распределение памяти
Register Access – Доступ к регистрам
Interrupt handling – Обработка прерываний

Рисунок 1.157 – Уровень API VPU

Ниже представлено краткое описание зон ответственности каждого из модулей и соответствующих файлов.

Главный модуль API: VpuApi.c

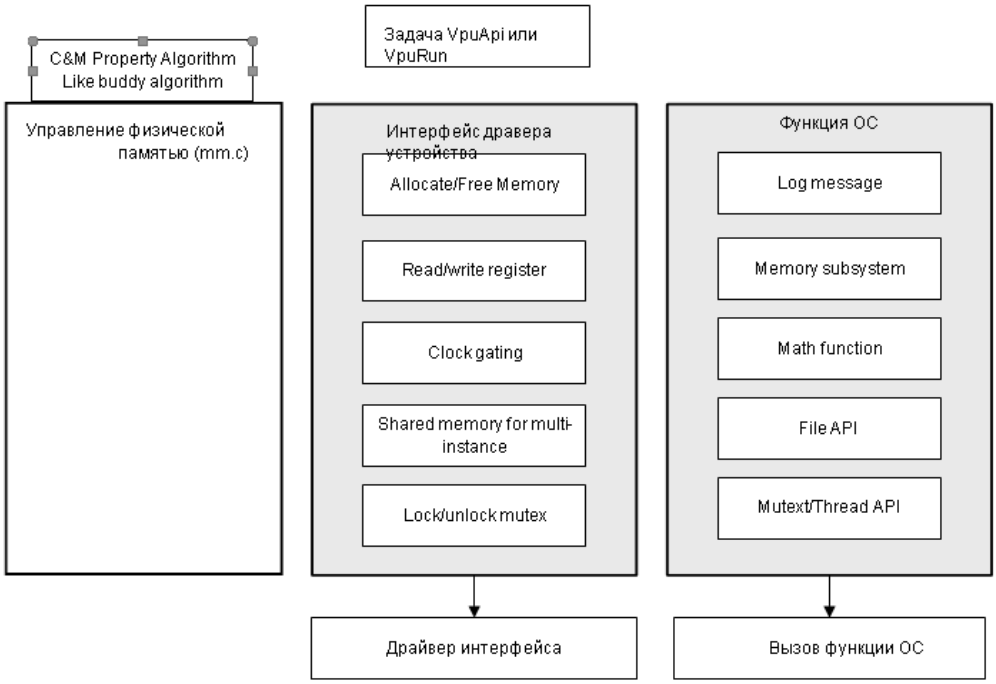
- Тела функций API декодера для управления аппаратным блоком VPU и работы с экземплярами.
- Все варианты реализации стандартов кодека используются главным интерфейсом в аппаратном блоке VPU, поэтому рассматриваемые функции API могут иметь простую архитектуру.
- Чтобы обеспечить поддержку нескольких экземпляров, под VpuApiFunc.c располагается открытый пул обработки экземпляров.
- Некоторые из функций API имеют прямой доступ к регистрам аппаратного блока VPU через функции VDI.
- Некоторые из функций API имеют доступ к функциям VDI для выделения памяти кодека и ожидания прерывания в VPU.
- Некоторые из функций API постепенно переходят к своим подфункциям под VpuApiFunc.c для доступа к регистрам аппаратного блока VPU.

1.3.6.4.3.2.2.3 Уровень VDI

Интерфейс драйвера устройства VPU (VDI) может обеспечивать сопряжение с уровнем ОС или непосредственно с VPU. Если в целевой системе есть режим ядра (например, Linux, Windows CE, Windows и т.д.), то блок VDI вызывает свой драйвер устройства. Для прочих ОС, таких, например, как RTOS или платформа Non OS, VDI может взаимодействовать с уровнем ОС или непосредственно с VPU.

Соответствующие коды находятся в директории vdi и имеют некоторый уровень портирования (коды платформ) для обеспечения интегрирования с новой целевой системой. Зависимость от VPU отсутствует.

На рисунке 1.158 представлены компоненты или функциональные модули, входящие в уровень VDI.



Allocate/Free Memory – Распределение/освобождение памяти
Read/Write Memory – Считывание/запись в память
Clock gating – Запрет подачи тактовых сигналов
Shared Memory for Multi-instance – Общая память для нескольких экземпляров
Lock/Unlock mutex – Блокирование/разблокирование мьютекса
Log message – Журнальное событие
Memory subsystem – Подсистема памяти
Math function – Математическая функция
File API – API файлов
Mutex/Thread API – API мьютекса/связки

Рисунок 1.158 – Уровень VDI

Уровень VDI отвечает за следующие функции:

- считывание/запись в регистр VPU;
- считывание/запись в физическую память;
- конвертация физического адреса в виртуальный;
- распределение/освобождение памяти;
- управление распределением памяти для VPU, если динамическое распределение использовать нельзя, но можно использовать системный вызов;
- управление экземплярами с общей памятью в режиме ядра;
- функции, характерные для систем на кристаллах (аппаратный сброс, запрет передачи тактовых сигналов, прерывание).

1.3.6.4.3.2.2.4 Уровень драйвера устройства

Уровень драйвера устройства имеет доступ и работает с аппаратным блоком VPU, осуществляет запуск в режиме ядра в случае, если в ОС установлено микро-ПО для драйвера устройства. Соответствующие коды находятся в директории VDI/[OS]/драйвера. На нем происходит распределение физического буфера посредством вызовов функций ОС, обработки прерываний, управления общей памятью, которая используется для работы с vpuapi в многозадачных приложениях.

1.3.6.4.4 УПРАВЛЕНИЕ БЛОКОМ VPU

1.3.6.4.4.1 Инициализация VPU

При первом включении VPU с хост-процессора и для обеспечения возможности последующего его использования необходимо выполнить операции, так называемой инициализации. Посредством вызова одной функции - API VPU_Init()выполняются следующие операции:

- отключение BIT-процессора при установке BIT_CODE_RUN (BASE + 0x000) = 0;
- запись микрокода (микро-ПО) BIT-процессора в память SDRAM, доступ к которой осуществляется с блока VPU непосредственно в ходе его работы;
- загрузка первых 4 кбайт микрокода (микро-ПО) в программную память BIT-процессора;
- установка указателей буфера BIT-процессора для рабочего буфера, буфера параметров и буфера кодов;
- включение опции управления (проверка наполненности/опустошения) и режима следования байтов в буфере битового потока;
- установка порядка следования байтов в буфере кадров и опции цветности (с чередованием коэффициентов CbCr);
- установка регистра разрешения прерывания;
- включение BIT-процессора при установке регистра BIT_CODE_RUN = 1.

Более подробная информация о каждом этапе инициализации и некоторые советы программистам будут представлены в последующих подразделах.

Примечание Общая длина кода микрокода BIT-процессора и объем внутренней программной памяти может меняться в соответствии с конфигурацией VPU и требованиями пользователя.

1.3.6.4.4.1.1 Проверка версии аппаратного и микропрограммного обеспечения VPU

С помощью команды GET_FW_VERSION из приложения можно проверить версию аппаратного и микропрограммного обеспечения VPU. Номер версии микропрограммного обеспечения представлен в виде 32-битного значения.

Version_number[15:12] – Главная версия

Version_number[11:8] – Модификация оборудования

Version_number[7:0] – Модификация программного обеспечения

Поэтому значение номера версии может находиться в диапазоне от 0.0.0 до 15.15.255. Для проверки версии используется специальная команда, которая поддерживается с помощью функции VPU_GetVersionInfo(), выполняемой после процесса инициализации.

1.3.6.4.4.1.2 Включение и отключение BIT-процессора

В BIT-процессоре существует специальный регистр, который отвечает за активацию или деактивацию BIT-процессора в ходе его работы - BIT_CODE_RUN (BASE + 0x000). В ходе инициализации обновляется программная память BIT-процессора, а также устанавливаются некоторые регистры конфигурации для управления операциями VPU. Поэтому во время данного процесса в целях безопасности BIT-процессор следует отключить.

После завершения инициализации хост-процессор должен включить BIT-процессор. Затем BIT-процессор запускает собственный внутренний процесс инициализации, и после этого он снова готов к работе.

1.3.6.4.4.2 Управление буфером данных BIT-процессора

BIT-процессору для выполнения операций декодирования или кодирования требуется определенный объем свободного пространства памяти SDRAM. Этот объем памяти BIT-процессора предназначен для размещения трех буферов – рабочего буфера, буфера кодов и буфера параметров.

Рабочий буфер - это разновидность временного объема памяти, в котором хранятся данные о слайсах и контекстные данные. При декодировании H.264/AVC BIT-процессору требуется дополнительный рабочий буфер для хранения SPS/PPS по каждому экземпляру.

Буфер кода – это область памяти, в которой хранится микрокод BIT-процессора (микро-ПО), а хост-процессор должен установить базовый адрес буфера кода.

Буфер параметра предназначен для размещения буфера кадров и совмещенного буфера.

Временной буфер - это буфер, в котором содержится информация, требуемая время от времени при выполнении текущего процесса декодирования - информация о деблокировании и коэффициентах AC/DC. Он используется, только когда не используется вспомогательная шина AXI памяти SRAM.

Объем каждого буфера конфигурируется следующим образом:

Таблица 1.504 – Объем буфера

Тип буфера	Объем
Рабочий буфер (WORK_BUF_SIZE)	80 кбайт (на экземпляр)
Буфер кода (CODE_BUF_SIZE)	266 кбайт
Буфер параметра (PARA_BUF_SIZE)	> 10 кбайт
Временной буфер (TEMP_BUF_SIZE)	470 кбайт

Всякий раз, когда хост-приложение открывает экземпляр, объем рабочего буфера увеличивается в зависимости от количества открытых экземпляров. На 1 экземпляр требуется 80 кбайт памяти. Поэтому, если, например, открыто 4 экземпляра, то общий объем рабочего буфера, который необходим BIT-процессору, равен 320 кбайтам.

Распределение буферов

С помощью функцию VPU_Init()хост-приложение может распределить все буферы кроме рабочего. Вызвав данную функцию можно автоматически настроить и управлять отдельными буферами через модуль VDI.

vdi_get_common_memory((unsigned long)coreIdx, &vb); codeBuffer = vb.phys_addr;

tempBuffer = codeBuffer + CODE_BUF_SIZE; paraBuffer = tempBuffer + TEMP_BUF_SIZE;

Рабочий буфер распределяется всякий раз, когда открывается экземпляр, и его объем зависит от объема, указанного в VPU_DecOpen()/VPU_EncOpen(). С помощью функций VPU API в выделенных регистрах главного интерфейса задается адрес каждого буфера.

Дополнительно к указанным буферам блоку VPU может потребоваться буфер большего объема для хранения RBSP SLICE в целях обеспечения FMO и ASO при декодировании потоков Базового профиля H.264/AVC. Такой буфер называется буфером хранения слайсов. Так же как рабочий буфер, буфер хранения слайсов распределяется в автоматическом режиме при каждом открытии экземпляра с помощью функции VPU_DecOpen()/ VPU_EncOpen(). Обычно для буфера хранения слайсов достаточно четверти исходного размера изображения YUV. Определить объем буфера можно по следующей формуле:

(SLICE_SAVE_SIZE) = (MAX_DEC_PIC_WIDTH*MAX_DEC_PIC_HEIGHT*3/4) // данный буфер для ASO/FMO

Если блоку VPU для декодирования потока Базового профиля H.264/AVC требуется буфер большего объема, VPU выдает данные о переполнении буфера.

1.3.6.4.4.3 Управление микрокодом (микро-ПО) BIT-процессора

Внутри блока VPU BIT-процессора есть собственная программная память, и содержимое этой программной памяти обновляется в динамическом режиме в зависимости от требуемого стандарта декодера. Преимуществом такой динамической перезагрузки микрокода является возможность снижения объема программной памяти. В этом есть особый смысл, поскольку чтобы управлять многими другими стандартами декодирования (несколькими экземплярами) в дуплексном режиме BIT-процессору обычно требуется несколько наборов микрокодов.

В силу того, что динамическая перезарузка, осуществляемая при работе нескольких экземпляров, выполняется самим блоком VPU, единственное, что должен сделать хост-процессор, это скопировать данный микрокод в зарезервированный буфер кода перед инициализацией VPU. Естественно, первая загрузка микрокода в программную память BIT-процессора должна выполняться хост-процессором отдельно.

Ниже представлен примерный код обработки микрокода BIT-процессора в ходе инициализации.

```
/* Скопировать полный вариант микрокода в буфер кода, зарезервированный в памяти SDRAM
*/ nCodeLen = ( sizeof(bit_code) / sizeof(bit_code[0]) );
pCodeData = malloc( nCodeLen * sizeof( unsigned short ) ); if( pCodeData )
{
// Отмена микрокода BIT-процессора
pData = pCodeData;
for( i = 0; i < nCodeLen; i += 2 )
{
*pData++ = bit_code[i + 1];
*pData++ = bit_code[i];
}
// Записать отмененный микрокод в память SDRAM
WriteSdramBurst( (unsigned char *)pCodeData, codeBuf, nCodeLen * sizeof( unsigned short ) , 0 );
free( pCodeData );
}
/* Загрузить микрокод BIT-процессора в программную память
*/ for (i = 0; i < 2048; ++i) {
data = bit_code[i]; VpuWriteMem(BIT_CODE_DOWN, (i << 16) | data);
}
```

1.3.6.4.4.4 Управление буфером битового потока

Буфер битового потока - это общий буфер между хост-процессором и блоком VPU, предназначенный для представления данных о закодированных изображениях, известных как битовый поток. В VPU в ходе декодирования или кодирования в целях максимизирования эффективности при фиксированном объеме области памяти применяется схема кольцевого буфера (известная также как схема циклического буфера).

Местоположение и объем буфера битового потока можно определить с помощью регистров главного интерфейса - CMD_ENC/DEC_SEQ_BB_START и CMD_ENC/DEC_SEQ_BB_SIZE. Поскольку циклический буфер подключается по способу сквозного соединения, он может моделировать линейный буфер в случае, если объем буфера достаточно большой. Помимо данных настроек хост-процессор может выбрать порядок следования байтов буфера потока и может включить или отключить опцию поверки наполненности/опустошения буфера.

Все эти опции управления данными буфера битового потока хранятся в специально выделенном регистре главного интерфейса - BIT_BITSTREAM_CTRL, и в ходе работы BIT-процессор к нему обращается.

1.3.6.4.4.4.1 Распределение буфера битового потока

Буфер битового потока должен быть выровнен по ширине шины памяти. Например, хост-процессором используется 64-разрядная шина, значит, битовый поток должен быть выровнен до границы 64 бит. Таким образом, устанавливать регистры CMD_ENC/DEC_SEQ_BB_START и CMD_ENC/DEC_SEQ_BB_SIZE на стороне хоста следует при полном соблюдении этого требования.

Значение объема буфера битового потока может находиться в диапазоне от 2 кбайт до 256 Мбайт. Рекомендуемым минимальным объемом буфера, необходимым для сглаживания изменений в экземпляре при работе с несколькими экземплярами, является объем выше 10 кбайт. В обычных условиях рекомендуется использовать буфер битового потока, максимальный объем которого менее 16 Мбайт (без ограничения). Условно оптимальный объем буфера битового потока сильно зависит от битовой скорости изображения кодека. Например, для скорости передачи потока 50 Мбит/с рекомендуется всегда использовать буфер объемом 8 Мбайт, между тем, для обеспечения большей эффективности при скорости 20 Мбит/с следует использовать буфер объемом 4 Мбайта.

1.3.6.4.4.4.2 Указатели для подкачки битового потока

Существует два типа указателей – указатель считывания (BIT_RD_PTR), при котором блок VPU считывает данные потока из буфера, и указатель записи (BIT_WR_PTR), при котором происходит наполнение буфера. Подобная схема предпочтительна для использования в ходе сеанса видеосвязи на базе пакетной передачи и в приложениях потокового видео, таких как телевизионное вещание или видеоконференцсвязь. При пакетной потоковой передаче на основе использования кольцевого буфера указатель считывания и указатель записи автоматически зацикливается на границах буфера.

Что касается управления этими указателями, то перед тем, как передать команду в VPU, на стороне хоста можно отрегулировать оба указателя. Поменять указатель считывания, пока VPU выполняет декодирование, нельзя, но можно отрегулировать указатель записи после подачи битового потока в буфер битового потока.

Перед тем как подавать новый блок битового потока в буфер, на стороне приложения должна быть проведена проверка того, есть ли в этом буфере битового потока пространство, доступное для записи. Оценить это можно с помощью указателя считывания, указателя записи и значения объема буфера. Для этих целей предназначена специальная функция API VPU_DecGetBitStreamBuffer(), которая оповещает приложения о местоположении указателя считывания и указателя записи, а также о наличии свободного пространства в буфере битового потока. Когда приходит возвращаемое значение этой функции API, приложение может загрузить в буфер битового потока новый блок данных, размер которого меньше объема свободного пространства буфера. Существует другая функция API - VPU_DecUpdateBitStreamBuffer(), которая позволяет приложениям получать уведомления об объеме бит, передаваемых в буфер битового потока (BIT_WR_PTR).

При работе с регистрами BIT_RD_PTR и BIT_WR_PTR на стороне хоста можно пользоваться функциями API:

VPU_DecSetRdPtr() обновляет регистр BIT_RD_PTR;

VPU_DecUpdateBitstreamBuffer() обновляет регистр BIT_WR_PTR.

1.3.6.4.4.4.3 Режимы обработки битового потока

Если битовый поток имеет достаточный объем, то для максимизирования эффективности блок VPU каждый раз загружает данные битового потока в 512-байтный модуль. Этот базовый загрузочный модуль называется блоком битового потока или просто блоком. Объем блока, как правило, составляет 512 байтов, и весь битовый поток загружается в модуль блоков. Если объем оставшихся в буфере битового потока данных меньше 512-байтного блока, то VPU будет действовать в зависимости от выбранного режима обработки битового потока.

Существует три режима обработки битового потока в VPU: режим прерывания, режим возврата и режим PicEnd. Выбрать один из режимов можно с помощью настройки StreamMode регистра BIT_BIT_STREAM_PARAM[4:3] на стороне хоста. Для декодирования по умолчанию установлен режим прерывания.

Режим обработки битового потока настраивается перед подачей команды декодирования изображения. В процессе работы VPU с хоста можно изменить режим прерывания на режим возврата или режим PicEnd. Однако чтобы переключиться обратно в режим прерывания хост должен дождаться пока не закончится декодирование текущего изображения. Также необходимо соблюдать осторожность при использовании регистра WR_PTR в режиме возврата или режиме PicEnd. После установки режима возврата или режима PicEnd хост не должен обновлять регистр WR_PTR до тех пор, пока не завершится декодирование одного изображения. В режиме возврата или PicEnd блок VPU выполняет декодирование до WR_PTR, даже если объем битового потока меньше 512 байт. Вот почему для безопасной работы при декодировании регистр WR_PTR должен находиться в статичном состоянии.

Режим прерывания (StreamMode = 0, режим, заданный по умолчанию)

Режим прерывания – это базовый режим обработки битового потока в VPU. Если в буфере битового потока остаются данные потока, объем которых меньше 512 байт, то блок VPU посылает на хост сигнал прерывания с запросом следующего объема битового потока. В этом случае устанавливается причина прерывания – опустошение буфера битового потока (14-й бит в регистре BIT_INT_REASON). После получения прерывания, хост должен заполнить буфер битового потока новыми данными закодированного изображения. Тем временем, VPU находится в состоянии ожидании блока данных битового потока, объемом более 512 байт.

Данный режим является самым простым и эффективным способом управления битовым потоком, но у него есть некоторые недостатки.

Если потоковая передача данных закодированного изображения кадра идет медленнее, чем эти данные расходуются, то в случае использования нескольких экземпляров это может привести к нежелательной задержке, вследствие того, что блок VPU переключает текущий экземпляр на другой только по завершении декодирования текущего экземпляра.

В неподвижных сценах с низкой битовой скоростью, таких как видеоконференцсвязь, объем данных закодированного изображения кадра зачастую бывает меньше блока объемом 512 байтов. В этом случае VPU в течение довольно продолжительного времени не сможет декодировать некоторые кадры.

Чтобы обработать такие данные, хост может переключить режим на режим возврата или режим PicEnd, что позволит продолжить декодирование лишь с небольшим битовым потоком.

Режим возврата (StreamMode = 1)

В режиме возврата до тех пор, пока, как и в режиме прерывания, весь 512-байтный блок не поступит в буфер, VPU не начнет декодирование даже небольшого битового потока. По окончании декодирования в VPU могут возникнуть следующие ситуации:

Если достигнуто окончание битового потока изображения, и декодирование всего кадра завершается без ошибок, то блок VPU возвращает результирующее изображение, как это обычно и происходит.

Однако если битовый поток имеет объем, недостаточный для декодирования кадра или заголовка последовательности, VPU возвращает регистр BIT_RD_PTR обратно на исходную позицию битового потока, где он и должен находиться перед выполнением команды PIC_RUN. (Вот почему данный режим называется режимом возврата.) В то же самое время VPU возвращает RetState = 0 и BitEmptyRollback = 1 в регистре RET_DEC_PIC_SUCCESS или RET_DEC_SEQ_SUCCESS. После поступления следующих данных закодированного изображения в буфер битового потока хост может снова начать декодирование или может сразу изменить экземпляр для других приложений.

Режим возврата очень эффективен для предотвращения «зависания» особенно в приложениях, в которых задействуется несколько экземпляров, или в случае очень медленной потоковой передачи закодированного изображения. Существует несколько требований, которые следует соблюдать при проведении настройки и управления на стороне хоста:

Изменить режим прерывания на режим возврата на стороне хоста можно в любой момент.

Перейти с режима возврата на режим прерывания на стороне хоста, пока работает VPU, нельзя.

Нельзя обновить регистр BIT_WR_PTR на стороне хоста, пока не завершится декодирование одного изображения.

Чтобы обеспечить плавную работу в режиме возврата буфер битового потока должен быть достаточно большим, чтобы в нем мог поместиться один кадр. Если буфер не большой, то хост снова должен подать данные битового потока для возвращенного кадра.

Режим PicEnd (StreamMode = 2)

Так же как в режиме возврата, в режиме PicEnd VPU декодирует битовый поток, объем которого меньше объема 512-байтного блока. После декодирования в VPU могут возникнуть следующие ситуации:

Если достигнуто окончание битового потока изображения, и декодирование всего кадра завершается без ошибок, то блок VPU возвращает результирующее изображение, как это обычно и происходит.

Однако если битовый поток имеет объем, недостаточный для декодирования кадра или заголовка последовательности, VPU «понимает», что что-то идет неправильно, и последующим макроблокам присваивается статус ошибочных. VPU генерирует соответствующие макроблоки вместо пропущенных макроблоков с помощью схемы маскирования ошибок. По окончании маскирования ошибок VPU возвращает RetState = 1 в регистре RET_DEC_PIC_SUCCESS и количество пропущенных макроблоков в регистре RET_DEC_PIC_ERR_MB.

Данный режим очень эффективен для приложений с малой задержкой, и если буфер битового потока имеет небольшой объем.

Моделирование воспроизведения файла с использованием режима PicEnd

Режим PicEnd позволяет смоделировать процесс воспроизведения файла посредством назначения начала и конца данных определенного файла с помощью регистров BIT_RD_PTR и BIT_WR_PTR, соответственно. VPU может декодировать сохраненные данные закодированного изображения непосредственно из регистра BIT_RD_PTR в регистр BIT_WR_PTR, без необходимости копировать данные, предназначенные для декодирования, в буфер битового потока. Чтобы обеспечить работу в таком режиме, регистры BIT_RD_PTR и BIT_WR_PTR должны находиться в области буфера битового потока. Другими словами, на стороне хоста следует установить буфер битового потока, имеющий объем, достаточно большой для размещения нескольких файлов. Обычно для моделирования воспроизведения файла в режиме PicEnd почти весь объем внешней памяти на стороне хоста выделяется под буфер битового потока.

Повторное использование блока файлов

В ходе моделирования воспроизведения файла в режиме PicEnd для каждого кадра хост должен установить регистры BIT_RD_PTR и BIT_WR_PTR. Однако в некоторых случаях объем закодированных данных изображения между регистрами BIT_RD_PTR и BIT_WR_PTR может превышать один кадр. Такое происходит по одной из нижеследующих причин:

с демультиплексора получены неточные данные;

это особые случаи, возникающие при кодировании стандартов, например, пакетированный кадр в MPEG-4;

закодированные данные изображения не расходовались вследствие того, что из-за неготовности на тот момент буфера кадра VPU не мог выполнить декодирование.

В этих случаях блок файлов должен использоваться повторно. Другими словами, хост должен снова подать файл в VPU. Учитывая каждую из указанных выше ситуаций, VPU может распознать и повторно использовать блок файлов.

Для случая 1: подача неточных данных битового потока

Диагностика: VPU, путем сравнения объема израсходованных при декодировании байт с реальным размером файла, определяет, сколько кадров находится в блоке. Если разница между этими объемами превышает внутреннее пороговое значение, то хост может распознать, где в блоке файлов находится больше одного кадра.

Решение: VPU продолжает декодирование с оставшимися данными блока.

Для случая 2: в MPEG-4 обнаружен пакетированный кадр

Диагностика: в ходе декодирования потока MPEG-4 VPU возвращает 1 в PackedChunkReuse - флаг 16-го бита регистра RET_DEC_PIC_SUCCESS.

Решение: хост повторно использует данные из второй части блока.

Для случая 3: буфер кадра заполнен

Диагностика: VPU возвращает -1 в RET_DEC_PIC_CUR_IDX из-за того, что буфер кадра непригоден для декодирования.

Решение: хост повторно использует весь битовый поток.

Для случая 4: в H.264 обнаружен NPF

Диагностика: VPU возвращает 0 в DecFrameInfo – флаг 15-го бита регистра RET_DEC_PIC_TYPE и одновременно с этим возвращает 1 в ChunkReuse - флаг 16-го бита регистра RET_DEC_PIC_SUCCESS.

Решение: хост повторно использует данные из оставшейся части блока.

Существует несколько ограничений, на которые следует обратить внимание при проведении настройки и управления на стороне хоста.

Изменить режим прерывания на режим возврата на хосте можно в любой момент.

Перейти с режим возврата на режим прерывания на хосте, пока работает VPU, нельзя.

Нельзя обновить регистр BIT_WR_PTR на хосте, пока не завершится декодирование одного изображения.

Возможность работы с несколькими экземплярами

Благодаря возможности работы с несколькими экземплярами и для идентификации каждого из экземпляров, независимо от того сколько экземпляров создается с помощью RunIndexes, на стороне хоста следует вести собственный список регистров BIT_RD_PTR и BIT_WR_PTR. BIT-процессором в регистре BIT_RD_PTR устанавливается адрес считывания битового потока внешней памяти SDRAM с текущим RunIndex, в результате чего на сторону хоста поступает уведомление об объеме битового потока, который был считан и декодирован для определенного экземпляра.

1.3.6.4.4.5 Управление сигнализацией прерывания

Чтобы добиться максимальной эффективности при управлении VPU, в блоке VPU для завершения требуемой операции и информирования о том, что буфер потока пуст/заполнен может использоваться сигнализация прерывания. Но некоторым командам с быстрым откликом опция сигнализации прерывания не предоставляется ввиду ее бесполезности в таких случаях.

На текущий момент “C&M” предоставляет опцию сигнализации прерывания следующим командам:

BIT_RUN_COMPLETE: инициализация BIT-процессора была выполнена после установки BIT_CODE_RUN в 1;

SEQ_INIT: инициализация последовательности VPU выполнена;

SEQ_END: завершение последовательности VPU выполнено;

PIC_RUN: обработка изображения VPU выполнена;

SET_FRAME_BUF: регистрация буфера кадра VPU выполнена;

ENC/DEC_PARA_SET: настройка параметров для VPU выполнена;

DEC_BUF_FLUSH: принудительный сброс буфера битового потока декодером выполнен.

Среди рассмотренных выше команд выполнение команд SEQ_INIT и PIC_RUN при определенных условиях может привести к останову VPU в случае заполнения или опустошения буфера битового потока. Поэтому, если в случае использования схемы кольцевого буфера при опустошении или наполненности буфера включается опция прерывания, то это поможет избежать излишнего расхода цикла в хост-приложениях.

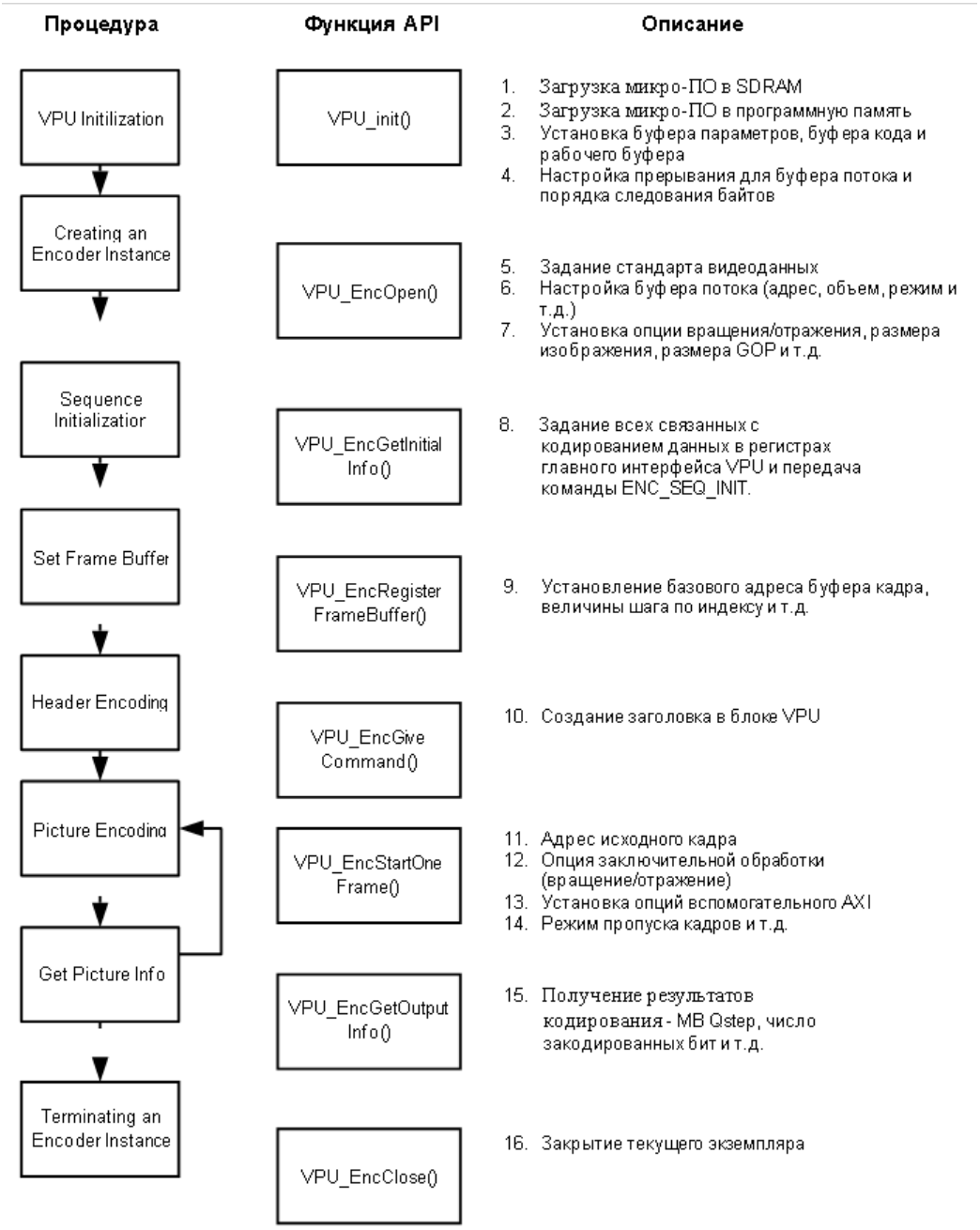
Включение или отключение каждого прерывания осуществляется при записи 0 или 1 в соответствующее битовое поле регистра включения прерывания. При получении сигнала прерывания приложение может определить источник прерывания при проверке значения регистра причины прерывания.

В ситуациях, когда сигнализацию прерывания использовать сложно, рассмотренные типы сигнализации прерывания можно заменить на схему опроса считывания регистра BIT_BUSY_FLAG в BIT-процессоре.

1.3.6.4.4.5.1 Управление кодером

1.3.6.4.4.5.1.1 Общий порядок кодирования

На рисунке 1.159 представлен краткий алгоритм управления кодером и показана взаимосвязь между потоком управления кодером и функцией API. Кроме того, на нем дано краткое описание каждого этапа такого взаимодействия.



VPU Initialization – Инициализация VPU
Creating an Encoder Instance – Создание экземпляра кодера
Sequence Initialization – Инициализация последовательности
Set Frame Buffer – Установка буфера кадров
Header Encoding – Кодирование заголовка
Picture Encoding – Кодирование изображения
Get Picture Info – Получение информации об изображении
Terminating an Encoder Instance – Закрытие экземпляра кодера

Рисунок 1.159 - Диаграмма управления кодером с функциями API

1.3.6.4.4.5.2 Создание экземпляра кодера

После инициализации VPU на первом этапе запуска кодера в работу создается экземпляр кодера, и выделяется указатель экземпляра, чтобы блок VPU мог идентифицировать то, какой из экземпляров работает в данный момент. Определить это можно с помощью одной функции API - VPU_EncOpen().

При создании нового экземпляра кодера, приложению с помощью структуры EncOpenParam следует задать внутренние параметры данного экземпляра кодера. В данную структуру входят следующие данные о новом экземпляре кодера:

Адрес и объем буфера битового потока

Физический адрес стартового адреса буфера битового потока и его объем

Стандарт кодека

Стандарт видеокодека - H.263, MPEG-4, H.264/AVC

Размер изображения

Ширина и высота изображения, подлежащего кодированию

Целевая частота кадров и целевая битовая скорость с параметрами модели VBV - initialDelay и vbvBufferSize

Параметры режима верификации буфера видеоданных (VBV) являются необязательными, даже если включена опция контроля передачи данных.

Размер GOP

Частота повторяющихся intra-изображений (или IDR) на выходе потока закодированных данных.

Включение/отключение слайсов, режим и размер слайса

Если слайс включен, VPU может генерировать столько слайсов, сколько конфигурируется в ходе процесса кодирования. VPU может начать новый слайс в одном из 2 режимов слайсов. Если количество сгенерированных бит равно значению в регистре размера слайса, или когда число макроблоков равно значению в регистре размера слайса, VPU переходит к следующему слайсу.

Прочие опции – enableAutoSkip, intraRefresh и т.д.

Включение автоматического пропуска изображений, когда контрольная сумма на выходе достаточно большая, а также включение опции intra-регенерации для выработки устойчивости к ошибкам и ряда макроблоков intra в изображении, не являющимся intra-изображением.

Включение режима кольцевого буфера

Приложением определяется, будет ли использоваться схема потоковой передачи на основе кольцевых буферов. Если данная опция отключена, то будет использоваться схема покадровой потоковой передачи совместно со схемой линейного буфера.

Примечание Добавление данного поля позволяет настроить потоковый режим отдельно для каждого экземпляра кодера.

Включение режима распределения динамического буфера

Динамическое распределение буфера битового потока изображения происходит на стороне приложения при включении опции распределения динамического буфера и только в случае, если отключен режим кольцевого буфера, другими словами, только в режиме линейного буфера. Если опция распределения динамического буфера отключена, то при кодировании изображения используется адрес и объем буфера битового потока, возвращаемые по команде VPU_EncGetInitialInfo(). Если опция распределения динамического буфера включена, то адрес и объем буфера битового потока изображения выдается приложениями в динамическом режиме при вызове операции кодирования изображения.

Параметры, характерные для стандартов видеокодирования

Определение параметров характерных для каждого стандарта видеокодеков - инструменты обеспечения устойчивости к ошибкам в MPEG-4, приложения в H.263 и параметры деблокирования в H.264/AVC. Например, у кодера MPEG-4 много параметров кодирования, включая такие параметры, как разделение данных, RVLC, идентификатор версии и т.д. Получить более подробную информацию о параметрах, характерных для определенных стандартов, можно в справочном руководстве к API.

Параметры, связанные с оценкой параметров движения

В VPU есть блок оценки параметров движения, который обеспечивает различные параметры приложений, используемые при конфигурировании - RcIntervalMode, MbInterval, MESearchRange, MEUseZeroPmv и IntraCostWeight. Подробное описание этих параметров можно найти в справочном руководстве к API.

С помощью рассмотренных выше опций приложения могут получать хорошо оптимизированные выходные данные кодирования, которые отвечают требованиям целевых приложений. В частности, такие выходные данные как sliceReport, mbReport, qpReport т.д. могут служить полезным инструментом для разработчиков приложений, которые сталкиваются с серьезными ограничениями в целевых приложениях.

Например, в случае, когда пакет данных имеет фиксированный размер, на стороне приложения может возникнуть необходимость добавить к определенному числу бит один слайс. Для приложения ввиду изменяющихся характеристик длительности процесса кодирования может оказаться сложным гарантировать, чтобы размер слайса на выходе был меньше установленного размера. По этой причине во многих приложениях слайс обычно делят на два пакета, вследствие чего процесс пакетирования оказывается крайне неэффективным. Чтобы добиться высокой эффективности, и упростить процесс пакетирования, на стороне приложения с помощью (packet_size: N) можно задать размер слайса в пределахN, что позволит сделать размер выходного слайса меньше размера пакета. Затем приложение может разместить слайс в пакете, просто указав границы слайса, определенные VPU в выходных данных кодера.

После создания экземпляра кодера с этими параметрами, приложение уже не сможет изменить исходные параметры, определенные на данном этапе. Теоретически, если приложение захочет изменить какие-либо базовые параметры, оно должно закрыть данный экземпляр и заново создать экземпляр кодера с новыми исходными параметрами. Однако на практике приложению требуется изменить лишь некоторые из исходных параметров, в зависимости от среды целевого приложения. С помощью команды динамического конфигурирования API блока VPU дает возможность приложениям динамически сконфигурировать часть этих исходных параметров. Подробное описание VPU_EncGiveCommand() находится в справочном руководстве к API.

Функция API - VPU_EncOpen() не требует проведения никаких операций на стороне VPU, но гарантирует, что все внутренние параметры и информация о буфере битового потока будут использоваться на последующих этапах.

1.3.6.4.4.5.3 Конфигурирование VPU для экземпляра кодера

1.3.6.4.4.5.3.1 Инициализация последовательности

После регистрации всей информации необходимой для нового экземпляра кодека на стороне хост-приложения необходимо сконфигурировать VPU должен быть сконфигурирован для возможности предоставить новый экземпляр кодека. Данная процедура выполняется посредством настройки всех необходимых данных кодера в регистрах главного интерфейса VPU, и последующей передачи в VPU команды ENC_SEQ_INIT для инициализации операции конфигурирования внутри VPU.

Все указанные процессы осуществляются, главным образом, с помощью функции API - VPU_EncGetInitialInfo().. В результате этого функция возвращает выходной параметр, который очень важен для операций кодирования – минимальное количество буферов кадра. Обычно данный процесс не требует много времени, и он выполняется всего один раз в начале работы экземпляра кодирования. Поэтому рекомендуется для данной функции использовать сигнализацию прерывания, но применить ее можно только после выполнения данной операции и при включении соответствующего бита в регистре включения прерывания.

1.3.6.4.4.5.3.2 Регистрация буферов кадра

Данный процесс конфигурирования завершается регистрацией буферов кадра в блоке VPU для их использования при последующем кодировании изображения. На данном заключительном этапе конфигурирования из VPU_EncGetInitialInfo()возвращается очень важный параметр – минимальное количество буферов кадра. Он показывает то, что для правильной работы кодера данным приложениям следует зарезервировать в VPU указанное (как минимум) число буферов

кадра. Конфигурирование можно выполнить с помощью функции API VPU_EncRegisterFrameBuffer().

1.3.6.4.4.5.3.3 Генерирование синтаксисов заголовков верхнего уровня

Когда при вызове функции VPU_EncGetInitialInfo()открывается экземпляр кодера, приложения с помощью функции VPU_EncGiveCommand()ДОЛЖНЫ сгенерировать синтаксисы заголовков верхнего уровня – например, заголовки VOS/VO/VOL в MPEG-4 и SPS/PPS в AVC.

Существует два возможных пути генерирования этих синтаксисов заголовков – с помощью регистра PARA_BUF или с помощью буфера битового потока. Рекомендуется получать синтаксисы заголовков с помощью команды ENC_PUT_AVC/MP4_HEADER через буфер битового потока. Если приложениями будет использоваться данный набор команд, то в результате синтаксисы заголовков будут находиться в буфере битового потока в соответствии с заданными настройками порядка следования битов.

Другой способ генерирования синтаксисов заголовков – через функцию PARA_BUF. Это можно сделать с помощью команд ENC_GET_VOS_HEADER, ENC_GET_VO_HEADER или ENC_GET_VOL_HEADER в случае MPEG-4, и с помощью команд

ENC_GET_SPS_RBSP или ENC_GET_PPS_RBSP в случае H.264/AVC. Несмотря на потоковый режим, данная команда позволит сгенерировать синтаксисы заголовков и записать их в PARA_BUF, а не в буфер битового потока. Но в этом случае порядок следования битов будет обратным. Поэтому, в системе с прямым порядком следования битов на стороне хост-процессора необходимо выполнить преобразование порядка следования битов.

Примечание Способ генерирования синтаксисов заголовков через функцию PARA_BUF считается устаревшим. Поэтому для генерирования синтаксисов заголовков верхнего уровня настоятельно рекомендуется пользоваться первым методом – с помощью буфера битового потока.

1.3.6.4.4.5.4 Работа кодера изображения в VPU

1.3.6.4.4.5.4.1 Загрузка входного изображения в режиме YUV

Перед запуском в работу кодера изображения хост-приложения должны предоставить исходное изображение в режиме YUV в формате 4:2:0 с предварительно установленным размером для H.263, MPEG-4 и H.264/AVC.

Новое исходное изображение может прийти от внешнего видеотерминального устройства, например, такого как сенсор CMOS. Чтобы предотвратить простой блока VPU в ходе ожидания завершения получения исходного изображения, рекомендуется воспользоваться схемой параллельной буферизации исходного изображения. При этом кодер перед началом работы не будет тратить цикл на простой.

1.3.6.4.4.5.4.2 Инициализация кодирования изображения

При активации операции кодирования изображения приложениям следует предоставить в VPU следующую информацию:

адрес исходного кадра;

В адресе исходного кадра содержится базовый адрес каждого компонента исходного изображения в режиме YUV, включая базовый адрес буфера кадра по компонентам яркости и цветности (или Cb и Cr).

опции принудительного пропуска кадра и принудительной передачи I-изображения;

Принудительный пропуск кадра – это безусловный пропуск кодирования текущего кадра, а в случае принудительной передачи I-изображения текущий кадр, безусловно, кодируется как I-кадр.

выходные данные – информация о макроблоках, информация о векторе движения и информация о слайсах;

На стороне приложения можно включать или отключать опцию генерирования блоком VPU в ходе кодирования таких информативных выходных данных, как информация о макроблоках (MB QP, границы слайса), информация о векторе движения и информация о слайсах.

опция заключительной обработки (вращение/отражение);

Установление опции вращения и отражения в VPU. Чтобы кодировать исходный кадр VPU считывает его и поворачивает.

1Благодаря этой дополнительной возможности опция автоматического генерирования синтаксисов заголовков (как VOL в MPEG-4, SPS/PPS в AVC) больше не поддерживается.

шаг квантизации;

Установление шага квантизации для кодирования текущего изображения. Если включена опция управления скоростью, данный этап пропускается.

вспомогательный AXI;

Установление опции использования вспомогательного AXI в целях экономии полосы пропускания. Существует 4 опции использования вспомогательного AXI - BIT_USE, IP_USE, DBLKY_USE и DBLKC_USE.

буфер кадра;

Установление режима буфера кадра - с чередованием компонент CbCr и порядок следования байтов.

После настройки всех этих параметров в блоке VPU хост-процессор, передав в VPU команду PIC_RUN, может начать кодирование изображения. Все предварительно установленные процессы выполняются при вызове всего одной функции API - VPU_EncStartOneFrame() со структурой EncParam. Данная функция API запускает операцию кодирования изображения. Однако ее возврат из API еще не означает, что кодирование изображения завершено.

Величина шага квантизации, передаваемая в VPU с регистром ENC_PIC_RUN, важна только в случае, когда отключена опция управления скоростью передачи. Эта дополнительная опция предоставляется в целях поддержания работы VBR при кодировании, ориентированном на конкретное приложение.

Опция принудительного пропуска кадра используется тогда, когда временно отсутствует возможность кодирования нового изображения по причинам внешнего характера, связанным с кодером. Другими словами, принудительный пропуск кадра выполняется в приложениях тогда, когда кодирование изображения по некоторым причинам внешнего характера представляется проблематичным. Например, можно посчитать, что состояние канала на текущий момент недостаточно хорошее, поэтому передача закодированного потока невозможна. После этого на стороне приложения операция кодирования приостанавливается до тех пор, пока не будет включена опция принудительного пропуска кадра.

Кроме того, режим автоматического пропуска кадра предусмотрен в алгоритме контроля скорости передачи, который используется для ограничения объема выходных данных битового потока при заданной целевой частоте битов. Например, если размер битов выходных данных превышает целевую битовую скорость, то алгоритм управления скоростью передачи увеличивает шаг квантизации, тем самым, уменьшает размер битов выходных данных. Однако, даже, несмотря на то, что значение шага квантизации можно увеличивать, бывают ситуации, когда VPU не может обеспечить соответствие целевой битовой скорости. В таком случае блок VPU может автоматически пропустить текущее изображение, чтобы обеспечить соответствие заданной на хост-процессоре целевой битовой скорости. С помощью настройки соответствующего регистра на стороне хоста можно включать и отключать режим автоматического пропуска кадра.

Опция принудительной передачи I-кадра является еще одной полезной для приложений опцией, которая позволяет в ходе операции декодирования передавать сообщения об ошибке к удаленному приемному устройству. Даже, несмотря на то, что на стороне декодера можно реализовать схему маскирования определенных ошибок или восстановления при возникновении ошибки, лучший способ восстановить работоспособность декодера после ошибки – это передать I-кадр. Поэтому благодаря опции принудительной передачи I-кадра приложение может очень быстро добиться восстановления работоспособности удаленного приемного устройства после ошибки. При инициализации кодирования изображения, чтобы изменить тип текущего

изображения на I-изображение, на стороне хоста следует установить опцию принудительной передачи I-кадра.

1.3.6.4.4.5.4.3 Выполнение операции кодирования изображения

Операция кодирования изображения занимает некоторое количество времени, и в ходе ожидания завершения выполнения этой операции приложение может перейти к выполнению других задач, например, пакетированию закодированного потока данных для их последующей передачи. Приложением могут использоваться две разные схемы распознавания завершения операции кодирования изображения: опрос регистра состояния и сигнализация прерывания. Если в приложении будет использоваться схема опроса, то нужно будет просто проверить BusyFlag в соответствующем регистре. Вызов функции VPI_IsBusy() приведет к таким же результатам.

Сигнализация прерывания может быть наиболее эффективным способом проверки выполнения заданной команды. Сигнал прерывания для команды ENC_PIC_RUN отображается в bit[3] регистра включения прерывания. Поэтому благодаря такому специальному сигналу прерывания, передаваемому из VPU, система легко «поймет», что операция кодирования изображения завершена.

1.3.6.4.4.5.4.4 Обработка потока данных кодера

Если буфер битового потока кодера имеет объем, достаточный для хранения потока изображений большого размера, то в ходе операции кодирования кодеру не потребуется битовый поток. Поле завершения операции кодирования хост-приложение может считывать закодированный битовый поток в соответствии с требованиями пакетирования.

Но если буфер битового потока кодера недостаточно большой для хранения всего потока изображений, может произойти переполнение буфера кодера, и пока ситуация с переполнением буфера не будет разрешена, кодер в блоке VPU не сможет выполнять никакие задачи и будет находиться в «зависшем» состоянии. Поэтому чтобы избежать подобного нежелательного эффекта «зависания» кодера, приложение должно продолжать считывать закодированный битовый поток из буфера потоков в процессе кодирования изображений. И когда буфер битового потока заполнится, VPU выдаст предупреждение о прерывании, чтобы уведомить хост-процессор об этом событии.

При использовании схемы кольцевого буфера с ограниченным объемом буфера битового потока кодера, рекомендуется считывать битовый поток в ходе операции кодирования. С помощью двух специальных функций - VPU_EncGetBitStreamBuffer() и VPU_EncUpdateBitStreamBuffer()приложение может управлять указателем считывания при кодировании, пока обеспечивается доступ к буферу битового потока кодера.

Однако если опция кольцевого буфера отключена, а в целях защиты VPU от «зависания», большого объема буфера битового потока (используется линейный буфер) достаточно для хранения данных одного кодированного изображения, то хост может считать закодированный битовый поток только после завершения кодирования каждого изображения. В этом случае пока в VPU идет кодирование изображения, система может успешно выполнять другие задачи. Если используется такая опция потоковой покадровой передачи, то использование функций VPU_EncGetBitStreamBuffer() и VPU_EncUpdateBitStreamBuffer() является не целесообразным.

1.3.6.4.4.5.5 Закрытие экземпляра кодера

Когда система собирается завершать операцию кодирования и закрывать экземпляр кодера, она просто сбрасывает указатель этого экземпляра и, передав в VPU команду SEQ_END, уведомляет его о том, что данный экземпляр закрывается. Сделать это можно просто вызвав функцию VPU_EncClose().

1.3.6.4.4.5.6 Команды динамического конфигурирования

Иногда в ходе процесса кодирования изображений приложению необходимо передать в блок VPU некоторые специальные команды, например, команду о необходимости развернуть исходные изображения перед началом кодирования, команду вставки синтаксисов заголовка верхнего уровня и т.д. Интерфейсом API VPU предоставляется набор команд для поддержания специальных запросов подобного типа от хост-приложений:

повернуть и отразить исходный кадр перед началом кодирования;

выделить синтаксисы заголовков верхнего уровня, например, заголовка VOS, VO и VOL в MPEG-4, и SPS и PPS в H.264/AVC для использования вне канала;

вставить синтаксисы заголовков верхнего уровня, например, заголовок VOS, VO и VOL в формате MPEG-4, и SPS и PPS в формате H.264;

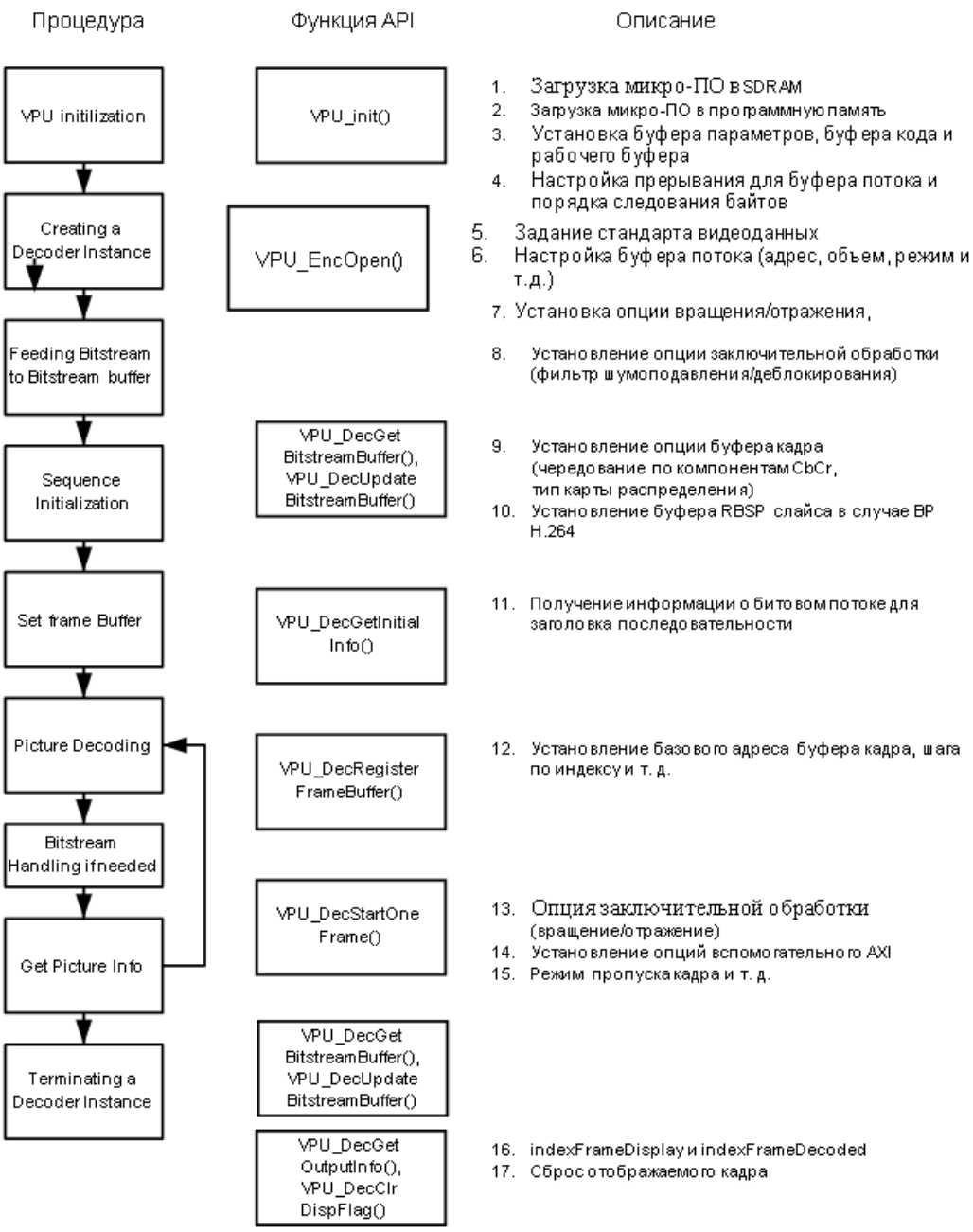
изменить параметры кодера, например, битовую скорость, частоту кадров, количество GOP, режим слайсов и т.д. в динамическом режиме в ходе кодирования изображений/

Данные значения можно обновить при выполнении функции VPU_EncStartOneFrame().

1.3.6.4.4.6 Управление декодером

1.3.6.4.4.6.1 Общий порядок декодирования

На рисунке 1.160 представлен краткий алгоритм управления декодером и показана взаимосвязь между потоком управления декодером и функцией API. Кроме того, на нем дано краткое описание каждого этапа этого взаимодействия.



VPU Initialization – Инициализация VPU
Creating an Decoder Instance – Создание экземпляра декодера
Feeding Bitstream to Bitstream Buffer – Подача битового потока в буфер битового потока
Sequence Initialization – Инициализация последовательности
Set Frame Buffer – Установка буфера кадров
Picture Decoding – Декодирование изображения
Get Picture Info – Получение информации об изображении
Terminating a Decoder Instance – Завершение экземпляра декодера

Рисунок 1.160 -Диаграмма управления декодером с функциями API

1.3.6.4.4.6.2 Создание экземпляра декодера

После инициализации VPU на первом этапе запуска декодера в работу создается экземпляр декодера, и выделяется указатель, определяющий экземпляр декодера. Определить это можно с помощью одной функции API - VPU_DecOpen().

При создании нового экземпляра кодера, приложению с помощью структуры DecOpenParam следует установить внутренние параметры данного экземпляра декодера. В данную структуру входят следующие данные о новом экземпляре декодера:

адрес и объем буфера битового потока;

Физический адрес стартового адреса буфера битового потока и его объем

стандарт кодека;

Стандарт декодера видеоданных - H.264/AVC, VC-1, MPEG-2, RV и т. д.

объем и адрес буфера для хранения RBSP СЛАЙСА.

В Базовом профиле H.264/AVC отсутствует возможность FMO/ASO. Для потока, закодированного по Базовому профилю H.264/AVC, VPU считывает поток данных, переупорядочивает их и записывает в буфер для хранения RBSP СЛАЙСА. После проведения этой предварительной операции VPU может корректно декодировать базовый поток H.264/AVC с использованием переупорядоченного потока в буфере для RBSP СЛАЙСА. Как уже ранее упоминалось, данный буфер действует только в Базовом профиле H.264/AVC. Это значит, что нет необходимости задавать объем и адрес буфера для хранения RBSP СЛАЙСА в других стандартах видеоданных.

1.3.6.4.4.6.3 Конфигурирование VPU для экземпляра декодера

1.3.6.4.4.6.3.1 Подача битового потока в буфер битового потока

При инициализации последовательности декодирования SEQ_INIT, чтобы считать конфигурации декодера, необходимо выполнить разбор синтаксисов заголовков верхнего уровня, таких как, VOS/VO/VOL в MPEG-4 и SPS/PPS в H.264/AVC. Поэтому чтобы запустить команду SEQ_INIT, система должна наполнить буфер битовым потоком достаточного объема.

В некоторых приложениях нельзя гарантировать, что данные типы синтаксисов заголовков находятся в начале битового потока. В этом случае система должна подавать поток входных данных в буфер битового потока до тех пор, пока из потока входных данных в VPU не поступит вся необходимая информация.

В режиме кольцевого буфера для подачи входного битового потока на стороне хост-приложения необходимо иметь данные относительно свободного пространства в буфере битового потока. Определить эту информацию можно с помощью указателя считывания, указателя записи и зная объем буфера битового потока. Если в буфере битового потока есть свободное пространство, система может загружать входной поток декодера напрямую в буфер битового потока. И по окончании загрузки потока, обновив указатель записи потока, система должна передать информацию об объеме загруженного потока.

Чтобы получить указатель считывания и указатель записи потока, а также информацию о свободном пространстве интерфейс API блока VPU при вызове VPU_DecGetBitstreamBuffer()предоставляет обновленную функцию API. Также с помощью функции API VPU_DecUpdateBitstreamBuffer() можно получить обновленный указатель записи.

Примечание Для лучшего понимания логики данной операции можно обратиться к функции WriteBsBufHelper(), описанной в файле src/ vpuhelper.c file.

1.3.6.4.4.6.3.2 Инициализация последовательности

После того, как будет создан новый экземпляр, а входной битовый поток будет подан в буфер битового потока, приложение может направить в VPU команду DEC_SEQ_INIT, чтобы получить из битового потока информацию конфигурирования декодера. По завершении разбора синтаксисов заголовков декодер возвращает следующую важную информацию относительно конфигурации декодера:

размер изображения;

Ширина и высота изображения

информация о прямоугольнике обрезки изображения;

Информация о прямоугольнике обрезки изображения декодера H.264/AVC, в которой отражен сдвиг левой верхней точки и нижней правой точки относительно оригинального изображения из буфера кадра.

частота кадров; Частота кадров декодера

минимальное количество буферов кадра;

дополнительная информация о формате MPEG-4;

Включены или отключены опции обеспечения устойчивости к ошибкам MPEG-4, такие как, разделение данных или двунаправленное VLC, а также режим коротких заголовков видеоданных * задержка в буфере кадров при переупорядочении отображения данных

Число задержек кадра для поддержания переупорядочения отображения данных в декодере H.264/AVC.

индикация опции деблокирования Приложения-J;

Данный флаг показывает, включена или отключена опция деблокирования декодера H.263. Если для формата H.263 используется внешний фильтр пост-деблокирования, то данный флаг позволяет избежать повторения использования внутриканального фильтра деблокирования H.263 и внешнего фильтра пост-деблокирования.

расчетные размеры буфера для хранения СЛАЙСА;

Объем буфера для хранения СЛАЙСА. Блок VPU выдает данные о двух разных объемах – рекомендованном объеме буфера и наихудшем варианте объема буфера.

Размер изображения, определенный из битового потока, может не быть кратным 16x16. Но для правильной работы декодера объем буфера кадра должен быть кратным размеру 16x16. Поэтому по окончании операции перекрывания, возвращаемые значения объема будут изменены на величину кратную 16x16. Зная размер изображения и минимальное количество буферов кадра, система должна зарезервировать буферы кадра и передать их VPU перед началом декодирования изображения.

Задержка в буфере кадра – это специальный параметр для формата H.264, который обеспечивает поддержку переупорядочения отображения данных. Если на стороне приложения требуется обеспечить поддержку переупорядочения отображения данных и для этого нужно, например, 5 дополнительных буферов кадра, то первый вывод декодированных данных на экран начнется после декодирования 6-го кадра. Теоретически, при переупорядочении отображения данных допустима задержка 16 кадров (максимально).

Для обработки операции SEQ_INIT блоком VPU обеспечивается функция API VPU_DecGetInitialInfo().

О завершении данной функции на хост-процессор будет передан специальный сигнал прерывания. Проверить это можно также с помощью флага опроса BusyFlag на хост-процессоре.

Важным вопросом в процессе SEQ_INIT является вопрос об обработке ошибок, потому что любые ошибки в синтаксисах заголовков верхнего уровня приводят к серьезным проблемам при декодировании. В качестве помощи при обнаружении ошибок к таким синтаксисам заголовков, как правило, добавляется несколько маркерных битов. Если синтаксисы заголовков, включенных в поток, содержат критические ошибки или синтаксисы заголовков не приходят в течение длительного времени, то блок VPU может «зависнуть» на выполнении данной задачи, и никакие другие экземпляры в VPU работать не будут. Поэтому API VPU предоставляет специальную функцию, которую можно использовать только в таких серьезных случаях - VPU_DecSetSeqInitEsc(). При вызове данной функции, при условии, что буфер потока пуст, блок VPU автоматически завершает операцию SEQ_INIT. Затем хост-приложение должно решить, стоит ли закрыть данный экземпляр и попытаться выполнить SEQ_INIT после запуска другого экземпляра декодера. После того как данная проблема будет решена, настоятельно рекомендуется сбросить внутренний флаг ESCAPE, вызвав еще раз функцию VPU_DecSetSeqInitEsc().

1.3.6.4.4.6.3.3 Регистрация буферов кадра

Процесс конфигурирования завершается регистрацией в VPU буферов кадра для последующего декодирования изображений. На данном заключительном этапе конфигурирования значение минимального количества буферов кадра, который является одним из возвращаемых после VPU_DecGetInitialInfo() параметров, будет очень большим. Данный параметр показывает, что для корректной работы декодера приложению следует зарезервировать в VPU указанное количество буферов кадра (как минимум).

Объем буферов кадра можно рассчитать, зная ширину и высоту изображения. Если и ширина, и высота изображения кратны 16, то размер изображения и будет показывать объем буферов потока. В противном случае приложению следует применить к ширине и высоте изображения операцию перекрывания, чтобы число наименьшее из кратных 16 было больше ширины или высоты изображения.

Дополнительно к регистрации буферов кадра в VPU на данном этапе выполняется регистрация буфера для хранения слайса. Указания по рекомендуемому объему буфера выдаются при вызове функции VPU_DecGetInitialInfo().

Адреса буферов кадра не обязательно должны быть непрерывными. Адреса цветовых плоскостей кадра также могут прерываться.

1.3.6.4.4.6.4 Запуск декодера изображений в VPU

1.3.6.4.4.6.4.1 Инициирование декодирования изображения

При активации операции декодирования изображения приложение должно предоставить VPU следующие опции:

включение поиска I-кадра;

Включение или отключение опции поиска I-кадра (IDR в формате H.264/AVC).

режим пропуска кадра;

Включение или отключение возможности пропуска битового потока при декодировании следующего кадра.

вспомогательный AXI;

VPU предоставляет опцию, при которой для доступа к внутренней SRAM в целях повышения эффективности декодирования при сокращении полосы пропускания системы используется вспомогательный AXI. VPU предоставляет для вспомогательного AXI шесть опций - BIT_USE, IP_USE, DBLKY_USE, DBLKC_USE, OVL_USE и BTP_USE. Поэтому на стороне хоста следует определить, какую из опций включить после анализа системных ресурсов и задать базовый адрес каждой области, используемой в этих целях. Данный процесс выполняется при вызове функции VPU_DecGiveCommand(). Если на хосте данный процесс пропускается, от опция вспомогательного AXI отключается.

Примечание Более подробную информацию об объеме памяти SRAM, необходимом для каждой из данных опций, можно найти в Excel-файле «Дополнительная полоса пропускания SRAM».

После предоставления всех этих параметров в VPU система, передав команду DEC_PIC_RUN, может начать декодирование изображения.

Для обработки этих сложных операций API VPU предоставляет API - VPU_DecStartOneFrame(), который инициирует операцию декодирования изображения и возвращается, как только в VPU начинается декодирование изображения. Ход выполнения операции декодирования изображения отслеживается способом, описанным в пункте 4.3.4.3 «Выполнение операции декодирования изображения».

1.3.6.4.4.6.4.2 Обработка потока данных декодера

Для режима кольцевого буфера VPU предоставляет функцию API, позволяющую получить данные об указателе считывания, указателе записи и свободном пространстве при вызове одной

только функции VPU_DecGetBitstreamBuffer(). С помощью данной функции API приложение может получить информацию о доступном пространстве в буфере битового потока и передать в буфер битового потока определенный объем данных потока, который не должен превышать доступный объем буфера. При передаче данных потока следует проявлять осторожность в конце буфера битового потока во избежание непредвиденного прерывания данных. В ходе передачи данных потока в буфер битового потока указатель записи может достичь конечного адреса буфера битового потока. В этом случае система во избежание перывания данных должна зациклить указатель записи на начало буфера битового потока и продолжить загрузку.

Обновление указателя записи также легко выполнить с помощью функции API - VPU_DecUpdateBitstreamBuffer(). При получении данных об объеме загруженного потока внутри данной функции API выполняется зацикливание указателя записи и обновление указателя записи. Перед обновлением указателя записи хост-приложение должно завершить передачу данных потока в буфер битового потока. В противном случае может возникнуть несовпадение во времени доступа, что приведет к проблемам в работе декодера.

1.3.6.4.4.6.4.3 Выполнение операции декодирования изображения

Операция декодирования изображения занимает некоторое количество времени, и в ходе ожидания завершения выполнения этой операции приложение может перейти к выполнению других задач, например, обработке отображения выходных данных, декодированных до этого. Приложением могут использоваться две разные схемы распознавания завершения операции декодирования изображения: опрос регистра состояния и сигнализация прерывания. Если в приложении будет использоваться схема опроса, то нужно будет просто проверить регистр BusyFlag VPU. Вызов функции VPI_IsBusy()обеспечивает такие же результаты.

Сигнализация прерывания может быть наиболее эффективным способом проверки выполнения заданной команды. Сигнал прерывания для команды DEC_PIC_RUN отображается в bit[3] регистра включения прерывания. Поэтому благодаря такому специальному сигналу прерывания, передаваемому из VPU, система легко «поймет», что операция декодирования изображения выполнена.

1.3.6.4.4.6.4.4 Получение данных декодирования

Не важно, когда завершилось декодирование изображения, хост-приложение в любое время может получить закодированные выходные данные, такие как индекс отображаемого кадра, индекс декодированного кадра, тип изображения декодированного кадра, число макроблоков с замаскированными ошибками и т. д.

API предоставляет функцию VPU_DecGetOutputInfo() для получения выходных данных декодированного изображения. При использовании API приложение всегда должно пользоваться сразу парой функций - VPU_DecStartOneFrame() и VPU_DecGetOutputInfo(). Это означает то, что без вызова функции запроса результатов - VPU_DecGetOutputInfo() вызов функции VPU_DecStartOneFrame() не приведет к инициированию операции декодирования следующего изображения. Это условие очень полезно для защиты декодированных результатов от ошибочной перезаписи из другого потока данных в среде, в которой используется несколько экземпляров. Поэтому система должна рассматривать функцию VPU_DecGetOutputInfo() в качестве команды, освобождающей VPU от операции декодирования текущего изображения.

Считывание отображаемых выходных данных из декодера VPU

Индекс кадра отображения - indexFrameDisplay показывает, какой из кадров готов к отображению. Он всегда обозначает индекс буфера кадра, который отображается в данный момент. Он может отличаться от индекса декодированного изображения, если включена опция контроля упорядочения отображаемых данных, например, переупорядочение отображаемых данных H.264/AVC, B-изображение в VC-1 и т. д.

При инициализации последовательности SEQ_INIT выходные данные декодера не обязательно должны отображаться, даже если завершилось декодирование нескольких кадров, потому что существует определенный порядок вывода данных на экран. В случае переупорядочения H.264/AVC первые выходные данные могут появиться на экране, при самом неблагоприятном исходе, по окончании декодирования 17-го кадра. Иногда может не быть подходящего индекса буфера изображения. В этом случае VPU возвращает индекс буфера кадра с отрицательным значением. В зависимости от опции пропуска кадра это могут быть значения -3 или -2. Только по окончании инициализации последовательности данное значение будет равно -1. Поэтому когда система получает индекс буфера равный -1, можно завершить работу текущего экземпляра декодера без потерь качества изображения. Итак, хост-приложение может легко выяснить статус выходных отображаемых данных и определить окончание последовательности декодирования:

неотрицательное значение indexFrameDisplay;

Значение индекса выходных данных показывает индекс буфера кадра выходных отображаемых данных

indexFrameDisplay = -1;

Означает, что когда в VPU поступает сигнал об окончании потока, выходных отображаемых данных уже нет. Это означает окончание декодирования последовательности.

indexFrameDisplay = -2 or -3.

Означает то, что из-за включенной функции пропуска кадров никаких временно отображаемых выходных данных нет.

Считывание декодированных выходных данных из декодера VPU

Индекс закодированного кадра indexFrameDecoded – это дополнительные выходные данные, поступающие в хост-приложение. Данный индекс используется для указания номера буфера кадра, в котором хранится закодированное изображение. Для хост-приложения данный индекс, как правило, не имеет большого значения. Для обработки выходных данных декодера VPU вполне достаточно индекса отображения данных indexFrameDisplay.

При включении опции контроля упорядочения отображения данных для обеспечения переупорядочения отображения данных H.264/AVC, передачи B-кадра VC-1 и т. д. в конце декодирования последовательности, хост-приложению необходимо извлечь декодированные кадры, подлежащие отображению. В ходе операции по извлечению кадров никакие текущие операции по декодированию не выполняются. В такой ситуации бит indexFrameDecoded будет равен -2 (0xFFFE), что означает, что на данный момент поток для декодирования отсутствует. Такой отрицательный декодированный индекс используется также в случае, если из-за включенной опции пропуска изображения или ошибки в заголовке изображения в ходе декодирования пропускаются изображения.

Если для записи данных декодированных изображений нет достаточного количества буферов кадра, данное значение будет равно -1 (0xFFFF). В этом случае после сброса флага отображения по команде VPU_DecClrDispFlag()приложение повторно вызывает функцию VPU_DecStartOneFrame().

Ниже приводится краткие выводы относительно использования индекса indexFrameDecoded

Положительное значение индекса indexFrameDecoded

Значение индекса выходных данных указывает на индекс буфера кадра, в котором хранится закодированный кадр.

indexFrameDecoded == -1

Текущее изображение было декодировано, однако для его хранения нет доступного буфера кадра (вариант ошибки)

indexFrameDecoded == -2

Отсутствует поток изображений для декодирования, поэтому VPU не может декодировать никакое изображение.

Ниже приведена таблица взаимодействия между индексами indexFrameDisplay и indexFrameDecoded.

Таблица 1.559 – indexFrameDecoded и indexFrameDisplay

indexFrameDecoded	indexFrameDisplay	Описание
Любое значение	-2 или -3	В VPU было выполнено декодирование изображения, но отсутствует кадр для вывода на экран. Это обычная ситуация в начале операции декодирования потока с учетом существующего порядка отображения данных (задержки).
Любое значение	Любое значение	Декодирование идет в нормальном режиме
-2	Любое значение	Было декодировано последнее изображение (отсутствует поток для декодирования), и есть кадр для вывода на экран. Когда выдается команда пропуска кадра. Когда VPU в ходе декодирования заголовка изображения обнаруживает ошибку.

indexFrameDecoded	indexFrameDisplay	Описание
Любое значение	-1	Окончание последовательности (больше нет декодированного кадра и кадра для вывода на экран)
-1	Любое значение	Когда все буферы кадра заняты декодированным изображением; это означает, что блок VPU не может записать в буфер кадра декодированный кадр ввиду того, что больше буферов кадра нет. (вариант ошибки)

Если для подачи в буфер битового потока больше нет потока данных, хост-приложение должно установить в регистре BIT_BIT_STREAM_PARAM флаг StreamEnd, чтобы уведомить об этом VPU. Если в буфере битового потока обнаруживается пустой буфер, VPU выдает на хост-процессор сигнал прерывания, чтобы запросить дополнительный поток данных для декодирования. Однако если больше потока нет, возникает блокировки. Поэтому, чтобы избежать наступления подобной ситуации хост-приложение должно установить флаг StreamEnd. Если VPU обнаружит пустой буфер, а StreamEnd установлен в 1, то он возвращает в indexFrameDecoded значение -2.

Обрезка при отображении в H.264/AVC

Опция обрезки при отображении в H.264/AVC принуждает хост-приложение отобразить часть буферов кадров. Информация об окне обрезки предоставляется в SPS. В SPS представлено четыре значения сдвига прямоугольников обрезки, и считать эти значения хост-приложение может из соответствующих регистров. С помощью этих четырех значений сдвигов хост-приложение может легко определить положение окна целевых выходных данных. Если опция обрезки при отображении данных отключена, размер окна обрезки будет равен 0.

1.3.6.4.4.6.4.5 Управление буферами изображений

В блоке VPU есть несколько флагов, показывающих, отображается или не отображается буфер кадра. Флаги устанавливаются после того, как VPU автоматически возвращает индекс кадра изображения. VPU никогда не использует буфер, у которого установлен флаг отображения. Перед началом процесса декодирования VPU проверяет, есть ли свободный буфер кадра, и если для записи декодированного изображения нет свободного буфера, он немедленно возвращает текущий декодированный индекс со значением -1. (indexFrameDecoded == -1)

После отображения буферов кадра при вызове функции

VPU_DecClrDispFlag()хост-приложение может сбросить флаг, но только в случае, если блок VPU находится в состоянии бездействия (по завершении декодирования изображения).

Флаг VPU_DecClrDispFlag() должен использовать значение indexFrameDisplay. Если DecOpenParam::wtlEnable истинно, и флаг VPU_DecClrDispFlag() использует значение indexFrameDisplayForTiled, то хост-процессор может получить неверные результаты декодирования.

1.3.6.4.4.7 Закрытие экземпляра декодирования

Обработка конца потока и последнего изображения в буфере битового потока

После того как хост-приложение достигнет окончания потока и передаст все данные потока в буфер битового потока, следует установить флаг StreamEnd в регистре BIT_BIT_STREAM_PARAM. Данный флаг предназначен для защиты VPU от остановки в случае опустошения буфера потока. Вместо этого хост-приложение может выдать 0 во второй аргумент функции VPU_DecUpdateBitstreamBuffer(), чтобы проинформировать VPU об окончании потока.

По окончании передачи последнего байта потока в буфер битового потока хост-приложению следует установить данный флаг и продолжить вызывать функцию VPU_DecStartOneFrame(). После того, как придет последнее изображение выходных отображаемых данных, индекс декодированного изображения indexFrameDecoded изменится на -1. Когда хост-приложение получит данный индекс, можно будет понять, что обработки последовательности завершилась.

Даже, несмотря на то, что indexFrameDecoded равен -1, некоторые из кадров могут быть еще не выведены на экран. Хост-приложение должно буферизировать выходной декодированный кадр даже после завершения текущей операции декодирования. В этом случае хост-приложение должно вызывать функцию VPU_DecStartOneFrame(). До тех пор, пока задержанные выходные отображаемые кадры не будут полностью извлечены, декодер VPU будет передавать в хост-приложение индекс кадра только что выведенного на экран. Если доступных выходных данных больше нет, декодер VPU возвращает индекс отображения, равный -1 (indexFrameDisplay = -1).

Закрытие текущего экземпляра

После того как приложение закончит декодирование последнего изображения и будет готово закрыть экземпляр декодера, оно должно сбросить указатель этого экземпляра и уведомить VPU с помощью передачи в него команды SEQ_END о закрытии данного экземпляра. Это можно сделать, просто вызвав функцию VPU_DecClose().

1.3.6.4.4.8 Другие способы управления VPU

1.3.6.4.4.8.1 Работа с несколькими экземплярами

В приложениях многоканального кодека экземпляр является обработчиком, помогающим идентифицировать каждую задачу, которую выполняет определенный декодер или кодер, поэтому может показаться, что ряд задач выполняется одновременно, даже, несмотря на то, что блок VPU фактически обеспечивает поочередное выполнение этих задач в режиме с разделением времени.

Если хост управляет блоком VPU с использованием уровня API VPU, этот уровень API VPU может управлять переходом между экземплярами. Следовательно, хосту не нужно заботиться ни о чем, что связано с работой с несколькими экземплярами.

Однако в случае если хост управляет VPU с помощью регистров главного интерфейса, то при переключении экземпляров хост должен управлять некоторыми общими регистрами. Перед тем как хост передаст новую команду, ему нужно зарезервировать содержимое регистров главного интерфейса для выполненного перед этим экземпляра, а также восстановить содержимое регистров главного интерфейса для возобновления экземпляра.

Ниже приведены регистры, которые резервируются и восстанавливаются перед переключением экземпляров.

BIT_BIT_STREAM_CTRL
BIT_FRAME_MEM_CTRL
BIT_BIT_STREAM_PARAM
BIT_RD_PTR
BIT_WR_PTR
BIT_AXI_SRAM_USE
BIT_FRM_DIS_FLAG

Пример работы экземпляров

На рисунке 1.161 показано, как VPU обеспечивает обслуживание двух экземпляров в ходе их работы.

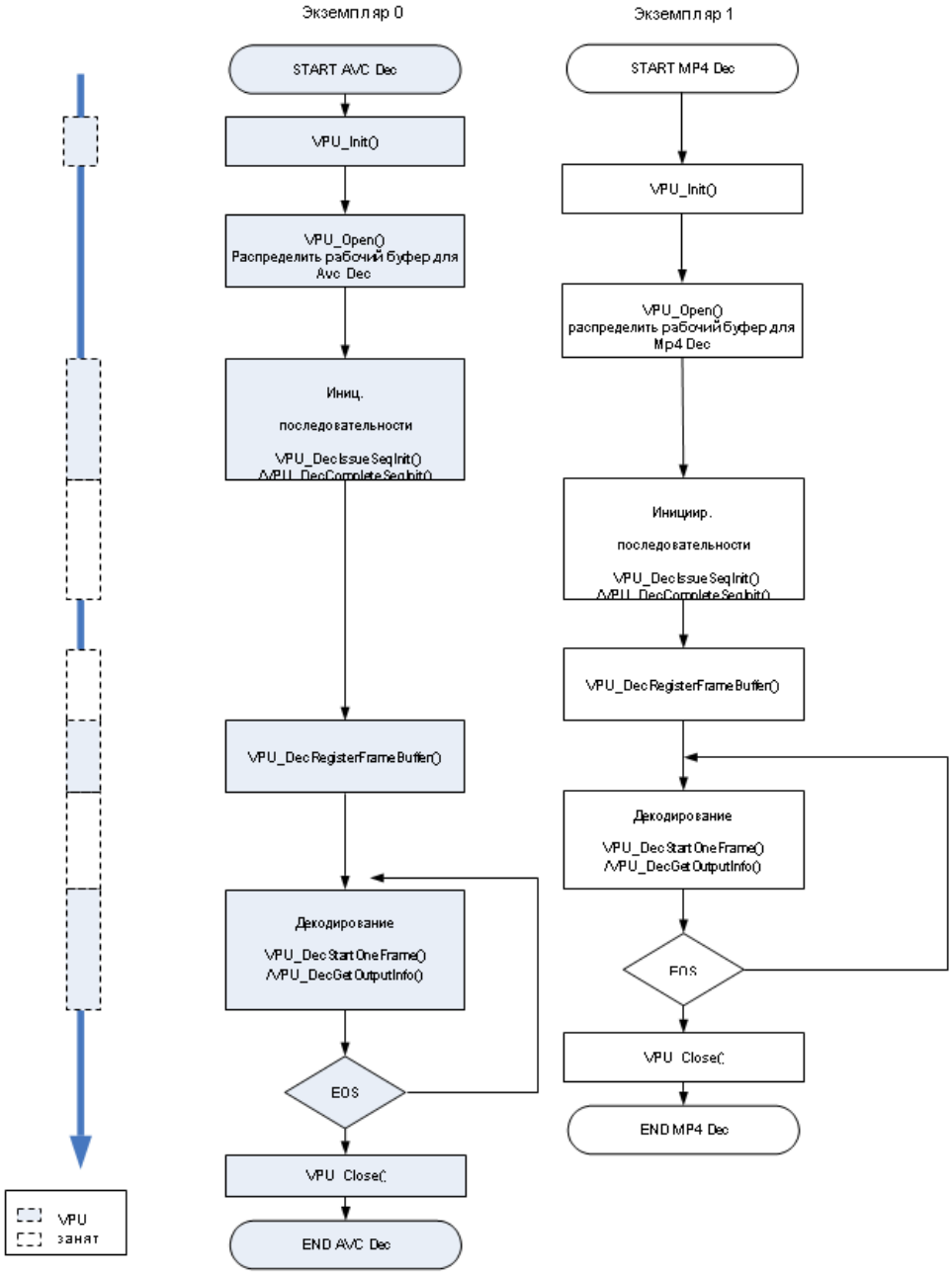


Рисунок 1.161 - Пример диаграммы работы с 2 экземплярами

Блок VPU работает с экземплярами по приведенной ниже схеме. Допустим, например, что экземпляр 0 - это AvcDec, а экземпляр 1 - это Mp4Dec.

В экземпляре 0 по команде VPU_Init() с хоста во внутреннюю программную память VPU загружается микро-ПО.

Когда выдается VPU_Init(), то для экземпляра 1 она не выполняется, потому что VPU уже инициировался в экземпляре 0.

VPU_DecOpen() позволяет назначить рабочий буфер для экземпляра 0. Информация об экземпляре 0 хранится в рабочем буфере.

Также как для экземпляра 0, VPU_DecOpen() позволяет назначить рабочий буфер для экземпляра 1.

Функция инициирования последовательности выполняется для экземпляра 0. При этом возвращается информация о размере изображения и количестве DPB.

Функция инициирования последовательности выполняется для экземпляра 1. При этом возвращается информация о размере изображения и количестве DPB.

Функция VPU_DecRegisterFrameBuffer() позволяет зарегистрировать буферы кадра для экземпляра 0 с информацией о буфере кадра и буфере mvCol, который назначается хостом.

Функция VPU_DecRegisterFrameBuffer() позволяет зарегистрировать буферы кадра для экземпляра 0 с информацией о буфере кадра и буфере mvCol, который назначается хостом.

Начинается декодирование кадра для экземпляра 0.

Начинается декодирование кадра для экземпляра 1.

Направленная вниз стрелка в левой части рисунка 4.3 “Пример диаграммы работы с 2 экземплярами” показывает состояние занятости блока VPU. Может показаться, что оба экземпляра должны использоваться одновременно. Однако на самом деле VPU, например, по команде SEQ_INIT или PIC_RUN, единовременно может работать только с одним экземпляром. Поэтому хост-приложению следует убедиться в том, что оно может вызывать функцию VPU тогда, когда VPU находится в состоянии занятости. Если вызов предыдущей функции завершен, то приложение может вызвать другую функцию.

Приложение В справочном ПО “Chips&Media”, устанавливаемом на устройства FPGA, для регистра максимального количества экземпляров выделено четыре экземпляра (см. код в vpuconfig.h, приведенный ниже). Пользователь может изменить это число, установив, какой объем памяти необходим каждому экземпляру (см. пункт 4.4.1.2 “Объем памяти для каждого экземпляра” .

// MAX_NUM_INSTANCE позволяет установить любое число, исходя из объема памяти системы.

// Приведенная ниже настройка предназначена специально для платы FPGA C&M.

1.3.6.4.4.8.1.1 Объем памяти для одного экземпляра

Для работы одного экземпляра требуется некоторый объем памяти. Величину этого объема можно определить исходя из приведенного ниже описания.

Рабочий буфер. Каждому экземпляру для хранения контекстной информации требуется собственный рабочий буфер. Объем рабочего буфера равен 80*1024. Кроме того, декодеру H.264 для хранения параметров ParameterSet требуется рабочий буфер большего объема (320*1024).

Буфер битового потока. Хост должен определять и задавать объем данного буфера исходя из битовой скорости и разрешения

Буферы кадров. Хост должен сам определить и задать этот объем.

MvColBuffer. Для разных вариантов разрешений и используемых кодеков объем может быть разным. Более подробную информацию можно найти в описании функции VPU_GetMvColBufSize().

И т. д.

Базовый H.264. Буфер слайсов для ASO/FMO

SLICE_SAVE_SIZE = (MAX_DEC_PIC_WIDTH*MAX_DEC_PIC_HEIGHT*3/4)

VP8. (Буфер хранения макроблоков для размещения информации макроблоков + векторов движения с разделением)*4*MbNumbyte

VP8_MB_SAVE_SIZE = (17*4*(MAX_DEC_PIC_WIDTH*MAX_DEC_PIC_HEIGHT/256))

Mp4 Enc. Вспомогательный буфер для хранения раздела данных

SIZE_MP4ENC_DATA_PARTITION = (MAX_ENC_PIC_WIDTH*MAX_ENC_PIC_HEIGHT*3/4)

1.3.6.4.4.8.2 Изменение последовательности

В последовательности, такой как SPS и PPS, содержится информация о разрешении изображения, количестве ссылочных кадров, профиле и уровне, которая необходима для инициализации декодера и его начальной настройки.

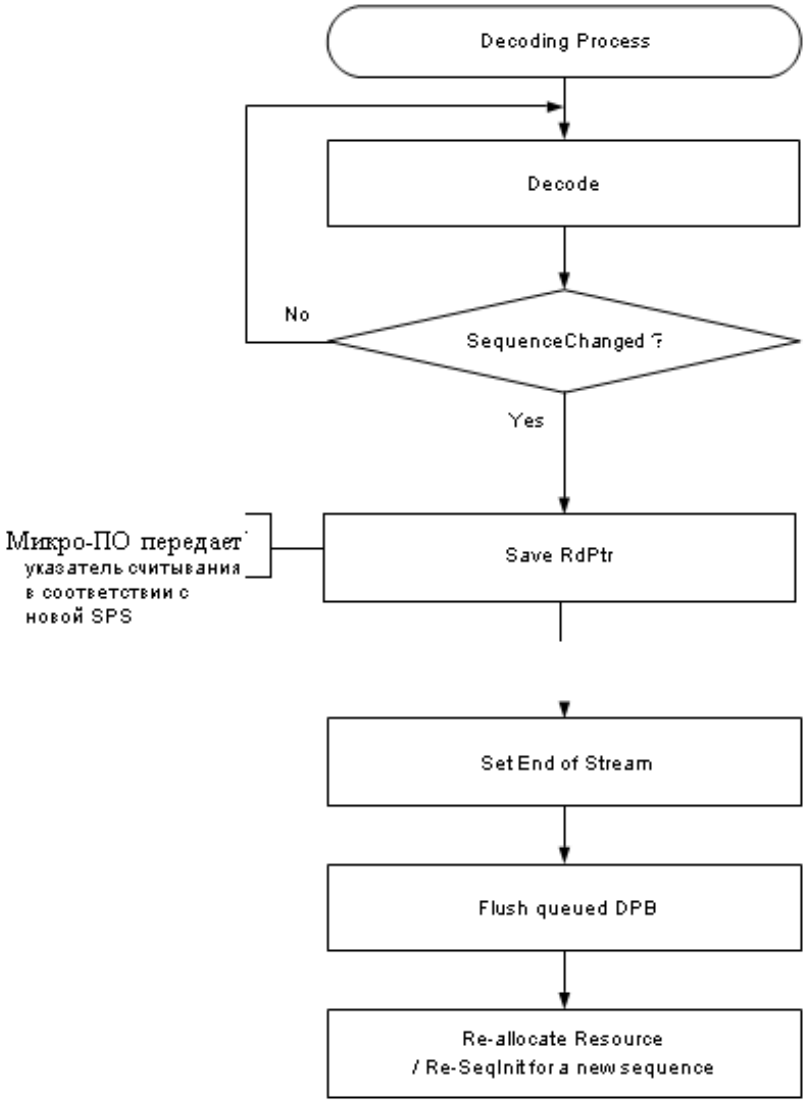
Иногда в VPU в нескольких потоках поступает несколько последовательностей SPS и PPS. Например, VPU в одном и том же потоке может получить информацию SPS и PPS с разными вариантами разрешений. В этом случае блок VPU должен перераспределить объем буфера кадра, используемого для декодирования нескольких последующих кадров, потому что, если размер декодированного кадра, поступившего на хранение, будет больше размера предыдущего кадра, то буфер будет переполнен.

Таким образом, хост должен учитывать возможность возникновения такой ситуации в условиях использования нескольких экземпляров и должен знать, как работать с ресурсами в течение всего жизненного цикла декодера. Кроме того, в контейнере файлов редко случается изменение файлов, такая возможность поддерживается только для форматов AVC и MPEG2.

Примечание Если профиль изменился на другой, но с таким же разрешением и DPB, последовательность меняться не будет.

1.3.6.4.4.8.3 Диаграмма работы при условии изменения последовательности

На рисунке 1.162 представлена диаграмма работы при изменении последовательности.



Decoding process – Процесс декодирования
Decode – Декодировать
No – Нет
Sequence Changed – Последовательность изменилась
Yes – Да
Save RdPtr – Сохранить RdPtr
Set End of Stream – Задать конец потока
Flush queued DPB – Сброс в DPB в порядке очереди
Re-allocate Resource / Re-SeqInit for a new sequence – Перераспределение ресурсов / Повторное инициирование последовательности для новой последовательности

Рисунок 1.162 – Функциональная диаграмма при изменении последовательности

В ходе декодирования функция VPU_DecGetOutputInfo() выдает 1 на флаге sequenceChanged для новой последовательности.

При обнаружении этого хост вынужден сохранять текущий указатель считывания, который укажет местоположение новой последовательности.

Задание конца последовательности с помощью представленных ниже функций API.

VPU_DecSetRdPtr(handle, seqChangedRdPtr, 1);

// указатель записи будет обновлен до нового заголовка последовательности

VPU_DecUpdateBitstreamBuffer(handle, 1);

// уведомление микро-ПО о состоянии конца потока в буфере битового потока

// принудительное уведомление о том, что буфер битового потока опустошается.

VPU_DecUpdateBitstreamBuffer(handle, STREAM_END_SET_FLAG);

// установление состояния конца потока, чтобы аннулировать буфер кадра с задержкой.

Поочередный сброс в DPB для вывода на экран. Вызывать функции VPU_DecStartOneFrame() и VPU_DecGetOutputInfo () пока outputInfo.indexFrameDisplay не станет равно -1.

Повторное инициирование последовательности с новой последовательностью.

1.3.6.4.4.8.3.1 Обнаружение изменения последовательности

Чтобы обнаружить изменение разрешения, блок VPU проводит проверку следующих элементов.

Таблица 1.560 – Проверка изменения последовательности

ЭЛЕМЕНТ	AvcDec	Mp2Dec
Размер изображения	pic_width_in_mbs_minus1, pic_height_in_map_units_minus1	horizontal_size_value, vertical_size_value
Количество кадров	DpbMaxDecBuffering	-

1.3.6.4.4.8.4 Модуль постобработки

После декодирования изображение с помощью модуля PPU (блока постобработки) в VPU подвергается дальнейшей обработке, например, поворачивается или отражается. PPU функционирует с канале декодирования, в который предыдущий кадр буфера отображения кадров приходит как исходный, и записывает его обратно в буфер кадра, назначенный ХОСТОМ.

Процесс работы PPU

PPU считывает буфер исходного кадра на основе макроблоков, выполняет поворот, отражение или преобразование Tiled2Linear, а также записывает данные в буфер-приемник кадра. Во время работы PPU эффективность декодирования может снизиться, поскольку кадры повторно записываются только после того, как будут обработаны, а это означает, что расход полосы пропускания будет больше.

Другой буфер кадра для PPU

Если ХОСТ желает воспользоваться одной из функций PPU, следует разместить другой буфер(ы) кадра, чтобы PPU мог записывать свои выходные данные в буфер-приемник кадра. Для обеспечения плавного отображения рекомендуется использовать схему двойной буферизации посредством назначения нескольких буферов кадра.

Индекс буфера кадра PPU

Работа PPU неизбежно приводит к тому, что один кадр будет задерживаться, поскольку PPU функционирует автономно, отдельно от процесса декодирования. По этой причине хост не должен моментально отключаться, когда по окончании потока и отображения перестают приходить кадры (индекс отображения = -1), пока PPU не обработает последний выходной кадр.

На рисунке 1.163 показан процесс того, как индекс буфера кадра для PPU goes along with decoded frame buffer index and display frame buffer index.

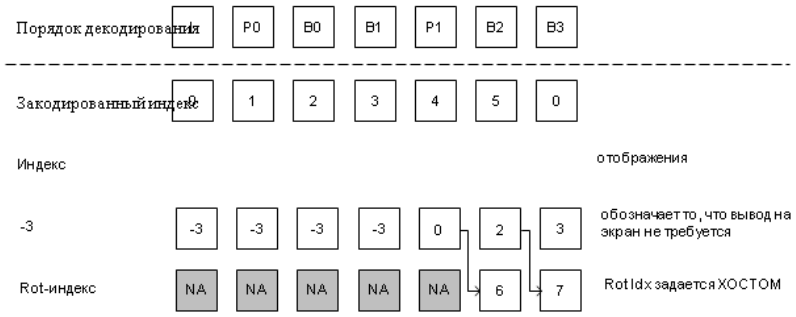
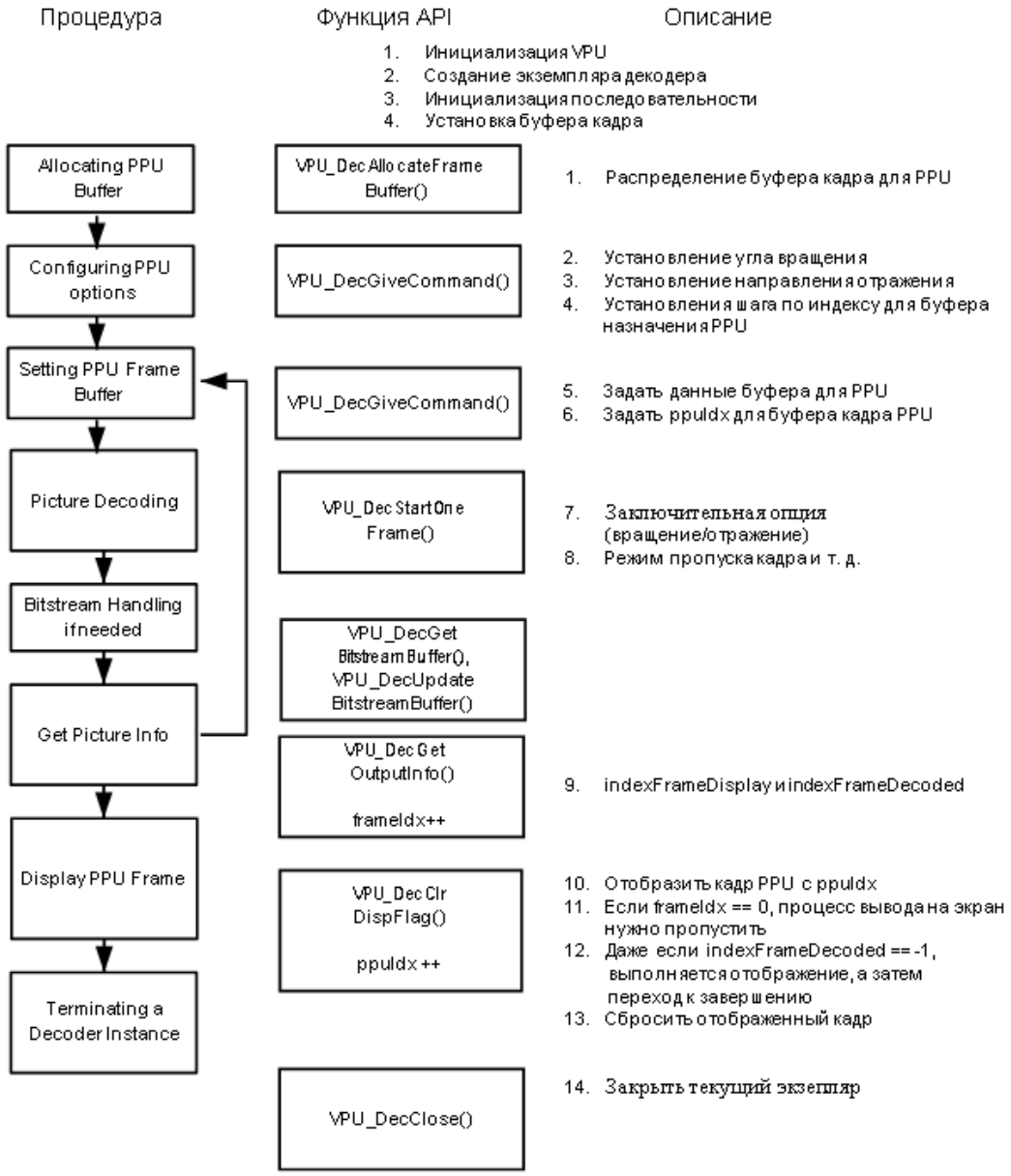


Рисунок 1.163 – Индекс буфера кадра PPU

Обработка PPU с помощью API

На рисунке 1.164 - “Процесс управления PPU с помощью API” представлена краткая схема работы модуля PPU. Здесь показаны некоторые этапы и связанные с ними функции API, предназначенные для управления PPU, с кратким описанием.



Allocating PPU buffer – Распределение буфера PPU
Configuring PPU options – Конфигурирование опций PPU
Setting PPU frame buffer – Установление буфера кадра PPU
Picture Decoding – Декодирование изображения
Bitstream handling if needed – Обработка битового потока (если необходимо)
Get Picture Info – Получение информации об изображении
Display PPU frame – Отображение кадра PPU
Terminating a Decoder Instance – Закрытие экземпляра декодера

Рисунок 1.164– Процесс управления PPU с помощью API

Распределение буфера PPU

Вызов функции VPU_DecAllocateFrameBuffer() для распределения буфера-приемника кадра для PPU. Хост не должен вызывать данную функцию, если буфер-приемник кадра уже выбран. Данная функция является необязательной.

Конфигурирование опции PPU

VPU_DecGiveCommand(handle, SET_ROTATION_ANGLE, &rotAngle);

Данная функция задает значение угла поворота.

0: без вращения
90: поворот на 90 градусов
180: поворот на 180 градусов
270: поворот на 270 градусов
VPU_DecGiveCommand(handle, SET_MIRROR_DIRECTION, &mirDir);
Данная функция задает направление отражения.
0: без отражения
1: по вертикали
2: по вертикали
3: по вертикали и по горизонтали
VPU_DecGiveCommand(handle, SET_ROTATOR_STRIDE, &rotStride); Данная функция задает шаг по индексу буфера кадра назначения для PPU. VPU_DecGiveCommand (handle, ENABLE_ROTATION, 0);

Данная функция включает вращение. Она позволяет вносить изменения в каждый кадр.

VPU_DecGiveCommand(handle, ENABLE_MIRRORING, 0); Данная функция включает функцию отражения. Она позволяет вносить изменения в каждый кадр. VPU_DecGiveCommand(handle, ENABLE_DERING, 0);

Данная функция включает функцию шумоподавления. Она позволяет вносить изменения в каждый кадр и действует только для MPEG-2/4.

Настройка буфера кадра PPU

VPU_DecGiveCommand(handle, SET_ROTATOR_OUTPUT, &fbPPU[ppIdx]);

Данная функция конфигурирует информацию об адресе буфера-приемника кадра для PPU. Она также позволяет вносить изменения в каждый кадр. Когда XOCT получает от драйвера видеоадаптера адрес буфера кадра, то этот адрес может быть передан данной функции.

1.3.6.4.5 МЕТОДИКА ОБРАБОТКИ ОШИБОК

1.3.6.4.5.1 Введение

Иногда видеоприложения сталкиваются с ошибками битового потока. Существует несколько источников, способных вызвать нарушение в передаче битового потока в видеоприложениях. Сильные искажения в канале передачи, особенно в каналах видеосвязи, уязвимых для ошибок, таких как каналы беспроводной сети, могут вызвать серьезные нарушения в передаче битового потока.

При рассмотрении вопроса об обработке ошибок в видеоприложениях следует с большой осторожностью относиться к использованию приведенных ниже терминов.

Обнаружение ошибок

Операция обнаружения ошибок служит для поиска ошибок, включенных в получаемые данные потока. Существует несколько разных способов обнаружения ошибок в протоколе транспортного уровня системы. При обнаружении ошибок в ходе передачи пакета битового потока, эти ошибки частично или полностью обрабатываются на транспортном уровне системы. В противном случае все обнаруженные ошибки будут обрабатываться самим видеодекодером.

Восстановление после ошибок

Операция восстановления после ошибок служит для нейтрализации ошибок в пораженном битовом потоке с использованием некоторых корректирующих инструментов, таких как коды с исправлением ошибок. Но чаще всего операция устранения ошибок полностью обрабатывается на транспортном уровне системы. Поэтому обычно данная операция не видеодекодером не выполняется.

Обеспечение устойчивости к ошибкам

Операция обеспечения устойчивости к ошибкам служит для обеспечения видеодекодера некоторыми специальными возможностями по достижению некоторого уровня устойчивости к ошибкам. С помощью инструментов по выработке устойчивости к ошибкам видеодекодер может с большей эффективностью выполнять операции обнаружения и маскирования ошибок.

Такие операции как

- локализация распространения ошибок,
- повторение определенной ключевой информации, и
- чередование данных

являются хорошими примерами операций, направленных на обеспечение устойчивости к ошибкам. Эти инструменты позволяют видеодекодеру получить некоторый объем битового потока изображения, не содержащего ошибки. Используя битовый поток, не содержащий ошибок, видеодекодер может легко свести к минимуму последствия ошибок.

Маскирование ошибок

Операция маскирования ошибок служит для того, чтобы минимизировать видимые последствия ошибок, возникающих в видеодекодере. Отбрасывание и сокрытие кадров с ошибками – это самый простой способ скрыть видимые последствия ошибок. Но на практике это может привести к пропуску в отображении нескольких кадров ввиду характера прогнозирующего кодирования при сжатии видеоданных. А при условии частых ошибок в канале это вызовет ухудшение качество видеосвязи, «замораживанию» изображения. Поэтому после минимизирования всех возможных визуальных артефактов важно показать выходные декодированные данные, содержащие ошибки.

В нижеследующих подразделах подробно описываются три главных вопроса, связанных с обработкой ошибок, возникающих при декодировании видеоданных.

1.3.6.4.5.2 Обнаружение ошибок

Для повышения эффективности кодирования в большинстве стандартов видеокодирования используются в основном инструменты VLC (кодирование с переменной длиной). Даже, несмотря на то, что механизм VLC очень важен для достижения более высокого коэффициента

сжатия, он приводит к распространению ошибок при их возникновении. Если в кодовой таблице VLC определяется пораженная последовательность, то символ, кодируемый с помощью VLC и имеющий ошибку, может быть декодирован с неверным значением. Еще хуже, что такое неверное интерпретирование повторяется до тех пор, пока неустановленная последовательность не будет выявлена на этапе декодирования. Поэтому когда в смежной последовательности кодовых данных VLC происходит ошибка, в целях безопасности следует считать всю последовательность последовательностью, содержащую ошибки.

Однако на практике вероятность неверного интерпретирования нескольких символов значительно ниже вероятности обнаружения ошибки. И отбрасывание всей последовательности закодированных данных VLD в некоторой степени нецелесообразно. Поэтому когда в ходе обработки макроблоков случается ошибка VLD, можно допустить, что во всех предыдущих макроблоках ошибок не было.

Выполняя проверку достоверности каждого значения, видео декодер может обнаруживать как ошибки VLD, так и ошибки в битовом потоке, особенно в синтаксисах заголовков. Пользуясь предварительно заданным значением - битом маркера, декодер также может обнаруживать ошибки в битовом потоке.

1.3.6.4.5.3 Устойчивость к ошибкам

Существует несколько инструментов обеспечения устойчивости к ошибкам, включенных в стандарты видеокодирования, таких как слайс (маркер ресинхронизации), разделение данных и RVLC для форматов MPEG-4, FMO, ASO, а также RS для AVC. Данные инструменты устранения ошибок обеспечивают возможность снижения степени поражения потока, но иногда результат их использования не слишком очевиден, и поэтому на практике более широко используется лишь ограниченный набор инструментов обеспечения устойчивости к ошибкам.

Особенно эффективным инструментом для предупреждения распространения ошибок является слайсовая структура. В середину данных VLD кодер может ввести предопределенную последовательность маркеров, которые показывают границы набора макроблоков (слайса). В этом случае, если при декодировании VLC случается ошибка, она распространится только в пределах этого слайса. Поэтому эффективно локализовать поражение можно будет только в пределах области слайса.

FMO и ASO – варианты расширений слайса, которые могут пригодиться в случае распространения пораженных фрагментов на изображение. Но эти схемы могут вызвать увеличение затрат вычислительных ресурсов, а также ухудшить эффективность кодирования. Использование инструментов устранения ошибок MPEG-4, такие как разделение данных и RVLC, повышает возможность маскирования ошибок. Но при этом требуются большие затраты вычислительных возможностей, а также падает эффективность кодирования. Поэтому данные инструменты не пользуются популярностью в видеоприложениях.

1.3.6.4.5.4 Маскирование ошибок

Маскирование ошибок – это одна из дополнительных схем, которая может использоваться для сокрытия последствий ошибок. Декодер, как правило, не может скрыть последствия ошибок в синтаксисах заголовков. Но ошибки, возникающие в ходе декодирования макроблоков, могут быть частично замаскированы при прогнозировании макроблоков, пораженные ошибками. Существует большое количество материалов, посвященных методам прогнозирования таких макроблоков, пораженных ошибками, то есть, методам прогнозирования пикселей в Intra-изображении, прогнозирования inter-изображения на основе информации вектора движения.

По окончании прогнозирования макроблоков, имеющих ошибки, декодер заменяет их спрогнозированными макроблоками. Если движение внутри изображения не слишком заметное, эффективность маскирования таких ошибок впечатляет. Но если в изображении заметен полукадр движения, эффективность маскирования ошибок относительно низкая. Следовательно, в некоторых декодерах схемы маскирования ошибок даже не используются, а выполняется

кадры, пораженные ошибками, отбрасываются. На практике схему маскирования ошибок полезно использовать, когда частота появления ошибок довольно велика. Если декодер аннулирует с экрана все изображения, пораженные ошибками, на экране никаких обновлений не будет.

1.3.6.4.5.5 Инструменты обработки ошибок VPU

1.3.6.4.5.5.1 Схема обнаружения ошибок

Обнаружение ошибок в заголовке

Выполняя проверку достоверности каждого значения, видеодекодер может обнаруживать как ошибки VLD, так и ошибки в синтаксисах заголовков. Пользуясь предварительно заданным диапазоном значений для каждого значения, битами маркера, кодами начала, декодер также может обнаруживать ошибки в битовом потоке. Если ошибка произошла в синтаксисах заголовков, то обработчику событий декодера требуются разные схемы обработки, в зависимости от уровня синтаксисов заголовков и имеющихся опций.

Заголовок уровня последовательности

Если ошибка возникает в синтаксисах заголовков уровня последовательности (VOL в MPEG-4, SPS в AVC, SEQ_META в VC-1 и т. д.), то декодер не будет обрабатывать последующие потоки изображений. У декодера VPU в этом случае есть две опции, и хосту необходимо решить какую из них следует использовать.

Декодер VPU может продолжить попытки найти синтаксисы заголовков уровня следующей последовательности, не переставая вызывать функцию. Данная опция может пригодиться приложениям цифрового видеовещания и приложениям видеосвязи.

Декодер VPU может остановить обработку. Если 0-ой бит регистра BIT_BIT_STREAM_PARAM (0x114) - SeqInitEscape активирован, то блок VPU может выйти из процесса инициализации с REC_DEC_SEQ_SUCCESS==0. Данная опция может быть пригодиться для локальных медиаплейеров, у которых синтаксисы верхнего уровня располагаются только в начале потока.

Заголовок нижнего уровня

Если ошибка возникает в синтаксисах заголовков нижнего уровня, таких как синтаксисы заголовков уровня изображения, заголовки слайса и т. д., декодер проигнорирует текущий поток изображений и продолжит декодировать следующий кадр. У декодера VPU в этом случае есть две опции, и хосту необходимо решить какую из них следует использовать.

Декодер VPU может продолжить попытки найти заголовок следующего изображения, не переставая вызывать функцию. Данная опция может использоваться либо в режиме воспроизведения файлов, либо в режиме кольцевого буфера. В случае VC-1 данный тип проверки границ изображения не возможен, потому что в поток MP VC-1 отсутствуют стартовые коды начала изображения. Управление границами потока осуществляется на стороне хост-приложения с анализатором файлов.

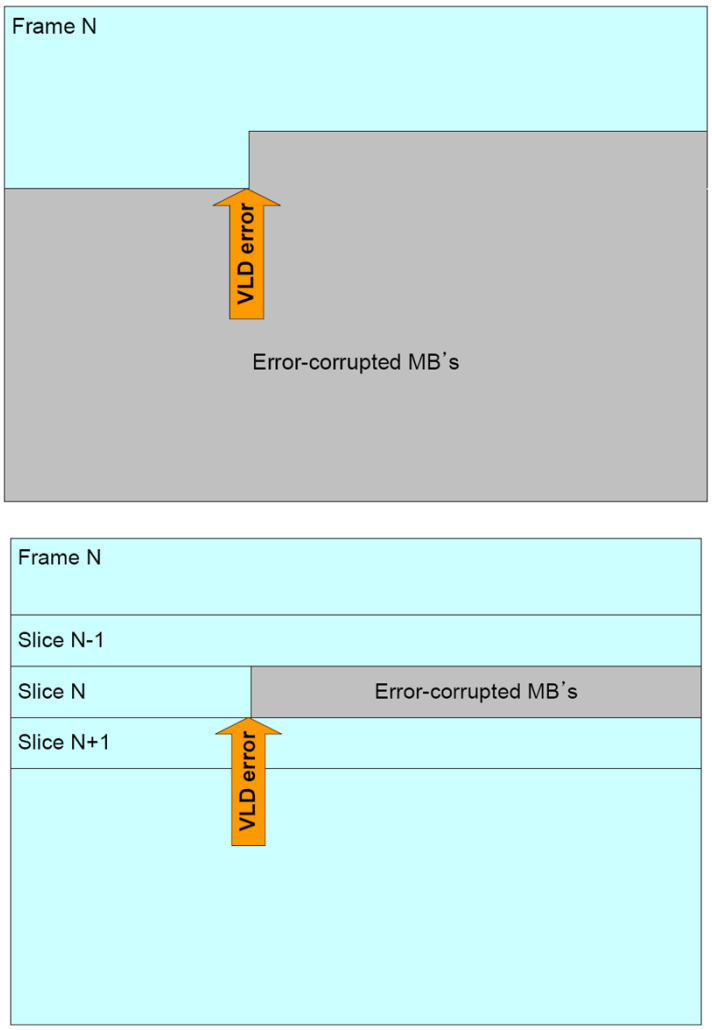
Декодер VPU может сразу же возвратить код ошибки. Данная опция используется в режиме воспроизведения файлов.

Примечание VPU выполняет процедуру обнаружения ошибок в заголовке с помощью микро-ПО, но текущая версия микро-ПО еще не поддерживает регистр главного интерфейса, в котором хранятся команды такого типа, передаваемые от хоста. Планируется усовершенствовать эту опцию.

Обнаружение ошибок VLD

Если ошибка VLD (кодирование с переменной длиной) возникает в ходе обработки микроблоков, блок VPU считает, что все предыдущие макроблоки не имеют ошибок. Только те макроблоки будут считаться макроблоками, пораженными ошибками, которые идут до следующего маркера синхронизации. VPU уведомляет об успешном проведении операции (REC_DEC_SEQ_SUCCESS==1) с указанием в регистре RET_DEC_PIC_ERR_MB (0x1C8) количества макроблоков с ошибками, а затем выполняет маскирование ошибок.

Примечание Коды ошибок, описывающие причины возникновения ошибок, еще не выявлены, но, впоследствии, будут реализованы.



Frame N – Кадр N
VLD Error – Ошибка VLD
Error-Corrupted MBs – Макроблоки, пораженные ошибками
Frame N – Кадр N
Slice N-1 – Слайс N-1
Slice N – Слайс N
Slice N+1 – Слайс N+1
VLD Error – Ошибка VLD
Error-Corrupted MBs – Макроблоки, пораженные ошибками

Рисунок 1.165 – Обнаружение ошибок в декодере VPU

1.3.6.4.5.5.2 Поддерживаемые инструменты обеспечения устойчивости к ошибкам

В декодере VPU все битовые потоки, основанные на инструментах обеспечения устойчивости к ошибкам, в зависимости от стандартов можно подвергнуть разбору. Поэтому не важно, какие используются опции устойчивости к ошибкам, декодер может успешно декодировать все битовые потоки. Например, могут быть успешно декодированы потоки формата MPEG-4 с маркером ресинхронизации, разделением данных и RVLC. Также успешно декодируются потоки AVC со слайсом, FMO, ASO и RS.

В действительности в режиме маскирования ошибок в VPU используется только ограниченное число инструментов обеспечения устойчивости к ошибкам. Это происходит потому, что для некоторых операций маскирования ошибок требуются большие затраты с точки

зрения объема программной памяти рабочей частоты. Принимая во внимание затраты, в особенности затраты программной памяти, в схеме маскирования ошибок в VPU используются следующие инструменты устранения ошибок.

MPEG-4

Слайс (включая маркер ресинхронизации)

Данная схема - это главный инструмент устранения ошибок, который позволяет предотвратить распространение ошибок в потоке изображений, пораженных ошибками. В схеме устранения ошибок в VPU используются все виды инструментов предотвращения распространения ошибок.

Разделение данных

Данная схема также основана на слайсовой структуре. Кроме того, в ней используется чередование между параметрами макроблока и коэффициентами трансформирования. Поскольку коэффициенты трансформирования имеют больший размер, это помогает локализовывать последствия ошибок для коэффициентов трансформирования.

Операция разделения данных помогает защитить некоторые важные данные, такие, например, как intra-DC, векторы движения, типы макроблоков, и т. д. посредством отделения параметров макроблоков от коэффициентов трансформирования. Потоки, содержащие структуры с разделением данных, могут быть успешно разобраны и декодированы в декодере VPU.

RVLC (реверсивное кодирование с переменной длиной)

Данная схема представляет собой дополнительную разновидность схемы с разделением данных. Благодаря использованию этой специальной преобразованной схемы декодер имеет больше возможностей для восстановления коэффициентов трансформирования, имеющих ошибки. Потоки, содержащие данные RVLC, могут быть успешно разобраны и декодированы в декодере VPU.

F.5.2.2. H.264

FMO (гибкий порядок макроблоков)

Функция FMO H.264 – это разновидность функции пространственного чередования между соседними макроблоками. Поэтому, в зависимости от типа карты распределения группы слайсов, ошибка в макроблоке не будет распространяться на другие соседние макроблоки. В случае возникновения ошибки макроблоки, пораженные ошибками, будут распространяться на кадр.

Декодер VPU поддерживает потоки H.264 с функцией FMO. Схемой маскирования ошибок в VPU также поддерживается возможность локализации ошибок на основе FMO, потому что, в сущности, она подобна схеме маскирования ошибок на основе слайсов. Для поддержания функции FMO требуется проведение предварительного сканирования всех данных потока, участвующих в построении потока изображений, а для этого, как правило, необходима более высокая рабочая частота. Увеличение объема микро-ПО для маскирования ошибок ничтожно мало.

ASO (Произвольный порядок слайсов)

Функция ASO H.264 – это разновидность функции переупорядочения между данными потока слайса. В случае возникновения пакетной ошибки некоторые из слайсов, соседствующих с этим пакетом, также могут быть поражены. С другой стороны, когда пакетная ошибка происходит в потоке ASO, область поражения может распространиться на слайсы, не являющиеся соседними.

Декодер VPU полностью поддерживает потоки H.264 с функцией ASO. Схемой маскирования ошибок VPU также поддерживается возможность локализации ошибок на основе ASO, потому что, в сущности, она подобна схеме маскирования ошибок на основе слайсов. Для поддержания функции ASO требуется проведение предварительного сканирования всех данных потока, участвующих в построении потока изображений, а для этого, как правило, необходима более высокая рабочая частота. Увеличение объема микро-ПО для маскирования ошибок ничтожно мало.

Избыточный слайс

Избыточный слайс H.264 – это разновидность инструментов, используемых для восстановления после ошибок. При многократной передаче данных слайса достигается определенный уровень возможности восстановления после ошибок. В зависимости от частоты избыточного слайса и объема данных слайса, эффективность кодирования может резко меняться.

Декодер VPU допускает использование в потоках H.264 избыточного слайса, но не декодирует, а игнорирует его. Схемой маскирования ошибок в VPU не поддерживается функция восстановления после ошибок, основывающаяся на избыточном слайсе, потому что в реальных приложениях она не всегда используется ввиду вызываемого при этом снижения эффективности кодирования. Для поддержания данной функции требуется проведение предварительного РАЗБОРА всех данных потока, используемых для построения потока изображений, что может потребовать слишком больших затрат вычислительных ресурсов и высокой рабочей частоты. Увеличение объема микро-ПО для маскирования ошибок также ощутимо.

1.3.6.4.5.5.3 Схема маскирования ошибок

1.3.6.4.5.5.3.1 Вложенная схема маскирования ошибок

Схема маскирования ошибок в VPU основана на принципе пропуска макроблоков и является одной из самых простых и, вместе с тем, самых эффективных схем. Функция пропуска макроблоков в зависимости от стандартов видеокодека имеет разное значение, но главный принцип работы в целом один и тот же, независимо от стандарта видеокодека.

По статистике пропускаемый макроблок – это наиболее вероятный тип макроблока. Поэтому когда происходит ошибка, и декодер не может восстановить определенный макроблок, этот макроблок предсказывается по наиболее вероятному макроблоку ссылочного кадра.

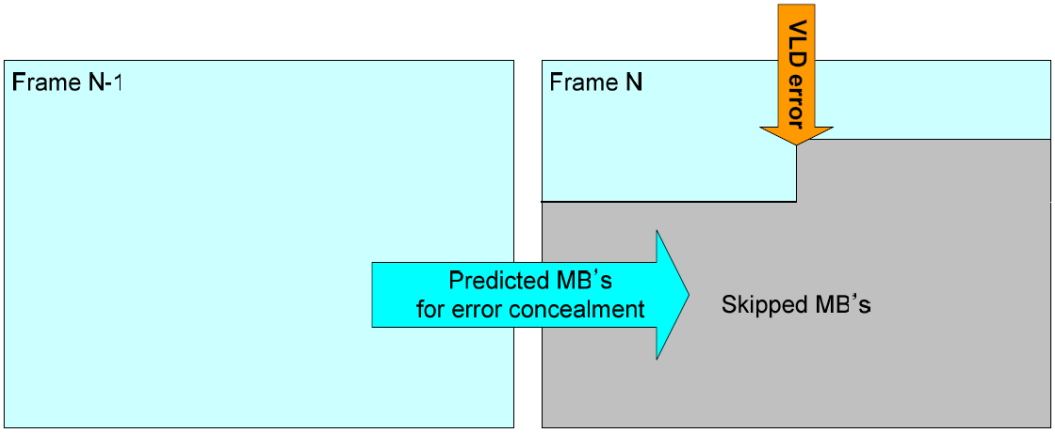
Если в случае ошибок бит применяется схема маскирования ошибок, то возможны два решения:

Заменить все макроблоки в разделе данных VLC, пораженном ошибками (слайс или кадр)

Заменить все макроблоки начиная с позиции, на которой была обнаружена ошибка.

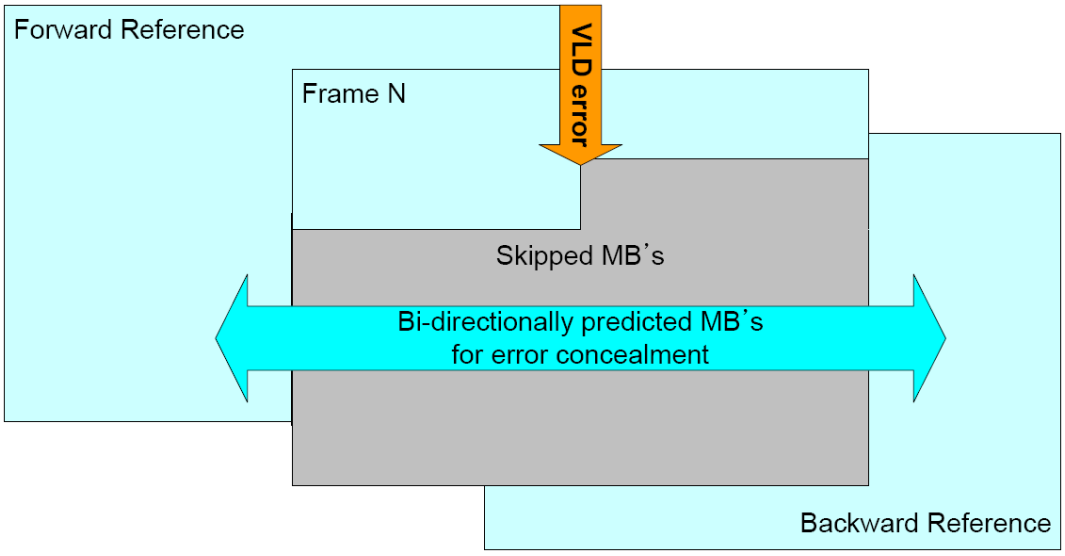
В схеме маскирования ошибок в VPU для всех видеодекодеров следует выбирать второй метод. Поэтому заменяться пропущенными макроблоками будут только те макроблоки, которые на этапе обнаружения ошибок классифицировались как макроблоки, пораженные ошибками. То есть, если блоком VLD во время разбора потока входных данных определенного макроблока обнаруживается ошибка, то все макроблоки, начиная с данного и заканчивая последним макроблоком, вплоть до следующего указателя ресинхронизации (маркер ресинхронизации, границы слайса, стартовый код и т. д.), будут заменяться пропущенными макроблоками.

Все эти схемы маскирования ошибок представлены на рисунке 1.166, 1.167 и 1.168.



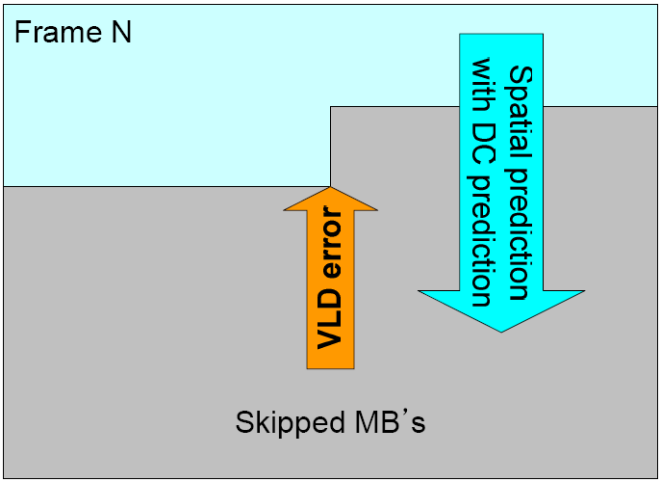
Frame N-1 – Кадр N-1
Frame N – Кадр N
VLD error – Ошибка VLD
Predicted MB's for error concealment – Спрогнозированные макроблоки, используемые для маскирования ошибок
Skipped MB's – Пропущенные макроблоки

Рисунок 1.166 – Маскирование ошибок P-кадров в декодере VPU



Forward Reference – Прямая ссылка
Frame N – Кадр N
VLD error – Ошибка VLD
Skipped MB's – Пропущенные макроблоки
Bi-directionally predicted MB's for error concealment – Спрогнозированные в обоих направлениях макроблоки, используемые для маскирования ошибок
Backward Reference – Обратная ссылка

Рисунок 1.167 – Маскирование ошибок B-кадров в декодере VPU



Frame N – Кадр N
VLD error – Ошибка VLD
Spatial prediction with DC prediction – Пространственное прогнозирование с предсказанием DC
Skipped MB's – Пропущенные макроблоки

Рисунок 1.168 – Маскирование ошибок I-кадров в декодере VPU

В случае P-кадров пропущенный макроблок копируется из кадра из прямой ссылки с вектором движения 0 (MPEG-2, MPEG-4, H.263) или с разницей векторов движения 0 (H.264, VC-1).

В случае B-кадров пропущенный макроблок копируется как из прямой, так и из обратной ссылок. Две эти копии чередуются для воспроизведения финальных макроблоков устройства прогнозирования.

В случае I-кадра для некоторых видеостандартов, поддерживающих возможность прогнозирования коэффициентов AC-DC, вместо временного прогнозирования может использоваться пространственное прогнозирование. Макроблоки с функцией прогнозирования коэффициента DC используются в качестве устройств предсказания макроблоков, пораженных ошибками. Если функция прогнозирования коэффициентов AC/DC не поддерживается, то будет проводиться копирование совмещенных макроблоков из ссылочного кадра.

Операция кодирования в нормальном режиме возобновляется с первого после следующего указателя ресинхронизации макроблока. Поэтому чем больше слайсов или пакетов данных имеют маркер ресинхронизации, тем значительно эффективнее будет осуществляться локализация ошибок. То есть, лишь небольшая часть изображения будет иметь замаскированные ошибки. Большая его часть будет успешно восстановлена. Однако это может потребовать большого расхода битов, а большое количество слайсов или пакетов данных вызовет ухудшение качества работы кодера. При подборе подходящего значения объема слайсов для каждого изображения приложению следует учитывать частоту появления ошибок, размер изображения, характеристики целевого приложения и т. д.

1.3.6.4.5.5.3.2 Поддержка внешней схемы маскирования ошибок

В зависимости от окружения данного приложения может требоваться более специфичная схема маскирования ошибок. Чтобы обеспечить собственную внешнюю схему маскирования ошибок VPU передает в хост-процессор некоторую внутреннюю информацию, которая необходима для работы внешней схемы. Например, VPU передать в хост-процессор данные о количестве макроблоков, пораженных ошибками. На данный момент такая информация указывается в регистре RET_DEC_PIC_ERR_MB (0x1C8). На основе этой информации хост-процессор может заменить фрагменты декодированного изображения, пораженные ошибками, другими фрагментами с замаскированными ошибками.

Для более эффективного маскирования ошибок в хост-процессоре следует передать информацию о векторе движения, информацию о слайсах, о значениях коэффициента DC intra, о карте распределения ошибок в макроблоках по каждому из изображений. Такая информация не является выходной по умолчанию. Но при необходимости ее можно легко добавить. Например, если в случае использования MPEG-4 с разделением данных хост-приложению необходимо воспользоваться собственными схемами маскирования ошибок, то блок VPU должен передать хост-процессору все параметры, которые не содержат ошибок, например, векторы движения, коэффициент DC intra и т. д. Хост-процессор может управлять данным процессом с использованием карты распределения ошибок, информации о слайсах битового потока и полученных внутренних параметров.

1.3.6.4.6 ПРИЧИНЫ ВОЗНИКНОВЕНИЯ ОШИБОК ПРИ ДЕКОДИРОВАНИИ ЗАГОЛОВКОВ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

1.3.6.4.6.1 AVC

Таблица 1.561 – Причины возникновения ошибок при декодировании заголовков AVC

Бит ошибки	ERROR_TYPE	Категория	Описание
0	AVCERR_DISABLE	Запрет декодирования ввиду того, что декодер AVC отключен	
1	AVCERR_MEETSEQEND _SPSINIT	В подаваемом потоке нет соответствия SPS.	
2	AVCERR_MEETSEQEND _PPSINIT	В подаваемом потоке нет соответствия PPS.	
3	AVCERR_NAL_REF_IDR	SPS	Если в блоке NAL содержится набор параметров последовательности, то nal_ref_idc не должен быть равен 0
4	AVCERR_RESERVED_ZERO _4BITS	SPS	reserved_zero_4bits должен быть равен 0.
5	AVCERR_BIT_DEPTH_LUMA _MINUS8	SPS	bit_depth_luma_minus8 должен быть равен 0.
6	AVCERR_BIT_DEPTH _CHROMA_MINUS8	SPS	bit_depth_chroma_minus8 должен быть равен 0.
7	AVCERR_LOG2_MAX _FRAME_NUM_MINUS4	SPS	log2_max_frame_num_minus4 должен быть менее 12.
8	AVCERR_PIC_ORDER_CNT _TYPE	SPS	pic_order_cnt_type должен быть менее 2.
9	AVCERR_LOG2_MAX_PIC _ORDER_CNT_LSB _MINUS4	SPS	log2_max_pic_order_cnt_lsb_minus4 должен быть менее 12 .
10	AVCERR_NUM_REF_FRAME _IN_PIC_ORDER _CNT_CYCLE	SPS	num_ref_frame_in_pic_order_cnt_cycle должен быть менее 255.
11	AVCERR_OFFSET_FOR _REF_FRAME	SPS	offset_for_ref_frame должен входить в диапазон значений от -2^31 до 2^31 - 1.
12	AVCERR_NUM_REF_FRAME	SSPS	num_ref_frames не должен превышать 16.
13	AVCERR_PIC_WIDTH_IN _MBS_MINUS1	SPS	pic_width_in_mbs_minus1 не должен превышать MAX_MB_NUM_X -1.

Продолжение таблицы 1.561

Бит ошибки	ERROR_TYPE	Категория	Описание
14	AVCERR_PIC_HEIGHT_IN_MAP_MINUS1	SPS	pic_height_in_map_minus1 не должен превышать MAX_MB_NUM_Y -1.
15	AVCERR_DIRECT_8X8_INFERENCE_FLAG	SPS	direct_8x8_inference_flag если frame_mbs_only_fag равен 0, то direct_8x8_inference_flag должен быть равен 1.
16	AVCERR_FRAME_CROP_OFFSET	SPS	frame_crop_left_offset, frame_crop_right_offset, frame_crop_top_offset, frame_crop_bottom_offset
17	AVCERR_OVER_MAX_MB_SIZE	SPS	pic_width_in_mbs и pic_height_in_map должны быть меньше 2048.
18	AVCERR_NOT_SUPPORT_PROFILEIDC	SPS	Поддерживаются значения 66 (Базовый), 77 (Главный), 100 (Высокий) и 128 (mvc).
19	AVCERR_NOT_SUPPORT_LEVELIDC	SPS	Поддерживаются значения до 51.
31	ERR_SEQ_HEADER_NOT_FOUND	Заголовок последовательности не найден	

1.3.6.4.6.2 VC-1(SP, MP, AP)

Таблица 1.562 – Причины возникновения ошибок при декодировании заголовков VC-1

Бит ошибки	ERROR_TYPE	Описание
1	VC1ERR_DISABLE	Запрет декодирования ввиду того, что декодер VC1 отключен
2	VC1ERR_0x00000004	Вторые 4 байта должны быть равны 4.(0x00000004)
3	VC1ERR_COMPLEX_PROFILE	Не поддерживается с потоком wmv3 со Сложным профилем. (profile=2)
4	VC1ERR_YUV411	YUV411 (3-й бит в ПРОФИЛЕ) должен быть равен 0.
5	VC1ERR_SPRITE	Спрайт (4-й бит в ПРОФИЛЕ) должен быть равен 0.
6	VC1ERR_X8INTRA	X8INTRA должен быть равен 0.
7	VC1ERR_BIT16XFORM	BIT16XFORM должен быть равен 1.
8	VC1ERR_DCT_TAB_SWITCH	DCT_TAB_SWITCH должен быть равен 0.
9	VC1ERR_BETA_RTM	BETA_RTM должен быть равен 1.
10	VC1ERR_PICSIZE	MAX_CODED_WIDTH и MAX_CODED_HEIGHT не должны превышать MAX_FRAME_X и MAX_FRAME_Y, соответственно.
31	ERR_SEQ_HEADER_NOT_FOUND	Заголовок последовательности не найден (только в AP VC-1)

Примечание SPS (набор параметров последовательности)

1.3.6.4.6.3 MPEG4 (короткий заголовок видеоданных)

Таблица 1.563 – Причины возникновения ошибок при декодировании MPEG4 (короткий заголовок видеоданных)

Бит ошибки	ERROR_TYPE	Категория	Описание
0	MP4ERR_DISABLE	Запрет декодирования ввиду того, что декодер MPEG-4 отключен	
1	MP4ERR_MARKER_BIT	PL	marker_bit не должен быть равен 0.
2	MP4ERR_ZERO_BIT	PL	zero_bit должен быть равен 0.
3	MP4ERR_MP4ERR_ANNEXD	PL	Не используется
4	MP4ERR_MP4ERR _ANNEX-EFG	PL	Приложение E (режим AC на основе синтаксисов), Приложение F (расширенный режим прогнозирования) и Приложение G (режим PB-кадра). Все эти биты должны быть равны 0.
5	MP4ERR_NUM_GOB	PL	num_gob не должен быть равен 0.
6	MP4ERR_UFEP	PL	Значения UFEP отличные от 000 и 001 резервируются.
7	MP4ERR_ANNEXD_PLUS PTYPE	PL	Не используются
8	MP4ERR_ANNEXEF_PLUS PTYPE	PL	Приложение E (режим AC на базе синтаксисов), Приложение F (расширенный режим прогнозирования). Все эти биты должны быть равны 0.
9	MP4ERR_ANNEXNRS _PLUSPTYPE	PL	Приложение N (выбор ссылочного изображения), Приложение R (декодирование независимого сегмента) и Приложение S (Альтернативное VLC Inter). Все эти биты должны быть равны 0.
10	MP4ERR_PICTURE_TYPE _CODE	PL	picture_type_code должен быть менее 1.
11	MP4ERR_ANNEXPQ_MPP TYPE	PL	Приложение P (ресэмплинг ссылочного изображения) и Приложение Q (обновление с уменьшенным разрешением). Все эти биты должны быть равны 0.
12	MP4ERR_EMUL_PREV	PL	Зарезервированный бит 2 и бит 1 для предотвращения моделирования должны быть равны 1.
13	MP4ERR_CMP	PL	CPM (мультиточка постоянного присутствия и мультиплексирование видеосигналов) должен быть равен 0.
14	MP4ERR_PAR	PL	PAR (зрительное соотношение пикселей) должен быть менее 5.
15	MP4ERR_CPCFC	PL	CPCFC (код частоты тактирования заказного изображения) не должен быть равен 0.
16	MP4ERR_UUI	PL	UUI (безусловный неограниченный индикатор вектора движения) должен быть равен 1 или 01.
17	MP4ERR_SSS	PL	SSS (подрежим слайсовой структуры) должен быть равен 0. Это значит, что функция упорядочения прямоугольных слайсов и произвольных слайсов не поддерживается.

Продолжение таблицы 1.563

Бит ошибки	ERROR_TYPE	Категория	Описание
18	MP4ERR_SEPB	PL	SEPB в sliceHeader263 должен быть равен 1.
30	COMERR_PIC_SIZE_ERR	PL	Размер изображения превышает допустимое значение.
31	ERR_SEQ_HEADER_NOT_FOUND	Заголовок последовательности не найден	

Примечание PL (уровень изображения)

1.3.6.4.6.4 MPEG4 (VOL, VOS)

Таблица 1.564 – Причины возникновения ошибок при декодировании заголовков MPEG4 (VOL, VOS)

Бит ошибки	ERROR_TYPE	Категория	Описание
0	MP4ERR_DISABLE	Не используется	
1	MP4ERR_VIDEO_OBJECT_LAYER_VERID	VOL	video_object_layer_verid должен находиться в диапазоне значений от 1 до 5 (рекомендуется значение 2).
2	MP4ERR_PAR_WIDTH	VOL	par_width не должен быть равен 0.
3	MP4ERR_PAR_HEIGHT	VOL	par_height не должен быть равен 0.
4	MP4ERR_CHROMA_FORMAT	VOL	chroma_format должен быть равен 1.
5	MP4ERR_MARKER_BIT	VOL	marker_bit должен быть равен 1.
6	MP4ERR_VIDEO_OBJECT_LAYER_SHAPE	VOL	video_object_layer_shape должен быть равен 0. (1: бинарный, 2: только бинарный, 3: оттенки серого)
7	MP4ERR_VOP_TIME_INC	VOL	vop_time_increment_resolution не должен быть равен 0.
8	MP4ERR_FIXED_VOP_TIME_INC	VOL	fixed_vop_time_increment не должен быть равен 0.
9	MP4ERR_OBMC_DISABLE	VOL	obmc_disable должен быть равен 1.
10	MP4ERR_SPRITE_ENABLE	VOL	sprite_enable должен быть равен 0 (GMC).
11	MP4ERR_NOT_8_BIT	VOL	not_8_bit должен быть равен 0.
12	MP4ERR_COMPLEXITY_EST_DISABLE	VOL	complexity_estimation_disable должен быть равен 1. В этом случае в каждом VOP отключается заголовок оценки сложности.
13	MP4ERR_SCALABILITY	VOL	Бит масштабирования должен быть равен 0.
30	COMERR_PIC_SIZE_ERR	VOL	Размер изображения превышает допустимое
31	ERR_SEQ_HEADER_NOT_FOUND	Заголовок последовательности не найден	

Примечание VOL (уровень визуальных объектов)

1.3.6.4.6.5 MPEG2

Таблица 1.565 – Причины возникновения ошибок при декодировании заголовков MPEG2

Бит ошибки	ERROR_TYPE	Категория	Описание
0	MP2ERR_DISABLE	Не используется	Запрет декодирования ввиду того, что декодер MP2 отключен
1	MP2ERR_HORIZONTAL_SIZE_VALUE	horizontal_size_value не должен быть равен 0.	
2	MP2ERR_VERTICAL_SIZE_VALUE	SH	vertical_size_value не должен быть равен 0.
3	MP2ERR_ASPECT_RATIO_INFORMATION	SH	aspect_ratio_information не должен быть равен 0.
4	MP2ERR_FRAME_RATE_CODE	SH	frame_rate_code не должен быть равен 0.
5	MP2ERR_MARKER_BIT	SH	marker_bit не должен быть равен 0.
6	MP2ERR_INTRA_QUANTISER_MATRIX	SH	intra_quantiser_matrix не должен быть равен 0.
7	MP2ERR_NON_INTRA_QUANTISER_MATRIX	SH	non_intra_quantiser_matrix не должен быть равен 0.
8	MP2ERR_CHROMA_FORMAT	SH	chroma_format должен быть равен 1.
9	MP2ERR_HORIZONTAL_SIZE_EXTENSION	SHE	horizontal_size_extension не должен быть равен 0.
10	MP2ERR_VERTICAL_SIZE_EXTENSION	SHE	vertical_size_extension не должен быть равен 0.
11	MP2ERR_BIT_RATE _EX-TENTION	SHE	Не используется
12	MP2ERR_MARKER_BIT _SE-QEXT	SHE	marker_bit должен быть равен 1.
13	MP2ERR_VBV_BUFFER_SIZE_EXTENTION	SHE	vbv_buffer_size_extention должен быть равен 0.
14	MP2ERR_FCODE	Array Fcode не должен быть равен 0.	
15	MP2ERR_PICTURE _STRUC-TURE	picture_structure не должен быть равен 0.	
16	MP2ERR_NOT_SUPPORTED_PROFILE	Неподдерживаемый профиль	
31	ERR_SEQ_HEADER_NOT_FOUND	Заголовок последовательности не найден	

Примечание SH (заголовок последовательности), SHE (расширение заголовка последовательности), SDE (расширение отображения последовательности)

1.3.6.4.6.6 RVx

Таблица 1.566 – Причины возникновения ошибок при декодировании заголовков RVx

Бит ошибки ERROR_TYPE		Описание
0	RVERR_DISABLE	Запрет декодирования ввиду того, что декодер RV отключен
1	RVERR_SUB_MOF _FLAG31TO16	SUB_MOF_FLAG[31:16] должен быть равен 5256h (RV).
2	RVERR_SUB_MOF _FLAG16TO0	SUB_MOF_FLAG[15:0] должен быть равен 3430h или 3330h (40 или 30).
3	RVERR_RPR_PICSIZE_NUM	RprPicSizeNum должен находиться в диапазоне значений от 0 до 7.
4	RVERR_FRAME_BYTE_SIZE	FrameByteSize должен быть больше 4 байтов.
5	RVERR_NUM_SEG0	num_segment не должен быть равен 0.
6	RVERR_NUM_SEG256	num_segment должен быть менее 256.
7	RVERR_SEG_VALID	is_valid[i] должен быть равен 1.
8	RVERR_ECC	ECC (version_encoding[3] в RV8) должен быть равен 1.
9	RVERR_PICQUANT	PicQuant должен находиться в диапазоне значений от 0 до 30.
10	RVERR_BITSTRM_VER	bitstream_version должен быть равен 0.
11	RVERR_X10_HEADER	x10_header должен быть равен 0.
12	RVERR_FIRSTMBADDR	FirstMbAddr должен быть равен 0.

1.3.6.4.6.7 VP8

Таблица 1.567 – Причины возникновения ошибок при декодировании заголовков VP8

Бит ошибки ERROR_TYPE		Описание
0	VP8ERR_DISABLE	Запрет декодирования ввиду того, что декодер VP8 отключен
1	VP8ERR_START_FROM _IN- TER	Инициализация последовательности начинается с кадра inter.
2	VP8ERR_SEQ_PIC_WIDTH _SPEC_OVER	Размер изображения по ширине превышает допустимое значение.
3	VP8ERR_SEQ_PIC_HEIGHT _SPEC_OVER	Размер изображения по высоте превышает допустимое значение.

1.3.6.4.6.8 AVS

Таблица 1.568 – Причины возникновения ошибок при декодировании заголовков AVS

Бит ошибки ERROR_TYPE		Описание
0	AVSERR_DISABLE	Запрет декодирования ввиду того, что декодер AVS отключен
1	AVSERR_PROFILE	Неподдерживаемый профиль. (Поддерживается только профиль Jizhun)
2	AVSERR_PICSIZE	Ошибка в размере изображения
31	ERR_SEQ_HEADER_NOT _FOUND	Заголовок последовательности не найден

1.3.6.4.7 ДЕКОДЕР. FAQ

Данный раздел представляет собой общий список часто задаваемых вопросов заказчиков и ответов на них, касающихся VPU (блок обработки видеосигналов). В нем рассматриваются случаи возникновения ошибок и ограничения, накладываемые в ходе работы блока VPU, а также объясняется то, как их следует обрабатывать.

1.3.6.4.7.1 Сбой при декодировании

В данном разделе рассматриваются случаи возникновения сбоев декодирования, а также некоторые ограничения, возникающие на двух основных этапах декодирования - SEQ_INIT и PIC_RUN.

1.3.6.4.7.1.1 Сбой при инициализации последовательности (SEQ_INIT)

После создания нового экземпляра и подачи входного битового потока в буфер потока, хост может передать в VPU команду DEC_SEQ_INIT, которая позволит получить из битового потока информацию о конфигурации декодера. После разбора синтаксисов заголовков декодер возвращает обратно важную информацию о конфигурации декодера, включая данные о размере изображения, частоте кадров, минимальном количестве буферов кадра и т. д.

Однако если блок VPU не выполнит декодирование заголовка последовательности, VPU не сможет декодировать изображение. Затем VPU через регистр RET_DEC_SEQ_SEQ_ERR_REASON (0x1E0) передает сообщение об ошибке. Более подробно вопрос сбоев рассмотрен в - ПРИЧИНЫ ВОЗНИКНОВЕНИЯ ОШИБОК ПРИ ДЕКОДИРОВАНИИ ЗАГОЛОВКОВ ПОСЛЕДОВАТЕЛЬНОСТЕЙ.

Ниже представлены наиболее часто возникающие проблемы, которые возникают после выполнения команды SEQ_INIT.

Не найден заголовок последовательности

Если в битовом потоке отсутствует стартовый код, или если в битовом потоке присутствует неопределенный бит nal_unit_type, то это может вызвать «зависание» блока VPU. Для успешного декодирования хост должен получить из синтаксического анализатора стартовый код и поместить его в битовый поток.

Настройка порядка следования байтов FPGA

Когда между порядком следования байтов в памяти, доступ к которой осуществляется со стороны хоста, и настройкой порядка следования байтов в VPU наблюдается несоответствие, программа может «зависнуть» на DES_SEQ_INIT, а регистр BUSY_FLAG покажет бесконечно высокое значение. Также возможно, чтобы команда DEC_SEQ_INIT успешно возвратилась, не смотря на ошибку в данных буфера кадра.

Пользователям следует быть внимательными при выполнении имитационного моделирования. После моделирования RTL на ПК пользователи, несмотря на разный порядок следования байтов при доступе к памяти, склонны делать то же самое на FPGA, не меняя никакие настройки. Перед началом работы VPU на стороне хоста необходимо установить регистр BIT_BIT_STREAM_CTRL (0x10C) в соответствии с порядком следования байтов в собственной модели RTL или FPGA. По умолчанию, установлен обратный порядок следования байтов в буфере битового потока.

Вне спецификации

Бывают ситуации, когда VPU не может выполнить декодирование, по причине того, что не поддерживает данный поток (все спецификации). Далее приводится перечень стандартов, которые не описываются в спецификации, и которые блок VPU, как правило, умеет декодировать, но не может поддерживать некоторые из функций видеоданных.

H.263: поддерживаются только Приложение I, J, K и T

MPEG4: GMC, OBMC и MS MPEG4 версии 1/2 не поддерживаются

MPEG2: vertical_size_extension/horizontal_size_extension не поддерживаются

RV: RV7 не поддерживаются

VC1: WMV7/8(FCC WMV1 и WMV2) не поддерживаются

1.3.6.4.7.1.2 Сбой при декодировании изображения (PIC_RUN или VPU_DecStartOneFrame)

При активации операции декодирования изображения хост должен передать в блок VPU информацию по следующим позициям:

включение опции поиска I-кадра;

режим попуска кадров;

PicFlush MPEG2;

режим использования фильтра деблокирования RV.

Примечание Более подробную информацию по данным параметрам можно найти в описании структуры DecParam в «Справочном руководстве к API».

После того, как все эти параметры будут предоставлены блоку VPU, хост по команде DEC_PIC_RUN сможет приступить к операции декодирования изображения.

Однако после выполнения команды DEC_PIC_RUN регистр главного интерфейса BIT_BUSY_FLAG (0x160) может сохранить значение 1, вызывая, тем самым, «зависание» системы. И, чтобы выявить основную проблему, пользователь может запустить проверку BIT_CUR_PC (0x018). Ниже рассмотрены 3 наиболее частые причины этого.

Подкачка потока данных

Для буфера битового потока, работающего в режиме прерывания, существует одно ограничение, на которое следует обратить внимание. Бит buffer_empty interrupt может передаваться только один раз. Объем данных битового потока на стороне хоста не должен превышать 512 байтов, даже после получения сигнала прерывания buffer_empty, и блок VPU не будет генерировать еще один сигнал прерывания BIT_EMPTY; это необходимо для предотвращения перерасхода ресурсов хоста вследствие генерирования бесконечного числа сигналов прерывания. Чтобы проверить обновление битового потока VPU должен находиться в канале.

Поток данных, содержащий серьезную ошибку

VPU, как правило, не декодирует поток, содержащий серьезную ошибку, которая вынуждает VPU постоянно находиться в состоянии занятости. Вот почему VPU выполняет рекурсивную программу или ожидает цикл микро-ПО. При этом хост должен проверить следящий таймер и перезапустить VPU.

1.3.6.4.7.1.3 RET_DEC_PIC_DECODED_IDX

После декодирования одного кадра VPU возвращает в регистр RET_DEC_PIC_DECODED_IDX (0x1DC) индекс декодированного кадра. Данный регистр может иметь отрицательное значение в одном из следующих случаев.

-1 (0xFFFF): Когда нет достаточного количества буферов кадра для продолжения процесса декодирования.

– H.264:

Даже если хост освобождает буферы кадров с помощью функции VPU_DecClrDispFlag(), VPU может возвратить -1, потому что некоторые из буферов используются в качестве буферов ссылочных кадров, особенно, при декодировании потока H.264, в котором в течение длительного времени применяется много ссылочных кадров и ссылок. Например, если хост освобождает второй кадр,

dispDoneIdx = 2; VPU_DecClrDispFlag(handle, dispDoneIdx);

а буфер второго кадра используется в качестве ссылочного кадра, VPU в ответ на RET_DEC_PIC_DECODED_IDX on возвращает-1.

– VC-1:

В случае использования VC1, если в VPU есть только один доступный буфер кадра, VPU в ответ на RET_DEC_PIC_DECODED_IDX возвращает -1, потому что для декодирования одного кадра, подлежащего пост-обработке, требуется два буфера кадра.

-2(0xFFFE): когда в заголовке изображения есть несколько ошибок, или когда отсутствуют декодированные данные кадра. В обоих случаях информация о декодированных выходных данных не доступна, в то время как информация об отображении доступна.

1.3.6.4.7.1.4 RET_DEC_PIC_DISPLAY_IDX (0x1C4)

После декодирования одного кадра VPU возвращает в регистр RET_DEC_PIC_DISPLAY_IDX (0x1C4) индекс отображаемого кадра. Данный регистр получает отрицательное значение в трех случаях:

-1 (0xFFFF): когда больше нет выходных данных, подлежащих отображению. XOCT приложение может легко «узнать» EOS это или нет.

-2 (0xFFFE): если был пропуск кадра или когда нет больше изображений, подлежащих отображению, после выполнения текущей команды PIC_RUN.

-3 (0xFFFD): если при упорядочении изображений произошла задержка отображения.

1.3.6.4.7.2 Минимальное число буферов кадра

После выдачи команды SEQ_INIT в регистре RET_DEC_SEQ_FRAME_NEED (0x1CC) хост может получить данные о минимальном числе буферов кадра, которые представляют собой области памяти, предназначенные для хранения закодированных изображений. В зависимости от формата видеопотока и информации в заголовке последовательности VPU может возвратить другое значение буфера кадра.

Расчет

Это минимальное количество кадров, которое VPU возвращает для каждого видеостандарта. Большая часть этих значений являются фиксированными величинами, и только для формата H.264 требуется некоторый расчет в соответствии с информацией синтаксисов потока.

AVS: 4 кадра (фиксированное значение)

DIV3: 3 кадра (фиксированное значение)

JPEG: 2 кадра (фиксированное значение, но для разных продуктов может быть разным)

MPEG-2/4: 4 кадра (фиксированное значение, но если включен фильтр деблокирования, к нему может быть добавлен еще один буфер.)

Theora: 3 кадра (фиксированное значение)

VP8: 4 кадра (фиксированное значение)

RV: 4 кадра (фиксированное значение)

VC1: 4 кадра (по умолчанию)

Но если (MaxBFrames 1!= 0 || профиль == Расширенный профиль), MinFrameNum = 6 кадров

• H.264/AVC :

Если (ReorderEnable (задано на хосте) == 0 || picture_order_cnt_type(in SPS) == 2), MinFrameNum = num_ref_frames + 2

В противном случае, (ReorderEnable == 1 || picture_order_cnt_type !=2)

Если в заголовке VUI есть синтаксис max_dec_frame_buffering, то MinFrameNum = max_dec_frame_buffering + 2

Если отсутствует max_dec_frame_buffering, то MaxDpbSize рассчитывается по приведенной ниже формуле в соответствии Приложением A спецификации H.264. (Таблица A-1 – Границы уровня) как показано на рисунке I.1 - “Приложение A H.264.

Таблица 1.179 – Границы уровня”.

calcMaxDpbSize(int levelIdc, int frmX, int frmY, int chromaIdc)

```
{
```
1MaxBFrames – это синтаксис в метаданных уровня последовательности, а num_ref_frames – это синтаксис в SPS.

```
int      maxDpb, maxDpbSize;
// MAX DPB BY LEVEL
if (levelIdc <= 10)
maxDpb = (int) (148.5 * 1024); else if (levelIdc <= 11)
maxDpb = (int) (337.5 * 1024); else if (levelIdc <= 20)
maxDpb = (int) (891.0 * 1024); else if (levelIdc <= 21)
maxDpb = (int) (1782.0 * 1024); else if (levelIdc <= 3)
maxDpb = (int) (3037.5 * 1024); else if (levelIdc <= 32)
maxDpb = (int) (7680.0 * 1024); else if (levelIdc <= 42)
maxDpb = (int) (13056.0 * 1024); else if (levelIdc <= 5)
maxDpb = (int) (41400.0 * 1024);
else
maxDpb = (int) (69120.0 * 1024);
maxDpbSize = maxDpb / (frmX*frmY + (frmX*frmY/2)*chromaIdc); maxDpbSize = MIN(maxDpbSize, 16);

return maxDpbSize;
}
```

MVC: Почти все значения, такие же как для форматов H.264/AVC, но для текущего кадра вместо +2 используется значение + 3

Если (ReorderEnable (задано на хосте) == 0 || picture_order_cnt_type(in SPS) == 2), MinFrameNum = num_ref_frames + 3

В противном случае (ReorderEnable == 1 || picture_order_cnt_type !=2)

Если в заголовке VUI есть синтаксис max_dec_frame_buffering, то MinFrameNum = max_dec_frame_buffering + 3

Представленная выше таблица объясняет и убеждает заказчиков в том, что значение DPB (буфер декодированных изображений) приходит от VPU, потому что им может показаться, что VPU должен возвращать значение минимального числа буферов кадра, необходимых для декодирования 264 потоков, как "количество ссылочных кадров + 2". И этот факт может вызвать у заказчиков сомнение относительно DPB и того, что это значение больше требуемого. Вот почему мы и представили формулу расчета для потока H.264.

Рисунок 1.169 представляет собой фрагмент Приложения A спецификации H.264 (Таблица A-1 - Границы уровня), и служит для быстрой ссылки.

Table A-1 – Level limits

Level number	Max macroblock processing rate MaxMBPS (MB/s)	Max frame size MaxFS (MBs)	Max decoded picture buffer size MaxDpbMbs (MBs)	Max video bit rate MaxBR (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s, or cpbBrNalFactor bits/s)	Max CPB size MaxCPB (1000 bits, 1200 bits, cpbBrVclFactor bits, or cpbBrNalFactor bits)	Vertical MV component range MaxVmvR (luma frame samples)	Min compression ratio MinCR	Max number of motion vectors per two consecutive MBs MaxMvsPer2Mb
1	1 485	99	396	64	175	[−64,+63.75]	2	-
1b	1 485	99	396	128	350	[−64,+63.75]	2	-
1.1	3 000	396	900	192	500	[−128,+127.75]	2	-
1.2	6 000	396	2 376	384	1 000	[−128,+127.75]	2	-
1.3	11 880	396	2 376	768	2 000	[−128,+127.75]	2	-
2	11 880	396	2 376	2 000	2 000	[−128,+127.75]	2	-
2.1	19 800	792	4 752	4 000	4 000	[−256,+255.75]	2	-
2.2	20 250	1 620	8 100	4 000	4 000	[−256,+255.75]	2	-
3	40 500	1 620	8 100	10 000	10 000	[−256,+255.75]	2	32
3.1	108 000	3 600	18 000	14 000	14 000	[−512,+511.75]	4	16
3.2	216 000	5 120	20 480	20 000	20 000	[−512,+511.75]	4	16
4	245 760	8 192	32 768	20 000	25 000	[−512,+511.75]	4	16
4.1	245 760	8 192	32 768	50 000	62 500	[−512,+511.75]	2	16
4.2	522 240	8 704	34 816	50 000	62 500	[−512,+511.75]	2	16
5	589 824	22 080	110 400	135 000	135 000	[−512,+511.75]	2	16
5.1	983 040	36 864	184 320	240 000	240 000	[−512,+511.75]	2	16

Table A-1 – Level limits – Таблица A-1 – Границы уровня

Level number – Количество уровней

Max macroblock processing rate MaxMBPS (MB/s) – Максимальная скорость обработки макроблока MaxMBPS (Мбит/с)

Max Frame Size MaxFS (MBs) – Максимальный размер кадра MaxFS (Мбайт)

Max decoded picture buffer size MaxDpbMbs (MBs) – Максимальный объем буфера закодированного изображения MaxDpbMbs (Мбайт)

Max video bit rate Max BR – Максимальная битовая скорость видеоданных Max BR

(1000 bits/s, 1200 bits/s

cpbBrVclFactor bits/s, or cpbBrNalFactor bits/s) – (1000 бит/с, 1200 бит/с

cpbBrVclFactor битс или cpbBrNalFactor бит/с)

Max CPB size MaxCPB – Максимальный размер CPB MaxCPB

(1000 bits/s, 1200 bits/s

cpbBrVclFactor bits/s, or

cpbBrNalFactor bits/s) – (1000 бит/с, 1200 бит/с

cpbBrVclFactor бит/с или

cpbBrNalFactor бит/с)

Vertical MV component range MaxVmvR – Диапазон компонент вектора движения по вертикали MaxVmvR

(Luma frame samples) – (Сэмплы кадра по компоненте яркости)

Min compression ratio MinCR – Минимальный коэффициент сжатия MinCR

Max number of motion vectors per two consecutive MBs MaxMvsPer2Mb – Максимальное число векторов движения на два последовательных макроблока MaxMvsPer2Mb

Рисунок 1.169 - Приложение A H.264. Таблица A-1 – Границы уровня

1.3.6.4.7.3 Поиск I-кадра и пропуск кадра

С помощью регистра интерфейса хост может воспользоваться опцией поиска I-кадра или опцией пропуска P/B- или B-кадра (см. таблицу 2.83 - “CMD_DEC_PIC_OPTION (0x194)”).

Примечание Подробное описание регистра хоста приведено в руководстве для программистов VPU.

1.3.6.4.7.3.1 Поиск I-кадра

Данная функция поиска I-кадра используется для случайного доступа в ходе телевизионного вещания и в приложениях хранения DVD. Поскольку I-кадры меньше всего поддаются сжатию и не требуют декодирования других видеокадров, для декодера они могут выступать в качестве точки случайного доступа.

С помощью настройки второго бита регистра CMD_DEC_PIC_OPTION на стороне хоста можно включать или отключать опцию поиска I-кадра.

В ходе декодирования потока данных H.264/AVC они могут иметь I-кадр и кадр IDR. Если ХОСТ хочет найти IDR-кадр, то 11-й бит регистра CMD_DEC_PIC_OPTION должен быть установлен в 1.

1.3.6.4.7.3.2 Режим пропуска кадра

Режим пропуска кадра позволяет пропускать определенное количество P/B- или B-кадров. Хост может включить режим пропуска кадров для следующего кадра, установив 3-й и 4-й биты регистра CMD_DEC_PIC_OPTION[4:3] с одним из следующих значений.

0: отключить пропуск изображений

1: пропустить все кроме I(IDR)-изображений, то есть пропускаются и P-, и B-изображения

2: пропустить B-изображение (если в H.264 nal_ref_idc == 0)

3: безусловный пропуск одного изображения

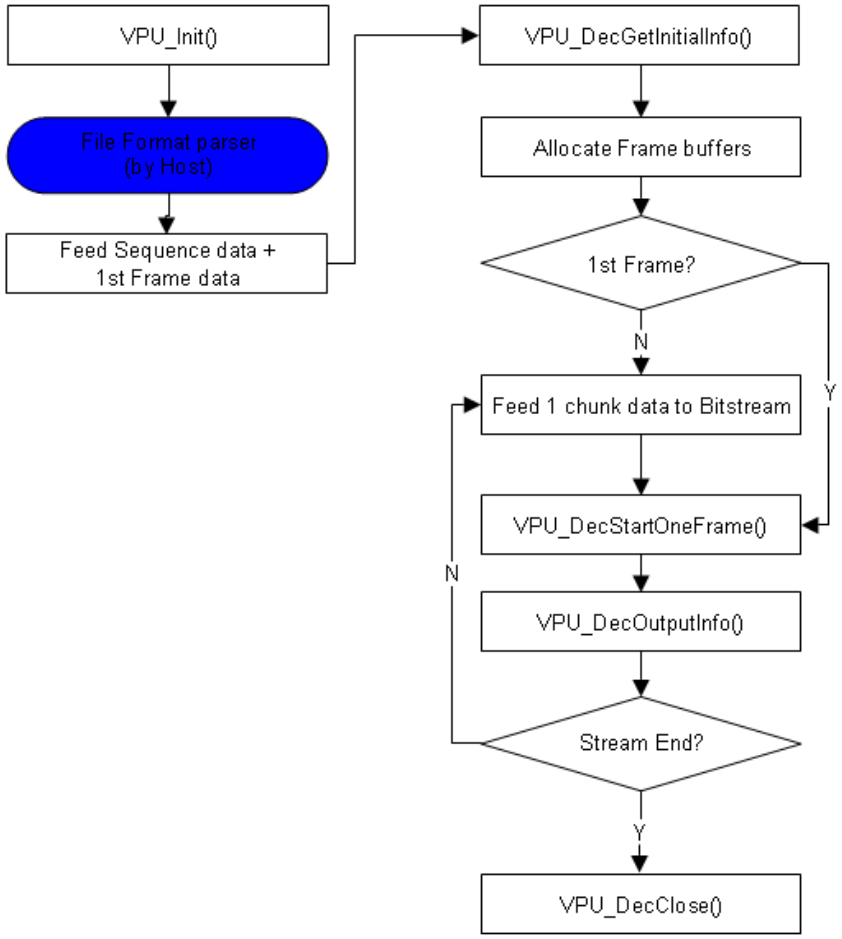
Исключения

Если функции iframeSearch и frame skip включены, то VPU игнорирует функцию пропуска кадра и работает только пропуск iframe.

1.3.6.4.7.4 Декодирование потока в контейнере файлов

1.3.6.4.7.4.1 Операция воспроизведения из контейнера файлов

На рисунке 1.170 представлена блок-схема процесса декодирования в режиме воспроизведения из контейнера файлов.



File Format parser (by Host) – Анализатор формата файлов (на стороне хоста)
Feed Sequence data – Данные подаваемой последовательности
1st Frame data – Данные 1-го кадра
Allocate Frame buffers? – Распределить буферы кадра?
1st Frame? – 1-й кадр?
N – Н
Y – Д
Feed 1 chunk data to Bitstream – Подача данных 1 блока в битовый поток
N – Н
Y – Д
Stream End – Окончание потока

Рисунок 1.170 – Блок-схема декодирования с использованием контейнера файлов

1.3.6.4.7.4.2 Ограничения

Декодер VPU, поддерживающий мультистандартное кодирование, имеет некоторые ограничения в работе с потоком контейнера файлов. Во избежание проблем, которые могут возникнуть в ходе подкачки битового потока в блок VPU, хост должен учитывать следующее:

Генерирование стартового кода

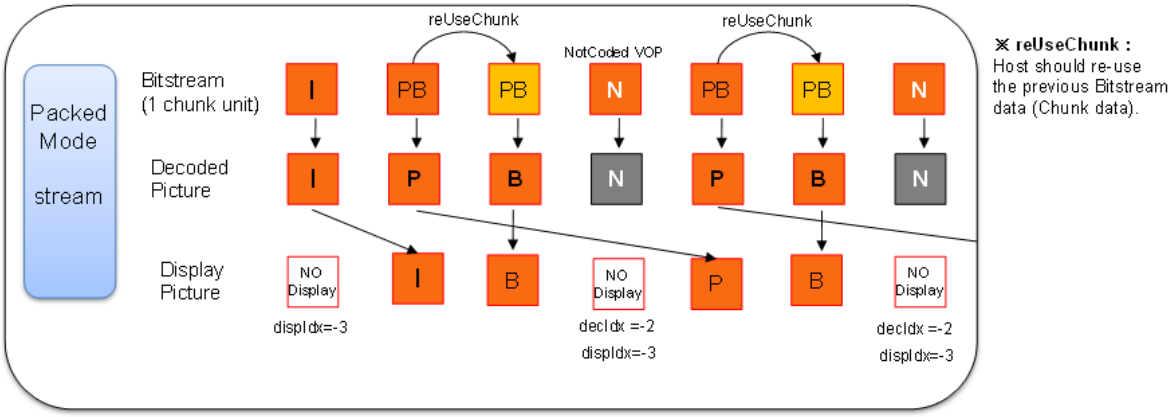
ES (элементарный поток) H.264 в контейнере файлов MPEG-4 не имеет префикса стартового кода (00 00 01). Анализатор или хост должен сам создать префикс стартового кода.

Генерирование заголовков последовательности и изображения

В форматах DIV3, RV, VC-1 и VP8, в ES не содержится никакой информации о заголовке изображения и последовательности. Поэтому анализатор или хост должны создать эту информацию в определенном формате потока байтов, определяемом “Chips&Media”. Особенности выполнения данной операции описываются в Приложении В – ФОРМАТЫ ПОТОКА БАЙТОВ CNM.

Пакетный режим MPEG-4

Для декодирования битового потока в пакетном режиме MPEG-4 (пакетированного битового потока) в данных 1 блока может быть 2 кадра. После выполнения команды CMD_DEC_PIC_RUN (по два раза на каждую строку) хост может начать подкачку данных блока. На рисунке I.3 - “Битовый поток в пакетном режиме MPEG-4” проиллюстрирован процесс обработки пакетированного потока MPEG-4, упорядочения декодированного изображения и отображения данных. В этом случае VPU возвращает в регистр chunkReuseRequired значение 1, и после этого хост должен вызвать еще раз функцию VPU_DecStartOneFrame() с тем же битовым потоком.



Packed Mode stream – Поток данных в пакетном режиме
Bitstream (1 chunk unit) – Битовый поток (1 блок)
Decoded Picture – Декодирование изображение
Display Picture – Отображаемое изображение
NO Display – БЕЗ отображения
NO Display – БЕЗ отображения
NO Display – БЕЗ отображения
Host should re-use the previous Bitstream data (Chunk data) – Хост должен повторно использовать данные предыдущего битового потока (данные блока)

Рисунок 1.171 – Битовый поток в пакетном режиме MPEG-4

PB: Пакетированный битовый поток

N: Фиктивный блок, следующий после пакетного режима; не кодированные данные.

reUseChunk: Хост должен пользоваться существующим потоком и не запрашивать новый поток из VPU.

Для битовых потоков в пакетном режиме MPEG-4, чтобы повторно использовать блок, следует декодировать 2 кадра. Индекс отображения при первом декодировании не изменяется, но изменяется при втором.

1 полукадр в 1 блоке

Блоком может быть изображение полукадра, но только если оно содержит 1 полукадр. В этом случае, для передачи в VPU всего битового потока данных кадра, содержащего верхний и нижний полукадры, VPU должен подать в буфер битового потока другой блок, потому что VPU может обрабатывать только блок кадров.

Если VPU выполняет функцию PIC_RUN только с данными 1 полукадра, VPU реагирует следующим образом:

Режим прерывания: происходит прерывание INT_BIT_BIT_BUF_EMPTY.

Режим отката: VPU приостанавливает декодирование и откатывается назад к состоянию перед выполнением команды PIC_RUN.

Режим PicEnd: маскируется 2-й полукадр.

1.3.6.4.7.5 FOURCC

FourCC (четырехсимвольный код) – это последовательность из четырех байтов, используемых для уникальной идентификации форматов данных. Одной из наиболее известных областей применения кода FourCC является идентификация видео-кодека, используемого в контейнере файлов, таком как AVI, MP4, MKV. К стандартным идентификаторам относятся DIVX, XVID и H264.

Ниже представлена таблица соответствия между кодом FourCC и видеостандартами, поддерживаемыми блоком VPU.

Таблица 1.569 – Таблица соответствия FourCC и стандартов

Стандарт	COD_STD	FourCC	Подтип	MP4_ASP_CLASS Reg- ister	AUX_STD	Комментарий
H.264	0	H264				
		X264				
		avc1				

Стандарт	COD_STD	FourCC	Подтип	MP4_ASP_CLASS Register	AUX_STD	Комментарий
		davc				
		NDXC				
		NDXP				
		NDXS				
		NDXH				
VC-1	1	WVC1	SMPTE VC1			
		WMV1	WM Video7			Не поддерж.
		WMV2	WM Video8			Не поддерж.
		WMV3	WM Video9			Частично поддерживается
MPEG2	2	MMES				
		LMP2				
		MP2V				
MPEG4	3	D263	H.263			
		DIV3	DivX3.11		1	
		DIV4		5		
		DIV5		1		
		DIVX		5		
		DX50	DivX V5	1		
		ILVR	H.263+			
		L263	H.263			
		M263	H.263			
		m4cc	MPEG-4			
		M4CC	MPEG-4			
		MPG4	MS MPEG4V1			Не поддерж.
		MP42	MS MPEG4V2			Не поддерж.
		MP43	MS MPEG4V3		1	DivX3.11
		RMP4	MPEG-4			
		SAN3	DivX3.11		1	
		SEDG	MPEG-4			
		SVQ1	Sorenson Spark			
		VDOW	H.263			
		VIVO	H.263			
		x263	H.263			
		XVID	XVID	2		
		ZyGo	H.263			
		MP4S	MPEG4SP			

Стандарт	COD_STD	FourCC	Подтип	MP4_ASP_CLASS Reg- ister	AUX_STD	Комментарий
RealVideo	4	rv20	RV7 G2			Не поддерж.
		rv30	RV8			
		rv40	RV9/10			
AVS	5	AVS2	профиль			
			jizhun AVS			
JPEG	6	DMB2	Motion JPG			
			FLJP Motion JPG (кодирование полукадров)			
		JPEG				
		LJ11	JPEG 4:1:1			
		LJ22	JPEG 4:2:2			
		LJPG	Motion JPG			
		mjpg	Motion JPG (w таблица Huffman)			
			PJPG			
VP8	7	VP80				

1.3.6.4.8 НАСТРОЙКА ПАРАМЕТРОВ КОДЕРА

Кодер Chips&Media семейства CODA9, благодаря наличию большого числа параметров кодирования, отличается гибкостью, и поэтому он может использоваться во множестве разных приложений. Он поддерживает различные функции, влияющие на работу кодера, каждая из которых может быть сконфигурирована независимо от остальных. В данном прикладном примечании описываются параметры кодирования, необходимые для настройки работы кодера, и, в целях наилучшего использования кодера, приводятся руководящие принципы в отношении рекомендованных значений некоторых параметров для нескольких общих сценариев использования, видеоконференцсвязи, телевизионного вещания и хранения видеофайлов.

1.3.6.4.8.1 Определение параметров кодирования

Ниже приводятся все параметры конфигурирования кодера семейства CODA9.

1.3.6.4.8.1.1 Базовые параметры

YUV_SRC_IMG

Имя файла, содержащего исходное изображение

FRAME_NUMBER_ENCODED

Количество кадров, подлежащих кодированию

PICTURE_WIDTH

Ширина исходного изображения (размер изображения по горизонтали). Данное значение должно быть выше 32 и кратно 16.

PICTURE_HEIGHT

Высота исходного изображения (размер изображения по вертикали). Данное значение должно быть выше 16 и кратно 16.

FRAME_RATE

Частота кадров, выраженная в к/с, например, 30 к/с. Данное значение используется для расчета числа целевых битов одного кадра в кодере по следующей формуле:

Число целевых битов на кадр = Битовая скорость / Частота кадров

УРОВЕНЬ

Уровень, определенный в Приложении A H.264. Данное значение должно в десять раз превосходить количество уровней. Например, согласно таблице A-1 в Приложении A, значение 32 показывает, что количество уровней равно 3.2. Когда данное значение 0, кодер автоматически рассчитывает количество уровней исходя из других параметров.

GOP_PIC_NUMBER

Количество GOP (групп изображений). Данное значение показывает количество изображений, отсчет которых начинается от одного I-изображения и заканчивается при появлении следующего I-изображения.

0: Первое изображение кодируется в I-изображении, а остальные кодируются в P-изображении.

1: Все изображения кодируются в I-изображении.

– 2: I, P, I, P, …

– 3: I, P, P, I, P, P, I, …

– 4: I, P, P, P, I, P, P, P, …

CONSTRAINED_INTRA

Constrained_intra_pred_flag в синтаксисе стандарта H.264.

0: Кодер использует inter-спрогнозированные макроблоки для intra-прогнозирования.

1: Кодер не использует inter-спрогнозированный макроблок для intra-прогнозирования.

В случае необходимости использования опции CIR (Циклическая регенерация Intra) настоятельно рекомендуется установить данное поле в 1.

IDR_INTERVAL

IDR_MODE может иметь значение в диапазоне от 0 до 15.

0: Только первое изображение является IDR.

1: Все I-изображения являются IDR.

n: Каждые n-ые I-изображения являются IDR.

INTERLACED_PIC

Режим чересстрочного кодирования

0: Кодирование кадра

1: Кодирование полукадра

Для чересстрочного закодированного изображения кодер может работать с форматом YUV чересстрочно закодированного полукадра, только если он входной.

FIELD_REFERENCE_MODE

Если регистр INTERLACED_PIC установлен в 1, данный флаг покажет, какой из полукадров является ссылочным для каждого полукадра.

0: Ссылка на полукадр той же четности

1: Ссылка на полукадр противоположной четности

SKIP_PIC_NUMS

Данный параметр позволяет определить количество кадров, подлежащих пропуску. Если FrameNum равно SKIP_PIC_NUM, то кодер пропускает этот кадр. В случае, когда кодер кодирует с помощью CBR, и общее число битов на текущий момент остается слишком низким, то данный параметр может использоваться для обработки события переполнения буфера потока.

INTRA_MB_REFRESH

Данный параметр позволяет включить опцию CIR (циклического обновления Intra) с заданным числом макроблоков intra.

0: Не использует CIR

N: N число макроблоков intra в кадре. Макроблок intra обновляется на предварительно установленном этапе по приведенной ниже формуле.

STEP = MAX (((MB_NUM_X>>3) * (MB_NUM_Y>>3)), 2)

Например, при кодировании изображения размером 160x80 с настроенным регистром INTRA_MB_REFRESH на 8, кодер добавляет прогоны CIR макроблока intra (см. рисунок 1.172) .

PICTURE_WIDTH: 160
PICTURE_HEIGHT: 80
INTERLACED_PIC: 0
INTRA_MB_REFRESH: 8
– STEP = (20 >> 3)*(10 >> 3) = 2*1

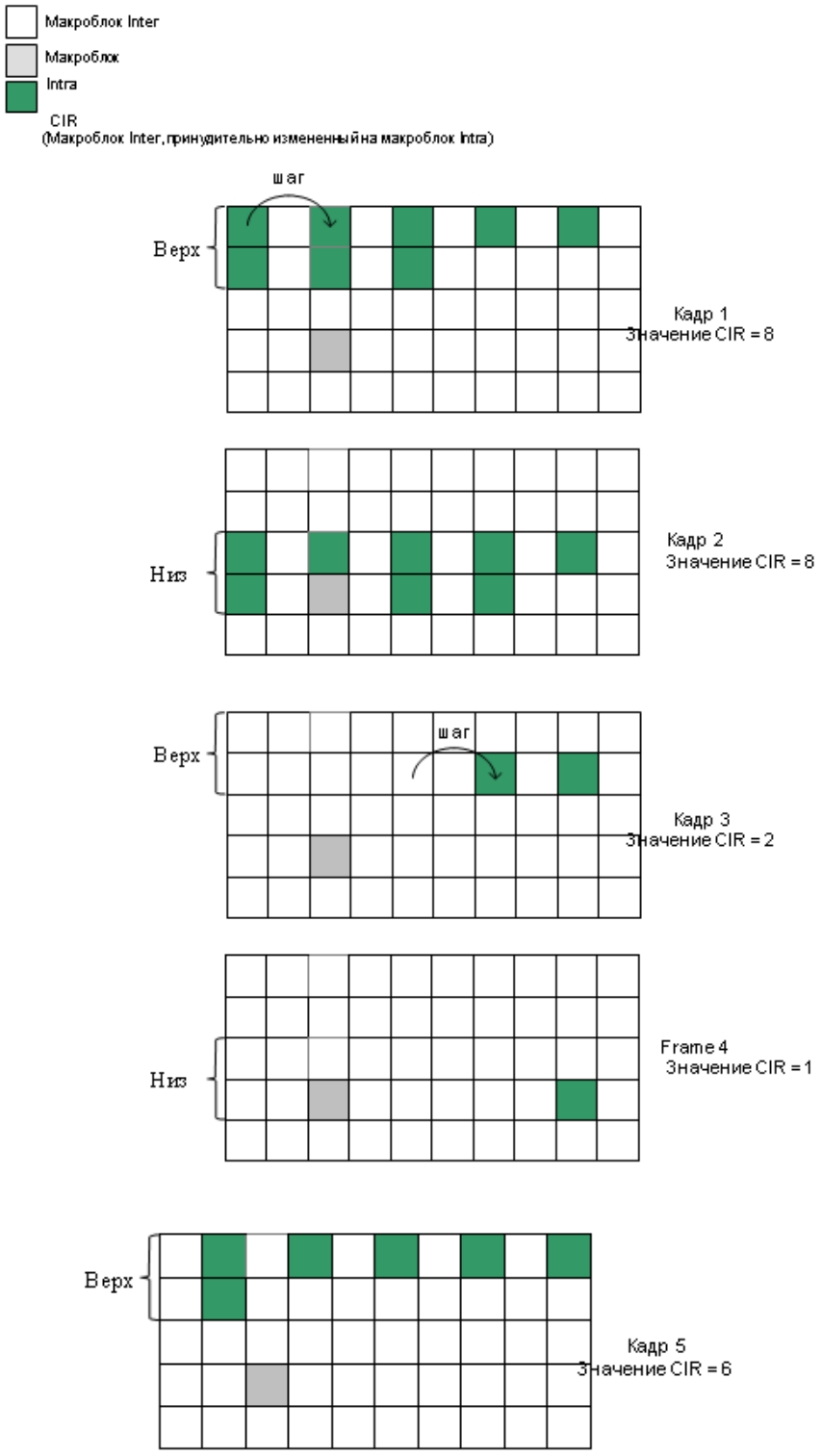


Рисунок 1.172 – Пример действия CIR

Опция CIR в основном действует на половину кадра (см. рисунок 1.172). CIR добавляются в первую половину кадра 1. Также CIR добавляются во вторую половину кадра 2. Если кодер считает макроблоки для добавления каждых двух макроблоков (ШАГ=2), макроблок intra он не добавляет.

Если счет CIR достигает заданного регистра INTRA_MB_REFRESH, то кодер перезапускается и снова начинает работать. Кроме того, кодер может продолжать подсчитывать CIR по кадрам, что можно увидеть на кадре 3 и кадре 5.

CHROMA_FORMAT_400

Формат субсэмплирования по компоненте цветности

– 0 - 4:2:0

1 - 4:0:0 (Данный режим МОНО разрешен только в HP)

SEARCH_RANGE_X

Диапазон поиска в направлении по горизонтали. Действительны следующие значения:

– 0: -64 ~ 63

– 1: -48 ~ 47

– 2: -32 ~ 31

– 3: -16 ~ 15

SEARCH_RANGE_Y

Диапазон поиска в направлении по вертикали. Действительны следующие значения:

– 0: -48 ~ 47

– 1: -32 ~ 31

– 2: -16 ~ 15

ME_USE_ZERO_PMV

Данный параметр используется в ходе оценки параметров движения для расчета расхода PMV в режиме пропуска. Если он равен 1, кодер берет PMV от (0,0), а не от усредненного значения MV от соседних блоков, что может оказывать влияние на качество кодирования или вызвать искажение.

Для обеспечения устойчивого функционирования, описываемого в каждой спецификации видеокодека, для кодера H.264/AVC лучше устанавливать 0, а для кодера MPEG-4 - 1.

– H.264/AVC: 0

MPEG-4: 1

WEIGHT_INTRA_COST

Данный параметр представляет величину расхода, которая добавляется к расходу в режиме intra при декодировании в режиме макроблоков. Чем выше данное значение, тем более вероятно режим макроблоков станет режимом inter. Рекомендуется использовать значение 0 и не изменять его, если, конечно, не требуется полное понимание решения режима.

MBMode = InterCost <= (IntraCost + IntraCostWeight) ? Inter Mode : Intra Mode

ENTROPY_CODING_MODE

Режим энтропийного кодирования

0: CAVLC

1: CABAC

2: Переход на режим энтропийного кодирования

В режиме 2 I-слайсы кодируются по методу CAVLC, а P-слайсы – по методу CABAC. I-слайсам на кодирование требуется больше времени, если они кодируются по методу CABAC, чем в случае кодирования по методу CAVLC, потому что в I-слайсах, как правило, больше битов, чем в P-слайсах, а чем больше длина битов, тем больше времени требуется для CABAC. Вот почему оба режима энтропийного кодирования используются при разработке эффективных методов кодирования.

TRANSFORM_8X8_MODE

Режим преобразования в формат 8X8

0: отключено (BP)

1: включено (HP)

CABAC_INIT_IDC
cabac_init_idc в синтаксисе стандарта H.264
CHROMA_QP_OFFSET
Chroma_qp_offset в синтаксисе стандарта H.264.
PIC_QP_Y
Данное значение используется в качестве параметра квантизации в случае, если отключена опция контроля скорости передачи.

1.3.6.4.8.1.2 Фильтр деблокирования

DISABLE_DEBLK
Флаг фильтра деблокирования
0: Фильтр деблокирования включен.
1: Фильтр деблокирования отключен.
2: фильтр деблокирования отключен на границе слайса.
DEBLK_ALPHA
Slice_alpha_c0_offset_div2 в синтаксисе стандарта H.264.
DEBLK_BETA
Slice_beta_offset_div2 в синтаксисе стандарта H.264.

1.3.6.4.8.1.3 Структура слайса

SLICE_MODE
0: В изображении есть один слайс.
1: В изображении есть несколько слайсов.
SLICE_SIZE_MODE
Данным значением определяется то, как истолковывается регистр SLICE_SIZE_NUMBER.
0: В каждом слайсе содержится число SLICE_SIZE_NUMBER битов.
1: В каждом слайсе содержится число SLICE_SIZE_NUMBER макроблоков.
2: Число (SLICE_SIZE_NUMBER + 1) слайсов имеет собственное число макроблоков.
Например, если регистр SLICE_SIZE_MODE установлен в 2, а регистр SLICE_SIZE_NUMBER установлен в 7, то это означает, что есть 8 слайсов, и количество макроблоков для каждого слайса определяется с помощью регистра SLICE_SIZE_GROUP_0000, SLICE_SIZE_GROUP_0001 и т. д.
SLICE_SIZE_NUMBER
По сути, это размер слайса, но в регистре SLICE_SIZE_MODE определяется единица измерения данного значения.
SLICE_SIZE_GROUP_0000 ~ SLICE_SIZE_GROUP_0007
Если слайсы имеют разный размер (то есть, SLICE_SIZE_MODE=2 и SLICE_SIZE_NUMBER=7), то количество макроблоков в каждом слайсе может задаваться в регистре SLICE_SIZE_GROUP (максимально 8).
SLICE_SIZE_GROUP_0000 7
SLICE_SIZE_GROUP_0001 24
SLICE_SIZE_GROUP_0002 36
..
.. SLICE_SIZE_GROUP_0007 12

1.3.6.4.8.1.4 Контроль скорости передачи

Кодер CODA9 осуществляет контроль битовой скорости при корректировке QP (параметра квантизации) на следующих двух уровнях:

- Контроль QP на уровне кадра
- Контроль QP на уровне макроблока

Во-первых, для соответствия целевой битовой скорости, биты назначаются исходя из QP уровня изображения и параметров QP уровня рядов макроблоков. Во-вторых, QP для макроблока или группы макроблоков выбираются на основе измерения уровня сложности макроблока, при котором кодеру важно только качество визуального отображения.

Для контроля скорости передачи существует много подходящих опций, однако главных параметров всего несколько - RATE_CONTROL_ENABLE, BIT_RATE_KBPS и DELAY_IN_MS (если CBR) . Остальные параметры служат для ограничения экспоненциальных изменений QP, таких как мин./макс. qp, delta qp или gamma-значение. Эти параметры легко адаптируются к целевому приложению.

RATE_CONTROL_ENABLE

Флаг включения контроля скорости передачи

0: отключить контроль скорости передачи (VBR)

1: включить контроль скорости передачи с CBR

2: включить контроль скорости передачи с ABR.

Более подробное описание каждой из настроек приводится далее.

Значение 0 отвечает за кодирование с постоянным QP. Требует установления компоненты Y QP PIC._

Значение 1 отвечает за контроль скорости передачи и воспроизведение битового потока, согласующегося с HRD, в буфере закодированного изображения, не допуская его переполнения и недозаполнения. Подходит для потоковой передачи по сети и приложений отображения данных в беспроводной сети.

Значение 2 также как значение 1 отвечает за регулирование процесса генерирования битовой скорости, но при этом допускаются колебания битовой скорости во времени в диапазоне целевой битовой скорости. Данное значение позволяет сконцентрироваться на управлении качеством визуального отображения, а также на поддержании целевого диапазона битовых скоростей. Целесообразно использовать его в мультимедийных приложениях, использующих хранилища данных.

BIT_RATE_KBPS

Целевая битовая скорость, выраженная в кбит/с. Наиболее эффективно использовать данное поле, когда регистр RATE_CONTROL_ENABLE установлен в 1 или 2.

DELAY_IN_MS

Данным параметром указывается начальная задержка в буфере декодера, выраженная в миллисекундах (мс).

20 – 200 (мс): Кодирование с небольшим периодом ожидания

Более 200 (мс): Кодирование с нормальным периодом ожидания

Это важный параметр для кодирования с небольшим периодом ожидания, потому что он используется для определения объема буфера потока:

BufferSize = DELAY_IN_MS * BIT_RATE_KBPS

BufferSize – это значение, используемое для обработки целевой битовой скорости и для защиты CBP от переполнения или недозаполнения при использовании текущей полосы пропускания.

Данное поле эффективно только, если регистр RATE_CONTROL_ENABLE установлен в 1. По умолчанию установлено значение 200. При установлении значения 0 данный параметр сохраняет значение 200.

RC_INTERVAL_MODE

Включение режима изменения параметра квантизации (Qp).

0: QP изменяет каждый макроблок. (по умолчанию).

3: QP меняет каждую группу макроблоков. Количество макроблоков в группе назначается в регистре RC_MB_INTERVAL.

Установления данного параметра в 3 приводит к небольшому снижению качества визуального отображения, поскольку кодер определяет уровень сложности для каждого макроблока и выбирает параметр QP. Другие значения не действуют для данной опции.

RC_MB_INTERVAL

Количество макроблоков в группе в случае изменения QP. По умолчанию установлено значение 1.

CONST_INTRAQP_ENABLE

Данный параметр позволяет определить параметр квантизации для I-кадра.

RC_INTRA_QP

В данном регистре определяется параметр шага квантизации изображения кадра intra для процесса кодирования. Когда он задается, intra-изображения кодируются с постоянным QP, и он влияет на прогнозирование QP последующих P-кадров. Допустимым диапазоном данного параметра является:

– В режиме MPEG-4/H.263: 1 ~ 31

В режиме H.264/AVC: 0 ~ 51

Данное значение действует, только если регистр RATE_CONTROL_ENABLE установлен в значение 1 или 2 , и установлен регистр CONST_INTRAQP_ENABLE.

MAX_DELTA_QP_SET_ENABLE

Данный параметр показывает предел разницы в QP между текущим и предыдущим макроблоками в регистре MAX_DELTA_QP. Например, если параметр MAX_DELTA_QP установлен в значение 5, а изменение в QP между текущим и предыдущим макроблоком оказывается равным 19, то кодер вынужден установить 19, чтобы обеспечить значение 5.

Данное значение действует только, если между параметрами QP есть разница. Другими словами, если параметры QP одинаковые, данный параметр не работает.

min < |QP difference| < max

MAX_DELTA_QP

Данный параметр задает максимальное значение параметра delta квантизации в диапазоне от 0 до 51 (по умолчанию установлено значение 51).

Он работает, только когда включен регистр MAX_DELTA_QP_SET_ENABLE и когда он меньше значения MAX_QP.

MIN_DELTA_QP_SET_ENABLE

Данный параметр указывает предел разницы в QP между текущим и предыдущим макроблоками для регистра MIN_DELTA_QP. Например, если MIN_DELTA_QP установлен в 1, и изменение QP между текущим и предыдущим макроблоками оказывается равным 5, кодер вынужден установить 5, чтобы обеспечить значение 1.

Данное значение действует только, если между параметрами QP есть различие. Другими словами, если параметры QP одинаковые, данный параметр не работает.

min < |QP difference| < max

MIN_DELTA_QP

Данный параметр задает минимальное значение параметра delta квантизации в диапазоне от 0 до 51 (по умолчанию установлено значение 0).

Он работает, только когда включен регистр MIN_DELTA_QP_SET_ENABLE и когда он меньше значения MIN_QP.

MIN_QP_SET_ENABLE

Данный параметр позволяет настроить минимальные значения в наборе параметров квантизации в файле конфигурирования.

MIN_QP

Данный параметр позволяет задать минимальное значение параметра квантизации в диапазоне от 0 до 51 (по умолчанию установлено значение 0).

Примечание Значение регистра MIN_QP не должно превышать значения регистра MAX_QP.

MAX_QP_SET_ENABLE

Данный параметр позволяет настроить максимальные значения в наборе параметров квантизации в файле конфигурирования.

MAX_QP

Данный параметр позволяет задать максимальное значение параметра квантизации в диапазоне от 0 до 51 (по умолчанию установлено значение 51).

GAMMA_SET_ENABLE

Данный параметр обеспечивает возможность настройки гамма в файле конфигурирования.

ГАММА

Данный параметр определяет значение гаммы, указывающее скорость настройки битовой скорости в зависимости от полноты буфера как параметра реагирования. Значение изменяется в диапазоне от 8 до 64.

Чем больше значение гаммы, тем более плавно настраивается битовая скорость для соответствия целевому биту в буфере. По умолчанию установлено значение гамма, равное 32. Данный параметр не работает, если регистр RATE_CONTROL_ENABLE установлен в 1 (CBR), а значение регистра DELAY_IN_MS для приложений с малой задержкой ниже 200 мс.

1.3.6.4.8.1.5 Сообщение VUI

Кодер CODA9 поддерживает некоторые параметры информации по удобству использования видеоданных (VUI) для приложений, определенных в Приложении E спецификации H.264/AVC.

VUI_PARAMETERS_PRESENT_FLAG

vui_parameters_present_flag в синтаксисах параметров VUI в формате H.264/AVC

VIDEO_SIGNAL_TYPE_PRESENT_FLAG

video_signal_type_present_flag в синтаксисах параметров VUI в формате H.264/AVC

VIDEO_FORMAT

video_format в синтаксисах параметров VUI в формате H.264/AVC

VIDEO_FULL_RANGE_FLAG

video_full_range в синтаксисах параметров VUI в формате H.264/AVC

COLOUR_DESCRIPTION_PRESENT_FLAG

colour_description_present_flag в синтаксисах параметров VUI в формате H.264/AVC

COLOUR_PRIMARIES

colour_primaries в синтаксисах параметров VUI в формате H.264/AVC

TRANSFER_CHARACTERISTICS

transfer_characteristics в синтаксисах параметров VUI в формате H.264/AVC

MATRIX_COEFFICIENTS

matrix_coefficients в синтаксисах параметров VUI в формате H.264/AVC

Примечание Подробное описание данных параметров представлено в Приложении E спецификации H.264/AVC.

1.3.6.4.8.1.6 QMAT для пользователей

Существуют различные флаги, определяемые по таблице qmatrix, которые применяются пользователями:

SEQ_SCALING_MATRIX_PRESENT
SEQ_SCALING_LIST_PRESENT
SEQ_SCALING_LIST_DEFAULT
PIC_SCALING_MATRIX_PRESENT
PIC_SCALING_LIST_PRESENT
PIC_SCALING_LIST_DEFAULT

Ниже приведена конфигурация масштабирующей матрицы, заданная по умолчанию.

SEQ_SCALING_MATRIX_PRESENT ./qmat.cfg SEQ_SCALING_LIST_PRESENT 0
SEQ_SCALING_LIST_DEFAULT 255

Схема использования пользователями таблицы масштабирующей матрицы.

SEQ_SCALING_MATRIX_PRESENT ./qmat.cfg SEQ_SCALING_LIST_PRESENT 255

SEQ_SCALING_LIST_DEFAULT 0

Bitfield of xxx_SCALING_LIST_PRESENT and xxx_SCALING_LIST_DEFAULT {intra4x4 Y, intra4x4 Cb, intra4x4 Cr, inter4x4 Y, inter4x4 Cb, inter4x4 Cr, intra8x8 Y, inter8x8 Y}

Конфигурация масштабирующей матрицы PPS такая же, как одной из матриц SPS.

1.3.6.4.8.2 Пример настройки параметров для различных приложений

1.3.6.4.8.2.1 Видеоконференцсвязь

Видеоконференцсвязь - это двусторонняя интерактивная дистанционная передача данных, в ходе которой аудио- и видеоданные передаются одновременно с обоих направлений. Для такого рода приложений реального времени, чувствительным к задержкам, кодер на низкой битовой скорости должен сжимать кадры в некоторый постоянный объем битов в секунду с возможностью обеспечения устойчивости к ошибкам. Кроме того, на кодирование при этом тратится меньше времени, чем в других случаях, что позволяет добиться высокой эффективности данного процесса в реальном масштабе времени.

Рекомендуемые значения параметров кодирования для обеспечения видеоконференцсвязи определяются в приведенной ниже таблице.

Таблица 1.570 – Рекомендуемые параметры для настройки видеоконференцсвязи

Параметры	Рекомендуемое значение	Примечания
GOP_PIC_NUMBER	0	
SLICE_MODE	1	
SLICE_SIZE_MODE SLICE_SIZE_NUMBER	0 8 000 ~ 11 200 битов	
INTRA_MB_REFRESH CONSTRAINED_INTRA	1 1	
SEARCH_RANGE_X	3	
SEARCH_RANGE_Y ENTROPY_CODING_MODE	2 0	
TRANSFORM_8x8_MODE	0	Рекомендуется к использованию во всех случаях кроме случаев, когда эффективность кодирования с 30 к/с в реальном времени в системе SoC
RATE_CONTROL_ENABLE BIT_RATE_KBPS	1 64 ~ 4 000	QCIF - 64 кбит/с @ 15 к/с / 128 кбит/с @ 30 к/с CIF - 256 - 384 кбит/с @ 15 к/с / 384 - 512 кбит/с @ 30 к/с • D1 - 1000 - 1500 кбит/с @ 30 к/с • 720p - 2000 - 4000 кбит/с @ 30 к/с
DELAY_IN_MS	20 ~ 200	Значение меньше 200 мс обеспечивает возможность управления скоростью при кодировании со сверхнизким периодом ожидания. (При значении, равном или превышающем 200 мс, используется другой алгоритм управления скоростью.)

1.3.6.4.8.2.2 Потоковое интернет-видео

Потоковое интернет-видео – это передача аудио и видеоконтекта по сети пользователям. На потоковом сервере находится заранее записанный контент, который и распространяется пользователям. В сервисах такого типа для кодирования с постоянной битовой скоростью в формат высококачественных потоковых данных с инструментами обеспечения устойчивости к ошибкам требуется кодер, потому что важно, чтобы вещание потокового видео не прерывалось и не «зависало» в середине воспроизведения. Кроме того, критичным для зрителя является также фактор соблюдения одинакового на протяжении всего вещания качества изображения. Что касается таких требований, как низкий период ожидания и длительное время кодирования, то в данном случае они являются не критичными.

Примеры значений параметров кодера для приложения потокового вещания определяются по таблице 1.571.

Таблица 1.571 – Рекомендуемые параметры для обеспечения потокового интернет-видео

Параметры	Рекомендуемое значение	Примечания
GOP_PIC_NUMBER	15-30	
SLICE_MODE	1	
SLICE_SIZE_MODE	0	
SLICE_SIZE_NUMBER	8 000 ~ 11 200 битов	
INTRA_MB_REFRESH	1	
CONSTRAINED_INTRA	1	
SEARCH_RANGE_X	0	
SEARCH_RANGE_Y	0	
ENTROPY_CODING_MODE	1	
TRANSFORM_8x8_MODE	1	
RATE_CONTROL_ENABLE	1	
BIT_RATE_KBPS	64 ~ 4 000	• 240p - 500 кбит/с @ 30 к/с • 480p - 800 - 1000 кбит/с @ 30 к/с • 720p - 2000 - 2900 кбит/с @ 30 к/с • 1080p - 3000 - 4300 кбит/с @ 30 к/с
DELAY_IN_MS	500	Более 200

1.3.6.4.8.2.3 Хранение данных

В качестве приложения по хранению данных может выступать любое цифровое ЗУ или диск ЗУ, установленный в устройстве, таком как DVR или портативный медиаплейер, которое может записывать видеоинформацию в цифровом формате и воспроизводить ее. В приложении по хранению данных для кодера не играют важной роли такие параметры, как ширина полосы пропускания сети, период ожидания и время кодирования, потому что нет передачи данных и нет задач, которые необходимо решать в реальном масштабе времени. Но такие факторы как высокое качество, условный контроль битовой скорости и кодирование с возможностью обеспечения устойчивости к ошибкам, являются важными при конфигурировании.

Для приложений такого типа значения параметров кодера определяются по таблице 1.572.

Таблица 1.572 – Рекомендуемые параметры для обеспечения потокового интернетвещания

Параметры	Рекомендуемое значение	Примечания
GOP_PIC_NUMBER	30 (1 сек.)	
SLICE_MODE	0	
SLICE_SIZE_MODE	0	
SLICE_SIZE_NUMBER		
INTRA_MB_REFRESH	0	
CONSTRAINED_INTRA	0	
SEARCH_RANGE_X	0	
SEARCH_RANGE_Y	0	
ENTROPY_CODING_MOD E	1	
TRANSFORM_8x8_MODE	1	
RATE_CONTROL_ENABLE	2	
BIT_RATE_KBPS	128 ~ 15000	QCIF - 128 кбит/с @15 к/с / 256 кбит/с @ 30 к/с CIF - 512 - 1000 кбит/с @ 30 к/с • D1 - 4000 - 8000 кбит/с @ 30 к/с • 720p - 4000 - 15000 кбит/с @30 к/с
	20000 ~ 40000	Стандарт Blu-ray
Параметры	Рекомендуемое значение	Примечания
DELAY_IN_MS	500	Более 200

1.3.6.4.9 КОДИРОВАНИЕ СО СВЕРХНИЗКИМ ПЕРИОДОМ ОЖИДАНИЯ

Ввиду того, что на сегодняшний день интерактивные приложения мультимедийной передачи данных по беспроводной сети, такие как видео-игры и видео-повторы, а также видеоконференцсвязь получают широкое распространение, кодирование с малой задержкой становится все более важной задачей, которая ставится перед системами передачи данных в реальном времени.

Кодер IP-видео H.264/AVC CODA980 Chips&Media характеризуется сверхмалой задержкой при кодировании, что дает возможность работы с некоторыми допустимыми опциями кодирования, такими как поддержка интерфейса синхронизации подкадра с ISP, тщательное контролирование скорости для кодирования на низкой битовой скорости и сигнализация SliceInfo для более быстрой передачи данных. Данный кодер соответствует требованиям, предъявляемым к обработке данных в реальном времени и, одновременно с этим, способен гарантировать высокое качество отображения в приложениях, чувствительных к задержкам.

В данном документе рассматриваются различные типы задержек, возникающих в процессе передачи видеоданных, а также описываются методы управления сверхнизкими задержками и минимизирования таких задержек.

1.3.6.4.9.1 Типы задержек в режиме сквозной передачи

В стандартной системе передачи сжатых видеоданных есть три основных источника задержки1: период ожидания при кодировании, период ожидания при передаче и период ожидания в работе декодера, проиллюстрированные на рисунке 1.173.

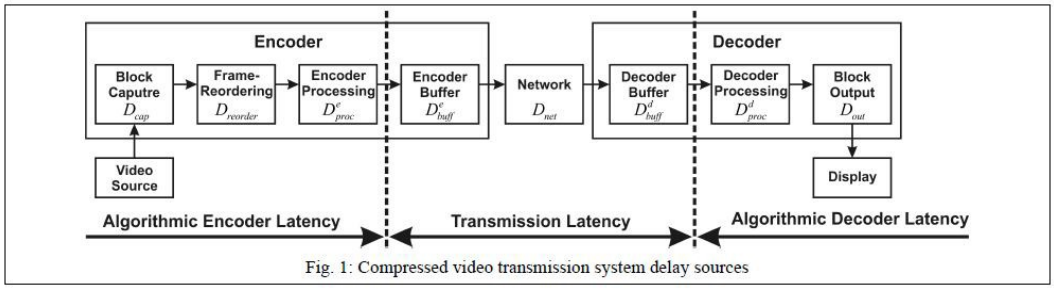


Fig. 1: Compressed video transmission system delay sources

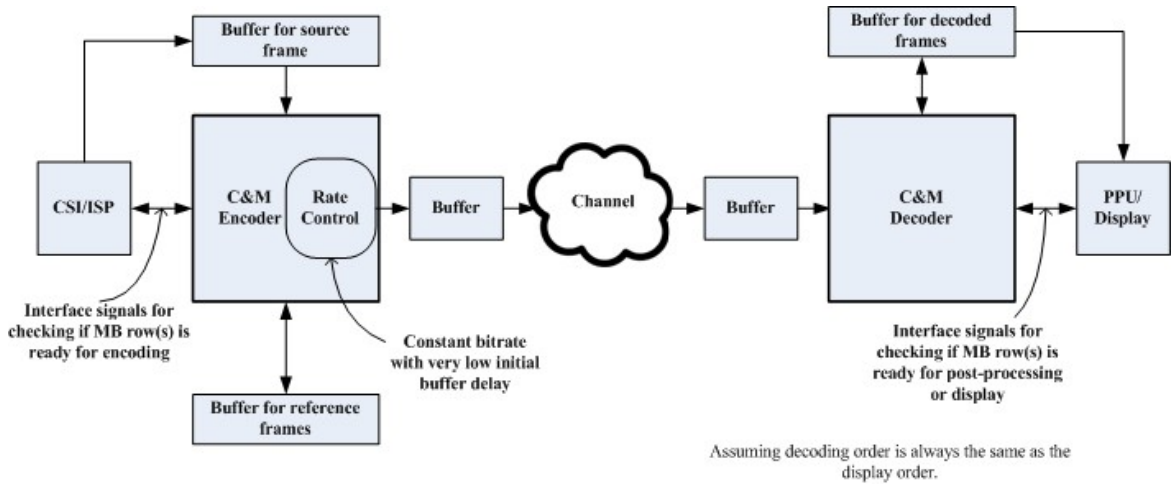
Encoder – Кодер
Decoder – Декодер
Block Capture – Захват блока
Frame-Reordering – Переупорядочение кадров
Encoder Processing – Обработка декодера
Encoder Buffer – Буфер кодера
Network – Сеть
Decoder Buffer – Буфер декодера
Decoder Processing – Обработка декодера
Block Output – Вывод блока
Video Source – Источник видеоданных
Display – Вывод на экран
Algorithmic Encoder Latency – Алгоритмический период ожидания в кодере
Transmission Latency – Период ожидания при передаче данных
Algorithmic Decoder Latency – Алгоритмический период ожидания в декодере

Рисунок 1.173 – Типы задержки при передаче сжатых видеоданных

Для захвата последовательности входных видеокадров, для переупорядочения (в случае наличия B-изображения), а также для сжатия кадров кодеру требуется время. В периоде ожидания при передаче данных учитывается время на буферизацию сжатых видеоданных и в кодере, и в декодере, а также непосредственно время передачи. Декодеру также требуется время на распаковку видеоданных и их обратной их записи для последующего отображения. Суммарное время, затрачиваемое на весь процесс, начиная с захвата блока и заканчивая выводом на экран, называют, как правило, сквозной задержкой.

1.3.6.4.9.2 Кодирование с низкой задержкой

На рисунке 1.174 - “Методы кодирования со сверхнизкой задержкой” представлена схема кодирования со сверхнизкой задержкой, используемая в блоках VPU (блок обработки видеосигналов), разработанном компанией “Chips&Media”.



Buffer for source frame – Буфер для исходного кадра
Buffer for decoded frames – Буфер для декодированных кадров
C&M Encoder – Кодер C&M
Rate Control – Контроль скорости
Buffer – Буфер
Channel – Канал
Buffer – Буфер
C&M Decoder – Декодер C&M
PPU/Display – PPU/Отображение
Interface signals for checking if MB row(s) is ready for encoding – Интерфейсные сигналы для проверки готовности ряда(ов) макроблоков к кодированию
Buffer for reference frames – Буфер для ссылочных кадров
Constant bitrate with very low initial buffer delay – Постоянная битовая скорость с низкой задержкой в буфере исходных данных
Interface signals for checking if MB row(s) is ready for post-processing or display – Интерфейсные сигналы для проверки готовности ряда(ов) макроблоков к постобработке или выводу на экран
Assuming decoding order is always the same as the display order – Предполагается, что порядок декодирования всегда совпадает с порядком отображения

Рисунок 1.174 – Методы кодирования со сверхнизкой задержкой

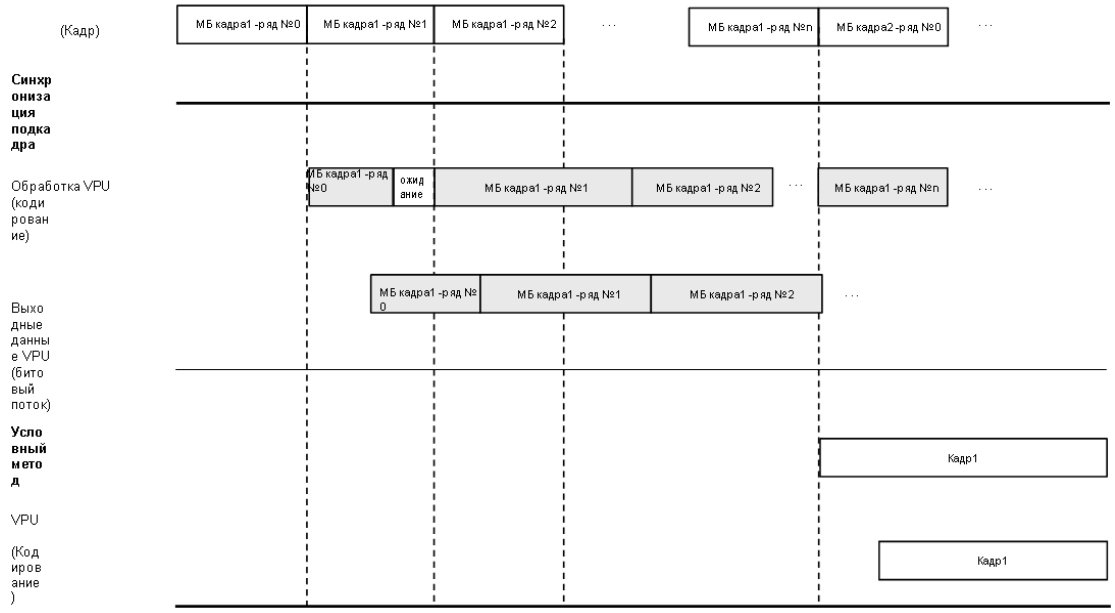
В случае кодирования с низкой задержкой VPU может кодировать видеоданные формата H.264/AVC, размером 1080п, скоростью 60 к/с и на частоте 266 МГц до 50 Мбит/с с CHP (Ограниченным высоким профилем) или CBP (Ограниченным базовым профилем) формата H.264/ AVC.

Благодаря интерфейсным сигналам, передающимся от CSI/ISP, VPU может начать работать сразу, как только будет подготовлен определенный минимальный объем данных. Блок VPU кодирует исходные кадры с контролем скорости и генерирует постоянную битовую скорость для буфера, чтобы не допустить переполнения или недозаполнения этого буфера при передаче видеоданных в реальном масштабе времени. Кроме того, регулирование параметров кодирования поможет снизить задержки при кодировании.

Примечание Кодер H.264/AVC CODA980 согласуется почти со всеми инструментами Ограниченного высокого профиля Miracast 2. Однако в нем не поддерживается режим чересстрочного кодирования, а поддерживается режим энтропийного кодирования CABAC.

Синхронизация подкадра

В процессе считывания исходных кадров из входного буфера, VPU с помощью специальных аппаратных сигналов готовности подкадра из модуля источника видеоданных или посредством настройки соответствующих регистров, может начать обработку части кадра - ряда макроблоков, чтобы ускорить ход процесса. Данная функция называется синхронизацией подкадра в VPU. На рисунке 1.175 показаны входные данные в рядах макроблоков кодера.



(Frame) – (Кадр)
Sub-frame Synchronization – Синхронизация подкадров
VPU Processing (Encoding) – Обработка VPU (кодирование)
VPU Output (Bitstream) – Выходные данные VPU (битовый поток)
Conventional Method – Условный метод
VPU (Encoding) – VPU (Кодирование)
(Bitstream) - (Битовый поток)

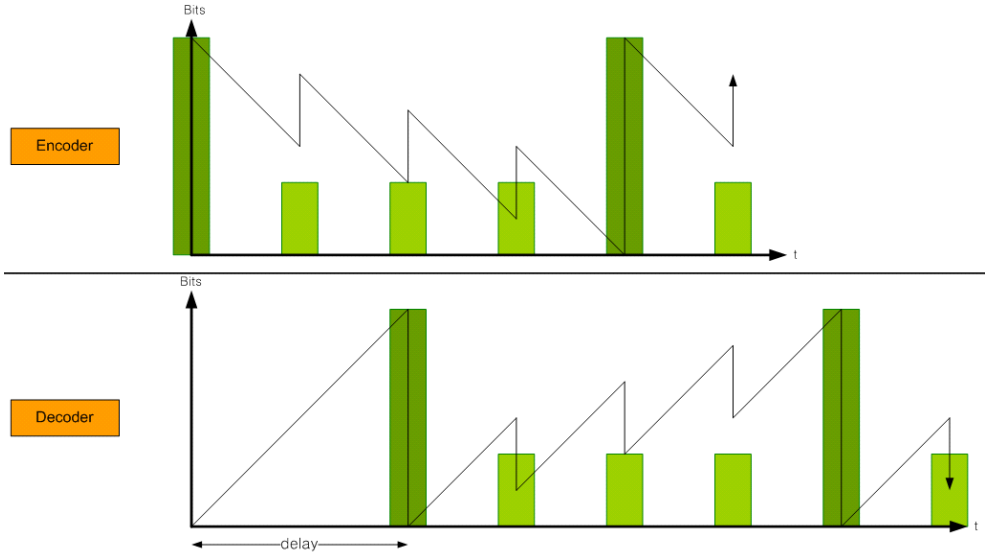
Рисунок 1.175 – Синхронизация подкадра

Кодер VPU может начать процесс кодирования сразу, как только данные одного ряда макроблоков поступят во входной буфер, в то время как в условном методе блок VPU ожидает, пока в буфер не поступят все биты кадра. По завершении обработки закодированный битовый поток сбрасывается на каждые 512 байтов. Синхронизация подкадров позволяет начать работу раньше и сократить период ожидания при буферизации источника кадра.

Примечание Более подробную информацию о синхронизации кадра можно найти в «Техническом описании CODA980».

1.3.6.4.9.3 Контроль скорости

При кодировании последовательности видеокадров кадров, как правило, имеют разную избыточность, и поэтому количество битов, генерируемых в расчете на один кадра, безусловно, изменяется. На рисунке 1.176 проиллюстрировано то, как по мере изменения кадров меняются закодированные биты, а также то, что вызывает в декодере более длительную начальную задержку.



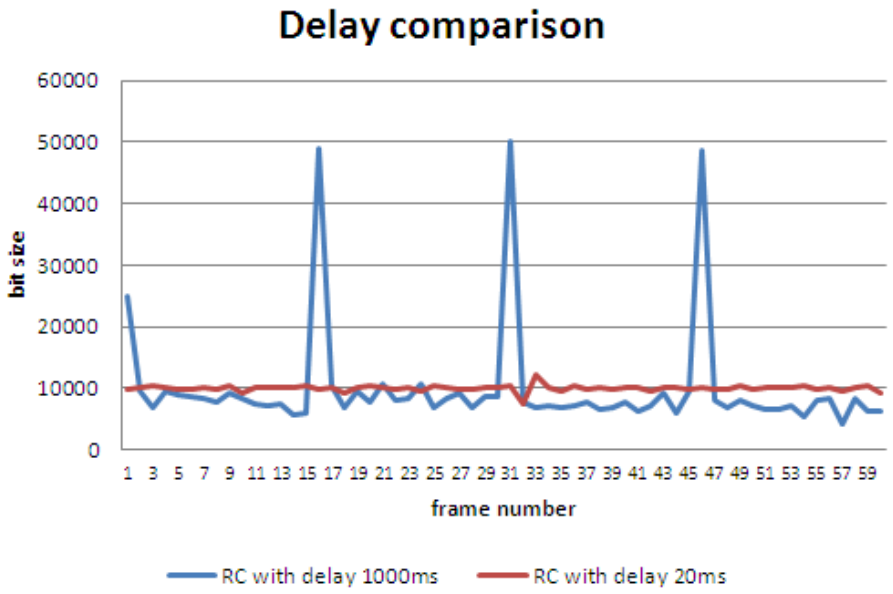
Encoder – Кодер
Decoder – Декодер
Bits – Биты
Bits – Биты
Delay - Задержка

Рисунок 1.176 – Отклонение битовой скорости

Как видно из рисунке 1.174 - “Методы кодирования со сверхнизкой задержкой”, между кодером и каналом передачи данных располагается буфер. Для выравнивания изменяющейся скорости выходных данных в нем собирается определенное количество битов, подлежащих передаче. Объем буфера зависит от битовой скорости в канале и разрешенной приложением исходной задержки.

В приложении с низкой задержкой закодированный битовый поток должен передаваться по каналу CBR за короткий интервал времени с еще более низкой исходной задержкой, следовательно, для этого требуется совсем небольшой буфер. Чтобы воспроизводить постоянный объем битов и не допускать переполнения или недозаполнения буфера в ходе процесса кодирования следует проводить контроль скорости.

На рисунке 1.177 представлен сравнительный анализ битовых скоростей при контроле скорости с разными задержками.



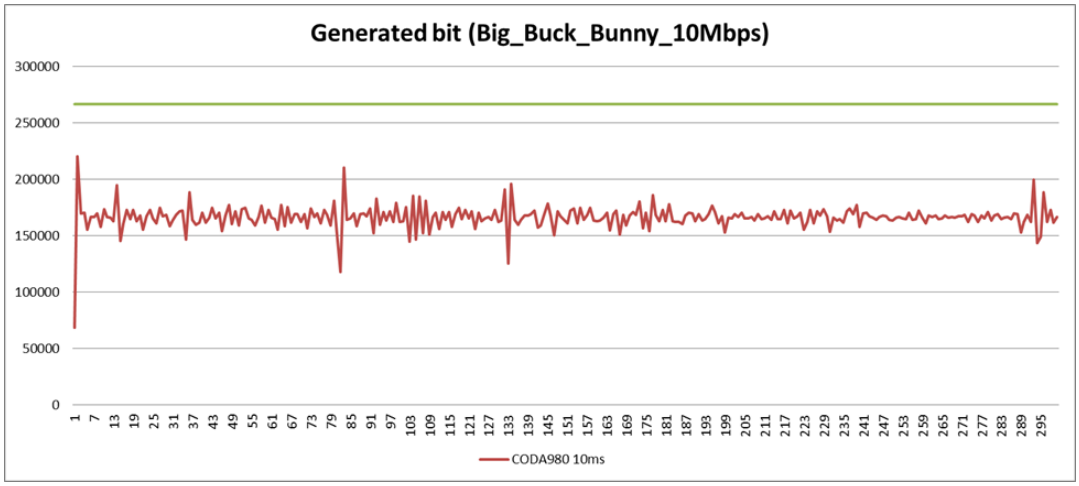
Delay comparison – Сравнительный анализ величины задержки
Bit size – Размер бита
Frame number – Число битов
RC with delay 1000mc – RC с задержкой 1000 мс
RC with delay 20ms - RC с задержкой 20 мс

Рисунок 1.177 – Сравнительный анализ величины задержки

Установка времени задержки оказывает влияние на объем генерируемых VPU битов в расчете на один кадр. Фактически, низкая задержка указывает на использование небольшого буфера, и в этом случае кадры битов усредняются на очень короткое время при изменении QP. Из приведенного выше рисунка понятно, что контроль скорости с задержкой 20 мс показывает фактическую битовую скорость каждого кадра, в то время как, контроль скорости с задержкой 1000 мс показывает битовую скорость с отклонением, внезапные всплески на определенных кадрах.

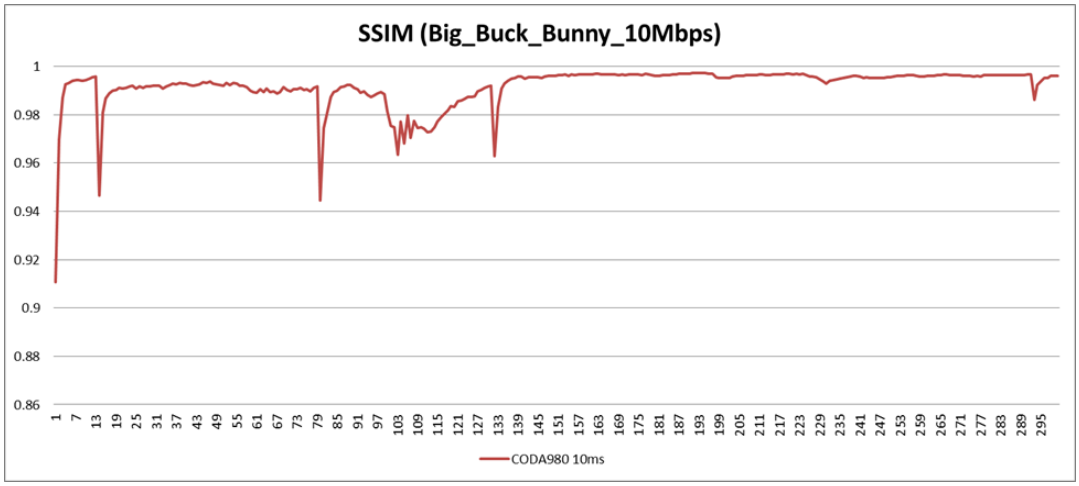
Эффективность контроля скорости

Кодер H.264/AVC CODA980 позволяет осуществлять контроль скорости на уровне макроблоков, прежде всего, в целях соответствия целевой битовой скорости. На нижеследующем рисунках 1.178 и 1.179 представлены результаты измерения битовой скорости SSIM 3 при кодировании последовательности видеоданных big buck bunny 1080p при 10 Мбит/с с включением функции контроля скорости.



Generated bit (Big_Buck_Bunny_10Mbps) – Сгенерированный бит (Big_Buck_Bunny_10Mbps)
CODA980 10ms – CODA980 10 мс

Рисунок 1.178 – Контроль битов кадра при 10 Мбит/с



CODA980 10ms – CODA980 10 мс

Рисунок 1.179 – Качество отображения кадров при включении контроля скорости

Из графика на рисунке 1.178 можно определить, что сгенерированные биты для каждого кадра являются постоянными и всегда располагаются существенно ниже максимальной битовой скорости, даже, несмотря на то, что наблюдаются некоторые незначительные колебания битов при изменении числа кадров. Кроме того на рисунке 1.179 представляет собой график качества отображения кадров, на котором представлена субъективная динамика изменения качества. В целом, график для большинства фрагментов выглядит плавным, за исключением нескольких «провалов» после смены сцен.

Примечание Дополнительную информацию о том, как установить регистра контроля скорости и API, можно найти в предоставляемом Руководстве для программистов и Справочном руководстве по API.

.

1.3.6.4.9.4 Сигнализация SliceInfo

VPU в целях маскирования ошибок в среде передачи видеоданных, где в ходе передачи битового потока по сети может возникнуть пакетная ошибка или потеря пакета, может закодировать кадр в несколько слайсов битового потока.

Как видно из рисунка 1.173 существует некоторая задержка буферизации битового потока перед его отправкой в сеть, расположенную между кодером и каналом передачи (сеть). Чтобы обработать и сократить период ожидания при буферизации из кодера, существует возможность передачи из VPU в буфер информации о слайсе, включающей в себя указание количества слайсов, указатель входа в битовом потоке для слайса, а также его размер в байтах, чтобы можно было посылать с хоста слайс битового потока без необходимости ожидать весь битовый поток данного кадра.

Уведомление с информацией о слайсе передается посредством прерывания или в регистре главного интерфейса всякий раз, когда завершается кодирование слайса. VPU сбрасывает счетчик слайсов в начале кодирования каждого нового кадра. Такой блок слайсов сигнализации для приложений, чувствительных к задержкам, позволяет передавать видеоданные чуть быстрее, и поэтому декодер может раньше приступить к послайсовой обработке.

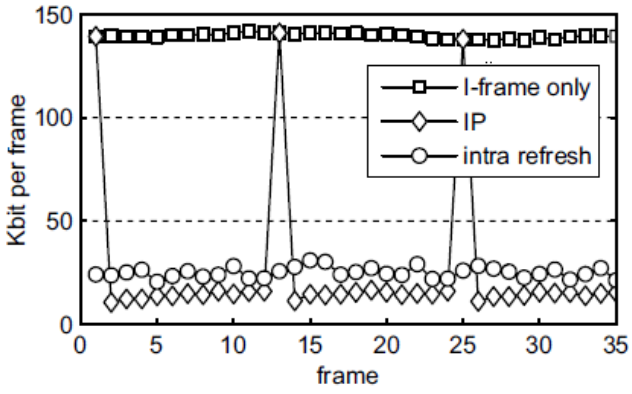
1.3.6.4.9.5 Инструменты кодирования

Чтобы добиться снижения периода ожидания первое и самое важное, что следует сделать - это запустить блок VPU в оптимальном режиме кодирования. У VPU много опций, которые определяют и влияют на поведение кодера и на величину периода ожидания. Для соответствия требованию в отношении низкой задержки, VPU необходимо сконфигурировать с использованием некоторых релевантных с точки зрения эффективности и качества кодирования параметров кодирования.

Intra-регенерация

При возникновении ошибок только у первых закодированных битовых потоков I-кадра проявляются слабые стороны. Если при передаче теряется или разрушается часть пакетов, особенно тех, которые находятся в начале последовательности видеоданных, то ошибка в I-изображении вероятнее всего будет распространяться и на декодирование, потому что последующие кадры будут на нее ссылаться. Поэтому наиболее безопасным для VPU будет добавить закодированный Intra-макроблок в течение очень короткого интервала регенерации, чтобы скрыть макроблоки с ошибками или пропущенные макроблоки (если они есть).

На рисунке 1.180 показаны варианты битовой скорости для стандартной последовательности видеоданных с использованием трех разных методов кодирования.



I-frame Only – Только I-кадр
Intra refresh –Регенерация Intra
Kbit per frame – кбит/кадр
Frame - Кадр

Рисунок 1.180 – Регенерация Intra

На рисунке 1.180 показано, что в результате кодирования только I-кадра воспроизводится значительное количество битов, но все они являются постоянными, в то время как при

кодировании IP производится существенно меньшее количество битов, но они имеют сильные всплески скорости на I-кадрах. Между тем, для регенерации Intra желательно иметь достаточно небольшой объем битв при относительно постоянной битовой скорости, в расчете на один кадр.

Чтобы с помощью временного прогнозирования добиться более высокого коэффициента сжатия, рекомендуется вставить в прогнозируемые кадры закодированный макроблок intra. Такая принудительная регенерация intra-блока поможет обеспечить отображение видео без ошибок в течение короткого интервала времени, а также даст возможность осуществлять передачи с низкой задержкой при относительно постоянной битовой скорости в расчете на один кадр.

Кроме того, в некоторых приложениях, в которых предоставляется полоса пропускания достаточной ширины, возможно кодирование с использованием только I-кадра. Была проведена оценка эффективности кодирования 1080п 30 к/с при 50 Мбит/с и 100 Мбит/с со всеми intra-блоками и с задержкой 10 мс. Она показала, что частота 129 МГц и 143 МГц дает усредненное количество циклов, требуемое для кодирования в реальном масштабе времени при 50 Мбит/с и 100 Мбит/с, соответственно.

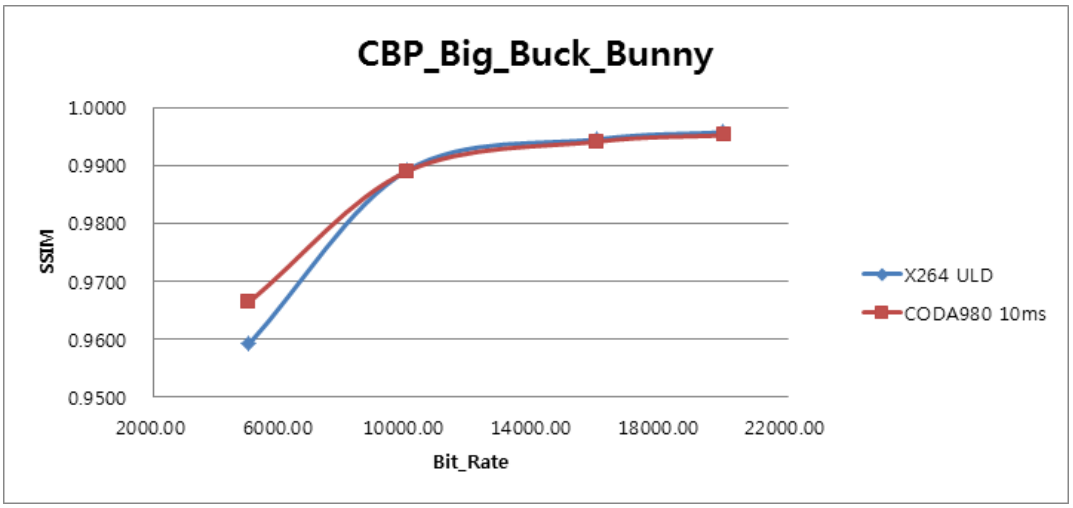
Ограниченный Базовый/Высокий профиль для I/P

При передаче видео с низкой задержкой следует использовать только упреждающее временное прогнозирование (I- и P-кадры), поскольку двунаправленное прогнозирование (B-кадры) приводит к задержкам в ходе переупорядочения кадров, при их кодировании, последующем декодировании и выводе на экран.

На рисунках 1.182 и 1.183 представлены графики сравнения качества изображения в случае, если видеофайл «Big Buck Bunny», имеющий разрешение 1080p, кодируется с использованием профилей CBP 5 и CHP 6 CODA980, которые не составляют B-изображения.

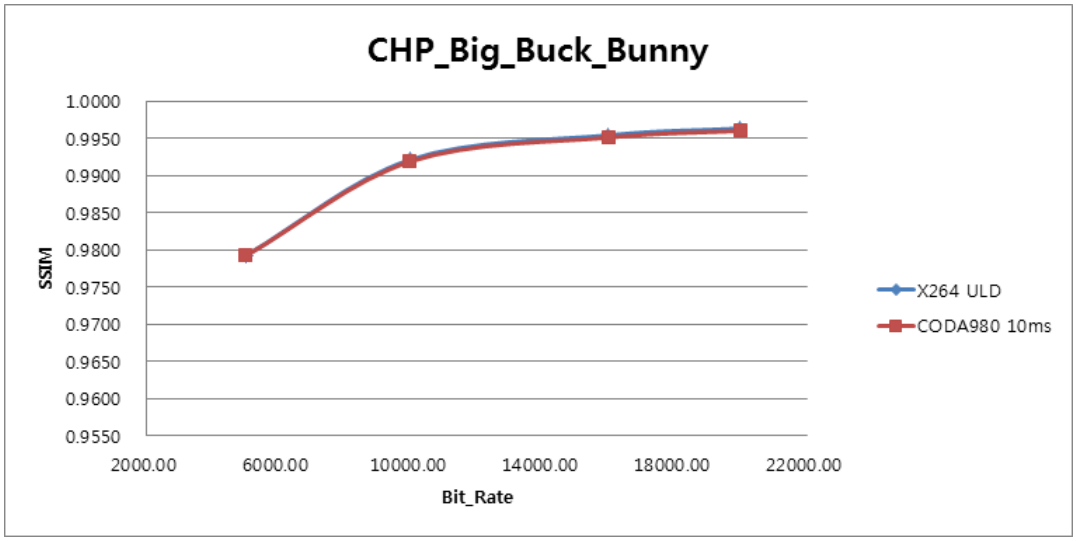


Рисунок 1.181– Исходная последовательность видеоданных «Big Buck Bunny 1080p»



10ms – 10 мс

Рисунок 1.182 – Качество видеоданных, закодированных с использованием профиля CBP H.264/AVC CODA980



10ms – 10 мс

Рисунок 1.183 – Качество видеоданных, закодированных с использованием профиля CHP H.264/AVC CODA980

На рассмотренных выше графиках показаны результаты SSIM в одинаковых точках контроля битовой скорости, а также в сравнении с ULD X264. 7 На рисунке 1.182 проиллюстрировано то, что при низких битовых скоростях, таких как скорости в диапазоне от 6000 бит/с до 10000 бит/с профиль CBP H.264/AVC CODA980 с задержкой 10 мс (при включенном контроле скорости) обладает более высоким качеством, чем обеспечивается ULD X264. При скорости превышающей 10000 бит/с их качество становится приблизительно одинаковым.

С другой стороны на рисунке 1.183 показано, что кривая графика более слаженная, и ее общее SSIM выше, чем на рисунке 1.182 при 6000 бит/с - выше почти на 0,0200, и изменение SSIM профиля CHP H.264/AVC полностью совпадает с кривой ULD X264.

1.3.6.4.9.6 Изменение параметра битовой скорости

Статус канала сети связи зачастую изменяет битовую скорость, в особенности, в среде беспроводной связи, такой как Miracast. В таком приложении скорость передачи битов в канале контролируется и передается в кодер. Поэтому, если канал связи находится в плохом состоянии или буфер близок к заполнению, битовая скорость снижается, и кодер может воспроизвести несколько битов одновременно.

Чтобы справиться с таким изменением битовой скорости VPU поддерживает функцию изменения параметров и может обеспечить возможность кодирования кадров с одновременным динамическим контролем битовой скорости (если эта функция установлена). Функция изменения параметров включается посредством настройки некоторых соответствующих регистров главного интерфейса.

Примечание Описание соответствующих регистров можно найти в Справочном руководстве к API CODA980.

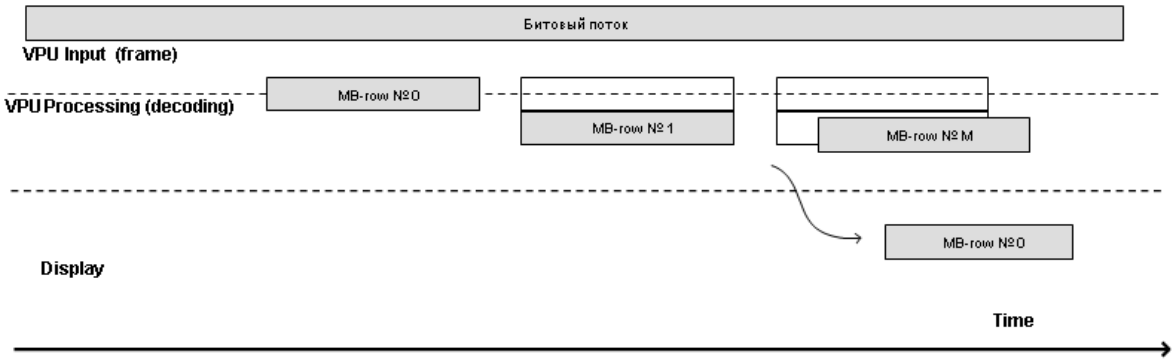
1.3.6.4.9.7 Декодирование с низкой задержкой

1.3.6.4.9.7.1 Сигнализация декодированных рядов макроблоков

Чтобы минимизировать период ожидания на стороне декодера, существует способ передачи информации о выполнении декодирования ряда макроблоков, поэтому, как только такая информация поступит на хост, он может начать выводить на экран ряд макроблоков изображения, не тратя время на ожидание записи всего кадра.

Ниже представлен пример конфигурации CBP X264 в условиях сверхнизкой задержки. Для моделирования ситуации используются следующие опции и команды кодирования: x264.exe --bframes 0 --partitions "p16x16, p8x8, i16x16, i4x4" --vbv-bufsize 1 --bitrate 5000 --me hex --trellis 0 --ref 1 --frames 300 --fps 30 --nal-hrd cbr --tune zerolatency -o output_stream.264 input_yuv --input-res 1920x1080 --keyint 60 --no-cabac --threads 1 --no-8x8dct --constrained-intra --intra-refresh --no-scenecut

На рисунке 1.184 показано, как осуществляется обработка данных ряда макроблоков в VPU и отображение данных ряда макроблоков при снижении периода ожидания в процессе их отображения.



VPU Input (Frame) – Вход VPU (кадр)
MB-row № 1 – Ряд МБ № 1
MB-row № 2 – Ряд МБ № 2 -
VPU Processing (Decoding) – Обработка VPU (Декодирование)
Display Time – Время отображения
MB-row № M – Ряд МБ № M
MB-row № 0 – Ряд МБ № 0

Рисунок 1.184 – Декодирование в условиях сверхнизкой задержки

Функционирование блока VPU осуществляется за счет обработки макроблоков. Как только в реальном времени начнется выведение изображения на экран, VPU сможет декодировать один кадр менее чем за 33 мс. Однако в VPU к каждому ряду макроблоков может применяться

отдельный цикл обработки, поскольку и битовая скорость, и степень сложности кодирования макроблоков будут неизбежно меняться. При декодировании определенного числа рядов макроблоков VPU генерирует и передает на хост сигнал прерывания с указанием адреса буфера декодированного на текущий момент кадра, а это означает, что декодированные ряды макроблоков кадра готовы к отображению. Вызов этой функции и количество рядов макроблоков (пороговое значение) конфигурируются в расчете на каждый канал (поток данных декодирования).

Примечание Информацию, касающуюся соответствующих регистров и настроек, можно найти в Руководстве для программистов.

1.3.6.4.9.8 Маскирование ошибок

Как правило, приложения с низкой задержкой наиболее подвержены ситуациям возникновения ошибок из-за сбоев в канале или каких-либо проблем, связанных со статусом буфера. VPU может обнаружить ошибку, связанную с недостаточным для декодирования кадра объемом битового потока, а также с ошибкой в самом битовом потоке. VPU в соответствии со схемой маскирования ошибок может воспроизвести последующие макроблоки для тех фрагментов, в которых были обнаружены потерянные или неправильные биты.

1.3.7 Комплект СФ-блока аудиоконтроллера

1.3.7.1 Описание и работа аудиоконтроллера

1.3.7.1.1 Назначение аудиоконтроллера

Аудиоконтроллер рассчитан на применение в составе СнК с системной шиной AMBA 3.0 (AXI) высокой степени интеграции и может использоваться в мультимедийных устройствах, предназначенных для воспроизведения звука.

Аудиоконтроллер реализует обмен данными между центральным процессором и внешним приемником цифровой звуковой информации по интерфейсам SPDIF или I2S. Аудиоконтроллер предназначен для вывода одноканального и многоканального (до 8-ми каналов при использовании интерфейса I2S) несжатого цифрового аудиосигнала и сжатого многоканального аудиосигнала (при использовании интерфейса SPDIF).

На вход аудиоконтроллера поступают массивы цифровых отсчетов аудиосигнала, которые преобразуются в последовательные битовые потоки в соответствии со спецификацией протокола I2S или SPDIF. Данные битовые потоки подаются на оконечные устройства, выводящие звук.

На рисунке 1.185 представлена схема типового применения аудиоконтроллера.

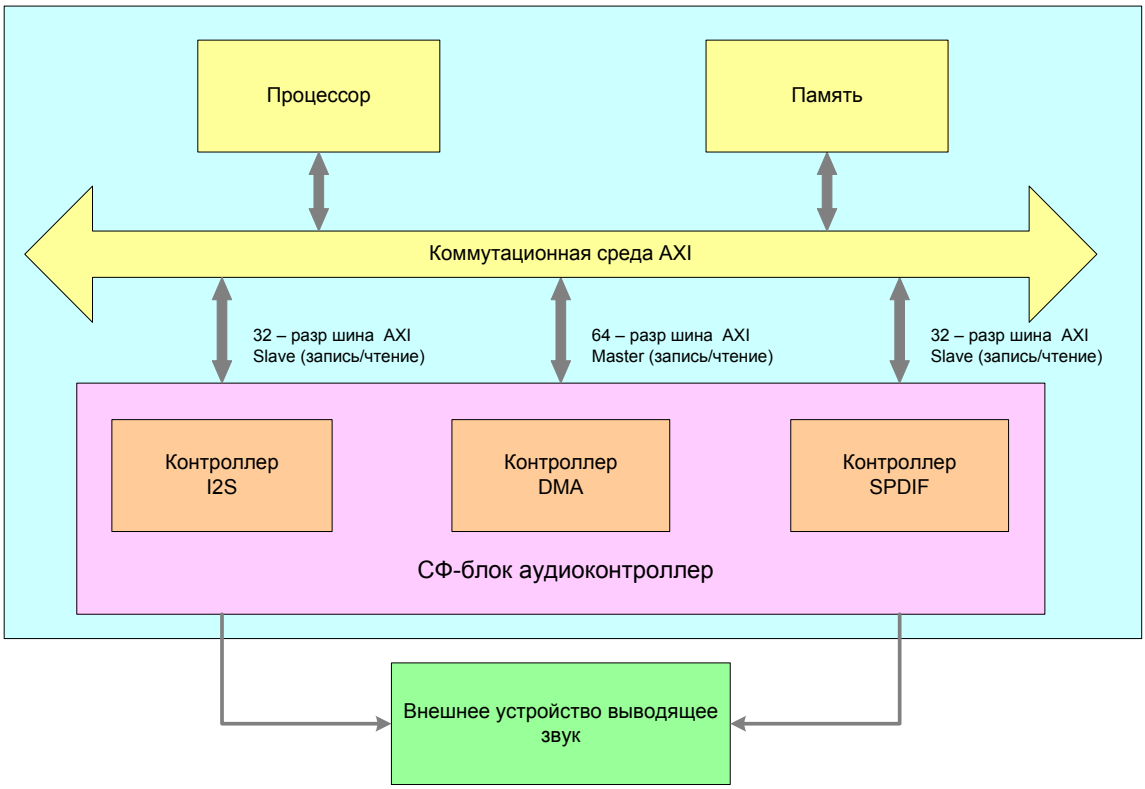


Рисунок 1.185 – Схема типового применения аудиоконтроллера

Контроллер обеспечивает передачу звуковых данных с частотами дискретизации от 22,05 КГц до 192 КГц. Частота передачи звука формируется на основе внешней звуковой частоты, которая наряду с системной синхронизацией поступает в звуковой контроллер от внешних программируемых генераторов синхросигналов.

1.3.7.1.2 Технические характеристики аудиоконтроллера

- Поддерживаемая разрядность цифровых отсчетов аудиосигнала (кол-во разрядов):
  - 16;
  - 20;
  - 24;
- Формат представления цифровых отсчетов аудиосигнала:
  - bigendian
  - littleendian;
- Поддерживаемые стандартные частоты дискретизации звука (КГц):
  - 192;
  - 176,4;
  - 96;
  - 88,2;
  - 48;
  - 44,1;
  - 32;
  - 22,05.
- Системная шина данных AMBA AXI с разрядностью по адресам 32 бит, по данным 32 и 64 бита;
- Системная частота: от 50 МГц и выше (тестирование велось при 162 МГц);
- Входная звуковая частота: кратна 22,5792 МГц, либо кратна 24,576 МГц.

1.3.7.1.3 Структура аудиоконтроллера

Аудиоконтроллер структурно состоит из трех блоков:

- контроллер многоканального звука с интерфейсом I2S;
- контроллер цифрового звука с интерфейсом SPDIF;
- контроллер AUDIO_DMA для обеспечения аудиоконтроллера звуковыми данными.

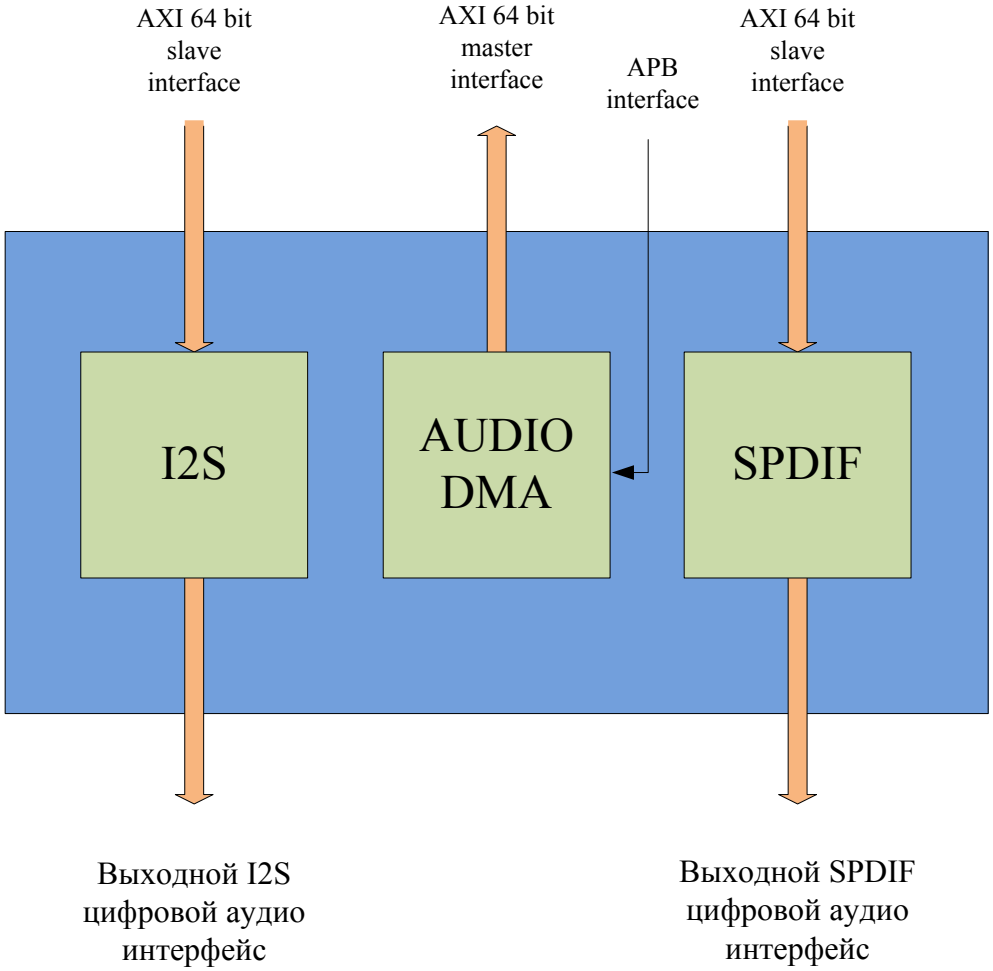


Рисунок 1.186 – Структурная схема аудиоконтроллера

Контроллер AUDIO_DMA читает по AXI шине 64-х разрядные данные из внешней памяти, преобразует их в 32-х разрядные данные и передает по AXI шине в один из звуковых контроллеров. Звуковые контроллеры I2S, SPDIF преобразуют получаемые данные в битовые потоки и передают их по соответствующему цифровому звуковому интерфейсу.

1.3.7.1.3.1 Контроллер многоканального звука с интерфейсом I2S

Контроллер многоканального звука с интерфейсом I2S (далее контроллер I2S) реализует интерфейс между центральным процессором и ЦАП с интерфейсом I2S, предназначенным для вывода аналогового многоканального аудиосигнала на звуковую аппаратуру.

На вход контроллера I2S поступают массивы цифровых отсчетов (далее, просто отсчеты) аудиосигнала разрядностью от 16 до 24 бит, которые преобразуются в последовательные битовые потоки в соответствии со спецификацией протокола I2S. Контроллер I2S способен принимать и преобразовывать массивы отсчетов для четырех стереоканалов одновременно, что позволяет поддерживать вывод многоканального звука вплоть до схемы 7.1.

Структурная схема контроллера I2S и схема его интеграции с центральным процессором представлена на рисунке 1.3.

Входные данные поступают в контроллер I2S через интерфейсный модуль AXI-Slave. Интерфейсный модуль AXI-Slave осуществляет следующие действия:

- взаимодействие с коммутационной средой посредством транзакций записи и чтения в соответствии с протоколом AXI 1.0;
- дешифрация поступающей в ходе транзакции адресной информации;
- запись полученных в ходе транзакции данных в блок регистров или блок буферов цифровых отчетов аудиосигнала на основе дешифрированной адресной транзакции.

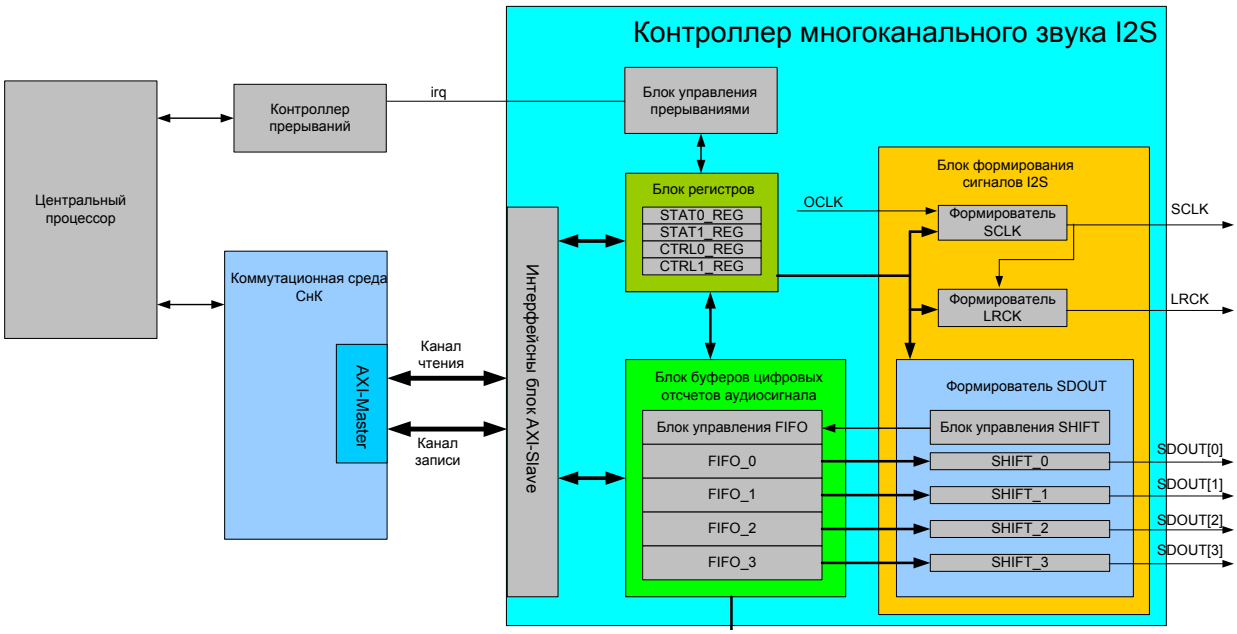


Рисунок 1.187 – Структурная схема контроллера I2S

Запись поступающего на вход контроллера I2S потока отсчетов производится в блок буферов цифровых отчетов аудиосигнала.

Блок буферов цифровых отчетов аудиосигнала представляет из себя набор из четырех программно доступных FIFO-буферов (FIFO_0, FIFO_1, FIFO_2, FIFO_3) и блока управления FIFO.

Каждый FIFO-буферов предназначен для промежуточного хранения массивов отсчетов одного стереоканала. Запись потока входных отсчетов в каждый FIFO-буфер ведется интерфейсный блоком AXI-Slave на основе дешифрированной адресной информации.

В случае, если контроллер I2S настраивается на работу в стереофоническом режиме, то используется только буфер FIFO_0. Если контроллер I2S настраивается на работу в многоканальном режиме (количество каналов превышает один), то используются остальные FIFO-буфера. Количество используемых FIFO-буферов зависит от количества выводимых каналов.

Блок управления FIFO управляет адресными счетчикам записи и чтения для каждого FIFO-буфера, а также отслеживает и информирует интерфейсный блок AXI-Slave и блок формирования сигналов I2S о статусе заполнения FIFO-буферов.

Блок формирования сигналов I2S реализует формирование сигналов SCLK, LRCK, SDOUT[3:0] в соответствии со спецификацией протокола I2S. Структурно, данный блок состоит из формирователя SCLK, формирователя LRCK и формирователя SDOUT.

Формирователь SCLK осуществляет генерацию сигнала SCLK требуемой частоты. Частота сигнала SCLK определяется частотой дискретизации исходного аудиосигнала (количество отсчетов в секунду) и разрядностью отсчетов. Генерация SCLK производится путем деления

опорного интерфейсного синхросигнала OCLK на целый коэффициент, задаваемый драйвером через программирование конфигурационных регистров контроллера I2S.

Формирователь SDOUT предназначен для преобразования отсчетов из параллельного кода в последовательный для передачи по одноразрядным линиям шины SDOUT[3:0].

Преобразование отсчетов из параллельного кода в последовательный происходит при помощи четырех сдвиговых регистров SHIFT_0, SHIFT_1, SHIFT_2, SHIFT_3. Каждый сдвиговый регистр однозначно соответствует FIFO-буферу с таким же индексом в блоке буферов цифровых отсчетов аудиосигнала.

Загрузка отсчетов из FIFO-буферов в сдвиговые регистры, а также управление работой сдвиговых регистров, осуществляется блоком управления SHIFT. Выдача битов со сдвиговых регистров на линии шины SDOUT тактируется сигналом SCLK.

Формирователь LRCK осуществляет генерацию сигнала LRCK, который является признаком принадлежности передаваемых по шине SDOUT[3:0] отсчетов к левому или правому каналу. Генерация LRCK тактируется сигналом SCLK и производиться на основе информации о разрядности отсчетов: сигнал LRCK удерживается в состоянии «1» или «0» строго n тактов сигнала SCLK, где n – разрядность передаваемых по линиям шины SDOUT[3:0] отсчетов.

1.3.7.1.3.2 Контроллер цифрового звука с интерфейсом SPDIF

Контроллер цифрового звука с интерфейсом SPDIF (далее контроллер SPDIF) реализует обмен данными между центральным процессором и внешним приемником цифровой звуковой информации по интерфейсу SPDIF. Контроллер предназначен для вывода одноканального и двухканального несжатого цифрового аудиосигнала и сжатого многоканального аудиосигнала.

На вход контроллера SPDIF поступает информация конфигурации аудиоканалов, пользовательские данные, массивы цифровых отсчетов аудиосигнала (далее, аудиосэмплы) разрядностью 16, 20 или 24 бита, которые преобразуются в последовательные битовые потоки в соответствие со стандартами IEC 60958 и IEC 61937 и выводятся наружу.

Структурная схема контроллера SPDIF представлена на рисунке 1.188.

Входные данные поступают в контроллер SPDIF через интерфейсный модуль AXI-Slave. Интерфейсный модуль AXI-Slave осуществляет следующие действия:

взаимодействие с коммутационной средой посредством транзакций записи и чтения в соответствии с протоколом AXI 1.0;

- дешифрация поступающей в ходе транзакции адресной информации;
- запись полученных в ходе транзакции данных в блок регистров, буферы статусной информации каналов, буфер пользовательских данных и буфер аудиосэмплов на основе дешифрированной адресной информации.

Буфер аудиосэмплов представляет собой программно доступный на запись FIFO-буфер и блок управления FIFO. FIFO-буфер предназначен для промежуточного совместного хранения аудиосэмплов левого и правого аудиоканалов. Запись потока звуковых отсчетов в FIFO-буфер ведется интерфейсным блоком AXI-Slave на основе дешифрированной адресной информации.

Блок управления FIFO управляет адресными счетчиками записи и чтения FIFO-буфера, а также передает информацию о статусе заполнения FIFO-буфера и ошибочных ситуациях в блок регистров. Схожее устройство имеют буферы статусной информации каналов и пользовательских данных.

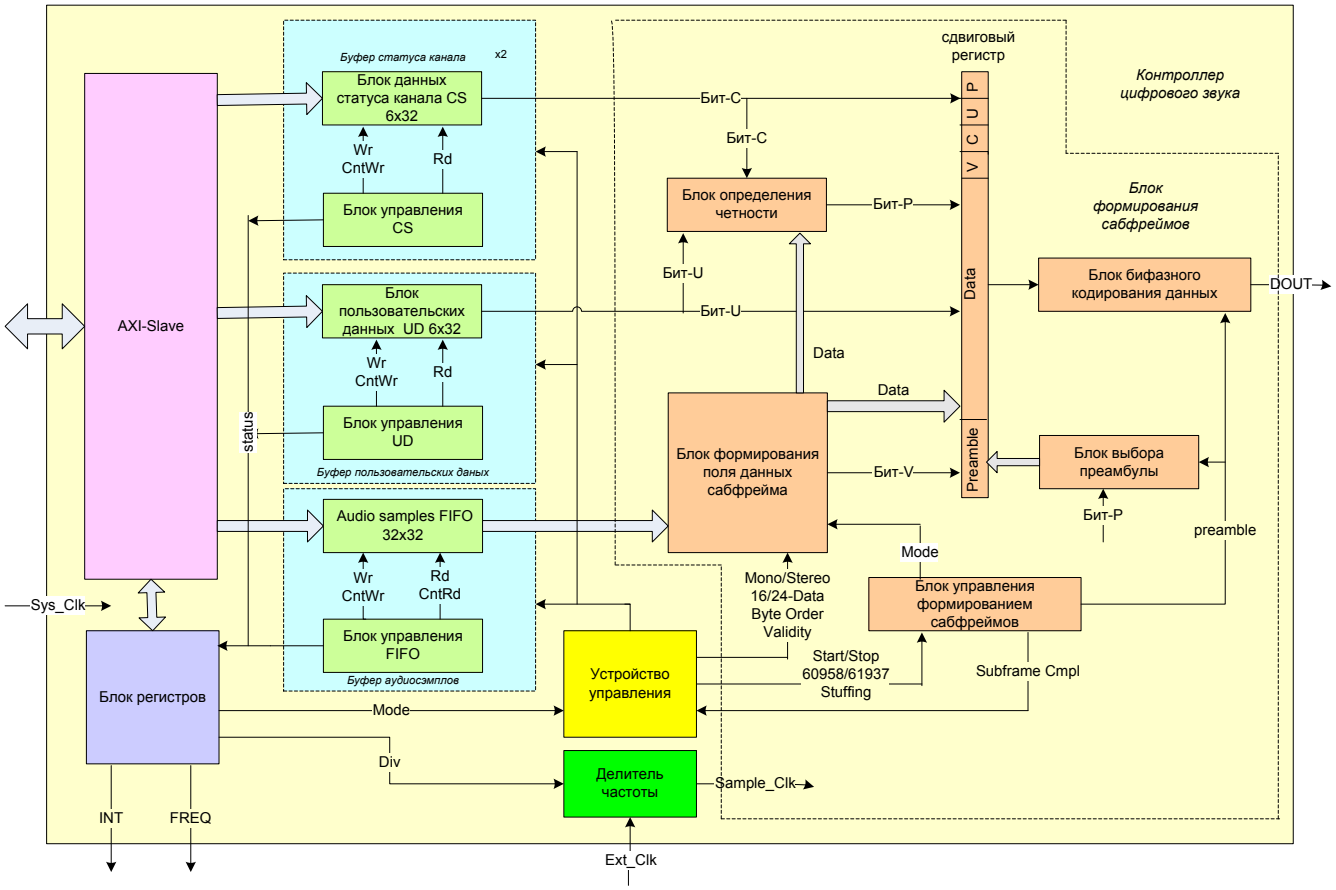


Рисунок 1.188 – Структурная схема контроллера SPDIF

Блок формирования сабфреймов предназначен для создания сабфреймов звуковых данных, которые затем побитно выдаются по интерфейсу SPDIF. Каждый сабфрейм состоит из:

- преамбулы, определением типа которой занимается блок выбора преамбулы;
- поля данных, которое заполняется информацией из буфера аудиосэмплов определенным образом, зависящим от режима работы. Порядок заполнения определяет блок формирования поля данных сабфрейма;
- битов статуса канала и пользовательских данных. Информация для этих полей побитно считывается из соответствующих буферов;
- бита контроля ошибки, определением которого занимается блок определения четности.

Блок управления формированием сабфрейма вырабатывает сигналы, позволяющие выбрать и собрать в определенном порядке информацию, формирующую звуковой сабфрейм, и вывести ее наружу. Поскольку информация вовне передается по одноразрядной линии DOUT, то для ее преобразования из параллельного кода в последовательный используется сдвиговый регистр, формирующий выходной побитовый поток данных, который перед выдачей наружу подвергается бифазному кодированию.

Устройство управления контроллера SPDIF следит за ходом передачи звуковой информации, выдавая сигналы структурирования данных – завершение вывода фреймов звуковой информации, блоков звуковой информации и пакетов, сжатых звуковых данных. Так же устройство управления контроллера SPDIF вырабатывает по мере необходимости сигналы чтения данных из буферов CS, UD и audiosamples.

Скорость передачи битов выходного битового потока звуковой информации, определяется частотой работы блока формирования сабфреймов (sample_clk), которая в свою очередь формируется из внешней звуковой частоты, путем ее деления на целочисленный коэффициент, задаваемый в конфигурационных регистрах контроллера SPDIF.

Блок регистров содержит в себе программно доступные на запись и чтение конфигурационные регистры, а именно:

- регистр управления, предназначенный для настройки скорости передачи аудиоданных и задания режимов работы контроллера SPDIF;
- регистр состояния, в котором отображается текущее состояние контроллера SPDIF;
- регистр масок прерываний, позволяющий выбирать события, по которым производится выработка сигнала прерывания процессора;
- регистр параметров сжатого звука, для настройки передачи многоканального сжатого аудиосигнала.

1.3.7.1.3.3 Контроллер AUDIO_DMA

В связи с тем, что контроллеры звука стандартов I2S и S/PDIF, входящие в аудиоконтроллер, являются ведомыми устройствами на шине AXI, подачу аудио данных на них должно осуществлять внешнее устройство. При этом подача данных может осуществляться центральным процессором системы, однако данный вариант вызовет необходимость обработки процессором большого количества прерываний, а также высокой скорости реакции на прерывание. Если это не является проблемой, то контроллер AUDIO_DMA может быть исключен из состава аудиоконтроллера. В противном случае в систему вводится контроллера AUDIO_DMA, который будет самостоятельно обеспечивать загрузку данных в контроллеры звука по мере опустошения их буферов.

Данное решение позволяет снизить частоту возникновения прерываний к центральному процессору, так как в этом случае необходимо отслеживать не заполнение буферов в контроллерах звука, а обеспечивать переключение страниц в памяти, размеры которых значительно превышают размеры внутренних буферов контроллеров звука.

Другим путем уменьшения частоты прерываний без использования AUDIO_DMA контроллера в составе аудиоконтроллера является увеличение внутренней буферной памяти контроллеров звука, что влечет за собой увеличение занимаемой площади.

Контроллер AUDIO_DMA предназначен для обеспечения загрузки данных в контроллеры цифрового звука аудиподсистемы. Контроллер AUDIO_DMA производит чтение данных из памяти, их буферизацию и запись данных в контроллер цифрового звука стандартов I2S и S/PDIF. Контроллер обеспечивает поддержку до четырех каналов чтения/записи данных, каждый из которых независимо конфигурируется для работы с собственным буфером как во внешней памяти, так и в контроллере звука.

При работе с SPDIF контроллером задействован только один (нулевой) канал контроллера AUDIO_DMA. При работе с I2S контролером, в зависимости от канальности звука, задействуются до четырех каналов контроллера AUDIO_DMA.

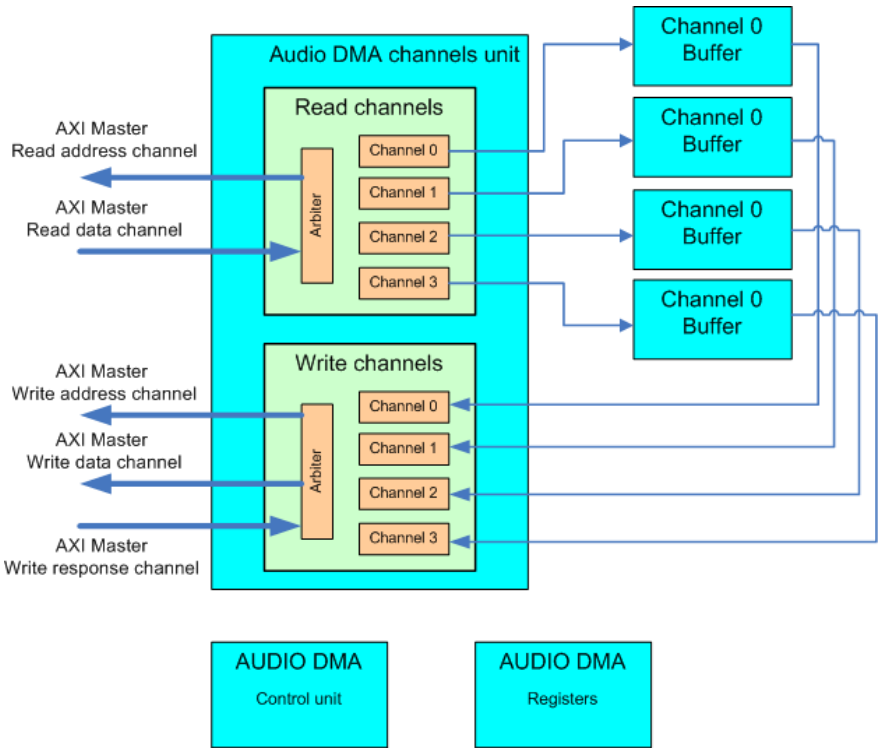


Рисунок 1.189 – Структура контроллера AUDIO_DMA аудиоподсистемы

Контроллер AUDIO_DMA имеет следующие интерфейсы стандарта AXI: канал чтения, канал записи, канал подтверждения записи. Посредством данных каналов контроллер AUDIO_DMA подключен к системной AXI шине системы. Данное подключение обеспечивает ему доступ к блокам памяти системы, а также к ведомым устройствам (контроллерам звука) по шине AXI.

Помимо данных интерфейсов контроллер AUDIO_DMA имеет управляющий интерфейс для конфигурирования его внутренних регистров, этот интерфейс не относится к стандартным интерфейсам.

Каждый канал контроллера AUDIO_DMA считывает звуковые данные из памяти страницами. Контроллеру задаются базовые адреса двух страниц в памяти. Пока контроллер читает данные из одной страницы, процессор может формировать данные другой. Как только формирование новой страницы закончено, контроллеру AUDIO_DMA разрешается перейти на чтение данных новой страницы по завершению чтения данных предыдущей страницы.

Читаемые 64-х разрядные данные помещаются в один из четырех внутренних буферов контроллера AUDIO_DMA. Далее 64-х разрядные данные преобразуются в 32-х разрядные и передаются в один из звуковых контролеров.

Для мониторинга уровня заполнения буферов в ведомых контроллерах звука на контроллер AUDIO_DMA непосредственно заведены внутренние статусные сигналы о количестве слов в буферах с контроллеров звука.

1.3.7.1.3.4 Интерфейс I2S

Интерфейс I2S состоит из 6 линий: линии синхронизации звуковых данных, линии указывающей данные какого канала передаются в настоящее время (левого или правого), 4-х линий данных (4 стереоканала).

На рисунке 1.190 представлена временная диаграмма передачи звука по I2S интерфейсу.

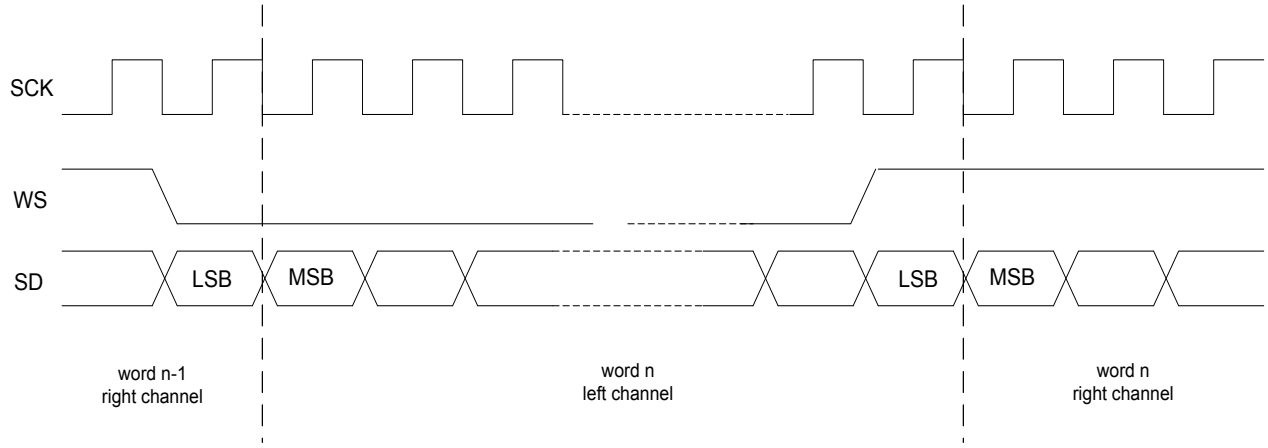


Рисунок 1.190 – Временная диаграмма протокола I2S

Цифровые отчеты звука, хранящиеся в 4-х FIFO I2S контроллера, поступают на соответствующие им сдвиговые регистры, которые обеспечивают выдачу отсчетов звука в последовательном коде по шине SDOUT[3:0]. Выдача бит на шину SDOUT[3:0] тактируется синхросигналом SCLK, который формируется из внешней частоты звуковой синхронизации. Отсчеты цифрового звука выдаются, начиная со старшего разряда. Выдаваемые по линиям шины SDOUT[3:0] данные сопровождаются сигналом LRCK, который является признаком принадлежности передаваемых отсчетов к левому или правому каналу. Сигнал LRCK удерживается в состоянии «1» или «0» n тактов сигнала SCLK, где n – разрядность передаваемых по линиям шины SDOUT[3:0] отсчетов. Сигнал LRCK на один такт опережает соответствующие ему данные. Все изменения на I2S интерфейсе происходят по срезу сигнала синхронизации SCK.

1.3.7.1.3.5 Интерфейс SPDIF

Интерфейс S/PDIF – это последовательный интерфейс, по которому передается звуковая информация, упакованная в определенные структуры – сабфреймы. Данные сабфреймы подвергнуты бифазному кодированию (Biphase-Mark Code) и потому, передаваемая по линии SPDIF информация представляет собой самосинхронизирующийся код, то есть состоит из звуковых данных и сигнала синхронизации.

Каждый бит информации, передаваемый по SPDIF интерфейсу представлен двумя временными интервалами. При передаче ‘0’ состояние линии SPDIF в течение двух временных интервалов, соответствующих этому биту, остается неизменным, а при передаче ‘1’ – меняется. К тому же состояние SPDIF линии меняется в начале передачи каждого нового бита. Таким образом, звуковые данные и тактовые импульсы объединяются в единый поток данных. На рисунке 1.191 представлен пример бифазного кодирования информации.

Частота дискретизации данных формируется на основе внешней звуковой частоты путем ее деления (смотри пункт 1.4.6).

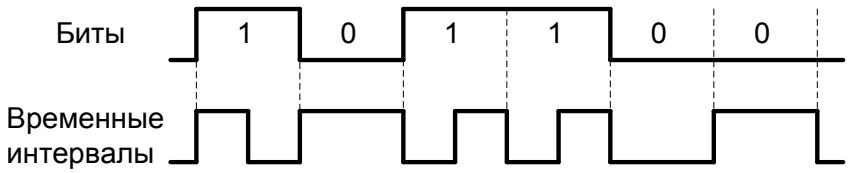


Рисунок 1.191 – Пример бифазного кодирования информации

Основной единицей при передаче звуковых данных по SPDIF интерфейсу является сабфрейм. Это относится как к передаче линейного двухканального звука (стандарт IEC 60958), так и сжатого многоканального звука (стандарт IEC 61937). Два сабфрейма формируют фрейм звуковой информации. Фреймы передаются блоками. Каждый блок содержит 192 фрейма (рисунок 1.193).



Рисунок 1.192 – Структура сабфрейма звуковой информации

На рисунке 1.192 представлена структура сабфрейма звуковой информации

Сабфрейм звуковой информации состоит из:

преамбулы, предназначенной для синхронизации и структурирования звуковой информации;

дополнительной информации пользователя Aux;

аудиосэмпла – отсчета цифрового звука;

бита значимости данных, указывающего на тип передаваемых данных (линейные или нелинейные звуковые данные);

бита пользовательских данных;

бита статусной информации канала;

бита четности.

Преамбула – четырехразрядная синхрогруппа, которая нарушает правила бифазного кодирования. Преамбула действует как сигнал синхронизации, обозначающий начало нового сабфрейма. Преамбуле соответствуют биты 0-3 сабфрейма (8 временных интервалов бифазного кода). Существует несколько видов преамбул:

“B” – 11101000, “Z” – 00010111;

“M” – 11100010, “X” – 00011101;

“W” – 11100100, “Y” – 00011011.

Сабфрейму присваивается преамбула в соответствии с правилом:

первый сабфрейм каждого фрейма (за исключением первого фрейма в блоке) имеет преамбулу “M” или “X”;

второй сабфрейм каждого фрейма имеет преамбулу “Y” или “W”;

первый сабфрейм первого фрейма в блоке имеет преамбулу B или “Z”;

преамбулы имеют значения “XYZ”, если второе состояние бифазного кода бита-P предшествующего преамбуле равно‘1’, и “MWB” если – ‘0’.

поскольку при выключенном контроллере на линию DOUT выдается ноль, то первая преамбула после включения контроллера будет иметь значение “B”.

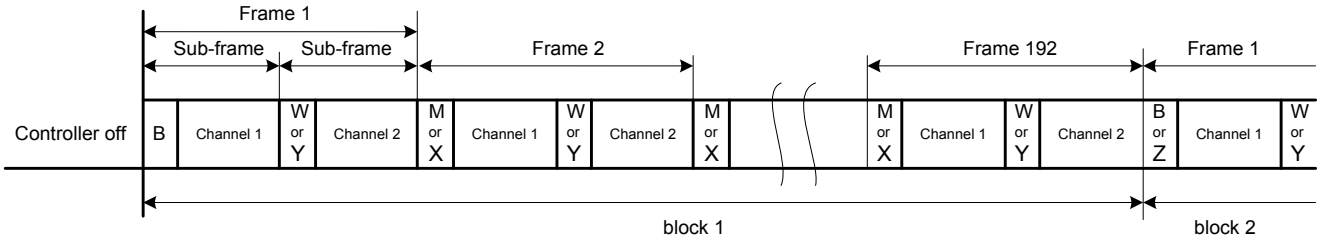


Рисунок 1.193 – Изменение преамбулы сабфрейма по ходу передачи звука

Восемь состояний преамбулы выводятся непосредственно на выходную линию DOUT, не подвергаясь бифазному кодированию.

Биты 4-27 в звуковом сабфрейме занимает поле данных (рисунок 1.192). Данные берутся из FIFO аудиосэмплов SPDIF контролера. В зависимости от значения бита SPDIF_DATA регистра управления она располагается так, как показано на рисунке 1.195. В каждом 32-х разрядном слове FIFO-аудиосэмплов может храниться:

один аудио-сэмпл размером 24, 20 или 16 бит – в данном случае контроллер считает значимыми данные с 31-ого по 8 разряд, которые подставляются в поле данных сабфрейма. При размере аудио-сэмпла 20 и 16 бит, пользователь может передавать дополнительную информацию в поле Aux.

два аудио-сэмпла размером по 16 бит – в данном случае контроллер дополняет 16-битные данные нулями до 24-х разрядов и подставляет в сабфрейм. В этом случае пользователь лишается возможности передавать данные в поле Aux. Контроллер заполнит его нулями.

В FIFO-аудиосэмплов данные могут храниться как в формате big-endian, так и little-endian (выбор осуществляется битом регистра управления BYTE_ORDER). При чтении из FIFO данные преобразуются в формат little-endian (т.е. в битах 31-23 слова располагается старший байт информации).

Бит-V указывает на тип передаваемых звуковых данных – при передаче линейного звука IEC 60958 этот бит устанавливается в ‘0’, а при передаче сжатых данных IEC 61937 в ‘1’.

Бит-C сабфрейма содержит в себе конфигурационную информацию канала (192 бита на каждый канал)

Бит-U содержит пользовательскую информацию

Бит-P – бит четности сабфрема:

‘0’ – если количество единиц в битах с 4-ого по 30-ый сабфрейма четное;

‘1’ – если количество единиц в битах с 4-ого по 30-ый сабфрейма нечетное.

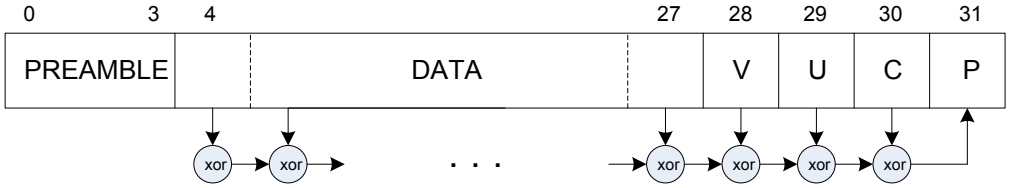


Рисунок 1.194 – Формирование бита четности сабфрейма

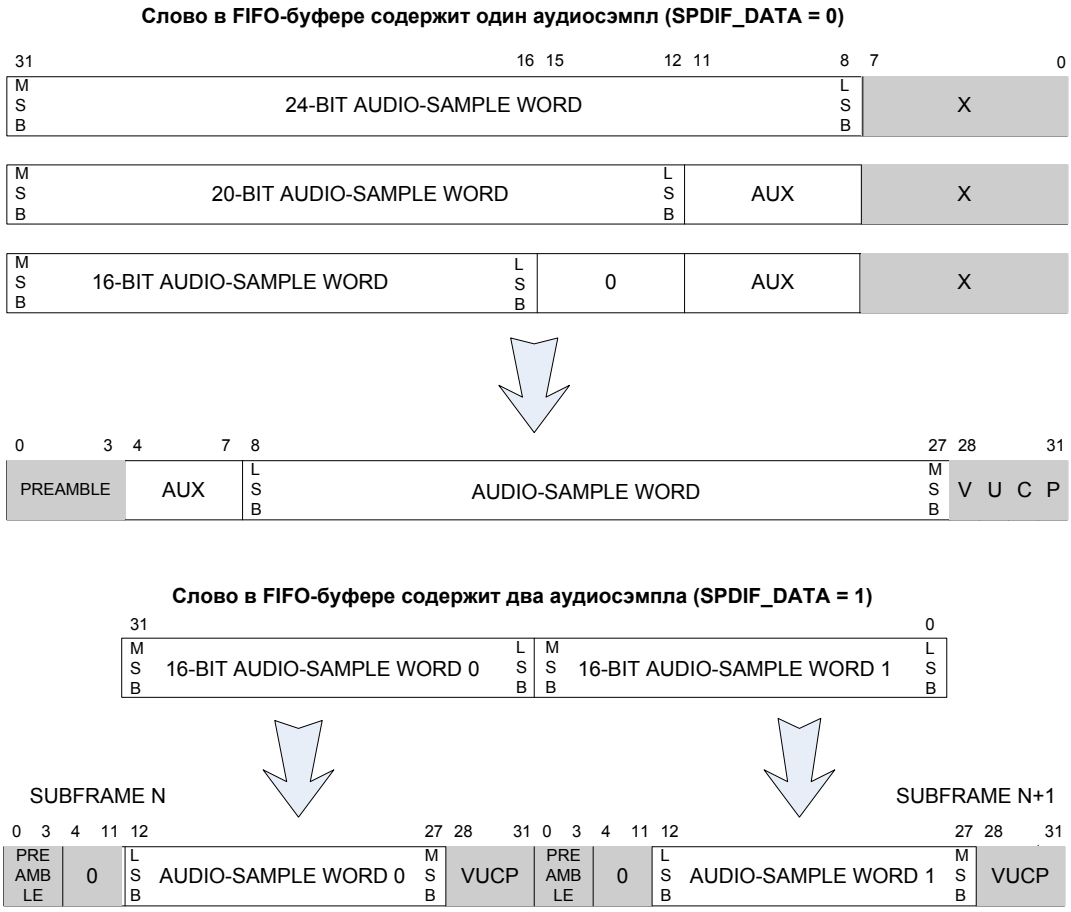


Рисунок 1.195 – Расположение аудиосэмплов в поле данных сабфрейма

Настройка скорости передачи звука по I2S интерфейсу

Частота дискретизации звука по I2S интерфейсу – это частота передача двух отсчетов цифрового звука (для левого и правого канала). Передача двух отсчетов N-ой разрядности занимает 2*N тактов синхросигнала I2S-интерфейса SCK. Для формирования сигнала SCK контроллер I2S использует программируемый целочисленный делитель входной звуковой частоты, поступающей на порт i2s_mclk_i. Таким образом,

Требуемая частота дискретизации = (Внешняя частота / SCLK_RATIO * 2N), где
N – разрядность передаваемых по интерфейсу I2S звуковых данных;
SCLK_RATIO – программируемый целочисленный делитель.

Настройка скорости передачи звука по SPDIF интерфейсу.

Настройка скорости звукового канала производится путем целочисленного деления частоты внешнего интерфейсного синхросигнала spdif_mclk_i до опорной, а затем опорной частоты до частоты, соответствующей желаемой частоте дискретизации звука. В качестве опорных частот выступают: 22,5792 МГц (соответствует частоте звуковой дискретизации 176,4 КГц) и 24,576 МГц (соответствует частоте звуковой дискретизации 192 КГц). Таким образом, для поддержки всех стандартных частот дискретизации на порт звуковой синхронизации аудиоконтроллера необходимо подавать две внешние частоты, для чего должен быть предусмотрен внешний мультиплексор синхросигналов. Входная звуковая частота не может быть меньше опорной. Из опорной частоты путем целочисленного деления можно получить желаемую частоту звуковой дискретизации (точнее частоту смены временных интервалов

бифазного кода на SPDIF интерфейсе). Частота звуковой дискретизации для SPDIF интерфейса – это частота выдачи звуковых фреймов.

Поле FREQ_BASE регистра управления контроллера SPDIF, задает целочисленный коэффициент деления внешней частоты spdif_mclk_i до опорной. Коэффициент деления опорной частоты до необходимой задается полем FREQ_DIV регистра управления контролера SPDIF.

Деление производится по формулам:

Опорная частота = Внешняя частота / (FREQ_BASE+1);

Искомая частота = Опорная частота / (FREQ_DIV+1) – получаемое значение является частотой смены состояний бифазного кодирования.

Искомая частота / 128 – частота передачи одного фрейма звуковой информации, то есть частота дискретизации.

Частота дискретизации = Внешняя частота / ((FREQ_BASE+1) * (FREQ_DIV+1) * 128);

Коэффициенты деления подбираются по формулам:

FREQ_BASE = (Внешняя частота / (22.5792 МГц или 24.576 МГц)) - 1
FREQ_DIV = ((22.5792 МГц или 24.576 МГц) / (Частота дискретизации * 128)) - 1

Настраивать делитель частоты контроллера SPDIF можно при помощи таблицы 1.3. Так, например, имея внешнюю входную частоту 245.76 МГц, и желая получить частоту дискретизации аудиосигнала 48 кГц, необходимо в регистре управления SPDIF контроллера задать следующие значения: FREQ_BASE = 9, FREQ_DIV = 3.

Таблица 1.573 – Настройка SPDIF контроллера на вывод звука с определенной частотой дискретизации

		FREQ	
		‘0’	‘1’
FREQ_BASE	15	361.2672	393.216
	14	338.688	368.64
	13	316.1088	344.064
	12	293.5296	319.488
	11	270.9504	294.912
	10	248.3712	270.336
	9	225.792	245.76
	8	203.2128	221.184
	7	180.6336	196.608
	6	158.0544	172.032
	5	135.4752	147.456
	4	112.896	122.88
	3	90.3168	98.304
	2	67.7376	73.728
	1	45.1584	49.152
	0	22.5792	24.576

Возможные значения частоты внешнего звукового синхросигнала spdif_mclk, МГц

Продолжение таблицы 1.573

FREQ_DIV	Значения частоты дискретизации аудиосигнала, кГц	
0	176.4	192
1	88.2	96
2	58.8	64
3	44.1	48
4	35.28	38.4
5	29.4	32
6	25.2	27.42857
7	22.05	24
8	19.6	21.33333

1.3.7.1.3.6 Сигналы прерываний

В аудиоконтроллере имеются три сигнала прерываний: от I2S контроллера, от SPDIF контроллера и от AUDIO_DMA. Прерывания являются уровневыми, поэтому после обработки их необходимо сбрасывать. Общая схема прерываний выглядит следующим образом: в каждом из устройств аудиоконтроллера существует ряд событий, по которым возможна выработка прерываний. Эти события отображаются в регистрах статуса/прерываний каждого из блоков аудиоконтроллера (I2S, SPDIF, AUDIO_DMA). Разрешение выработки прерываний по этим событиям задается через регистр масок прерываний. При возникновении события, в соответствующем бите регистра статуса устанавливается единица, при разрешенной маске прерываний эта единица попадет на линию внешнего прерывания. После обработки прерывания необходимо записать в соответствующий бит регистра статуса единицу – данное действие сбросит бит регистра статуса в ноль и вместе с ним снимет формируемое им прерывание.

1.3.7.1.4 Формат входных данных аудиоконтроллера

Исходными звуковыми данными для I2S, SPDIF контроллеров являются 32-х разрядные слова, передаваемые по AXI шине, и записываемые в FIFO аудиоданных. Каждое 32-х разрядное слово может нести в себе либо один аудиосэмпл разрядностью 16, 20 или 24 бита, либо два аудиосэмпла разрядностью 16 бит. Таким образом, возможны два вида работы с 16-битными звуковыми данными: когда они передаются отдельно в каждом 32-х разрядном слове, либо укладываются в 32-х разрядное слово попарно. Второй случай предпочтительней, поскольку позволяет в два раза сократить объем данных, передаваемых звуковым контроллерам.

32-х разрядное слово, передаваемое в звуковой контроллер, может иметь формат big-endian, либо little-endian.

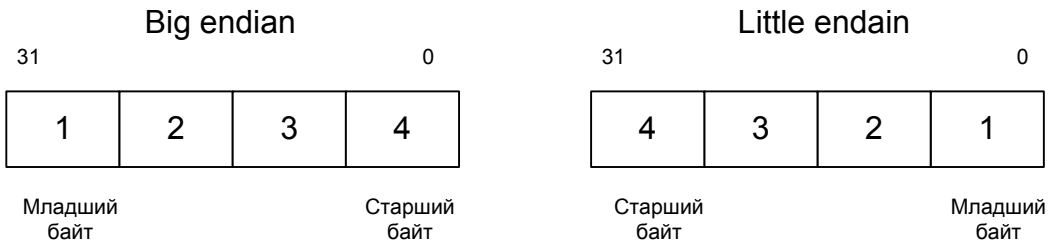


Рисунок 1.196 – Порядок расположения байт в 32-х разрядном слове данных

Каждый канал I2S интерфейса и SPDIF интерфейс (при передаче линейного звука) являются стереоканалами, т.е. по ним поочередно передаются данные левого и правого каналов. Таким образом, звуковые данные в звуковые контроллеры передаются попарно.

Моно звук – это передача одинаковых данных по левому и правому каналам. Программно монозвучание достигается путем записи двух одинаковых сэмплов в буфер звукового контроллера, которые затем выводится на соответствующий звуковой интерфейс.

Контроллер SPDIF позволяет автоматически реализовать моно звуковой режим. Аппаратный моно режим активируется путем установки бита SPDIF_CHANNELS в ‘1’. В этом случае каждый аудиосэмпл из FIFO аудиоданных подставляется как в поле данных первого (левого), так и в поле данных второго (правого) сабфрейма. В этом случае буфер SPDIF контроллера должен содержать аудиосэмплы только для левого канала. Таким образом, аппаратная поддержка моно звука позволяет в два раза сократить объем передаваемых SPDIF контроллеру данных.

1.3.7.1.5 Программный интерфейс аудиоконтроллера

Данный раздел содержит информацию о программном интерфейсе аудиоконтроллера: описание регистров, режимов доступа к ним и особенности задания режимов.

Задание конфигурационных параметров и отображение текущего состояния аудиоконтроллера ведется через регистры. Доступ к регистрам AUDIO_DMA ведется через системный интерфейс управления (пункт 1.4.3). Доступ к регистрам SPDIF и I2S контроллеров ведется через AXI Slave интерфейс (пункт 1.4.1). Каждое обращение к регистрам имеет выравнивание до 32-х разрядов.

При обращении к регистрам аудиоконтроллера по зарезервированным адресам никаких ошибок не возникает, все пересылки завершаются нормально. При этом операции записи игнорируются, операции чтения возвращают все нули. При попытке записи в регистры, предназначенные только для чтения, операция записи завершается успешно, но никакой записи не производится и состояние регистров не изменяется.

Внутренние регистры аудиоконтроллера имеют связь со всеми блоками аудиоконтроллера, обеспечивая их необходимыми данными.

1.3.7.1.5.1 Группы регистров аудиоконтроллера

Управление аудиоконтроллером и индикация информации о его состоянии ведется посредством регистров. Регистры аудиоконтроллера объединены в группы по функциональному признаку (таблица 1.574). Каждая группа регистров должна иметь свой уникальный базовый адрес. Базовые адреса должны быть выровнены в соответствии с адресным пространством соответствующей группы регистров.

Таблица 1.574 – Список групп регистров аудиоконтроллера

Наименование	Адресное пр-во	Описание
REG_I2S	0x00-0xFF	Регистры I2S контроллера
REG_SPDIF	0x00-0xFF	Регистры SPDIF контроллера
REG_AUDIO_DMA	0x800-0x8FF	Регистры AUDIO_DMA

При описании режимов доступа к отдельным полям регистров аудиоконтроллера в дальнейшем используются следующие условные обозначения:

- “RO” – поле регистра предназначено только для чтения. Запись любых значений в поле игнорируется;
- “WO” – поле регистра предназначено только для записи. При чтении возвращает ноль;

- “RW” – поле регистра предназначено как для чтения, так и для записи;
- “RW1” – отдельный бит регистра предназначен для чтения. Возможна запись ‘1’. Запись ‘0’ игнорируется;
- “RC1” – отдельный бит регистра предназначен для чтения. При записи ‘1’ значение бита сбрасывается в ‘0’. Запись ‘0’ игнорируется;

Описания регистров аудиоконтроллера и принципов работы с отдельными полями регистров даны ниже.

1.3.7.1.5.2 Регистры контроллера I2S

По AXI интерфейсу I2S контроллера доступны как его программируемые регистры, так и внутренние буферы звуковых данных (таблица 1.575). Настройкой I2S контроллера через его программируемые регистры занимается пользователь (процессор), в то время как пополнением буферов звуковых данных занимается AUDIO_DMA.

Контроллер I2S является 32-х разрядным устройством, которое подключается к младшим разрядам 64-х разрядной шины данных AXI интерфейса. В зависимости от адреса, 32-х разрядные данные передаются либо в младших, либо в старших разрядах 64-х разрядной шины данных. Для того чтобы значащие данные передавались именно в младших разрядах шины данных, адреса регистров контроллера I2S выровнены по границе 64 бита (8 байт).

Таблица 1.575 – Регистры контроллера I2S

Наименование	Адрес	Описание
REG_I2S_CTRL0	0x00	Регистр 0 конфигурации-управления
REG_I2S_CTRL1	0x08	Регистр 1 конфигурации-управления
REG_I2S_STAT0	0x10	Регистр статуса 0
REG_I2S_STAT1	0x18	Регистр статуса 1
I2S_FIFO_0	0x20	Буфер 0 цифровых отсчетов аудиосигнала
I2S_FIFO_1	0x28	Буфер 1 цифровых отсчетов аудиосигнала
I2S_FIFO_2	0x30	Буфер 2 цифровых отсчетов аудиосигнала
I2S_FIFO_3	0x38	Буфер 3 цифровых отсчетов аудиосигнала

REG_I2S_CTRL0 – регистр управления I2S контроллера

Смещение x00, обращение только словом.

Данный регистр позволяет настроить I2S контроллер на определенный режим работы, управлять запуском и выключением I2S контроллера (таблица 1.576).

Таблица 1.576 – Разряды регистра управления I2S контроллера (REG_I2S_CTRL0)

Биты	Доступ	После RESET	Описание
31-21	RO	0	Поле зарезервировано. При чтении возвращает ноль.
20-16	RW	0	RES (dac_resolution) – разрядность цифровых отсчетов аудиосигнала (разрешение внешнего ЦАП). Значение разрядности равно UNSIGNED(RES) + 1. Используемая разрядность: 32 бита (значение RES – «11111»); 24 бита (значение RES – «10111»); 20 бит (значение RES – «10011»); 16 бит (значение RES – «01111»). При двух цифровых отсчетах в слове значение RES аппаратно принимается равным 16 бит (значение RES – «01111»).
15-12	RO	0	Поле зарезервировано. При чтении возвращает ноль.
11	RW	0	EL_3_en – маска для прерывания, возникающего по достижении буфером 3 ”уровня заполнения”, равного или меньшего половины размера буфера. EL_3_en = «1» - прерывание разрешено; EL_3_en = «0» - прерывание запрещено.

Продолжение таблицы 1.576

Биты	Доступ	После RESET	Описание
10	RW	0	EL_2_en – маска для прерывания, возникающего по достижении буфером 2 ”уровня заполнения”, равного или меньшего половины размера буфера. EL_2_en = «1» - прерывание разрешено; EL_2_en = «0» - прерывание запрещено.
9	RW	0	EL_1_en – маска для прерывания, возникающего по достижении буфером 1 ”уровня заполнения”, равного или меньшего половины размера буфера. EL_1_en = «1» - прерывание разрешено; EL_1_en = «0» - прерывание запрещено.
8	RW	0	EL_0_en – маска для прерывания, возникающего по достижении буфером 0 ”уровня заполнения”, равного или меньшего половины размера буфера. EL_0_en = «1» - прерывание разрешено; EL_0_en = «0» - прерывание запрещено.
7	RW	0	E3_en (E3_mask) – маска для прерывания, возникающего при пустом буфере 3. E3_en = «1» - прерывание разрешено; E3_en = «0» - прерывание запрещено.
6	RW	0	E2_en (E2_mask) – маска для прерывания, возникающего при пустом буфере 2. E2_en = «1» - прерывание разрешено; E2_en = «0» - прерывание запрещено.
5	RW	0	E1_en (E1_mask) – маска для прерывания, возникающего при пустом буфере 1. E1_en = «1» - прерывание разрешено; E1_en = «0» - прерывание запрещено.
4	RW	0	E0_en (E0_mask) – маска для прерывания, возникающего при пустом буфере 0. E0_en = «1» - прерывание разрешено; E0_en = «0» - прерывание запрещено.
3	RW	0	CH (channel configuration) – способ упаковки цифровых отсчетов аудиосигнала: CH = «0» - один цифровой отсчет аудиосигнала на 32-х разрядное слово; CH = «1» - два цифровых отсчета аудиосигнала на 32-х разрядное слово.
2	RW	0	BR (byte re-order) – перестановка байт принимаемого по AXI слова данных: «0» - сохраняется исходный порядок байт (little endian); «1» - исходный порядок байт меняется на противоположный (1234 -> 4321).
1	RW	0	CSD (Ctrl_Sync_Domain) – управление выбором частоты генератора для формирования синхроимпульса OCLK.
0	RW1	0	PRS (preset) – программный сброс. Активным является высокий уровень сигнала, по которому происходит установка всех внутренних счетчиков и сигналов управления в начальное состояние.

REG_I2S_CTRL1 – регистр управления I2S контроллера

Смещение x10, обращение только словом

Данный регистр позволяет установить скорость выдачи звуковых данных, управлять запуском и выключением I2S контроллера (таблица 1.577).

Таблица 1.577 – Разряды регистра 1 управления I2S контроллера (REG_I2S_CTRL1)

Биты	Доступ	После RESET	Описание
31-9	RO	0	Поле зарезервировано. При чтении возвращает ноль.
8	RW	0	EN (work_en) – разрешение работы. Активным является высокий уровень, при котором разрешается работа генератора внешнего синхросигнала и блока управления выдачей цифровых отсчетов аудиосигнала. Описать сразу ли выключается.

Продолжение таблицы 1.577

Биты	Доступ	После RESET	Описание
7-0	RW	0	SCLK_RATIO[8:0] – коэффициент деления входной опорной частоты (более подробное описание в пункте 1.4.6). При задании нулевого значения этого поля SCLK_RATIO аппаратно принимается равным 1.

REG_I2S_STAT0 – регистр 0 статуса I2S контроллера

Смещение x18, обращение только словом

Данный регистр индицирует состояние буферов цифровых отсчетов каналов 0, 1 (таблица 1.578).

Таблица 1.578 – Разряды регистра 0 статуса I2S контроллера (REG_I2S_STAT0)

Биты	Доступ	После RESET	Описание
31	RO	1	EL_1 – признак полупустого буфера 1
30	RO	1	E_1 – признак пустого буфера 1
29	RO	0	F_1 – признак заполненности буфера 1
28-24	RO	0	Поле зарезервировано. При чтении возвращает ноль.
23-16	RO	0	CTW_1 – количество слов в буфере цифровых отсчетов аудиосигнала 1
15	RO	1	EL_0 – признак полупустого буфера 0
14	RO	1	E_0 – признак пустого буфера 0
13	RO		F_0 – признак заполненности буфера 0
12-8	RO	0	Поле зарезервировано. При чтении возвращает ноль.
7-0	RO	0	CTW_0 – количество слов в буфере цифровых отсчетов аудиосигнала 0

REG_I2S_STAT1 – регистр 1 статуса I2S контроллера

Смещение x18, обращение только словом

Данный регистр индицирует состояние буферов цифровых отсчетов каналов 2, 3 (таблица 1.579).

Таблица 1.579 – Разряды регистра 1 статуса I2S контроллера (REG_I2S_STAT1)

Биты	Доступ	После RESET	Описание
31	RO	1	EL_3 – признак полупустого буфера 3
30	RO	1	E_3 – признак пустого буфера 3
29	RO	0	F_3 – признак заполненности буфера 3
28-24	RO	0	Поле зарезервировано. При чтении возвращает ноль.
23-16	RO	0	CTW_3 – количество слов в буфере цифровых отсчетов аудиосигнала 3
15	RO	1	EL_2 – признак полупустого буфера 2
14	RO	1	E_2 – признак пустого буфера 2
13	RO	0	F_2 – признак заполненности буфера 2
12-8	RO	0	Поле зарезервировано. При чтении возвращает ноль.
7-0	RO	0	CTW_2 – количество слов в буфере цифровых отсчетов аудиосигнала 2

CTW_i (counter_of_words) – количество слов в буфере цифровых отсчетов аудиосигнала.

E_i (empty) – признак пустого буфера, активный – высокий уровень. E_i Устанавливается в ‘1’ по аппаратному и программному сбросу, а в рабочем режиме при наступлении условия «полностью пустой буфер» (CTW_i = 0). Сбрасывается аппаратно, когда условие «полностью пустой буфер» не выполняется (CTW_i # 0).

EL_i (empty-level) – признак полупустого буфера, активный – высокий уровень. Возникает при достижении буфером ”уровня заполнения”, равного или меньшего половины размера буфера.

Сбрасывается аппаратно, когда условие «полупустой буфер» не выполняется.

F_i (full) – признак полного заполнения буфера, активный – высокий уровень. Возникает при достижении буфером ”уровня заполнения”, равного размеру буфера.

Сбрасывается аппаратно, когда условие «полный буфер» не выполняется.

1.3.7.1.5.3 Регистры контроллера SPDIF

По AXI интерфейсу SPDIF контроллера доступны как его программируемые регистры, так и внутренние буферы данных (таблица 1.580). Настройкой SPDIF контроллера через его программируемые регистры занимается пользователь (процессор), в то время как пополнением буфера звуковых данных занимается AUDIO_DMA.

Контроллер SPDIF является 32-х разрядным устройством, которое подключается к младшим разрядам 64-х разрядной шины данных AXI интерфейса. В зависимости от адреса, 32-х разрядные данные передаются либо в младших, либо в старших разрядах 64-х разрядной шины данных. Для того чтобы значащие данные передавались именно в младших разрядах шины данных, адреса регистров контроллера SPDIF выровнены по границе 64 бита (8 байт).

Таблица 1.580 – Таблица регистров контроллера SPDIF

Наименование	Смещение	Описание
REG_SPDIF_CTRL	0x00	Регистр управления
REG_SPDIF_STAT	0x08	Регистр прерываний/статуса
REG_SPDIF_MASK	0x10	Регистр маскирования прерываний
REG_NONLIN_PARAMS	0x18	Регистр параметров блока сжатых аудиоданных
SPDIF_BUF_CS1	0x20	Буфер статуса первого аудиоканала
SPDIF_BUF_CS2	0x28	Буфер статуса второго аудиоканала
SPDIF_BUF_UD	0x30	Буфер пользовательской информации
SPDIF_FIFO_AUDIOSAMPLES	0x38	Буфер аудиосэмплов

REG_SPDIF_CTRL – регистр управления SPDIF

Смещение x00, обращение только словом

Данный регистр позволяет настроить SPDIF контроллер на определенный режим работы, установить скорость выдачи звуковых данных, запустить SPDIF контроллер, выключить SPDIF контроллер путем его программного сброса (таблица 1.581).

Таблица 1.581– Разряды регистра управления SPDIF (REG_SPDIF_CTRL)

Биты	Доступ	После RESET	Описание
31	RW	0	Бит SPDIF_START – запись ‘1’ включает контроллер (инициирует начало передачи сабфреймов звуковой информации по линии SPDIF). Бит сбрасывается в ‘0’ программным сбросом.
30	R,W1	0	Бит SPDIF_RESET – программный сброс. Запись ‘1’ выключает контроллер, путем его сброса. Сброс контроллера длится четыре такта системной частоты – в течение этого времени бит SPDIF_RESET находится в состоянии ‘1’, по завершению сброса бит устанавливается в ‘0’. В течение четырех тактов сброса регистры контроллера не доступны.
29	RW	0	Бит SPDIF_MODE – вид передаваемых по SPDIF данных. ‘0’ – линейные аудиоданные (IEC 60958), ‘1’ – сжатые аудиоданные (IEC 61937).
28	RW	0	Бит SPDIF_CHANNELS – выбор количества каналов: ‘0’ – двухканальный линейный звук (стерео) или многоканальный сжатый звук; ‘1’ – одноканальный линейный звук (моно).
27-24	RO	0	Поле зарезервировано. Читается нулями.

Продолжение таблицы 1.581

Биты	Доступ	После RESET	Описание
23	RW	0	Бит SPDIF_DATA – количество адиосэмплов, содержащихся в одном 32-х разрядном слове данных: '0' – один 24-х, 20-ти или 16-ти битный аудисэмпл; ‘1’ – два 16-ти битных аудиосэмпла.
22	RW	0	Бит BYTE_ORDER – порядок байтов в слове данных аудиосэмпла: ‘0’ – little endian ([31:0] = ст. байт … мл. байт);. ‘1’ – big endian ([31:0] = мл. байт … ст. байт). Пользовательские данные и данные статуса канала вне зависимости от значения бита BYTE_ORDER хранятся в формате little_endian.
21	RW	0	Бит UD_EN – разрешение передачи пользовательских данных: ‘0’ – пользовательские данные не передаются. Буфер пользовательских данных заполнять не следует. Контроллер автоматически подставляет ноль в бит-U сабфреймов; ‘1’ – пользовательские данные передаются. Следует пополнять буфер пользовательских данных по мере его опустошения.
20-16	RO	0	Поле зарезервировано. Читается нулями.
15-14	RW	01	PROG_LEVEL – программируемый уровень заполненности FIFO аудиоданных, по которому вырабатывается сигнал статуса FIFO_LEVEL: '00’ – заполнен на четверть; ‘01’ – заполнен на половину; ‘10’ – заполнен на три четверти; ‘11’ – режим программирования уровня отключен. Сигнал FIFO_LEVEL всегда в нуле.
13-8	RO	0	Поле зарезервировано. Читается нулями.
7	RW	0	FREQ – бит выбора опорной частоты передачи звука. '0' – 22,5792 МГц , ‘1’ – 24,576 МГц.
6-3	RW	0	FREQ_BASE - коэффициент деления входной частоты до опорной частоты передачи звука. В качестве опорных частот выступают: 22,5792 МГц и 24,576МГц. Деление производится по формуле: Опорная частота = Входная частота / (FREQ_BASE+1)
2-0	RW	0	FREQ_DIV - коэффициент деления опорной частоты передачи данных. Настройка на необходимую частоту дискретизации аудиосигнала осуществляется в соответствии с таблицей: FREQ_DIV FREQ = 22,5792 FREQ = 24,576 0 - 176.4 192 1 - 88.2 96 2 - 58.86 64 3 - 44.1 48 4 - 35.28 38.4 5 - 29.4 32 6 - 25.2 27.42 7 - 22.05 24

REG_SPDIF_STAT – регистр статуса SPDIF

Смещение x08, обращение только словом

Данный регистр предназначен для индикации текущего состояния SPDIF контроллера (таблица 1.582). Существует два вида отображения состояния SPDIF контроллера: поточный и триггерный. При поточном отображении, меняющееся во времени событие в реальном времени отображается в соответствующем бите регистра REG_SPDIF_STAT (это относится к битам 15, 16 данного регистра). При триггерном отображении работает следующая схема: по возникновению определенного события, соответствующий ему бит в регистре REG_SPDIF_STAT устанавливается в единицу, и остается в этом состоянии до тех пор, пока программист не сбросит его в ноль, записав в этот бит единицу. По таким событиям возможна выработка прерываний, если она разрешена в регистре REG_SPDIF_INT_EN.

Таблица 1.582 – Разряды регистра статуса SPDIF (REG_SPDIF_STAT)

Биты	Доступ	После RESET	Описание
31	RC1	0	Бит FRAME_COMPLETE – бит устанавливается в ‘1’ по завершении вывода очередного фрейма. Обнуляется при записи в этот бит ‘1’.
30	RC1	0	Бит BLOCK_COMPLETE – бит устанавливается в единицу по завершении вывода блока из 192-х фреймов. Обнуляется при записи в этот бит ‘1’.
29	RC1	0	Бит BURST_COMPLETE – бит устанавливается в единицу по завершении выдачи блока данных сжатого звука. Обнуляется при записи в этот бит ‘1’.
28-24	RO	0	Поле зарезервировано. Читается нулями.
23	RC1	0	CS_HALF – бит устанавливается при вычитывании половины буфера CS. Прерывание сигнализирует о том, что можно подгружать данные в высвободившуюся часть буфера (3 слова по 32 разряда). Бит сбрасывается записью в него '1'.
22	RC1	0	UD_HALF – бит устанавливается при вычитывании половины буфера UD. Прерывание сигнализирует о том, что можно подгружать данные в высвободившуюся часть буфера (3 слова по 32 разряда). Бит сбрасывается записью в него '1'.
21-20	RO	0	Поле зарезервировано. Читается нулями.
19	RC1	1	FIFO_EMPTY – прерывание по пустоте FIFO аудиоданных. Бит устанавливается при опустошении FIFO аудиоданных. Бит сбрасывается записью в него '1'.
18	RC1	0/1	FIFO_LEVEL – прерывание по границе заполненности FIFO. Устанавливается в единицу, при опустошении FIFO до границы, заданной полем PROG_LEVEL. После сброса бит находится в состоянии ‘0’ при PROG_LEVEL равном “11”, и в состоянии ‘1’ – в остальных случаях. Бит сбрасывается записью в него '1'.
17	RC1	0	FIFO_OVERFLOW – переполнение FIFO аудиоданных. Бит устанавливается в ‘1’ при записи в полное FIFO. При этом новые записываемые данные теряются. Бит сбрасывается записью в него ‘1’.
16	RC1	0	FIFO_UNDERFLOW – бит устанавливается при чтении из пустого FIFO (читаемые в этом случае данные, повторяют предыдущие). Бит сбрасывается записью в него ‘1’.
15	RO	1	FIFO_EMPTY_STAT – бит статуса FIFO аудиоданных. Находится в единице при пустом FIFO аудиоданных.
14	RO	0/1	FIFO_LEVEL_STAT – бит статуса FIFO аудиоданных. Находится в единице, если заполненность FIFO меньше границы, заданной полем PROG_LEVEL.
13-0	RO	0	Поле зарезервировано. Читается нулями.

REG_SPDIF_INT_EN – регистр разрешения прерываний

Смещение x10, обращение только словом

Данный регистр предназначен для выбора событий, по которым вырабатывается сигнал системного прерывания (таблица 1.583). Каждому биту-маске из регистра REG_SPDIF_INT_EN соответствует бит-прерывание из регистра REG_SPDIF_STAT. Вне зависимости от разрешения в регистре REG_SPDIF_STAT устанавливаются те или иные события, но лишь при наличии разрешения прерывания по этим событиям передаются на выходную линию interrupt SPDIF контроллера. Чтобы разрешить выработку системного прерывания по тому или иному событию, необходимо в бит регистра REG_SPDIF_INT_EN, соответствующего этому событию, записать ‘1’.

Таблица 1.583 – Разряды регистра разрешения прерываний (REG_SPDIF_INT_EN)

Биты	Доступ	После RESET	Описание
31	RW	0	CS_INT_EN – разрешение выработки прерывания по сигналу статуса CS_HALF
30	RW	0	UD_INT_EN – разрешение выработки прерывания по сигналу статуса UD_HALF
29	RW	0	FIFO_INT_EN – разрешение выработки прерывания по достижению программируемой заполненности FIFO (сигнал FIFO_LEVEL)
28	RW	0	EMPTY_INT_EN – разрешение выработки прерывания по пустому FIFO (сигнал FIFO_EMPTY)
27	RW	0	UNDERFLOW_INT_EN – разрешение выработки прерывания по сигналу ошибки FIFO_UNDERFLOW.
26	RW	0	OVERFLOW_INT_EN – разрешение выработки прерывания по переполнению FIFO аудиосэмплов. (сигнал FIFO_OVERFLOW).
25	RW	0	BLOCK_INT_EN – разрешение выработки прерывания по выдаче блока данных из 192-х фреймов.
24	RW	0	BURST_INT_EN – разрешение выработки прерывания по выдачи блока сжатых аудио данных.
23-0	RO	0	Поле зарезервировано. Читается и записывается нулями.

REG_NONLINEAR_PARAMS – регистр параметров блока сжатых аудио данных

Смещение x18, обращение только словом

Регистр REG_NONLINEAR_PARAMS предназначен для реализации аппаратной поддержки формирования стаффинг интервала (определенного количества нулевых данных) при передаче многоканального сжатого звука (таблица 1.584). В текущей версии звукового контроллера аппаратная поддержка формирования стаффинг интервала не доступна. В связи с этим запрещается писать в поле STUFFING_LEN значения отличные от нуля. Стаффинг интервалы реализуются в текущей версии звукового контроллера программно – то есть нулевые значения, которые необходимо передавать в течение данного интервала должны быть записаны в FIFO аудиоданных SPDIF контроллера.

Таблица 1.584 – Разряды регистра параметров блока сжатых аудио данных (REG_NONLINEAR_PARAMS)

Биты	Доступ	После RESET	Описание
31-16	RW	0	Поле STUFFING_LEN – размер интервала стаффинга (передачи нулевых данных) в фреймах. В текущей версии звукового контроллера запись в это поле значений отличных от нуля запрещена!
15-0	RW	0	Поле DATA_BURST_LEN – размер блока сжатых аудиоданных (задается в фреймах). В текущей версии звукового контроллера в это поле необходимо записывать 0 – при выборе режима IEC 60958 (SPDIF_MODE=0) и 1 при выборе режима IEC 61937 (SPDIF_MODE=1).

SPDIF_BUF_CS_1 – буфер статусной информации 1-ого канала (объем: шесть 32-х разрядных слова – 192 бита);

Смещение x20, доступна только запись 32-х разрядных слов. При чтении возвращает 0.

SPDIF_BUF_CS_2 – буфер статусной информации 2-ого канала (объем: шесть 32-х разрядных слова – 192 бита);

Смещение x28, доступна только запись 32-х разрядных слов. При чтении возвращает 0.

Буферы статусной информации заполняются пользователем единожды, перед включением звукового контроллера. Установленная пользователем статусная информация, то есть информация о свойствах передаваемого звука не может меняться при работающем SPDIF-контроллере. При желании воспроизводить звук с параметрами отличными от текущих,

необходимо выключить SPDIF контроллер, произвести его настройку на новый звуковой режим, заполнить буферы BUF_CS1, BUF_CS2 новой статусной информацией и снова включить контроллер. Таким образом, контроллер SPDIF не поддерживает передачу звука с меняющимися параметрами (например, с изменяющимся битрейтом).

SPDIF_BUF_UD – буфер пользовательских данных (объем: шесть 32-х разрядных слова – 192 бита);

Смещение x30, доступна только запись 32-х разрядных слов. При чтении возвращает 0.

Учитывая, что в большинстве случаев пользовательские данные (user data) в сабфреймах SPDIF не передаются, использование данного буфера в текущей версии звукового контроллера в полном объеме не поддерживается. Рекомендуется устанавливать управляющий бит UD_EN в ноль, при котором SPDIF контроллер не производит чтение из буфера BUF_UD, а непосредственно подставляет нули в бит-U каждого сабфрейма.

SPDIF_FIFO_AUDIOSAMPLES – FIFO аудиоданных (объем 32x32 – 1КБ).

Смещение x38, доступна только запись 32-х разрядных слов. При чтении возвращает 0.

FIFO аудиосэмплов SPDIF контроллера предназначено для промежуточного хранения звуковых данных, за счет чего обеспечивается бесперебойный вывод звуковой информации. FIFO аудиосэмплов доступен через шину AXI как для программиста так и для AUDIO_DMA, но именно AUDIO_DMA звукового контроллера будет заполнять FIFO по мере необходимости.

1.3.7.1.5.4 Регистры контроллера AUDIO_DMA

Программируемые регистры AUDIO_DMA, доступны по системному интерфейсу управления (нестандартный интерфейс, описание смотри в пункте 1.4.3)

AUDIO_DMA содержит 34 32x разрядных регистра, для адресации которых требуются 8 разрядов адресной шины (адресное пространство 256 байт). В тоже время в текущей версии аудиоконтроллера анализируются 12 младших разрядов адреса на системном интерфейсе управления. В 11-ом разряде адреса регистра обязательно должна присутствовать единица, то есть регистры доступны смещением 12’h800. Данную ситуацию можно обойти, подавая на порты aud_dma_rg_addr_i[11:8] константу 4’b1000, а на порты aud_dma_rg_addr_i [7:0] соответствующие линии шины адреса системного интерфейса управления.

Таблица 1.585 – Регистры AUDIO_DMA

Наименование	Адрес	Описание
DMA_ENA_REG	12'h 000	Регистр разрешения
CH0_BASE0	12'h 004	Регистр базового адреса (начальный адрес) нулевой страницы в памяти для нулевого канала чтения
CH0_BASE1	12'h 008	Регистр базового адреса (начальный адрес) первой страницы в памяти для нулевого канала чтения
CH1_BASE0	12'h 00C	Базовый адрес для канала чтения 1 (аналогично каналу 0)
CH1_BASE1	12'h 010	Базовый адрес для канала чтения 1 (аналогично каналу 0)
CH2_BASE0	12'h 014	Базовый адрес для канала чтения 2 (аналогично каналу 0)
CH2_BASE1	12'h 018	Базовый адрес для канала чтения 2 (аналогично каналу 0)
CH3_BASE0	12'h 01C	Базовый адрес для канала чтения 3 (аналогично каналу 0)
CH3_BASE1	12'h 020	Базовый адрес для канала чтения 3 (аналогично каналу 0)
CH0_END0	12'h 024	Конечный адрес нулевой страницы для нулевого канала чтения.
CH0_END1	12'h 028	Конечный адрес первой страницы для нулевого канала чтения. (аналогично странице 0)
CH1_END0	12'h 02C	Конечный адрес для канала чтения 1 (аналогично каналу 0)
CH1_END1	12'h 030	Конечный адрес для канала чтения 1 (аналогично каналу 0)
CH2_END0	12'h 034	Конечный адрес для канала чтения 2 (аналогично каналу 0)

Продолжение таблицы 1.585

Наименование	Адрес	Описание
CH2_END1	12'h 038	Конечный адрес для канала чтения 2 (аналогично каналу 0)
CH3_END0	12'h 03C	Конечный адрес для канала чтения 3 (аналогично каналу 0)
CH3_END1	12'h 040	Конечный адрес для канала чтения 3 (аналогично каналу 0)
SLV0_BASE	12'h 044	Адрес ведомого устройства для канала записи 0
SLV1_BASE	12'h 048	Адрес ведомого устройства для канала записи 1
SLV2_BASE	12'h 04C	Адрес ведомого устройства для канала записи 2
SLV3_BASE	12'h 050	Адрес ведомого устройства для канала записи 3
CH0_TRW	12'h 054	Количество 32х разрядных слов, для записи в ведомое устройство канала записи 0.
CH1_TRW	12'h 058	Количество 32х разрядных слов, для записи в ведомое устройство канала записи 1.
CH2_TRW	12'h 05C	Количество 32х разрядных слов, для записи в ведомое устройство канала записи 2.
CH3_TRW	12'h 060	Количество 32х разрядных слов, для записи в ведомое устройство канала записи 3.
SLV_OVRH	12'h 064	Задает количество слов для прекращения запись в ведомое устройство.
SLV0_BSIZE	12'h 068	Размер буфера устройства ведомого в канале 0
SLV1_BSIZE	12'h 06C	Размер буфера устройства ведомого в канале 1
SLV2_BSIZE	12'h 070	Размер буфера устройства ведомого в канале 2
SLV3_BSIZE	12'h 074	Размер буфера устройства ведомого в канале 3
AXI_PARAM	12'h 078	Параметры шины AXI
AUD_DMA_INT_MASK	12'h 07C	Регистр маски прерываний
AUD_DMA_INT	12'h 080	Регистр состояния прерываний
AUD_DMA_BUF_STATUS	12'h 084	Состояние внутренних буферов

DMA_ENA_REG – регистр управления AUDIO_DMA

Смещение x00, обращение только словом.

Данный регистр предназначен для управления включением/выключением 4-х каналов AUDIO_DMA , а также для разрешения переключения страниц в памяти (таблица 1.586).

Таблица 1.586 – Разряды регистра управления AUDIO_DMA (DMA_ENA_REG)

Биты	Доступ	После RESET	Описание
31-12	RO	0	Поле зарезервировано. При чтении возвращает ноль.
11	RW	0	Ch3_rd_sw_ena – разрешение переключения страниц в памяти для канала 3 (аналогично Ch0_rd_sw_ena)
10	RW	0	Ch2_rd_sw_ena – разрешение переключения страниц в памяти для канала 2 (аналогично Ch0_rd_sw_ena)
9	RW	0	Ch1_rd_sw_ena – разрешение переключения страниц в памяти для канала 1 (аналогично Ch0_rd_sw_ena)
8	RW	0	Ch0_rd_sw_ena – разрешение переключения страниц в памяти для канала 0: ‘1’ – следующая страница в памяти готова, переключение разрешено. По завершению чтения данных из текущей страницы, контролер AUDIO_DMA сразу начнет читать данные из дугой; ‘0’ – переключение не производится, AUDIO_DMA ожидает готовности данных во внешней памяти.
7-5	RO	0	Поле зарезервировано. При чтении возвращает ноль.
4	RW	0	i2s_spdif – бит определяет с каким из контроллеров работает AUDIO_DMA: ‘0’ – I2S ; ‘1’ – SPDIF.
3	RW	0	Ch3_active – разрешение функционирования канала 3 DMA:
2	RW	0	Ch2_active – разрешение функционирования канала 2 DMA:
1	RW	0	Ch1_active – разрешение функционирования канала 1 DMA:
0	RW	0	Ch0_active – разрешение функционирования канала 0 DMA: ‘0’ – канал деактивирован; ‘1’ – канал активен.

В таблице 1.587 представлен список регистров, предназначенный для задания 32-х разрядных базовых адресов, по которым AUDIO_DMA будет обращаться к внешним памятям и памятям контроллеров I2S, SPDIF. Базовый адрес задается в формате little endian (0-й бит – наименее значащий). Все регистры доступны на чтение-запись (RW). Базовые адреса начала и конца страницы для чтения звуковых данных должны быть выровнены по границе 64х разрядного слова.

Таблица 1.587 – Регистры базовых адресов AUDIO_DMA

Регистр	Смещение	Доступ	Описание
CH0_BASE0	0x004	RW	Регистр базового адреса (начальный адрес) нулевой страницы в памяти для нулевого канала чтения, по окончанию считывания из памяти данных страницы хранимой по адресу CH0_BASE1 и при равенстве ch0_rd_sw_ena == 1 происходит переключение AUDIO_DMA на данный адрес, и данный адрес становится текущим, а CH0_BASE1 – следующим
CH0_BASE1	0x008	RW	Регистр базового адреса (начальный адрес) первой страницы в памяти для нулевого канала чтения, работа AUDIO_DMA начинается с данной страницы, в начальный момент времени данный адрес является текущим, при этом адрес CH0_BASE0 является следующим
CH1_BASE0	0x00c	RW	Базовый адрес для канала чтения 1 (аналогично каналу 0)
CH1_BASE1	0x010	RW	Базовый адрес для канала чтения 1 (аналогично каналу 0)
CH2_BASE0	0x014	RW	Базовый адрес для канала чтения 2 (аналогично каналу 0)
CH2_BASE1	0x018	RW	Базовый адрес для канала чтения 2 (аналогично каналу 0)
CH3_BASE0	0x01c	RW	Базовый адрес для канала чтения 3 (аналогично каналу 0)
CH3_BASE1	0x020	RW	Базовый адрес для канала чтения 3 (аналогично каналу 0)
CH0_END0	0x024	RW	Конечный адрес нулевой страницы для нулевого канала чтения. Адрес, следующий за последним словом в памяти страницы ноль. AUDIO_DMA ведет чтение в диапазоне адресов между CH0_BASE0 (включительно) и CH0_END0 (не включительно).
CH0_END1	0x028	RW	Конечный адрес первой страницы для нулевого канала чтения (аналогично странице 0)
CH1_END0	0x02c	RW	Конечный адрес для канала чтения 1 (аналогично каналу 0)
CH1_END1	0x030	RW	Конечный адрес для канала чтения 1 (аналогично каналу 0)
CH2_END0	0x034	RW	Конечный адрес для канала чтения 2 (аналогично каналу 0)
CH2_END1	0x038	RW	Конечный адрес для канала чтения 2 (аналогично каналу 0)
CH3_END0	0x03c	RW	Конечный адрес для канала чтения 3 (аналогично каналу 0)
CH3_END1	0x040	RW	Конечный адрес для канала чтения 3 (аналогично каналу 0)
SLV0_BASE	0x044	RW	Адрес ведомого устройства для канала записи 0
SLV1_BASE	0x048	RW	Адрес ведомого устройства для канала записи 1
SLV2_BASE	0x04c	RW	Адрес ведомого устройства для канала записи 2
SLV3_BASE	0x050	RW	Адрес ведомого устройства для канала записи 3

В таблице 1.588 представлен список регистров, предназначенный для задания 32-х разрядных параметров для AUDIO_DMA. Значения задаются в формате little endian (0-й бит – наименее значащий). Все регистры доступны на чтение-запись (RW).

Таблица 1.588 – 32-х разрядные параметры для AUDIO_DMA

Регистр	Смещение	Доступ	Описание
CH0_TRW	0x054	RW	Количество 32х разрядных слов, которое AUDIO_DMA должно записать в ведомое устройство канала записи 0. (Данное количество слов должно соответствовать заданным параметрам для канала чтения CH0_BASE0 - CH0_END0 и не может отличаться для разных страниц)
CH1_TRW	0x058	RW	Количество 32х разрядных слов, которое AUDIO_DMA должно записать в ведомое устройство канала записи 1. (аналогично каналу 0)
CH2_TRW	0x05c	RW	Количество 32х разрядных слов, которое AUDIO_DMA должно записать в ведомое устройство канала записи 2. (аналогично каналу 0)
CH3_TRW	0x060	RW	Количество 32х разрядных слов, которое AUDIO_DMA должно записать в ведомое устройство канала записи 3. (аналогично каналу 0)
SLV_OVRH	0x064	RW	Задает количество слов для прекращения записи в ведомое устройство. AUDIO_DMA прекращает запись в ведомое устройство, когда уровень заполнения буфера, ведомого становится равным или больше разницы SLVx_BSIZE - SLV_OVRH. Параметр задается для всех каналов.
SLV0_BSIZE	0x068	RW	Размер буфера устройства ведомого в канале 0 (кол-во 32-х разрядных слов - 1).
SLV1_BSIZE	0x06c	RW	Размер буфера устройства ведомого в канале 1 (кол-во 32-х разрядных слов - 1).
SLV2_BSIZE	0x070	RW	Размер буфера устройства ведомого в канале 2 (кол-во 32-х разрядных слов - 1).
SLV3_BSIZE	0x074	RW	Размер буфера устройства ведомого в канале 3 (кол-во 32-х разрядных слов - 1).

AXI_PARAM – регистр параметров транзакций на AXI шине

Смещение x078, обращение только словом.

Данный регистр предназначен для задания параметров AXI шины, которые будет использовать AUDIO_DMA при формировании транзакций чтения-записи звуковых данных.

В таблице 1.589 представлены разряды регистра AXI_PARAM.

Таблица 1.589 – Разряды регистра параметров AXI шины (AXI_PARAM)

Биты	Доступ	После RESET	Описание
31-29	RW	0	AWPROT
28-26	RO	0	Поле зарезервировано. При чтении возвращает ноль.
25-24	RW	0	AWLOCK
23-20	RW	0	WID
19	RO	0	Поле зарезервировано. При чтении возвращает ноль.
18-16	RW	0	ARPROT
15-14	RO	0	Поле зарезервировано. При чтении возвращает ноль.
13-12	RW	0	ARLOCK
11-8	RW	0	ARID
3-0	RW	0	ARLEN - запись любого значения на работу блока не влияет

AUD_DMA_INT_MASK – регистр масок прерываний.

Смещение x07c, обращение только словом.

Данный регистр предназначен для разрешения прерываний по ряду событий, отображаемых в регистре состояния прерываний. Если в соответствующем разряде регистра масок установлена ‘1’ – прерывание разрешено, ‘0’ – прерывание запрещено. Маска влияет только на формирование сигнала прерывания, но не на запись признаков в регистр состояния прерываний. Разряды регистра масок прерываний описаны в таблице 1.590.

Таблица 1.590 – Разряды регистра масок прерываний (AUD_DMA_INT_MASK)

Биты	Доступ	После RESET	Описание
31-20	RO	0	Поле зарезервировано. При чтении возвращает ноль.
19	RW	0	ch3_stoped_mask
18	RW	0	ch2_stoped_mask
17	RW	0	ch1_stoped_mask
16	RO	0	ch0_stoped_mask
15-12	RO	0	Поле зарезервировано. При чтении возвращает ноль.
11	RW	0	bresp_error_mask
10	RW	0	wid_error_mask
9	RW	0	rresp_error_mask
8	RW	0	arid_error_mask
7-5	RO	0	Поле зарезервировано. При чтении возвращает ноль.
3	RW	0	ch3_wr_end_mask
2	RW	0	ch2_wr_end_mask
1	RW	0	ch1_wr_end_mask
0	RW	0	ch0_wr_end_mask

AUD_DMA_INT – регистр состояния прерываний

Смещение x080, обращение только словом.

В случае возникновения прерывания оно фиксируется в соответствующем разряде данного регистра (таблица 1.591). При возникновении прерывания оно записывается в регистр и сохраняется до его программного сброса. Для того чтобы сбросить прерывание необходимо в соответствующий разряд регистра состояния прерываний записать единицу, запись нуля в регистр состояния прерываний не оказывает никакого действия на хранимые данные.

Таблица 1.591 – Разряды регистра состояния прерываний (AUD_DMA_INT)

Биты	Доступ	После RESET	Описание
31-20	RO	0	Поле зарезервировано. При чтении возвращает ноль.
19	RC1	0	ch3_stoped – бит устанавливается в ‘1’ после того как канал 3 деактивирован (ch3_active = 0) и по каналу 3 звуковому контроллеру переданы все данные текущей страницы.
18	RC1	0	ch2_stoped – бит устанавливается в ‘1’ после того как канал 2 деактивирован (ch3_active = 0) и по каналу 2 звуковому контроллеру переданы все данные текущей страницы.
17	RC1	0	ch1_stoped – бит устанавливается в ‘1’ после того как канал 1 деактивирован (ch3_active = 0) и по каналу 1 звуковому контроллеру переданы все данные текущей страницы.
16	RC1	0	ch0_stoped – бит устанавливается в ‘1’ после того как канал 0 деактивирован (ch3_active = 0) и по каналу 0 звуковому контроллеру переданы все данные текущей страницы.
15-12	RO	0	Поле зарезервировано. При чтении возвращает ноль.
11	RC1	0	bresp_error
10	RC1	0	wid_error
9	RC1	0	rresp_error
8	RC1	0	arid_error
7-4	RO	0	Поле зарезервировано. При чтении возвращает ноль.
3	RC1	0	ch3_wr_end – бит устанавливается в единицу по завершению передачи звуковому контроллеру данных очередной страницы по каналу 3. Бит сбрасывается записью в него ‘1’.
2	RC1	0	ch2_wr_end – бит устанавливается в единицу по завершению передачи звуковому контроллеру данных очередной страницы по каналу 2. Бит сбрасывается записью в него ‘1’.

Продолжение таблицы 1.591

Биты	Доступ	После RESET	Описание
1	RC1	0	ch1_wr_end – бит устанавливается в единицу по завершению передачи звуковому контроллеру данных очередной страницы по каналу 1. Бит сбрасывается записью в него ‘1’.
0	RC1	0	ch0_wr_end – бит устанавливается в единицу по завершению передачи звуковому контроллеру данных очередной страницы по каналу 0. Бит сбрасывается записью в него ‘1’.

AUD_DMA_BUF_STATUS – регистр статуса

Смещение x07c, обращение только словом.

Регистр предназначен для отображения состояния внутренних буферов контроллера AUDIO_DMA (таблица 1.592). Регистр доступен только на чтение.

Таблица 1.592 – Разряды регистра состояния буферов AUDIO_DMA (AUD_DMA_BUF_STATUS)

Биты	Доступ	После RESET	Описание
31-24	RO	0	количество слов в 3 буфере
23-16	RO	0	количество слов в 2 буфере
15-8	RO	0	количество слов в 1 буфере
7-0	RO	0	количество слов в 0 буфере

Контроллер AUDIO_DMA имеет четыре канала, каждый из которых может быть независимо сконфигурирован. Запуск каналов контроллера на передачу данных осуществляется записью единичного значения в соответствующий признак ch(X)_active. До запуска контроллера необходимо сконфигурировать все каналы, которые будут функционировать в данном режиме работы.

Для задания области памяти и количество слов, которые надо считать из памяти используются следующие параметры:

CH(X)_BASE0 и CH(X)_BASE1 - базовые (начальные) адреса для текущей и следующей страниц, читаемых из памяти. После сброса начинает работать с _BASE1

CH(X)_END0 и CH(X)_END1 - конечные адреса для текущей и следующей страниц, читаемых из памяти. Контроллер будет осуществлять чтение данных из диапазона CH(X)_BASE(Y) до CH(X)_END(Y) (не включая CH(X)_END(Y)). Рекомендуется формировать страницы размером не менее 256 байт.

Для определения параметров области памяти, куда ведется копирование, необходимо задать следующие параметры:

SLV(X)_BASE – адрес, в который будет вестись запись данных (контроллер рассчитан на запись данных в память типа FIFO (запись ведется по одному адресу единичными транзакциями на шине AXI)

CH(X)_TRW - количество 32х разрядных слов, которые необходимо записать в ведомое устройство (значение задается уменьшенным на единицу). Количество слов должно соответствовать указанному диапазону чтения CH(X)_BASE(Y) : CH(X)_END(Y). При этом следует учитывать, что чтение ведется в 64х разрядных словах, а запись в 32х разрядных.

SLV(X)_BSIZE - указывает размер буфера в ведомом устройстве, размер указывается в 32х разрядных словах минус один, т.е. если размер буфера равен 32 слова в регистр необходимо записать значение 31.

SLV_OVRH - задает количество слов для прекращения записи в ведомое устройство, как только выполняется условие SLV(X)_BSIZE - (количество слов записанных в буфер – отслеживается аппаратно) < SLV_OVRH контроллер приостанавливает запись в ведомое

устройство. Размер SLV_OVRH определяется из параметров системы – количества уровней иерархии в коммутационной среде и регистров на пути данных.

1.3.7.2 Использование по назначению аудиоконтроллера

Работа с аудиоконтроллером состоит из нескольких фаз: сброс аудиоконтроллера – конфигурирование аудиоконтроллера – запуск аудиоконтроллера – отслеживание хода работы по прерываниям и статусу – выключение аудиоконтроллера.

1.3.7.2.1 Сброс аудиоконтроллера

После включения питания необходимо произвести сброс аудиоконтроллера.

Сброс аудиоконтроллера осуществляется путем сброса его трех основных блоков: AUDIO_DMA, SPDIF, I2S. Сброс аудиоконтроллера может быть либо программным, либо аппаратным.

Аппаратный сброс осуществляется посредством подачи низкого уровня сигнала на порты i2s_sys_reset_i, spdif_sys_reset_i, aud_dma_sys_reset_i. Активный уровень аппаратного сброса должен длиться не менее восьми тактов системной частоты.

Программный сброс аудиоконтроллера осуществляется через его программируемые регистры. Контроллер I2S сбрасывается записью ‘1’ в бит PRS регистра REG_I2S_CTRL0.

Контроллер SPDIF сбрасывается записью ‘1’ в бит SPDIF_RESET регистра REG_SPDIF_CTRL.

Программный сброс длится несколько тактов системной частоты, во время которых доступ к регистрам звуковых контроллеров запрещен.

После прохождения операции сброса значение битов сброса в соответствующих регистрах управления автоматически обнуляется. Управляющие схемы звуковых контроллеров при этом переходят в исходное состояние.

Программный сброс AUDIO_DMA в текущей версии аудиоконтроллера не предусмотрен. В тоже время сброс AUDIO_DMA может быть осуществлен из вне: сигнал программного сброса может вырабатываться внешней по отношению к аудиоконтроллеру схемой. Затем этот сигнал подмешивается к сигналу аппаратного сброса и подается на порт aud_dma_sys_reset_i.

Программный сброс аудиоконтроллера рекомендуется осуществлять каждый раз перед началом работы с ним.

Рекомендуется после подачи сигнала системного или программного сброса провести программную инициализацию всех регистров видеоконтроллера.

1.3.7.2.2 Настройка аудиоконтроллера

Конфигурирование аудиоконтроллера на требуемый режим работы производится перед его запуском. Изменение конфигурации аудиоконтроллера по ходу его работы (за исключением управления несколькими параметрами) может привести к некорректной работе аудиоконтроллера.

Аудиоконтроллер единовременно может выводить звуковую информацию только по одному из звуковых интерфейсов: I2S или SPDIF. Для этого производится конфигурация регистров звукового контроллера, которому предстоит выводить информацию.

Конфигурирование I2S контроллера

Для правильного конфигурирования I2S контроллера, необходимо:

- Указать формат входных данных. В регистре REG_I2S_CTRL0 в поле BR указать порядок байт в 32-х разрядном слове данных, поступающем в контроллер I2S по AXI шине. В поле CH указать количество сэмплов, передаваемых в одном 32-х разрядном слове. В поле RES указать разрядность сэмплов;

- Разрешить прерывания, которые планируется обрабатывать. Учитывая, что с I2S контроллером будет работать AUDIO_DMA, рекомендуется разрешить прерывание лишь по пустоте внутренних буферов звуковых данных I2S котроллера (биты Ex_en регистра REG_I2S_CTRL0). По этому прерыванию можно будет узнать, когда I2S контроллер завершит выдачу всех данных.
- Настроить I2S контроллер на частоту выдачи звуковой информации (частоту дискретизации звука). В поле SCLK_RATIO регистра REG_I2S_CTRL1 должен быть записан коэффициент деления внешней звуковой частоты, подаваемой на порт i2s_mclk_i аудиоконтроллера (см п.1.4.6).

Конфигурирование SPDIF контроллера

Для правильного конфигурирования I2S контроллера, необходимо:

- Указать формат входных данных. В поле BYTE_ORDER регистра REG_SPDIF_CTRL указать порядок байт в 32-х разрядном слове данных поступающем по AXI шине. В поле SPDIF_DATA указать количество сэмлов передаваемых в одном 32-х разрядном слове.
- В поле SPDIF_MODE указать вид звуковых данных, которые будут передаваться по SPDIF интерфейсу: линейные или сжатые
- В зависимости от вида передаваемых данных настроить регистр REG_NONLINEAR_PARAMS. Записать нули в оба поля при линейном звуке. При сжатом звуке – записать 1 в поле DATA_BURST_LEN.
- При желании использовать аппартную поддержку вывода линейного моно звкуа записать ‘1’ в бит SPDIF_CHANNELS.
- Запретить передачу пользовательской информации в бите-U звуковых сабфреймов записав ‘0’ в бит UD_EN регистра REG_SPDIF_CTRL.
- Разрешить прерывания, которые планируется обрабатывать. Учитывая, что с SPDIF контролером будет работать AUDIO_DMA, рекомендуется разрешить прерывание лишь по пустоте буфера звуковых данных SPDIF котроллера (FIFO_INT_EN регистра REG_SPDIF_INT_EN). По этому прерыванию можно будет узнать, когда SPDIF контроллер завершит выдачу всех данных. Для контроля корректности заполнения буфера рекомендуется разрешить выработку прерываний по событиям undeflow, overflow.
- Настроить SPDIF контроллер на частоту выдачи звуковой информации (частоту дискретизации). В поле FREQ_BASE регистра управления REG_SPDIF_CTRL должен быть записан коэффициент деления внешней звуковой частоты, подаваемой на порт spdif_mclk_i аудиоконтроллера до опорной, а в поле FREQ_DIV коэффициент деления опорной частоты для получения желаемой частоты дискретизации (см п.1.4.6).
- Записать в буферы BUF_CS1, BUF_CS2 контролера SPDIF статусную информацию канала о свойствах передаваемого звукового потока (в соответствии со стандартами IEC 60958, IEC 61937). В каждый буфер необходимо записать по шесть 32-х разрядных слова (в формате little endian) статусной информации.

После настройки соответствующего звукового контроллера необходимо настроить AUDIO_DMA.

Конфигурирование AUDIO_DMA:

Для правильного конфигурирования котроллера AUDIO_DMA, необходимо:

- В поле i2s_spdif регистра DMA_ENA_REG указать с каким из звуковых контроллеров будет работать AUDIO_DMA

- В регистре указать объем буфера FIFO звукового контроллера. При работе с SPDIF только для нулевого канала. При работе с I2S от одного до четырех – в зависимости от канальности звука.
- В регистре SLVX_BASE указать адрес буфера(-ов) звукового контролера;
- В регистрах SLVX_BSIZE указать, за сколько слов до заполнения буфера ведомого звукового контроллера прекращать его наполнение звуковыми данными. Таким образом, буфер звукового контроллера не будет заполняться выше определенной отметки. Рекомендуемое значении для параметра – 3.
- В регистрах CHX_BASE1, CHX_END1 указать адреса начала и конца страницы с готовыми звуковыми данными. При включении AUDIO_DMA начнет читать данные именно этой страницы. Если вторая страница звуковых данных уже готова или известен ее адрес, то его так же можно сразу указать в регистрах CHX_BASE0, CHX_END0. Стоит учитывать, что адреса задаются выровненными по границе 8 байт.
- Задать размер страницы в регистре. Размер страницы не должен быть меньше 256 байт, и должен находиться в соответствии с адресами начала и конца страницы.
- Задать параметры транзакций формируемых AUDIO_DMA на AXI шине. Параметры задаются в регистре AUD_DMA_AXI_PARAM. Рекомендуется оставить значения в регистре, которые находятся там после сброса.
- Разрешить все прерывания в регистре AUD_DMA_INT_MASK

1.3.7.2.3 Включение аудиоконтроллера

После настройки аудиоконтроллера его необходимо включить. Делается это в два этапа: сначала включается AUDIO_DMA затем один из звуковых контроллеров.

Включение AUDIO_DMA

Для активации каналов AUDIO_DMA необходимо в его регистре управления DMA_ENA_REG в биты chX_rd_sw_ena и chX_active записать ‘1’. При работе с SPDIF контроллером следует запускать только нулевой канал. При работе с I2S контролером одновременно могут быть запущены до 4-х каналов.

Как только канал X AUDIO_DMA включен, он начинает читать 64-х разрядные данные из памяти начиная с адреса OSDX_BASE1. Читаемые данные он затем передает в буферы ведомого звукового контроллера. Как только звуковой контроллер располагает необходимыми для вывода данными его можно активировать.

Включение I2S контроллера

Активация I2S контроллера осуществляется записью ‘1’ в бит EN регистра REG_I2S_CTRL1.

В случае, когда контролер I2S запускается до или одновременно с запуском AUDIO_DMA на интерфейс I2S будут выводиться нулевые звуковые сэмплы (тишина) до тех пор, пока буферы I2S пусты. Как только буферы получат звуковые данные от AUDIO_DMA начнется их вывод. Такой способ запуска контроллера I2S является некорректным, поскольку возможна рассинхронизация данных передаваемых по параллельным каналам. Звуковые данные записываются в буферы последовательно, таким образом, звуковой контроллер может начать выдавать данные в то время, когда в одном из буферов данные уже имеются, а в других еще нет. Таким образом, на одном канале I2S интерфейса появятся звуковые данные, а по другим будут передаваться нулевые сэмплы.

Правильный способ включения I2S контроллера – это дождаться появления звуковых данных во всех буферах, а только затем запустить контроллер I2S. Появление данных в буферах можно обнаружить постоянно, опрашивая поля CTW_X регистров REG_I2S_STAT0, REG_I2S_STAT1.

Активация SPDIF контроллера

Активация SPDIF контроллера осуществляется записью ‘1’ в бит SPDIF_START регистра REG_SPDIF_CTRL.

SPDIF контролер нельзя включать до тех пор, пока в его буфере не появятся данные. При попытке сделать это, контроллер автоматически выключается с ошибкой underflow (чтение из пустого буфера). Появление данных в буфере SPDIF контроллера можно обнаружить постоянно, опрашивая поле FIFO_EMPTY_STAT регистра REG_SPDIF_STAT.

1.3.7.2.4 Ход работы аудиоконтроллера

После того как все компоненты аудиоконтроллера включены – происходит цикличная работа контроллера. AUDIIO_DMA считывает данные из текущей страницы в памяти и передает их в буфер звукового контроллера. Буферы звукового контроллера заполняется до определенной границы, после чего AUDIO_DMA ожидает момент, когда из буфера звукового контролера будут прочитаны очередные данные, а затем снова дополняет его до границы. Таким образом, скорость чтения контроллером AUDIO_DMA данных из памяти в среднем будет такой же, как скорость вывода звуковых данных. Это означает что процессор гарантировано успеет подготовить новую страницу звуковых данных до того, как текущая будет прочитана. Поэтому нет необходимости сбрасывать бит разрешения переключения страниц chX_rd_sw_ena, необходимо лишь назначать AUDIO_DMA базовые адреса новой страницы. Следить за ходом работы AUDIO_DMA можно читая статусную информацию из соответствующих регистров (AUD_DMA_BUF_STATUS) и получая прерывания. Существуют два вида прерывания – прерывание по прочтению из памяти текущей страницы и по записи в буфер звукового контроллера данных текущей страницы.

Звуковые контроллеры I2S, SPDIF в свою очередь последовательно выводят звуковые данные, преобразуя их в битовые потоки. Критерием нормального хода работы звуковых контролеров служит отсутствие прерываний по пустоте буферов. То есть во время работы буферы звуковых контроллеров не должны опустошаться. К тому же у SPDIF контролера есть прерывания по переполнению и нехватке данных в FIFO (overflow, underflow). Возникновение перечисленных выше прерываний не должно происходить во время передачи звуковых данных. Установка этих прерываний может свидетельствовать о некорректной настройке аудиоконтроллера, либо о проблемах на шине AXI.

1.3.7.2.5 Выключение аудиоконтроллера

После того как получено прерывание от канала AUDIO_DMA о том, что он прочитал предпоследнюю страницу данных в памяти и перешел на чтение последней, необходимо запретить для этого канала переключение страниц, установив соответствующий бит chX_rd_sw_ena в ноль. После того как канал AUDIO_DMA прочитает последнюю страницу, он выдаст соответствующее прерывание и перейдет в режим ожидания (поскольку ему запрещено переключать страницы). Теперь можно выключить канал контроллера AUDIO_DMA, записав в бит chX_active ноль. Контроллер AUDIO_DMA считается выключенным, когда все его каналы остановлены.

Несмотря на то что AUDIO_DMA остановлен, звуковому контроллеру потребуется еще некоторое время на выдачу данных из его буфера(-ов).

I2S контроллер после выдачи всех данных выработает прерывание о пустоте буферов FIFO, к тому же в поле CTW_X будет указано, что буфер содержит 0 аудиосэмлов. По этим событиям следует произвести выключение I2S контроллера путем записи ‘0’ в бит EN регистра REG_I2S_CTRL1. Стоит учитывать, что после опустошения буфера в течение некоторого времени контроллер I2S будет выводить последний сэмпл, таким образом, выключать контроллер I2S стоит с некоторой задержкой после опустошения FIFO. После выдачи последнего сэмпла и до выключения контроллера на интерфейс I2S будут подаваться нулевые сэмплы.

SPDIF контроллер после выдачи всех данных из буфера вначале сформирует прерывание по пустоте его FIFO, а затем прерывание underflow (чтение из пустого FIFO). По этому событию он автоматически выключится, установив бит SPDIF_START в ноль, и произведя программный сброс. В общем же случае контроллер SPDIF останавливается путем его программного сброса, то есть записью ‘1’ в бит SPDIF_RESET регистра REG_SPDIF_CTRL. Таким образом, возникновение прерывания underflow считается нормальным при завершении работы контролера SPDIF, возникновение этого прерывания во время работы – свидетельствует об ошибке.

При нормальном завершении работы аудиоконтроллер возвращается в исходное состояние: он готов к настройке на новый режим работы и включению. В тоже время настоятельно рекомендуется произвести программный сброс аудиоконтроллера до возобновления работы с ним.

Внимание! *Стоит проявлять осторожность при работе с прерываниями звуковых контролеров I2S и SPDIF и по возможности отказаться от нее в пользу работы со статусом. Как уже отмечалось, звуковые контролеры работают только с одиночными транзакциями на шине AXI, причем порядок записываемых в звуковой контроллер данных должен соответствовать порядку адресов, по которым к нему обращаются. Работа с прерываниями требует сброса битов прерываний в соответствующих регистрах звуковых контроллеров. Это в свою очередь означает, что процессор должен записывать в регистры звуковых контроллеров некоторые значения. В тоже время AUDIO_DMA обращается к звуковым контроллерам, записывая в них аудиосэмплы. Таким образом, два ведущих устройства обращаются к одному ведомому по каналу записи AXI интерфейса. Коммутационная среда в этом случае должна строго следить за разделением адресов и данных транзакций от разных мастеров. В тоже время такое ограничение можно обойти, если работать со статусом звуковых контролеров – то есть постоянно опрашивать регистры состояния. В этом случае процессор будет обращаться к устройствам только на чтение, AUDIO_DMA – только на запись. Такая организация работы позволит избежать указанных выше ошибочных ситуаций.*

1.3.8 Подсистема контроллера внешней динамической памяти DDR3

Данная глава описывает подсистему контроллера внешней динамической памяти DDR3, которая предназначена для обеспечения доступа изделия во внешнюю память типа DDR3 SDRAM.

1.3.8.1 Структура подсистемы контроллера динамической памяти DDR3

В состав подсистемы контроллера динамической памяти DDR3 входят блоки, которые позволяют поддерживать DDR3 SDRAM протоколы обмена данных, адресов и контрольных сигналов. Структура DDR3 подсистемы приведена на рисунке 1.100.

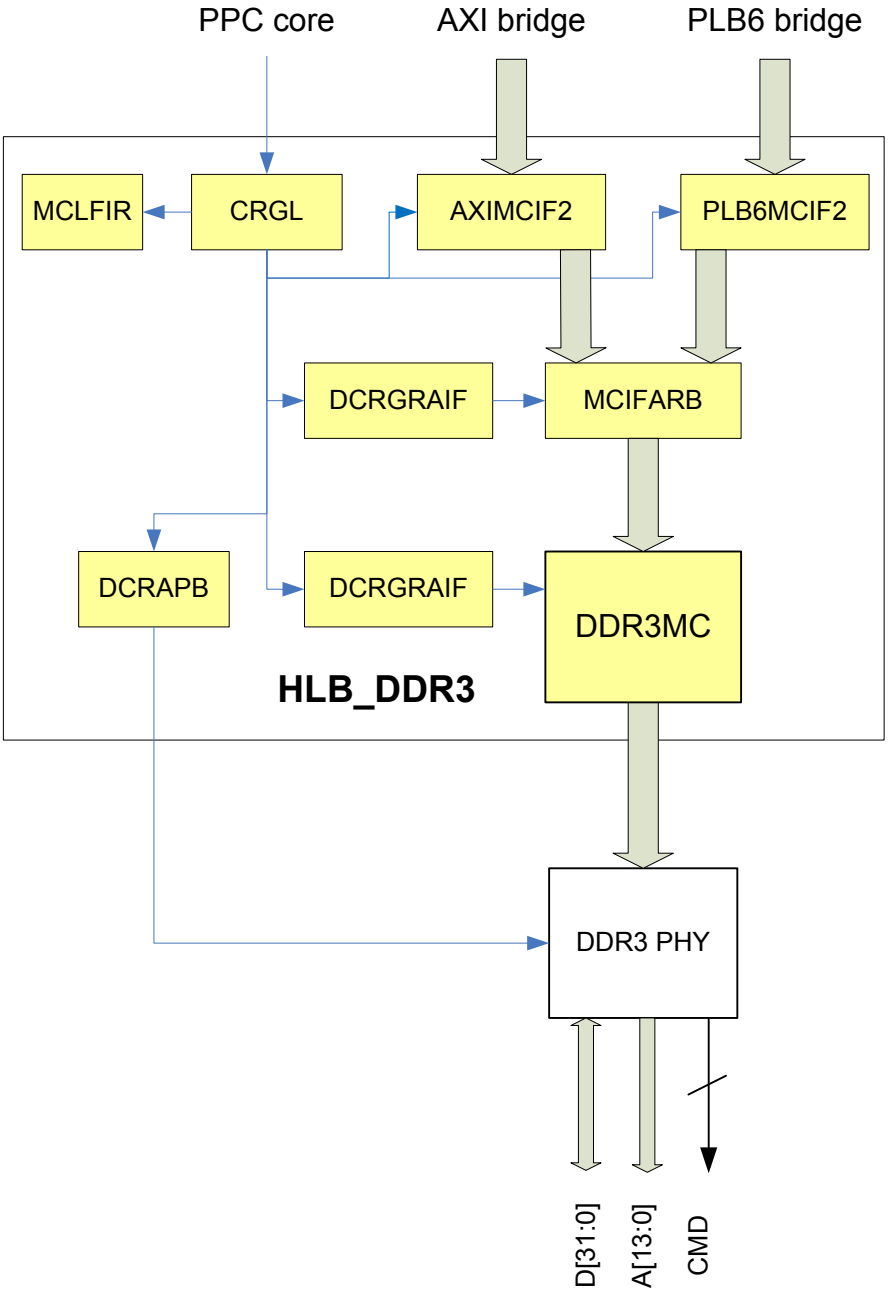


Рисунок 1.100 – Структура DDR3 подсистемы микросхемы

Подсистема включает в себя следующие блоки:

- контроллер памяти типа DDR3 SDRAM – DDR3MC;
- блок моста с шиной типа PLB6 – PLB6MCIF2;
- блок моста с шиной типа AMBA AXI – AXIMCIF2;

- блок шинного арбитра – MCIF2ARB;
- блоки, обеспечивающие управление подсистемой в целом и отдельными ее компонентами по шине управления типа DCR – CRGL, MCLFIR, DCRGRAIF.

Следует отметить, что DDR3 подсистема не включает в своем составе блоков физического интерфейса DDR3 PHY, так как эти блоки должны быть расположены на верхнем уровне иерархии микросхемы. В то же время, управление блоками DDR3 PHY осуществляется из данной подсистемы по интерфейсу APB. Используется блок моста с шины DCR на шину APB – DCRAPB.

1.3.8.2 Технические характеристики подсистемы контроллера динамической памяти DDR3

Контроллер динамической памяти имеет следующие характеристики:

- Тип поддерживаемой памяти - DDR3 SDRAM;
- Интерфейсный протокол обмена данных между DDR3MC и PHY DDR3 - DFI3.1;
- Частота шины - 400 МГц;
- Максимальная скорость передачи данных - 1600 мегабит в секунду;
- Разрядность внешней шины данных, бит – 32 (40 бит с ECC, кода коррекции ошибок);
- Разрядность внешней шины адреса - 16 бит;
- Количество сигналов выбора микросхемы (chip select) – 2 по 4 бита;
- Количество банков в микросхеме памяти – 16;

Программирование контроллера DDR3MC осуществляется процессорным ядром PowerPC. Перед тем, как производить обращения к внешней памяти, необходимо инициализировать контроллер DDR3MC.

1.3.8.3 Конфигурация подсистемы контроллера динамической памяти DDR3

Ниже описана выбранная конфигурация подсистемы контроллера динамической памяти DDR3. Данная конфигурация определялась путем установления параметров системы в выбранное значение:

- Определена разрядность внутренней шины данных типа MCIF2 - 128, (MCIF_DATA_WIDTH=128);
- Определена разрядность внутренней шины данных ECC типа MCIF2 - 16, (MCIF_ECC_WIDTH=16);
- Задан способ выдачи данных контроллером памяти DDR3MC из входной 128-битной шины данных и 16-битной шины данных ECC типа MCIF2 в выходную 32-битную шину данных и 8-битную шину данных ECC типа DFI: 4 канала по 32 бита для шины данных и 4 канала по 4 бита для ECC, (установлен DDR34LMC_MUX4_1MODE_ON);
- Интерфейс типа PLB6 асинхронно развязан с интерфейсом типа MCIF2 в блоке моста PLB6MCIF2, (установлен DPLB6MCIF2_ASYNC_CLOCKS);
- В контроллер динамической памяти DDR3 включена логика проверки кода коррекции ECC, (BUILD_IN_ECC=1);
- Установлено значение по умолчанию глубины команд - 4, (не активны DDR34LMC_Q_8_ENTRIES, PLB6MCIF2_Q_16_ENTRIES);
- Задан расширенный набор проверок для интерфейса типа MCIF с поддержкой контроля четности (установлен DDR34LMC_RAS_EN, MCIF2ARB4_RAS);

1.3.8.4 Регистровая модель подсистемы контроллера динамической памяти DDR3

После системного сигнала сброса все конфигурационные регистры должны быть настроены соответствующим образом, после чего подсистема готова к работе. Все конфигурационные регистры программируются прямым обращением по шине типа DCR (Direct Control Registers). 26-битная шина I_DCR_BASE_ADDR используется для задания базового адреса для осуществления доступа к конкретному конфигурационному регистру в общем DCR адресном пространстве.

1.3.8.4.1 Программно реализуемая модель блока AXIMCIF2

1.3.8.4.1.1 Список программно доступных регистров блока AXIMCIF2

В данном разделе описываются программно доступные регистры блока моста с шиной типа AMBA AXI – AXIMCIF2. Таблица 1. содержит список программно доступных регистров.

Таблица 1.569 – Список регистров блока AXIMCIF2

Название	Тип	Смещение относительно базового адреса	Выполняемая функция	Значение после инициализации
BESR	ЗП/ЧТ	x00, x01	Регистр состояния ошибки шины AXI	x0000_0000
ERRID	ЧТ	x02	Регистр состояния ошибки при обращении по ID	x0000_0000
AEARL	ЧТ	x03	Регистр состояния ошибки при обращении по младшим адресам	x0000_0000
AEARU	ЧТ	x04	Регистр состояния ошибки при обращении по старшим адресам	x0000_0000
AXIERRMSK	ЗП/ЧТ	x05	Регистр для маскирования ошибки, генерации прерываний	x0000_0000
AXIASYNC	ЗП/ЧТ	x06	Регистр для включения асинхронной логики	x0000_0000
AXICFG	ЗП/ЧТ	x07	Конфигурационный регистр AXI моста	x0000_0000
AXISTS	ЧТ	x09	Статусный регистр AXI моста	x8000_0000
MR0CF	ЗП/ЧТ	x10	Конфигурационный регистр памяти RANK0	x0000_0000
MR1CF	ЗП/ЧТ	x11	Конфигурационный регистр памяти RANK1	x0000_0000
MR2CF	ЗП/ЧТ	x12	Конфигурационный регистр памяти RANK2	x0000_0000
MR3CF	ЗП/ЧТ	x13	Конфигурационный регистр памяти RANK3	x0000_0000
RID	ЧТ	x20	Регистр ID контроля	x0000_0101

1.3.8.4.1.2 Описание программно доступных регистров блока AMBA AXI

Таблица 1.570 – Регистр состояния ошибки шины AXI – BESR

Биты	Название	Значение после инициализации	Выполняемая функция
31	MCIF_RDERR	0	Ошибка чтения данных по шине MICF. 1: ошибка, 0: ошибки нет
30	MCIF_PARERR	0	Ошибка четности по шине MICF. 1: ошибка, 0: ошибки нет
29	MCIF_WRERR	0	Ошибка записи данных по шине MICF. 1: ошибка, 0: ошибки нет
28	MCIF_RMWERR	0	Ошибка маскирования данных по шине MICF. 1: ошибка, 0: ошибки нет

Продолжение таблицы 1.571

Биты	Название	Значение после инициализации	Выполняемая функция
27:25	MCIF_INTERR	0	27: Длина очереди, выставляемая мостом AXIMCIF2 превышает глубину команд, способную принят контроллером памяти; 1: ошибка, 0: ошибки нет 26: Чтение по шине MCIF произошло без запроса; 1: ошибка, 0: ошибки нет 25: Запись по шине MCIF произошло без запроса; 1: ошибка, 0: ошибки нет
24	MULTI_HIT	0	Обращение при записи по одному и тому же ID адресу 1: событие произошло, 0: событие не произошло
23:0		0	Зарезервировано

Таблица 1.572 – Регистр состояния ошибки при обращении по ID – ERRID

Биты	Название	Значение после инициализации	Выполняемая функция
31:11		0	Зарезервировано
10	MCIF RW Error	1'b0	1: Ошибка обращения по чтению, 0: Ошибка обращения по записи
9:0	Master ID	10'b0	Ошибка передачи транзакций по ID

Таблица 1.573 – Регистр состояния ошибки при обращении по младшим адресам – AEARL

Биты	Название	Значение после инициализации	Выполняемая функция
31:11	AEARL	0	Ошибка передачи по младшим 32-х битам адреса. Адрес – MCIFII O_M_ADR[4:35]

Таблица 1.574 – Регистр состояния ошибки при обращении по старшим адресам – AEARU

Биты	Название	Значение после инициализации	Выполняемая функция
31:4		0	Зарезервировано
3:0	AEARU	0	Ошибка передачи по старшим 4-м битам адреса. А AEARU[3:1] – банки адресов. MCIFII O_M_RS. AEARU[3] – константа 1'b0; AEARU[0] – адресный бит O_M_ADR[3]. Остальные 3 бита O_M_ADR[0:2] всегда константа 3'b0 .

Таблица 1.575 – Регистр для маскирования ошибки, генерации прерываний – AXIERRMSK

Биты	Название	Значение после инициализации	Выполняемая функция
31:24	ERRMSK	0	Маскирование внутренней ошибки 1: Маскирование ошибки без генерации прерываний; 0: Не маскировать ошибку
23:0		0	Зарезервировано

Таблица 1.576 – Регистр для включения асинхронной логики – AXIASYNC

Биты	Название	Значение после инициализации	Выполняемая функция
31:28		0	Зарезервировано
27:25	AXINUM	3’b000	Данное поле используется для генерации сигнала переключения после выставления запроса по AXI. Определяется количество циклов, необходимых после появления запроса по шине MCIF для генерации сигнала переключения: 3’b000: Используется значение по умолчанию. 3’b001: Для AXI: MCIF частота превышает AXI в 4 раза и более, тогда выставляется оптимальное значение программистом.
24:18	MCIFNUM	7’b0000000	Задается количество циклов, необходимое для корректного захвата M_REQ_ACK на стороне AXI. Поскольку соотношение частот шины MCIFII и AXI может быть различным (до 32 раз). Данный регистр используется для хранения данной информации, чтобы корректно осуществлялся захват на стороне AXI. Базовое значение поля задается, исходя из соотношения MCIFII и AXI тактовых сигналов. 7’b0000000 – прописывается базовое значение, исходя из кратности тактовых сигналов шины соотношения AXI и MCIFII, поле DCR_CLK_RATIO, (рекомендовано)
17:12		0	Зарезервировано
11:8	RDRDY_TUNE	4’b0000	Этап проверки готовности данных по чтению перед выставлением сигнала M_RD_DT_DONE_ERLY, необходим для того, чтобы контролировать готовность данных по чтению к следующей транзакции. Данное значение используется, чтобы быть уверенным, что данные на шине MCIFII готовы до того, как AXI мост начнет транзакции по передачи данных на шину данных по чтению. Расшифровка: 4’b0000: Значение по умолчанию (равно 4 MCIFII тактовым сигналам) Остальные: Переписывают данное значение для оптимального решения (равно реальной задержке по линии между сигналами I_M_RD_DT_DONE_ERLY и I_M_RD_DT_DONE)
7:6		0	Зарезервировано
5	DCR_CLK_MN	0	Соотношение частот AXI:MCIFII 0: 1:N 1: 2:N Данный бит используется для указания соотношения AXI частоты к MCIFII. Если данный бит установлен в 1, один дополнительный цикл добавляется для синхронизации пути, чтобы избежать полу-тактовых путей между регистрами.
4:0	DCR_CLK_RATIO	5’b00000	Соотношение частот между MCIFII и AXI. Максимальное соотношение между MCLK и ACLK – это 16. Если отношение не целое, то происходит округление числа. 5’b00000: равной частоты 5’b00001: 1<MCLK/ACLK <=2; 5’b00010: 2< MCLK/ACLK <=3; … 5’b11111: 31< MCLK/ACLK <=32.

Таблица 1.577 – Конфигурационный регистр AXI моста – AXICFG

Биты	Название	Значение после инициализации	Выполняемая функция
31:16		0	Зарезервировано
15	IT_OOO	0	MCIFII сигнал разрешения для задания внутри поточного возврата данных по чтению 0: Отключен, 1: Разрешен Всегда должен быть 1’b0
14:13	TOM	2’b00	Режим задания порядка следования транзакций по MCIFII шине: 00 – в строгом заданном порядке; 10 – разрешено частично, вне очереди; 11 – разрешено вне очереди; Другие – зарезервировано.
12:2		0	Зарезервировано
1	EQ_ID	0	0: разное значение AWID во время чередования данных по записи; 1: одинаковое значение AWID во время чередования данных по записи.
0	EXM_ID	0	0: специальная опция, включает функцию отслеживания данных во время их записи мастером, который впоследствии их считывает; 1: специальная опция, не включает функцию отслеживания данных во время их записи мастером, который впоследствии их считывает;

Таблица 1.578 – Статусный регистр AXI моста – AXISTS

Биты	Название	Значение после инициализации	Выполняемая функция
31		1’b1	Статус транзакции: 1: Пустой 0: Активный Данный бит используется для проверки есть ли запрос, ожидаемый на исполнение. 1: Все транзакции выполнены; 0: Есть запрос.
30:0			Зарезервировано

Таблица 1.579 – Конфигурационный регистр памяти RANK0-3 - MRxCF

Биты	Название	Значение после инициализации	Выполняемая функция
31:20	R_RA	0	Основной адрес ранка памяти, сопоставим с AXI шиной адреса [35:24]
19:16	R_RS	b’0000	0000 – 16 MB; 0001 – 32 MB; 0010 – 64 MB; 0011 – 128 MB; 0100 – 256 MB; 0101 – 512 MB; 0110 – 1 GB; 0111 – 2 GB; 1000 – 4 GB; 1001 – 8 GB; Остальные зарезервированы.
15:1	R_RO	0	Сдвиг AXI адреса относительно RANK адреса
0	R_RV	0	0: отключение RANK 1: включение RANK

Таблица 1.580 – Регистр ID контроля – RID

Биты	Название	Значение после инициализации	Выполняемая функция
31:00	RID	x0000_0101	Значение ID, в константе. Указывается в конфигурационном файле.

1.3.8.4.2 Программно реализуемая модель блока PLB6MCIF2

1.3.8.4.2.1 Список программно доступных регистров блока PLB6MCIF2

В данном разделе описываются программно доступные регистры блока моста с шиной типа PLB6 – PLB6MCIF2. Таблица 1.581 содержит список программно доступных регистров.

Таблица 1.581 – Список регистров блока PLB6

Название	Тип	Смещение относительно базового адреса	Выполняемая функция	Значение после инициализации
BESR	ЗП/ЧТ	x00, x01	PLB регистр состояния ошибки	x0000_0000
BEARL	ЧТ	x02	PLB регистр состояния ошибки при обращении по младшим адресам	x0000_0000
BEARU	ЧТ	x03	PLB регистр состояния ошибки при обращении по старшим адресам	x0000_0000
INTR_EN	ЗП/ЧТ	x04	Регистр прерываний	xFFE0_01C0
	N/A	x05- x06	Зарезервировано	
PLBASYNC	ЗП/ЧТ	x07	PLB регистр для включения асинхронной логики	x0000_0000
PLBORD	ЗП/ЧТ	x08	PLB регистр задания порядка следования транзакций	x0100_0000 или x4000_0000 или x2000_0000
PLBCFG	ЗП/ЧТ	x09	PLB конфигурационный регистр	x2000_00F0
	N/A	x0A- x0E	Зарезервировано	
MAXMEM	ЗП/ЧТ	x0F	PLB регистр для определения максимального адресного пространства	xC000_0000
PUABA	ЗП/ЧТ	x10	PLB регистр для задания актуального адресного пространства	x0000_0000
MR0CF	ЗП/ЧТ	x11	Конфигурационный регистр памяти RANK 0	x0000_0000
MR1CF	ЗП/ЧТ	x12	Конфигурационный регистр памяти RANK 1	x0000_0000
MR2CF	ЗП/ЧТ	x13	Конфигурационный регистр памяти RANK 2	x0000_0000
MR3CF	ЗП/ЧТ	x14	Конфигурационный регистр памяти RANK 3	x0000_0000
MR4CF	ЗП/ЧТ	x19	Конфигурационный регистр памяти RANK 4	x0000_0000
MR5CF	ЗП/ЧТ	x1A	Конфигурационный регистр памяти RANK 5	x0000_0000
MR6CF	ЗП/ЧТ	x1B	Конфигурационный регистр памяти RANK 6	x0000_0000
MR7CF	ЗП/ЧТ	x1C	Конфигурационный регистр памяти RANK 7	x0000_0000
SR0CF	ЧТ	x15	Конфигурационный регистр SUPER RANK 0	Зависит от настроек
SR1CF	ЧТ	x16	Конфигурационный регистр SUPER RANK 1	Зависит от настроек

Продолжение таблицы 1.582

Название	Тип	Смещение относительно базового адреса	Выполняемая функция	Значение после инициализации
SR2CF	ЧТ	x17	Конфигурационный регистр SUPER RANK 2	Зависит от настроек
SR3CF	ЧТ	x18	Конфигурационный регистр SUPER RANK 3	Зависит от настроек
P6BMTAG1	ЗП/ЧТ	x1D	PLB регистр 1 соответствия ID мастеров	x0000_0000
P6BMTAG2	ЗП/ЧТ	x1E	PLB регистр 2 соответствия ID мастеров	x0000_0000
P6BMTAG3	ЗП/ЧТ	x1F	PLB регистр 3 соответствия ID мастеров	x0000_0000
STATUS	ЧТ	x20	Статусный PLB регистр	x8000_0000
RevID	ЧТ	x21	Регистр ID контроля	Вводный знак
MAP0CF	ЗП/ЧТ	x22	Регистр разрешения доступа в адресное пространство памяти RANK 0	x0000_0000
MAP1CF	ЗП/ЧТ	x23	Регистр разрешения доступа в адресное пространство памяти RANK 1	x0000_0000
MAP2CF	ЗП/ЧТ	x24	Регистр разрешения доступа в адресное пространство памяти RANK 2	x0000_0000
MAP3CF	ЗП/ЧТ	x25	Регистр разрешения доступа в адресное пространство памяти RANK 3	x0000_0000
MAP4CF	ЗП/ЧТ	x26	Регистр разрешения доступа в адресное пространство памяти RANK 4	x0000_0000
MAP5CF	ЗП/ЧТ	x27	Регистр разрешения доступа в адресное пространство памяти RANK 5	x0000_0000
MAP6CF	ЗП/ЧТ	x28	Регистр разрешения доступа в адресное пространство памяти RANK 6	x0000_0000
MAP7CF	ЗП/ЧТ	x29	Регистр разрешения доступа в адресное пространство памяти RANK 7	x0000_0000
		x2A-x3F	Зарезервировано	

1.3.8.4.2.2 Описание программно доступных регистров блока PLB6MCIF2

Таблица 1.583 – PLB регистр состояния ошибки - BESR

Биты	Название	Значение после инициализации	Выполняемая функция
0	MxRW	0	Ошибка в операции: 0 – Запись 1 – Чтение
1:5	MxID	b’00000	ID PLB мастера, во время работы с которым обнаружилась ошибка
6:10	MTAG	b’00000	PLB MTag, во время работы с которым обнаружилась ошибка
11:14	BufferID	b’0000	Буфер ID, во время работы с которым обнаружилась ошибка
15	RS_Err[0]	0	Для 8-ми сегментной конфигурации памяти {BESR[15], BESR[21:22]} записывается сегмент eDRAM памяти, во время работы с которым обнаружилась ошибка Для 4-рех сегментной конфигурации памяти eDRAM или для ранка памяти DDR3, данный бит зарезервирован.
16:20	MxET1[1:5]	b’00000	Идентификация наличия ошибки типа 1 MxET1[0:5] – определяет ошибку типа 1 000000 – нет ошибки; 100000 – ошибка в доступе мастера по PLB6; 010000 – ошибка декодирования адреса 001000 – ошибка четности по адресной шине или сигналам управления (Address-or-byte-enable-parity-error) 000100 – ошибка записи данных по шине PLB; 000010 – ошибка чтения данных по шине MCIFII; 000001 – ошибка четности чтения данных по шине MCIFII;
21:22	RS_Err[1:2]	b’00	Для 8-ми сегментной конфигурации памяти {BESR[15], BESR[21:22]} записывается сегмент eDRAM памяти, во время работы с которым обнаружилась ошибка Для 4-рех сегментной конфигурации памяти eDRAM или для ранка памяти DDR3, BESR[21:22] записывается сегмент eDRAM памяти или номер ранка памяти DDR3, во время работы с которым обнаружилась ошибка
23:25	MxET2	b’000	Идентификация наличия ошибки типа 2 00 – нет ошибки; 1xx – PLB ошибка четности по записи; 01x – MCIFII ошибка четности по записи; 001 – MCIFII ошибка по записи, RMW ошибка (read-modify-write)
26	MxET1[0]	b’0	Самый значимы бит ошибки типа 1, см описание MxET1[1:5]
27	RESV	b’0	Зарезервирован
28:31	MxET3	b’000	Идентификация наличия ошибки типа 3 0000 – нет флага; 1000 – флаг декодирования; 0100 – MCIFII флаг; 0010 – флаг записи; 0001 – флаг чтения.

Таблица 1.584 – PLB регистр состояния ошибки при обращении по младшим адресам - BEARL

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	BEARL	0	32 бита младших адресов шины PLB, по которым при обращении соответствующим мастером обнаружена ошибка в выполняемой транзакции

Таблица 1.585 – PLB регистр состояния ошибки при обращении по старшим адресам - BEARU

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	BEARU	0	32 бита старших адресов шины PLB, по которым при обращении соответствующим мастером обнаружена ошибка в выполняемой транзакции

Таблица 1.586 – Регистр прерываний – INTR_EN

Биты	Название	Значение после инициализации	Выполняемая функция
0:15	ERR_EN	xFFE0	Вывод ошибки Соответствует полю BESR[16:31] 0 – выводить ошибки; 1 – не выводить ошибки
16:31	INTR_EN	x01C0	Ошибка прерываний Соответствует полю BESR[16:31] 0 – выводить прерывание; 1 – не выводить прерывание Учтите, что INTR_EN[5:6,11] зарезервированы и должны быть всегда установленв ‘b000

Таблица 1.587 – Регистр включения асинхронной логики – PLBASYNC

Биты	Название	Значение после инициализации	Выполняемая функция
0	PLBgtMCIF		Частота больше PLB, чем MCIF 0 – частота PLB6 <= 0.95x(частота MCIF) 1 – частота PLB6 >= 1.05x(частота MCIF)
1	PLBeqMCIF		Одинаковые частоты PLB и MCIF 0 – частота PLB6 не равна частоте MCIF 1– частота PLB6 равна частоте MCIF, ±5%
2:3		0	Зарезервировано
4:7	PLB6NUM		Количество PLB6 тактовых сигналов, необходимых для осуществления перехода с PLB6 в MCIFII тактовый домен. Для достижения оптимального значения, используется формула: $PLB6NUM = \max(\text{ceil}[(3.5 + SYNC\_3\_STAGE) \times Tmc$ SYNC_3_STAGE = ‘0’, если ‘PLB6MCIF2_3STAGE_SYNC’ не установлен или ‘1’, если ‘PLB6MCIF2_3STAGE_SYNC’ установлен; Tmcif = частота MCIF; Tplb = частота PLB6.
8:11	MCIFNUM		Количество MCIFII тактовых сигналов, необходимых для осуществления перехода с MCIFII в PLB6 тактовый домен. Для достижения оптимального значения, используется формула: $MCIFNUM = \max(\text{ceil}[(3.5 + SYNC\_3\_STAGE) \times Tplb$ SYNC_3_STAGE = ‘0’, если ‘PLB6MCIF2_3STAGE_SYNC’ не установлен или ‘1’, если ‘PLB6MCIF2_3STAGE_SYNC’ установлен; Tmcif= частота MCIF; Tplb = частота PLB6.
12:19		0	Зарезервировано
20:23	RDRDY_TUNE		RDRDY – количество тактов перезаписи. Для синхронных переходов (‘PLB6MCIF2_3STAGE_SYNC’ не установлен), поле должно быть равно ‘b0010. Для асинхронных переходов (‘PLB6MCIF2_3STAGE_SYNC’ установлен), это поле должно быть рано 0.
24:31		0	Зарезервировано

Таблица 1.588 – PLB регистр задания порядка следования транзакций – PLBORD

Биты	Название	Значение после инициализации	Выполняемая функция
0	RaW	0	PLB Read-Around_Write: Осуществление транзакций чтения в обход транзакций записи, при учете, что обращение идет по разным адресам. 0 – отключение режима; 1 – включение режима.
1:3	Threshold[1:3]	См п.1	Младшие (LSb) биты порогового значения для режима RaW
4:6		0	Зарезервировано
7	Threshold[0]	См п.1	Старшие (MSb) биты порогового значения для режима RaW
8:15		0	Зарезервировано
16	IT_OOO	0	MCIFII внутри поточный возврат данных на чтение вне очереди (Intrathread Out-Of_Order Read Data Return). 0 – отключение режима; 1 – включение режима.
17:18	TOM	0	Режим задания порядка следования транзакций по MCIFII шине: 00 – в строгом заданном порядке; 10 – разрешено частично, вне очереди; 11 – разрешено вне очереди; Другие – зарезервировано.
19:31		0	Зарезервировано

Пункт 1: Сигнал сброса регистра Threshold[0:3] задает глубину конфигурируемой очереди следующим образом:

- 1000 – глубина очереди 32 (16 транзакций на чтение и 16 транзакций на запись);
- 0100 – глубина очереди 16 (8 транзакций на чтение и 8 транзакций на запись);
- 0011 – глубина очереди 8 (4 транзакций на чтение и 4 транзакций на запись);

Таблица 1.589 – PLB конфигурационный регистр – PLBCFG

Биты	Название	Значение после инициализации	Выполняемая функция
0	Noncoherent	0	Когерентность: 0 – когерентна; 1 – не когерентна.
1:4	Tsnoop	100	0100 – Tsnoop=4 0101 – Tsnoop=5 0110 – Tsnoop=6 0111 – Tsnoop=7 1000 – Tsnoop=8 Остальные недоступны
5	PmstrID_chk_EN	0	Доступ всех мастеров к шине PLB6. Если 0, функция защиты памяти отключена и все мастера имеют доступ к шине PLB6. В противном случае, мастерам шины PLB6, определенным в регистрах P6BMTAGx и MAPxCF, разрешено писать и читать через ядро PLB6MCIF2
6:23		0	Зарезервировано
24:27	Parity Enables	1111	Параметры ошибок четности MCIF: [24] adr_rs_par_en [25] be_par_en [26] cmd_par_en [27] mcif_par_en

Продолжение таблицы 1.590

Биты	Название	Значение после инициализации	Выполняемая функция
28:30	CG_ENABLE	000	Управление тактовыми сигналами (т.с). Разрешено управлять тактовыми сигналами во время базовых операций для каждого тактового домена. PLBCFG[28] – управление т.с домена 1; PLBCFG[29] – управление т.с домена 2; PLBCFG[30] – управление т.с домена 3; 0 – управление т.с для соответствующего домена отключено; 1 – управление т.с для соответствующего домена включено;
31	Br_Enable	0	Активация работы моста в нормальном режиме: 0 – мост отключен; 1 – мост включен в нормальном режиме. Перед тем, как установить 1, необходимо программным образом настроить все соответствующие DCR регистры в нужное значение.

Таблица 1.591 – PLB регистр максимального адресного пространства – MAXMEM

Биты	Название	Значение после инициализации	Выполняемая функция
0:1	MAXMEM	2'b11	Задается максимальное адресное пространство шины PLB: 2'b11 – 64Гб; 2'b10 – 32Гб; 2'b01 – 16Гб; 2'b00 – 8Гб;
2:31		0	Зарезервировано

Таблица 1.592 – PLB регистр актуального адресного пространства – PUABA

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	PUABA	0	Значение данного регистра определяется в соответствии со значениями регистров MAXMEM и MRxCF для задания реального PLB адресного пространства в соответствии с PLB6MCIF2. Данный адрес соотносится с 31-битной шиной адреса PLB: Для работы с DDR3 MC, PUABA[0:30] определяет 64/32/8Гб область, в которой PLB6 мост активен. PUABA[31] зарезервирован. Для работы с EDRAM MC, PUABA[0:31] определяет 4Гб область, в которой PLB6 мост активен.

Для примера:
Если MAXMEM=2'b11, PUABA[0:27] соотноситься с I_S_ADDR[0:27];
Если MAXMEM=2'b10, PUABA[0:28] соотноситься с I_S_ADDR[0:28];
Если MAXMEM=2'b01, PUABA[0:29] соотноситься с I_S_ADDR[0:29];
Если MAXMEM=2'b00, PUABA[0:30] соотноситься с I_S_ADDR[0:30];

Таблица 1.593 – Конфигурационные регистр памяти RANK – MRxCF

Биты	Название	Значение после инициализации	Выполняемая функция
0:12	M_BA[0:12]	0	Для работы с DDR3 MC, M_BA[0:12] задают исходное значение смещения в максимально возможном адресном пространстве, определенным в MAXMEM, и заданным реальным адресом в PUABA. Соответствуют [28:40] битам PLB шины адреса. Для работы с EDRAM MC, M_BA[0:9] задают исходное значение смещения в 4Гб адресном пространстве, определенным в MAXMEM, и заданным реальным адресом в PUABA. Соответствуют [32:41] битам PLB шины адреса. M_BA[10:12] зарезервированы

Продолжение таблицы 1.594

Биты	Название	Значение после инициализации	Выполняемая функция
13:15		0	Зарезервировано
16:19	M_SZ	0000	Для работы с DDR3 MC: 0000 – 8 Мб; 0001 – 16 Мб; 0010 – 32 Мб; 0011 – 64 Мб; 0100 – 128 Мб; 0101 – 256 Мб; 0110 – 512 Мб; 0111 – 1 Гб; 1000 – 2 Гб; 1001 – 4 Гб; 1010 – 8 Гб; 1011 – 16 Гб; 1100 – 32 Гб; Остальные зарезервированы; Для работы с EDRAM MC: 0100 – 4 MB Остальные зарезервированы.
20:30		0	Зарезервировано
31	M_RE	0	Если 1, обеспечивается доступ системы к ранку памяти

Таблица 1.595 – Конфигурационные регистр памяти SUPER RANK – SRxCF

Биты	Название	Значение после инициализации	Выполняемая функция
0:12	S_BA[0:12]	I_SUPER_BASE{0-3}[0:12]	Соответствуют [28:40] битам перераспределенных адресов.
13:15		0	Зарезервировано
16:19	S_SZ	I_SUPER_SIZE{0-3}[0:3]	0000 – 8 Мб; 0001 – 16 Мб; 0010 – 32 Мб; 0011 – 64 Мб; 0100 – 128 Мб; 0101 – 256 Мб; 0110 – 512 Мб; 0111 – 1 Гб; 1000 – 2 Гб; 1001 – 4 Гб; 1010 – 8 Гб; 1011 – 16 Гб; 1100 – 32 Гб; Остальные зарезервированы.
20:30		0	Зарезервировано
31	S_RE	I_SUPER_RE{0-3}	Если 1, обеспечивается доступ системы к памяти SUPER RANK

Данный регистр обеспечивает соответствие ID мастеров шины PLB6 (I_S_MAST_ID[0:4]) со специальными ID n мастерами (n от 0 до 5).

Когда ID мастеру шины PLB6 разрешен доступ к памяти, то происходит сравнение полей MstIDWrAcc и MstIDRdAcc регистра MAPxCF. На основе полученного сравнения делается заключение о возможности писать и читать в выбранную область памяти EDRAM.

Данный регистр поддерживается только для работы с контроллером EDRAM.

Таблица 1.596 – PLB регистр 1 соответствия ID мастеров – P6BMTAG1

Биты	Название	Значение после инициализации	Выполняемая функция
0:4	Mst0ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 0
5:9	Mst1ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 1
10:14	Mst2ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 2
15:19	Mst3ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 3
20:24	Mst4ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 4
25:29	Mst5ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 5
30:31		0	Зарезервировано

Данный регистр обеспечивает соответствие ID мастеров шины PLB6 (I_S_MAST_ID[0:4]) со специальными ID n мастерами (n от 6 до 11).

Когда ID мастеру шины PLB6 разрешен доступ к памяти, то происходит сравнение полей MstIDWrAcc и MstIDRdAcc регистра MAPxCF. На основе полученного сравнения делается заключение о возможности писать и читать в выбранную область памяти EDRAM.

Данный регистр поддерживается только для работы с контроллером EDRAM.

Таблица 1.597 – PLB регистр 2 соответствия ID мастеров – P6BMTAG2

Биты	Название	Значение после инициализации	Выполняемая функция
0:4	Mst6ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 6
5:9	Mst7ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 7
10:14	Mst8ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 8
15:19	Mst9ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 9
20:24	Mst10ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 10
25:29	Mst11ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 11
30:31		0	Зарезервировано

Данный регистр обеспечивает соответствие ID мастеров шины PLB6 (I_S_MAST_ID[0:4]) со специальными ID n мастерами (n от 6 до 11).

Когда ID мастеру шины PLB6 разрешен доступ к памяти, то происходит сравнение полей MstIDWrAcc и MstIDRdAcc регистра MAPxCF. На основе полученного сравнения делается заключение о возможности писать и читать в выбранную область памяти EDRAM.

Данный регистр поддерживается только для работы с контроллером EDRAM.

Таблица 1.598 – PLB регистр 3 соответствия ID мастеров – P6BMTAG3

Биты	Название	Значение после инициализации	Выполняемая функция
0:4	Mst12ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 12
5:9	Mst13ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 13
10:14	Mst14ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 14
15:19	Mst15ID	0	Прописывается ID мастера шины PLB6 (I_S_MAST_ID[0:4]), которое соотносится с мастером 15
20:31		0	Зарезервировано

Регистр должен соотноситься с установками, прописанными в регистре MRxCF. Данный регистр поддерживается только для работы с контроллером EDRAM.

Таблица 1.599 – Регистр разрешения доступа в адресное пространство памяти RANK0 – MAPxCF

Биты	Название	Значение после инициализации	Выполняемая функция
0:15	MstIDWrAcc	0	ID мастера для разрешения записи
16:31	MstIDRdAcc	0	ID мастера для разрешения чтения

Таблица 1.600 – Статусный регистр – STATUS

Биты	Название	Значение после инициализации	Выполняемая функция
0	IDLE	1	0 – Занят; 1 – Пустой. Означает, что нет текущих запросов, находящихся в статусе ожидания
1:31		0	Зарезервировано

Таблица 1.601 – Регистр контроля ID – RID

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	RID	Вводный знак	Значение ID

1.3.8.4.2.3 Последовательность сигналов сброса блока PLB6MCIF2

Существует два сигнала сброса для блока PLB6MCIF2: I_RESET и I_RESET_RECOVER.

Не допускается одновременная установка данных сигналов в ’b1, в противном случае результаты не предсказуемы.

Для переведения блока PLB6MCIF2 в начальное состояние, необходимо подать ‘1’ на ‘I_RESET’ и удерживать не менее 40 тактовых сигналов PLB частоты, при условии, что ‘I_RESET_RECOVER’ будет находиться в ‘0’. Данный сигнал сброса обнулит полностью состояние конфигурационных регистров и регистров, хранящие значения шин данных, в ядре данного блока для выполнения полной последовательности инициализации ядра.

Для частичного сброса состояния блока PLB6MCIF2, необходимо подать ‘1’ на ‘I_RESET_RECOVER’ и удерживать не менее 40 тактовых сигналов PLB частоты, при условии, что ‘I_RESET’ будет находиться в ‘0’. Данный сигнал сброса обнулит состояние регистров, хранящие значения шин данных, в ядре данного блока и оставит нетронутыми значения конфигурационных регистров.

1.3.8.4.3 Программно реализуемая модель блока MCIF2ARB

1.3.8.4.3.1 Описание программно доступного интерфейса CREG блока MCIF2ARB

Регистровый интерфейс блока арбитра MCIF2 является полностью асинхронным, с помощью которого можно программно доступными средствами сконфигурировать систему. Протокол состоит из стандартной последовательности - ‘запрос/подтверждение’. Сигнал запроса активируется после установки адреса, команд (на чтение или запись) и данных (для записи). После обнаружения запроса выполняется текущая транзакция логикой CREG, с последующим выставлением подтверждения о выполнении данной транзакции. Последовательно снимается запрос и подтверждение.

1.3.8.4.3.2 Список программно доступных регистров блока MCIF2ARB

Таблица 1.602 – Список регистров блока MCIF2ARB

Название	Тип	Смещение относительно базового адреса	Выполняемая функция	Значение после инициализации
MACR	ЗП/ЧТ	x00	Конфигурационный MCIF регистр арбитража	x0000_0000

1.3.8.4.3.3 Описание программно доступных регистров блока MCIF2ARB

Таблица 1.603 – Конфигурационный MCIF регистр арбитража – MACR

Биты	Название	Значение после инициализации	Выполняемая функция
31:28	MPM	0	Определяет очередность арбитража 4’b0000: Свободный; 4’b0001: Заданный. Остальные зарезервированы. 0 – Приоритетность отдается более приоритетному мастеру с запросом, ожидающим выполнение, таблица приоритетностей; 1 – Приоритетность отдается по заранее выбранному механизму задания приоритетности (master0 -> master 3).
27	WD_DLY	0	Задержка между запросом и данными на запись: 0 – 1 такт; 1 – 2 такта.
26:10		0	Зарезервировано
9:8	ORDER_ATT	0	Определяется значение выходного сигнала O_M_ORDER_ATT[0:1]: 00 – в заданном порядке; 01 – зарезервировано; 10 – частично вне очереди; 11 – вне очереди.
7:0		0	Зарезервировано

1.3.8.4.4 Программно реализуемая модель блока MCLFIR

1.3.8.4.4.1 Список программно доступных регистров блока MCLFIR

В данном разделе описываются программно доступные регистры блока контроля и статуса состояний – MCLFIR. Таблица 1.604 содержит список программно доступных регистров.

Таблица 1.604 – Список регистров блока MCLFIR

Название	Тип	Сдвиг адреса	Функция	Значение после сброса
MC_ERR0	ЧТ	0x0	Регистры для вывода ошибок контроллера MC DDR3	0x0000_0000
MC_ERR1	ЧТ	0x1		0x0000_0000
MC_ERR_MSK0	ЧТ	0x6	Регистры для задания маскируемых и немаскируемых ошибок	0xFFFF_FFFF
MC_ERR_MSK1	ЧТ	0x7		0xFFFF_FFC0
MC_ERR_ACTION0	ЗП/ЧТ	0x0C	Регистры для задания признака перехода немаскируемой ошибки с первого уровня на второй уровень обработки ошибок	0x0000_0000
MC_ERR_ACTION1	ЗП/ЧТ	0x0D		0x0000_0000
MC_ERR_WOF0	ЗП/ЧТ	0x0E	Регистры для протоколирования всех немаскированных ошибок после снятия сигнала сброса	0x0000_0000
MC_ERR_WOF1	ЗП/ЧТ	0x0F		0x0000_0000
MC_ERR_AND0	ЗП	0x02	Регистры для обнуления немаскируемых ошибок	N/A
MC_ERR_AND1	ЗП	0x03		N/A
MC_ERR_OR0	ЗП	0x04	Регистры для перезаписи немаскируемых ошибок	N/A
MC_ERR_OR1	ЗП	0x05		N/A
MC_ERR_MSK_AND0	ЗП	0x08	Регистры для обнуления признака маскируемых и немаскируемых ошибок	N/A
MC_ERR_MSK_AND1	ЗП	0x09		N/A
MC_ERR_MSK_OR0	ЗП	0x0A	Регистры для перезаписи признака маскируемых и немаскируемых ошибок	N/A
MC_ERR_MSK_OR1	ЗП	0x0B		N/A
MC_LFIR	ЧТ	0x10	Регистр для задания ошибок, которые можно устранить	0x0000_0000
MC_LFIR_AND	ЗП	0x11	Регистр для обнуления ошибок, которые можно устранить	N/A
MC_LFIR_OR	ЗП	0x12	Регистр для перезаписи ошибок, которые можно устранить	N/A
MC_LFIR_MSK	ЧТ	0x13	Регистр для задания признака перехода немаскируемой ошибки со второго уровня обработки ошибок на верхний уровень системы	0xC000_0000
MC_LFIR_MSK_AND	ЗП	0x14	Регистр для обнуления признака перехода немаскируемой ошибки со второго уровня обработки ошибок на верхний уровень системы	N/A
MC_LFIR_MSK_OR	ЗП	0x15	Регистр для перезаписи признака перехода немаскируемой ошибки со второго уровня обработки ошибок на верхний уровень системы	N/A
MC_CONSOL_STAT	ЧТ	0x20	Регистр статусного состояния контроллера MC DDR3	0x0000_0000
MC_CONSOL_STAT_AND	ЗП	0x21	Регистр для обнуления статусного состояния контроллера MC DDR3	N/A

Продолжение таблицы 1.605

Название	Тип	Сдвиг адреса	Функция	Значение после сброса
MC_CONSOL_STAT_OR	ЗП	0x22	Регистр для перезаписи статусного состояния контроллера MC DDR3	N/A
MC_CONSOL_STAT_TRIG	ЧТ	0x23	Регистр для задания признака события	0x0000_0000
MC_CONSOL_STAT_TRIG_AND	ЗП	0x24	Регистр для обнуления признака события	N/A
MC_CONSOL_STAT_TRIG_OR	ЗП	0x25	Регистр для перезаписи признака события	N/A
MC_CONSOL_STAT_MSK	ЧТ	0x26	Регистр для задания признака перехода немаскируемого события с первого уровня на второй уровень обработки ошибок	0xFFF0_8000
MC_CONSOL_STAT_MSK_AND	ЗП	0x27	Регистр для обнуления признака перехода немаскируемого события с первого уровня на второй уровень обработки ошибок	N/A
MC_CONSOL_STAT_MSK_OR	ЗП	0x28	Регистр для перезаписи признака перехода немаскируемого события с первого уровня на второй уровень обработки ошибок	N/A
MC_CONSOL_STATSUM	ЧТ	0x30	Регистр для вывода информации о текущем событии	0x0000_0000
MC_CONSOL_STATSUM_AND	ЗП	0x31	Регистр для сброса информации о текущем событии	N/A
MC_CONSOL_STATSUM_OR	ЗП	0x32	Регистр для перезаписи информации о текущем событии	N/A
MC_CONSOL_STATSUM_MSK	ЧТ	0x33	Регистр для задания признака перехода немаскируемой информации о текущем событии со второго уровня обработки ошибок на верхний уровень системы	0x8000_0000
MC_CONSOL_STATSUM_MSK_AND	ЗП	0x34	Регистр для сброса перехода немаскируемой информации о текущем событии со второго уровня обработки ошибок на верхний уровень системы	N/A
MC_CONSOL_STATSUM_MSK_OR	ЗП	0x35	Регистр для перезаписи признака перехода немаскируемой информации о текущем событии со второго уровня обработки ошибок на верхний уровень системы	N/A

1.3.8.4.4.2 Описание программно доступных регистров об ошибках контроллера MC DDR3 первого уровня

Таблица 1.606 – Регистр для вывода ошибок MC DDR3 – MC_ERR0

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	I_ERROR[0:31]	0	Выводы MC DDR3 ‘I_ERROR[0:31]‘ подключены к регистру

Таблица 1.607 – Регистр для вывода ошибок MC DDR3 – MC_ERR1

Биты	Название	Значение после инициализации	Выполняемая функция
0:25	I_ERROR[32:57]	0	Выводы MC DDR3 ‘I_ERROR[0:31]‘ подключены к регистру
26:31		0	Зарезервировано

Таблица 1.608 – Регистр для задания маскируемых и немаскируемых ошибок MC DDR3 – MC_ERR_MSK0

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	MASK	для каждого m бита: ‘b1	Регистр зеркально отображает информацию по регистру MC_ERR0. Для каждого m бита: ‘1’b – маскируемая ошибка, соответствующий бит регистра MC_ERR0 запрещено передавать на второй уровень обработки ошибки; ‘0’b – немаскируемая ошибка, соответствующий бит регистра MC_ERR0, равный 1’b1, разрешено передать на второй уровень обработки ошибки с учетом настроек регистра MC Action0, заданных программным образом.

Таблица 1.609 – Регистр для задания маскируемых и немаскируемых ошибок MC DDR3 – MC_ERR_MSK1

Биты	Название	Значение после инициализации	Выполняемая функция
0:25	MASK	‘для каждого m бита: ‘b1	Регистр зеркально отображает информацию по регистру MC_ERR1. Для каждого m бита: ‘1’b – маскируемая ошибка, соответствующий бит регистра MC_ERR1 запрещено передавать на второй уровень обработки ошибки; ‘0’b – немаскируемая ошибка, соответствующий бит регистра MC_ERR1, равный 1’b1, разрешено передать на второй уровень обработки ошибки с учетом настроек регистра MC Action1, заданных программным образом.
26:31		0	Зарезервировано

Таблица 1.610 – Регистр для задания признака перехода немаскируемой ошибки с первого уровня на второй уровень обработки ошибок – MC_ERR_ACTION0

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	ACTION	N/A	Регистр зеркально отображает информацию по регистру MC_ERR0. Для каждого m бита: ‘1’b – признак неустранимой ошибки: когда соответствующий бит регистра MC_ERR0 равен 1’b1 и соответствующий бит регистра MC_ERR_MSK0 равен 1’b0, возможна установка ‘UNRECOVERABLE_ERR’ бита в ‘MC_LFIR’ регистре; ‘0’b – признак устранимой ошибки: когда соответствующий бит регистра MC_ERR0 равен 1’b1 и соответствующий бит регистра MC_ERR_MSK0 равен 1’b0, возможна установка ‘RECOVERABLE_ERR’ бита в ‘MC_LFIR’ регистре.

Таблица 1.611 – Регистр для задания признака о переходе немаскируемой ошибки с первого уровня на второй уровень обработки ошибок – MC_ERR_ ACTION1

Биты	Название	Значение после инициализации	Выполняемая функция
0:25	ACTION	N/A	Регистр зеркально отображает информацию по регистру MC_ERR1. Для каждого m бита: ‘1’b – признак неустранимой ошибки: когда соответствующий бит регистра MC_ERR1 равен 1’b1 и соответствующий бит регистра MC_ERR_MSK1 равен 1’b0, возможна установка ‘UNRECOVERABLE_ERR’ бита в ‘MC_LFIR’ регистре; ‘0’b – признак устранимой ошибки: когда соответствующий бит регистра MC_ERR1 равен 1’b1 и соответствующий бит регистра MC_ERR_MSK1 равен 1’b0, возможна установка ‘RECOVERABLE_ERR’ бита в ‘MC_LFIR’ регистре.
26:31		0	Зарезервировано

Таблица 1.612 – Регистр для протоколирования всех немаскированных ошибок после снятия сигнала сброса – MC_ERR_WOF0

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	WOF	0	Регистр зеркально отображает информацию по ошибкам регистра MC_ERR0 с момента снятия сигнала сброса до появления первой немаскируемой ошибки. Значения регистра хранятся до момента, когда не произойдет обнуление (регистр MC_ERR_WOF_AND0) или перезапись (регистр MC_ERR_WOF_OR0) значений программным образом. Обнуление регистра MC_ERR0 приводит к автоматическому сбросу данного регистра. Для каждого m бита: ‘1’b – немаскируемый бит регистра MC_ERR0 равен 1’b1 сразу после снятия сигнала, первая немаскируемая ошибка (Who’s On First); ‘0’b – немаскируемый бит регистра MC_ERR0 не установлен сразу после снятия сигнала.

Таблица 1.613 – Регистр для протоколирования всех немаскированных ошибок после снятия сигнала сброса – MC_ERR_WOF1

Биты	Название	Значение после инициализации	Выполняемая функция
0:25	WOF	0	Регистр зеркально отображает информацию по ошибкам регистра MC_ERR1 с момента снятия сигнала сброса до появления первой немаскируемой ошибки. Значения регистра хранятся до момента, когда не произойдет обнуление (регистр MC_ERR_WOF_AND1) или перезапись (регистр MC_ERR_WOF_OR1) значений программным образом. Обнуление регистра MC_ERR1 приводит к автоматическому сбросу данного регистра. Для каждого m бита: ‘1’b – немаскируемый бит регистра MC_ERR1 равен 1’b1 сразу после снятия сигнала, первая немаскируемая ошибка (Who’s On First); ‘0’b – немаскируемый бит регистра MC_ERR1 не установлен сразу после снятия сигнала.
26:31		0	Зарезервировано

Таблица 1.614 – Регистр для обнуления немаскированных ошибок – MC_ERR_AND0

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_ERR0. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_ERR0 будет обнулен; ‘1’b – соответствующий бит регистра MC_ERR0 не будет изменен.

Таблица 1.615 – Регистр для обнуления немаскированных ошибок – MC_ERR_AND1

Биты	Название	Значение после инициализации	Выполняемая функция
0:25	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_ERR1. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_ERR1 будет обнулен; ‘1’b – соответствующий бит регистра MC_ERR1 не будет изменен.
26:31		N/A	Зарезервировано

Таблица 1.616 – Регистр для перезаписи немаскированных ошибок – MC_ERR_OR0

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_ERR0. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_ERR0 будет установлен; ‘0’b – соответствующий бит регистра MC_ERR0 не будет изменен.

Таблица 1.617 – Регистр для перезаписи немаскированных ошибок – MC_ERR_OR1

Биты	Название	Значение после инициализации	Выполняемая функция
0:25	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_ERR1. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_ERR1 будет установлен; ‘0’b – соответствующий бит регистра MC_ERR1 не будет изменен.
26:31		N/A	Зарезервировано

Таблица 1.618 – Регистр для обнуления признака маскируемых или немаскированных ошибок – MC_ERR_MSK_AND0

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_ERR_MSK0. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_ERR_MSK0 будет обнулен; ‘1’b – соответствующий бит регистра MC_ERR_MSK0 не будет изменен.

Таблица 1.619 – Регистр для обнуления признака маскируемых или немаскированных ошибок – MC_ERR_MSK_AND1

Биты	Название	Значение после инициализации	Выполняемая функция
0:25	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_ERR_MSK1. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_ERR_MSK1 будет обнулен; ‘1’b – соответствующий бит регистра MC_ERR_MSK1 не будет изменен.
26:31		N/A	Зарезервировано

Таблица 1.620 – Регистр для перезаписи признака маскируемых или немаскированных ошибок – MC_ERR_MSK_OR0

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_ERR_MSK0. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_ERR_MSK0 будет установлен; ‘0’b – соответствующий бит регистра MC_ERR_MSK0 не будет изменен.

Таблица 1.621 – Регистр для перезаписи признака маскируемых или немаскированных ошибок – MC_ERR_MSK_OR1

Биты	Название	Значение после инициализации	Выполняемая функция
0:25	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_ERR_MSK1. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_ERR_MSK1 будет установлен; ‘0’b – соответствующий бит регистра MC_ERR_MSK1 не будет изменен.
26:31		N/A	Зарезервировано

1.3.8.4.4.3 Описание программно доступных регистров об ошибках контроллера MC DDR3 второго уровня

Таблица 1.622 – Регистр для ошибок, которые можно устранить – MC_LFIR

Биты	Название	Значение после инициализации	Выполняемая функция
0	RECIVERABLE_ERR	0	Устраняемая ошибка. Признак ошибки устанавливается на первом уровне обработки ошибки
1	UNRECIVERABLE_ERR	0	Неустранимая ошибка. Признак ошибки устанавливается на первом уровне обработки ошибки.
26:31		0	Зарезервировано

Таблица 1.623 – Регистр для обнуления признака ошибок, которые можно устранить – MC_LFIR_AND

Биты	Название	Значение после инициализации	Выполняемая функция
0:1	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_LFIR. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_LFIR будет обнулен; ‘1’b – соответствующий бит регистра MC_LFIR не будет изменен.
2:31		N/A	Зарезервировано

Таблица 1.624 – Регистр для перезаписи признака ошибок, которые можно устранить – MC_LFIR_OR

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_LFIR. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_LFIR будет установлен; ‘0’b – соответствующий бит регистра MC_LFIR не будет изменен.

Таблица 1.625 – Регистр для задания признака перехода немаскируемой ошибки со второго уровня обработки ошибок на верхний уровень системы – MC_LFIR_MSK

Биты	Название	Значение после инициализации	Выполняемая функция
0	RECOV_MASK	‘1’b	Регистр зеркально отображает информацию по регистру MC_LFIR. ‘1’b – маскируемая ошибка, соответствующий бит регистра MC_ LFIR запрещено передавать на верхний уровень системы (O_MC_RECOV_ERR = ‘0’b); ‘0’b – немаскируемая ошибка, соответствующий бит регистра MC_LFIR разрешено передать на верхний уровень системы, как сигнал прерывания (O_MC_RECOV_ERR присваивает значение бита RECOVERABLE_ERR в регистре MC_LFIR).
1	UNRECOV_MASK	‘1’b	Регистр зеркально отображает информацию по регистру MC_LFIR. ‘1’b – немаскируемая ошибка, соответствующий бит регистра MC_ LFIR запрещено передавать на верхний уровень системы (O_MC_UNRECOV_ERR = ‘0’b); ‘0’b – немаскируемая ошибка, соответствующий бит регистра MC_LFIR разрешено передать на верхний уровень системы, как сигнал прерывания (O_MC_UNRECOV_ERR присваивает значение бита UNRECOVERABLE_ERR в регистре MC_LFIR).
2:31		0	Зарезервировано

Таблица 1.626 – Регистр для обнуления признака перехода немаскируемой ошибки со второго уровня обработки ошибок на верхний уровень системы – MC_LFIR_MSK _AND

Биты	Название	Значение после инициализации	Выполняемая функция
0:1	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_LFIR_MSK. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_LFIR_MSK будет обнулен; ‘1’b – соответствующий бит регистра MC_LFIR_MSK не будет изменен.
2:31		N/A	Зарезервировано

Таблица 1.627 – Регистр для перезаписи перехода немаскируемой ошибки со второго уровня обработки ошибок на верхний уровень системы – MC_LFIR_MSK _OR

Биты	Название	Значение после инициализации	Выполняемая функция
0:31	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_LFIR_MSK. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_LFIR_MSK будет установлен; ‘0’b – соответствующий бит регистра MC_LFIR_MSK не будет изменен.

1.3.8.4.4.4 Описание программно доступных регистров о статусе контроллера MC DDR3 первого уровня

Таблица 1.628 – Регистр статусного состояния контроллера MC DDR3 – MC_CONSOL_STAT

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	I_STATUS[0:11]	0	Биты регистра соответствуют битам выходной шины статусных данных, I_STATUS[0:MCSTAT_WIDTH-2].
12:15		0	Зарезервировано
16	I_STATUS[12]	0	Бит регистра соответствует биту выходной шины статусных данных, I_STATUS[0:MCSTAT_WIDTH-1].
17:31		0	Зарезервировано

*Значение по умолчанию шины статусных данных MCSTAT_WIDTH = 13

Таблица 1.629 – Регистр для обнуления статусного состояния контроллера MC DDR3 – MC_CONSOL_STAT _AND

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STAT. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_ CONSOL_STAT будет обнулен; ‘1’b – соответствующий бит регистра MC_CONSOL_STAT не будет изменен.
12:15		N/A	Зарезервировано
16	RESET_BIT	N/A	‘0’b – соответствующий бит регистра MC_ CONSOL_STAT будет обнулен; ‘1’b – соответствующий бит регистра MC_CONSOL_STAT не будет изменен.
17:31		N/A	Зарезервировано

Таблица 1.630 – Регистр для перезаписи статусного состояния контроллера MC DDR3 – MC_CONSOL_STAT_OR

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STAT. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_CONSOL_STAT будет установлен; ‘0’b – соответствующий бит регистра MC_CONSOL_STAT не будет изменен.
12:15		N/A	Зарезервировано
16	SET_BIT	N/A	Для каждого m бита: ‘1’b – соответствующий бит регистра MC_CONSOL_STAT будет установлен; ‘0’b – соответствующий бит регистра MC_CONSOL_STAT не будет изменен.
17:31		N/A	Зарезервировано

Таблица 1.631 – Регистр для задания признака события – MC_CONSOL_STAT_TRIG

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	TRIGGER_LEVEL	0	Регистр зеркально отображает информацию по регистру MC_CONSOL_STAT. Установка битов данного регистра может определить статусное событие. Для каждого m бита: ‘0’b – если соответствующий бит регистра MC_ CONSOL_STAT равен ‘0’b, статусное событие может быть зафиксировано; ‘1’b – если соответствующий бит регистра MC_ CONSOL_STAT равен ‘1’b, статусное событие может быть зафиксировано.
12:15		0	Зарезервировано
16	TRIGGER_LEVEL	0	Для каждого m бита: ‘0’b – если соответствующий бит регистра MC_ CONSOL_STAT равен ‘0’b, статусное событие может быть зафиксировано; ‘1’b – если соответствующий бит регистра MC_ CONSOL_STAT равен ‘1’b, статусное событие может быть зафиксировано.
17:31		0	Зарезервировано

Таблица 1.632 – Регистр для обнуления признака события – MC_CONSOL_STAT_TRIG_AND

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STAT_TRIG. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_CONSOL_STAT_TRIG будет обнулен; ‘1’b – соответствующий бит регистра MC_CONSOL_STAT_TRIG не будет изменен.
12:15		N/A	Зарезервировано
16	RESET_BIT	N/A	‘0’b – соответствующий бит регистра MC_CONSOL_STAT_TRIG будет обнулен; ‘1’b – соответствующий бит регистра MC_CONSOL_STAT_TRIG не будет изменен.
17:31		N/A	Зарезервировано

Таблица 1.633 – Регистр для перезаписи признака события – MC_CONSOL_STAT_TRIG_OR

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STAT_TRIG. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_CONSOL_STAT_TRIG будет установлен; ‘0’b – соответствующий бит регистра MC_CONSOL_STAT_TRIG не будет изменен.
12:15		N/A	Зарезервировано
16	SET_BIT	N/A	Для каждого m бита: ‘1’b – соответствующий бит регистра MC_CONSOL_STAT_TRIG AT будет установлен; ‘0’b – соответствующий бит регистра MC_CONSOL_STAT_TRIG не будет изменен.
17:31		N/A	Зарезервировано

Таблица 1.634 – Регистр для задания признака перехода немаскируемого события с первого уровня на второй уровень обработки ошибок – MC_CONSOL_STAT_MSK

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	MASK_BIT	xFFF	Регистр зеркально отображает информацию по регистру MC_CONSOL_STAT. Для каждого m бита: ‘0’b – Если соответствующий бит регистра MC_CONSOL_STAT совпадает с соответствующим битом регистра MC_CONSOL_STAT_TRIG, бит ‘INFO_STATUS’ регистра ‘MC_CONSOL_STATSUM’ будет установлен; ‘1’b – бит ‘INFO_STATUS’ регистра ‘MC_CONSOL_STATSUM’ не будет установлен.
12:15		0	Зарезервировано
16	MASK_BIT	1	‘0’b – Если соответствующий бит регистра MC_CONSOL_STAT совпадает с соответствующим битом регистра MC_CONSOL_STAT_TRIG, бит ‘INFO_STATUS’ регистра ‘MC_CONSOL_STATSUM’ будет установлен; ‘1’b – бит ‘INFO_STATUS’ регистра ‘MC_CONSOL_STATSUM’ не будет установлен.
17:31		0	Зарезервировано

Таблица 1.635 – Регистр для обнуления признака перехода немаскируемого события с первого уровня на второй уровень обработки ошибок – MC_CONSOL_STAT_MSK _AND

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STAT_MSK. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_CONSOL_STAT_MSK будет обнулен; ‘1’b – соответствующий бит регистра MC_CONSOL_STAT_MSK не будет изменен.
12:15		N/A	Зарезервировано
16	RESET_BIT	N/A	‘0’b – соответствующий бит регистра MC_CONSOL_STAT_MSK будет обнулен; ‘1’b – соответствующий бит регистра MC_CONSOL_STAT_MSK не будет изменен.
17:31		N/A	Зарезервировано

Таблица 1.636 – Регистр для перезаписи признака перехода немаскируемого события с первого уровня на второй уровень обработки ошибок – MC_CONSOL_STAT_MSK _OR

Биты	Название	Значение после инициализации	Выполняемая функция
0:11	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STAT_MSK. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_CONSOL_STAT_MSK будет установлен; ‘0’b – соответствующий бит регистра MC_CONSOL_STAT_MSK не будет изменен.

Продолжение таблицы 1.637

Биты	Название	Значение после инициализации	Выполняемая функция
12:15		N/A	Зарезервировано
16	SET_BIT	N/A	Для каждого m бита: ‘1’b – соответствующий бит регистра MC_CONSOL_STAT_MSK будет установлен; ‘0’b – соответствующий бит регистра MC_CONSOL_STAT_MSK не будет изменен.
17:31		N/A	Зарезервировано

1.3.8.4.4.5 Описание программно доступных регистров о статусе контроллера MC DDR3 второго уровня

Таблица 1.638 – Регистр для вывода информации о текущем событии – MC_CONSOL_STATSUM

Биты	Название	Значение после инициализации	Выполняемая функция
0	INFO_STATUS	0	В регистр программным образом прописывается ‘1’b, когда происходит статусное событие (регистры первого уровня). Сброс происходит программным образом.
1:31		0	Зарезервировано

Таблица 1.639 – Регистр для сброса информации о текущем событии – MC_CONSOL_STATSUM_AND

Биты	Название	Значение после инициализации	Выполняемая функция
0	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STATSUM. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_CONSOL_STATSUM будет обнулен; ‘1’b – соответствующий бит регистра MC_CONSOL_STATSUM не будет изменен.
1:31		N/A	Зарезервировано

Таблица 1.640 – Регистр для перезаписи информации о текущем событии – MC_CONSOL_STATSUM_OR

Биты	Название	Значение после инициализации	Выполняемая функция
0	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STATSUM. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_CONSOL_STATSUM будет установлен; ‘0’b – соответствующий бит регистра MC_CONSOL_STATSUM не будет изменен.
1:31		N/A	Зарезервировано

Таблица 1.641 – Регистр для задания признака перехода немаскируемой информации о текущем событии со второго уровня обработки информации на верхний уровень системы – MC_CONSOL_STATSUM_MSK

Биты	Название	Значение после инициализации	Выполняемая функция
0	MASK_BIT	‘1’b	Регистр зеркально отображает информацию по регистру MC_CONSOL_STATSUM. ‘1’b – маскируемая ошибка, соответствующий бит регистра MC_CONSOL_STATSUM запрещено передавать на верхний уровень системы (O_MC_INFSTAT_INTRPT = ‘0’b); ‘0’b – немаскируемая ошибка, соответствующий бит регистра MC_CONSOL_STATSUM разрешено передать на верхний уровень системы, как сигнал прерывания (O_MC_INFSTAT_INTRPT присваивает значение бита INFO_STATUS в регистре MC_CONSOL_STATSUM).
1:31		0	Зарезервировано

Таблица 1.642– Регистр для сброса признака перехода немаскируемой информации о текущем событии со второго уровня обработки информации на верхний уровень системы – MC_CONSOL_STATSUM_MSK_AND

Биты	Название	Значение после инициализации	Выполняемая функция
0	RESET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STATSUM_MSK. Для каждого m бита: ‘0’b – соответствующий бит регистра MC_CONSOL_STATSUM_MSK будет обнулен; ‘1’b – соответствующий бит регистра MC_CONSOL_STATSUM_MSK не будет изменен.
1:31		N/A	Зарезервировано

Таблица 1.643 – Регистр для перезаписи признака перехода немаскируемой информации о текущем событии со второго уровня обработки информации на верхний уровень системы – MC_CONSOL_STATSUM_MSK_OR

Биты	Название	Значение после инициализации	Выполняемая функция
0	SET_BIT	N/A	Запись в данный регистр может изменить соответствующие значения регистра MC_CONSOL_STATSUM_MSK. Для каждого m бита: ‘1’b – соответствующий бит регистра MC_CONSOL_STATSUM_MSK будет установлен; ‘0’b – соответствующий бит регистра MC_CONSOL_STATSUM_MSK не будет изменен.
1:31		N/A	Зарезервировано

1.3.8.4.5 Программно реализуемая модель контроллера DDR3MC

1.3.8.4.5.1 Список программно доступных регистров контроллера DDR3MC

В таблице 1.638 приведен полный список регистров контроллера памяти DDR3MC. Смотрите соответствующий раздел для подробного описания каждого регистра.

Таблица 1.644 – Регистры конфигурации контроллера DDR3MC

Название регистра	Описание	Адресное смещение	ЗП/ЧТ	Значение сброса
MCSTAT	Регистр статуса контроллера памяти	x‘10’	ЧТ	x‘6000_0000’
MCOPT1	Регистр настроек контроллера памяти (1)	x‘20’	ЧТ/ ЗП	x‘0102_0000’
MCOPT2	Регистр настроек контроллера памяти (2)	x‘21’	ЧТ/ ЗП	x‘80F3_0000’
MCOPTE1	Регистр настроек порта 1 контроллера памяти	x‘22’	ЧТ/ ЗП	x‘0000_0000’
MCOPTE2	Регистр настроек порта 2 контроллера памяти	x‘23’	ЧТ/ ЗП	x‘0000_0000’
MCOPTE3	Регистр настроек порта 3 контроллера памяти.	x‘24’	ЧТ/ ЗП	x‘0000_0000’
IOCNTL	Регистр управления входом/выходом (Задается пользователем)	x‘30’	ЧТ/ ЗП	x‘0000_0000’
PHYCNTL	Регистр управления PHY(Задается пользователем)	x‘31’	ЧТ/ ЗП	x‘0000_0000’
PHYSTAT	Регистр статуса PHY(Задается пользователем)	x‘32’	ЧТ	Зависит от системы
PHYMASK	Регистр маски статуса прерывания PHY	x‘33’	ЧТ/ ЗП	x‘0000_0000’
CFGR0	Регистр конфигурации ранга памяти 0	x‘40’	ЧТ/ ЗП	x‘0000_0000’
CFGR1	Регистр конфигурации ранга памяти 1	x‘41’	ЧТ/ЗП	x‘0000_0000’
CFGR2	Регистр конфигурации ранга памяти 2	x‘42’	ЧТ/ ЗП	x‘0000_0000’
CFGR3	Регистр конфигурации ранга памяти 3	x‘43’	ЧТ/ ЗП	x‘0000_0000’
RHZEN	Регистр включения перехода в высокоимпедансное состояние	x’48’	ЧТ/ ЗП	x‘0000_0000’
INITSEQn	Регистр последовательности инициализации	x‘50’ - x‘6E’ (четные)	ЧТ/ ЗП	x‘0000_0000’
INITCMDn	Регистр команд инициализации	x‘51’ - x‘6F’ (нечетные)	ЧТ/ ЗП	x‘0000_0000’
SDTR0	Регистр временных параметров DDR SDRAM (0)	x‘80’	ЧТ/ ЗП	x‘0000_0000’
SDTR1	Регистр временных параметров DDR SDRAM (1)	x‘81’	ЧТ/ ЗП	x‘0000_0000’
SDTR2	Регистр временных параметров DDR SDRAM (2)	x‘82’	ЧТ/ ЗП	x‘0000_0000’
SDTR3	Регистр временных параметров DDR SDRAM (3)	x‘83’	ЧТ/ ЗП	x‘0000_0000’
SDTR4	Регистр временных параметров DDR SDRAM (4)	x‘84’	ЧТ/ ЗП	x‘0000_0000’
SDTR5	Регистр временных параметров DDR SDRAM (5)	x‘85’	ЧТ/ ЗП	x‘0000_0000’
DBG0	Регистр отладки (0)	x‘8A’	ЧТ	x‘0000_0000’
SMR0	Регистр теневого режима (0)	x‘90’	ЧТ/ ЗП	x‘0000_0000’
SMR1	Регистр теневого режима (1)	x‘91’	ЧТ/ ЗП	x‘0000_0000’

Продолжение таблицы 1.645

Название регистра	Описание	Адресное смещение	ЗП/ЧТ	Значение сброса
SMR2	Регистр теневого режима (2)	x‘92’	ЧТ/ ЗП	x‘0000_0000’
SMR3	Регистр теневого режима (3)	x‘93’	ЧТ/ ЗП	x‘0000_0000’
SMR4	Регистр теневого режима (4)	x‘94’	ЧТ/ ЗП	x‘0000_0000’
SMR5	Регистр теневого режима (5)	x‘95’	ЧТ/ ЗП	x‘0000_0000’
SMR6	Регистр теневого режима (6)	x‘96’	ЧТ/ ЗП	x‘0000_0000’
ODTR0	Регистр встроенной терминационной логики для ранга 0	x‘A0’	ЧТ/ ЗП	x‘0000_0000’
ODTR1	Регистр встроенной терминационной логики для ранга 1	x‘A1’	ЧТ/ ЗП	x‘0000_0000’
ODTR2	Регистр встроенной терминационной логики для ранга 2	x‘A2’	ЧТ/ ЗП	x‘0000_0000’
ODTR3	Регистр встроенной терминационной логики для ранга 3	x‘A3’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_CNTL	Регистр статуса и управления механизмом очистки	x‘AA’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_INT	Регистр таймера интервала обновления механизма очистки	x‘AB’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_CUR	Регистр текущего адреса исполнения механизма очистки	x‘AC’	ЧТ	x‘0000_0000’
SCRUB_CUR_EXT	Регистр расширенного текущего адреса исполнения механизма очистки	x‘AD’	ЧТ	x‘0000_0000’
SCRUB_PREFILL_DATA0-1	Регистр данных для предварительного заполнения при механизме очистки	x‘AE’ - x‘AF’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_ST_RANKm_PORT0	Регистр адреса начала механизма очистки для порта 0	x‘B0’, x‘B4’, x‘B8’, x‘BC’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_ST_RANKm_PORT1	Регистр адреса начала механизма очистки для порта 1	x‘B1’, x‘B5’, x‘B9’, x‘BD’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_ST_RANKm_PORT2	Регистр адреса начала механизма очистки для порта 2	x‘B2’, x‘B6’, x‘BA’, x‘BE’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_ST_RANKm_PORT3	Регистр адреса начала механизма очистки для порта 3	x‘B3’, x‘B7’, x‘BB, x‘BF	ЧТ/ ЗП	x‘0000_0000’
CALSTAT	Регистр статуса калибровки	x‘C1’	ЧТ/ ЗП	x‘0000_0000’
T_PHYUPD0-3	Регистр таймера обновления DFI PHY 0-3	x‘CC’ - x‘CF’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_ED_RANKm_PORT0	Регистр адреса окончания очистки для порта 0	x‘D0’, x‘D4’, x‘D8’, x‘DC’	ЧТ/ ЗП	x‘0000_0000’
SCRUB_ED_RANKm_PORT1	Регистр адреса окончания очистки для порта 1	x‘D1’, x‘D5’, x‘D9’, x‘DD’	ЧТ/ ЗП	x‘0000_0000’

Продолжение таблицы 1.646

Название регистра	Описание	Адресное смещение	ЗП/ЧТ	Значение сброса
SCRUB_ED_RANKm_PORT2	Регистр адреса окончания очистки для порта 2	x‘D2’, x‘D6’, x‘D9’, x‘DE’	ЧТ /ЗП	x‘0000_0000’
SCRUB_ED_RANKm_PORT3	Регистр адреса окончания очистки для порта 3	x‘D3’, x‘D7’, x‘DA’, x‘DF’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_ADDR_PORT0	Регистр адреса ECC-ошибки порт 0	x‘E0’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_ADDR_PORT1	Регистр адреса ECC-ошибки порт 1	x‘E1’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_ADDR_PORT2	Регистр адреса ECC-ошибки порт 2	x‘E2’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_ADDR_PORT3	Регистр адреса ECC-ошибки порт 3	x‘E3’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_COUNT_PORT0	Регистр счетчика ECC- ошибок порт 0	x‘E4’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_COUNT_PORT1	Регистр счетчика ECC- ошибок порт 1	x‘E5’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_COUNT_PORT2	Регистр счетчика ECC- ошибок порт 2	x‘E6’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_COUNT_PORT3	Регистр счетчика ECC- ошибок порт 3	x‘E7’	ЧТ/ ЗП	x‘0000_0000’
CE_TCNT_CLR	Регистр очистки количества исправляемых ECC-ошибок	x‘E8’	ЧТ/ ЗП	x‘0000_0000’
CE_TCNT_SET	Регистр установки порогового значения количества исправляемых ECC-ошибок	x‘E9’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_ADDR_EXT	Регистр расширенного адреса ECC-ошибки	x‘EA’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_PORT0	Регистр статуса ECC-ошибки порта 0	x‘F0’, x‘F1’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_PORT1	Регистр статуса ECC-ошибки порта 1	x‘F2’, x‘F3’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_PORT2	Регистр статуса ECC-ошибки порта 2	x‘F4’, x‘F5’	ЧТ/ ЗП	x‘0000_0000’
ECCERR_PORT3	Регистр статуса ECC-ошибки порта 3	x‘F6’, x‘F7’	ЧТ/ ЗП	x‘0000_0000’
ECC_CHECK_PORT0	Регистр проверки ECC-ошибки порта 0	x‘F8’	ЧТ/ ЗП	x‘0000_0000’
ECC_CHECK_PORT1	Регистр проверки ECC-ошибки порта 1	x‘F9’	ЧТ/ ЗП	x‘0000_0000’
ECC_CHECK_PORT2	Регистр проверки ECC-ошибки порта 2	x‘FA’	ЧТ/ ЗП	x‘0000_0000’
ECC_CHECK_PORT3	Регистр проверки ECC-ошибки порта 3	x‘FB’	ЧТ/ ЗП	x‘0000_0000’

1.3.8.4.5.2 Конфигурация устройства

Конфигурационные регистры контроллера динамической памяти DDR34LMC должны быть настроены согласно требованиям подсистемы памяти микросхемы и быть использованы для реализации корректной работы контроллера с внешней памятью типа SDRAM. Для защиты подсистемы памяти от незапланированных обращений, аппаратным образом устанавливается запрет на запись для определенного количества регистров после того, как контроллер памяти DDR34LMC приступил к работе (MCOPT2[MC_ENABLE] равен 1’b1).

В таблице (см. Таблица 1.647) представлены все конфигурационные регистры контроллера DDR34LMC и их зависимость от транзакций записи и чтения при включенном и выключенном состоянии контроллера DDR34LMC.

Важно: Зарезервированные поля конфигурационных регистров памяти не должны быть изменены во время транзакции записи. Изменение каких-либо зарезервированных полей значениями, отличных от тех, что установлены по умолчанию после снятия сигнала сброса, может привести к непредсказуемым результатам в работе всей системы контроллера динамической памяти DDR34LMC.

Таблица 1.647 - Зависимость конфигурационных регистров от MCOPT2[MC_ENABLE]

Название регистра	Описание	Адресное смещение	MC_ ENABLE	ЧТ/ЗП	Результат
MCSTAT	Регистр статуса контроллера памяти	x‘10’	X	ЗП	Не меняется
MCOPT1	Регистр настроек контроллера памяти (1)	x‘20’	X	ЗП	Обновляется
MCOPT2	Регистр настроек контроллера памяти (2)	x‘21’	X	ЗП	Обновляется
MCOPTE1	Регистр настроек порта 1 контроллера памяти	x‘22’	X	ЗП	Обновляется
MCOPTE2	Регистр настроек порта 2 контроллера памяти	x‘23’	X	ЗП	Обновляется
MCOPTE3	Регистр настроек порта 3 контроллера памяти	x‘24’	X	ЗП	Обновляется
IOCNTL	Регистр управления входом/выходом	x‘30’	X	ЗП	Обновляется
PHYCNTL	Регистр управления PHY	x‘31’	X	ЗП	Обновляется
PHYSTAT	Регистр статуса PHY	x‘32’	X	ЗП	Не меняется
PHYMASK	Регистр маски статуса прерывания PHY	x‘33’	X	ЗП	Обновляется
CFGR0-3	Регистры конфигурации рангов памяти 0-3	x‘40’ - x‘43’	0	ЗП	Обновляется
			1	ЗП	Не меняется
RHZEN	Регистр включения перехода в высокоимпедансное состояние	x’48’	X	ЗП/ЧТ	Обновляется
INITSEQn	Регистр последовательности инициализации	x‘50’ - x‘6E’	X	ЗП	Обновляется
INITCMDn	Регистр команд инициализации	x‘51’ - x‘6F’	X	ЗП	Обновляется

Продолжение таблицы 1.648

Название регистра	Описание	Адресное смещение	MC_ ENABLE	ЧТ/ЗП	Результат
SDTR0-5	Регистры временных параметров DDR SDRAM 0-5	x‘80’ - x‘85’	X	ЗП*	Обновляется
SMR0-7	Регистры теневого режима (0-7)	x‘90’ - x‘97’	0	ЗП	Обновляется
			1	ЗП	Не меняется
ODTR(n)	Регистры встроенной терминационной логики для рангов 0-3	x‘A0’ - x‘A3’	X	ЗП	Обновляется
SCRUB_CNTL	Регистр статуса и управления механизмом очистки	x‘AA’	X	ЗП	Обновляется
SCRUB_INT	Регистр таймера интервала обновления механизма очистки	x‘AB’	X	ЗП	Обновляется
SCRUB_CUR	Регистр текущего адреса исполнения механизма очистки	x‘AC’	X	ЗП	Не меняется
SCRUB_CUR_EXT	Регистр расширенного текущего адреса исполнения механизма очистки	x‘AD’	X	ЗП	Не меняется
SCRUB_PREFILL_ DATA0-1	Регистр данных для предварительного заполнения при механизме очистки	x‘AE’ - x‘AF’	X	ЗП	Обновляется
SCRUB_ST_RANK m_PORTx	Регистры адреса начала механизма очистки для портов 0-3	x‘B0’ - x‘BF’	X	ЗП	Обновляется
CALSTAT	Регистр статуса калибровки	x‘C1’	X	ЗП	Обновляется
T_PHYUPD0-3	Регистр таймера обновления DFI PHY 0-3	x‘CC’ - x‘CF’	X	ЗП	Обновляется
SCRUB_ED_RANK m_ PORTx	Регистры адреса окончания очистки для портов 0-3	x‘D0’ - x‘DF’	X	ЗП	Обновляется
ECCERR_ADDR_P ORTx	Регистры адреса ECC-ошибки портов 0-3	x‘E0’ - x‘E3’	X	ЗП	Обновляется
ECCERR_COUNT_ PO RTx	Регистры счетчика ECC- ошибок портов 0-3	x‘E4’ - x‘E7’	X	ЗП	Не меняется
CE_TCNT_CLR	Регистр очистки количества исправляемых ECC-ошибок	x‘E8’	X	ЗП	Обновляется
CE_TCNT_SET	Регистр установки порогового значения количества исправляемых ECC-ошибок	x‘E9’	X	ЗП	Обновляется
ECCERR_ADDR_E XT	регистр расширенного адреса ECC-ошибки	x‘EA’	X	ЗП	Обновляется
ECCERR_PORTx	Регистры статуса ECC-ошибки портов 0-3	x‘F0’ - x‘F7’	X	ЗП	Очистка/ установка маски

Продолжение таблицы 1.649

Название регистра	Описание	Адресное смещение	MC_ ENABLE	ЧТ/ЗП	Результат
ECC_CHECK_POR Tx	Регистры проверки ECC-ошибки портов 0-3	x‘F8’ - x‘FB’	X	ЗП	Не обновляется
Любой	Любой	Любой	X	ЧТ	Чтения данных

Примечание: *X означает «любое значение»*
“Port x” (где x = 0 - 3) относятся к одному из 4-х MCIF интерфейсов.
“Rank m” (где m = 0 - 3) относятся к одному из 4-х SDRAM рангов.

1.3.8.4.5.2.1 Начальная конфигурация после сброса при включенном питании

Конфигурация всех регистров контроллера динамической памяти DDR3MC должна быть задана до его включения. Включение производится установкой MCOPT2[MC_ENABLE] в 1.

Запись в MCOPT1 и SDTR0-4 возможна после включения контроллера, и не зависит от значения MCOPT2[MC_ENABLE].

Внимание: Необходимо гарантировать, что контроллер памяти DDR34LMC бездействует во время изменения или обновления значений MCOPT1. В противном случаи, текущие обращения к памяти могут проходить не корректно.

Транзакции чтение данных из регистров происходит независимо от значения MCOPT2[MC_ENABLE].

1.3.8.4.5.2.2 Изменение начальной конфигурации

Изменение конфигурации некоторых регистров контроллер памяти DDR34LMC может осуществляться в любой момент после задания начальной конфигурации. Перед изменением конфигураций регистров, все текущие и ожидающие в очереди запросы, направленные в контроллер памяти должно быть завершены. Далее описана последовательность действий при изменении конфигурации регистров:

- Обеспечить завершение всех текущих и ожидающих в очереди запросов к памяти. Подтверждается установкой MCSTAT[IDLE] в 1.
- Отключить контроллер памяти, записав в MCOPT2[MC_ENABLE] значение 0.
- Обновить регистры временных параметров.
- Если были внесены какие-либо изменения в регистры SMR0-3, устройство памяти должно быть повторно инициализировано, чтобы эти изменения загрузились в запоминающее устройство. Это должно быть сделано с помощью регистров INITSEQ и CMD.
- Кроме того, при существенных изменениях в функциональном содержимом этих регистров, может потребовать повторной калибровки интерфейса PHY-to-SDRAM. Если это так, выполните необходимые процедуры калибровки.
- Включить контроллер памяти при помощи записи единицы в MCOPT2[MC_ENABLE]. Теперь память может быть использована.

1.3.8.4.5.3 Описание программно доступных регистров контроллера DDR34LMC

1.3.8.4.5.3.1 Регистр статуса контроллера памяти – MCSTAT

Адресное смещение: x‘10’
Значение Сброса: x‘6000_0000’
Доступ: ЧТ

Этот регистр (Рисунок 1.101) хранит статус системы о работе ядра контроллера динамической памяти DDR34LMC в режиме реального времени. Эта информация показывает статус контроллера памяти в момент чтения данных из регистра.

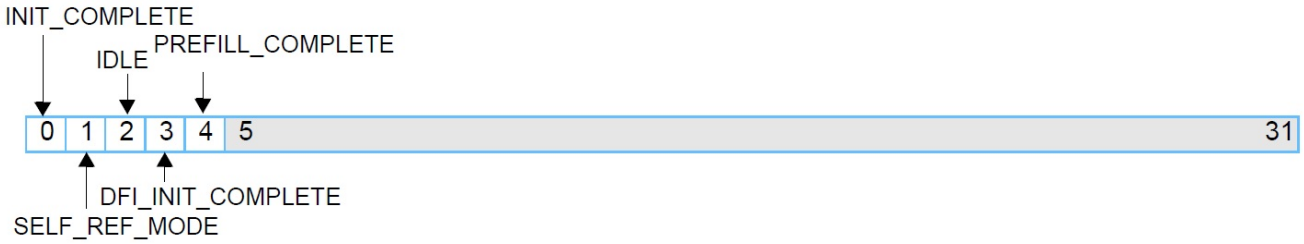


Рисунок 1.101 - MCSTAT

Таблица 1.650 - Регистр статус контроллера памяти

Бит	Название	Значение сброса	Описание
0	INIT_COMPLETE	0	Статус завершения последовательности инициализации памяти. Установка этого бита в 1 указывает на завершение предварительной загрузки последовательности инициализации в устройство памяти. Значение устанавливается в 0 всякий раз, когда бит начала инициализации в регистре MCOPT2 установлен в 1.
1	SELF_REF_MODE	1	Статус режима регенерации 0 - SDRAM не в режиме регенерации 1 - SDRAM в режиме регенерации Этот бит устанавливается в 1, когда устройства SDRAM находится в режиме регенерации. После завершения регенерации этот бит устанавливается в 0. Этот бит относится как к программно включаемому режиму регенерации, который устанавливается с помощью MCOPT2[SELF_REF_EN], и к аппаратно включаемому режиму регенерации.
2	IDLE	1	Статус простоя контроллера памяти. 0 – Не простаивает 1 – Простаивает Этот бит показывает на отсутствие текущих и ожидающих в очереди транзакций чтения/записи в контроллер памяти.
3	DFI_INIT_COMPLETE	0	Статус завершения инициализации DFI. Установленный в 1 этот бит показывает, что PHY устанавливает свой выходной сигнал завершения инициализации. На основании установки этого бита происходит установка MCOPT2[DFI_INIT_START], что, в свою очередь запускает инициализацию PHY.
4	PREFILL_COMPLETE	0	Завершение предварительного заполнения памяти. (Не используется)
5:31		0s	Зарезервировано

1.3.8.4.5.3.2 Регистр настроек контроллера памяти 1 – MCOPT1

Адресное смещение: x‘20’
Значение Сброса: x‘0102_0000’
Доступ: ЗП/ЧТ

Этот регистр (см. Рисунок 1.102) позволяет задавать настройки для контроллера динамической памяти DDR34LMC. Все настройки, задаваемые в MCOPT1, MCOPT2 и SDTR0-4, должны быть заданы до включения контроллера динамической памяти DDR34LMC.

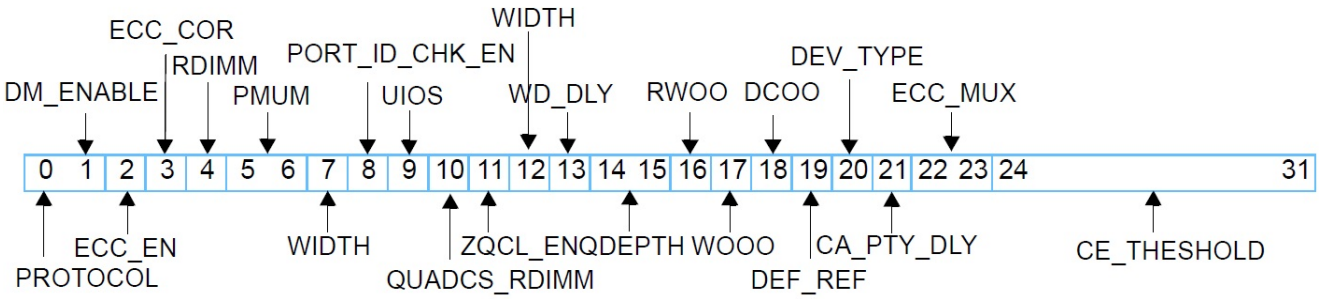


Рисунок 1.102 - MCOPT1

Таблица 1.651 - Регистр настроек контроллера памяти (1)

Бит	Название	Значение сброса	Описание
0	PROTOCOL	0	Протокол устройства памяти. 0 - DDR3
1	DM_ENABLE	0	Активация маски данных. 0 – Отключение маски данных. Принудительное использование последовательности чтение-модификация-запись (RMW) для частичной записи. 1 – Использование маски данных для частичной записи, если это возможно.
2	ECC_EN	0	Включение, в контроллере памяти режима встроенного ECC. 0 – Отключено 1 – Включено (не поддерживается)
3	ECC_COR	0	Включение, встроенной в контроллер памяти коррекции и проверки ECC. 0 – Отключено 1 – Включено (не поддерживается)
4	RDIMM	0	Включение DIMM. 0 - Отключен 1 - Включен **Примечание:** Этот бит влияет на временной промежуток между адресом и данными, передаваемыми в SDRAM, что провидит к дополнительным задержкам, вносимым с помощью регистров адреса/управления в модуле DIMM.
5:06	PMUM	0	Режим управления страницами 00 - Закрытие. Всегда закрытая страница после завершения команд чтения или записи. 01 – Автоматическое закрытие. Закрытая страница после транзакции чтения или записи, если отсутствует очередной запрос к этой странице 10 - Открыто. После использования, страница остается открытой до обращения к другой странице или до завершения обновления памяти. 11 – Зарезервировано

Продолжение таблицы 1.652

Бит	Название	Значение сброса	Описание
7	WIDTH[0]	1	WIDTH[0:1]: Выбор режима размера данных DDR-to-MCIF 2'b10 – Зарезервировано 2'b00 – Режим 1\4 ширины данных MCIF (Режим 1\4) 2'bx1 – Режим 1\8 ширины данных MCIF (Режим 1\8), только для пакета, длинной BL8
8	PORT_ID_CHK_EN	0	Функция проверки ID порта. 0 - Отключение проверки ID внеочередных операций MCIF 1 - Включение проверки ID внеочередных операций MCIF
9	UIOS	0	Состояния неиспользуемых входов/выходов. 0 - Активен (Высокоимпедансное состояние только при транзакциях чтения) 1 – Третье состояние (Всегда высокоимпедансное состояние) Это поле может использоваться для управления состоянием неиспользуемых входов/выходов, которые могут быть отключены соответствующими сигналами.
10	QUADCS_RDIMM	0	Включение 4x-ранговый RDIMM (MCOPT1[RDIMM] должен быть установлен в 1, когда QUADCS_RDIMM=1)
11	ZQCL_EN	0	Включение ZQCL команд на выходе из регенерации. 0 - Нет ZQCL команд. 1 - ZQCL команда выдается на все включенные ранги. После выхода из режима регенерации и выждав tXS, команда ZQCL выдается (если разрешена) на все включенные ранги.Далее, после задержки tXSDLL ядру разрешается осуществлять транзакции чтения и записи.
12	WIDTH[1]	1	Выбор режима размера данных для DDR SDRAM (Описание смотреть выше)
13	WD_DLY	0	Задержка между запросом данных записи MCIF и выдачей данных записи (в тактовых циклах MCIF) 0 - 1 ответный цикл 1 - 2 ответный цикл
14:15	QDEPTH	10	Глубина очереди команд MCIF 10 – глубина очереди команд 4 11 – зарезервировано Другие значения - Зарезервированы
16	RWOO	0	Включение out-of-order операций чтения и записи (Out-of-order операции должны быть включены всеми элементами управления). 0 - Выключено 1 - Включено
17	WOOO	0	Включение out-of-order операций записи 0 - Выключено 1 - Включено
18	DCOO	0	Отключение всех out-of-order операций, включая управление страницами 0 - Включение out-of-order управления страницами 1 - Выключение out-of-order управления страницами
19	DEF_REF	0	Включение приостановки обновления. Эта функция изменяет сроки автоматического обновления, чтобы избежать задержки в очереди команд чтения и записи. 0 – Обычное обновление 1 – Приостанавливает до 4x команд регенерации, если контроллер занят транзакциями записи/чтения **Примечание**: Функция приостановки обновления должна быть включена после завершения калибровки.

Продолжение таблицы 1.653

Бит	Название	Значение сброса	Описание
20	DEV_TYPE	0	Тип устройства памяти 0 - устройство памяти x4 или x8 1 – устройство памяти x16
21	CA_PTY_DLY	0	Задержка сигнала проверки четности команд/адреса 0 – дополнительная задержка отсутствует 1 - сигнал задерживается на 1 DFI тактовый цикл по отношения к Командам/Адресу, которые он проверяет (рекомендуется)
22:23	ECC_MUX	0s	Включение байтового мультиплексора. Не используется
24:31	CE_THRESHOLD	0s	Пороговое значение счетчика CE Не используется

1.3.8.4.5.3.3 Регистр настроек контроллера памяти 2 – MCOPT2

Адресное смещение: x‘21’

Значение сброса: x‘80F3_0000’

Доступ: ЧТ/ЗП

Регистр (Рисунок 1.102) содержит настройки конфигурации контроллера памяти DDR3, последовательности инициализации, управления мощностью, а так же особенностями регенерации.

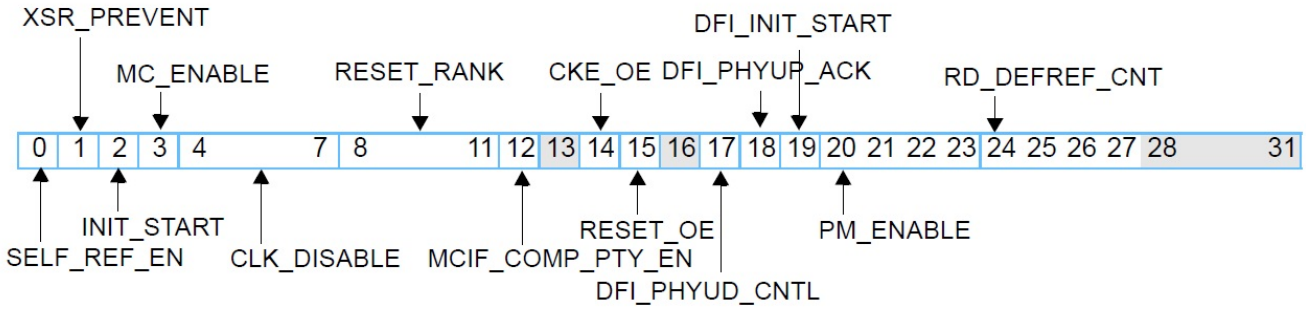


Рисунок 1.103 - MCOPT2

Таблица 1.654 - Регистр настроек контроллера памяти 2

Бит	Название	Значение сброса	Описание
0	SELF_REF_EN	1	Управление регенерацией 0 – Выход из режима регенерации 1 – Вход в режим регенерации Этот бит используется для управления входом и выходом в режим регенерации. Он устанавливается в 1 во время сброса с включенным питанием, а так же может быть установлен в 1 при помощь входного сигнала. Программная или аппаратная установка этого бита приводит к очистке бита ENABLE из MCOPT2 в 0, отключению контроллера и остановке автоматического обновления. Этот бит может быть очищен программно и аппаратно что приведет к выходу из режима регенерации при помощи реактивации CKE в памяти.
1	XSR_PREVENT	0	Предотвращение выхода из режима регенерации. 0 – Позволяет выйти из режима регенерации при любых условиях. 1 – Предотвращает выход контроллера и памяти из режима регенерации кроме первого выхода после включении/выключения или сброса. Этот бит должен устанавливаться и очищаться программно. Когда он установлен, память может войти в режим регенерации, но не может из него выйти, пока бит не очиститься программно или сбросом.
2	INIT_START	0	Статус и запуск инициализации. 0 – Генератор последовательности инициализации в режиме IDLE (ожидание) 1 – Выполнение предварительной инициализации или полной инициализация Запись единицы в INIT_START запускает предварительную последовательность инициализации направленную в устройство памяти. Настройки инициализации задаются в регистрах INITSEQn и CMDn. В результате MCSTAT[INIT_COMPLETE] автоматически очищается (в 0). Однажды установленный этот бит остается таким до тех пор, пока аппаратные средства не завершат запрограммированную последовательность инициализации. После завершения последовательности предварительной инициализации, это бит очищается, а MCSTAT[INIT_COMPLETE] устанавливается в 1. Этот бит никогда не должен программно устанавливаться в 0.Он может быть установлен только в 1 (в случае, если генератор последовательности инициализации находится в режиме ожидания, т.е. INIT_START = 0)
3	MC_ENABLE	0	Включение контроллера внешней динамической памяти DDR34LMC памяти. 0 – Контроллер выключен 1 – Контроллер включен для стандартных операций. Это бит позволяет программно включать или выключать стандартные операции с памятью и автоматическое обновление. Перед установкой этого бита в 1, необходимо программно выйти из режима регенерации, если эта функция включена, запрограммировать все регистры управления и временных параметров в допустимые значения и произвести последовательность инициализации. Очистка этого бита предотвращает прием команд памяти. Это бит очищается сбросом или входом в режим регенерации. Он автоматически устанавливается, когда выход из режима регенерации осуществляется аппаратно.

Таблица 1.655 - Регистр настроек контроллера памяти 2

Бит	Название	Значение сброса	Описание
4:7	CLK_DISABLE	0000	Отключение тактовых сигналов на рангах DFI xxx1 - Отключает тактовый сигнал на ранге 0 xx1x - Отключает тактовый сигнал на ранге 1 x1xx - Отключает тактовый сигнал на ранге 2 1xxx - Отключает тактовый сигнал на ранге 3 Эти биты обеспечивает независимый контроль тактовых сигналов интерфейса SDRAM. В активном состоянии ожидается, что DDR3 PHY будет удерживать тактовые сигналы в константе. В неактивном состоянии тактовые сигналы должны быть стандартными.
8:11	RESET_RANK	1111	Сброс для рангов DFI xxx1 - Активирует сброс для ранга 0 xx1x - Активирует сброс для ранга 1 x1xx - Активирует сброс для ранга 2 1xxx - Активирует сброс для ранга 3 Этот бит обеспечивает независимый программный контроль сбросом 4х рангов памяти.
12	MCIF_COMP_PTY_EN	0	Включение подтверждения проверки четности, запрашиваемым ведомым интерфейсом MCIF 0 - Выключено 1 - Включено
13		0	Зарезервировано
14	CKE_OE	1/0	Управление включением выхода CKE. 0 - отключено 1 - включено Этот бит используется для активации или деактивации сигнала, который используется для управления выхода CKE.
15	RESET_OE	0/1	Управление включения выхода RESET_N Этот сигнал используется для активации и деактивации сигнала O_MC_RESET_OE.
16		0	Зарезервировано
17	DFI_PHYUD_CNTL	0	Режим управление обновлением DFI PHY 0 – Аппаратное подтверждение запроса обновления PHY на интерфейсе DFI 1 – Программное подтверждение запроса обновления PHY на интерфейсе DFI
18	DFI_PHYUD_ACK	0	Управление подтверждением обновления от DFI PHY к PHY Этот бит управляет подтверждением в ответ на запрос. Этот бит действителен только DFI_PHYUD_CNTL=1
19	DFI_INIT_START	0	Управление началом инициализации DFI. Когда бит установлен в 1, начинается процедура инициализации PHY.

Продолжение таблицы 1.656

Бит	Название	Значение сброса	Описание
20:23	PM_ENABLE	0000	Используется для включения всех функций управления питанием, включая возврат в высокоимпедансное состояние и Clock Gating в MCP 1xxx - Конфигурационно-зависимый Clock Gating для тактового домена #1 x1xx – Конфигурационо-исполнимый Clock Gating для тактового домена #1 xx1x - Конфигурационо-исполнимый Clock Gating для тактового домена #1 xxx1 – Возврат в высокоимпедансное состояние 0 – Соответствующая функция управления питанием отключена (по умолчанию) 1 - Соответствующая функция управления питанием включена Это поле должно быть стабильно, когда MC_ENABLE установлен.
24:27	RD_DEFREF_CNT	0000	Определяет максимальное количество отложенных команд обновления во время ожидания системой команд чтения. “0000” = Не откладывать команды обновления во время ожидания системной команды чтения “0001”-”0111” = Откладывать от 1 до 7 команды обновления во время ожидания системной команды чтения Остальные значения зарезервированы Если MCOPT1.DEF_REF действителен и RD_DEFREF_CNT меньше 4, то до 4х команд обновления могут быть отложены до прихода команды чтения. Это поле должно быть стабильно, когда установлен MC_ENABLE.
28:31		0s	Зарезервировано

1.3.8.4.5.3.4 Регистр настроек порта X контроллера памяти – MCOPTEx

Адресное смещение: x‘22’ - x‘24’
Значение сброса : x‘0000_0000’
Доступ: ЧТ/ЗП

Эти регистры используются только для встроенного механизма ECC (встроенный механизм ECC отсутствует).

1.3.8.4.5.3.5 Регистр управления входом/выходом – IOCNTL

Адресное смещение : x‘30’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Этот регистр (Рисунок 1.104) зарезервирован для статических сигналов, которые определяют атрибуты внешних входных/выходных и приемников внутри DDR PHY. Биты этого регистра не могут быть предопределены и определяются пользователем или интегратором этого ядра на уровне чипа. Эти биты используются для управления функциями входа/выхода такими как: Терминационное Сопротивление, Сопротивления Драйвера, сигналы ZCNTL, сигналы VREF и другие.

Внимание: Биты с 0 по 31 в этом регистре непосредственно отображаются на выходе O_PHY_IO_CONTROL[31:0]. Обратите внимание на порядок битов.

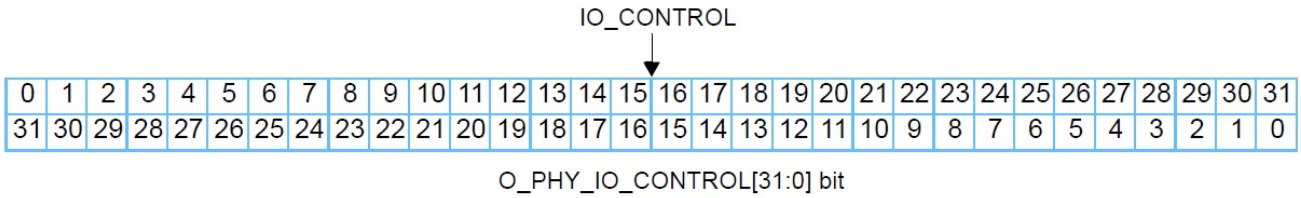


Рисунок 1.104 - IOCNTL

Таблица 1.657 - Регистр управления входом/выходом

Бит	Название	Значение сброса	Описание
0:31	IO_CONTROL	0s	0 = выход O_PHY_IO_CONTROL в 0 1 = выход O_PHY_IO_CONTROL в 1

1.3.8.4.5.3.6 Регистр управления PHY – PHYCNTL

Адресное смещение: x‘31’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Этот регистр (Рисунок 1.105) зарезервирован для статических сигналов, которые определяют атрибуты внешних входных/выходных драйверов и приемников внутри DDR PHY. Биты этого регистра не могут быть предопределены и определяются пользователем или интегратором этого ядра на уровне чипа. Эти биты могу использоваться для управления функциями, которые определенны пользователем.

Биты с 0 по 31 в этом регистре непосредственно отображаются на выходе O_PHY_CONTROL[31:0]. Обратите внимание на порядок битов.

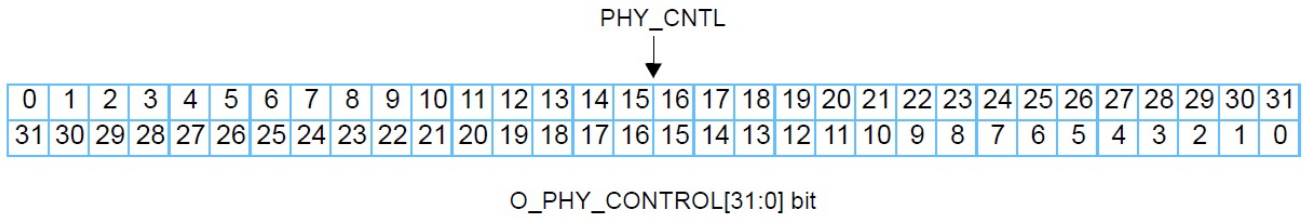


Рисунок 1.105 - PHYCNTL

Таблица 1.658 - Управление выходами DDR3 PHY

Бит	Название	Значение сброса	Описание
0:31	PHY_CNTL	0s	0 = выход O_PHY_CONTROL в 0 1 = выход O_PHY_CONTROL в 1

1.3.8.4.5.3.7 Регистр статуса PHY – PHYSTAT

Адресное смещение: x‘32’
Значение сброса: x‘0000_0000’
Доступ: Только ЧТ

Биты с 0 по 31 в этом регистре (Рисунок 1.106Рисунок 1.106) непосредственно отображают логическое состояние на вход I_PHY_STATUS[31:0]. Это - индикаторы статуса, чувствительные к уровню, которые обеспечивают обнаружение спада, фронта или фиксации состояния на входе.

Примечание: Обратите внимание на порядок битов.

Все биты этого регистра могут использоваться для активации выходных прерываний при помощи включения соответствующих битов в регистре PHYMASK (Адресное смещение 0x33).

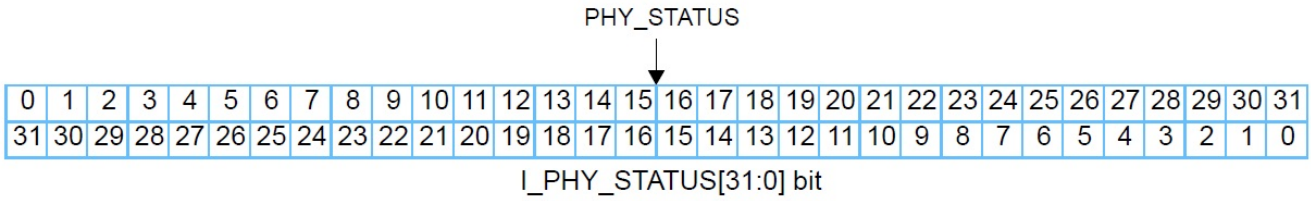


Рисунок 1.106 - PHYSTAT

Таблица 1.659 - Регистр статуса DDR3 PHY

Бит	Название	Значение сброса	Описание
0:31	PHY_STATUS	Зависит от системы	0 = вход I_PHY_STATUS в 0 1 = вход I_PHY_STATUS в 1

Примечание: Биты PHYSTAT[0:15] стационарные (программным образом все эти биты удерживаются в 1 до сброса), а PHYSTAT[16:31] не стационарные (все эти биты захватывают текущее входное состояние).

Примечание: Только биты PHYSTAT[0:15] могут быть заданы, и запись 1 очищает соответствующие биты.

1.3.8.4.5.3.8 Регистр маски статуса прерывания PHY – PHYMASK

Адресное смещение: x‘33’
Значения сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Этот регистр (Рисунок 1.107) используется для активации (1) или деактивации (0) включения соответствующих битов в регистре PHYSTAT для активации выхода прерываний. Сигнал прерывания не появляется до тех пор, пока не активны биты (установлены в 1), участвующие в создании сигнала прерывания и соответствующие биты PHYSTAT не установлены в 1.

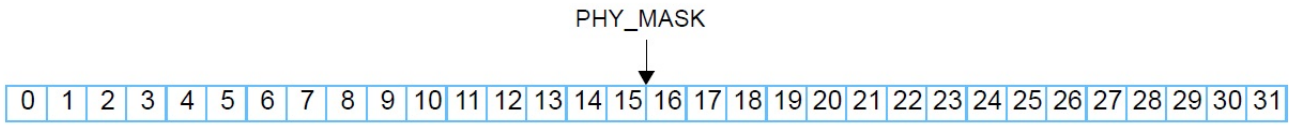


Рисунок 1.107 - PHYMASK

Таблица 1.660 - Регистр маски статуса прерывания DDR3 PHY

Бит	Название	Значение сброса	Описание
0:31	PHY_MASK	0s	0 = Соответствующий бит PHY_STATUS не участвует в создании прерывания 1 = Соответствующий бит PHY_STATUS участвует в создании прерывания

1.3.8.4.5.3.9 Регистры конфигурации рангов памяти – CFGR0-3

Адреса смещения: x‘40’, x‘41’, x‘42’, x‘43’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Регистры CFGRx (Рисунок 1.108) должны быть запрограммированы для определения режима адресации для каждого из рангов памяти. Котроллер памяти DDR3 поддерживает до 4х рангов памяти. Не пытайтесь использовать недействительные или выходящие за диапазон адреса, поступающие на интерфейс MCIF-2. Вся проверка недействительных адресов должна проводиться внешним блоком и все запросы на интерфейсе MCIF-2 должны соответствовать размеру ранга, к которому осуществляется обращение.

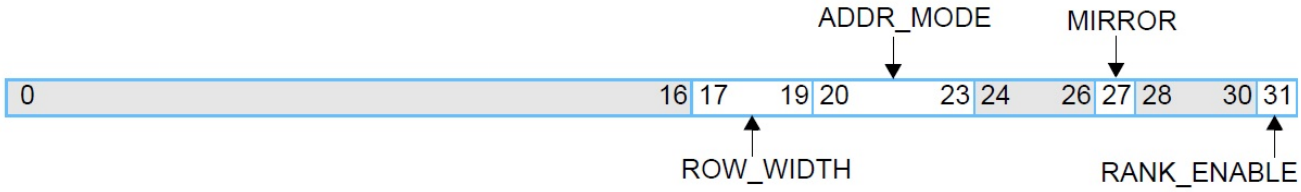


Рисунок 1.108 - CFGRx

Таблица 1.661 - Регистр конфигурации рангов памяти 0-3

Бит	Название	Значение сброса	Описание
0:16		0s	Зарезервировано
17:19	ROW_WIDTH	0s	Размер адреса строки (в битах) 000 - 12 бит 001 - 13 бит 010 - 14 бит 011 - 15 бит 100 - 16 бит 101 - 17 бит Другие - Зарезервированы Это поле устанавливает размер адреса строки ранга (в битах). Оно используется при наличии встроенного механизма ECC и встроенного механизма очистки (в дизайне не присутствуют).
20:23	ADDR_MODE	0	Режимы адресации (размер адреса CAS) 0000 - Режим 0 = Зарезервировано (Nx8) 0001 - Режим 1 = Зарезервировано (Nx9) 0010 - Режим 2 = Nx10 (8 банков) 0011 - Режим 3 = Nx11 (8 банков) 0100 - Режим 4 = Nx12 (8 банков) Другое - Зарезервировано
24:26		0s	Зарезервировано
27	MIRROR	0	Включение зеркального отражения (mirroring) адреса 0 - Отключение зеркального отражения (mirroring) адреса 1 - Включение зеркального отражения (mirroring) адреса Этот бит включает функцию зеркального отражения адреса, доступную на некоторых DIMM с двойным рангом.
28:30		0s	Зарезервировано
31	RANK_ENABLE	0	Включение ранга памяти 0 – Отключить данный ранг 1 – Включить данный ранг

1.3.8.4.5.3.10 Регистр включения перехода в высокоимпедансное состояние – RHZEN

Адресное смещение: x48
Значение сброса: x0000_0000
Доступ: ЧТ/ЗП

Когда MCOPT2[PM_ENABLE] установлен в xxx1 каждый бит регистра (Рисунок 1.109) указывает, должен или нет соответствующий сигнал DDR-интерфейса быть в высокоимпедансном состоянии в период неактивности MC.

0 = Высокоимпедансное состояние отсутствует (по умолчанию)
1 = Устанавливает высокоимпедансное состояние

Во время основных операций, переход в высокоимпедансное состояние осуществляется, когда MC выводит на DDR интерфейс команду дезактивации устройства (DES). Во время регенерации, переход в высокоимпедансное состояние осуществляется после выдачи команды входа в режим регенерации и держится до выдачи команды выхода из режима регенерайции.

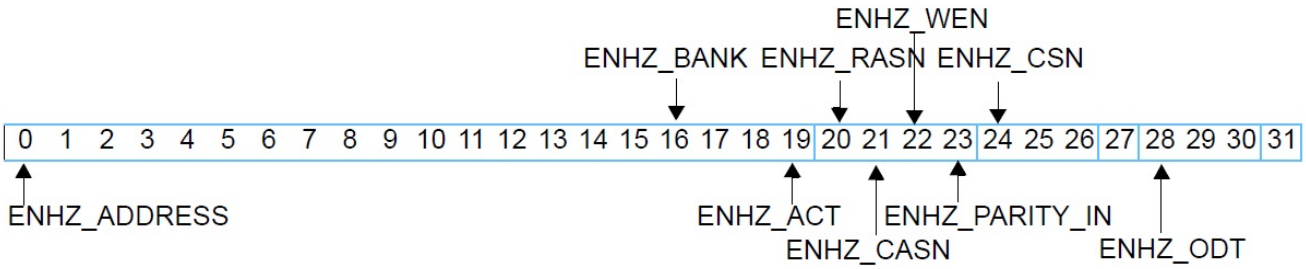


Рисунок 1.109 - RHZEN

Таблица 1.662 - Регистр включения перехода в высокоимпедансное состояние

Бит	Название	Значение сброса	Описание
0:15	ENHZ_ADDRESS	0s	Отвечают за выходные сигналы адреса
16:18	ENHZ_BANK	0	Отвечают за выходные сигналы адреса банка
19	ENHZ_ACTN	0	(Не используется в DDR3)
20	ENHZ_RASN	0	Отвечает за выходной сигнал выбора адреса строки
21	ENHZ_CASN	0	Отвечает за выходной сигнал выбора адреса столбца
22	ENHZ_WEN	0	Отвечает за выходной сигнал разрешения чтения
23	ENHZ_PARITY_IN	0	Отвечает за выходной сигнал проверки четности Команд/Адреса
24:27	ENHZ_CSN	0	Отвечают за выходные сигналы выбора чипа Обратите внимание, что когда этот бит, задан, переход в высокоимпедансное состояние этих сигналов разрешено только во время регенерации SDRAM.
28:31	ENHZ_ODT	0	Отвечает за выходной сигнал встроенной терминационной логики Обратите внимание, что когда этот бит задан, переход в высокоимпедансное состояние этих сигналов разрешено только во время регенерации SDRAM.

1.3.8.4.5.3.11 Регистры последовательности инициализации и команд инициализации SDRAM – INITSEQn and INITCMDn

Адресное смещение: x‘50’ - x‘6F’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Эта группа из 16 пар регистров (Рисунок 1.110 Рисунок 1.111) обеспечивает программное управление конфигурацией различных шагов последовательности инициализации SDRAM. Регистры последовательности с четными адресами, совместно с регистрами команд с нечетными адресами используются для создания временных и логических параметров.

Примечание: Команды настройки регистров режима, выдаваемые во время инициализирующей последовательности, должны настраивать SDRAM в режиме, совместимый с конфигурацией контроллера памяти, которая запрограммирована в регистрах теневого режима SDRAM (SMR0-3).

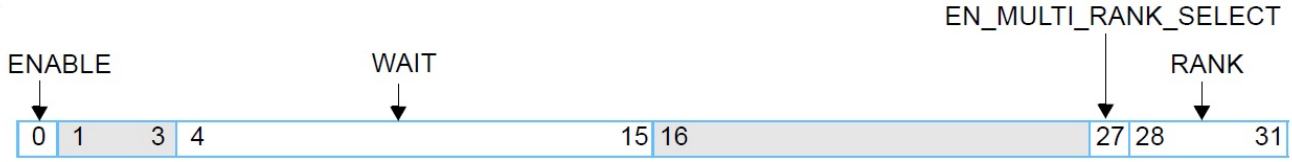


Рисунок 1.110 - INITSEQn (четные адреса)

Таблица 1.663 - Регистр последовательности инициализации

Бит	Название	Значение сброса	Описание
0	ENABLE	0	0 - Отключить передачу данной команды на интерфейс памяти 1 - Включить передачу данной команды на интерфейс памяти
1:03		0s	Зарезервировано.
4:15	WAIT	0s	Минимальное ожидание (в периодах тактового сигнала контроллера памяти). Это поле определяет, через какое минимальное время после данной команды на устройство памяти будет выдана следующая команда. Это значение минимально, но на самом деле, оно может быть больше. Протокол памяти часто выставляет требование на задержку между командами, и она должна быть задана здесь. **Примечание:** Значения, программируемые внутри регистров SDTRn, не используются, пока выполняется INITSEQ/CMD последовательность.
16:26		0s	Зарезервировано
27	EN_MULTI_RANK_ SELECT	0	Включение выбора нескольких рангов. Для пар INITSEQn/ INITCMDn если осуществляется выбор нескольких рангов (несколько бит RANK[0:3] установлены в 1), этот бит управляет, будут ли сигналы выбора чипа утверждены в одном и том же тактовом цикле DDR. 0 - Выборы нескольких чипов утверждаются через несколько тактов DDR 1 - Выборы нескольких чипов утверждаются в одном и том же такте DDR
28:31	RANK	0	Выбор ранга инициализации. Это 4х-битное поле определяет сигнал выбора чипа, выдаваемого при установке соответствующей команды, к устройствам памяти. Обратите внимание, что сигналы команд и адреса приходят на интерфейс памяти за один тактовый цикл до сигнала выбора чипа. Обратите так же внимание, что если активны несколько рангов, они последовательно активируются (время между активациями в 1 тактовый цикл). **Примечание:** Биты CFGRx[RANK_ENABLE] соединены с соответствующими битами данного регистра по схеме «И» для получения набора рангов, которые будут инициализированы.

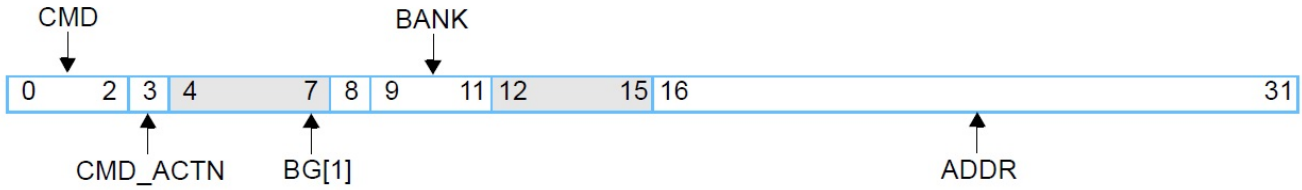


Рисунок 1.111 - INITCMDn (нечетные адреса)

Таблица 1.664 - Регистр команд инициализации

Бит	Название	Значение сброса	Описание
0:2	CMD	0s	Команда инициализации Это 3х-битовое выводится соответственно на сигнал строба адреса строки, строба адреса столбца и разрешения записи к устройству памяти. В частности если любой из этих битов установлен в 1, то соответствующий сигнал активен, если ноль, то не активен.
3	CMD_ACTN	0	Команда инициализации для ACT_N (Не используется в DDR3)
4:7		0	Зарезервировано
8	BG[1]	0	Адрес группы банка инициализации (Не используется в DDR3)
9:11	BANK	0s	Адрес банка инициализации. Это 3х-битовое поле выдается к устройству памяти и определяет адрес банка, в соответствии с выбором чипа.
12:15		0s	Зарезервировано
16:31	ADDR	0s	Адрес памяти инициализации Это 16ти-битовое поле определяет адрес в устройстве памяти, в соответствии с выбором чипа.

Набор регистров INITSEQn обеспечивает аппаратную поддержку механизма, позволяющего программно взаимодействовать непосредственно с устройством памяти DDR3 для различных шагов последовательности инициализации. Серия команд, необходимых для инициализации устройства памяти DDR, может быть получены путем программирования содержимого регистров INITSEQn/CMDn. Этот полностью программируемый механизм позволяет контроллеру согласовывать различные последовательности инициализации и любые будущие изменения.

При появлении сброса, память отключается вместе с удерживанием CKE в нуле. Перед инициализацией, на CKE должна быть установлена единица. Это обеспечивается очисткой бита SELF_REF_EN в регистре MCOPT2. Также последовательность инициализации должна активироваться путем установки бита INIT_START в регистре MCOPT2. Статус последовательности инициализации можно наблюдать при помощи состояний бита MCSTAT[INIT_COMPLETE].

Этот регистр обеспечивает доступность только команд от DDR3 к SDRAM. Другие команды не поддерживаются.

Примечание: Когда используется MRS инициализация для памяти DDR3 RDIMM, необходимо проводить MRS инициализацию для каждого отдельного ранга с промежутком в 3 тактовых цикла. (Устанавливается только 1 бит в INITSEQn.RANK для каждой последовательности).

Например, процедура инициализации DDR3 с CAS задержкой = 5, длинной пакета = 8, типом пакета = последовательным, выглядит так:

reg_write x0050, x8004000F // INITSEQ0 - доступно, задержка =tMRD, все ранги
reg_write x0051, x00200000 // INITCMD0 - MR2 (RAS=0,CAS=0,WE=0)
reg_write x0052, x8004000F // INITSEQ1 - доступно, задержка = tMRD, все ранги

reg_write x0053, x00300000 // INITCMD1 - MR3 (RAS=0,CAS=0,WE=0)
reg_write x0054, x8004000F // INITSEQ2 - доступно, задержка = tMRD, все ранги
reg_write x0055, x00100856 // INITCMD2 - MR1 (RAS=0,CAS=0,WE=0)
reg_write x0056, x800C000F // INITSEQ3 - доступно, задержка =t MOD, все ранги
reg_write x0057, x00001510 // INITCMD3 - MR0 (RAS=0,CAS=0,WE=0)
reg_write x0058, x8200000F // INITSEQ4 - доступно, задержка = tZQinit, все ранги
reg_write x0059, xC0000400 // INITCMD5 - ZQCL (RAS=1,CAS=1,WE=0,A10=1)
reg_write x005A, x00000000 // INITCMD5 – доступ отключен
reg_write x005B, x00000000 // INITCMD5

1.3.8.4.5.3.12 Регистр временных параметров SDRAM (0) – SDTR0

Адресное смещение: x‘80’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

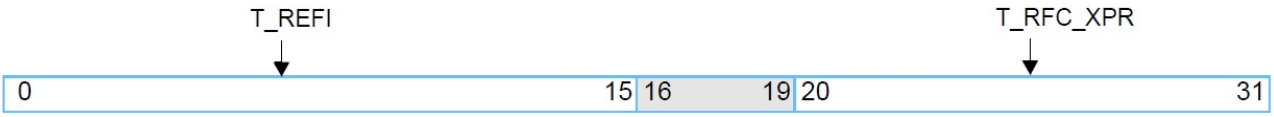


Рисунок 1.112 - SDTR0

Таблица 1.665 - Регистр временных параметров SDRAM (0)

Бит	Название	Значение сброса	Описание
0:15	T_REFI	0s	Интервал периодического обновления (В периодах тактового сигнала контроллера памяти) **Примечание:** Количество тактов между каждым запросом обновления на 1 такт больше, чем значение, заданное в этом поле (так при значении 0x185F, в среднем необходимо 6240 тактов между последовательностями обновления).
16:19		0s	Зарезервировано
20:31	T_RFC_XPR	0s	Команда обновления для ACT или REF (tRFC, min) (В периодах тактового сигнала контроллера памяти) Минимальное функциональное значение x002. Обратите внимание, что регистра временных задержек так же используется для установки значение tXPR, которое устанавливает задержку на выход сброса и на установку высокого уровня сигнал CKE к первой действительной команде. Это значение должно удовлетворять значению в спецификации JEDEC max(5nCK, tRFC(min) + 10 ns).

16-ти битное поле T_REFI из SDTR0 определяет частоту обновления для SDRAM. Внутренний счетчик работает на тактовой частоте контроллера. На частоте 800 MHz при значении 0x185F обеспечивается интервал обновления 7.8 µs(6240 x 1.25 ns = 7.8 µs). Этот регистр должен быть запрограммирован для обеспечения интервалов обновления устройства SDRAM на целевой рабочей частоте.

12-ти битное поле T_RFC_XPR определяет время (в тактах), которое необходимо выждать между предыдущей командой обновления и следующей (или командой активации, подаваемой на устройство памяти). Этот параметр сильно зависит от объема памяти (Mb или Gb).

Приостановка обновления включается установкой бита MCOPT1[DEF_REF]. Когда этот бит установлен, запросы обновления приостанавливаются до тех пор, пока очередь команд не станет пустой или пока число приостановленных команд обновления не станет равна 4. Когда одно из этих условий выполняется, контроллер выдает накопленные запросы обновления без разрывов между ними.

Используйте обычные операции обновления (не приостанавливаемые) если обычные операции памяти не могут задерживаться дольше, чем на запрограммированный период обновления.

1.3.8.4.5.3.13 Регистр временных параметров SDRAM (1) – SDTR1

Адресное смещение: x‘81’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Эти величины (Рисунок 1.113) влияют на временные параметры шины команд и данных. Активность T_LEADOFF заставляет все команды памяти удерживаться 2 такта, а выбор ранга (CS) происходит во время второго такта команд.

Обратите внимание, что значения, запрограммированные здесь, не могут быть использованы операциями памяти при использовании регистра INITSEQn/CMDn. Любые временные задержки последовательности инициализации должны быть заданы в поле INITSEQn[WAIT].

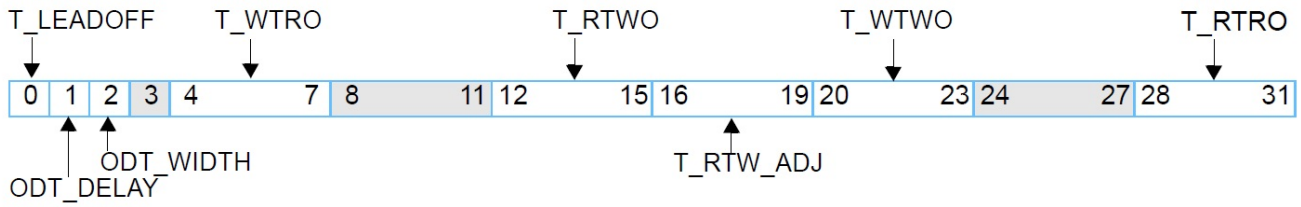


Рисунок 1.113 - SDTR1

Таблица 1.666 - Регистр временных параметров (1)

Бит	Название	Значение сброса	Описание
0	T_LEADOFF	0	Включение увеличенной длительности команд 0 – Команда в 1 такт 1 – Команда в 2 такта (2T timing)
1	ODT_DELAY	0	Задержка активации ODT (В тактовых циклах DDR) 0 - Активация сигнала ODT в том же такте что и сигнал выбора чипа 1 - Активация сигнала ODT через 1 такт после сигнал выбора чипа Обратите внимание, что команда чтения провидит к активации применяемого сигнала ODT на RL-WL тактов позже, чем активация для команды записи.
2	ODT_WIDTH	0	Управление длинной ODT 0 – Длинна сигнала ODT 4 или 6 тактов (BC4 или BL8) 1 - Длинна сигнала ODT 5 или 7 тактов (BC4 или BL8)
3		0	Зарезервировано
4:07	T_WTRO	0	Задержка переключения шины с транзакции записи на транзакцию чтение (разный CS) (в тактах DDR). 0000 - 0 тактов . . . 1010 - 10 тактов Остальные - зарезервированы Это поле определяет необходимое количество тактов между командой записи в один ранг и команды чтения из другого ранга. Полное количество тактов определяется как: WL - RL + BL/2 + T_WTRO

Таблица 1.667 - Регистр временных параметров (1)

Бит	Название	Значение сброса	Описание
8:11		0	Зарезервировано
12:15	T_RTWO	0	Задержка переключения шины с транзакции чтения на транзакцию запись (разные CS) (в тактах DDR). 0000 - 0 тактов - - - 1010 - 10 тактов Остальные – Зарезервированы Это поле определяет необходимое количество тактов между командой чтения из одного ранга и командой записи в другой ранг. Общее число тактов определяется как: RL - WL + BL/2 + T_RTWO
16:19	T_RTW_ADJ	0	Задержка переключения шина с транзакции чтения на транзакцию запись (тот же CS) (в тактах DDR) 0000 - 0 тактов - - - 1010 - 10 тактов
20:23	T_WTWO	0	Задержка переключения шины с транзакции записи на транзакцию запись (разные CS) (в тактах DDR) 0000 - 0 тактов - - - 1010 - 10 тактов Остальные - зарезервированы Это поле определяет необходимое количество тактов между командой записи в один ранг и командой записи в другой ранг. Полное число тактов определяется как: BL/2 + T_WTWO
24:27		0s	Зарезервировано
28:31	T_RTRO	0	Задержка переключения шины с транзакции чтения на транзакцию чтение (разные CS) (в тактах DDR) 0000 - 0 тактов - - - 1010 - 10 тактов Остальные - зарезервированы Это поле определяет необходимое количество тактов между командой чтения с одного ранга и командой чтений с другого ранга. Полное число тактов определяется как: BL/2 + T_RTRO

Примечание: Проверьте спецификацию на DDR3 PHY для дополнительных программируемых временных ограничений для задержек при обращении к одному рангу и разным рангам. В некоторых случаях, PHY требует дополнительные тактовые циклы для настройки линий временных задержек и других деталей.

Примечание: Если T_WTRO,T_RTWO, T_WTWO или T_RTRO меньше чем 3, ODT_WIDTH должен быть установлен в 0.

1.3.8.4.5.3.14 Регистр временных параметров SDRAM (2) – SDTR2

Адресное смещение: x‘82’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Этот регистр (Рисунок 1.114) должен программироваться для настройки конфигурации определенных временных параметров устройства SDRAM, которые считаются в единичных тактах, как того требует выбранное устройство.

Обратите внимание, что значения, программируемые здесь, не применяются для операций памяти, исполняемых в результате использования регистров INITSEQn/CMDn. Все задержки протокола последовательности инициализации должны быть определены в поле INITSEQn[WAIT].

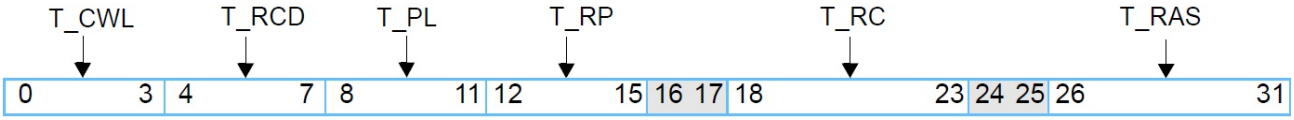


Рисунок 1.114 - SDTR2

Таблица 1.668 - Регистр временных параметров

Бит	Название	Значение сброса	Описание
0:3	T_CWL	0s	CAS write latency (Используется только в режиме DDR4)
4:7	T_RCD[1:4]	0	T_RCD[0:4] Время между активацией и ЧТ/ЗП (tRCD, min.) (В тактовых циклах DDR) 00010 - 2 такта . . . 1111 0- 31 такт Остальные - Зарезервированы
8:11	T_PL	0	Задержка проверки четности (В тактовых циклах DDR) Эти биты должны быть установлены в 0
12:15	T_RP[1:4]	0	T_RP[0:4] Период команды предварительной зарядки (tRP, min) (В тактовых циклах DDR) 00010 - 2 такта . . . 11111 – 31 такт Остальные - Зарезервированы
16	T_RP[0]	0	Старший бит of T_RP[0:4]. Смотрите описание выше
17	T_RCD[0]	0	MSB of T_RCD[0:4]. Смотрите описание выше
18:23	T_RC	0	Время между активацией и автоматической регенерацией(или активацией) (tRC, min) (В тактовых циклах DDR) 000010 - 2 такта . . . 111111 - 63 такта Остальные - Зарезервированы
24:25		0s	Зарезервировано
26:31	T_RAS	0	Время между активацией и предварительной зарядкой(tRAS, min) (В тактовых циклах DDR) 000010 - 2 такта . . . 110000 - 48 тактов Остальные - Зарезервированы

1.3.8.4.5.3.15 Регистр временных параметров SDRAM (3) – SDTR3

Адресное смещение: x‘83’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Этот регистр (Рисунок 1.115) должен программироваться для настройки конфигурации определенных временных параметров устройства SDRAM, которые считаются в единичных

тактах, как того требует выбранное устройство. Настройки, выходящие за пределы этой спецификации, могут привести к недействительным операциям.

Обратите внимание, что значения, программируемые здесь, не применяются для операций памяти, исполняемых в результате использования регистров INITSEQn/CMDn. Все задержки протокола последовательности инициализации должны быть определены в поле INITSEQn[WAIT].

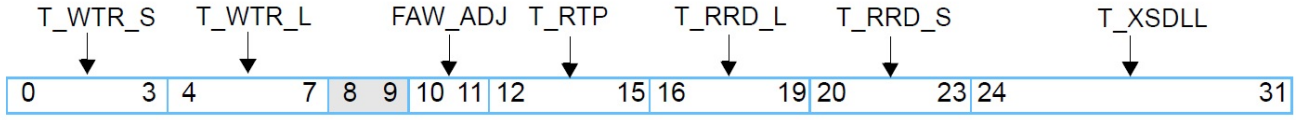


Рисунок 1.115 - SDTR3

Таблица 1.669 - Регистр временных параметров SDRAM (3)

Бит	Название	Значение сброса	Описание
0:3	T_WTR_S	0s	Задержка от начала транзакции записи до команды чтения для разных групп банков (не используется для DDR3)
4:7	T_WTR/T_WTR_L	0	Задержка от начала транзакции записи до команды чтения для разных групп банков (В тактовых циклах DDR) 0010 - 2 такта . . . 1111 - 15 тактов Остальные - зарезервированы
8:9		0s	Зарезервировано
10:11	FAW_ADJ	0s	Переменная настройки окна FAW(В тактовых циклах DDR)
12:15	T_RTP	0	Задержка между командой чтения и предварительной зарядкой (tRTP min) (В тактовых циклах DDR) 0010 - 2 такта . . . 1111 - 15 тактов Остальные – зарезервированы
16:19	T_RRD_L	0	Включение задержки между активациями в группе банков (тот же ранг, tRRD_L, min) (Не используется в DDR3)
20:23	T_RRD/T_RRD_S	0	Включение задержки между активациями в разных группах банка (тот же ранг, tRRD_S, min) (В тактовых циклах DDR) 0010 - 2 такта . . . 1111 - 15 тактов Остальные - зарезервированы Примечание: В системе DDR3, это параметр T_RRD
24:31	T_XSDLL	0	Задержка выхода из режима регенерации и блокировки DLL (tXSDLL). (В периодах тактового сигнала контроллера памяти) Этот счетчик задержки кратный 16 тактам и его типичное значение 00100000, что соответствует задержки в 512 тактов. Эта задержка необходима после активации CKE и перед любой командой срабатывания INITSEQ, применяемой к памяти.

Если наихудшая задержка записи DQ или DQS из PHY к устройству памяти (WDQ_DQS_DLY_PM) больше чем наихудшая задержка сигнала команды из PHY к устройству памяти (CMD_DLY_PM), параметры T_WTR, T_WTR_L и T_WTR_S заменятся на T_WTR*, как представлено в следующей формуле:
T_WTR*=round_up(tWTR_M/MCLK)+round_up([WDQ_DQS_DLY_PM-CMD_DLY_PM]/MCLK)

Если наихудшая задержка записи DQ или DQS из PHY к устройству памяти (WDQ_DQS_DLY_PM) не больше, чем наихудшая задержка сигнала команды из PHY к устройству памяти (CMD_DLY_PM), параметры T_WTR, T_WTR_L и T_WTR_S могут быть установлены в минимальное значение, необходимое для устройства памяти: T_WTR*=round_up(tWTR_M /MCLK)

tWTR_M - Задержка от начала внутренней транзакции записи до внутренней команды чтения устройства памяти

MCLK – Период тактового сигнала памяти

Последующая команда активации к другому банку может выдаваться, пока доступен первый банк, что приводит к сокращению количества перекрытий доступа к строке первого банка. Минимальное внутреннее время между последующей командой активации к другому банку определяется с помощью tRRD. Не больше 4х команд активации банка может быть выдано в течении определенного периоду tFAW (MIN), и так же применяется ограничение tRRD (MIN). Параметр tFAW (MIN) применяется, несмотря на количество открытых или закрытых банков.

Таблица 1.670 - Параметры

Обозначение		**DDR3-800**	**DDR3-1066**	**DDR3-1333**	**DDR3-1600**	**DDR3-1866**	**DDR3-2133**	**Единица**
nFAW (min)	**1 KB размер страницы**	**16**	**20**	**20**	**24**	**27**	**25**	**nCK**
	2 KB размер страницы	**20**	**27**	**30**	**32**	**35**	**35**	**nCK**
nRRD (min)	**1 KB размер страницы**	**4**	**4**	**4**	**5**	**5**	**6**	**nCK**
	2 KB размер страницы	**4**	**6**	**5**	**6**	**6**	**7**	**nCK**
FAWADJ	**1 KB размер страницы**	**0**	**1**	**1**	**1**	**2**	**0**	
	2 KB размер страницы	**1**	**1**	**2**	**2**	**3**	**2**	

1.3.8.4.5.3.16 Регистр временных параметров SDRAM (4) – SDTR4

Адресное смещение: x84

Значение сброса: x0000_0000

Доступ: ЧТ/ЗП

Этот регистр (Рисунок 1.116) должен программироваться для настройки конфигурации определенных временных параметров устройства SDRAM, которые считаются в единичных тактах, как того требует выбранное устройство. Определенные производственным стандартом конкретные значения величин описаны в таблице в столбце «Описание» (tCL = Cas задержка; tAL = Дополнительная задержка, и так далее).

Обратите внимание, что значения, программируемые здесь, не применяются для операций памяти, исполняемых в результате использования регистров INITSEQn/CMDn. Все задержки протокола последовательности инициализации должны быть определены в поле INITSEQn[WAIT].

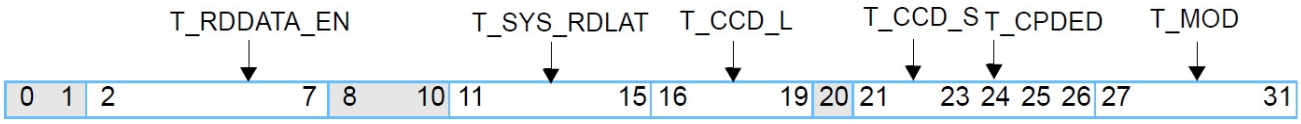


Рисунок 1.116 - SDTR4

Таблица 1.671 - Регистр временных параметров SDRAM (4)

Бит	Название	Значение сброса	Описание
0		0	Зарезервировано
1	T_RDDATA_EN[0]	0	MSB в T_RDDATA_EN[0:6]. Этот бить должен быть установлен в 0
2:7	T_RDDATA_EN[1:6]	000000	T_RDDATA_EN[0:6] Задержка включения транзакции данных чтения PHY (В тактовых циклах DDR) 000000 - 0 тактов . . . 111111 - 63 такта Другие - Зарезервированы Определенное число тактов DFI от утверждения команды чтения на шине команд DFI, до утверждения сигнала активации чтения данных. Эта величина зависит от PHY.
8:9		0s	Зарезервировано
10:15	T_SYS_RDLAT	000000	Задержка транзакции чтения системы (В тактовых циклах DDR) 000000 - 0 тактов - - - 111111 - 63 тактов Указывает задержку чтения системы, необходимую для задержки транзакции чтения памяти (RL = CL + AL) и показывает, когда данные транзакции чтения становятся действительными и доступными на шине. Она включает в себя любые задержки, связанные с исходящими от PHY сигналами команд, задержки захвата входящих данных транзакции чтения и любые дополнительные задержки каналов (в обоих направлениях), которые добавляются к PD на чип и временным задержкам.
16:19	T_CCD_L	000s	Задержка между командами CAS одной и той же группы банка (tCCD_L, min) (Не используется)
20		0	Зарезервированы
21:23	T_CCD	0	Задержка между командами CAS в разные группы банка (tCCD_s, min.) (В тактовых циклах DDR). 010 - 2 тактов . . . 111 - 7 тактов Остальные - Зарезервированы

Продолжение таблицы 1.672

Бит	Название	Значение сброса	Описание
24:26	T_CPDED	000	Задержка отключения прохождения команд (В тактовых циклах DDR) Когда активен возврат в высокоимпедансное состояние, (MCOPT2[23]=1), указывает количество тактовых циклов DDR, от входа в режим регенерации, до тех пор, пока высокоимпедансное состояние не применено к этим сигналам при помощи регистра включения перехода в высокоимпедансное состояние (RHZEN) 000 -1 такт (минимальное значение для операций DDR3) 001 -2 такта 010 -3 такта 011 -4 такта (минимальное значение для операций) . . . 111 - 8 тактов
27:31	T_MOD	0	Задержка обновления команды установки регистров режима (tMOD, min.) (В тактовых циклах DDR). 00001 - 1 такт . . . 11111 - 31 такт Остальные - Зарезервированы

1.3.8.4.5.3.17 Регистр временных параметров SDRAM (5) – SDTR5

Адресное смещение: x‘85’

Значение сброса: x‘0000_0000’

Доступ: ЧТ/ЗП

Этот регистр (Рисунок 1.117 - SDTR5) должен быть запрограммирован в соответствии с требованиями конкретного PHY, который подключен к контроллеру памяти, и в соответствии со спецификацией DFI-3.1.

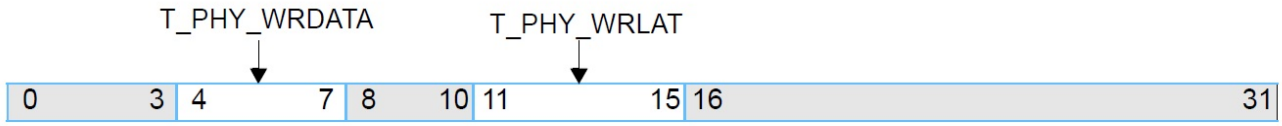


Рисунок 1.117 - SDTR5

Таблица 1.673 - Регистр временных параметров SDRAM (5)

Бит	Название	Значение сброса	Описание
0:04		0s	Зарезервировано
5:07	T_PHY_WRDATA	0	Задержка транзакции данных чтения PHY (Смотрите спецификацию на DFI-3) (в периодах тактового сигнала контроллера памяти) 000 - 0 тактов - - - 111 - 7 тактов Устанавливает число тактовых циклов DFI от детектирования сигнала разрешения записи до утверждения данных записи на шине чтения.
8:31		0s	Зарезервировано

Примечание по временному протоколу DDR:

Следующие уравнения используются в контроллере памяти DDR34LMC для определения количества тактов между соответствующими операциями на интерфейсе памяти. Они используют значения, запрограммированные в регистрах SDTR0-4, а также значения в регистрах SMR0-6.

Активация чтения или записи = T_RCD - AL

Между транзакциями чтениями (внутри ранга) = T_CCD

Между транзакциями записями (внутри ранга) = T_CCD

От транзакции чтения к транзакции записи (внутри ранга, BL8/OTF) = CL - CWL + T_CCD(T_CCD_S) + 2 + T_RTW_ADJ

От транзакции чтения к транзакции записи (внутри ранга, BC4) = CL - CWL + T_CCD/2(T_CCD_S/2) + 2 + T_RTW_ADJ

От транзакции записи к транзакции чтения (внутри ранга, BL8/OTF) = CWL + AL + 4+ T_WTR

Транзакция записи после транзакции чтения (внутри ранга, BC4) = CWL + AL + 2+ T_WTR

Между транзакциями чтениями (между рангами) = BL/2 + T_RTRO

Между транзакциями записями (между рангами) = BL/2 + T_WTWO

От транзакции чтения к транзакции записи (между рангами) = CL - CWL + BL/2 + T_RTWO

От транзакции записи к транзакции чтению (между рангами) = CWL - CL + BL/2 + T_WTRO

От транзакции чтения к предварительной зарядке (внутри ранга) = AL + T_RTP

От транзакции записи к предварительной зарядке (внутри ранга, BL8/OTF) = AL + CWL + 4+ WR

От транзакции записи к предварительной зарядке (внутри ранга, BC4) = AL + CWL + 2+ WR

От транзакции чтения с автоматической предварительной зарядкой к активации = AL + T_RTP + T_RP

От транзакции записи с автоматической предварительной зарядкой к активации (BL8/OTF) = AL + CWL + 4+ WR + T_RP

От транзакции запись с автоматической предварительной зарядкой к активации (BC4) = AL + CWL + 2+ WR + T_RP

В вышеупомянутом списке - BL, это размер пакета, представленный для первой команды памяти.

1.3.8.4.5.3.18 Регистр отладки (0) – DBG0

Адресное смещение: x‘8A’
Значение сброса: x‘0000_0000’
Доступ: ЧТ
Этот регистр используется для отладки.

Таблица 1.674 - Регистр отладки 0

Бит	Название	Значение сброса	Описание
0:10		0s	Зарезервировано
11:15	T_SYS_RDLAT_DBG	0	(В тактовых циклах DDR) Это поле показывает количество тактов между первой командой транзакции чтения и инициализацией после сброса. Если это значение не совпадает с в SDTR4[T_SYS_RDLAT], это свидетельствует об ошибке.
16:31		0s	Зарезервировано

1.3.8.4.5.3.19 Регистр теневого режима 0 – SMR0

Адресное смещение: x‘90’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Это регистр (Рисунок 1.118 - SMR0) должен быть запрограммирован в соответствии со значениями регистра режима SDRAM 0 (MR0). Это значение должно отражать режим «эксплуатации» выбранного устройства памяти. Некоторые памяти могут определять эти значения по-разному. Смотрите спецификацию на выбранную память для окончательного выбора значений.

Значения в этом регистре не программируются внутри используемого устройства памяти, и поэтому регистр называются «Теневой». Регистр режима внутри устройства памяти должен быть запрограммирован при помощи регистров и процедур INITSEQn и CMDn.

Некоторые поля битов отмечены как «не используются контроллером памяти». Эти поля отражают рабочее состояние регистра в устройстве SDRAM и не влияют на работу контроллера памяти. Все биты обозначенные как RFU должны быть запрограммированы в значение 0.

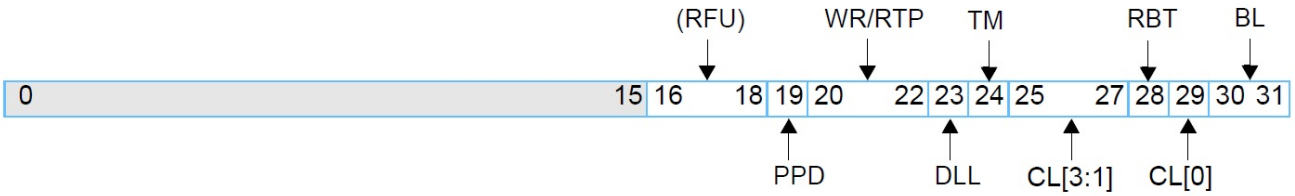


Рисунок 1.118 - SMR0

Таблица 1.675 - Регистр теневого режима (0)

Бит	Название	Значение сброса	Описание
0:15		0s	Зарезервировано
16:18	(RFU)	0	Зарезервировано
19	PPD	0	PPD – Отключение предварительной зарядки (не используется контроллером памяти) 0 – Медленный выход (DLL выключено) 1 – Быстрый выход (DLL включено)
20:22	WR/RTP	0	Восстановление чтения. (Используется контроллером памяти). 000 - 16 тактов 001 - 5 тактов 010 - 6 тактов 011 - 7 тактов 100 - 8 тактов 101 - 10 тактов 110 - 12 тактов 111 - 14 тактов
23	DLL	0	Сброс DLL (не используется контроллером памяти)
24	TM	0	TM (не используется контроллером памяти)
25:27	CL[3:1]	0	Задержка CAS (Используется контроллером памяти) 001.0 - 5 тактов 010.0 - 6 тактов 011.0 - 7 тактов 100.0 - 8 тактов 101.0 - 9 тактов 110.0 - 10 тактов 111.0 - 11 тактов 000.1 - 12 тактов 001.1 - 13 тактов 010.1 - 14 тактов 011.1 - 15 тактов 100.1 - 16 тактов Остальные - зарезервированы
28	RBT	0	Тип пакета транзакции чтения 0 - Последовательный 1 - Зарезервированный (Режим чередования не поддерживается данным контроллером памяти)
29	CL[0]	0	Младший бит задержка CAS (Смотрите выше)

Продолжение таблицы 1.676

Бит	Название	Значение сброса	Описание
30:31	BL	0	Размер Пакета (Используется контроллером памяти) 00 - BL8 (Фиксированный) 01 - BC4/BL8 On-The-Fly (Оптимизирует производительность, позволяя МС переключаться между 4 и 8-кадровой передачей без необходимости выдачи команды MRS. Действителен только для транзакций данных SDRAM в режиме 1\2 и 1\4) 10 - BC4(Фиксированный) (Пожалуйста, обратите внимание, что для режима с фиксированным BC4, PHY должен поддерживать калибровку фиксированного режима BC4 и транзакции от контроллера памяти. Действителен только для транзакций данных SDRAM в режиме 1\2 и 1\4)

1.3.8.4.5.3.20 Регистр теневого режима SDRAM 1 – SMR1

Адресное смещение: x‘91’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Это регистр (Рисунок 1.119) должен быть запрограммирован в соответствии со значениями регистра режима SDRAM 1. Это значение должно отражать режим эксплуатации выбранного устройства памяти. Некоторые памяти могут определять эти значения по-разному. Смотрите спецификацию на выбранную память для окончательного выбора значений.

Значения в этом регистре не программируются внутри используемого устройства памяти, и поэтому регистр называются «теневым». Регистры режима внутри устройства памяти должны программироваться, используя регистры последовательности INITSEQn/CMDn.

Некоторые поля битов отмечены как «не используются контроллером памяти». Эти поля отражают рабочее состояние регистра в устройстве SDRAM и не влияют на работу контроллера памяти. Все биты обозначенные как RFU должны быть запрограммированы в значение 0.

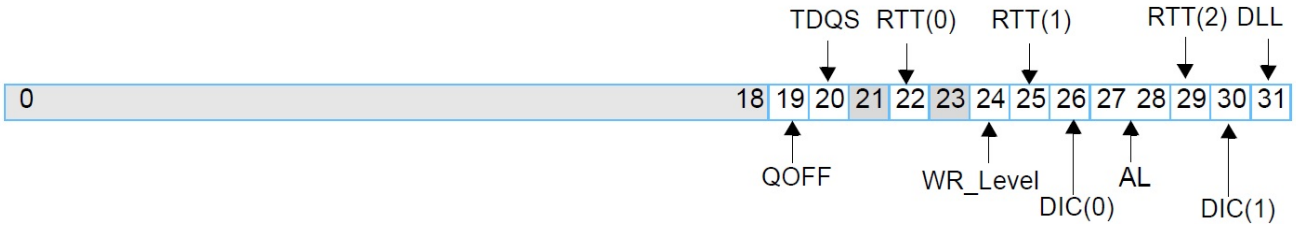


Рисунок 1.119 - SMR1

Таблица 1.677 - Регистр теневого режима SDRAM 1

Бит	Название	Значение сброса	Описание
0:18		0s	Зарезервировано
19	QOFF	0	Q-отключение (не используется контроллером памяти) 0 – Включение выходного буфера 1 – Выключение выходного буфера
20	TDQS	0	Управление TDQS (не используется контроллером памяти) 0 - Отключено 1 - Включено
21	(RFU)	0	Зарезервировано
22	RTT_Nom(2)	0	RTT_Nom бит 2(старший бит), связан с 25-м и 29-м битами.
23	(RFU)	0	Зарезервировано

Продолжение таблицы 1.678

Бит	Название	Значение сброса	Описание
24	WR_Level	0	Управление выравниванием записи (не используется контроллером памяти) 0 - Отключено 1 - Включено
25	RTT_Nom(1)	0	RTT_Nom бит 1, Связан с 22-м и 29-м битами.
26	DIC(1)	0	Управление импедансом. Связан с 30-м битом.
27:28	AL	0	Дополнительная задержка в тактах (используется контроллером памяти) 00 - 0 (Отключено) 01 - CL-1 10 - CL-2 11 - Зарезервировано
29	RTT_Nom(0)	0	Управления терминацией сигналов устройства памяти (не используется контроллером памяти) RTT_Nom(0:2) 000 - RTT_Nom отключен 001 - RZQ/4 Ом 010 - RZQ/2 Ом 011 - RZQ/6 Ом 100 - RZQ/12 Ом 101 - RZQ/8 Ом Для задания этих настроек используются биты SMR1[22], SMR1[25] и SMR1[29]
30	DIC(1)	0	Управление импедансом (не используется контроллером памяти) DIC(0:1) 00 - RZQ/6 01 - RZQ/7 10 - RZQ/TBD 11 - RZQ/TBD Для задания этих настроек используются биты SMR1[30] и SMR1[26]
31	DLL	0	Включение DLL (не используется контроллером памяти) 0 - Отключен 1 - Включен

1.3.8.4.5.3.21 Регистр теневого режима 2 – SMR2

Адресное смещение: x‘92’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Это регистр (Рисунок 1.120) должен быть запрограммирован в соответствии со значениями регистра режима SDRAM 2. Это значение должно отражать режим эксплуатации выбранного устройства памяти. Некоторые памяти могут определять эти значения по-разному. Смотрите спецификацию на выбранную память для окончательного выбора значений.

Значения в этом регистре не программируются внутри используемого устройства памяти, и поэтому регистр называются «теневым». Регистры режима внутри устройства памяти должны программироваться, используя регистры последовательности INITSEQn/CMDn.

Некоторые поля битов отмечены как «не используются контроллером памяти». Эти поля отражают рабочее состояние регистра в устройстве SDRAM и не влияют на работу контроллера памяти. Все биты обозначенные как RFU должны быть запрограммированы в значение 0.

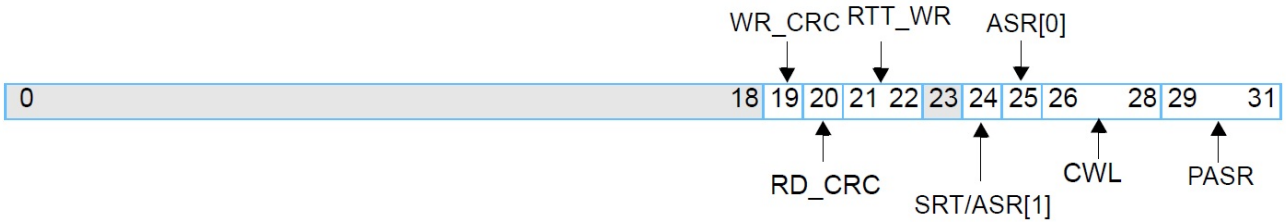


Рисунок 1.120 - SMR2

Таблица 1.679 - Регистр теневого режима SDRAM 2

Бит	Название	Значение сброса	Описание
0:18		0s	Зарезервировано
19	WR_CRC	0	CRC Запись (не используется в DDR3)
20	RD_CRC	0	CRC Чтение (не используется в DDR3)
21:22	RTT_WR	0	Rtt для операций записи (не используется контроллером памяти) 00 – Отключение динамического ODT (запись не влияет на значение Rtt) 01 - RZQ/4 10 - RZQ/2 11 - Зарезервировано
23	RFU	0	Зарезервировано
24	SRT/ASR[1]	0	(не используется контроллером памяти) Диапазон температур регенерации 0 - Обычный 1 - Расширенный
25	ASR[0]	0	Автоматическая регенерация (не используется контроллером памяти) ASR[0] 0 - Ручной 1 - Включен (опционально)
26:28	CWL	0	Задержка записи 000 - 5 тактов 001 - 6 тактов 010 - 7 тактов 011 - 8 тактов 100 - 9 тактов 101 - 10 тактов 110 - 11 тактов 111 - 12 тактов
29:31	PASR	0	Частичная регенерация массива (опционально) (не используется контроллером памяти) Для подробного описания смотреть спецификацию JEDEC-79.

1.3.8.4.5.3.22 Регистр теневого режима 3 – SMR3

Адресное смещение: x‘93’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Это регистр (Рисунок 1.121) должен быть запрограммирован в соответствии со значениями регистра режима SDRAM 3. Это значение должно отражать режим эксплуатации выбранного устройства памяти. Некоторые памяти могут определять эти значения по-разному. Смотрите спецификацию на выбранную память для окончательного выбора значений.

Значения в этом регистре не программируются внутри используемого устройства памяти, и поэтому регистр называются «теневым». Регистры режима внутри устройства памяти должны программироваться, используя регистры последовательности INITSEQn/CMDn.

Некоторые поля битов отмечены как «не используются контроллером памяти». Эти поля отражают рабочее состояние регистра в устройстве SDRAM и не влияют на работу контроллера памяти. Все биты обозначенные как RFU должны быть запрограммированы в значение 0.

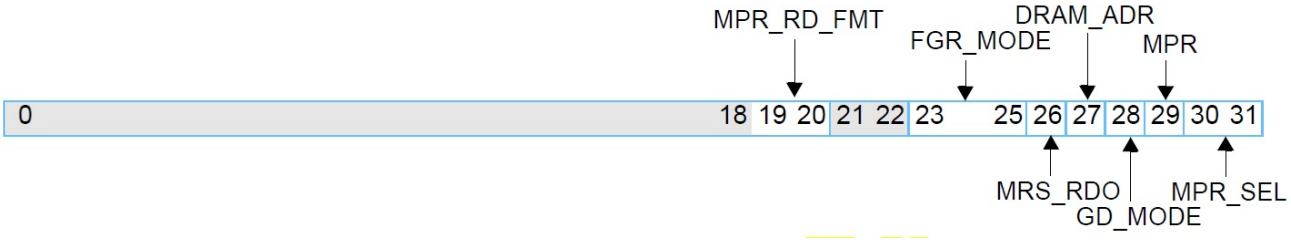


Рисунок 1.121 - SMR3

Рисунок 1.122 - Регистр теневого режима SDRAM 3

Бит	Название	Значение сброса	Описание
0:18		0s	Зарезервировано
19:20	MPR_RD_FMT	0s	RFU
21:22	RFU	0s	Зарезервировано
23:25	FGR_MODE	0	RFU
26	MRS_RDO	0	Считывание MRS (Режим DDR4) (не используется контроллером памяти) 0 - Отключено 1 – Включено
27	DRAM_ADR	0	RFU
28	GD_MODE	0	RFU
29	MPR	0	Операции MPR (не используется контроллером памяти) 0 – Обычные операции 1 – Поток данных из MPR
30:31	MPR_SEL	0	RFU

1.3.8.4.5.3.23 Регистр теневого режима 4 – SMR4

Адресное смещение: x‘94’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Не используется для DDR3.

1.3.8.4.5.3.24 Регистр теневого режима 5 – SMR5

Адресное смещение: x‘95’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Не используется для DDR3.

1.3.8.4.5.3.25 Регистр теневого режима 6 – SMR6

Адресное смещение: x‘96’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Не используется для DDR3.

1.3.8.4.5.3.26 Регистры управления встроенной терминацией памяти – ODTRn

Адресное смещение: x‘A0’ - x‘A3’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Эти 4 регистра (Рисунок 1.123) управляют операциями на контактах встроенной терминации (для всех рангов памяти). Они позволяют управлять динамикой терминации для повышения эффективности переключения и уменьшения потребления в режиме ожидания.

Каждый из этих 4х регистров управляет одним из 4х ODT выходов контроллера памяти. Каждый ODT контакт может быть запрограммирован на активность во время транзакций чтения или записи (пока активна шина данных) в любой ранг памяти. Обратитесь к документации на память DDR3 для ODT операций для 1, 2 ,3 или 4-го ранга системы памяти.

ODTR0 с адресом смещения xA0 управляет ODT сигналом для ранга 0.
ODTR1 с адресом смещения xA1 управляет ODT сигналом для ранга 1.
ODTR2 с адресом смещения xA2 управляет ODT сигналом для ранга 2.
ODTR3 с адресом смещения xA3 управляет ODT сигналом для ранга 3.

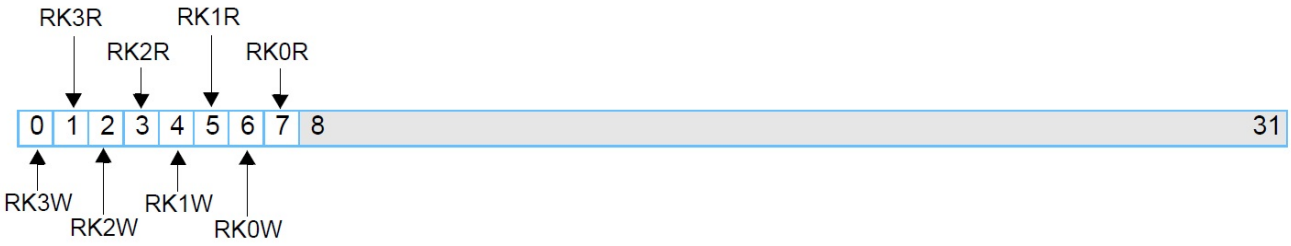


Рисунок 1.123 - ODTRn

Таблица 1.680 - Регистр управления встроенной терминацией памяти для ранга n

Бит	Название	Значение сброса	Описание
0	RK3W	0	1 = Включение ODT(n) с записью в ранг 3
1	RK3R	0	1 = Включение ODT(n) с чтением из ранга 3*
2	RK2W	0	1 = Включение ODT(n) с записью в ранг 2
3	RK2R	0	1 = Включение ODT(n) с чтением из ранга 2*
4	RK1W	0	1 = Включение ODT(n) с записью в ранг 1
5	RK1R	0	1 = Включение ODT(n) с чтением из ранга 1*
6	RK0W	0	1 = Включение ODT(n) с записью в ранг 0
7	RK0R	0	1 = Включение ODT(n) с чтением из ранга 0*
8:31		0s	Зарезервировано
* - Смотрите примечание по программированию после таблицы			

Устройство памяти DDR3 дополнительно поддерживает встроенную терминацию (ODT). ODT устраняет необходимость во внешних сигналах терминации от сети. ODT в памяти DDR3 управляется настройками регистров SMR1 и SMR2, а так же состояниями на ODT входах устройства памяти.

Таблица 1.681 - Параметры ODT

Функциональные биты и биты управления в регистре	0	1	2	3	4	5	6	7
Отключенный ODT	0	0	0	0	0	0	0	0
Утверждение ODT, когда происходит чтение из ранга 0	-	-	-	-	-	-	-	1
Утверждение ODT, когда происходит запись в ранг 0	-	-	-	-	-	-	1	-
Утверждение ODT, когда происходит чтение из ранга 1	-	-	-	-	-	1	-	-
Утверждение ODT, когда происходит запись в ранг 1	-	-	-	-	1	-	-	-
Утверждение ODT, когда происходит чтение из ранга 2	-	-	-	1	-	-	-	-
Утверждение ODT, когда происходит запись в ранг 2	-	-	1	-	-	-	-	-
Утверждение ODT, когда происходит чтение из ранга 3	-	1	-	-	-	-	-	-
Утверждение ODT, когда происходит запись в ранг 3	1	-	-	-	-	-	-	-
Примечание: Значение “-” показывает, что соответствующий бит в регистре может быть 0 или 1.								

Примечание по программированию: Для устройства памяти неприемлема установка бита RKnR в регистре ODTRn для нескольких “n”. Следовательно, неприемлема одновременная установка битов ODTR0[RK0R], ODTR1[RK1R], ODTR2[RK2R], и ODTR3[RK3R] в 1.

1.3.8.4.5.3.27 Регистр статуса и управления механизмом очистки – SCRUB_CNTL

Адресное смещение: x‘AA’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данный регистр не используется. Все его биты зарезервированы.

1.3.8.4.5.3.28 Регистр таймера очистки в интервалах регенерации – SCRUB_INT

Адресное смещение: x‘AB’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данный регистр не используется. Все его биты зарезервированы.

1.3.8.4.5.3.29 Регистр текущего адреса исполнения механизма очистки – SCRUB_CUR

Адресное смещение: x‘AC’
Значение сброса: x‘0000_0000
Доступ: ЧТ
Данный регистр не используется. Все его биты зарезервированы.

1.3.8.4.5.3.30 Регистр расширенного текущего адреса исполнения механизма очистки – SCRUB_CUR_EXT

Адресное смещение: x‘AD’
Значение сброса: x‘0000_0000’
Доступ: ЧТ
Данный регистр не используется. Все его биты зарезервированы.

1.3.8.4.5.3.31 Регистры данных для предварительного заполнения при механизма очистки – SCRUB_PREFILL_DATA0-1

Адресное смещение: x‘AE’ - x‘AF’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данный регистр не используется. Все его биты зарезервированы.

1.3.8.4.5.3.32 Регистр адреса начала механизма очистки – SCRUB_ST_RANKm_PORTx

Адресное смещение: x‘B0~xBF’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данный регистр не используется. Все его биты зарезервированы.

1.3.8.4.5.3.33 Регистр статуса калибровки – CALSTAT

Адресное смещение: x‘C1’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Этот регистр (Рисунок 1.124) задает статус калибровочной логики DDR3.

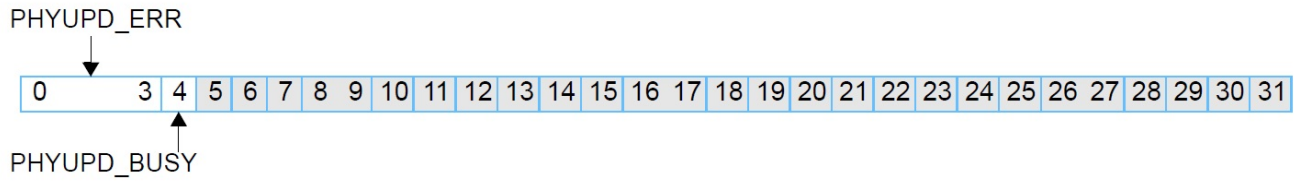


Рисунок 1.124 - CALSTAT

Таблица 1.682 - Регистр статуса калибровки

Бит	Название	Значение сброса	Описание
0:03	PHYUPD_ERR	0	Ошибка обновления PHY. Ошибка тайм-аута происходит во время одного из четырех типов запросов обновления PHY. xxx1 - Тайм-аут обновления PHY типа 3 xx1x - Тайм-аут обновления PHY типа 2 x1xx - Тайм-аут обновления PHY типа 1 1xxx - Тайм-аут обновления PHY типа 0
4	PHYUPD_BUSY	0	Калибровка обновления PHY в процессе. Этот бит доступен только для чтения.
5:31		0s	Зарезервировано

1.3.8.4.5.3.34 Регистр таймера обновления DFI PHY 0-3 – T_PHYUPD0-3

Адресное смещение: x‘CC’ - x‘CF’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Эти 4 регистра (Рисунок 1.125) содержат значение максимального числа тактов, которое DDR3 MC ждет деактивации запроса обновления от PHY после активации сигнала подтверждения обновления от PHY. Выбор регистра, значение которого будет определять тайм-аут, зависит от значения на шине типа обновления PHY в момент запроса.

Например, регистр T_PHYUPD0 содержит значение тайм-аута для запроса 0-го типа.

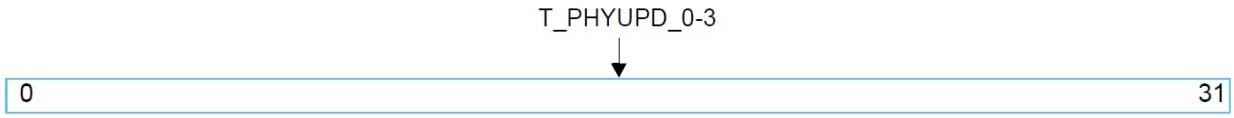


Рисунок 1.125 - T_PHYUPD0-3

Таблица 1.683 - Регистр таймера обновления DFI PHY

Бит	Название	Значение сброса	Описание
0:31	T_PHYUPD_0-3	0s	(В тактовых циклах контроллера памяти) Максимально число тактов ожидания отключения запроса обновления из PHY после активации подтверждения обновления от PHY.

1.3.8.4.5.3.35 Регистры адреса окончания очистки – SCRUB_ED_RANKm_PORTx

Адресное смещение: x‘D0’~x‘DF’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП

Данные регистры не используется. Все его биты зарезервированы.

1.3.8.4.5.3.36 Регистр адреса ECC-ошибки – ECCERR_ADDR_PORTx

Адресное смещение: x‘E0’~x‘E3’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данные регистры не используется. Все его биты зарезервированы.

1.3.8.4.5.3.37 Счетчик ошибок ECC – ECCERR_COUNT_PORTx

Адресное смещение: x‘E4’ - x‘E7’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данные регистры не используется. Все его биты зарезервированы.

1.3.8.4.5.3.38 Регистр очистки значения счетчика исправляемых ошибок ECC – CE_TCNT_CLR

Адресное смещение: x‘E8’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данные регистры не используется. Все его биты зарезервированы.

1.3.8.4.5.3.39 Регистр установки значения счетчика исправляемых ошибок ECC – CE_TCNT_SET

Адресное смещение: x‘E9’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данные регистры не используется. Все его биты зарезервированы.

1.3.8.4.5.3.40 Расширенный адрес ошибки ECC – ECCERR_ADDR_EXT

Адресное смещение: x‘EA’
Значение сброса: x‘0000_0000’
Доступ: ЗП/ЧТ
Данные регистры не используется. Все его биты зарезервированы.

1.3.8.4.5.3.41 Регистр статуса ошибки ECC – ECCERR_PORTx

Адресное смещение: x‘F0’ ~ x‘F7’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП;
Данные регистры не используется. Все его биты зарезервированы.

1.3.8.4.5.3.42 Байты проверки ошибки ECC – ECC_CHECK_PORTx

Адресное смещение: x‘F8’ ~ x‘FB’
Значение сброса: x‘0000_0000’
Доступ: ЧТ/ЗП
Данные регистры не используется. Все его биты зарезервированы.

1.3.8.4.5.4 Сигналы сброса контроллера внешней динамической памяти DDR34LMC

У ядра DDR34LMC существует три различных сценария сброса. Это системный сброс, частичный сброс и внешний сброс самообновления. Они активируются входными сигналами I_MC_RESET, I_MC_RECOVER_RESET и I_MC_REF_RESET соответственно.

1.3.8.4.5.4.1 Системный сигнал сброса

Установка сигнала I_MC_RESET в 1 и I_MC_RECOVER_RESET с I_MC_REF_RESET в 0, сбрасывает всю логику ядра, машину состояний и регистры в их начальное состояние, а так же устанавливает сигнал сброса DDR (O_DFI_RESET_N[3:0]), сигнал разрешения сброса (O_RESET_OE), сигнал включения тактового сигнала (O_DFI_CKE[3:0]) и сигнал разрешения включения тактового сигнала (O_CKE_OE) для сброса интерфейса DDR. (MCOPT2[RESET_RANK(3:0),CKE_OE,RESET_OE] устанавливается в 000011, и {O_RESET_OE,CKE_OE, O_DFI_RESET_N[3:0], O_DFI_CKE[3:0]} устанавливаются в 1100000000). (Рисунок 1.126)

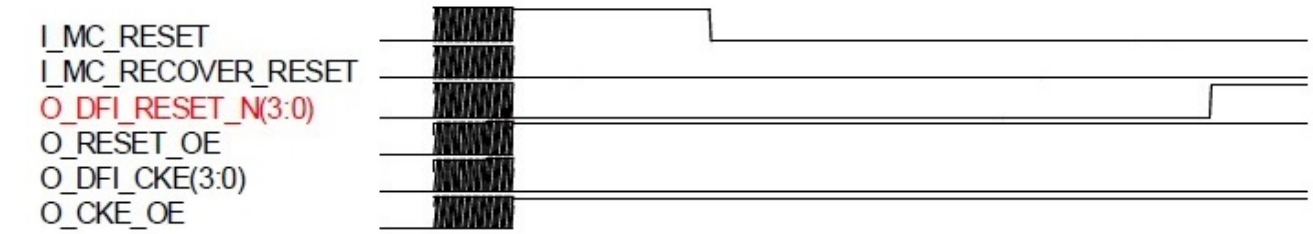


Рисунок 1.126 – Системный сигнал сброса

Примечание: Минимальное рекомендуемое время удерживания сигнала для сброса все логики составляет 200 тактовых циклов.

1.3.8.4.5.4.2 Сигнал сброса без обнуления конфигурационных регистров

Установка входного сигнала ‘I_MC_RECOVER_RESET’ в ‘1’ и установление ‘I_MC_RESET’ с ‘I_MC_REF_RESET’ в ’0’, позволяет сбрасывать контроллер динамической памяти MC DDR3 без обнуления его конфигурационных регистров.

Список регистров, на которые не действует данный сброс, представлен в таблице (см. Таблица 1.684).

Выходы контроллера ‘O_DFI_RESET_N[3:0]’, ‘O_DFI_CKE[3:0]’, ‘O_CKE_OE’, ‘O_RESET_OE’ не меняют своего значения при сбросе сигналом ‘I_MC_RECOVER_RESET’.

Для того, чтобы обнулить состояние регистров контроллера MC DDR3 сигналом ‘I_MC_RECOVER_RESET’, необходимо выполнить следующие условия:

- Контроллер MC DDR3 и блок физического интерфейса DDR3 PHY, а так же внешняя память SDRAM должны быть успешно обнулены, проинициализированы и настроены для последующей работы;
- Все ожидающие транзакции должны быть завершены;
- Внешняя память SDRAM должна быть в режиме регенерации (self-refresh).

Таблица 1.684 - Игнорируемые регистры

Адрес	Имя регистра
x‘0020’	MCOPT1
x‘0022’	MCOPTE1
x‘0023’	MCOPTE2
x‘0024’	MCOPTE3
x‘0030’	IOCNTL
x‘0031’	PHYCNTL
x‘0040’	CFGR0
x‘0041’	CFGR1
x‘0042’	CFGR2
x‘0043’	CFGR3
x‘0048’	RHZEN
x‘0080’	SDTR0
x‘0081’	SDTR1
x‘0082’	SDTR2
x‘0083’	SDTR3
x‘0084’	SDTR4
x‘0085’	SDTR5
x‘0090’	SMR0
x‘0091’	SMR1
x‘0092’	SMR2
x‘00a0’	ODTR0
x‘00a1’	ODTR1
x‘00a2’	ODTR2
x‘00a3’	ODTR3

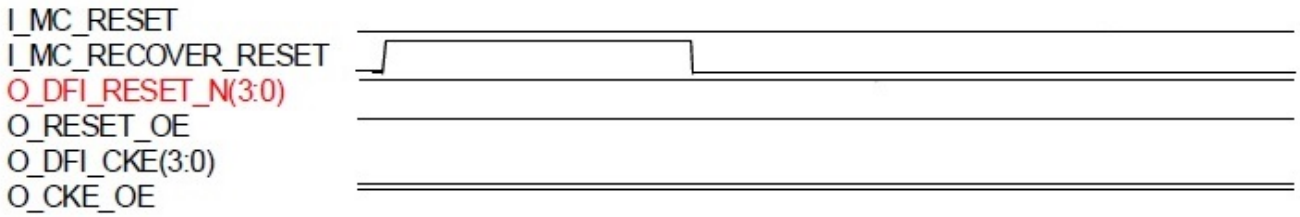


Рисунок 1.127 – Сигнал сброса без обнуления конфигурационных регистров

1.3.8.4.5.4.3 Сигнал сброса с сохранением данных во внешней памяти типа SDRAM

Сигнал сброса ‘I_MC_REF_RESET’ может применяться для сохранения данных во внешней памяти типа SDRAM при реализации сценария по управлению питанием микросхемы. Возможность подавать или снимать питание с контроллера динамической памяти MC DDR3 во время его неактивного состояния позволяет понижать потребляемую мощность всей системы.

После выполнения заданной последовательности подачи напряжения и сброса системы: активация сигналов сброса (‘I_MC_REF_RESET’ и ‘I_MC_RESET’ в ‘1’) с последующей деактивацией (‘I_MC_REF_RESET’ в ‘0’, на такт раньше, чем ‘I_MC_REF_RESET’) –

происходит возврат всей логики контроллера, ее регистров и автомата состояний в их начальное состояние. Для реализации регенерации (self refresh) внешней памяти предусмотрено управление выходными сигналами ({O_RESET_OE, O_CKE_OE} в ‘00’), что позволяет удерживать источник сигналов RESET_N[3:0] и CKE[3:0] внешней памяти типа SDRAM в высокоимпедансном состоянии. Внешний контроль сигналов RESET_N[3:0], CKE[3:0] и удержание внешней памяти типа SDRAM в состоянии регенерации (RESET_N[3:0] в ‘1111’ и CKE[3:0] ‘0000’) позволяет подготовить микросхему к отключению питания.

На рисунке (Рисунок 1.128) показана временная диаграмма данной последовательности.

Описание сигналов:

DVDD_PGOOD: Установка в 1 указывает на стабильность питания SDRAM

DDR_RESET_N[3:0]: Сигнал сброса ко всем рангам памяти

DDR_CKE[3:0]: Сигнал CKE ко всем рангам памяти

ASIC_VDD_PGOOD: Установка в 1 указывает на стабильное питание чипа

ASIC_DVDD_PGOOD: Установка в 1 указывает на стабильное питание портов

RESET: Сигнал обычного сброса, который генерируется логикой сброса чипа. Он присоединен к входу I_MC_RESET.

REF_RESET: Сигнал сброса восстановления, который генерируется логикой сброса чипа. Он присоединен к I_MC_REF_RESET.

DDR_MC_CLOCK: Генерируется PLL и присоединен к I_MC_CLOCK.

O_DDR_RESET[3:0]: Выходной сигнал сброса из чипа

O_DDR_CKE[3:0]: Выходной сигнал CKE из чипа

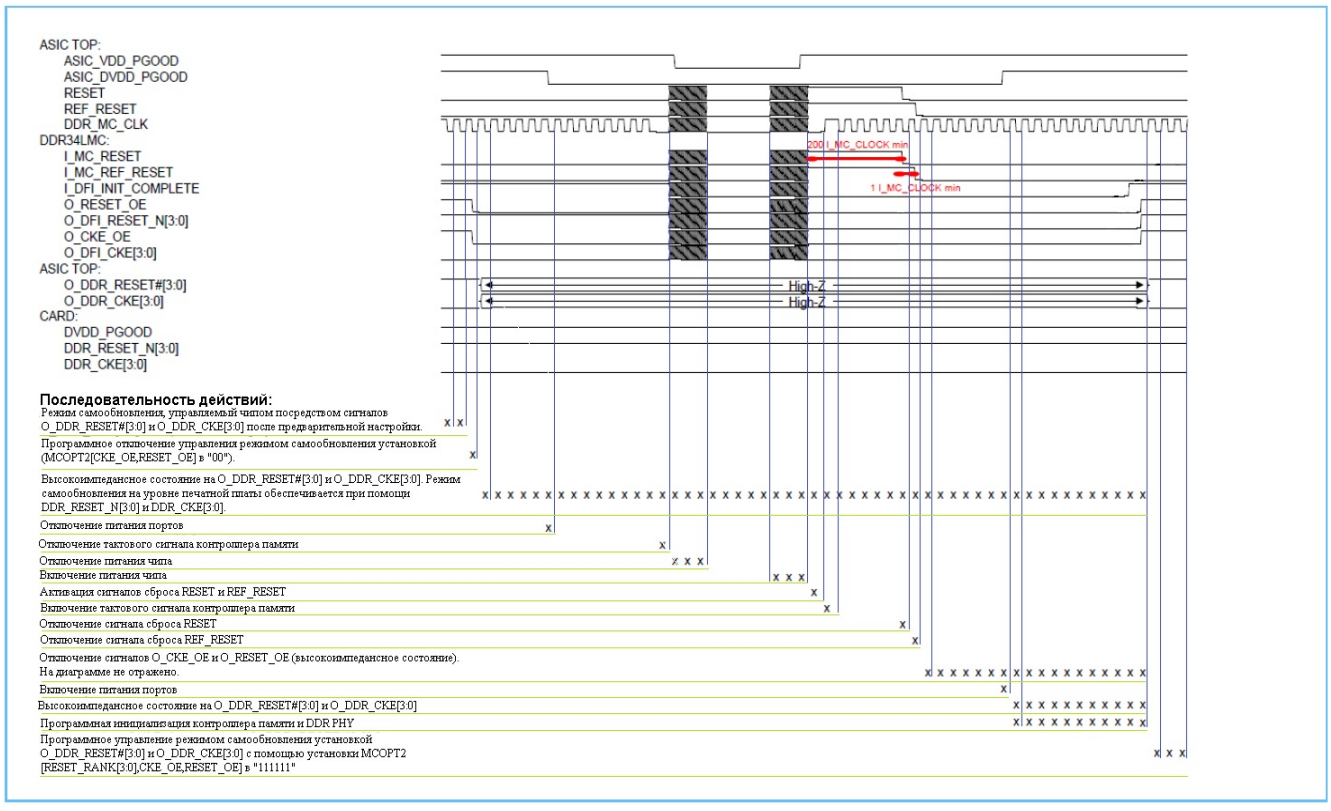


Рисунок 1.128 - Временная диаграмма сброса с сохранением данных в SDRAM

1.3.8.4.5.5 Инициализация контроллера внешней динамической памяти DDR34LMC

Контроллер внешней динамической памяти DDR34LMC должен быть инициализирован перед операциями, осуществляемыми над ним. Далее приведена последовательность инициализации:

- Включить питания, обеспечить стабильности тактового сигнала I_MC_CLOCK и удерживание I_MC_RESET в течение, по крайней мере, 200 циклов тактового сигнала при устойчивом питании. В зависимости от I_MC_REF_RESET, в то время как I_MC_RESET удерживается 1 такт и затем сбрасываемся, интерфейс памяти DDR находится в состоянии сброса или в состоянии регенерации.
- Настроить DDR PHY, чтобы гарантировать завершение им основной последовательности сброса, стабильность выхода PHY PLL, и возможность интерпретировать DFI команды от контроллера памяти. Этот шаг обеспечивает стабильность тактового сигнала на устройстве памяти SDRAM.
- SDRAM RESET_N может быть деактивирован (установлен в 1) при помощи установки в MCOPT2[RESET_RANK] значения "0000" через определенный период времени (обычно не меньше 200 мкс) после первого шага инициализации. После деактивации SDRAM RESET_N, минимальный период ожидания установки CKE равен 500 мкс. Во время периода ожидания, SDRAM CK и CK#, которые управляются из вне контроллера памяти, должны быть запущены и включены.
- После описанных выше трех шагов, контроллер памяти входит в режим «включенного питания» с MCOPT2[SELF_REF_EN]=1 и неактивным сигналом CKE, направленным к устройству памяти (0).
- Затем система готовиться к инициализации устройства памяти SDRAM при помощи загрузки регистров управления (MCOPT1, IOCNTL, PHYCNTL, ODTRx), конфигурации (CFGRx, SMRx), временных параметров (SDTRx) внутри контроллера памяти и регистров последовательности инициализации (INITSEQn и INITCMDn.
- Настроить счетчики времени завершения калибровки T_PHYUPD0-3. Хоть это и не требуется, данная настройка предлагается, поскольку это позволяет отлаживать ошибки калибровки.
- Выход из режима «включенного питания» при помощи записи в MCOPT2[SELF_REF_EN] значения 0 и в MCOPT2[INIT_START] значение 1. Это действие устанавливает сигнал CKE в 1, и затем выжидается задержка в tXPR тактов (определенная в регистре SDTR0) до последовательности инициализации (задаваемую в регистрах INITSEQn и INITCMDn), которая запускается при помощи бита INIT_START. Когда последовательность инициализации завершается, MCSTAT[INIT_COMPLETE] устанавливается в 1.
- Настроить регистры, связанные с калибровкой DDR PHY.
- Вызвать начальную калибровку DDR PHY при помощи установки MCOPT2[DFI_INIT_START] в 1 до тех пор, пока MCSTAT[DFI_INIT_COMPLETE] не установится в 0.
- Опросить регистр DDR PHY о статусе калибровки проверки. Когда калибровка закончится, проверить, есть ли сообщения об ошибке от DDR PHY в регистре статуса калибровки инициализации. После успешного завершения калибровки, установить MCOPT2[MC_ENABLE] в 1. Если необходимо отложить обновление, необходимо установить MCOPT1[REF_DEF] в 1.

Ядро готово для работы с памятью.

1.3.9 Подсистема высокоскоростных интерфейсов HSIF

1.3.9.1 Описание подсистемы HSIF

1.3.9.1.1 Структурная схема и состав блоков подсистемы

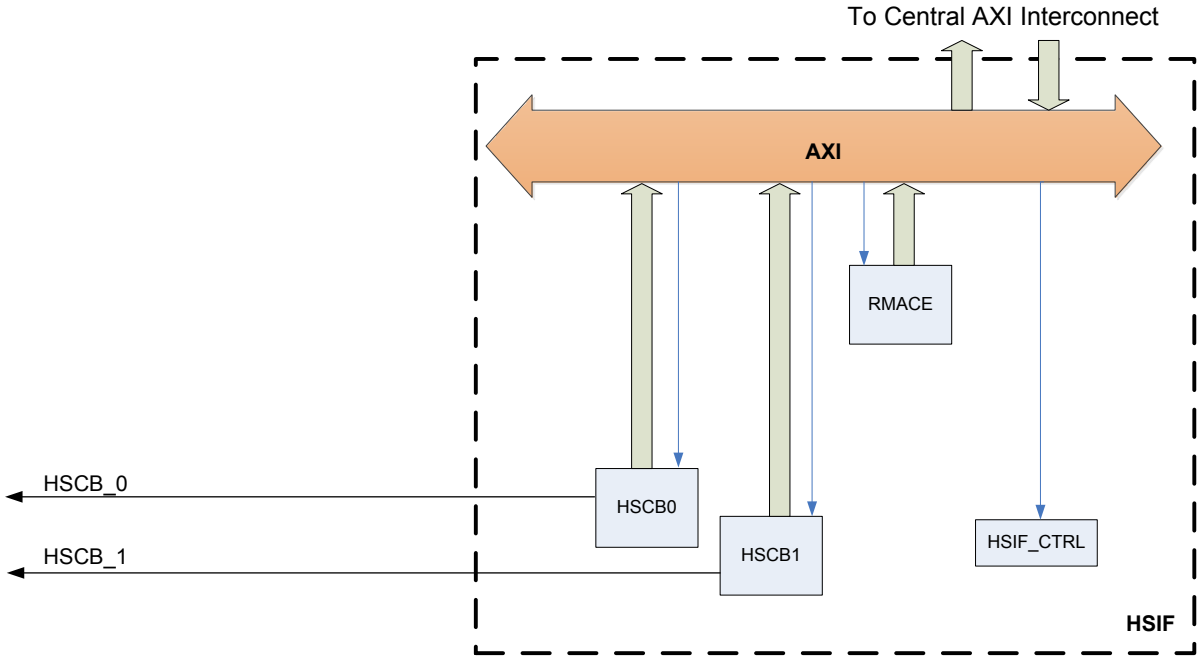


Рисунок 1.129 - Структурная схема подсистемы HSIF

Подсистема HSIF состоит из следующих блоков:

- Два контроллера интерфейса SpaceWire (HSCB0, HSCB1).
- Блок криптообработки RMACE.
- Блок управляющих регистров HSIF_CTRL.

1.3.9.1.2 Адресация блоков подсистемы HSIF

Таблица 1.685 - Карта адресов подсистемы HSIF.

Блок	Адрес на шине APB
HSIF_CTRL	0x3C023000 – 0x3C023FFF
RMACE	0x3C024000 – 0x3C024FFF
HSCB0	0x3C025000 – 0x3C025FFF
HSCB1	0x3C026000 – 0x3C026FFF

1.3.9.1.3 Блок управляющих регистров

Блок управляющих регистров HSIF_CTRL подсистемы HSIF микросхемы предназначен для настройки работы подсистемы. Блок позволяет управлять следующими функциями:

- Управлять сигналами LVDS буферов для блоков HSCB0(1);
- Расширение адресного пространства для контроллеров HSCB0(1) и блока RMACE.

Таблица 1.686 - Карта регистров блока HSIF_CTRL

Название	Адрес	Доступ	Значение после сброса	Описание
HSIF_ID_REG	0x0	RO	0x 46495348	Регистр идентификации
LVDS_CONTROL0	0x04	RW	0x0	Регистр управления сигналами LVDS буферов для блоков HSCB0(1)
HSCB0_RADDR_EXTEND	0x08	RW	0x0	Расширение адреса для 36-разрядной адресации канала чтения из контроллера HSCB0: Биты [3:0] - соответствуют 4 старшим разряда адреса
HSCB0_WADDR_EXTEND	0x0C	RW	0x0	Расширение адреса для 36-разрядной адресации канала записи контроллера HSCB0: Биты [3:0] - соответствуют 4 старшим разряда адреса
HSCB1_RADDR_EXTEND	0x10	RW	0x0	Расширение адреса для 36-разрядной адресации канала чтения из контроллера HSCB1: Биты [3:0] - соответствуют 4 старшим разряда адреса
HSCB1_WADDR_EXTEND	0x14	RW	0x0	Расширение адреса для 36-разрядной адресации канала записи контроллера HSCB1: Биты [3:0] - соответствуют 4 старшим разряда адреса
RMACE_RADDR_EXTEND	0x18	RW	0x0	Расширение адреса для 36-разрядной адресации канала чтения из блока RMACE: Биты [3:0] - соответствуют 4 старшим разряда адреса
RMACE_WADDR_EXTEND	0x1C	RW	0x0	Расширение адреса для 36-разрядной адресации канала записи блока RMACE: Биты [3:0] - соответствуют 4 старшим разряда адреса

1.3.9.1.3.1 Поля регистра LVDS_CONTROL0

Управляет сигналами LVDS буферов для блоков HSCB0 и HSCB1 .

Таблица 1.687 - Поля регистра LVDS_CONTROL0

Биты	Поле	Знач. по умолч.	Описание
0	REB_HSCB0_DI	0x0	Управление сигналом REB буфера HSCB0_DI
1	CM_ENF_HSCB0_DI	0x0	Управление сигналом CM_ENF буфера HSCB0_DI
2	REB_HSCB0_SI	0x0	Управление сигналом REB буфера HSCB0_SI
3	CM_ENF_HSCB0_SI	0x0	Управление сигналом CM_ENF буфера HSCB0_SI
4	ENB_HSCB0_DO	0x0	Управление сигналом ENB буфера HSCB0_DO
5	MODE_HSCB0_DO	0x0	Управление сигналом MODE буфера HSCB0_DO
6	ENB_HSCB0_SO	0x0	Управление сигналом ENB буфера HSCB0_SO
7	MODE_HSCB0_SO	0x0	Управление сигналом MODE буфера HSCB0_SO
8	REB_HSCB1_DI	0x0	Управление сигналом REB буфера HSCB1_DI
9	CM_ENF_HSCB1_DI	0x0	Управление сигналом CM_ENF буфера HSCB1_DI
10	REB_HSCB1_SI	0x0	Управление сигналом REB буфера HSCB1_SI
11	CM_ENF_HSCB1_SI	0x0	Управление сигналом CM_ENF буфера HSCB1_SI
12	ENB_HSCB1_DO	0x0	Управление сигналом ENB буфера HSCB1_DO
13	MODE_HSCB1_DO	0x0	Управление сигналом MODE буфера HSCB1_DO
14	ENB_HSCB1_SO	0x0	Управление сигналом ENB буфера HSCB1_SO
15	MODE_HSCB1_SO	0x0	Управление сигналом MODE буфера HSCB1_SO
31:4	-	-	reserve

1.3.9.2 Описание блоков подсистемы HSIF

1.3.9.2.1 Контроллер прямого доступа к памяти AXI-DMA

Контроллер прямого доступа к памяти AXI-DMA (AXI-DMA) не является самостоятельным блоком - он входит в состав более крупных блоков для обеспечения взаимодействия этих блоков с коммутационной средой микросхемы.

Блоки, в которых используется AXI-DMA:

- Блок криптообработки RMACE в составе HSIF (раздел 1.3.9.2.2)
- Контроллер HSCB в составе HSIF (раздел 1.3.9.2.3)
- Контроллер интерфейса Fiber Channel в составе XHSIFx (раздел 1.3.10).

В данном разделе приведено описание работы с AXI-DMA, общее для всех блоков, использующих его функционал, а нюансы работы в составе более крупных блоков описаны в разделах, соответствующих описанию этих блоков.

Блок AXI-DMA рассчитан на работу с дескрипторами, расположенными во внешней памяти.

Блок AXI-DMA включает два независимых канала: канал чтения из памяти и канал записи в память. Принципы прямого доступа в память для каждого канала идентичны. Для каждого канала AXI-DMA задаётся отдельный набор дескрипторов.

Блок верхнего уровня AXI-DMA-TOP включает в себя непосредственно контроллер ПДП с интерфейсом AXI – AXI-DMA и вспомогательные блоки APB Slave для записи регистров и выполнения программного сброса, и обработчик прерываний IRQ handler.

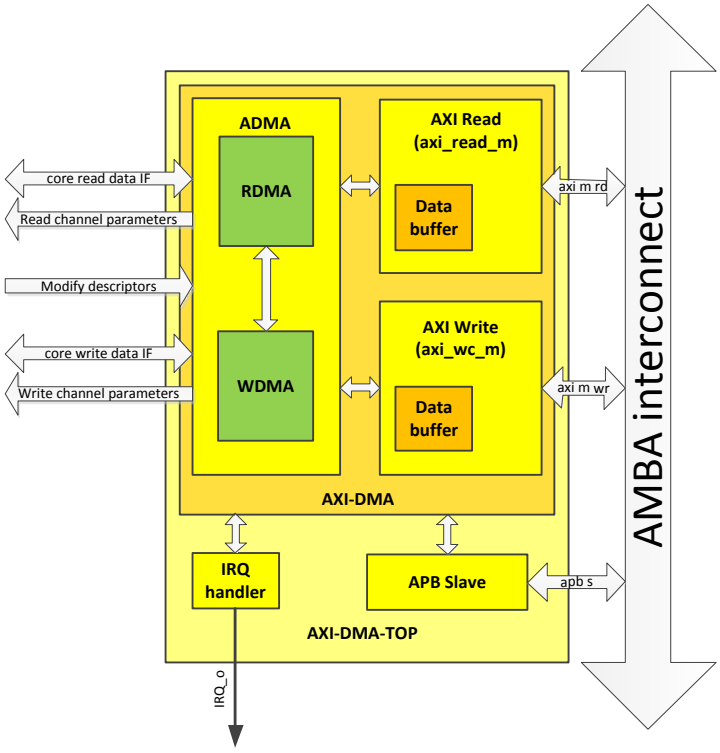


Рисунок 1.130 - Структурная схема AXI-DMA.

В состав блока AXI-DMA входят блоки, реализующие интерфейс AXI (канал чтения данных из внешней памяти (AXI Read) и канал записи данных в память (AXI Write)), и контроллер прямого доступа к памяти (ADMA) с отключаемой поддержкой внешней таблицы дескрипторов.

Блок ADMA поддерживает 32-разрядную адресацию и работу с блоками памяти от 1 байта до (1 Мбайта - 1) (с возможным расширением до (64 Мбайт - 1)).

1.3.9.2.1.1 Формат дескриптора

Таблица дескрипторов представляет собой непрерывный список из 8-байтовый строк. Начальный адрес таблицы (адрес первого дескриптора) и размер таблицы задаются при настройке AXI-DMA в соответствующих регистрах. В ходе работы AXI-DMA построчно считывает таблицу. После того, как AXI-DMA считает последний дескриптор из таблицы, указатель на новый дескриптор автоматически переходит на начальный адрес таблицы и чтение дескрипторов продолжается.

Доступными для работы AXI-DMA являются только дескрипторы типа Tran. В случае, если AXI-DMA дошло до чтения дескриптора с типом отличным от Tran оно останавливает свою работу. Более подробно работа AXI-DMA рассмотрена в разделах 1.3.9.2.1.3 и далее.

Размер дескриптора составляет 8 байт. Формат полей дескриптора представлен в таблице ниже.

Таблица 1.688 – Формат дескриптора AXI-DMA

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Custom*		[31:26]	6 бит для использования в зависимости от задачи
Length		[25:6]	Длина блока данных (до 1M)
Атрибуты	Act2	[5]	Тип дескриптора 00 Compl Дескриптор выполнен 01 Compl Дескриптор выполнен 10 Tran Передача данных 11 Compl Дескриптор выполнен
	Act1	[4]	
	Act0*	[3]	бит для использования в зависимости от задачи
	Int	[2]	1 – Выработка прерывания по завершении выполнения
	Err**	[1]	1 – Ошибка при выполнении дескриптора
	Valid*	[0]	бит для использования в зависимости от задачи

*Данные поля содержат флаги, определяемые функционалом блока, в состав которого входит AXI-DMA

** Поле Err используется только при модификации дескриптора.

Размер блока в памяти, соответствующего дескриптору, определяется полем Length.

Таблица 1.689 – Пример укороченного поля Length.

20-bit Length	Длина (байт)
0x0_0000	0
0x0_0001	1
0x0_0002	2
...	...
0xf_ffff	1M – 1

Также возможно увеличение размера блока в памяти, соответствующего дескриптору, за счёт поля Custom (задаётся в настройках блока AXI-DMA – регистр SETTINGS).

Таблица 1.690 – Пример поля Length максимального размера.

26-bit {Custom, Length}	Длина (байт)
0x000_0000	0
0x000_0001	1
0x000_0002	2
...	...
0x3ff_ffff	64M – 1

В случае, если при передаче или приёме данных дескриптора ядром произошла ошибка, ядро может установить атрибут Err = 1 при модификации ***дескриптора***, при этом R(W)DMA выполняет действия аналогичные остановке работы с завершением операций текущего буфера (см. 1.3.9.2.1.3.3.3).

1.3.9.2.1.2 Регистровая карта

Регистровая карта ADMA разбита на три блока:

- Регистры общего назначения и настройки интерфейса AXI.
- Регистры настройки DMA каналов чтения.
- Регистры настройки DMA каналов записи.

Таблица 1.691 – Карта регистров блока AXI-DMA.

Название	Адрес	Доступ	Значение после сброса	Описание
Общие регистры ADMA				
ADMA_ID	0x800	RO	0x414d4441	Идентификатор устройства
REVISION	0x804	RO	0x11	Номер ревизии и количество каналов чтения и записи DMA
SW_RST	0x808	RW	0x0	Программный сброс
CH_STATUS	0x80C	RO	0x0	Статус каналов DMA.
Регистры настройки AXI				
ARLEN	0x810	RW	0xF	Значение ARLEN для транзакций по AXI (значимы только младшие четыре бита – [3:0])
AWLEN	0x814	RW	0xF	Значение AWLEN для транзакций по AXI (значимы только младшие четыре бита – [3:0])
BRESP	0x818	RO	0x0	Значение BRESP для транзакций по AXI (значимы только младшие два бита – [1:0])
RAXI_ERR_ADDR	0x81C	RO	0x0	Адрес AXI-транзакции чтения, вызвавшей ошибку на шине AXI.
ARCACHE	0x820	RW	0x3	Значение ARCACHE для транзакций по AXI (значимы только младшие четыре бита – [3:0])
AWCACHE	0x824	RW	0x3	Значение AWCACHE для транзакций по AXI (значимы только младшие четыре бита – [3:0])
ARPROT	0x828	RW	0x2	Значение ARPROT для транзакций по AXI (значимы только младшие три бита – [2:0])
AWPROT	0x82c	RW	0x2	Значение AWPROT для транзакций по AXI (значимы только младшие три бита – [2:0])
ARLOCK	0x830	RW	0x0	Значение ARLOCK для транзакций по AXI (значимы только младшие два бита – [1:0])
AWLOCK	0x834	RW	0x0	Значение AWLOCK для транзакций по AXI (значимы только младшие два бита – [1:0])
ARBURST	0x838	RW	0x1	Значение ARBURST для транзакций по AXI (значимы только младшие два бита – [3:0])
AWBURST	0x83c	RW	0x1	Значение AWBURST для транзакций по AXI (значимы только младшие два бита – [1:0])
Регистры DMA канала чтения (RDMA)				
RDMA_SETTINGS	0x900	RW	0x0	Настройки DMA канала записи
RDMA_STATUS	0x904	RO	0x0	Статус канала записи
RDMA_SYS_ADDR	0x908	RW	0x0	Адрес таблицы дескрипторов для чтения из памяти

Продолжение таблицы 1.692

Название	Адрес	Доступ	Значение после сброса	Описание
RDMA_TBL_SIZE	0x90C	RW	TBL_SIZE	Размер таблицы дескрипторов (по умолчанию 2 кбайта)
RDMA_ADDR	0x910	RW	0x0	Текущий адрес
RDMA_LEN	0x914	RW	0x0	Длина блока данных
RDMA_DESC_ADDR	0x918	RW	0x0	Адрес текущего дескриптора
Регистры DMA канала записи (WDMA)				
WDMA_SETTINGS	0xA00	RW	0x0	Настройки DMA канала чтения
WDMA_STATUS	0xA04	RO	0x0	Статус канала чтения
WDMA_SYS_ADDR	0xA08	RW	0x0	Адрес таблицы дескрипторов для чтения из памяти
WDMA_TBL_SIZE	0xA0C	RW	TBL_SIZE	Размер таблицы дескрипторов (по умолчанию 2 кбайта)
WDMA_ADDR	0xA10	RW	0x0	Текущий адрес
WDMA_LEN	0xA14	RW	0x0	Длина блока данных
WDMA_DESC_ADDR	0xA18	RW	0x0	Адрес текущего дескриптора

Ниже приведено более подробное описание регистров

1.3.9.2.1.2.1 REVISION

Таблица 1.693 – Поля регистра REVISION

Биты	Поле	Описание
3:0	r_chan_num	Количество каналов чтения в устройстве
7:4	w_chan_num	Количество каналов записи в устройстве
31:8	-	Резерв. Особенности работы устройства

1.3.9.2.1.2.2 SW_RST

Регистр программного сброса.

Для выполнения сброса в данный регистр нужно записать 0x1. По завершении процедуры сброса в регистре аппаратно установится значение 0x0.

На выполнение программного сброса требуется время, т.к. необходимо дождаться завершения текущих AXI-транзакций, несоблюдение этого требования может привести к зависанию всей системы. Ожидание завершения AXI-транзакций выполняется аппаратно.

1.3.9.2.1.2.3 CH_STATUS

В регистре CH_STATUS отображается информация о прерываниях от разных каналов AXI-DMA. В текущей версии поддерживается только один канал чтения и один канал записи.

Таблица 1.694 – Поля регистра CH_STATUS.

Биты	Поле	Описание
0	rdma_irq	Возникло прерывание канала чтения.
15:1	reserved	Резерв. Прерывания от других каналов чтения.
16	wdma_irq	Возникло прерывание канала записи.
31:17	reserved	Резерв. Прерывания от других каналов записи.

Содержимое регистра сбрасывается после чтения**.**

1.3.9.2.1.2.4 R(W)DMA_SETTINGS

Регистр содержит настройки и маску прерываний DMA канала чтения (записи).

Таблица 1.695 – Поля регистров R(W)DMA_SETTINGS.

Биты	Поле	Описание
0	r(w)_desc_int	Маска прерывания «Закончено выполнение дескриптора канала чтения (записи) с 1 в поле Int»
1	r(w)_desc_end	Маска прерывания «Закончено выполнение остановки R(W)DMA»
2	r(w)_bad_desc	Маска прерывания «Дескриптор канала чтения помечен как выполненный (compl)»
3	r(w)_last_addr	Маска прерывания «Достигнут последний адрес блока данных качала чтения при работе без поддержки внешней таблицы дескрипторов»
4	-	Резерв
5	r(w)_axi_error	Маска прерывания «Сигнала прерывания при возникновении ошибки на системной шине (RRESP = SLVERR для канала чтения или BRESP = SLVERR для канала записи)»
6	r(w)_buf_full	Маска прерывания «Буфер канала чтения (записи) полон»
7	r(w)_buf_empty*	Маска прерывания «Буфер канала чтения (записи) пуст»
27:8	-	Резерв
28	en_r(w)dma**	Включен канал чтения (записи) (1 – активно)
29	en_ r(w)dma_desc_tbl	Поддержка внешней таблицы дескрипторов (1 – активно)
30	r(w)dma_long_len	Расширение поле Length за счёт поля Custom (1 – активно)
31	cancel_r(w)dma***	Остановка работы R(W)DMA (1 – активно)

* Прерывания r(w)_buf_empty могут приходить только после завершения работы R(W)DMA, т.е. при опустошении буфера в то время, пока R(W)DMA выдаёт адреса, прерывания r(w)_buf_empty не приходят.

** Запись 1 в поле en_r(w)dma включает R(W)DMA, когда R(W)DMA завершает работу данное поле аппаратно сбрасывается в 0 (более подробно см. 1.7).

*** Запись 1 в поле cancel_r(w)dma запускает процесс остановки R(W)DMA (более подробно см. 1.7.3), по завершении остановки данное поле аппаратно сбрасывается в 0 и выдается соответствующее прерывание, если установлена маска.

1.3.9.2.1.2.5 RDMA_STATUS

Регистр статуса канала чтения.

Таблица 1.696 – Поля регистра RDMA_STATUS.

Биты	Поле	Описание
0	r_desc_int	Закончено выполнение дескриптора канала чтения с 1 в поле Int
1	r_desc_end	Закончено выполнение остановки RDMA
2	r_bad_desc	Дескриптор канала чтения помечен как выполненный (compl)
3	r_lsat_addr	Достигнут последний адрес блока данных качала чтения при работе без поддержки внешней таблицы дескрипторов
4	-	Резерв
5	r_axi_error	Сигнал прерывания при возникновении ошибки на системной шине (RRESP = SLVERR) канал чтения
6	r_buf_full	Буфер канала чтения полон
7	r_buf_empty	Буфер канала чтения пуст
8	active_trans	Признак активной AXI-транзакции: если в текущий момеонт выполняется хотя бы одна AXI-транзакция, то данный бит находится в состоянии «1», в противном случае – в состоянии «0».
11:9	-	Резерв
12	act_req	Наличие активных (не завершенных) запросов на чтение данных
11	-	Резерв
22:16	cnt_word	Количество слов в буфере данных канала чтения
31:23	-	Резерв

Содержимое регистра сбрасывается после чтения**.**

1.3.9.2.1.2.6 WDMA_STATUS

Таблица 1.697 – Поля регистра WDMA_STATUS.

Биты	Поле	Описание
0	w_desc_int	Закончено выполнение дескриптора канала записи с 1 в поле Int
1	w_desc_end	Закончено выполнение остановки WDMA
2	w_bad_desc	Дескриптор канала записи помечен как выполненный (compl)
3	w_lsat_addr	Достигнут последний адрес блока данных качала записи при работе без поддержки внешней таблицы дескрипторов
4	-	Резерв
5	w_axi_error	Сигнал прерывания при возникновении ошибки на системной шине (BRESP = SLVERR) канал записи
6	w_buf_full	Буфер канала записи полон
7	w_buf_empty	Буфер канала записи пуст
31:8	-	Резерв

Содержимое регистра сбрасывается после чтения.

1.3.9.2.1.2.7 R(W)DMA_TBL_SIZE

32-битный регистр размера таблицы дескрипторов.

В начале работы в этот регистр программно записывается размер таблицы дескрипторов в байтах. Размер должен быть выровнен по 8 байт (длина одного дескриптора).

По достижении конца таблицы AXI-DMA автоматически возвращается к адресу первого дескриптора таблицы, и работа продолжается в нормальном режиме.

Текущее значение регистра не изменяется. Запись в данный регистр возможна только при выключенном R(W)DMA (en_r(w)dma == 0). Попытка записи в данный регистр во время работы R(W)DMA игнорируется.

1.3.9.2.1.2.8 R(W)DMA_SYS_ADDR

32-битный адрес таблицы дескрипторов.

В начале работы в этот регистр программно записывается адрес таблицы дескрипторов (если она используется).

Значение этого регистра в процессе работы не изменяется.

Запись в данный регистр возможна только при выключенном R(W)DMA (en_r(w)dma == 0). Попытка записи в данный регистр во время работы R(W)DMA игнорируется.

При чтении дескрипторов R(W)DMA из таблицы не анализируется граница в 4 Кбайта. Установка не выровненного по 8 байт адреса таблицы дескрипторов может привести к ошибке.

1.3.9.2.1.2.9 R(W)DMA_ADDR

32-битный регистр текущего адреса в памяти.

В данном регистре отображается адрес текущей ячейки памяти.

При работе с внешней таблицей дескрипторов в этот регистр автоматически переписывается адрес из дескриптора.

При работе без поддержки внешней таблицы дескрипторов, в этот регистр программно записывается адрес, начиная с которого будет идти обращение в память.

Значение регистра увеличивается на 8 при каждом обращении в память. Таким образом, регистр содержит указатель на текущий адрес в памяти.

Во время работы R(W)DMA считанное значение регистра может отличаться от реального.

Запись в данный регистр возможна только при выключенном R(W)DMA (en_r(w)dma == 0). Попытка записи в данный регистр во время работы R(W)DMA игнорируется.

1.3.9.2.1.2.10 R(W)DMA_LEN

32-битный регистр длины блока в памяти, к которому идет.

При работе с внешней таблицей дескрипторов в этот регистр автоматически переписывается размер блока данных из дескриптора.

При работе без поддержки внешней таблицы дескрипторов, в этот регистр программно записывается размер блока данных, к которому будет идти обращение в память.

Значение регистра уменьшается на 8 при каждом обращении в память. Таким образом, значение в регистре указывает на оставшийся объем памяти, выделенной этим дескриптором.

Во время работы R(W)DMA считанное значение регистра может отличаться от реального.

Запись в данный регистр возможна только при выключенном R(W)DMA (en_r(w)dma == 0). Попытка записи в данный регистр во время работы R(W)DMA игнорируется.

1.3.9.2.1.2.11 R(W)DMA_DESC_ADDR

32-битный адрес текущего дескриптора (указатель текущего дескриптора).

Значение в данном регистре задаёт адрес дескриптора, с которым в данный момент работает R(W)DMA. Если записей в данный регистр не было, то в начале работы в него аппаратно переписывается адрес таблицы дескрипторов, т.е. адрес первого дескриптора в таблице. При переходе к новой строке таблицы, значение этого регистра меняется – в нём устанавливается адрес текущего дескриптора. Если программно записать адрес дескриптора из таблицы в R(W)DMA_DESC_ADDR, то после включения R(W)DMA начнёт работу с этого дескриптора. Запись в регистр R(W)DMA_DESC_ADDR допустима при выключенном R(W)DMA (0 в поле en_r(w)dma (бит 28) регистра R(W)DMA_SETTINGS) за исключением случай описанного в п. 1.7.3 «Интерактивное изменение таблицы дескрипторов».

Во время работы R(W)DMA считанное значение регистра может отличаться от реального.

Адрес, записанный в регистр R(W)DMA_DESC_ADDR должен попадать в таблицу, задаваемую значениями в регистрах R(W)DMA_SYS_ADDR и R(W)DMA_TBL_SIZE, т.к. R(W)DMA_DESC_ADDR задаёт положение в текущей таблице.

После того, как будет отработан последний дескриптор в таблице, указатель текущего дескриптора перейдёт на первый дескриптор в таблице (адрес указан в регистре R(W)DMA_SYS_ADDR).

При чтении дескрипторов R(W)DMA из таблицы не анализируется граница в 4 Кбайта. Установка не выровненного по 8 байт адреса таблицы дескрипторов может привести к ошибке.

1.3.9.2.1.3 Алгоритм работы с AXI-DMA

Ниже приведена блок-схема, описывающая основной алгоритм работы AXI-DMA.

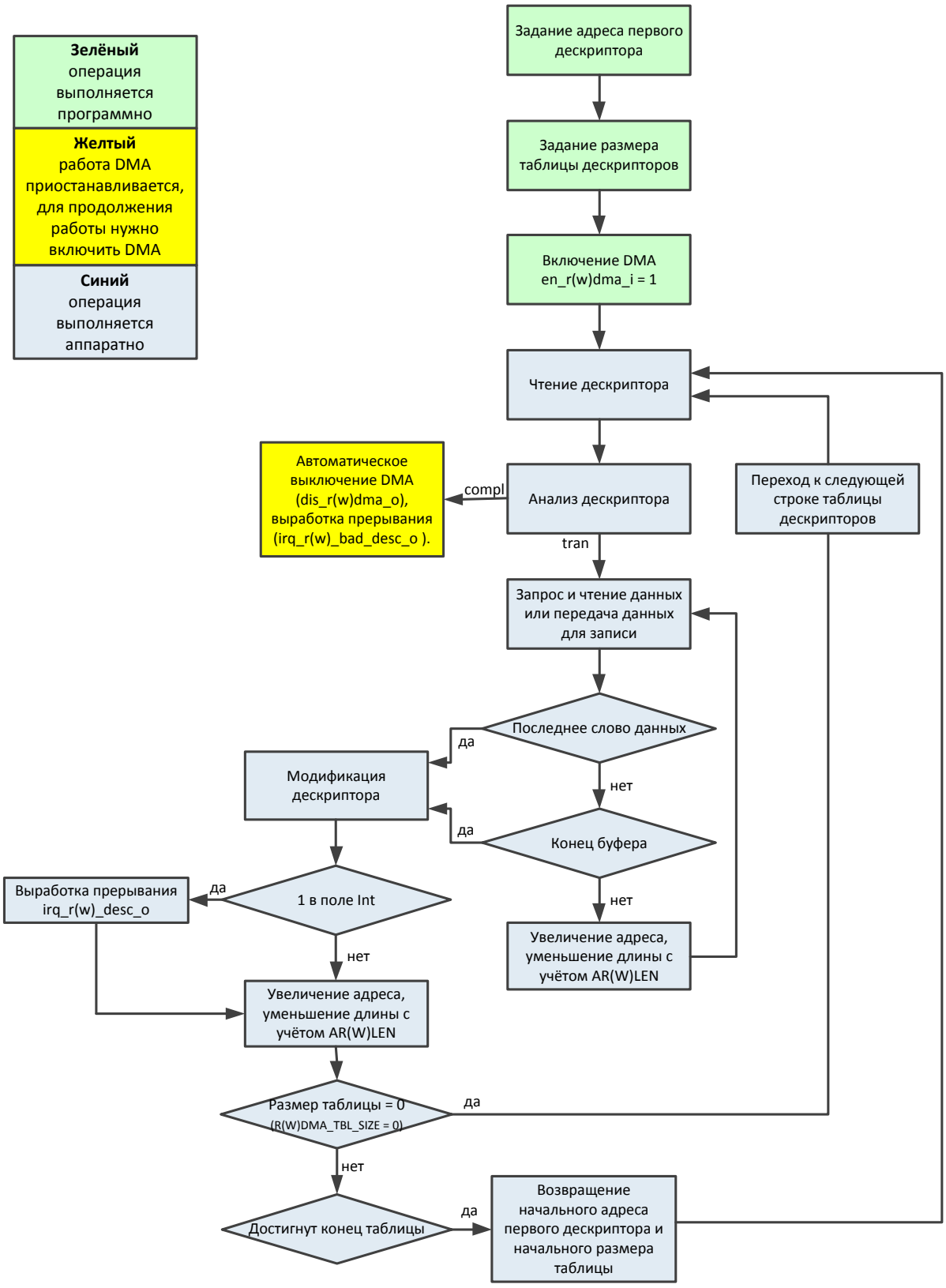


Рисунок 1.131 – Алгоритм работы с AXI-DMA

При работе с поддержкой таблицы дескрипторов сначала из таблицы считывается строка-дескриптор, описывающая буфер в памяти, затем выполняются операции чтения из памяти или записи в память по адресам соответствующим данному буферу.

1.3.9.2.1.3.1 Определение типа дескриптора

После чтения дескриптора из памяти в первую очередь анализируются поля Act2 и Act1.

- Compl – дескриптор выполнен, выдается прерывание, т.к. область данных, на которую он указывает уже была обработана.
- Tran – нормальная работа, выполняется чтение или запись данных.

1.3.9.2.1.3.2 Нормальная работа

Значения полей Address и Length дескриптора переписываются в регистры R(W)DMA_ADDR и R(W)DMA_LEN. DMA машина генерирует адреса. По мере выработки адресов смещается указатель текущего адреса в регистре R(W)DMA_ADDR, и уменьшается количество оставшихся байт в регистре R(W)DMA_LEN. По достижении конца блока данных либо при приходе сигнала окончания передачи данных (только при записи в память) генерируется прерывание (если в дескрипторе установлены соответствующие флаги) и переписывается дескриптор типа tran. При переписывании дескриптора:

- Значение поля Address дескриптора не изменяется.
- Для битов Act2(5) и Act1(4) блок AXI-DMA аппаратно устанавливает значение 01 (compl), указывающее, что дескриптор отработан.
- Остальные биты могут содержать значение, определяемое пользователем для применения в конкретном устройстве, они записываются в соответствии с данными полученными от ядра по шинам out_r_attr_i и out_w_attr_i для дескрипторов канала чтения и канала записи соответственно.
- Для начала перезаписи дескриптора ядро выдает строб out_r(w)_valid_i.
- В случае если таблица дескрипторов содержит более двух дескрипторов:
- Если таблица дескрипторов для RDMA содержит более двух дескрипторов, то RDMA приступает к обработке следующего tran-дескриптора сразу после завершения операций с текущим, не дожидаясь перезаписи дескриптора. Если таблица дескрипторов для RDMA содержит два или один дескриптор, то RDMA приступает к обработке следующего tran-дескриптора только после перезаписи текущего дескриптора.

WDMA для ускорения работы поддерживает механизм предсчитывания дескрипторов. Если таблица дескрипторов для WDMA содержит более четырёх дескрипторов, то WDMA читает первый дескриптор, начинает выполнять операции связанные с ним, и на фоне этого читает второй, таким образом к моменту завершения первого буфера для данных (описывается первым дескриптором), уже готов второй буфер. Если таблица дескрипторов для WDMA содержит четыре или менее дескрипторов, то WDMA считывает следующий дескриптор только после перезаписи текущего дескриптора.

Замедление в работе RDMA и WDMA при использовании таблиц дескрипторов небольшого размера сделано для того, чтобы предотвратить ситуацию, когда R(W)DMA считывает дескриптор сразу после того, как модифицировало его.

1.3.9.2.1.3.3 Механизмы остановки AXI-DMA

Остановка AXI-DMA может выполняться в любой момент времени.
AXI-DMA поддерживает 3 различных механизма остановки работы.

1.3.9.2.1.3.3.1 Программный сброс

Программный сброс выполняется одновременно для всего блока AXI-DMA.

При программном сбросе завершается текущая AXI-транзакция, после чего сбрасывается содержимое всех внутренних буферов, а внутренние регистры возвращаются в исходное состояние.

Программный сброс выполняется записью 1 в регистр SW_RST. Завершение текущей AXI-транзакции требует некоторого времени, поэтому программный сброс выполняется не мгновенно. По завершении сброса канала соответствующее поле в регистре SW_RST аппаратно устанавливается в 0.

Для дальнейшей работы необходимо снова выполнить программирование AXI-DMA.

1.3.9.2.1.3.3.2 Приостановка работы

Приостановка работы каждого канала выполняется независимо. При приостановке работы содержимое регистров AXI-DMA сохраняется. Значение регистров R(W)DMA_ADDR и R(W)DMA_LEN указывает на текущее положение R(W)DMA. Значение регистра R(W)DMA_DESC_ADDR указывает на обрабатываемый на момент приостановки дескриптор, в случае работы с поддержкой таблицы дескрипторов.

Для приостановки работы необходимо записать 0 в поле en_r(w)dma регистра R(W)DMA_SETTINGS. При этом RDMA перестаёт передавать данные ядру, завершает текущую AXI-транзакцию и перестает запрашивать данные из памяти. WDMA перестаёт принимать данные от ядра и завершает текущую транзакцию.

Для продолжения работы необходимо записать 1 в поле en_r(w)dma регистра R(W)DMA_SETTINGS.

Перезапись каких-либо регистров за исключением R(W)DMA_SETTINGS во время приостановки работы запрещена.

При приостановке работы функционирование RDMA и WDMA одинаково как для режима прямой адресации и для работы с поддержкой таблицы дескрипторов.

1.3.9.2.1.3.3.3 Остановка работы с завершением операций текущего буфера

Остановка работы с завершением операций текущего буфера возможна только для режима работы с поддержкой таблицы дескрипторов. При остановке работы с завершением операций текущего буфера содержимое регистров AXI-DMA сохраняется. Значение регистра R(W)DMA_DESC_ADDR указывает на текущее положение R(W)DMA в таблице дескрипторов.

Для остановки работы необходимо записать 1 в поле cancel_r(w)dma регистра R(W)DMA_SETTINGS. После завершения процесса остановки в поле cancel_r(w)dma регистра R(W)DMA_SETTINGS будет установлен 0 и выдано прерывание r(w)_desc_end.

RDMA завершает выполнение текущего дескриптора (адрес дескриптора в регистре RDMA_DESC_ADDR), модифицирует его, останавливается, выдаёт прерывание r_desc_end и аппаратно устанавливает в 0 поля en_rdma и cancel_rdma регистра RDMA_SETTINGS. Значение регистра RDMA_DESC_ADDR указывает на последний завершенный дескриптор. Внутренние буферы RDMA автоматически опустошаются.

Для WDMA возможно два варианта поведения.

- В случае если команда остановки пришла до того, как WDMA начало передавать данные, соответствующие дескриптору, остановка будет выполнена сразу. WDMA остановится, выдаст прерывание w_desc_end и аппаратно устанавливает в 0 поля en_wdma и cancel_wdma регистра WDMA_SETTINGS. При этом не будет произведено модификации дескриптора. Таким образом, в регистре WDMA_DESC_ADDR будет содержаться адрес дескриптора следующего за последним выполненным, а дескриптор с соответствующим адресом в таблице будет помечен как tran (готовый к выполнению).

- В случае если WDMA начало передавать данные до прихода команды остановки, WDMA завершает выполнение текущего дескриптора (адрес дескриптора в регистре WDMA_DESC_ADDR), модифицирует его, останавливается, выдаёт прерывание w_desc_end и аппаратно устанавливает в 0 поля en_wdma и cancel_wdma регистра WDMA_SETTINGS. Значение регистра WDMA_DESC_ADDR указывает на последний завершенный дескриптор. Внутренние буферы WDMA автоматически опустошаются.

Процедура возобновления работы описана в разделе 1.3.9.2.1.3.3.5.

1.3.9.2.1.3.3.4 Аппаратная остановка работы

Если в данных для модификации дескриптора out_r(w)_attr_i установлен признак Err (out_r(w)_attr_i[1]), то выполняется аппаратная остановка работы с завершением операций с текущим буфером. Дальнейшая работа R(W)DMA аналогична программной остановке работы с завершением операций текущего буфера, как описано в 1.3.9.2.1.3.3.3.

1.3.9.2.1.3.3.5 Возобновление работы AXI-DMA и интерактивное изменение таблицы дескрипторов

Ниже приведены различные сценарии возобновления работы AXI-DMA после выполнения процедуры остановки.

1.3.9.2.1.3.3.6 Возобновление работы без изменения настроек R(W)DMA

После завершения процесса остановки R(W)DMA (поля cancel_r(w)dma и en_r(w)dma регистра R(W)DMA_SETTINGS содержат 0 и выдано прерывание r(w)_desc_en, если оно не замаскировано) допустимо включение R(W)DMA без изменения настроек. Для этого необходимо записать 1 в поле en_r(w)dma регистра R(W)DMA_SETTINGS. После включения R(W)DMA продолжит работу с места остановки. Если дескриптор, адрес которого записан в R(W)DMA_DESC_ADDR был выполнен, то будет зачитан следующий дескриптор в таблице. Если дескриптор, адрес которого записан в WDMA_DESC_ADDR не был выполнен (см. объяснение в п.1.3.9.2.1.3.3.3), он будет зачитан снова. После этого продолжится обычная работа.

1.3.9.2.1.3.3.7 Возобновление работы с изменением настроек R(W)DMA

После завершения процесса остановки R(W)DMA допустима установка указателя текущего дескриптора на любой дескриптор из таблицы, кроме того, допустимо изменение базового адреса и размера таблицы.

Изменение базового адреса и(или) размера таблицы (регистры R(W)DMA_SYS_ADDR и R(W)DMA_TBL_SIZE) необходимо производить до задания указателя текущего дескриптора (регистр R(W)DMA_DESC_ADDR). В случае изменения базового адреса и(или) размера таблицы необходимо дополнительно произвести запись в регистр R(W)DMA_DESC_ADDR.

Для включения R(W)DMA (возобновления работы) необходимо записать 1 в поле en_r(w)dma регистра R(W)DMA_SETTINGS. После этого R(W)DMA начнёт обработку дескрипторов, начиная с адреса, записанного в регистре R(W)DMA_DESC_ADDR. Далее продолжается стандартная работа.

1.3.9.2.1.3.3.8 Интерактивное изменение таблицы дескрипторов RDMA

AXI-DMA допускает интерактивное изменение таблицы дескрипторов канала чтения из памяти (RDMA), т.е. внесение корректив в таблицу без необходимости ожидания завершения работы или остановки RDMA.

Алгоритм интерактивного изменения таблицы дескрипторов:

- Подать сигнал остановки RDMA (записать 1 в поле cancel_rdma регистра RDMA_SETTINGS).
- Считать адрес текущего (n) дескриптора (регистр RDMA_DESC_ADDR).
- Заменить следующий (n+1) дескриптор или изменить настройки RDMA (изменение настроек описано в 1.7.2).
- Запустить RDMA (записать 1 в поле en_rdma регистра RDMA_SETTINGS), не дожидаясь завершения остановки.

В результате выполнения такой последовательности действий после обработки n-го дескриптора RDMA перейдет к выполнению следующего (n+1), который будет содержать необходимое значение (в случае изменения настроек RDMA перейдет к дескриптору с адресом, записанным в регистре RDMA_DESC_ADDR). Если не выполнять остановку, то механизм предзачитывание дескрипторов не позволяет однозначно определить, какой дескриптор считан RDMA в текущий момент.

Прерывание о завершении процесса остановки (r_desc_end) будет выдано только в том случае, если RDMA успеет завершить работу с дескриптором до того, как будет снова запущено. Для приведенного алгоритма рекомендуется замаскировать прерывание r_desc_end, тогда работа по приведенному алгоритму будет однообразна вне зависимости от того успело ли RDMA завершить работу с дескриптором.

Изменение таблицы дескрипторов по приведенному алгоритму допустимо неограниченное количество раз.

1.3.9.2.1.3.4 Работа с дескрипторами нулевой длины

Наличие дескриптора со значением 0 в поле Length не является ошибочной ситуацией.

При чтении такого дескриптора R(W)DMA перезаписывает его как выполненный в соответствии с очередью его прихода. Флаги при перезаписи дескриптора нулевой длины не сохраняются (значение всегда будет 0x_address[63:32]_00000010)

В случае если дескриптор нулевой длины содержал признак Int – прерывание будет выработано как только завершатся все операции с текущим дескриптором. Регистр R(W)DMA_IRQ_ADDR будет содержать адрес нулевого дескриптора, вызвавшего прерывание.

1.3.9.2.2 Блок криптообработки RMACE

Блок криптообработки RMACE (далее блок RMACE) реализует обмен данными между блоками памяти, минуя центральный процессор. Данные могут передаваться без изменения или подвергаться шифрованию. Так же RMACE позволяет выполнять логические операции над несколькими блоками памяти.

Блок RMACE поддерживает следующие алгоритмы работы:

- Прямая передача данных (direct).
- Операция XOR над двумя или более операндами (xor).
- Перестановка байт и/или бит в каждом байте (permutation).
- Шифрование/дешифрование по алгоритму DES в различных режимах (des).
- Шифрование/дешифрование по алгоритму 3DES в различных режимах (3des).
- Шифрование/дешифрование по алгоритму AES-128 в различных режимах (aes-128).
- Шифрование/дешифрование по алгоритму AES-192 в различных режимах (aes-192).
- Шифрование/дешифрование по алгоритму AES-256 в различных режимах (aes-256).
- Шифрование/дешифрование по алгоритму GOST в различных режимах (gost).
- Выработка имитовставки по алгоритму GOST (gost-imit)

Структурная схема блока RMACE представлена на рисунке ниже.

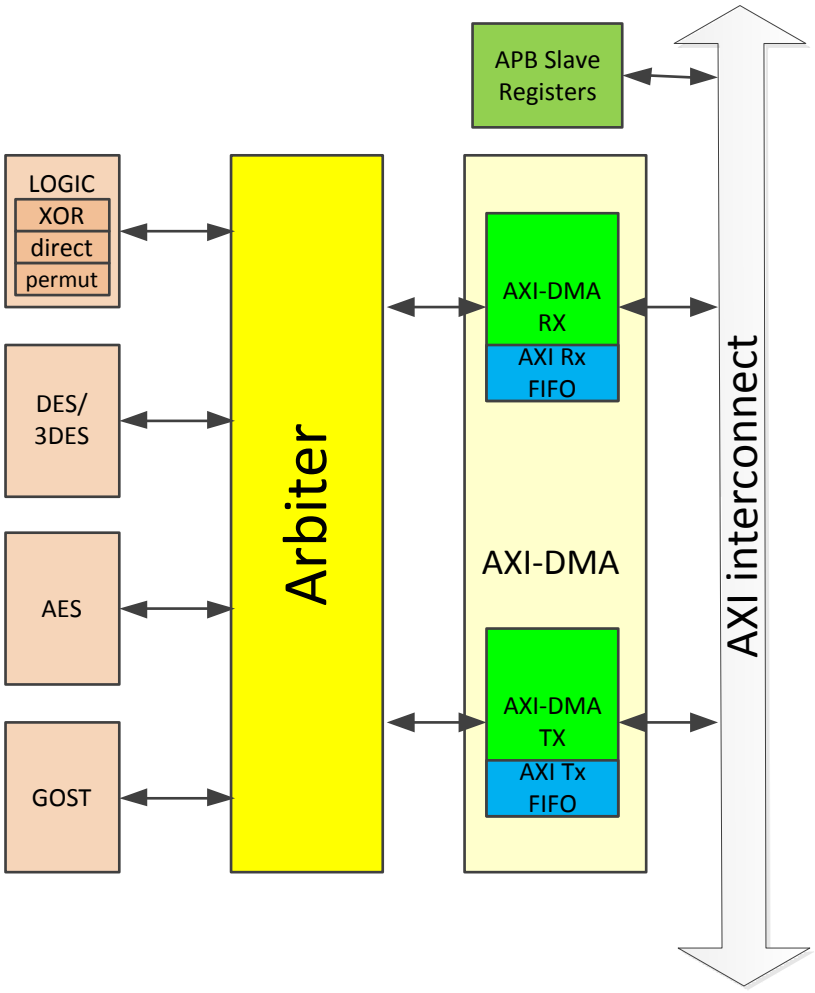


Рисунок 1.132 – Структурная схема блока криптообработки RMACE

В состав блока RMACE входят:

- Регистры управления (APB slaves registers) через APB интерфейс.
- AXI интерфейс для доступа в памяти (AXI-DMA). Каналы записи и чтения AXI интерфейса реализуют прямой доступ к памяти, работают независимо и предполагают работу с внешней таблицей дескрипторов.
- Управляющая логика (Arbiter).
- Набор блоков, выполняющих операции по передаче и преобразованию данных (в том числе реализующих различные алгоритмы шифрования).

1.3.9.2.2.1 Карта регистров

Регистры блока RMACE разделены на несколько групп.

Таблица 1.698 – Карта регистров RMACE.

Название	Адрес	Доступ	Значение после сброса	Описание
Системные регистры				
ID	0x000	RO	0x4543414d	Идентификатор устройства
VERSION	0x004	RO	0x00000101	Номер версии устройства
SW_RST	0x008	WO	0x0	Программный сброс ядра RMACE
Общие регистры блока AXI-DMA				
ADMA_ID	0x800	RO	0x414d4441	Идентификатор устройства
ADMA_REVISION	0x804	RO	0x11	Номер ревизии и количество каналов чтения и записи DMA
ADMA_SW_RST	0x808	RW	0x0	Программный сброс
ADMA_CH_STATUS	0x80C	RO	0x0	Статус каналов DMA.
Регистры настройки шины AXI				
ARLEN	0x810	RW	0xF	Значение ARLEN для транзакций по AXI (значимы только младшие четыре бита – [3:0])
AWLEN	0x814	RW	0xF	Значение AWLEN для транзакций по AXI (значимы только младшие четыре бита – [3:0])
BRESP	0x818	RO	0x0	Значение BRESP для транзакций по AXI (значимы только младшие два бита – [1:0])
RAXI_ERR_ADDR	0x81C	RO	0x0	Адрес AXI-транзакции чтения, вызвавшей ошибку на шине AXI.
ARCACHE	0x820	RW	0x3	Значение ARCACHE для транзакций по AXI (значимы только младшие четыре бита – [3:0])
AWCACHE	0x824	RW	0x3	Значение AWCACHE для транзакций по AXI (значимы только младшие четыре бита – [3:0])
ARPROT	0x828	RW	0x2	Значение ARPROT для транзакций по AXI (значимы только младшие три бита – [2:0])
AWPROT	0x82c	RW	0x2	Значение AWPROT для транзакций по AXI (значимы только младшие три бита – [2:0])
ARLOCK	0x830	RW	0x0	Значение ARLOCK для транзакций по AXI (значимы только младшие два бита – [1:0])
AWLOCK	0x834	RW	0x0	Значение AWLOCK для транзакций по AXI (значимы только младшие два бита – [1:0])
ARBURST	0x838	RW	0x1	Значение ARBURST для транзакций по AXI (значимы только младшие два бита – [1:0])
AWBURST	0x83c	RW	0x1	Значение AWBURST для транзакций по AXI (значимы только младшие два бита – [1:0])
Регистры DMA канала чтения (RDMA)				
RDMA_SETTINGS	0x900	RW	0x0	Настройки DMA канала записи
RDMA_STATUS	0x904	RO	0x0	Статус канала записи

Продолжение таблицы 1.699

Название	Адрес	Доступ	Значение после сброса	Описание
RDMA_SYS_ADDR	0x908	RW	0x0	Адрес таблицы дескрипторов для чтения из памяти
RDMA_TBL_SIZE	0x90C	RW	0x00000800	Размер таблицы дескрипторов (по умолчанию 2 кбайта)
RDMA_ADDR	0x910	RO	0x0	Текущий адрес
RDMA_LEN	0x914	RO	0x0	Длина блока данных
RDMA_DESC_ADDR	0x918	RW	0x0	Адрес текущего дескриптора
Регистры DMA канала записи (WDMA)				
WDMA_SETTINGS	0xA00	RW	0x0	Настройки DMA канала чтения
WDMA_STATUS	0xA04	RO	0x0	Статус канала чтения
WDMA_SYS_ADDR	0xA08	RW	0x0	Адрес таблицы дескрипторов для чтения из памяти
WDMA_TBL_SIZE	0xA0C	RW	0x00000800	Размер таблицы дескрипторов (по умолчанию 2 кбайта)
WDMA_ADDR	0xA10	RO	0x0	Текущий адрес
WDMA_LEN	0xA14	RO	0x0	Длина блока данных
WDMA_DESC_ADDR	0xA18	RW	0x0	Адрес текущего дескриптора

1.3.9.2.2.2 Алгоритм работы с RMACE

- Сформировать таблицу дескрипторов.
- Настроить DMA контроллеры каналов чтения из памяти (RDMA) и записи в память (WDMA).
- Включить RDMA и WDMA.
- Дождаться прерывания.
- Т.к. RMACE по сути контролер прямого доступа к памяти, не возможна работа только канала чтения или только канала записи.
- Дескрипторы RMACE делятся на два типа:
- Дескриптор конфигурации (Valid = 1).
- Дескриптор данных (Valid = 0).
- После завершения работы с данными, соответствующими дескриптору, происходит модификация дескриптора в таблице – дескриптор помечается как выполненный (compl).

Формат дескриптора канала чтения из памяти (RDMA) приведен в таблице 1.911:

Таблица 1.700 – Формат дескриптора канала чтения для RMACE

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Length		[31:6]	Длина блока данных (до 64M-1)
Атрибуты	Act2	[5]	Тип дескриптора 00 compl Дескриптор выполнен 01 compl Дескриптор выполнен 10 tran Передача данных 11 compl Дескриптор выполнен Должно быть 10
	Act1	[4]	
	Act0	[3]	0 – НЕ последний дескриптор массива данных 1 – последний дескриптор массива данных
	Int	[2]	1 - Выработка прерывания по завершении выполнения
	-	[1]	Зарезервировано
	Valid	[0]	1 – Дескриптор содержит конфигурацию 0 – дескриптор содержит данные

Для RDMA дескриптор конфигурации указывает на область памяти, которая содержит информацию о способе передачи данных (выполнять шифрование, логическое преобразование или передавать без изменений) и необходимую ключевую информацию.

Формат данных для дескриптора конфигурации (размер любого поля кратен 8 байтам):

Таблица 1.701 – Поля области данных для дескриптора конфигурации RMACE

Header
Key
IV
Custom

Дескриптор данных для RDMA указывает на область памяти, содержащей данные для обработки. Для выполнения логических операций над несколькими массивами данных используется поле Act0. Изменение значения этого поля позволяет разграничить дескрипторы одного массива данных от дескрипторов другого. Количество операндов в логических операциях указывается при конфигурации. Так же 1 в поле Act0 указывает на последний дескриптор в массиве данных для шифрования. Это нужно для шифрования массивов данных не выровненных по длине слова шифроалгоритма, т.к. неполное слово в этом случае обрабатывается особым образом.

При модификации дескрипторов RDMA изменяется только значение полей {Act2, Act1} – аппаратно устанавливается 10 (compl).

Формат дескриптора канала записи

Таблица 1.702 – Формат дескриптора канала записи для RMACE

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Length		[31:6]	Размер области памяти (до 64M-1)
Атрибуты	Act2	[5]	Тип дескриптора 00 compl Дескриптор выполнен 01 compl Дескриптор выполнен 10 tran Передача данных 11 compl Дескриптор выполнен Должно быть 10
	Act1	[4]	
	Act0	[3]	0 – НЕ последний дескриптор массива данных 1 – последний дескриптор массива данных
	Int	[2]	1 - Выработка прерывания по завершении выполнения
	-	[1]	Зарезервировано
	Valid	[0]	1 – Дескриптор чтения конфигурации 0 – Дескриптор записи данных

Для WDMA получение дескриптора конфигурации означает, что текущее состояние RMACE будет записано в память по содержащемуся в дескрипторе адресу.

После модификации дескриптор конфигурации имеет вид:

Таблица 1.703 – Формат дескриптора канала записи для RMACE после считывания конфигурации.

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Length		[31:6]	Размер конфигурационных данных
Атрибуты	Act2	[5]	01 (compl)
	Act1	[4]	
	Act0	[3]	1
	Int	[2]	1 - Выработка прерывания по завершении выполнения
	-	[1]	Зарезервировано
	Valid	[0]	1 (дескриптор конфигурации)

Дескриптор данных для WDMA указывает области в памяти, куда нужно поместить результат.

После модификации дескриптор данных имеет вид:

Таблица 1.704 – Формат дескриптора канала записи для RMACE после записи данных.

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Length		[31:6]	Размер записанного блока данных
Атрибуты	Act2	[5]	01 (compl)
	Act1	[4]	
	Act0	[3]	0 – НЕ последний дескриптор массива данных 1 – последний дескриптор массива данных
	Int	[2]	1 - Выработка прерывания по завершении выполнения
	-	[1]	Зарезервировано
	Valid	[0]	0 (дескриптор данных)

Дескриптор WDMA может описывать область памяти, большую, чем требуется для записи всех полученных данных, в этом случае, в поле Length будет указан реальный размер записанного массива данных, а поле Act0 будет содержать 0. В случае, если область памяти, описываемая дескриптором, будет недостаточна для записи всех данных, поле Length будет содержать тоже значение, что и при чтении дескриптора, поле Act0 будет содержать 1, а недостающие данные будут записаны в область памяти, соответствующую следующему дескриптору.

Для всех операций за исключением xor и gost-imit размеры данных после обработки RMACE не изменяется. Размер данных для xor уменьшается пропрционально количеству операндов. Размер данных для gost-imit определяется настройкой алгоритма GOST и не превосходит 8 байт.

1.3.9.2.2.3 Конфигурация

Ниже приведен формат области данных, определяющей конфигурацию RMACE

Таблица 1.705 – Формат области данных для дескриптора конфигурации RMACE.

header (8 байт)									
7		6	5	4		3	2	1	0
				39:36	35:32				
type		features	mode	endian	enc/dec	reserv			
0	nop	-	-	-	-	-			
1	direct	-	-	-					
2	xor	operands 2 - 255	-	1 – in 2 – out					
3	permut-ation	-	0 – bit 1 – byte	-					
4	des	-	0 – ecb 1 – cbc-ci 2 – cbc-atsc 3 – ofb	1 – in 2 – out	0 – enc 1 – dec				
5	3des	0 – ddd 1 – dde … 5 – ede 6 – eed 7 – eee	0 – ecb 1 – cbc-ci 2 – cbc-atsc 3 – ofb	1 – in 2 – out	0 – enc 1 – dec	-			
6	aes-128	-	0 – ecb 1 – cbc-ci 2 – cbc-atsc 3 – ofb	1 – in 2 – out	0 – enc 1 – dec				
7	aes-192	-	0 – ecb 1 – cbc-ci 2 – cbc-atsc 3 – ofb	1 – in 2 – out	0 – enc 1 – dec				
8	aes-256	-	0 – ecb 1 – cbc-ci 2 – cbc-atsc 3 – ofb	1 – in 2 – out	0 – enc 1 – dec				
9	gost	0x80 - ch_tbl	0 – ecb 1 – gamma 2 – gamma-fb	1 – in 2 – out	0 – enc 1 – dec				
10	gost-imit	0x80 - ch_tbl length 0 – 63	-	-					

KEY	
type	size (байт)
permutation	40
des	8
3des	24
aes-128	16
aes-192	24
aes-256	32
gost	32
gost-imit	32

IV	
type	size (байт)
des	8
3des	8
aes-128	16
aes-192	16
aes-256	16
gost	8
gost-imit	8

CUSTOM	
type	size (байт)
gost	64 – permutation table
gost-imit	64 – permutation table

1.3.9.2.2.3.1 Перестановка

Операция перестановки (permutation) выполняется в два этапа:

- Перестановка байтов в слове
- Перестановка битов в каждом байте

Перестановка битов или байтов выполняется по одной схеме: каждые 4 разряда кодируют номер положения бита или байта в новом слове.

Таблица 1.706 – Схема перестановки байтов в слове или битов в байте RMACE

Исходный порядок байтов / битов								
Номера байтов/битов	7	6	5	4	3	2	1	0
Порядок после перестановки								
Разряды, кодирующие номер байта/бита	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11: 8]	[7: 4]	[3: 0]

Пример (перестановка байтов):

Таблица перестановки: 0x01234567
Исходное слово: 0x0001020304050607
Результат: 0x0706050403020100

Формат таблицы перестановки:

Таблица 1.707 – Формат таблицы перестановки RMACE.

Номер строки таблицы	Байт 7 [63:56]	Байт 6 [55:48]	Байт 5 [47:40]	Байт 4 [39:32]	Байт 3 [31:24]	Байт 2 [23:16]	Байт 1 [15:8]	Байт 0 [7:0]
1	Перестановка байт				Резерв (0x0)			
2	Перестановка бит в байте 7				Перестановка бит в байте 6			
3	Перестановка бит в байте 5				Перестановка бит в байте 4			
4	Перестановка бит в байте 3				Перестановка бит в байте 2			
5	Перестановка бит в байте 1				Перестановка бит в байте 0			

Значение в поле mode заголовка конфигурационного дескриптора определяет будет ли выполнена только перестановка байт или же дополнительно будет выполняться и битовая перестановка в каждом байте.

Битовая перестановка применяется к слову, полученному посте перестановки байт.

Перестановка байт допустима только для данных кратных 8 байт, т.е. состоящих из целых слов. Перестановка байт в неполном слове (меньше 8 байт) приведет к некорректному результату.

Битовая перестановка допустима как для полных, так и для неполных слов. Однако, при применении битовой перестановки к неполному слову недопустимо изменять порядок байт, т.е. строка №1 таблицы должна иметь вид 0x7654321000000000.

1.3.9.2.2.3.2 Операция XOR

Блок RMACE позволяет выполнять операцию XOR над несколькими операндами (от 2 до 255).

Внутренний буфер RMACE имеет размер 4 кбайт, поэтому максимальный размер операнда для операции XOR составляет 4 кбайт. Для разграничения операндов используется поле Act0. Все дескрипторы, описывающие один операнд, кроме последнего, должны

Внутренний буфер RMACE имеет размер 4 кбайт, поэтому для выполнения операции XOR необходимо предварительно подготовить данные. Для RDMA нужно сформировать набор дескрипторов для каждого операнда. Группы дескрипторов для двух последовательных операндов должны иметь разные значение в поле Act0. Общий размер группы дескрипторов не должен превышать 4 кбайт.

Для того чтобы произвести операцию XOR над данными размером более 4 кбайт, необходимо разбить эти данные на части не более 4 кбайт и расположить эти часто последовательно в таблице дескрипторов. Ниже приведен пример, иллюстрирующий эту операцию.

Пример: d = a ^ b ^ c, где a = b = c = 6 кбайт.

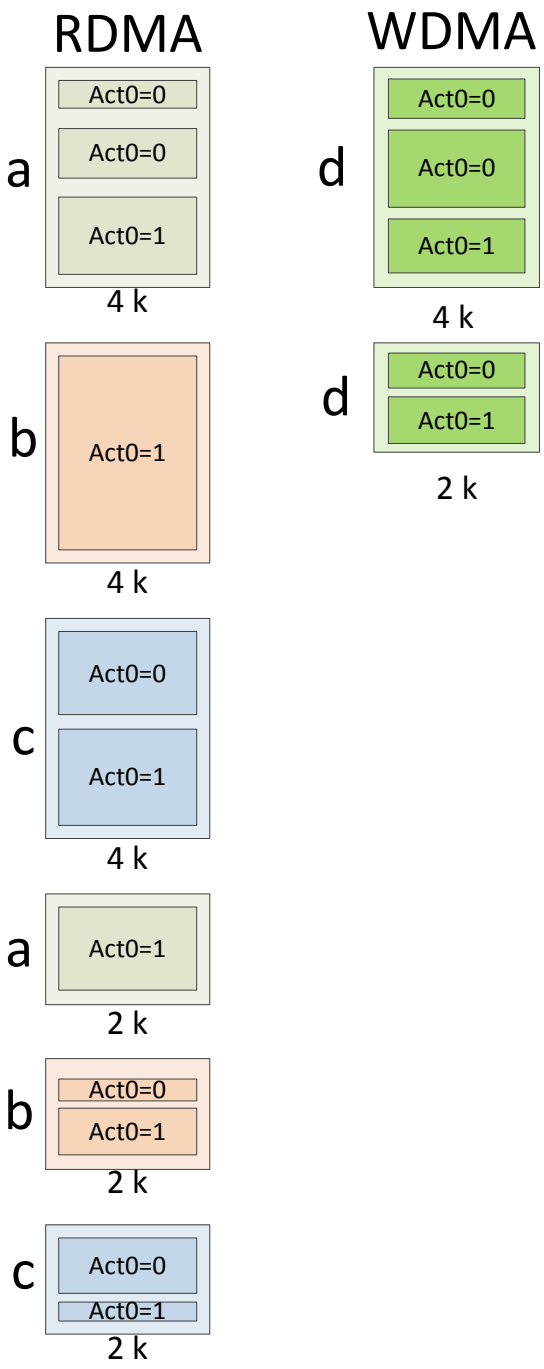


Рисунок 1.133 – Пример расположения данных в памяти для выполнения XOR над большими объемами данных.

1.3.9.2.2.4 Подготовка к работе

После включения блока RMACE или после выполнения программного сброса аппаратно происходит инициализация внутреннего буфера памяти, необходимого для выполнения

операции XOR (требуется 512 тактов, т.е. 5,12 мкс после снятия сигнала сброса), поэтому начало выполнения этой операции может быть задержано.

1.3.9.2.3 Контроллер интерфейса Space Wire (HSCB)

1.3.9.2.3.1 Описание

1.3.9.2.3.1.1 Технические характеристики

Скорость передачи данных от 2Мбит/с (минимальная скорость по стандарту) до 400 Мбит/с.

Интеграция в систему:

- Интерфейс APB slave для программирования регистров.
- Интерфейс AXI master с модулем DMA для передачи и приёма больших объемов данных.
- DMA с поддержкой внешней таблицы дескрипторов.
- Поддержка протокола RMAP.

1.3.9.2.3.1.2 Структурная схема

Структурная схема контроллера HSCB представлена на рисунке.

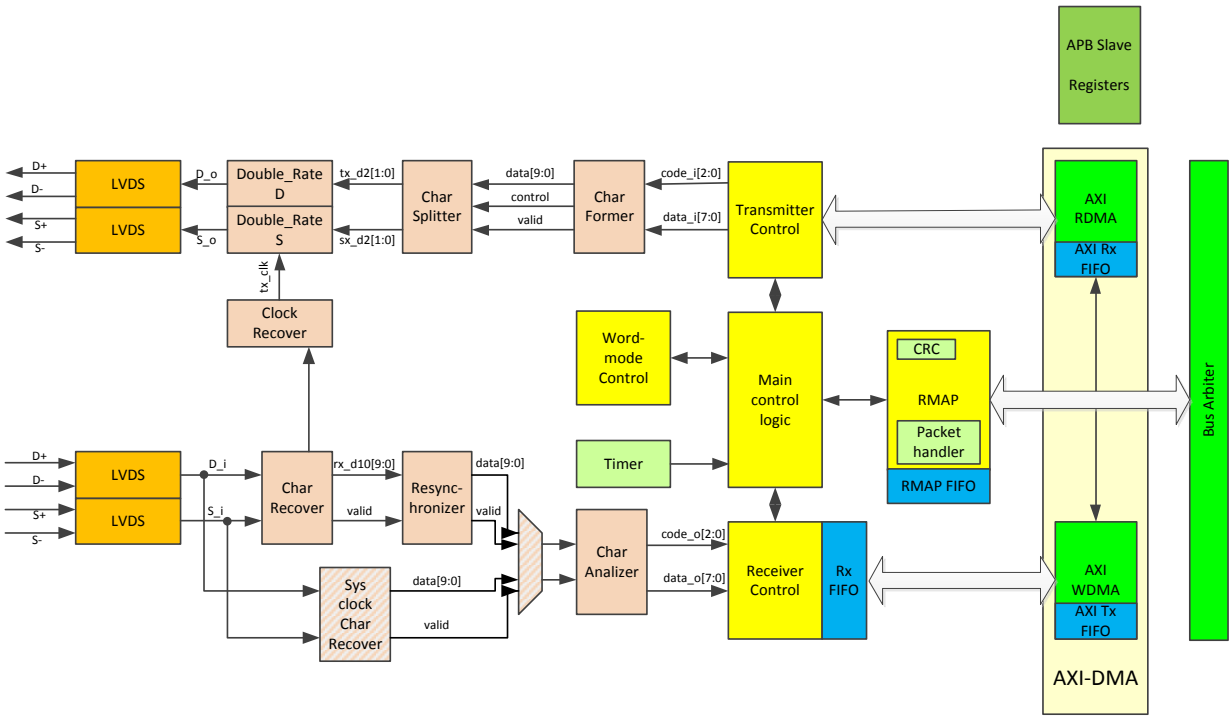


Рисунок 1.134 – Структурная схема контроллера HSCB

Контроллера HSCB включает в себя четыре подсистемы:

Управляющая логика выполняет анализ состояния соединения, арбитраж данных, и вырабатывает команды. Также управляющая логика реализует протокол RMAP предоставляющий доступ внешним устройствам к внутренней памяти системы.

Регистры управления через APB интерфейс.

Канал передачи, выполняющий преобразование данных и управляющих команд в DS-последовательности (Data-Strobe кодирование), и канал приёма, восстанавливающий данные и

команды из DS-последовательности. Приём и передача DS-последовательностей производится через LVDS буфера.

AXI интерфейс для доступа в памяти. Каналы записи и чтения AXI интерфейса реализуют прямой доступ к памяти, работают независимо и предполагают работу с внешней таблицей дескрипторов.

1.3.9.2.3.2 Программная модель

1.3.9.2.3.2.1 Карта регистров

Регистры контроллера HSCB разделены на несколько групп.

Таблица 1.708 – Карта регистров HSCB.

Название	Адрес	Доступ	Значение после сброса	Описание
Системные регистры				
ID	0x000	RO	0x43425348	Идентификатор устройства
VERSION	0x004	RO	0x00000101	Номер версии устройства
SW_RST	0x008	RW	0x0	Программный сброс (запись 1 сбрасывает устройство, когда снова устанавливается 0 – сброс выполнен)
SETTINGS	0x00C	RW	0x00000000	Регистр настроек
STATUS	0x010	RO	0x0	Регистр состояния устройства
IRQ_MASK	0x014	RW	0x0	Маска прерываний
TRANS_CLOCK	0x018	RW	0x00c81414	Коэффициент частоты передатчика
TIMINGS	0x01C	RW	0xc8a00500	Настройка временных параметров контроллера
Настройки Time-code				
INTERVAL	0x020	RW	0x7d0	Интервал между выдачей Time-кодов (в тактах системной частоты) По умолчанию установлен интервал 10 мс (для частоты 200 МГц)
TIME_CODE_INNER	0x024	RW	0x0	Внутреннее (системное) значение time-code (значимы только [7:0])
TIME_CODE_OUTER	0x028	RO	0x0	Значение time-code полученное от другого устройства [7:0] – new_code – полученное значение [15:8] – prev_code – предыдущее полученное значение
Настройка доступа по протоколу RMAP				
RMAP_PARAMS	0x04C	RW	0x00017799	Регистр настройки параметров RMAP
RMAP_STATUS	0x050	RO	0x0	Статус соединения RMAP
RMAP_C_STATUS	0x054	RO	0x0	Статус предыдущего соединения RMAP
Общие регистры блока AXI-DMA				
ADMA_ID	0x800	RO	0x414d4441	Идентификатор устройства
ADMA_REVISION	0x804	RO	0x11	Номер ревизии и количество каналов чтения и записи DMA
ADMA_SW_RST	0x808	RW	0x0	Программный сброс

Продолжение таблицы 1.709

Название	Адрес	Доступ	Значение после сброса	Описание
ADMA_CH_STATUS	0x80C	RO	0x0	Статус каналов DMA.
Регистры настройки шины AXI				
ARLEN	0x810	RW	0xF	Значение ARLEN для транзакций по AXI (значимы только младшие четыре бита – [3:0])
AWLEN	0x814	RW	0xF	Значение AWLEN для транзакций по AXI (значимы только младшие четыре бита – [3:0])
BRESP	0x818	RO	0x0	Значение BRESP для транзакций по AXI (значимы только младшие два бита – [1:0])
RAXI_ERR_ADDR	0x81C	RO	0x0	Адрес AXI-транзакции чтения, вызвавшей ошибку на шине AXI.
ARCACHE	0x820	RW	0x3	Значение ARCACHE для транзакций по AXI (значимы только младшие четыре бита – [3:0])
AWCACHE	0x824	RW	0x3	Значение AWCACHE для транзакций по AXI (значимы только младшие четыре бита – [3:0])
ARPROT	0x828	RW	0x2	Значение ARPROT для транзакций по AXI (значимы только младшие три бита – [2:0])
AWPROT	0x82c	RW	0x2	Значение AWPROT для транзакций по AXI (значимы только младшие три бита – [2:0])
ARLOCK	0x830	RW	0x0	Значение ARLOCK для транзакций по AXI (значимы только младшие два бита – [1:0])
AWLOCK	0x834	RW	0x0	Значение AWLOCK для транзакций по AXI (значимы только младшие два бита – [1:0])
ARBURST	0x838	RW	0x1	Значение ARBURST для транзакций по AXI (значимы только младшие два бита – [3:0])
AWBURST	0x83c	RW	0x1	Значение AWBURST для транзакций по AXI (значимы только младшие два бита – [1:0])
Регистры DMA канала чтения (RDMA)				
RDMA_SETTINGS	0x900	RW	0x0	Настройки DMA канала записи
RDMA_STATUS	0x904	RO	0x0	Статус канала записи
RDMA_SYS_ADDR	0x908	RW	0x0	Адрес таблицы дескрипторов для чтения из памяти
RDMA_TBL_SIZE	0x90C	RW	0x800	Размер таблицы дескрипторов (по умолчанию 2 кбайта)
RDMA_ADDR	0x910	RO	0x0	Текущий адрес
RDMA_LEN	0x914	RO	0x0	Длина блока данных
RDMA_DESC_ADDR	0x918	RW	0x0	Адрес текущего дескриптора

Продолжение таблицы 1.710

Название	Адрес	Доступ	Значение после сброса	Описание
Регистры DMA канала записи (WDMA)				
WDMA_SETTINGS	0xA00	RW	0x0	Настройки DMA канала чтения
WDMA_STATUS	0xA04	RO	0x0	Статус канала чтения
WDMA_SYS_ADDR	0xA08	RW	0x0	Адрес таблицы дескрипторов для чтения из памяти
WDMA_TBL_SIZE	0xA0C	RW	0x800	Размер таблицы дескрипторов (по умолчанию 2 кбайта)
WDMA_ADDR	0xA10	RO	0x0	Текущий адрес
WDMA_LEN	0xA14	RO	0x0	Длина блока данных
WDMA_DESC_ADDR	0xA18	RW	0x0	Адрес текущего дескриптора

1.3.9.2.3.2.1.1 SETTINGS (0x00C)

Таблица 1.711 – Поля регистра SETTINGS.

Биты	Поле	Описание
0	en_hscb	Включение Space Wire интерфейса 0 – интерфейс выключен, 1 – интерфейс включен
1	tx_endian	Порядок байт данных передающего канала 1 – big-endian, 0– little-endian
2	rx_endian	Порядок байт данных приёмного канала 1– big-endian, 0– little-endian
3	time_mode	Режим передачи Time-code 0 – приёмник (slave), 1 – передатчик (master)
4	-	Резерв, должно быть 0
5	loop_back	Включен режим петли (loopback) 0 –выключен, 1 –включен
6	en_rmap	Обработка RMAP 0 –выключен, 1 –включен
7	en_trim_clk	Автоподстройка частоты. 0 –выключен, 1 –включен
8	rx_fix_en*	Восстановление тактового сигнала приёмника из принимаемых данных 0 –выключен, 1 –включен
31	-	Резерв

* устойчивый приём возможен только если частоте приёмника в 4 или более раз превышает частоту передатчика

1.3.9.2.3.2.1.2 STATUS (0x010)

Таблица 1.712 – Поля регистра STATUS

Биты	Поле	Описание
0	disconnect_err	Потеря связи по каналу (либо не удалось подключиться, либо превышен интервал между символами)
1	parity_err	Ошибка бита четности
2	ESC_err	Получена ошибочная последовательность символов
3	-	Резерв
4	-	Резерв
5	Time-code	Получен time-code
6	Time-code_err	Полученный time-code не соответствует ожидаемому
7	TX_credit_err	Превышен допустимый размер буфера приёмника (число полученных FCT превышает допустимое по стандарту)
8	RX_credit_err	Переполнение приёмного буфера (количество принятых символов превышает значение, соответствующее числу отправленных FCT)
9	active_link	Соединение установлено
27:10	-	Резерв
29:28	state	Текущее состояние контроллера HSCB
31:30	-	Резерв

После чтения сбрасываются все биты регистра.

1.3.9.2.3.2.1.3 IRQ_MASK (0x014)

Таблица 1.713 – Поля регистра IRQ_MASK

Биты	Поле	Описание
0	disconnect_err	Маска прерывания потери связи по каналу
1	parity_err	Маска прерывания об ошибке бита четности
2	ESC_err	Маска прерывания ошибочной последовательность символов
3	RX_word	Маска прерывания о приёме слова при работе без DMA
4	TX_word	Маска прерывания о передаче слова при работе без DMA
5	Time-code	Маска прерывания о приёме time-code
6	Time-code_err	Маска прерывания об ошибке time-code
7	TX_credit_err	Маска прерывания о превышении допустимый размер буфера приёмника
8	RX_credit_err	Маска прерывания о переполнении приёмного буфера
9	active_link	Маска прерывания об успешном установлении соединения
31:10	-	Резерв

1.3.9.2.3.2.1.4 TRANS_CLOCK (0x018)

Таблица 1.714 – Поля регистра TRANS_CLOCK

Биты	Поле	Описание
7:0	freq	Коэффициент частоты передатчика (регулирует текущую скорость передачи)
15:8	init_freq	Коэффициент частоты передатчика, соответствующий скорости установления соединения – 10 Мбит/с. (коэффициент рассчитывается так же как и freq)
23:16	limit*	Минимально допустимая скорость передачи. (коэффициент рассчитывается так же как и freq)
31:24	-	Резерв

* Значение limit необходимо для подстройки частоты передачи при приёме на максимальной скорости (400 Мбит/с). Если данные принимаются, но в течение limit тактов системной частоты не регистрируется ни одного изменения восстановленного синхросигнала (регистратор работает на системной частоте), то считается, что приём ведется на скорости 400 Мбит/с, и передатчик начинает работать на максимально возможной (системной) частоте.

1.3.9.2.3.2.1.5 TIMINGS (0x01C)

Регистр настройки временных параметров обмена по интерфейсу Space Wire.

Значения счётчиков задаются в тактах. По умолчанию счётчики настроены на частоту работы блока 200 МГц и не требуют изменения.

Таблица 1.715 – Поля регистра TIMINGS

Биты	Поле	Описание
11:0	time_64	Счётчик 6,4 мкс (между ErrRST и ErrWAIT)
23:12	time_128	Счётчик 12,8 мкс (между ErrWAIT и CONNECT)
31:24	silence_time	Максимальное время между символами, при котором соединения сохраняется

1.3.9.2.3.2.1.6 RMAP_PARAMS (0x04C)

Регистр настройки параметров RMAP транзакций. В регистре содержится RMAP адрес устройства, ключ и флаг проверки ключа.

Таблица 1.716 – Поля регистра RMAP_PARAMS

Биты	Поле	Описание
7:0	terget_addr	Адрес в RMAP пакетах
15:8	key	Ключ для RMAP пакетов
16	en_key	Использовать ключ
19:17	-	Резерв
20	read_type	Буферизация при чтении из памяти 0 – буферизация выключена 1 – буферизация включена
31:21	-	Резерв

Изменение значения полей регистра рекомендуется только при выключенном HSCB.

1.3.9.2.3.2.1.7 RMAP_STATUS (0x050) и RMAP_PREV_STATUS (0x054)

Регистр RMA_C_STATUS содержит информацию о предыдущей транзакции. Таким образом по RMAP протоколу можно выяснить причины обрыва предыдущей транзакции.

Таблица 1.717 – Поля регистров RMAP_STATUS и RMAP_PREV_STATUS

Биты	Поле	Описание
7:0	rmap_status	Поле статуса в соответствии со стандартом
15:8	custom_status	Дополнительная информация об ошибках
31:16	-	резерв

Формат поля custom_status

Таблица 1.718 – Формат поля custom_status регистров RMAP_STATUS и RMAP_PREV_STATUS

Значение	Ошибка	Описание
0	OK	Транзакция прошла корректно
1	Chanel error	Ошибка в канале HSCB *
2	Receive Replay message	Получено сообщение RMAP, соответствующее ответу (уточнение в случае ошибки 1 *
3	AXI read error	Ошибка при выполнении AXI транзакции чтения из памяти *
4	AXI write error	Ошибка при выполнении AXI транзакции записи в память *
5	Too big address	Указанный в команде адрес не поддерживается (больше максимально допустимого в системе) **
6	Incorrect RMW length	Для RMW операции запрашивается не верная длина для указанного адреса **
7	Channel error after operation	Был разрыв соединения, но пакет RMAP был принят полностью и корректно, поэтому завершены операции записи по AXI (Write, Write-verify и Read-Modify-Write команды) *
8-255		резерв

* уточнение в случае ошибки 1 (General error) в поел rmap_status
** уточнение в случае ошибки 10 (RMAP command not implemented) в поел rmap_status

1.3.9.2.3.3 Общее описание работы контролера HSCB

1.3.9.2.3.3.1 Общие настройки и работа управляющей логики интерфейса HSCB

Общие настройки контроллера HSCB задаются в регистре SETTINGS.

Поле en_hscb отвечает за включение контроллера. При включении контроллер начинает работу в соответствии с диаграммой состояний, приведенной на рисунке.

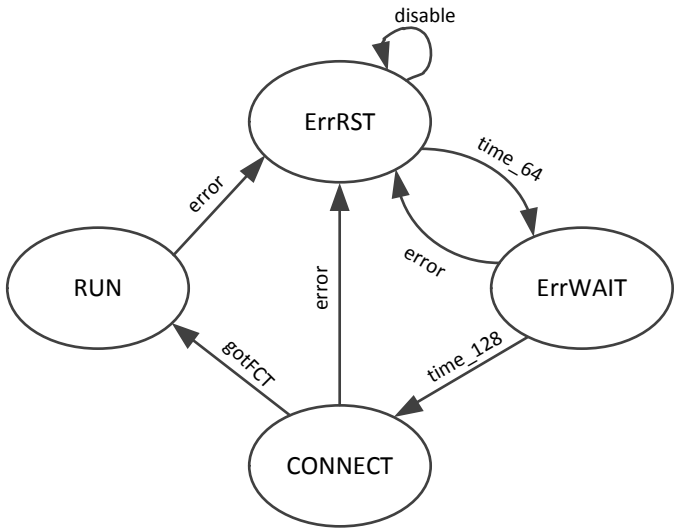


Рисунок 1.135 – Диаграмма состояний контроллера HSCB

При en_hscb = 0 контролера находится в выключенном состоянии ErrRST.

После включения (или перехода в состояние ErrRST из другог осостояния) запускается таймер, отсчитывающий 6,4 мкс. Предел таймера (в количестве тактов системной частоты) задается в поле time_64. Любой полученный в этом состоянии символ игнорируется.

По достижении значения time_64 контроллер переходит в состояние ErrWAIT, разрешает приём данных и запускает таймер, отсчитывающий 12,8 мкс. Предел таймера (в количестве тактов системной частоты) задается в поле time_128.

При приёме первого символа отличного от NULL или при превышении максимального интервала между символами (задаётся в поле silence_time регистра TIMINGS) контроллер переходит в состояние ErrRST и пытается заново установить соединение.

По достижении значения time_128 контроллер переходит в состояние CONNECT, включает передатчик и начинает посылать NULL.

Если первый принятый символ не NULL или в состоянии CONNECT принят символ отличный от NULL или FCT или превышении максимального интервала между символами, или в течение времени time_128 не получено ни одного символа, или возникла ошибка бита чётности контроллер переходит в состояние ErrRST и пытается заново установить соединение.

Если в состояниях ErrWAIT или CONNECT был получен символ NULL, контроллер начинает посылать символы FCT в соответствии с установленным режимом работы (поле word_mode регистра SETTINGS).

После приёма символа FCT контроллер переходит в состояние RUN. В этом состоянии доступны любые операции по интерфейсу HSCB.

При возникновении ошибки в работе интерфейса HSCB, контроллер переходит в состояние ErrRST и пытается заново установить соединение.

1.3.9.2.3.3.2 Дополнительные настройки

1.3.9.2.3.3.2.1 Скорость передачи данных

Контроллера HSCB позволяет изменять скорость передачи данных путём изменения частоты работы передающей логики.

Скорость передачи данных регулируется значением поля freq в регистре TRANS_CLOCK. По умолчанию это значение соответствует скорости передачи 10 Мбит/с – на этой скорости выполняется установка соединения.

Значение поля freq в регистре TRANS_CLOCK – коэффициент деления системной частоты. Значение freq задаёт длительность (в тактах системной частоты) стабильного состояния синхросигнала передающей логики. Передающая логика работает по фронту и по срезу синхросигнала, поэтому каждый новый бит по каналу HSCB будет передаваться каждые freq тактов (см. рис. Рисунок 1.136). Для установки максимальной скорости передачи (при системной частоте 200 МГц – 400 Мбит/с), в поле freq регистра TRANS_CLOCK необходимо записать 0.

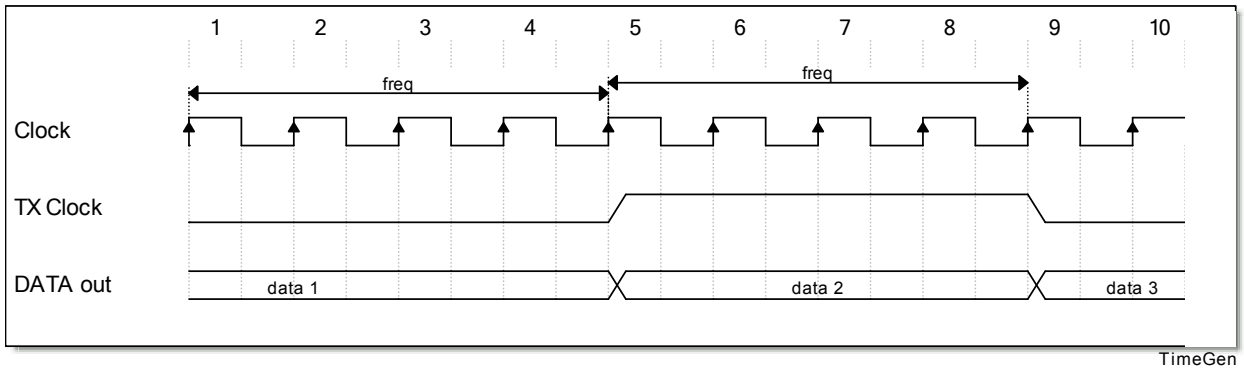


Рисунок 1.136 – Синхронизация и скорость передачи данных

Скорость передачи данных (под данными понимается любой символ протокола Space Wire) на основе коэффициента freq рассчитывается по формуле:

$$\boldsymbol{V}[\text{Мбит/с}] = \frac{\boldsymbol{F}[\text{МГц}]}{\boldsymbol{freq}[\text{такт}]},$$

где V – скорость передачи, F – системная частота.

Соответствие некоторых значений freq и скоростей передачи для системной частоты 200 МГц представлены в таблице.

Таблица 1.719 – Значения коэффициента частоты передатчика (freq) для различных скоростей передачи

Скорость передачи (Мбит/с)	**freq (тактов)**	**Частота передатчика (МГц)**
400	0	200
200	1	100
100	2	50
66,7	3	33,3
…	…	…
20	10	10
…	…	…
10	20	5
…	…	…
2	100	1

1.3.9.2.3.3.2.2 Автоподстройка скорости передачи данных

Контроллера HSCB позволяет выполнять автоподстройку скорости передачи данных в зависимости от скорости приёма. Для включения этой функции необходимо записать 1 в поле en_trim_clk.

При включённой автоподстройке контроллер на основании скорости приёма данных автоматически подбирает коэффициент деления системной частоты. Текущее значение этого коэффициента отображается в поле freq регистра TRANS_CLOCK.

При нестабильной скорости работы значение, прочитанное из поля freq регистра TRANS_CLOCK, может отличаться от реального, т.к. за время чтения может быть рассчитан новый коэффициент деления частоты.

В случае если соединение сохраняется, но коэффициент деления достигает значения, указанного в поле limit регистра TRANS_CLOCK, считается, что приём ведется на максимально доступной скорости 400 Мбит/с, и для передачи выставляется коэффициент 0, соответствующий максимальной скорости передачи.

1.3.9.2.3.3.2.3 Настройки порядка байт данных

Контроллера HSCB позволяет изменять порядок байт 8-байтовых слов при передаче или приёме по интерфейсу HSCB (передача идёт побайтно – один символ представляет собой один байт).

По умолчанию байты слова [63:0] передаются и принимаются как big-endian, т.е. [63:56], [55:48] … [7:0].

При записи 1 в поле tx_endian (для передачи) или rx_endian (для приёма) регистра SETTINGS порядок байт будет изменен на little-endian. Т.е. байты слова [63:0] будут отправлены или получены в следующем порядке: [7:0], [15:8] … [63:56].

1.3.9.2.3.3.2.4 Работа в режиме петли

Контроллера HSCB поддерживает отладочный режим – режим петли.

Для перехода в этот режим необходимо записать 1 в поле loop_back регистра SETTINGS.

В режиме петли сигнал D_o замыкается на D_i, а S_o – на S_i. Сигнал до LVDS буферов не доходит.

1.3.9.2.3.3.2.5 Работа с Time-code

Контроллера HSCB поддерживает механизм обмена символами Time-code.

Контроллера может выступать либо как приёмник, либо как источник этих символов.

При записи 1 в поле time_mode регистра SETTINGS контроллер будет генерировать Time-code последовательно с интервалом, определяемым значением регистра INTERVAL (число тактов системной частоты), начиная с значения заданного в регистра TIME_CODE_INNER.

При работе контроллера в режиме приёма Time-code. Принятый символ записывается в поле new_code регистра TIME_CODE_OUTER, а предыдущее значение Time-code отображается в поле prev_code. При получении символа Time-code устанавливается флаг Time-code в регистре STATUS и вырабатывается прерывание Time-code, если установлена соответствующая маска.

В случае если полученный Time-code не равен текущему значению Time-code увеличенному на 1, устанавливается флаг Time-code_err и выдается прерывание Time-code_err (если установлена соответствующая маска). Работа контроллера HSCB в этом случае не останавливается.

При получении первого Time-code после установления соединения, его значение может отличаться от значения в младшем байте регистра TIME_CODE_OUTER, поэтому будет установлен флаг Time-code_err и выдано прерывание Time-code_err (если установлена

соответствующая маска). Аналогичное поведение будет при получении повторно того же значения Time-code, что и предыдущее. Для некоторых применений получение двух одинаковых Time-code подряд не является ошибкой.

1.3.9.2.3.3.3 Передача данных

Для передачи данных используется интерфейс AXI master. Каналы чтения из памяти и записи в память работают независимо. Для уменьшения нагрузки на центральный процессор используется контроллер прямого доступа к памяти с поддержкой таблицы дескрипторов.

Дескриптор канала чтения описывает область памяти, содержащую полный пакет или часть пакета данных (если размера области памяти недостаточно для описания пакета). Формат дескриптора описан ниже:

Таблица 1.720 – Формат дескриптора канала чтения из памяти.

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Length		[31:6]	Длина блока данных (до 64M-1)
Атрибуты	Act2	[5]	Тип дескриптора 00 compl Дескриптор выполнен 01 compl Дескриптор выполнен 10 tran Передача данных 11 compl Дескриптор выполнен Должно быть 10
	Act1	[4]	
	Act0	[3]	1 – последний адрес дескриптора является концом пакета 0 – дескриптор содержит не полный пакет, продолжение пакета будет следующем дескрипторе
	Int	[2]	1 - Выработка прерывания по завершении выполнения
	-	[1]	Зарезервировано
	Valid	[0]	Поле анализируется если Act0 = 1, если Act0 = 0, значение не анализируется 1 – Данные верные (в конце пакета устанавливается код EOP) 0 – Данные не верные (в конце пакета устанавливается код EEP)

После передачи всех данных, соответствующих дескриптору, происходит модификация дескриптора в таблице – дескриптор помечается как выполненный. Модифицированный дескриптор канала чтения имеет вид:

Таблица 1.721 – Формат дескриптора канала чтения из памяти после модификации.

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Length		[31:6]	Длина блока данных (количество переданных данных)
Атрибуты	Act2	[5]	01
	Act1	[4]	
	Act0	[3]	Значение поля не меняется
	Int	[2]	0
	Err	[1]	Ошибка соединения 1 – возникла ошибка, приведшая к разрыву соединения 0 – передача прошла успешно
	Valid	[0]	Значение поля не меняется

Дескриптор канала записи описывает область памяти, в которую будут помещаться принятые данные. Формат дескриптора описан ниже:

Таблица 1.722 – Формат дескриптора канала записи в память

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Length		[31:6]	Длина блока данных (до 64M-1)
Атрибуты	Act2	[5]	Тип дескриптора 00 compl Дескриптор выполнен 01 compl Дескриптор выполнен 10 tran Передача данных 11 compl Дескриптор выполнен Должно быть 10
	Act1	[4]	
	Act0	[3]	Зарезервировано
	Int	[2]	1 - Выработка прерывания по завершении выполнения
	-	[1]	Зарезервировано
	Valid	[0]	Зарезервировано

После приёма символа конца пакета или по достижении конца области памяти описанной дескриптором, производится модификация дескриптора.

Модифицированный дескриптор канала записи имеет вид:

Таблица 1.723 – Формат дескриптора канала записи в память после модификации

Поле		Биты	Описание
Address		[63:32]	32-битный стартовый адрес
Length		[31:6]	Длина блока данных
Атрибуты	Act2	[5]	01
	Act1	[4]	
	Act0	[3]	1 – дескриптор содержит полный пакет 0 – дескриптор содержит не полный пакет, продолжение пакета будет в следующем дескрипторе
	Int	[2]	1 – в пакете присутствует байт с ошибкой бита чётности 0 – бит чётности во всех байтах пакета верен
	Err	[1]	Ошибка соединения 1 – возникла ошибка, приведшая к разрыву соединения 0 – приём прошел успешно
	Valid	[0]	1 – в конце пакета устанавливается код EOP 0 – в конце пакета устанавливается код EEP

1.3.9.2.3.4 Работа по протоколу RMAP

Контроллер HSCB способен выполнять операции в соответствии с протоколом RMAP.

Доступ по протоколу RMAP может быть отключен в регистре SETTINGS. По умолчанию доступ по протоколу RMAP допустим сразу после сброса контроллера и установки соединения (соединение устанавливается автоматически после сброса).

Поддерживаются все режимы, описанные в спецификации RMAP со следующими ограничениями:

Максимальный размер поля данных для сообщения типа Write verify составляет 256 байт.

Запись по одному адресу (no inc) допустима только для адресов, выровненных по 4 байта.

Чтение из одного адреса (no inc) доступно только для адресов и длин данных, выровненных по 4.

Read Modify Write команда должна укладываться в одиночную транзакцию AXI (максимум 4 байта с адресов выровненных по 4 байта). Т.е. с адреса выровненного по 4 допустимо чтение 1, 2, 3 или 4 байт, с адреса 4n + 1 – 1, 2 или 3 байт, с адреса 4n + 2 – 1 или 2 байта и с адреса 4n + 3 – только 1 байт.

Кроме этого, реализация поддержки протокола RMAP в контроллере HSCB позволяет выполнять буферизированные транзакции чтения из памяти, т.е. данные сначала будут считаны в отдельный буфер и только после успешного чтения всего необходимого массива данных, они будут переданы в ответном сообщении. Данный механизм позволяет предотвратить ошибки, связанные с ошибками шины AXI. Для включения режима буферизации необходимо установить флаг buf_read в регистре RMAP_PARAMS, при этом максимальный размер поля данных для RMAP команды не должен превышать 256 байт. По умолчанию, буферизация транзакция чтения отключена.

При использовании буферизированного чтения в случае возникновения ошибки чтения по AXI будет выслан ответ со статусом 10. Если буферизация при чтении не используется, то размер блока данных для чтения ограничен только полем data length RMAP пакета, однако в случае ошибки AXI транзакции чтения будет выставлен признак EEP в конце пакета.

1.3.10 Подсистема суперскоростных интерфейсов XHSIF0 и XHSIF1

1.3.10.1 Описание подсистем XHSIF0 и XHSIF1

1.3.10.1.1 Структурная схема и состав блоков подсистем XHSIF0 и XHSIF1

Микросхема включает две идентичные подсистемы суперскоростных интерфейсов XHSIF0 и XHSIF1. Ниже приведены их структуры.

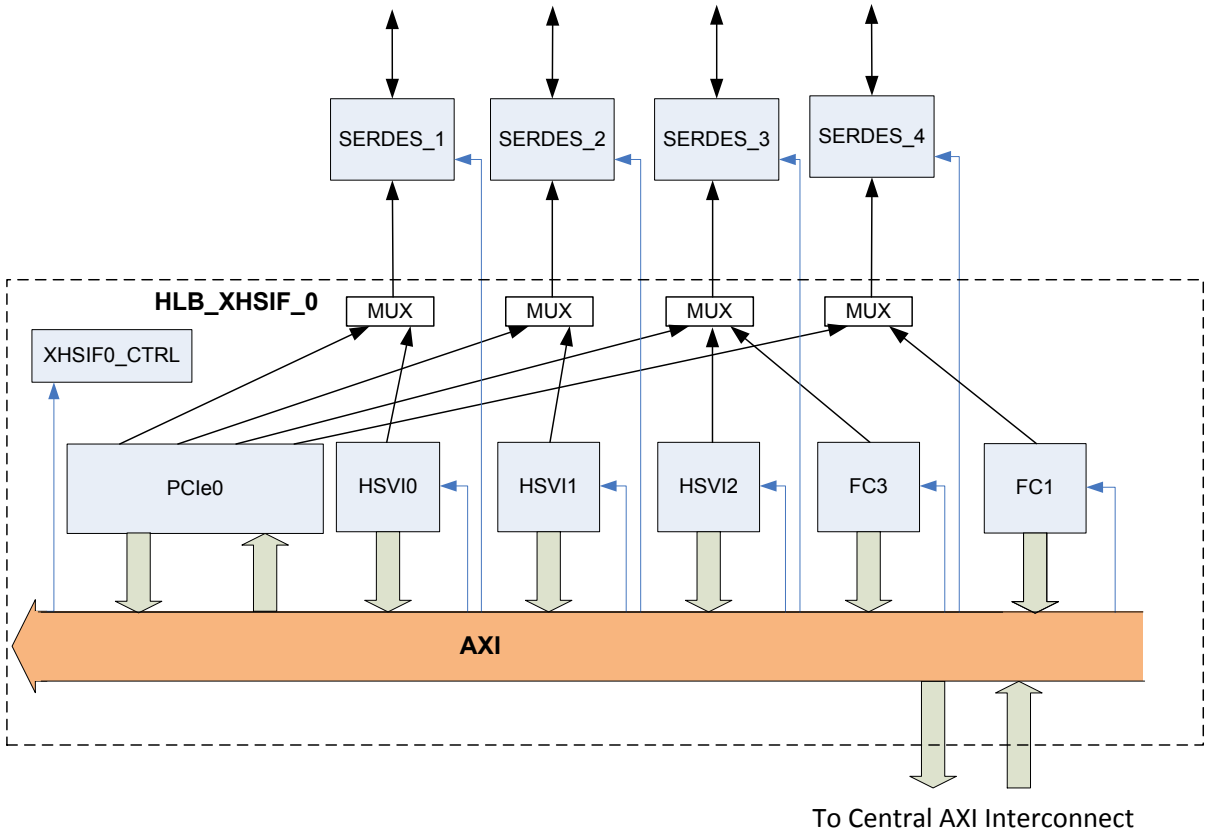


Рисунок 1.137 - Структурная схема подсистемы XHSIF_0

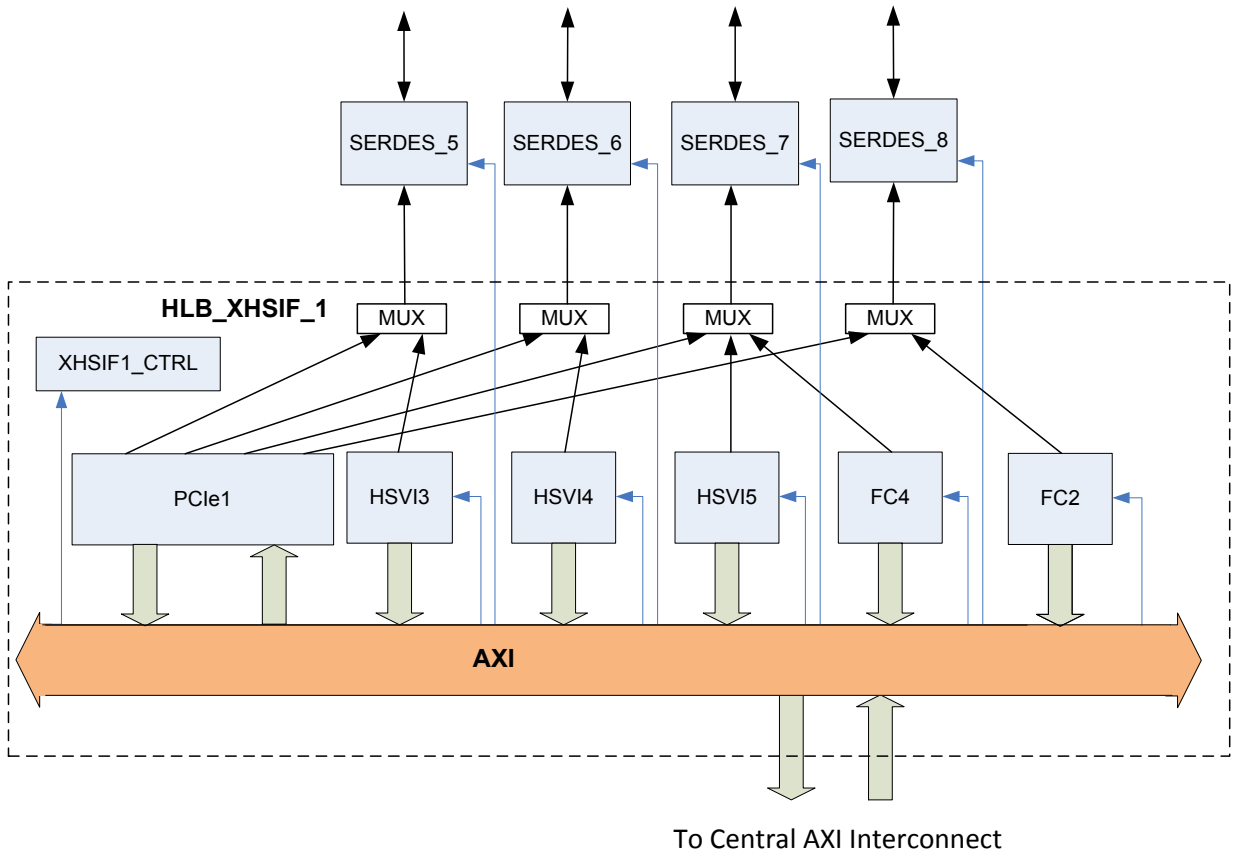


Рисунок 1.138 - Структурная схема подсистемы XHSIF_1

В дальнейшем описании подсистем XHSIF0 и XHSIF1 (XHSIFx) нумерация контролеров HSVI и FC приведена для XHSIF0, а в скобках нумерация соответствующих контроллеров HSVI и FC для XHSIF1 (Пример: HSVI0(3)).

Каждая из подсистем XHSIF0 и XHSIF1 состоит из следующих блоков:

- Контроллер интерфейса PCI Express (PCIe0(1), описание приведено в разделе 1.3.10.2.1)
- Три контроллера интерфейса ARINC-818 (HSVI0(3), HSVI1(4), HSVI2(5)
- Два контроллера интерфейса FiberChannel (FC1(2), FC3(4), описание приведено в разделе 1.3.10.2.3)
- Блок управляющих регистров XHSIFx_CTRL

1.3.10.1.2 Адресация блоков подсистемы XHSIF0

Таблица 1.724 - Карта адресов подсистемы XHSIF0

Блок	**Адрес на шине APB**
PCIe0 x4	0x3C000000 – 0x3C000FFF
HSVI0	0x3C001000 – 0x3C001FFF
HSVI1	0x3C002000 – 0x3C002FFF
HSVI2/FC3	0x3C003000 – 0x3C003FFF
FC1	0x3C004000 – 0x3C004FFF
XHSIF0_CTRL	0x3C005000 – 0x3C005FFF
SERDES1	0x3CB00000 – 0x3CB0FFFF
SERDES2	0x3CB10000 – 0x3CB1FFFF
SERDES3	0x3CB20000 – 0x3CB2FFFF
SERDES4	0x3CB30000 – 0x3CB3FFFF

1.3.10.1.3 Адресация блоков подсистемы XHSIF1

Таблица 1.725 - Карта адресов подсистемы XHSIF1

Блок	Адрес на шине APB
PCIe1 x4	0x3C010000 – 0x3C010FFF
HSVI3	0x3C011000 – 0x3C011FFF
HSVI4	0x3C012000 – 0x3C012FFF
HSVI5/FC4	0x3C013000 – 0x3C013FFF
FC2	0x3C014000 – 0x3C014FFF
XHSIF1_CTRL	0x3C015000 – 0x3C015FFF
SERDES5	0x3CB40000 – 0x3CB4FFFF
SERDES6	0x3CB50000 – 0x3CB5FFFF
SERDES7	0x3CB60000 – 0x3CB6FFFF
SERDES8	0x3CB70000 – 0x3CB7FFFF

1.3.10.1.4 Блок управляющих регистров XHSIFx_CTRL

Блок управляющих регистров XHSIFx_CTRL подсистемы XHSIFx микросхемы предназначен для настройки работы подсистемы. Блок позволяет управлять следующими функциями:

- Настройка конфигурации контроллера PCIe;
- Мониторинг статуса контроллера PCIe;
- Настройка мультиплексирования SERDES PHY;
- Мониторинг статуса SERDES PHY;
- Расширение адресного пространства контроллеров PCIe, HSVI, FC;
- Управление сигналами arcache, awcache интерфейсов AXI Master контроллеров HSVI и FC;
- Блок имеет защиту от случайной записи.

Таблица 1.726 - Карта регистров блока XHSIFx_CTRL

Название	Адрес	Доступ	Значение после сброса	Описание
Регистры управления контроллером PCI Express				
PCIe_setup0	0x000	RW	0x0	Регистр конфигурации контроллера PCIe
PCIe_setup1	0x004	RW	0xffff	Регистр конфигурации контроллера PCIe
PCIe_setup2	0x008	RW	0x0	Регистр конфигурации контроллера PCIe
PCIe_setup3	0x00C	RW	0x0	Регистр конфигурации контроллера PCIe
PCIe_setup4	0x010	RW	0x0	Регистр конфигурации контроллера PCIe
PCIe_setup5	0x014	RW	0x0	Регистр конфигурации контроллера PCIe
PCIe_setup6	0x018	RW	0x0	Регистр конфигурации контроллера PCIe
PCIe_status0	0x01C	RO	0x0	Регистр статуса контроллера PCIe
PCIe_status1	0x020	RO	0x0	Регистр статуса контроллера PCIe
PCIe_status2	0x024	RO	0x0	Регистр статуса контроллера PCIe

Продолжение таблицы 1.727

Название	Адрес	Доступ	Значение после сброса	Описание
PCIe_status3	0x028	RO	0x0	Регистр статуса контроллера PCIe
PCIe_status4	0x02C	RO	0x0	Регистр статуса контроллера PCIe
Регистры управления мультиплексорами SERDES PHY				
serdes_mux_control0	0x030	RW	0x80000000	Регистр управления мультиплексорами SERDES PHY
serdes_mux_control1	0x034	RW	0x1110000f	Регистр управления мультиплексорами SERDES PHY
serdes_mux_control2	0x044	RW	0x00001000	Регистр управления мультиплексорами SERDES PHY
serdes_mux_status0	0x03C	RO	0x0	Регистр статуса интерфейсов SERDES PHY
Регистры управления контроллерами HSVI				
hsvi_axi_addr_control	0x038	RW	0x0	Регистр управления старшими разрядами адреса интерфейсов AXI Master контроллеров HSVI и FC
hsvi_axi_axcache_control	0x040	RW	0x33333333	Регистр управления сигналами arcache, awcache интерфейсов AXI Master контроллеров HSVI и FC
Регистры общего назначения				
write_enable	0xf00	RW	0x0	Защита от случайной записи регистров XHSIFx_CTRL

1.3.10.1.4.1 Поля регистра PCIe_setup0

Таблица 1.728 - Поля регистра PCIe_setup0

Биты	Знач. по умолч.	Описание
5:0	6'h0	Период частоты работы ядра PCI Express в наносекундах, по умолчанию 4нс (250 МГц).
6	0	Управление сигналом p_ltssm_disable. Установка в 1 инициирует принудительную задержку в установлении связи, не рекомендуется использовать (подробности стр.130 документа Expresso_3.0_Core_User_Guide_1_33.pdf).
7	0	Управление сигналом set_attention_button_pressed (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 190 [559:544]).
8	0	Управление сигналом set_power_fault_detected (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 190 [559:544]).
9	0	Управление сигналом set_command_completed (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 190 [559:544]).
10	0	Управление сигналом state_of_mrl_sensor (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 190 [559:544]).
11	0	Управление сигналом state_of_presence_detect (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 190 [559:544]).
12	0	Управление сигналом state_of_em_interlock (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 190 [559:544]).
18:14	5'h00	Управление сигналом security_internal (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 32), 0 – secure, 1 – non-secure
26:19	8'h00	Управление сигналом security_egress (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 32), 0 – secure, 1 – non-secure
31:4	-	reserve

1.3.10.1.4.2 Поля регистра PCIe_setup1

Таблица 1.729 - Поля регистра PCIe_setup1

Биты	Знач. по умолч.	Описание
7:0	8'hff	Управление сигналом security_ingress (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 32), 0 – secure, 1 – non-secure
15:8	8'hff	Управление сигналом security_dma (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 32), 0 – secure, 1 – non-secure
31:16	-	reserve

1.3.10.1.4.3 Поля регистра PCIe_setup2

Таблица 1.730 - Поля регистра PCIe_setup2

Биты	Знач. по умолч.	Описание
0	0	1 – Включение аппаратной поддержки сохранения состояния контроллера PCIe при отключении тактовой частоты и питания ядра (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 223).
1	0	Управление сигналом pm_l2_enter_ack (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 218).
2	0	Управление сигналом pm_d3cold_n_pme_assert (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 218).
3	0	Управление сигналом pm_aux_power_detected (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 218).
8:4	5'h0	Управление сигналом pm_dpa_status (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 218).
31:9	-	reserve

1.3.10.1.4.4 Поля регистра PCIe_setup3

Таблица 1.731 - Поля регистра PCIe_setup3

Биты	Знач. по умолч.	Описание
0	0	Управление сигналом pm_ltr_nosnoop_req.
1	0	Управление сигналом pm_ltr_snoop_req.
14:2	13'h0	Управление сигналом pm_ltr_nosnoop.
27:15	13'h0	Управление сигналом pm_ltr_snoop.
28	0	Управление сигналом pm_ltr_msg_send.
31:29	-	reserve

1.3.10.1.4.5 Поля регистра PCIe_setup4

Таблица 1.732 - Поля регистра PCIe_setup4

Биты	Знач. по умолч.	Описание
31:0	32'h0	Управление сигналом pm_pb_data_reg_rd (подробности Expresso_3.0_Core_User_Guide_1_33.pdf стр. 222).

1.3.10.1.4.6 Поля регистра PCIe_setup5

Таблица 1.733 - Поля регистра PCIe_setup5

Биты	Знач. по умолч.	Описание
27:0	27'h0	Управление старшими битами адресной шины AXI Slave канала записи (сигнал s_awaddr[63:36]).
31:29	-	reserve

1.3.10.1.4.7 Поля регистра PCIe_setup6

Таблица 1.734 - Поля регистра PCIe_setup6

Биты	Знач. по умолч.	Описание
27:0	32’h0	Управление старшими битами адресной шины AXI Slave канала чтения (сигнал s_araddr[63:36]).
31:28	-	reserve

1.3.10.1.4.8 Поля регистра serdes_mux_control0

Таблица 1.735 - Поля регистра serdes_mux_control0

Биты	Знач. по умолч.	Описание
0	0	0 – на интерфейсе phy0 работает контроллер PCI Express 1 – на интерфейсе phy0 работает контроллер HSVI0(3)
1	0	0 – на интерфейсе phy1 работает контроллер PCI Express 1 – на интерфейсе phy1 работает контроллер HSVI1(4)
2	0	0 – на интерфейсе phy2 работает контроллер PCI Express 1 – на интерфейсе phy2 работает контроллер HSVI2(5) или FC3(4) (выбор осуществляется дополнительным битом регистра serdes_mux_control2[12])
3	0	0 – на интерфейсе phy3 работает контроллер PCI Express 1 – на интерфейсе phy3 работает контроллер FC1(2)
5:4	2’h0	Выбор источника тактового сигнала для контроллера PCI Express: 00 – сигнал phy0_soc_clk_i[1] 01 – сигнал phy1_soc_clk_i[1] 10 – сигнал phy2_soc_clk_i[1] 11 – сигнал phy3_soc_clk_i[1]
6	0	0 – в режимах работы контроллеров HSVI и FC на SERDES подается 0 в качестве данных 1 – в режимах работы контроллеров HSVI и FC на SERDES подаются реальные данные
8:7	2’h0	Значение из данного поля присваивается сигналу phy0_pipe_pd_o в случае, если на интерфейсе phy0 работает контроллер HSVI0(3) (serdes_mux_control0[0]=1)
10:9	2’h0	Значение из данного поля присваивается сигналу phy1_pipe_pd_o в случае, если на интерфейсе phy1 работает контроллер HSVI1(4) (serdes_mux_control0[1]=1)
12:11	2’h0	Значение из данного поля присваивается сигналу phy2_pipe_pd_o в случае, если на интерфейсе phy2 работает контроллер HSVI2(5) или FC3(4) (serdes_mux_control0[2]=1)
14:13	2’h0	Значение из данного поля присваивается сигналу phy3_pipe_pd_o в случае, если на интерфейсе phy3 работает контроллер FC1(2) (serdes_mux_control0[3]=1)
16:15	2’h0	Значение из данного поля присваивается сигналу phy0_pipe_rate_o в случае, если на интерфейсе phy0 работает контроллер HSVI0(3) (serdes_mux_control0[0]=1)
18:17	2’h0	Значение из данного поля присваивается сигналу phy1_pipe_rate_o в случае, если на интерфейсе phy1 работает контроллер HSVI1(4) (serdes_mux_control0[1]=1)
20:19	2’h0	Значение из данного поля присваивается сигналу phy2_pipe_rate_o в случае, если на интерфейсе phy2 работает контроллер HSVI2(5) или FC3(4) (serdes_mux_control0[2]=1)
22:21	2’h0	Значение из данного поля присваивается сигналу phy3_pipe_rate_o в случае, если на интерфейсе phy3 работает контроллер FC1(2) (serdes_mux_control0[3]=1)
24:23	2’h0	Значение из данного поля присваивается сигналу phy0_pipe_databuswidth_o в случае, если на интерфейсе phy0 работает контроллер HSVI0(3) (serdes_mux_control0[0]=1)
26:25	2’h0	Значение из данного поля присваивается сигналу phy1_pipe_databuswidth_o в случае, если на интерфейсе phy1 работает контроллер HSVI1(4) (serdes_mux_control0[1]=1)
28:27	2’h0	Значение из данного поля присваивается сигналу phy2_pipe_databuswidth_o в случае, если на интерфейсе phy2 работает контроллер HSVI2(5) или FC3(4) (serdes_mux_control0[2]=1)
30:29	2’h0	Значение из данного поля присваивается сигналу phy3_pipe_databuswidth_o в случае, если на интерфейсе phy3 работает контроллер FC1(2) (serdes_mux_control0[3]=1)
31	1	Значение из данного поля присваивается сигналу phy3_pipe_rxterm_o

1.3.10.1.4.9 Поля регистра serdes_mux_control1

Таблица 1.736 - Поля регистра serdes_mux_control1

Биты	Знач. по умолч.	Описание
0	1	Программный сброс интерфейса phy0 (подается одновременно на сигналы phy0_por_n_o и phy0_pipe_rstn_o), активный уровень – высокий, снимается записью 0 в данный регистр
1	1	Программный сброс интерфейса phy1 (подается одновременно на сигналы phy1_por_n_o и phy1_pipe_rstn_o), активный уровень – высокий, снимается записью 0 в данный регистр
2	1	Программный сброс интерфейса phy2 (подается одновременно на сигналы phy2_por_n_o и phy2_pipe_rstn_o), активный уровень – высокий, снимается записью 0 в данный регистр
3	1	Программный сброс интерфейса phy3 (подается одновременно на сигналы phy3_por_n_o и phy3_pipe_rstn_o), активный уровень – высокий, снимается записью 0 в данный регистр
4	0	Программный сброс интерфейса PIPE контроллера PCI Express, активный уровень – высокий, снимается записью 0 в данный регистр
5	0	Значение из данного поля присваивается сигналу phy0_ pipe_phymode_o[0] в случае, если на интерфейсе phy0 работает контроллер HSVI0(3) (serdes_mux_control0[0]=1)
6	0	Значение из данного поля присваивается сигналу phy1_ pipe_phymode_o[0] в случае, если на интерфейсе phy1 работает контроллер HSVI1(4) (serdes_mux_control0[1]=1)
7	0	Значение из данного поля присваивается сигналу phy2_ pipe_phymode_o[0] в случае, если на интерфейсе phy2 работает контроллер HSVI2(5) или FC3(4) (serdes_mux_control0[2]=1)
8	0	Значение из данного поля присваивается сигналу phy3_ pipe_phymode_o[0] в случае, если на интерфейсе phy3 работает контроллер FC1(2) (serdes_mux_control0[3]=1)
9	0	Значение из данного поля присваивается сигналу phy0_ pipe_txdetectrx_lb_o в случае, если на интерфейсе phy0 работает контроллер HSVI0(3) (serdes_mux_control0[0]=1)
10	0	Значение из данного поля присваивается сигналу phy1_ pipe_txdetectrx_lb_o в случае, если на интерфейсе phy1 работает контроллер HSVI1(4) (serdes_mux_control0[1]=1)
11	0	Значение из данного поля присваивается сигналу phy2_ pipe_txdetectrx_lb_o в случае, если на интерфейсе phy2 работает контроллер HSVI2(5) или FC3(4) (serdes_mux_control0[2]=1)
12	0	Значение из данного поля присваивается сигналу phy3_ pipe_txdetectrx_lb_o в случае, если на интерфейсе phy3 работает контроллер FC1(2) (serdes_mux_control0[3]=1)
13	0	Значение из данного поля присваивается сигналу phy0_ pipe_rxpolarity_o в случае, если на интерфейсе phy0 работает контроллер HSVI0(3) (serdes_mux_control0[0]=1)
14	0	Значение из данного поля присваивается сигналу phy1_ pipe_rxpolarity_o в случае, если на интерфейсе phy1 работает контроллер HSVI1(4) (serdes_mux_control0[1]=1)
15	0	Значение из данного поля присваивается сигналу phy2_ pipe_rxpolarity_o в случае, если на интерфейсе phy2 работает контроллер HSVI2(5) или FC3(4) (serdes_mux_control0[2]=1)
16	0	Значение из данного поля присваивается сигналу phy3_ pipe_rxpolarity_o в случае, если на интерфейсе phy3 работает контроллер FC1(2) (serdes_mux_control0[3]=1)
17	0	Значение из данного поля присваивается сигналу phy0_pipe_ebuffmode_o
18	0	Значение из данного поля присваивается сигналу phy0_pipe_txoneszeros_o
19	0	Значение из данного поля присваивается сигналу phy0_pipe_rxeqtraining_o
20	1	Значение из данного поля присваивается сигналу phy0_pipe_rxterm_o
21	0	Значение из данного поля присваивается сигналу phy1_pipe_ebuffmode_o
22	0	Значение из данного поля присваивается сигналу phy1_pipe_txoneszeros_o
23	0	Значение из данного поля присваивается сигналу phy1_pipe_rxeqtraining_o
24	1	Значение из данного поля присваивается сигналу phy1_pipe_rxterm_o
25	0	Значение из данного поля присваивается сигналу phy2_pipe_ebuffmode_o
26	0	Значение из данного поля присваивается сигналу phy2_pipe_txoneszeros_o
27	0	Значение из данного поля присваивается сигналу phy2_pipe_rxeqtraining_o
28	1	Значение из данного поля присваивается сигналу phy2_pipe_rxterm_o
29	0	Значение из данного поля присваивается сигналу phy3_pipe_ebuffmode_o
30	0	Значение из данного поля присваивается сигналу phy3_pipe_txoneszeros_o
31	0	Значение из данного поля присваивается сигналу phy3_pipe_rxeqtraining_o

1.3.10.1.4.10 Поля регистра serdes_mux_control2

Таблица 1.737 - Поля регистра serdes_mux_contro2

Биты	Знач. по умолч.	Описание
3:0	4'b0	Ручное управление сигналами power-on reset блоков SERDES PHY: [0] PHY линии 0: 1 – сброс активен, 0 – не активен [1] PHY линии 1: 1 – сброс активен, 0 – не активен [2] PHY линии 2: 1 – сброс активен, 0 – не активен [3] PHY линии 3: 1 – сброс активен, 0 – не активен
11:4	8'b0	Управление мультиплексорами опорной частоты (reference_clock) блоков SERDES PHY: [5:4] PHY линии 0: 00 – частота с внешнего входа REFCLK0 01 – частота с внешнего входа REFCLK1 10 и 11 – 100МГц из внутреннего источника [7:6] Аналогично для PHY линии 1 [9:8] Аналогично для PHY линии 2 [11:10] Аналогично для PHY линии 3
12	1'h1	0 – используется контроллер FC3(4) (при serdes_mux_control0[2]=1) 1 – используется контроллер HSVI2(5) (при serdes_mux_control0[2]=1)
31:13	-	reserve

1.3.10.1.4.11 Поля регистра serdes_mux_status0

Таблица 1.738 - Поля регистра serdes_mux_status0

Биты	Знач. по умолч.	Описание
0	-	Состояние сигнала phy0_pipe_rxdatak_i[1]
3:1	-	Состояние сигнала phy0_pipe_rxstatus_i
4	-	Состояние сигнала phy0_pipe_phystatus_i
5	-	Состояние сигнала phy0_pipe_rxelecidle_i
6	-	Состояние сигнала phy1_pipe_rxdatak_i[1]
9:7	-	Состояние сигнала phy1_pipe_rxstatus_i
10	-	Состояние сигнала phy1_pipe_phystatus_i
11	-	Состояние сигнала phy1_pipe_rxelecidle_i
12	-	Состояние сигнала phy2_pipe_rxdatak_i[1]
15:13	-	Состояние сигнала phy2_pipe_rxstatus_i
16	-	Состояние сигнала phy2_pipe_phystatus_i
17	-	Состояние сигнала phy2_pipe_rxelecidle_i
18	-	Состояние сигнала phy3_pipe_rxdatak_i[1]
21:19	-	Состояние сигнала phy3_pipe_rxstatus_i
22	-	Состояние сигнала phy3_pipe_phystatus_i
23	-	Состояние сигнала phy3_pipe_rxelecidle_i
31:13	-	reserve

1.3.10.1.4.12 Поля регистра hsvi_axi_addr_control

Таблица 1.739 Поля регистра hsvi_axi_addr_control

Биты	Знач. по умолч.	Описание
3:0	4'h0	Значение из данного поля присваивается сигналу awaddr [35:32] контроллера HSVI0(3)
7:4	4'h0	Значение из данного поля присваивается сигналу araddr[35:32] контроллера HSVI0(3)
11:8	4'h0	Значение из данного поля присваивается сигналу awaddr[35:32] контроллера HSVI1(4)
15:12	4'h0	Значение из данного поля присваивается сигналу araddr[35:32] контроллера HSVI1(4)

Продолжение таблицы 1.740

Биты	Знач. по умолч.	Описание
19:16	4’h0	Значение из данного поля присваивается сигналу awaddr[35:32] контроллеров HSVI2(5) или FC3(4)
23:20	4’h0	Значение из данного поля присваивается сигналу araddr[35:32] контроллеров HSVI2(5) или FC3(4)
27:24	4’h0	Значение из данного поля присваивается сигналу awaddr[35:32] контроллера FC1(2)
31:28	4’h0	Значение из данного поля присваивается сигналу araddr[35:32] контроллера FC1(2)

1.3.10.1.4.13 Поля регистра hsvi_axi_axcache_control

Таблица 1.741 - Поля регистра hsvi_axi_axcache_control

Биты	Знач. по умолч.	Описание
7:0	8'h33	reserve
11:8	4’h3	Значение сигнала awcache канала записи интерфейса AXI Master контроллера HSVI1(4)
15:12	4’h3	Значение сигнала arcache канала чтения интерфейса AXI Master контроллера HSVI1(4)
19:16	4’h3	Значение сигнала awcache канала записи интерфейса AXI Master контроллера HSVI2(5)
23:20	4’h3	Значение сигнала arcache канала чтения интерфейса AXI Master контроллера HSVI2(5)
27:24	4’h3	Значение сигнала awcache канала записи интерфейса AXI Master контроллера HSVI0(3)
31:28	4’h3	Значение сигнала arcache канала чтения интерфейса AXI Master контроллера HSVI0(3)

1.3.10.1.4.14 Поля регистра write_enable

Таблица 1.742 - Поля регистра write_enable

Биты	Знач. по умолч.	Описание
31:0	32’h0	Для разрешения записи в регистры подсистемы XHSIF необходимо записать в данный регистр значение 32’h98fa2c45, для запрещения записи – записать любое другое значение. По умолчанию запись запрещена (в регистре 32’h0).

1.3.10.1.5 Схема мультиплексирования портов интерфейсных контроллеров подсистем XHSIF0 и XHSIF1

Информация о мультиплексировании портов контроллеров внешних интерфейсов, реализованных в подсистеме XHSIF0 приведена в таблице Таблица 1.743, в подсистеме XHSIF1 приведена в таблице

Таблица 1.744.

Таблица 1.743 - Мультиплексирование контроллеров в XHSIF0

SERDEC	Управлящий регистр XHSIF0	Значение	Подключенный контроллер	
SERDEC_0	serdes_mux_control0[0]	0	PCI Express 0	
		1	HSVI0	
SERDEC_1	serdes_mux_control0[1]	0	PCI Express 0	
		1	HSVI1	
SERDEC_2	serdes_mux_control0[2]	0	PCI Express 0	
		1	serdes_mux_control2[12] =	0: HSVI2
				1: FC3
SERDEC_3	serdes_mux_control0[3]	0	PCI Express 0	
		1	FC1	

Таблица 1.744 -Мультиплексирование контроллеров в XHSIF1

SERDEC	Управлящий регистр XHSIF1	Значение	Подключенный контроллер	
SERDEC_5	serdes_mux_control0[0]	0	PCI Express 1	
		1	HSVI3	
SERDEC_6	serdes_mux_control0[1]	0	PCI Express 1	
		1	HSVI4	
SERDEC_7	serdes_mux_control0[2]	0	PCI Express 1	
		1	serdes_mux_control2[12] =	0: HSVI5
				1: FC4
SERDEC_8	serdes_mux_control0[3]	0	PCI Express 1	
		1	FC2	

1.3.10.2 Описание блоков подсистем XHSIF0 и XHSIF1

1.3.10.2.1 Контроллер интерфейса PCI Express

1.3.10.2.1.1 Общие сведения

Ядро Expresso 3.0 предоставляет гибкий, высокопроизводительный и удобный для использования интерфейс уровня транзакций для шины PCI Express®

Основные характеристики:

- Совместимость с PCI Express™ Base Specification Revision 3.0 version 1.0
  - Обратная совместимость с PCI Express 2.1/2.0/1.1/1.0a
- Линии PCI Express x16, x8, x4, x2, или x1
- Поддержка скорости линии 8.0 GT/s, 5.0 GT/s, и 2.5 GT/s
- Поддержка 1-8 физических функций
- Поддержка SR_IOV, до 255 виртуальных функций
- Полная поддержка приложений:
  - Оконечная точка
  - Корневой порт
  - Оконечная точка/Корневой порт
  - Восходящий порт коммутатора
  - Нисходящий порт коммутатора
  - Опции бифуркации
  - Поддержка кросс-канала
- Совместимый с PIPE интерфейс PHY для простого подключения к PIPE PHYSupport для автономного и программного выравнивания
- Гибкие методы выравнивания (алгоритмический, предустановка, таблица пользователя, аппаратный)
- Опция обнаружения и исправления ошибок четности RAM и потока данных
- Интерфейсные опции ядра: обход уровня транзакций и частичная реализация уровня транзакций
- Доступно четыре варианта разрядности данных ядра и пять вариантов ширины линии для максимально гибкой поддержки широкого спектра полупроводниковых устройств с различными возможностями
  - 32-бит (x1, x2, x4)
  - 64-бит (x2, x4, x8)
  - 128-бит (x4, x8, x16)
  - 256-бит (x8, x16)
- Необыкновенно широкая поддержка PCI Express PHY
  - Уже поддерживаются ASIC, FPGA и дискретные PHY для PCI Express

- Интеграция с PIPE PHY пользователя включена в лицензию ядра (если уже не поддерживается)
- Полная поддержка обработки ошибок
  - Генерация AER, ECRC/проверка и восстановление ошибок четности и ECC
  - Поддержка обнаружения многочисленных дополнительных ошибок, встроенная симуляция контроля/активизации ошибок
- Гибкие опции ядра позволяют добиваться компромисса между сложностью и характеристиками
  - Конфигурируемый размер буферов приема и повтора
  - Конфигурируемая задержка для ASIC и FPGA
- Поддержка таких функций, как реверсирование линии, конфигурирование восходящих и нисходящих устройств, автономная ширина и скорость соединения
- Опция поддержки TPH
- Опция поддержки PASID
- Характеристики управления питанием
  - Поддержка PCI-PM
  - Бюджетирование питания
  - Динамическое распределение питания
  - Определяемая PHY поддержка ASPM (L0s, L1)
  - Определяемая PHY поддержка состояний экономии энергопотребления L1, L2, L3
  - Определяемая PHY PM подстостояния L1 PM с CLKREQ
- Механизм обмена значениями задержек
- Реализованы регистры конфигурации типа 0 в режиме оконечной точки
- Реализованы регистры конфигурации типа 1 в режимах корневого порта и коммутатора
  - Полная реализация регистра конфигурации коммутатора и корневого порта
- Поддерживает расширение конфигурационного пространства пользователем
- Простота использования
  - Декодирует принятые пакеты, чтобы предоставить маршрутную информацию (попадание в BAR, тег, и т. д.) пользователю
  - Реализует все аспекты требуемого пространства конфигурации PCIe; пользователь может добавлять свои списки настроек и регистры конфигурации посредством интерфейса расширения конфигурации
  - Обрабатывает пакеты TLP сообщений PCI и предоставляет их содержание в виде удобного в использовании интерфейса сообщений
  - Полная и простая в использовании генерация прерываний Legacy/MSI/MSI-X
  - Согласованная синхронизация и работа интерфейсов во всех режимах работы
  - Предоставляется большой объем диагностической информации для отладки на верхнем уровне системы и мониторинга связей
- Реализованы все три уровня PCI Express (транзакций, канальный и физический)
- Всесторонне проверено с помощью Avery PCI-Xactor для PCI Express, а также внутренних тестов Northwest Logic
- Поставляется с PCI Express Testbench
- Доступны сервисы кастомизации и интеграции

В ядре Expresso 3.0 реализованы все три уровня (транзакций, канальный и физический), определенные стандартом PCI Express .

Ядро доступно в различных конфигурациях, включая линии x16, x8, x4, x2 и x1, и поддерживает различные ASIC, FPGA и дискретные PHY. В ядре реализован интерфейс PIPE PHY для непосредственного подключения к PIPE PHY. В лицензию ядра обычно включена интеграция с пользовательским PIPE-совместимым PHY (если уже не поддерживается).

Свяжитесь с Northwest Logic для получения перечня поддерживаемых PHY или для запроса о поддержке нового PHY.

Ядро поставляется с PCI Express Testbench, обладающим возможностью создавать скрипты и тестовые воздействия, которые позволяют полностью проверить разработку пользователя до применения в аппаратной части.

1.3.10.2.1.2 Архитектура

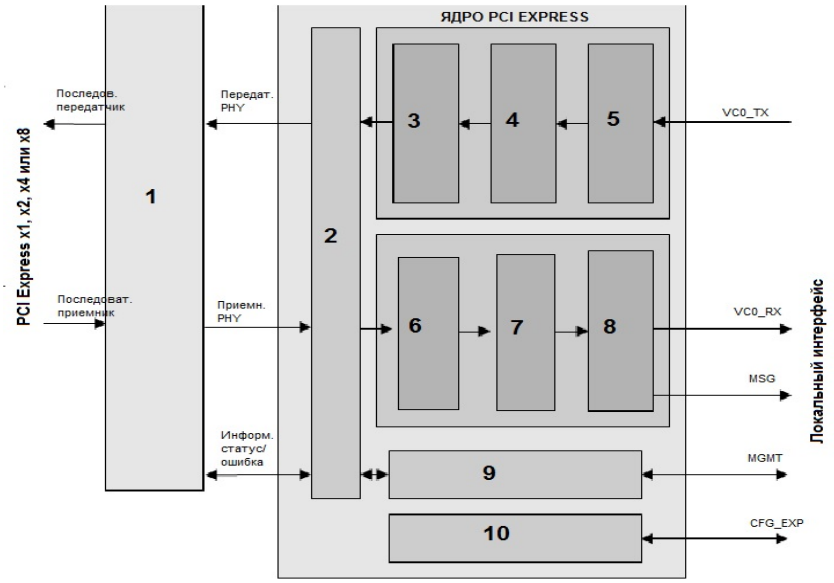


Рисунок 1.139 - Ядро Expresso 3.0

1. PHY (интегрированный или дискретный)
2. Интерфейс PHY
3. Передатчик физического уровня
4. Передатчик канального уровня
5. Передатчик уровня транзакций
6. Приемник физического уровня
7. Приемник канального уровня
8. Приемник уровня транзакций
9. Тактовый сигнал, конфигурация и управление
10. Регистры конфигурации

1.3.10.2.1.3 Конфигурации оконечной точки/корневого порта/коммутатора

Ядро Expresso 3.0 поддерживает приложения оконечной точки, корневого порта и коммутатора

- Оконечная точка
- Интерфейс PCI Express для устройства ввода/вывода (сетевые, хранения, сбора данных)
- Корневой порт
  - Реализует верхний узел иерархии PCI Express; часть корневого комплекса
  - Для использования со встроенным процессором, выполняющим операционную систему
- Оконечная точка/Корневой порт

- Работает и как оконечная точка, и как корневой порт с портами, которые выбирают режим работы
- Восходящий/нисходящий порт коммутатора
  - Работает как восходящий или нисходящий порт коммутатора PCI Express с портом, который выбирает режим работы

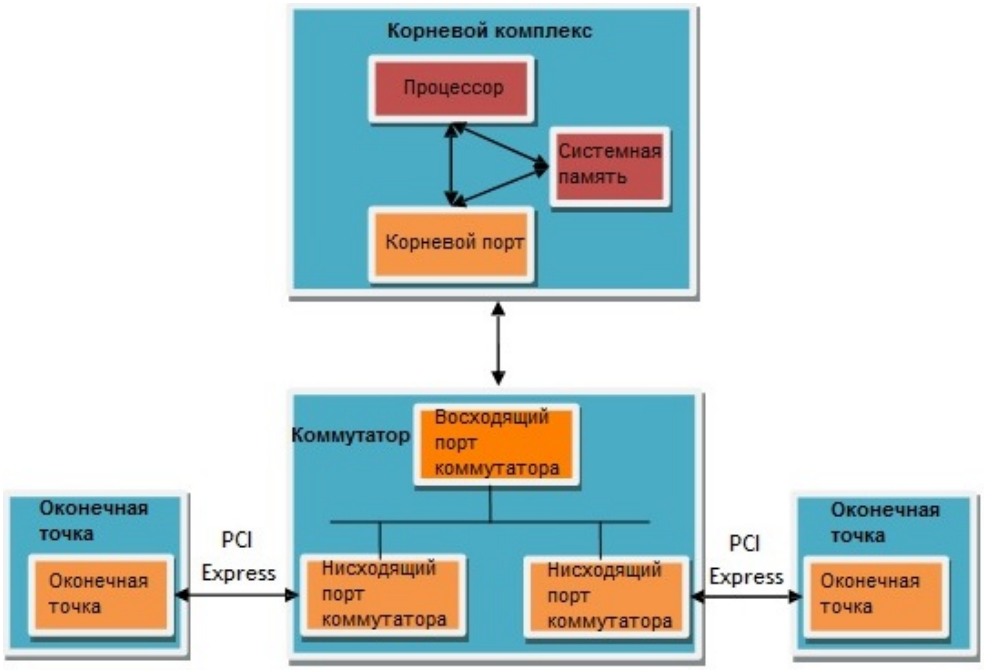


Рисунок 1.140 - Поддерживаемые конфигурации ядра Orange

1.3.10.2.1.4 Режим корневого порта/коммутатора

Ядро Expresso 3.0 доступно в следующих конфигурациях:

- Оконечная точка
- Корневой порт
- Оконечная точка/Корневой порт
- Восходящий/нисходящий порт коммутатора
-

1.3.10.2.1.4.1 Настройка физического уровня

При работе в режиме нисходящего порта (корневой порт и нисходящий порт коммутатора) ядро Expresso 3.0 настраивается как восходящий компонент (нисходящий порт/линии). При работе в режиме восходящего порта (оконечная точка и восходящий порт коммутатора) ядро Expresso 3.0 настраивается как нисходящий компонент (восходящий порт/линии). Настройки нисходящего порта и восходящего порта аналогичны, но в некоторых состояниях LTSSM имеются важные различия в поведении. В соединении PCI Express ближайшее к корневому комплексу устройство использует протокол настройки нисходящего порта, а другое устройство использует протокол настройки восходящего порта.

Трафик, идущий от различных оконечных точек к корневому комплексу, называется восходящим трафиком. Трафик, идущий от корневого комплекса к различным оконечным точкам, называется нисходящим трафиком.

1.3.10.2.1.4.2 Варианты mgmt_cfg_constants в конфигурациях корневого порта/порта коммутатора

Некоторые биты mgmt_cfg_constants действительны только в ядрах, поддерживающих конфигурацию корневой порт/коммутатор, а некоторые биты mgmt_cfg_constants принимают другое значение в режиме корневой порт/нисходящей порт. В данном разделе рассматриваются эти отличия.

1.3.10.2.1.4.2.1 Выбор режима корневой порт/восходящий коммутатор/нисходящий коммутатор

Для ядер, поддерживающих корневой порт/оконечную точку и для ядер, поддерживающих восходящий/нисходящий коммутатор, режим работы нисходящий/восходящий выбирается при помощи mgmt_cfg_constants[459, 456].

Битовое поле	Описание
[456]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И КОММУТАТОР: mgmt_rp_mode - этот бит совместно с mgmt_cfg_constants[459] == mgmt_sw_mode определяет режим работы ядра.
[459]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И КОММУТАТОР: бит mgmt_sw_mode – mgmt_sw_mode совместно с битом mgmt_rp_mode (mgmt_cfg_constants[456]) управляет режимом работы ядра: {mgmt_sw_mode, mgmt_rp_mode} 11 - Восходящий порт коммутатора Первичный - PCI Express Вторичный - локальный интерфейс Восходящий порт настройки LTSSM Ядро реализует тип 1 конфигурационных регистров на первичной стороне Ядро декодирует TLP на первичном и вторичном интерфейсе и передает только TLP, предназначенные для ресурсов, находящихся за портом коммутатора 10 - Корневой порт / Нисходящий порт коммутатора Первичный - локальный интерфейс Вторичный - PCI Express Нисходящий порт LTSSM Training Ядро реализует тип 1 конфигурационных регистров на первичной стороне Ядро декодирует TLP на первичном и вторичном интерфейсе и передает только TLP, предназначенные для ресурсов, находящихся за портом коммутатора 01 - Зарезервировано; не используется 00 – Оконечная точка Ядро реализует тип 0 конфигурационных регистров Восходящий порт настройки LTSSM

Поскольку выбор конфигурации ядра выполняется через порты ввода, то ядро по сути является многорежимным. То же ядро может быть сконфигурировано для переключения между режимами Оконечная точка/Корневой порт или Восходящий/Нисходящий коммутатор посредством изменения указанных выше битов конфигурации.

1.3.10.2.1.4.2.2 Идентификатор конфигурации Корневой порт/Порт коммутатора

Каждому устройству PCI Express присваивается уникальный идентификатор (ID), который при формировании TLP для PCI Express используется в полях ID инициатора и ID исполнителя. При работе в режиме восходящего порта (оконечной точки, восходящего порта коммутатора) этот идентификатор получается из принятых запросов записи конфигурации. При работе в режиме нисходящего порта (корневого порта, нисходящего порта коммутатора) идентификатор инициатора/исполнителя нисходящего порта обычно назначается командой записи конфигурации, полученной коммутатором/мостом. Если корневой порт напрямую подключен к процессору, значение для данного приложения может быть назначено как:

- Bus==0, Device==0 и Function==0 (0x0000)

Идентификатор инициатора/исполнителя определяется как Bus[7:0]Device[4:0]Function[2:0] и присваивается при помощи mgmt_cfg_constants[455:440]:

Битовое поле	Описание
[455:440]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И КОММУТАТОР: Идентификатор нисходящего порта - Это 16-битовое поле используется PCIe для ID инициатора и ID исполнителя, когда ядро работает в режиме нисходящего порта (корневой порт, нисходящий порт коммутатора). Когда ядро работает в режиме восходящего порта (оконечная точка, восходящий порт коммутатора, то ядро получает свой идентификатор

Идентификатору нисходящего порта не должен присваиваться тот же номер шины, что и в mgmt_cfg_constants: Поле номера вторичной шины, поскольку идентификатор нисходящего порта принадлежит восходящему порту коммутатора/моста, который находится на шине с другим номером, не таким, как для нисходящих устройств PCI Express.

1.3.10.2.1.4.2.3 Предоставление информации корневого порта/порта коммутатора

Оконечные точки должны предоставить условно-бесконечный размер буферов для приема данных и заголовков (CD и CH Credits). (Оконечные точки должны гарантировать, что они могут принять данные завершения от всех запросов чтения, чтобы не останавливать иерархию PCIe из-за неспособности принять ответные пакеты). Корневой порт и порт коммутатора должны предоставить действительный размер буферов для приема данных и заголовков, поскольку они направляют транзакции от других устройств и поэтому не способны управлять количеством запросов чтения, которые ожидают выполнения. При необходимости предоставить условно-бесконечный размер буферов для приема данных и заголовков для корневого порта и порта коммуникатора выделяется mgmt_rp_adv_inf_ch_cd. Тем не менее, это не рекомендуется, поскольку возникнет переполнение, если mgmt_rp_adv_inf_ch_cd == 1, и принятые ответные пакеты не могут быть отправлены должным образом.

Битовое поле	Описание
[457]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И КОММУТАТОР: mgmt_rp_adv_inf_ch_cd - Этот бит определяет, когда ядро, работающее в режиме восходящего порта, нисходящего порта или корневого порта предоставляет (0) действительный размер буферов для приема данных и заголовков (ожидаемое поведения для коммутаторов/корневых портов) или (1) предоставить условно-бесконечный размер буферов для приема данных и заголовков (требуемое поведение оконечных точек). Когда ядро работает в режиме оконечной точки, всегда предоставляется условно-бесконечный размер буферов для приема данных и заголовков, независимо от этого бита.

1.3.10.2.1.4.2.4 Конфигурация регистра базового адреса (BAR) корневого порта/порта коммутатора

При работе в режиме корневого порта или порта коммутатора ядро реализует тип 1 регистров конфигурации. Реализованы только два местоположения BAR, определенные в конфигурационном пространстве типа 1, а не шесть местоположений BAR, определенных в конфигурационном пространстве и реализованных в режиме оконечной точки. За исключением количества BAR, начальная установка их такая же, как в режиме оконечной точки.

Битовое поле	Описание
[175:144]	Base Address Cfg0[31:0] - В режиме корневого порта пользователь может определить до двух 32-битовых BAR ввода/вывода или памяти или до одного 64-битового BAR памяти при помощи Base Address Cfg0 и Base Address Cfg1. Base Address Cfg0 и Base Address Cfg1 предназначены для реализации регистра базового адреса 0 и регистра базового адреса 1 в конфигурационном пространстве типа 1. Начальная установка Base Address Cfg0 и Base Address Cfg1 производится по-отдельности, как в режиме оконечной точки, за исключением того, что в режиме корневого порта существуют только два местоположения 32-битового
[207:176]	Base Address Cfg1[31:0] – См. выше, описание Base Address Cfg0.
[239:208]	Base Address Cfg2[31:0] – См. выше, описание Base Address Cfg0. Не используется для режима корневого порта.
[271:240]	Base Address Cfg3[31:0] – См. выше, описание Base Address Cfg0. Не используется для режима корневого порта.
[303:272]	Base Address Cfg4[31:0] – См. выше, описание Base Address Cfg0. Не используется для режима корневого порта.
[335:304]	Base Address Cfg5[31:0] – См. выше, описание Base Address Cfg0. Не используется для режима корневого порта.
[367:336]	Expansion ROM Cfg[31:0] – Определяет, реализован ли регистр базового адреса дополнительного ROM, и если реализован, то указывает его размер. Биты [31:11] предназначены для определения размера BAR. Биты [10:0] должны быть установлены в 0, чтобы минимальный размер BAR составлял 2 килобайта. Для определения размера BAR дополнительного ROM биты устанавливаются последовательно с бита 31 вниз до последнего бита адреса, который должен быть реализован. Все оставшиеся биты устанавливаются в ноль. Например, 64-килобайтный BAR дополнительного ROM создается установкой ROM Cfg = 0xFFFF0000. BAR дополнительного ROM используется для хранения конкретной инициализации устройства или загрузочных инструкций, которые должны выполняться в процессе загрузки. BAR дополнительного ROM используется редко. В случае реализации в этом местоположении BAR должна быть реализована допустимая структура дополнительного ROM - в противном случае система может не загрузиться. Если не используется, значение поля должно быть 0x00000000. Expansion ROM Cfg одинаков в режимах оконечной точки и корневого порта.

Для режимов оконечной точки и корневого порта/порта коммутатора BAR дополнительного ROM реализуется одинаково.

При использовании в конфигурациях корневой порт/порт коммутатора BAR0, BAR1 и BAR дополнительного ROM не должны быть распределены в области ввода/вывода и памяти, определяемые I/O Base, I/O Limit, Mem Base, Mem Limit, Prefetchable Mem Base, и Prefetchable Mem Limit (см. ниже), поскольку эти ресурсы логически находятся по восходящей от корневого порта.

1.3.10.2.1.4.3 Конфигурация корневого порта/порта коммутатора

В режиме корневого порта в ядре реализованы регистры конфигурации типа 1, а не типа 0.

Когда ядро работает как нисходящий порт (корневой порт, нисходящий порт коммутатора), запросы конфигурации не могут быть переданы по восходящей, а любые принятые запросы конфигурации PCI Express будут обрабатываться как неподдерживаемые запросы.

В конфигурациях корневой порт/порт коммутатора области ввода/вывода и памяти, которые находятся по нисходящей от корневого порта, определяются (в соответствии с пространством конфигурации типа 1) при помощи регистров конфигурации I/O Base, I/O Limit, Mem Base, Mem Limit, Prefetchable Mem Base, Prefetchable Mem Limit, Secondary Bus Number, и Subordinate Bus Number.

При работе как нисходящий порт (корневой порт и нисходящий порт коммутатора) ядро будет принимать полученные транзакции, которые выходят за пределы этой области (и,

следовательно, целевых ресурсов, которые расположены по восходящей от ядра) и передавать их в пользовательский локальный интерфейс приема. Транзакции внутри этих областей предназначены для устройств, расположенных по нисходящей от корневого порта и при приеме будет считаться неподдерживаемыми запросами.

При работе как восходящий порт коммутатора ядро будет принимать полученные транзакции, которые попадают внутрь этой области (и, следовательно, целевых ресурсов, которые расположены по нисходящей от ядра) и передавать их в пользовательский локальный интерфейс приема. Транзакции, попадающие за пределы этих областей, предназначены для устройств, расположенных по восходящей от корневого порта и при приеме будет считаться неподдерживаемыми запросами.

Область запрещается (указывая, что ресурсы этого типа отсутствуют по нисходящей от порта) программированием базового значения, большего, чем значение Limit.

1.3.10.2.1.4.4 Определение mgmt_rp_status

Когда ядро сконфигурировано как корневой порт или порт коммутатора, оно передает информацию, необходимую приложениям корневого порта и коммутатора, посредством mgmt_rp_status. mgmt_rp_status доступен только для ядер, лицензированных с поддержкой корневого порта.

Битовое поле	Описание
[5:0]	Зарезервировано.
[9:6]	Зарезервировано.
[12:10]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И ПОРТ КОММУТАТОРА: Действительно только при работе в режиме нисходящего порта – Downstream Port System Error[2:0] - Эти порты сообщают об ошибках PCI Express, обнаруженных нисходящим портом или полученных нисходящим портом как сообщения об ошибках от нисходящих устройств. При возникновении ошибки соответствующий бит ошибки активизируется для одного цикла синхронизации. Ошибки, сообщаемые этим портом, должны регистрироваться и сообщаться системному ПО. [2] – (1) Фатальная ошибка обнаружена ядром или нисходящим устройством; (0) ошибка не выявлена [1] – (1) Нефатальная ошибка обнаружена ядром или нисходящим устройством; (0) ошибка не выявлена [0] – (1) Исправляемая ошибка обнаружена ядром или нисходящим устройством; (0) ошибка не выявлена Спецификация PCI Express позволяет объединять одинаковые ошибки (исправляемые, нефатальные или фатальные), если они возникают слишком близко, чтобы передавать их по отдельности. Ядро объединяет одинаковые ошибки, которые возникают в одном и том же цикле синхронизации. При работе в качестве оконечной точки, восходящего коммутатора и нисходящего порта коммутатора исправляемые, нефатальные и фатальные ошибки сигнализируются/пересылаются корневому комплексу посредством соответствующих сообщений об ошибке. Когда ядро работает как нисходящий порт, эти события перенаправляются в системную ошибку нисходящего порта.
[15:13]	Зарезервировано. Функция этих битов в предыдущих версиях ядра была объединена в системную ошибку корневого порта.

Битовое поле	Описание
[19:16]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И ПОРТ КОММУТАТОРА: Действительно только при работе в качестве нисходящего порта Downstream Port Legacy Interrupts[3:0] - следующие порты сообщают об унаследованных прерываниях, генерируемых ядром и реконструированных из принятых сообщений о прерывании активации/деактивации от нисходящих устройств: Бит[3] - (1) прерывание INTD активно и требует обслуживания; (0) неактивно Бит[2] - (1) прерывание INTC активно и требует обслуживания; (0) неактивно Бит[1] - (1) прерывание INTB активно и требует обслуживания; (0) неактивно Бит[0] - (1) прерывание INTA активно и требует обслуживания; (0) неактивно
[23:20]	Зарезервировано. Функция этих битов в предыдущих версиях ядра была объединена в унаследованные прерывания корневого порта.
[24]	Зарезервировано. Функция этого бита в предыдущих версиях ядра была объединена в mgmt_interrupt_o.
[31:25]	Зарезервировано.
[37:32]	Зарезервировано.
[63:38]	Зарезервировано.
[127:64]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И ПОРТ КОММУТАТОРА: Адрес BAR0; используется для приложений коммутатора; также доступен в mgmt_cfg_status
[191:128]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И ПОРТ КОММУТАТОРА: Маска адреса BAR0; используется для приложений коммутатора; также доступна в mgmt_cfg_status
[223:192]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И ПОРТ КОММУТАТОРА: Адрес BAR1; используется для приложений коммутатора; также доступен в mgmt_cfg_status
[287:224]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И ПОРТ КОММУТАТОРА: Маска адреса BAR1; используется для приложений коммутатора; также доступна в mgmt_cfg_status
[351:288]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И ПОРТ КОММУТАТОРА: Адрес дополнительного ROM; используется для приложений коммутатора; также доступен в mgmt_cfg_status
[387:352]	ТОЛЬКО ЯДРА, ПОДДЕРЖИВАЮЩИЕ КОНФИГУРАЦИИ КОРНЕВОЙ ПОРТ И ПОРТ КОММУТАТОРА: Маска адреса дополнительного ROM; используется для приложений коммутатора; также доступна в mgmt_cfg_status
[511:388]	Зарезервировано.

1.3.10.2.1.4.5 Генерирование запросов

Пакеты TLP, которые направляются на интерфейс передачи, не проверяются на достоверность и передаются в точности так, как предоставляются. Предполагается, что передаваемые пакеты уже проверены для направления к ресурсам, нисходящим по отношению к корневому порту, прежде чем они получили разрешение на пересылку на интерфейс передачи.

1.3.10.2.1.4.6 Конфигурация корневого порта PCI

Конфигурация устройств, нисходящих по отношению к корневому порту, должна проводиться в соответствии со стандартными правилами конфигурации PCI при помощи интерфейсов передачи корневого порта, для того, чтобы переслать запросы записи конфигурации и чтения конфигурации, а также интерфейса чтения корневого порта, чтобы получить результирующие завершения.

Конфигурирование иерархии PCI - это сложная задача, которую лучше выполнять при помощи подпрограммы, а не специальной аппаратной части. Дополнительную информацию см. в спецификациях PCI и PCI Express. По возможности рекомендуется максимально использовать существующую программу конфигурации PCI.

1.3.10.2.1.4.7 Блокирующие транзакции

Ядро Expresso 3.0 Core не поддерживает блокирующие транзакции в режиме корневого порта или оконечной точки. Блокирующие транзакции требуются только для унаследованных функций и их использование настоятельно не рекомендуется спецификацией PCI Express из-за серьезной потери производительности реализации блокирующих транзакций, а также невозможности определить, будут ли блокирующие транзакции поддерживаться устройствами в иерархии PCI Express.

1.3.10.2.1.4.8 Управление питанием

Ядро The Expresso 3,0 поддерживает состояния канала L1 и L2 при работе в качестве корневого порта/порта коммутатора.

Процесс управления питанием для корневого порта отличается от процесса управления питанием для оконечной точки тем, что ядро выступает в роли системного устройства, а не оконечной точки.

1.3.10.2.1.4.8.1 Вход нисходящего порта в состояние L1 и выход из него

1.3.10.2.1.4.8.1.1 Вход в состояние L1

До начала входа в состояние L1 программа управления питанием операционной системы проинформирует драйверы нисходящих устройств ядра о том, что канал перейдет в состояние L1. Драйверы должны завершить все незавершенные транзакции и сохранить необходимую информацию о состоянии, прежде чем разрешить программе управления питанием перевести канал в состояние L1.

Программа управления питанием инициирует процесс перевода канала PCI Express в состояние L1, записывая в регистр конфигурации состояния питания нисходящего порта (оконечной точки/нисходящего порта коммутатора) D-состояние, отличное от D0. Запись состояния питания, отличного от D0, заставляет нисходящий порт остановить транзакции, ждать подтверждения всех транзакций, ожидающих выполнения, и сообщить о своей готовности перейти в состояние L1 передачей пакетов DLLP (пакет канального уровня) PM_ENTER_L1.

Когда автомат состояния управления питанием ядра Expresso 3.0, работающий как нисходящий порт, получает пакеты DLLP PM_ENTER_L1, ядро начинает процесс перехода канала в состояние L1. Ядро прекращает пересылку TLP, ждет подтверждение всех TLP, ожидающих выполнения, а затем передает PM_REQ_ACK пакеты DLLP в ответ на PM_ENTER_L1 пакеты DLLP порта, чтобы сообщить восходящему порту, что нисходящий порт также готов к переходу в состояние L1.

Когда восходящий порт получит от ядра PM_REQ_ACK пакеты DLLP, восходящий порт передаст 1 или 2 (в зависимости от скорости канала) стандартные последовательности электрического простоя и переключит свой передатчик в состояние электрического простоя. Когда автомат состояния управления питанием, работающий как нисходящий порт, обнаруживает, что восходящий порт переходит в состояние электрического простоя, ядро передаст 1 или 2 (в зависимости от скорости канала) стандартные последовательности электрического простоя и переключит свой передатчик в состояние электрического простоя. Теперь канал перешел в состояние L1. PHY обоих устройств переведены в состояние с низким энергопотреблением для экономии энергии. Поддерживается общее питание, опорный тактовый сигнал PCI Express, тактовый сигнал PHY и тактовый сигнал ядра.

1.3.10.2.1.4.8.1.2 Выход из состояния L1

Выход из состояния L1 начинается когда-либо ядро, либо восходящий порт выводит свой PHY из состояния низкого энергопотребления и выходит из состояния электрического простоя, чтобы передать TLP. Устройство, желающее передать TLP, выходит из состояния электрического простоя и переходит к восстановлению, чтобы повторно настроить канал. Другое устройство обнаруживает выход удаленного устройства из состояния электрического простоя, выводит свой PHY из состояния низкого энергопотребления, выводит свой передатчик из состояния электрического простоя и также переходит к восстановлению. Во время восстановления канал повторно настраивается, и PHY обоих устройств восстанавливают синхронизацию на уровне символов. После успешного восстановления оба устройства возвращаются в состояние L0 и возобновляют нормальную работу, включая передачу и прием TLP. Обратите внимание, что при нормальном, успешном переходе в состояние L1 и выходе из него ядро не сбрасывается, не отключается от питания и сохраняет всю текущую информацию.

Удаленное устройство обычно тоже сохраняет всю текущую информацию. Поведение восходящего порта при сбросе в состоянии L1 определяется его предложением No_Soft_Reset регистра возможностей управления питанием. Если для восходящего поста значение регистра No_Soft_Reset == 1, то восходящий порт не будет сброшен при переходе в состояние L1. Если No_Soft_Reset == 0, то канал сбрасывается переходом в состояние L1, и восходящий порт должен быть заново пронумерован и инициализирован BIOS/OS до обращения к нему.

1.3.10.2.1.4.8.2 Вход нисходящего порта в состояние L2 и выход из него

1.3.10.2.1.4.8.2.1 Вход в состояние L2

До начала входа в состояние L2 программа управления питанием операционной системы проинформирует драйверы нисходящих устройств ядра о том, что канал перейдет в состояние L2. Драйверы должны завершить все транзакции, ожидающие выполнения и сохранить необходимую информацию о состоянии, прежде чем разрешить программе управления питанием перевести канал в состояние L2.

Программа управления питанием инициирует процесс перевода оконечной точки в состояние L2 передачей сообщения PME_Turn_Off в нисходящем направлении (от корневого комплекса). Промежуточные коммутаторы транслируют это сообщение всем устройствам в иерархии. Каждое устройство в иерархии подтверждает запрос PME_Turn_Off передачей сообщения PME_TO_Ack в локальный канал. Промежуточные коммутаторы собирают отклики PME_TO_Ack от всех устройств и направляют финальный совокупный отклик PME_TO_Ack корневому комплексу. После того, как восходящий порт передаст сообщение PME_TO_Ack, восходящий пор ждет подтверждения от всех TLP, ожидающих выполнения, и сообщает о готовности перехода в состояние L2 передачей PM_ENTER_L23 пакетов DLLP.

Когда автомат состояния управления питанием ядра Expresso 3.0, работающий как нисходящий порт, получает пакеты DLLP PM_ENTER_L23, ядро начинает процесс перехода канала в состояние L2. Ядро прекращает пересылку TLP, ждет подтверждения всех TLP, ожидающих выполнения, а затем передает PM_REQ_ACK пакеты DLLP в ответ на PM_ENTER_L23 пакеты DLLP оконечной точки, чтобы сообщить восходящему порту, что нисходящий порт также готов к переходу в состояние L2. Когда восходящий порт получит от ядра PM_REQ_ACK пакеты DLLP, восходящий порт передаст 1 или 2 (в зависимости от скорости канала) стандартные последовательности электрического простоя и переключит свой передатчик в состояние электрического простоя. Когда автомат состояния управления питанием, работающий как нисходящий порт, обнаружит, что восходящий порт переходит в состояние электрического ожидания, ядро передаст 1 или 2 (в зависимости от скорости канала) стандартные последовательности электрического простоя и переключит свой передатчик в состояние электрического простоя. Теперь канал перешел в состояние L2. PHY обоих устройств переведены в состояние с низким энергопотреблением для экономии энергии. Система может

отключить общее питание, и в этом случае устройства будут находиться в состоянии D3cold. Если питание не отключается, то должны также сохраняться тактовый сигнал PHY и тактовый сигнал, и устройство будет находиться в состоянии D3hot.

1.3.10.2.1.4.8.2.2 Выход из состояния L2

Выход из состояния L2 начинается либо когда снова подается питание, которое было отключено, либо когда ядро или восходящий порт выводит свой PHY из состояния низкого энергопотребления и выходит из состояния электрического простоя, чтобы повторно настроить соединение для передачи TLP. В отличие от выхода из состояния L1, при выходе из состояния L2 у обоих устройств выключены физический и канальный уровни, и их требуется повторно инициализировать. Вся текущая информация в обоих устройствах теряется, за исключением тех значений регистра конфигурации, которые сохраняются устройствами, имеющими вспомогательный источник энергии.

1.3.10.2.1.5 Обработка ошибок

Для большинства типов ошибок ядро Expresso 3.0 обнаруживает ошибки и реализует соответствующий ответ без вмешательства пользователя. В общем случае пользователю необходимо обрабатывать обнаружение ошибок и сообщения только для тех ошибок, тип которых ядро не может определить из-за недостатка информации.

Далее в таблицах определены типы ошибок PCI:

В столбце TYPE указывается серьезность ошибки PCI Express:

COR - исправляемая

NFAT - неисправляемая - нефатальная

FAT - неисправляемая - фатальная

Ядро не проверяет пересланные пользователем пакеты (пакеты, пересланные через интерфейс передачи) не достоверность и передает и в том виде, в котором они получены.

1.3.10.2.1.5.1 Общие ошибки PCI Express

Ошибка	Тип
Исправляемая внутренняя ошибка	COR
Неисправляемая внутренняя ошибка	FAT
Переполнение журнала заголовков	COR

1.3.10.2.1.5.2 Ошибки физического уровня

Ошибка	Тип
Ошибка приемника	COR

1.3.10.2.1.5.3 Ошибки канального уровня

Ошибка	Тип
Ошибка пакета TLP	COR
Ошибка пакета DLLP	COR
Тайм-аут повтора	COR
Переполнение счетчика REPLAY_NUM (Rollover)	COR
Ошибки протокола канального уровня	FAT
Внезапный разрыв соединения (Surprise Down)	FAT

1.3.10.2.1.5.4 Ошибки уровня транзакции

ОШИБКА	ТИП
Получен искаженный пакет TLP	NFAT
Не прошла проверка ECRC	NFAT
Неподдерживаемый запрос	NFAT
Тайм-аут завершения	NFAT
Аварийное завершение	NFAT
Неожиданное завершение	NFAT
Нарушение ACS	NFAT
Блокировка пакета MC TLP (MC Blocked TLP)	NFAT
Блокировка внешнего запроса AtomicOp (AtomicOp Egress Blocked)	NFAT
Переполнение приемника	FAT
Ошибка протокола управления потоком данных	FAT
Искаженный пакет TLP	FAT

1.3.10.2.1.6 Управление питанием

Ядро Expresso 3.0 поддерживает состояния канала L0, L0s, L1, L2 и L3. Поддержка состояний L0 (полностью рабочее состояние) и L3 (выключено) всегда разрешена. Остальные состояния разрешаются/запрещаются при помощи полей mgmt_cfg_constants. Если поддержка состояний L0s, L1 или L2 разрешена, то разработка пользователя должна конфигурировать возможности ядра управлять питанием, а для некоторых состояний канала выполнять дополнительные действия при входе или выходе. В данном разделе описаны рекомендуемые действия, которые должна выполнять логика пользователя для управления состояниями L0s, L1 и L2, а также в качестве реакции на них.

Спецификация PCI Express определяет следующие состояния канала:

- L0 – активное
  - Питание включено
  - Тактовый сигнал и PLL активны; тактовый сигнал ядра активен
  - Все транзакции и операции PCI Express разрешены
- L0s – малая задержка восстановления работы энергосберегающий режим "простоя"
  - Питание включено
  - Тактовый сигнал и PLL активны; тактовый сигнал ядра активен
  - PHY переведен в состояние электрического простоя
  - Удаленный PHY приемника должен повторно установить синхронизацию соединения на уровне символов при выходе из состояния L0s
  - Если состояние L0s разрешено программой управления питания, ядро самостоятельно переходит в L0s, когда передающая сторона канала свободна, и выходит из L0s , когда есть информация, ожидающая пересылки.
  - Необходима периодическая передача пакетов DLLP управления каналом, и поэтому когда канал свободен, он будет регулярно входить в состояние L0s и выходить из него, чтобы пересылать пакеты DLLP управления каналом.
- L1 – большая задержка восстановления, режим "простоя" с низким энергопотреблением
  - Питание включено
  - Тактовый сигнал и PLL активны; тактовый сигнал ядра должен оставаться активен
  - Питание значительной части PHY отключено
    - PHY переведен в состояние электрического простоя
    - PHY приемника в состоянии электрического простоя
  - Удаленный и локальный PHY должны повторно установить синхронизацию соединения на уровне символов при выходе из состояния L0s

- Вход в состояние L1 выполняется под управлением программы управление питания и автономно аппаратной частью посредством механизма ASPM L1.

- L2 – питание канала от дополнительного источника, состояние значительной экономии энергии
  - Питание отключено; дополнительный источник питания остается включенным
  - Тактовый сигнал и PLL неактивны; опорный тактовый сигнал PCI Express недоступен; тактовый сигнал ядра выключен
  - В этом состоянии питание ядра Expresso 3.0 отключено
  - Перед входом с состояние L2 (и соответственно отключением питания ядра) пользовательская логика управления питанием, подключенная к дополнительному источнику питания, должна сохранить информацию о состоянии ядра, которая не должна уничтожиться при входе с состояние L2 и выходе из него, а затем восстановить состояние ядра при выходе из L2. В состоянии L2 контекст устройства теряется (физический уровень и канальный уровень отключаются, и в регистры конфигурации записываются значения по умолчанию), за исключением информации, которая должна быть сохранена пользовательской логикой управления питанием. При подаче питания ядро начнет работу в состоянии LTSSM DETECT, и канал будет повторно настроен. Во время этого процесса ядро восстанавливает информацию, сохраненную пользовательской логикой управления питанием.
  - Пользовательская логика может реализовывать поддержку WAKE# или сигнала готовности Beacon для пробуждения системы, находящейся в этом состоянии. Передача сигналов WAKE#/Beacon должна выполняться пользователем, поскольку в этом состоянии ядро отключено. Если в этом состоянии поступают сигналы WAKE# или Beacon, то пользовательская логика передает эту информацию ядру для включения в процесс восстановления, так чтобы ядро смогло генерировать сообщение системе для идентификации его устройства как источника события пробуждения.
  - Переход в состояние L2 выполняется программой управления питанием.
- L3 - отключено
  - Питание отключено; дополнительный источник питания отключен
  - В этом состоянии все питание отключено, и ядро, PHY и пользовательская логика находятся в нерабочем состоянии.
  - Вся информация состояния теряется

1.3.10.2.1.6.1 Конфигурирование ядра для поддержки управления питанием

Ядро Expresso 3.0 позволяет логике пользователя реализовать разнообразные функциональные возможности управления питанием. Возможности управления питанием предлагаются и управляются в первую очередь посредством использования порта mgmt_cfg_control.

Поддержка управления питанием: Настройки управления питанием (PMC) == mgmt_cfg_control[143:128]:

- Биты[15:11] - PME_Support[4:0] указывается следующим образом:
  - 1xxxx – PME# может быть подтвержден из D3cold
  - x1xxx - PME# может быть подтвержден из D3hot
  - xx1xx - PME# может быть подтвержден из D2
  - xxx1x - PME# может быть подтвержден из D1
  - xxxx1 - PME# может быть подтвержден из D0
- Бит[10] – 1 == Аппаратная часть пользователя поддерживает состояние D2; 0 - не поддерживает; этот бит не следует устанавливать в 1, если не разрешена поддержка ядром состояния L1 (mgmt_cfg_constants[377] == 1).

- Бит[9] – 1 == Аппаратная часть пользователя поддерживает состояние D1; 0 - не поддерживает; этот бит не следует устанавливать в 1, если не разрешена поддержка ядром состояния L1 (mgmt_cfg_constants[377] == 1).
- Биты[8:6] - Aux_Current[2:0] указывает мощность дополнительного источника, необходимую для данного устройства:
  - 111 – 375 mA
  - 110 – 320 mA
  - 101 – 270 mA
  - 100 – 220 mA
  - 011 – 160mA
  - 010 – 100 mA
  - 001 – 55 mA
  - 000 - Дополнительного источника питания не требуется, или требования к дополнительному источнику питания указываются посредством регистра данных
- Бит[5] – Device Specific Initialization (DSI) – 1 == Устройству требуется специальная инициализация (помимо нормального конфигурирования PCI), прежде чем им может пользоваться универсальный драйвер. 0 - специальной инициализации на требуется.
- Бит[4] – Зарезервирован. Установлен в 0.
- Бит[3] – PME_Clock – 1 == Функция зависит от присутствия тактового сигнала PCI для работы PME# 0 указывает, что тактовый сигнал PCI не требуется, или что генерация PME# на поддерживается. Функции не должны зависеть от тактового сигнала PCI, когда поддерживают выдачу PME# из D3cold, поскольку ядро Expresso 3.0 и тактовый сигнал PCI в этом состоянии отключены.
- Биты[2:0] – Version - Должны быть установлены в “011”, указывая, что функция соответствует версии 1.2 спецификации интерфейса управления питанием PCI.

Если разработка пользователя требует дополнительного источника питания, то требования пользователя к дополнительному источнику питания должны указываться либо в Aux_Current, либо в регистре данных.

Пользователь может реализовать функциональность регистра данных при помощи полей Data_Scale и Data_Select регистров поддержки управления питанием: Power Management Control/Status (PMCSR) == mgmt_cfg_control[159:144] и Power Management Data Register Control[7:0] == mgtm_cfg_control[175:168].

Для базовой разработки с ограниченными возможностями управления питанием рекомендуются следующие установки:

- Поддержка управления питанием: Power Management, D2 или D1; не требуется дополнительный источник питания; версия == 3) Capabilities (PMC) == mgmt_cfg_control[143:128] == 16’h0003. (Нет поддержки PME
- Поддержка управления питанием: Power Management Control/Status (PMCSR) == mgmt_cfg_control[159:144] == 16’h0000 (Нет поддержки PME, нет поддержки Data Register)

1.3.10.2.1.6.2 L0s

Ядро Expresso 3.0 поддерживает управление питанием в активном состоянии (APSM) L0s. Поддержка ASPM L0s разрешается в mgmt_cfg_constants[378].

Когда поддержка ASPM L0s разрешена, TX Entry Time == mgmt_cfg_constants [399:384] конфигурируется пользователем для установки требуемого времени, в течение которого пересылки TLP и DLLP неактивны, прежде чем передатчик перейдет в состояние L0s.

Количество последовательностей NFTS, необходимых локальному PHY для восстановления синхронизации соединения на уровне символов при выходе из состояния L0s, программируется в mgmt_cfg_constants[407:400].

NFTS Timeout Extend, mgmt_cfg_constants[495:488], определяет, период ожидания ядра после истечения установленного времени выхода из состояния L0s, прежде чем перейти к восстановлению канала, чтобы восстановиться после не выполненного выхода из L0s. Из-за больших задержек между выходом приемника PHY, находящимся в состоянии электрического простоя, и соответствующими данными приемника, обычно необходимо выбирать относительно большие величины тайм-аута NFTS и NFTS Timeout Extend. Дополнительную информацию см. в описании NFTS Timeout Extend.

Replay Timer Rx_L0s_Adjustment == mgmt_cfg_constants[423:410] устанавливается в зависимости от PHY и должно указывать время, требуемое приемным цепям PHY для перехода из L0s в L0.

TX L0s Adjustment == mgmt_cfg_constants[425] и ACK Timer TX L0s Adjustment == mgmt_cfg_constants[439:426], устанавливаются в зависимости от PHY и должны указывать время, требуемое передающим цепям PHY для перехода из L0s в L0.

PCI Express Device Capabilities Register Control[31:0] == mgmt_cfg_control[63:32] имеет следующие поля L0s:

- Биты[8:6] – Endpoint L0s Acceptable Latency – Из базовой спецификации PCI Express, версия 2.1 раздел 7.8.3: “Допустимая общая задержка, которую может выдержать оконечная точка вследствие перехода из состояния L0s в состояние L0. Это, в основном, является косвенным отражением внутренней буферизации оконечной точки. Программа управления питанием использует указанную допустимую величину задержки L0s с задержками выхода, указанными всеми компонентами, составляющими тракт данных от этой оконечной точки до корневого порта корневого комплекса, чтобы определить, можно ли использовать вход ASPM L0s без потери производительности”. Следует отметить, что уровень буферизации относится не к буферизации ядра Expresso 3.0, а к буферизации пользовательского приложения. Пользователи должны устанавливать это поле в соответствии с задержкой, допустимой для их приложения.
  - 000 - Максимум 64 ns
  - 001 - Максимум 128 ns
  - 010 - Максимум 256 ns
  - 011 - Максимум 512 ns
  - 100 - Максимум 1 µs
  - 101 - Максимум 2 µs
  - 110 - Максимум 4 µs
  - 111 - Не ограничено
  - Устройства, отличные от оконечных точек, должны аппаратно устанавливать это поле в 000.
- PCI Express Link Capabilities Register Control[31:0] == mgmt_cfg_control[111:80] имеет следующие поля L0s:
- Биты[14:12] – L0s Exit Latency - Время, требуемое для полного перехода из L0s в L0:
  - 000 - Менее 64 ns
  - 001 - от 64 ns до 128 ns
  - 010 - от 128 ns до 256 ns
  - 011 - от 256 ns до 512 ns
  - 100 - от 512 ns до 1 µs
  - 101 - от 1 µs до 2 µs
  - 110 - от 2 µs до 64 µs

- 111 - Более 4 μs
- Задержки выхода могут быть существенно увеличены, если опорные тактовые сигналы PCI Express, используемые двумя устройствами в канале, общие или раздельные.

- Биты[11:10] – Поддержка управления питанием в активном состоянии (ASPM) должны быть установлены в “01” или “11”, если поддержка L0s разрешена, и в “00”, если запрещена.

1.3.10.2.1.6.3 L1

Ядро Expresso 3.0 поддерживает L1.

Программный вход в состояние L1 разрешается в mgmt_cfg_constants[377].

Аппаратный автономный вход в состояние L1 (Управление питанием в активном состоянии (APSM) L1) разрешается в mgmt_cfg_constants[409].

Программный переход в состояние L1 для восходящих потов (оконечная точка/восходящий порт коммутатора) выполняется следующим образом:

- Программа инициализирует переход канала в L1, записывая информацию о поддержке управления питанием: В регистре конфигурации Power State значение, отличное от 00 == D0 (обычно программа записывает 11 == D3hot). Следует учитывать, что драйвер устройства участвует в этом процессе и должен убедиться в отсутствии трафика, прежде чем разрешить системе переходить в состояние L1.
- Когда ядро обнаружит, что значение Power State изменилось и отлично от D0, автомат состояния управления питанием ядра, который отвечает за протокол высокого уровня управления питанием, выполняет следующие действия:
  - Блокирует последующие пересылки пакетов TLP.
  - Ждет завершения всех текущих пакетов TLP.
  - Ждет очистки буфера повтора (все переданные пакеты TLP подтверждены)
  - Ядро передает PM_ENTER_L1 пакеты DLLP пока не получит PM_REQ_ACK пакет DLLP от удаленного устройства.
  - Ядро переводит автомат состояний LTSSM в состояние L1.
  - При наличии TLP, ожидающего пересылки, и если автомат состояний LTSSM указывает на выход из состояния L1, ядро возвращается в состояние L0.

Программный переход в состояние L1 для нисходящих потов (корневой порт/нисходящий порт коммутатора) аналогичен, но ядро нисходящего порта выполняет запрос L1 восходящего порта:

- Ядро получает PM_ENTER_L1 пакет DLLP.
- Блокирует последующие пересылки пакетов TLP.
- Ждет завершения всех текущих пакетов TLP.
- Ждет очистки буфера повтора (все переданные пакеты TLP подтверждены)
- Пересылает PM_REQ_ACK пакет DLLP удаленному устройству, пока удаленное устройство не отключает передатчик
- Ядро переводит автомат состояний LTSSM в состояние L1.
- При наличии TLP, ожидающего пересылки, и если автомат состояний LTSSM указывает на выход из состояния L1, ядро возвращается в состояние L0.

PCI Express Device Capabilities Register Control[31:0] == mgmt_cfg_control[63:32] имеет следующие поля L1:

- Биты[11:9] – Endpoint L1 Acceptable Latency – Из базовой спецификации PCI Express, версия 2.1 раздел 7.8.3: “Это поле указывает допустимую общую задержку, которую может выдержать оконечная точка вследствие перехода из состояния L1 в состояние L0. Это, в основном, является косвенным отражением внутренней буферизации оконечной точки. Программа управления питанием использует указанную допустимую величину задержки L1 с задержками выхода, указанными (см. ниже) всеми компонентами, составляющими тракт данных от этой оконечной

точки до корневого порта корневого комплекса, чтобы определить, можно ли использовать вход ASPM L1 без потери производительности". Следует отметить, что уровень буферизации относится не к буферизации ядра Expresso 3.0, а к буферизации пользовательского приложения. Пользователи должны устанавливать это поле в соответствии с задержкой, допустимой для их приложения.

- 000 - Максимум 1 µs
- 001 - Максимум 2 µs
- 010 - Максимум 4 µs
- 011 - Максимум 8 µs
- 100 - Максимум 16 µs
- 101 - Максимум 32 µs
- 110 - Максимум 64 µs
- 111 - Не ограничено
- Устройства, отличные от оконечных точек, должны аппаратно устанавливать это поле в 000.

PCI Express Device Capabilities Register Control[31:0] == mgmt_cfg_control[111:80] имеет следующие поля L1:

- Биты[17:15] – L1 Exit Latency - Время, требуемое для полного перехода из L1 в L0:
  - 000 - менее 1µs
  - 001 - от 1 µs до 2 µs
  - 010 - от 2 µs до 4 µs
  - 011 - от 4 µs до 8 µs
  - 100 - от 8 µs до 16 µs
  - 101 - от 16 µs до 32 µs
  - 110 - 32 µs-64 µs
  - 111 - Более 64 µs
  - Задержки выхода могут быть существенно увеличены, если опорные тактовые сигналы PCI Express, используемые двумя устройствами в канале, общие или раздельные.
- Биты[11:10] – Active State Power Management (ASPM) Support должны быть установлены в "10" или "11", если поддержка L1 разрешена, и в "00", если запрещена.

Аппаратный автономный вход в состояние L1 (ASPM L1) инициируется только восходящими портами (оконечная точка, восходящий порт коммутатора). Функциональность ASPM L1 ядра должна быть разрешена и предложена в настройках соединения PCIe, и программа должна разрешить поддержку ASPM L1, чтобы осуществился аппаратный автономный вход в L1. Когда поддержка ASPM L1 присутствует и разрешена для восходящего порта, ядро запросит перевод канала в состояние L1 при помощи протокола ASPM L1, когда канал свободен (не пересылаются пакеты TLP или ACK/NAK пакеты DLLP) для запрашиваемого периода простоя.

Нисходящий порт (корневой порт/нисходящий порт коммутатора) получает запросы ASPM L1 от удаленного устройства и может выбирать, принять или отклонить запрос. Ядро примет запросы ASPM L1, когда отсутствуют пакеты TLP или ACK/NAK пакеты DLLP, ожидающие передачи4 в противном случае запрос будет отклонен.

1.3.10.2.1.6.4 L2

Ядро Expresso 3.0 поддерживает L2. Поддержка L2 разрешается в mgmt_cfg_constants[376]. Процесс перехода в состояние L2:

- Следует учитывать, что драйвер устройства участвует в этом процессе и должен убедиться в отсутствии трафика, прежде чем разрешить системе переходить в состояние L2.

- Программа может (но не обязательно) сначала перевести канал в состояние L1, а затем инициировать переход в состояние L2.
- Чтобы инициировать переход в состояние L2, системное программное обеспечение вызывает программу управления питанием, чтобы переслать PME_Turn_Off_Message ядру оконечной точки.
- Когда оконечная точка получит PME_Turn_Off_Message, автомат управления питанием ядра, который отвечает за протокол высокого уровня управления питанием, выполняет следующие действия:
  - Передает сообщение PME_TO_ACK, чтобы подтвердить PME_Turn_Off_Message, когда пользовательская аппаратная часть готова к выключению системы. Ядро позволяет два варианта генерации сообщения PME_TO_Ack посредством Disable Automatic PME_TO_Ack Message Generation == mgmt_cfg_constants[382]:
    - 0 (рекомендуется по умолчанию) - Ядро автоматически передает сообщение PME_TO_Ack в течение 256 тактов ядра после получения PME_Turn_Off_Message. В этот период времени пользовательская аппаратная часть должна хранить всю информацию о состоянии, которая уже не записана драйвером.
    - 1 – Пользовательская аппаратная часть отвечает за передачу требуемого сообщения PME_TO_Ack на интерфейс передатчика ядра. Эта опция позволяет пользовательской аппаратной части получить больше времени для сохранения информации состояния, хотя обычно этого не требуется; в решении перейти в режим L2 участвует драйвер устройства, и он уже должен был сохранить всю необходимую информацию состояния, прежде чем разрешить переход в L2. Задержка передачи сообщения PME_TO_Ack не должна быть слишком велика. Системной программе управления питанием разрешено реализовывать задержку (спецификация PCIe рекомендует от 1mS до 10mS), чтобы ограничить продолжительность ожидания подтверждения сообщения PME_Turn_Off, и она может отключать питание по истечению этого времени независимо от готовности пользовательской аппаратной части.
    - Независимо от того, какой метод используется, в промежутке между получением сообщения PME_Turn_Off_Message и передачей сообщения PME_TO_Ack пользователь должен записать всю информацию состояния, которая должна сохраниться после выключения питания. Потребуется определенное время, чтобы сообщение PME_TO_Ack было передано в корневой комплекс и обнаружено программой, а также дополнительное время, пока программа отключит питание, и поэтому общее время процесса неизвестно, и на него рассчитывать не следует.
      - После передачи сообщения PME_TO_Ack ядро блокирует передачу последующих пакетов TLP.
      - Ждет завершения всех текущих пакетов TLP.
      - Ждет очистки буфера повтора (все переданные пакеты TLP подтверждены)
      - Ядро передает PM_ENTER_L23 пакеты DLLP пока не получит PM_REQ_ACK пакет DLLP.
      - Ядро переводит автомат состояний LTSSM в состояние L2.
- Теперь следует отключить общее питание системы; ядро обычно получает общее питание и будет отключено; пользовательская логика управления питанием L2 (если присутствует, и разрешено использование дополнительного источника питания) продолжает работать, сохраняет контекст PME, и может снова активировать канал посредством Beacon или WAKE#, если это разрешено.

- При включении питания, или если в состоянии L2 питание ядра не было отключено, автомат состояний LTTSM начинает работу в состоянии DETECT. Физический уровень, канальный уровень и уровень транзакций выключены в состоянии DETECT, и устройство следует повторно инициализировать, как при сбросе.
- При выходе из состояния L2 пользовательская аппаратная часть управления питанием, работающая от дополнительного источника питания, должна восстановить контекст PME для ядра и инициировать генерацию сообщения PME, если пользовательская аппаратная часть управления питанием инициирует пробуждение системы посредством WAKE# или Beacon.

1.3.10.2.1.6.5 Использование дополнительного источника питания

Форм-факторы PCI Express позволяют подключать дополнительный источник питания, который может использоваться устройством для инициализации пробуждения системы или для управления системой, определяемого приложениями.

Потребность в дополнительном источнике питания указывается следующим образом:

- Поддержка управления питанием: Aux_Current != 0 или Power Management Capability: Реализован регистр данных
- Поддержка управления питанием: PME_Support[4] == 1(PME может генерироваться из D3cold)

В соответствии со спецификацией PCI Express и спецификацией интерфейса управления питанием шины PCI устройствам запрещено потреблять ток более 20mA от дополнительного источника питания, за исключением случаев, когда это разрешено программой.

Приборы могут потреблять менее 375mA, или ток, указанный в требованиях к дополнительному источнику питания, если: Auxiliary Power PM Enable == 1 или Power Management Control/Status: PME_En == 1.

1.3.10.2.1.6.6 Сохранение и восстановление информации для состояния L2

Ядро Expresso 3.0 поддерживает состояние D3cold с поддержкой от пользовательской логики, подключенной к дополнительному источнику питания. В состоянии D3cold питание ядра отключено, и оно не может хранить информацию или отслеживать события. Ядро переводится в состояние D3cold (питание отключено) или D3hot (питание сохраняется) через состояние канала L2. Ядро предоставляет контекстную информацию на интерфейс управления питанием, которая должна сохраняться в состоянии D3cold. При выходе из состояния D3cold пользователь передает эту информацию ядру через интерфейс управления питанием, включающий дополнительный сигнал, если пользователь осуществляет передачу от D3cold при помощи подтверждения WAKE# или Beacon.

1.3.10.2.1.7 Конфигурационное пространство PCI Express

Ядро Expresso 3.0 реализует тип заголовка 00 и тип заголовка 01 (когда присутствует поддержка корневого порта и/или коммутатора) для регистров конфигурации, включая настройки и расширенные настройки, как описано в PCI Express Base Specification, Rev 3.0, PCI Local Interface Specification Revision 3.0, иPCI Bus Power Management Interface Specification Revision 1.2.

В регистрах конфигурации типа 00 и типа 01 по-разному реализованы первые 64 байта конфигурационного пространства:

- Тип 00
  - Реализовано для оконечных точек
- Тип 01
  - Реализовано для корневых портов и порта моста/коммутатора

Настройки и расширенные настройки расположены по тем же самым адресам, независимо от того, какой тип заголовка реализован;

Регистры конфигурации ядра легко конфигурируются. В двухрежимных приложениях (корневой порт/оконечная точка и восходящий/нисходящий коммутатор) регистры конфигурируют сами себя в соответствии с режимом работы,

например, переходят с типа 00 на тип 01 при переключении с режима оконечной точки на режим корневого порта.

Регистры конфигурации обеспечивают стандартному программному обеспечению BIOS/OS для PCI/PCIe возможность обнаружить устройство, определить его возможности и сконфигурировать характеристики ядра. Поскольку количество приложений очень велико, регистры конфигурации легко конфигурируются. Порты mgtm_cfg_control и mgmt_cfg_constants содержат множество полей, определяющих реализацию регистра конфигурации PCIe и контрольное чтение установленных настроек. Более подробную информацию см. в описании портов mgtm_cfg_control и mgmt_cfg_constants Многие из характеристик, указанных в регистрах конфигурации, могут быть разрешены/запрещены или модифицированы с помощью этих портов.

Таблица 1.745 - Конфигурационный заголовок типа 00

Адрес	Байт3	Байт2	Байт1	Байт0
00	Device ID		Vendor ID	
04	Status		Command	
08	Class Code			Revision ID
0C	BIST	Header Type	Latency Timer	Cache Line Size
10	Base Address Register 0			
14	Base Address Register 1			
18	Base Address Register 2			
1C	Base Address Register 3			
20	Base Address Register 4			
24	Base Address Register 5			
28	Cardbus CIS Pointer			
2C	Subsystem ID		Subsystem Vendor ID	
30	Expansion ROM Base Address			
34	Зарезервировано			Capabilities Pointer
38	Зарезервировано			
3C	Max Latency	Min Grant	Interrupt Pin	Interrupt Line

Таблица 1.746 - Конфигурационный заголовок типа 01

Адрес	Байт3	Байт2	Байт1	Байт0
00	Device ID		Vendor ID	
04	Status		Command	
08	Class Code			Revision ID
0C	BIST	Header Type	Primary Latency Timer	Cache Line Size
10	Base Address Register 0			
14	Base Address Register 1			
18	Secondary Latency Timer	Subordinate Bus Number	Secondary Bus Number	Primary Bus Number
1C	Secondary Status		I/O Limit	I/O Base
20	Memory Limit		Memory Base	
24	Prefetchable Memory Limit		Prefetchable Memory Base	
28	Prefetchable Base старшие 32 бита			
2C	Prefetchable Limit старшие 32 бита			
30	I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits	
34	Зарезервировано			Capability Pointer
38	Expansion ROM Base Address			
3C	Bridge Control		Interrupt Pin	Interrupt Line

Таблица 1.747 - Настройки и расширенные настройки

Адрес	Байт3	Байт2	Байт1	Байт0
7B-40	PCI Express Capability (Настройка PCI Express)			
7F-7C	Зарезервировано			
87-80	Power Management Capability (Поддержка управления питанием)			
8F-88	Зарезервировано			
9B-90	MSI-X Capability (Поддержка MSI-X, реализована только в том случае, когда разрешена поддержка прерываний и поддержка MSI-X) **			
9F-9C	Зарезервировано			
AD-A0	MSI-X Capability (поддержка MSI, (реализована только в том случае, когда разрешена поддержка прерываний и поддержка MSI) **			
BF-AE	Зарезервировано			
C7-C0	VPD Capability (Поддержка VPD) **			
147-100	Advanced Error Reporting Capability (Расширенная настройка Advanced Error Reporting (AER), реализована только если поддержка AER разрешена)			
14F-148 (107-100)	ARI Capability (Поддержка ARI. Поддержка ARI начинается с 0x100 если поддержка AER не разрешена)			
173-150	NWL Vendor-Specific Extended Capability			
17f-174	Зарезервировано			
1AB-180	Secondary PCI Express Extended Capability (Дополнение к расширенным настройкам PCI Express. Реализовано только при наличии поддержки PCIe 3.0) **			
1AF-1AC	Зарезервировано			
1FF-1C0	SRIOV Capability (Поддержка SRIOV. Реализована только в том случае, если поддержка SRIOV присутствует и разрешена) **			
207-200	ATS Capability (Поддержка ATS) **			
20F-208	PASID Capability (Поддержка PASID) **			
21B-210	DSN Capability (Поддержка DSN) **			
26B-240	User Vendor-Specific Extended Capability (Дополнительные возможности пользователя, определяемые производителем) **			
2BF-280	Resizable BAR Capability (Поддержка BAR переменной длины) **			
38B-300	TPH Capability and ST Table (Поддержка TPH и ST Table) **			
38F-38C	Зарезервировано			
39F-390	Power Budgeting Capability (Поддержка распределения питания)			
3CF-3A0	Dynamic Power Allocation (DPA) Capability (Поддержка динамического распределения питания)			
3DF-3D0	L1 PM Substates Extended Capability (Расширенная поддержка подсостояний L1 PM)			
3E7-3E0	Latency Tolerance Reporting (LTR) Capability (Поддержка механизма обмена значениями задержек)			
3FF-3E8	Зарезервировано			
FFF-400	User Configuration Register Region (Область регистров конфигурации пользователя, см. интерфейс расширения регистров конфигурации)			

Таблица 1.748 - Регистры конфигурации типа 00

Адрес	Регистр конфигурации	Описание регистра
01 – 00	Vendor ID	Только для чтения: Это поле указывает производителя устройства. При чтении возвращает значение поля Vendor ID mgmt_cfg_constants.
03 – 02	Device ID	Только для чтения: Это поле указывает конкретное устройство. При чтении возвращает значение поля Device ID mgmt_cfg_constants.

Продолжение таблицы 1.749

Адрес	Регистр конфигурации	Описание регистра
05 – 04	Command Register	Биты регистра команд: Биты 10, 8, 6 и 2..0 предназначены для чтения/записи. Биты[15:11] = 00000. Не реализовано. Бит[10] – Interrupt Disable – Установленный бит запрещает прерывания; сброшенный бит - разрешает Бит[9] = 0. Не реализовано. Бит[8] – SERR Enable – Установленный бит разрешает сообщение корневому комплексу о фатальных и нефатальных ошибках, обнаруженных устройством. Бит[7] = 0. Не реализовано. Бит[6] – Parity Error Enable – Влияет на приведение ошибок PCI Express к унаследованным ошибкам PCI. Более подробную информацию см. в PCI Express Base Specification Rev1.1, раздел 6.2. Бит[5] = 0. Не реализовано. Бит[4] = 0. Не реализовано. Бит[3] = 0. Не реализовано. Бит[2] – Bus Master Enable – Если этот бит установлен, запросы памяти и ввода/вывода могут генерироваться только на интерфейсе уровня транзакций. Бит[1] – Memory Space Enable – Если этот бит установлен, ядро будет декодировать пакеты, чтобы обнаружить обращение к BAR памяти; сброшенный бит запрещает BAR памяти. Бит[0] – I/O Space Enable – Если этот бит установлен, ядро будет декодировать пакеты, чтобы обнаружить обращение к BAR ввода/вывода; сброшенный бит запрещает BAR ввода/вывода.
07 – 06	Status Register	Биты регистра статуса: Биты 15..11 и 8 предназначены для чтения/записи. Запись в 1 в местоположение бита очищает этот бит. Запись в 0 в местоположение бита не влияет на этот бит. Бит[15] – Устанавливается устройством при получении искаженного пакета TLP. Бит[14] – Устанавливается, когда устройство посылает сообщение ERR_FATAL или ERR_NONFATAL, и в командном регистре установлен бит SERR Enable. Бит[13] – Устанавливается, когда запрашивающее устройство получает завершение со статусом нераспознанного завершения (Unrecognized Request Completion) Бит[12] – Устанавливается, когда инициатор получает завершение со статусом аварийного завершения (Completer Abort Completion Status) Бит[11] – Устанавливается, когда устройство завершает запрос, используя статус аварийного завершения (Completer Abort Completion Status) Биты[10:9] = 00. Не реализовано. Бит[8] – Master Data Parity Error – Этот бит устанавливается запросчиком, если установлен его бит Parity Error Enable, а также получено завершение, отмеченное как искаженное, или запросчик исказил запрос записи. Биты[7:5] = 000. Не реализовано. Бит[4] = 1 указывает наличие списка настроек. Бит[3] – Interrupt Status – Отражает значение mgmt_interrupt Биты[2:0] = 000. Зарезервировано.
08	Revision ID	Только для чтения: Этот регистр указывает версию конкретного устройства. При чтении возвращает значение поля Revision ID mgmt_cfg_constants.
0B – 09	Class Code	Только для чтения: Код класса указывает обобщенную функцию устройства. При чтении возвращает значение поля Class Code mgmt_cfg_constants.

Продолжение таблицы 1.750

Адрес	Регистр конфигурации	Описание регистра
0C	0x0C: Cache Line Size	Чтение/запись: Cache Line Size не используется с PCI Express, но все же реализуется как регистр чтения/записи для целей совместимости унаследования.
0D	0x0D: Latency Timer	Только для чтения, возвращает 0x00.
0E	0x0E: Header Type	Только для чтения: Этот регистр возвращает значение 0x00, чтобы указать, что ядро соответствует стандартному для PCI распределению регистров, и что это устройство с одной функцией.
0F	0x0F: BIST	Не реализовано. При чтении возвращает 0x00.
13 – 10	Base Address Register 0	Чтение/запись: регистр базового адреса 0, регистр базового адреса 1, регистр базового адреса 2, регистр базового адреса 3, регистр базового адреса 4 и регистр базового адреса 5 информируют системное программное обеспечение о требованиях устройства к ресурсам, и поэтому они программируются для выделения устройству ресурсов памяти и ввода/вывода. Эти регистры отражают конфигурацию Base Address Cfg0, Base Address Cfg1, Base Address Cfg2, Base Address Cfg3, Base Address Cfg4, и Base Address Cfg5 в порту mgmt_cfg_constants. Более подробную информацию о Base Address Cfg0 см. в табл. 11-2 mgmt_cfg_constants Bit Fields и в PCI Local Bus Specification Revision 2.3, раздел 6.2.5.
17 – 14	Base Address Register 1	См. описание регистра базового адреса 0
1B – 18	Base Address Register 2	См. описание регистра базового адреса 0
1F – 1C	Base Address Register 3	См. описание регистра базового адреса 0
23 – 20	Base Address Register 4	См. описание регистра базового адреса 0
27 – 24	Base Address Register 5	См. описание регистра базового адреса 0
2B – 28	Card Bus CIS Pointer	Только для чтения: При чтении возвращает значение поля Cardbus CIS Pointer mgmt_cfg_constants.
2D – 2C	Subsystem Vendor ID	Только для чтения. Дополнительная информация о производителе. При чтении возвращает значение поля Subsystem Vendor ID mgmt_cfg_constants.
2F – 2E	Subsystem ID	Только для чтения. Дополнительная информация об устройстве. При чтении возвращает значение поля Subsystem ID mgmt_cfg_constants.
33 – 30	Expansion ROM Base Addr. Reg.	Информирует системное программное обеспечение о требованиях дополнительного ROM устройства к ресурсам, и поэтому он программируется для выделения устройству ресурсов памяти. Этот регистр отражает конфигурацию Expansion ROM Cfg в порте mgmt_cfg_constants. Более подробную информацию о Expansion ROM Cfg см. в табл. 11-2 mgmt_cfg_constants Bit Fields и в PCI Local Bus Specification Revision 2.3, раздел 6.2.5. Чтение/запись: Регистр базового адреса дополнительного ROM Биты[31:11] – Записываются для указания, где разместить эту область в пространстве памяти Биты[10:1] = 0..0 Зарезервированы Бит[0] = Устанавливается S/W, чтобы разрешить декодирование дополнительного ROM, и сбрасывается, чтобы запретить

Продолжение таблицы 1.751

Адрес	Регистр конфигурации	Описание регистра
34	Capabilities Pointer	Только для чтения: При чтении возвращает 0x40, что является начальным адресом PCI Express Capabilities Item.
37 - 35	Зарезервировано	Не реализовано. При чтении возвращает 0x000000.
3B – 38	Зарезервировано	Не реализовано. При чтении возвращает 0x00000000.
3C	Interrupt Line	Поддержка прерывания разрешается/запрещается полем Interrupt Enable порта mgmt_cfg_constants. Когда прерывания разрешены, Interrupt Line работает в режиме чтения/записи, и записывается программой конфигурации, чтобы указать, к какой линии системного прерывания подключен INTAn. Когда прерывания запрещены, Interrupt Line работает в режиме только для чтения и возвращает 0x00.
3D	Interrupt Pin	Поддержка прерывания разрешается/запрещается полем Interrupt Enable порта mgmt_cfg_constants. Когда прерывания разрешены, Interrupt Pin возвращает 0x01, указывая, что ядро реализует INTA#, и когда прерывания запрещены, Interrupt Pin возвращает 0x00, указывая, что прерывания не используются.
3E	Minimum Grant	Только для чтения. Возвращает 0x00.
3F	Maximum Latency	Только для чтения. Возвращает 0x00.

Таблица 1.752 - Настройки PCI Express

Адрес	Регистр конфигурации	Описание регистра
40	PCI Express Capability ID	Только для чтения = 0x10 (Начало PCI Express Capability Item)
41	Next Capability Pointer	Только для чтения = 0x80 (Начало Power Management Capability)
43-42	PCI Express Capabilities	Только для чтения Биты[15:14] – Зарезервированы = 00 Биты[13:9] – Interrupt Message Number[4:0]; MSI/MSI-X вектор прерываний ассоциируется с прерываниями, генерируемыми событиями регистра конфигурации (изменение полосы пропускания канала, ошибка корневого порта и т. д.) Бит[8] – Slot Implemented; только нисходящий коммутатор/корневой порт Биты[7:4] – Device/Port Type - Должен соответствовать применению ядра, поскольку запрограммированное значение разрешает/прячет регистры конфигурации и функциональность, которая применима только для некоторых типов устройств/портов: 0000 - оконечная точка PCI Express Требуется для приложений оконечной точки 0001 - Унаследованная оконечная точка PCI Express 0100 - корневой порт корневого комплекса PCI Express Требуется для приложений корневого порта 0101 - восходящий порт коммутатора PCI Express Требуется для восходящих портов коммутатора 0110 - нисходящий порт коммутатора PCI Express Требуется для нисходящих портов коммутатора 0111 - Мост от PCI Express к PCI/PCI-X 1000 - Мост PCI/PCI-X к PCI Express 1001 - интегрированная оконечная точка корневого комплекса 1010 – коллектор событий корневого комплекса Биты[3:0] – Capability Version - Для PCIe 3.0 должны быть 0x2

Продолжение таблицы 1.753

Адрес	Регистр конфигурации	Описание регистра
47-44	Device Capabilities Register	Только для чтения; Биты[14:6] и [2:0] переносятся напрямую из поля PCI Express Device Capabilities Register Control в порту mgmt_cfg_control. Биты[31:29] – Зарезервированы. Бит[28] - Function Level Reset Capability 1 - возможность присуствует 0 - возможность отсутствует Биты[27:26] – Captured Slot Power Limit Scale Биты[25:18] – Captured Slot Power Limit Value Биты[17:16] = 00. Зарезервировано. Бит[15] = 1. Ролевое сообщение об ошибках Бит[14] = 0 - Зарезервирован Бит[13] = 0 - Зарезервирован Бит[12] = 0 - Зарезервирован Биты[11:9] Допустимая задержка L1 для оконечной точки Биты[8:6] Допустимая задержка L0s для оконечной точки Бит[5] - Extended Tag Field Supported Биты[4:3] - Phantom Functions Supported Биты[2:0] - Поддержка максимального размера содержимого 000 - максимальный размер содержимого 128 байт 001 - максимальный размер содержимого 256 байт 010 - максимальный размер содержимого 512 байт 011 - максимальный размер содержимого 1024 байт 100 - максимальный размер содержимого 2048 байт 101 - максимальный размер содержимого 4096 байт 110 – Зарезервировано 111 - Зарезервировано
49-48	Device Control Register	Чтение/запись (подробную информацию о значении битов разрешения ошибок [3:0] см. в PCI Express Base Specification, Rev 1.1, раздел 6.2) Бит[15] - Bridge Configuration Retry Enable/Initiate Function Level Reset Биты[14:12] – Max Read Request Size; интерфейс передачи может не передать пакет TLP запроса чтения с большей длиной, чем указано в Max Read Request Size: o 000 == 128 байт o 001 == 256 байт o 010 == 512 байт o 011 == 1024 байт o 100 == 2048 байт o 101 == 4096 байт o 110 == Зарезервировано o 111 == Зарезервировано Бит[11] – Enable No Snoop Бит[10] – Aux Power PM Enable Бит[9] – Phantom Functions Enable Бит[8] – Extended Tag Field Enable Биты[7:5] – Max Payload Size; интерфейс передачи может не передать пакет TLP с содержимым с большей длиной, чем указано в Max Payload Size: o 000 == 128 байт o 001 == 256 байт o 010 == 512 байт o 011 == 1024 байт o 100 == 2048 байт o 101 == 4096 байт o 110 == Зарезервировано o 111 == Зарезервировано Бит[4] – Enable Relaxed Ordering Бит[3] – Unsupported Request Reporting Enable Бит[2] – Fatal Error Reporting Enable Бит[1] – Non-Fatal Error Reporting Enable Бит[0] – Correctable Error Reporting Enable

Продолжение таблицы 1.754

Адрес	Регистр конфигурации	Описание регистра
4B-4A	Device Status Register	Биты[15:4] только для чтения. Биты[3:0] очищаются записью 1 в соответствующее местоположение бита Биты[5:4] берутся прямо из поля PCI Express Device Status Register Control порта mgmt_cfg_control. Биты[15:6] = 0000000000. Зарезервировано Бит[5] – Transactions Pending Бит[4] – AUX Power Detected Бит[3] – Unsupported Request Detected Бит[2] – Fatal Error Detected Бит[1] – Non-Fatal Error Detected Бит[0] – Correctable Error Detected
4F-4C	Link Capabilities Register	Только для чтения. Биты[31:24] берутся прямо из поля PCI Express Device Link Capabilities Control порта mgmt_cfg_control. Биты[31:24] – Port Number Биты[22] = 1. ASPM Optionalit Compliance Бит[21] – Link Bandwidth Notification Capability == 1 при работе в качестве нисходящего порта; в остальных случаях 0 Бит[20] – Data Link Layer Active Reporting Capable == 1 при работе в качестве нисходящего порта; в остальных случаях 0 Бит[19] – Surprise Down Error Reporting Capable == 1 при работе в качестве нисходящего порта; в остальных случаях 0 Бит[18] = 0. Clock Power Management Биты[17:15] – L1 Exit Latency Биты[14:12] – L0s Exit Latency Биты[11:10] – Active State Power Management (ASPM) Support 00 – Нет поддержки ASMP 01 – Поддержка L0s 10 – Поддержка L1 11 – Поддержка L0s и L1 Биты[9:4] – Maximum Link Width 000001 - x1 000010 - x2 000100 - x4 001000 - x8 010000 - x16 Биты[3:0] – Maximum Link Speed 0001 (2.5GT/s) 0010 (5GT/s) 0011 (8GT/s)
51-50	Link Control Register	Чтение/запись Биты[15:12] = 0000. Зарезервировано. Бит[11] – Link Autonmous Bandwidth Interrupt Enable Бит[10] – Link Bandwidth Management Interrupt Enable Бит[9] – Hardware Autonomous Width Disable Бит[8] = 0. Enable Clock Power Management Бит[7] – Extended Sync Бит[6] – Common Clock Configuration Бит[5] – Retrain Link Бит[4] – Link Disable Бит[3] – Read Completion Boundary (RCB) o 0 == 64 байт o 1 == 128 байт Бит[2] = 0. Зарезервировано. Биты[1:0] – Active State Power Management (ASPM) Control 00 – Запрещено 01 – L0s разрешено 10 – L1 разрешено 11 – L0s и L1 разрешены

Продолжение таблицы 1.755

Адрес	Регистр конфигурации	Описание регистра
53-52	Link Status Register	Только для чтения; Бит[12] берется прямо из поля PCI Express Device Status Register Control порта mgmt_cfg_control. Бит[15] – Link Autonomous Bandwidth Status Бит[14] – Link Bandwidth Management Status Бит[13] – Data Link Layer Active Бит[12] – Slot Clock Configuration Бит[11] – Link Training Бит[10] = 0. Зарезервировано. Биты[9:4] Negotiated Link Width – указывает количество используемых в данный момент линий 010000 = x16 001000 = x8 000100 = x4 000010 = x2 000001 = x1 Биты[3:0] Link Speed o 0001 (2.5 GT/s) o 0010 (5.0 GT/s) o 0011 (8.0 GT/s)
57-54	Slot Capabilities Только корневой порт/коммутатор	Обычно только для чтения; разрешена запись, когда HW Init Write Enable == 1 (см. NWL Vendor-Specific Extended Capability); начальные предложения управляются mgmt_cfg_control[543:512] Биты[31:19] - Physical Slot Number Бит[18] - No Command Completed Support Бит[17] – Electromechanical Interlock Present Биты[16:15] – Slot Power Limit Scale[1:0] Биты[14:7] – Slot Power Limit Value[7:0] Бит[6] – Hot-Plug Capable Бит[5] – Hot-Plug Surprise Бит[4] – Power Indicator Present Бит[3] – Attention Indicator Present Бит[2] – MRL Sensor Present Бит[1] – Power Controller Present Бит[0] – Attention Button Present
59-58	Slot Control Только корневой порт/коммутатор	Только для чтения Биты[15:13] = 0. Зарезервировано. Бит[12] - Data Link Layer State Changed Enable Бит[11] = 0. Electromechanical Interlock Control Бит[10] - Power Controller Control Бит[9:8] - Power Indicator Control Бит[7:6] - Attention Indicator Control Бит[5] - Hot-Plug Interrupt Enable Бит[4] - Command Completed Interrupt Enable Бит[3] – Presence Detect Changed Enable Бит[2] – MRL Sensor Changed Enable Бит[1] – Power Fault Detected Enable Бит[0] - Attention Button Pressed Enable

Продолжение таблицы 1.756

Адрес	Регистр конфигурации	Описание регистра
5b-5a	Slot Status Только корневой порт/коммутатор	Только для чтения Биты[15:9] = 0. Зарезервировано. Бит[8] - Data Link Layer State Changed Бит[7] - Electromechanical Interlock Status Бит[6] - Presence Detect State Бит[5] - MRL Sensor State Бит[4] - Command Completed Бит[3] – Presence Detect Changed Бит[2] – MRL Sensor Changed Бит[1] – Power Fault Detected Бит[0] - Attention Button Pressed
5d-5c	Root Control Только корневой порт	Чтение/запись Биты[15:5] = 0. Зарезервировано. Бит[4] = CRS Software Visibility Enable Бит[3] – PME Interrupt Enable Бит[2] – System Error on Fatal Error Enable Бит[1] – System Error on Non-Fatal Error Enable Бит[0] - System Error on Correctable Error Enable
5f-5e	Root Capabilities Только корневой порт	Только для чтения; Бит[16] – Запись 1 для очистки. Биты[15:1] = 0. Зарезервировано Бит[0] = 1. Поддержка CRS Software Visibility
63-60	Root Status Только корневой порт	Только для чтения; Бит[16] – Запись 1 для очистки. Биты[31:18] = 0. Зарезервировано Бит[17] – PME Pending Бит[16] – PME Status Биты[15:0] – PME Requester ID
67-64	Device Capabilities 2	Только для чтения Биты[31:24] = 0. Зарезервировано Биты[23:22] = 00. Max End-End TLP Prefixes Бит[21] = 0. End-End TLP Prefix Supported Бит[20] = 0. Extended Fmt Field Supported Бит[19:18] = 00. OBFF Supported □ Биты[17:14] = 0000. Reserved Биты[13:12] = 00. TPH Completer Supported Бит[11] = LTR Mechanism Supported Бит[10] = 0. No RO-enabled PR-PR Passing Бит[9] = 0. 128-bit CAS Completer Supported Бит[8] = 0. 64-bit AtomicOp Completer Supported Бит[7] = 0. 32-bit AtomicOp Completer Supported Бит[6] = 0. AtomicOp Routing Supported Бит[5] = 0. ARI Forwarding Supported Бит[4] – Completion Timeout Disable Supported Биты[3:0] – Completion Timeout Ranges Supported

Продолжение таблицы 1.757

Адрес	Регистр конфигурации	Описание регистра
69-68	Device Control 2	Чтение/запись, если mgmt_cfg_control: PCIe Device Capabilities 2[4] == 1, в противном случае только для чтения, при чтении возвращает 0x00000000. Бит[15] – End-End TLP Prefix Blocking Биты[14:13] – OBFF Enable; not supported Биты[12:11] = 00. Reserved. Бит[10] – LTR Mechanism Enable Бит[9] – IDO Completion Enable Бит[8] – IDO Request Enable Бит[7] – AtomicOp Egress Blocking; not supported Бит[6] – AtomicOp Request Enable; not supported Бит[5] – ARI Forwarding Enable Бит[4] – Completion Timeout Disable - Устанавливается системным ПО, чтобы запретить данному устройству генерировать тайм-аут завершения. Пользователи должны запретить генерацию ошибки тайм-аута завершения, когда установлен этот бит (Completion Timeout Disable == mgmt_cfg_status[836]). Биты[3:0] – Completion Timeout Value - Устанавливается системным ПО, чтобы выбрать диапазон тайм-аута завершения, который должны использовать пользователи, реализующие тайм-ауты завершения. Более подробную информацию см. в табл. 7-24 спецификации PCI Express (Completion Timeout Value[3:0] == mgmt_cfg_status[835:832]).
6B-6A	Device Status 2	Зарезервировано PCI SIG для будущего использования. При чтении возвращает 0x00000000.
6F-6C	Link Capabilities 2	Только для чтения Биты[15:9] = 0. Зарезервировано. Бит[8] – Crosslink Supported Биты[7:1] – Supported Link Speeds Vector Бит[0] = 0. Зарезервировано
71-70	Link Control 2	Чтение/запись Бит[15:12] – Compliance Preset/De-emphasis[3:0] Бит[11] – Compliance SOS Бит[10] – Enter Modified Compliance Биты[9:7] – Transmit Margin Бит[6] – Selectable De-emphasis Бит[5] – Hardware Autonomous Speed Disable Бит[4] – Enter Compliance Биты[3:0] – Target Link Speed[3:0] o 0001 (2,5 GT/s) o 0010 (5,0 GT/s) o 0011 (8,0 GT/s)
73-72	Link Status 2	Только для чтения; Бит[5] – запись 1 для очистки. □ Биты[15:6] = 0000000000. Зарезервировано. Бит[5] – Link Equalization Reset Бит[4] – Equalization Phase 3 Successful Бит[3] – Equalization Phase 2 Successful Бит[2] – Equalization Phase 1 Successful Бит[1] – Equalization Complete Бит[0] – Current De-emphasis Level o 1==-3.5dB o 0==-6dB
77-74	Slot Capabilities 2 Только корневой порт/коммутатор	Зарезервировано PCI SIG для будущего использования. При чтении возвращает 0x00000000.
79-78	Slot Control 2 Только корневой порт/коммутатор	Зарезервировано PCI SIG для будущего использования. При чтении возвращает 0x00000000.

Продолжение таблицы 1.758

Адрес	Регистр конфигурации	Описание регистра
7b-7a	Slot Status 2 Только корневой порт/коммутатор	Зарезервировано PCI SIG для будущего использования. При чтении возвращает 0x00000000.
7F-7C	Зарезервировано	При чтении возвращает 0x00000000.

Таблица 1.759 - Расширенные настройки NWL Vendor-Specific

Адрес	Регистр конфигурации	Описание регистра
153-150	Vendor-Specifc PCI Express Extended Capability Header	Начало расширенных настроек NWL Vendor-Specific Extended Capability (VSEC) Биты[31:20] - Только для чтения. Указатель следующей расширенной настройки в списке. Биты[19:16] – Только для чтения = 0x1 == Capability Version Биты[15:0] - Только для чтения = 0x000B == Capability ID
	Vendor-Specific Header	Только для чтения Биты[31:20] – VSEC Length = 0x24 (36 байт) Биты[19:16] – VSEC Rev = 0x1 Биты[15:0] – VSEC ID == mgmt_cfg_control[575:560]
15B-158	HW Init	Чтение/запись Биты[31:1] = 0. Зарезервировано Бит[0] – HW Init Write Enable - Используется для разрешения программе записи в некоторые регистры конфигурации типа “HWInit”, запись в которые в остальных случаях запрещена; значение по умолчанию == 0 1 – Разрешение записи HW Init - Позволяет определенным полям HW Init записываться программой; действительно только для регистров конфигурации в этом документе, для которых указана возможность записи при HW Init Write Enable == 1 (например, PCI Express Capability: Slot Capabilities) 0 – Запрещение записи HW Init
15F-15C	Link Power Down Только корневой порт/нисходящий порт коммутатора	Чтение/запись - Используется системным ПО в приложении корневой порт или нисходящий порт коммутатора для передачи на PCI Express сообщения PME_Turn_Off для запроса, чтобы нисходящая иерархия PCI Express подготовилась к отключению питания • Биты[31:3] = 0. Зарезервировано • Бит[2] – L2 Request Timeout; указывает, когда запрос L2 завершен в результате тайм-аута; L2 Request Timeout очищается (0), когда в L2_Request записывается 1, или когда 1 записывается в этот регистр; L2 Request Timeout устанавливается (1), сообщение PME_TO_ACK не получено в ответ на переданное PME_Turn_Off в течение ожидаемых 100 mS (100 uS для симуляции, когда mgmt_short_sim == 1) окна тайм-аута Бит[1] – L2 Request Status; указывает, когда запрос L2 Request завершился либо в результате получения ожидаемого ответного сообщения PME_TO_Ack, или в результате тайм-аута; L2 Request Status очищается (0) когда в L2_Request записывается 1, или когда 1 записывается в этот регистр; L2 Request Status устанавливается (1), когда получено сообщение PME_TO_ACK или происходит тайм-аут Бит[0] – L2 Request; записывается 1, чтобы сообщение PME_Turn_Off передавалось по нисходящей в иерархии PCI Express; после того, как все нисходящие устройства подготовились к отключению питания, ядро должно получить в ответ сообщение PME_TO_Ack, указывающее, что нисходящая иерархия PCIe готова к отключению питания; L2 Request остается установленным до тех пор, пока не будет получено сообщение PME_TO_ACK или не произойдет тайм-аут

Продолжение таблицы 1.760

Адрес	Регистр конфигурации	Описание регистра
163-160	Автономное восстановление, скорость и ширина канала	Чтение/запись - Используется системным ПО в порте US для автономного изменения скорости, изменения ширины канала, или перехода к восстановлению Биты[31:16] – Lane Width Mask. 1 указывает, что линия может быть использована [16] = Lane 0 .. [31] = Lane 15. Биты[15:12] – Зарезервированы □ Биты[11:8] – Target Speed. (1=2.5G, 2=5G,3=8G). Биты[7:3] – Зарезервированы Бит[2] – Autonomous Entry to Recovery Command. При установке в 1 оба бита [1] и [0] должны быть установлены в 0. Установка этого бита в 1 приведет к немедленному переходу канала в режим восстановления. Это может использоваться для посылки указания на запрос выравнивания link partner при помощи соответствующих битов mgmt_cfg_8g_constants. Бит[1] – Autonomous Width Change Command. При установке в 1 канал перейдет к режиму восстановления, чтобы изменить ширину канала при помощи поля Lane Width Mask. Этот бит игнорируется, если в регистре Link Control было установлено HW Autonomous Width Disable. Бит[0] – Autonomous Speed Change Command. При установке в 1 канал перейдет к режиму восстановления, чтобы изменить скорость канала при помощи поля Target Speed. Этот бит игнорируется, если в регистре Link Control 2 было установлено HW Autonomous Speed Disable. Сигнал об изменении скорости и ширины канала может подаваться одновременно, однако сигнал о входе в режим восстановления должен подаваться отдельно от сигналов об изменении скорости или ширины канала.
173-164	Зарезервировано	Зарезервировано

Таблица 1.761 - Расширенные настройки ARI

Адрес	Регистр конфигурации	Описание регистра
14B-148 или 103-100	ARI Capability Extended Capability Header	Начало Расширенных настроек ARI – Только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x000E == Capability ID
14D-14C или 105-104	ARI Capability Register	Только для чтения Биты[15:8] – Next Function Number = 0 (не реализовано) Биты[7:2] – Зарезервированы = 0 Бит[1] - ACS Function Groups Capability = 0 (не реализовано) Бит[0] – MFVC Function Groups Capability = 0 (не реализовано)
14F-14E или 107-106	ARI Control Register	Только для чтения Биты[15:7] – Reserved Бит[6:4] – Function Group = 0 (не реализовано) Биты[3:2] – Зарезервированы = 0 Бит[1] - ACS Function Groups Enable = 0 (не реализовано) Бит[0] – MFVC Function Groups Enable = 0 (не реализовано)

Продолжение таблицы 1.762

Адрес	Регистр конфигурации	Описание регистра
14B-148 или 103-100	ARI Capability Extended Capability Header	Начало Расширенных настроек ARI – Только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x000E == Capability ID

Таблица 1.763 - Дополнение к расширенным настройкам PCI Express

Адрес	Регистр конфигурации	Описание регистра
183-180	Secondary PCI Express Extended Capability Header	Начало дополнения к расширенным настройкам PCI Express; это дополнение присутствует только в том случае, если в исполнении разрешена поддержка PCIe 3.0. При отсутствии поддержки AER это дополнение располагается со сдвигом 0x100. Биты[31:20] - Только для чтения. Указатель следующей настройки в списке. Биты[19:16] – Только для чтения = 0x1 == Capability Version Биты[15:0] - Только для чтения = 0x0019 == Capability ID
187-184	Link Control 3	Чтение/запись Биты[31:2] – Зарезервированы = 0 Бит[1] - Link Equalization Request Interrupt Enable Бит[0] - Perform Equalization
18B-188	Lane Error Status	Только для чтения. Указывает статус ошибки конкретной линии Биты[31:NUM_LANES] – Зарезервированы = 0 Бит[Lane#] – 1 == Обнаружена ошибка на линии[[Lane#]; 0 == нет ошибки
1AB-18C	Lane Equalization Control Register	Только для чтения Поля управления и статуса для выравнивания канала; 16 бит на линию, начиная с Lane[0]; линии с большими номерами расположены по мере возрастания адреса Формат для каждой линии: Бит[15] – Зарезервирован= 0 Биты[14:12] - Upstream Port Receiver Preset Hint Биты[11:8] – Upstream Port Transmitter Preset Бит[7] – Зарезервирован= 0 Биты[6:4] - Downstream Port Receiver Preset Hint Биты[3:0] –Downstream Port Transmitter Preset

Таблица 1.764 - Область расширенных настроек SRIOV

Адрес	Регистр конфигурации	Описание регистра
1C3-1CO	SRI-OV Capability Extended Capability Header	Начало расширенных настроек SR-IOV – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x0010 == Capability ID
1C7-1C4	SR-IOV Capabilities	Только для чтения Биты[31:21] - VF Migration Interrupt Message Number = 0 (не не реализовано) Биты[20:2] – Зарезервированы = 0 Бит[1] - ARI Capable Hierarchy Preserved = 1 Бит[0] - VF Migration Capable = 0 (не реализовано)
1C9-1C8	SR-IOV Control	Чтение/запись Биты[15:5] – Зарезервированы = 0 (только для чтения) Бит[4] – ARI Capable Hierarchy (чтение/запись) Все VF должны быть расположены на шине с тем же номером, что и физические функции; значение в ARI Capable Hierarchy не влияет на распределение VF Offset и VF Stride Бит[3] – VF MSE (чтение/запись) – Пространство памяти, доступно для виртуальных функций; должен быть установлен в 1, прежде чем VF сможет принимать запросы памяти, предназначенные для BAR виртуальных функций. Бит[2] – VF Migration Interrupt Enable (чтение/запись) – Не поддерживается Бит[1] – VF Migration Enable = 0 (только для чтения) – Не поддерживается Бит[0] – VF Enable (чтение/запись) – 1 == разрешение виртуальных функций; 0 == запрет; должен быть установлен в 1, прежде чем VF сможет отвечать на транзакции PCI Express.
1CB-1CA	SR-IOV Status	Только для чтения Биты[15:1] – Зарезервированы Бит[0] - VF Migration Status = 0 (не реализовано)
1CD-1CC	Initial VFs	Только для чтения Биты[15:8] = 0 Биты[7:0] – Значение mgmt_cfg_constants : TotalVFs[7:0]
1CF-1CE	TotalVFs	Только для чтения Биты[15:8] = 0 Биты[7:0] – Значение mgmt_cfg_constants : TotalVFs[7:0]
1D1-1D0	Num VFs	Чтение/запись Биты[15:8] = 0 Биты[7:0] – Номер разрешенной VF (когда VF Enable также 1)
1D2	Function Dependency Link	Только для чтения Биты[15:0] = 0 (No dependcies)
1D3	Зарезервировано	Только для чтения Биты[15:0] = 0
1D5-1D4	VF Offset	ToТолько для чтения Биты[15:0] = VFOffset[pf] где pf == # физической функции VF Offset[pf] = ((номер PF) - pf) + (pf * ((номер VF поддерживаемой для PF)-1))
1D7-1D6	VF Stride	Только для чтения Биты[15:0] = 0x0001 (VF последовательны)
1D9-1D8	Зарезервировано	Только для чтения Биты[15:0] = 0
1DB-1DA	VF Device ID	Только для чтения Биты[15:0] – Значение mgmt_cfg_constants : VF Device ID[15:0]

Продолжение таблицы 1.765

Адрес	Регистр конфигурации	Описание регистра
1DF-1DC	Supported Page Sizes	Только для чтения Биты[31:16] = 16'h0 (Размеры страниц >= 256 MB не поддерживаются) Биты[15:0] – Значение mgmt_cfg_constants : Support Page Sizes Low[15:0]; указывает, какие размеры страниц поддерживаются Размер страницы 2n+12 поддерживается, если бит установлен. Например, Бит[0] == 4KB, Бит[1] == 8KB, … , Бит[15] == 128 MB
1E3-1E	System Page Size	Чтение/запись Биты[31:0] = System Page Size[31:0] Может быть установлен только один бит, который указывает размер страницы, используемый системным ПО для разметки областей BAR памяти VF. System Page Size[n] == 1 соответствует размеру страницы 2n+12. Например, Бит[0] == 4KB, Бит[1] == 8KB, … , Бит[15] == 128 MB
1E7-1E4	VF BAR0	Чтение/запись Биты[31:0] = VF Base Address Register 0 Конфигурируется при помощи mgmt_cfg_constants : VF Base Address Cfg0[31:0]
1EB-1E8	VF BAR1	Чтение/запись Биты[31:0] = VF Base Address Register 1 Конфигурируется при помощи mgmt_cfg_constants : VF Base Address Cfg1[31:0]
1EF-1EC	VF BAR2	Чтение/запись Биты[31:0] = VF Base Address Register 2 Конфигурируется при помощи mgmt_cfg_constants : VF Base Address Cfg1[31:0]
1F3-1F0	VF BAR3	Чтение/запись Биты[31:0] = VF Base Address Register 3 Конфигурируется при помощи mgmt_cfg_constants : VF Base Address Cfg1[31:0]
1F7-1F4	VF BAR4	Чтение/запись Биты[31:0] = VF Base Address Register 4 Конфигурируется при помощи mgmt_cfg_constants : VF Base Address Cfg1[31:0]
1FB-1F8	VF BAR5	Чтение/запись Биты[31:0] = VF Base Address Register 5 Конфигурируется при помощи mgmt_cfg_constants : VF Base Address Cfg1[31:0]
1FF-1FC	VF Migration State Array Offset	Только для чтения Биты[31:0] = 0 (не реализованы)

Таблица 1.766 - Расиширенные настройки ATS

Адрес	Регистр конфигурации	Описание регистра
203-20O	ATS Capability Extended Capability Header	Начало расширенных настроек ATS – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 — Capability Version
205-204	ATS Capability Register	Только для чтения Биты[4:0] - Invalidate Queue Depth Бит[5]Page Aligned Request Биты[15:6] – Зарезервированы
207-206	ATS Control Register	Чтение/запись Биты[4:0] – Smallest Translation Unit Биты[14:5] – Зарезервированы Бит[15] – Разрешено

Таблица 1.767 - Расширенные настройки PASID

Адрес	Регистр конфигурации	Описание регистра
20B-208	PASID Capability Extended Capability Header	Начало расширенных настроек PASID– только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x001B == Capability ID
20D-20C	PASID Capability Register	Только для чтения Бит[0] - Зарезервировано Бит[1] – Execute Permission Supported Бит[2] – Privileged Mode Supported Биты[7:3] – Зарезервированы Биты[12:8] – Max PASID Width Биты[15:13] - Зарезервировано
20F-20E	PASID Control Register	Чтение/запись Бит[0] – Разрешено Бит[1] – Exexcute Permission Enable Бит[2] – Privileged Mode Enable Биты[15:3] – Зарезервированы

Таблица 1.768 - Расширенные настройки DSN

Адрес	Регистр конфигурации	Описание регистра
213-210	DSN Capability Extended Capability Header	Начало расширенных настроек DSN – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x0003 == Capability ID
21B-214	DSN Serial NUmber	Только для чтения Биты[63:0] – DSN Serial Number

Таблица 1.769 -Расширенные настройки пользователя VSEC

Адрес	Регистр конфигурации	Описание регистра
243-240	VSEC Capability Extended Capability Header	Начало расширенных настроек пользователя VSEC – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x000B == Capability ID
247-244	VSEC Header	Только для чтения Биты[31:20] – VSEC Length = 0x2c (44 байта) Биты[19:16] – VSEC Rev = 0x0 Биты[15:0] – VSEC ID == mgmt_cfg_control[607:592]
24B-248	VSEC Control	Чтение/запись Биты[n-1:0] – Input Bits Биты[31:n] – Output Bits Этот регистр содержит комбинацию входных и выходных битов, которые будут появляться как порты на ядре. Количество входных битов n указано в mgmt_cfg_control[589:584].
26B-24C	VSEC Data Registers	Чтение/запись Биты[255:0] – Данные общего назначения

Таблица 1.770 - Настройка BAR с изменяемым регистром

Адрес	Регистр конфигурации	Описание регистра
283-280	Resizable BAR Extended Capability Header	Resizable BAR Capability Header – Только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x0015 == Capability ID
287-284	Resizable BAR Capability(0)	Только для чтения Биты[31:24] – Зарезервированы Биты[23:4] – Когда бит установлен, то BAR, на который указывает указатель BAR в управляющем регистре, будет работать с BAR размером в 2^(n+16) байт. Например, Бит[4] = 2^20 Bytes = 1MB Биты[3:0] – Зарезервированы
28B-288	Resizable Bar Control Register(0)	Только для чтения Биты[31:13] – Зарезервированы Чтение/запись Биты[12:8] – BAR Size. Для определения размера этого BAR должно использоваться закодированное значение. Только для чтения Биты[7:5] – Количество Resizable BARs. Значение должно быть в диапазоне от 1 до 6. Эти биты действительны только в Resizable BAR Control Register (0). В управляющих регистрах (1) и выше эти биты зарезервированы. Биты[2:0] – Указатель BAR для этого BAR: 0 = BAR расположен со сдвигом 0x10 1 = BAR расположен со сдвигом 0x14 2 = BAR расположен со сдвигом 0x18 3 = BAR расположен со сдвигом 0x1C10 4 = BAR расположен со сдвигом 0x20 5 = BAR расположен со сдвигом 0x24 Остальные значения зарезервированы. Для 64-битового BAR этот индекс должен указывать на младшее DWORD.
2BF-28C	Resizable BAR Capability and Control Registers (1..6)	См. Resizable BAR Capability (0) См. Resizable Bar Control Register(0) Количество реализованных регистров BAR Capability and Control зависит от установки “Number of Resizable BARs” управляющего регистра (0).

Таблица 1.771 - Область расширенных настроек TPH

Адрес	Регистр конфигурации	Описание регистра
303-300	TPH Capability Extended Capability Header	Начало области расширенных настроек TPH – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x0017 == Capability ID
307-304	TPH Capability Register	Только для чтения Бит[0] – No ST Mode Supported Бит[1] – Interrupt Vector Mode Supported Бит[2] – Device Specific Mode Supported Биты[7:3] – Зарезервированы Бит[8] – Extended TPH Requester Supported Биты[10:9] – ST Table Location Биты[26:16] – ST Table Size Биты[31:27] - Зарезервированы
30B-308	TPH Control Register	Чтение/запись Биты[2:0] – ST Mode Select Биты[7:3] – Зарезервированы Биты[9:8] – TPH Requester Enable Биты[31:10] – Зарезервированы

Продолжение таблицы 1.772

Адрес	Регистр конфигурации	Описание регистра
38B-30C	ST Table	Чтение/запись Эта таблица состоит из 64 16-битовых значений: Биты[7:0] – ST Lower Биты[15:8] – ST Upper

Таблица 1.773. Настройки распределения питания

Адрес	Регистр конфигурации	Описание регистра
393-390	Power Budgeting Capability Extended Capability Header	Начало настроек распределения питания – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x0004 == Capability ID
394	Data Select Register	Чтение/запись Биты[7:0] – Data Select Register
397-395	Зарезервировано	Зарезервировано
39B-398	Data Register	Только для чтения Биты[31:21] - Зарезервированы □ Биты[20:18] – Power Rail (0:12V,1:3.3V,2:1.5/1.8V,7:Thermal) Биты[17:15] – Type (0:PME Aux,1:Aux,2:Idle,3:Sustained,7:Max) □ Биты[14:13] – PM State (0:D0,1:D1,2:D2,3:D3) Биты[12:10] – PM Sub State (0:Default,others: Device Specific) Биты[9:8] – Data Scale (0:1x,1:0.1x,2:0.01x,3:0.001x) Биты[7:0] – Base Power
39C	Capabilities Register	Только для чтения Биты[7:1] – Зарезервированы Бит[0] - System Allocated: Устанавливается в 1 для указания, что распределение питания устанавливается системой, и значения из регистра данных НЕ используются для распределения питания системы. Устанавливается в 0 для указания, что значения из регистра данных должны использоваться для распределения питания системы.

Таблица 1.774. Настройки динамического распределения питания (DPA)

Адрес	Регистр конфигурации	Описание регистра
3A3-3A0	DPA Capability Extended Capability Header	Начало настройки DPA – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x0016 == Capability ID
3A7-3A4	DPA Capability Register	Только для чтения Биты[31:24] – Transition Latency Value1 (xlcy1) Биты[23:16] – Transition Latency Value0 (xlcy0) Биты[15:14] – Зарезервированы Биты[13:12] – Power Allocation Scale (PAS) Биты[11:10] – Зарезервированы Биты[9:8] – Transition Latency Unit (tlunit) Биты[7:5] – Зарезервированы Биты[4:0] – Substate_Max
3AB-3A8	DPA Latency Indicator Register	Только для чтения Биты[31:Substate_Max+1] – Зарезервированы Биты[Substate_Max:0] – Transition Latency Indicator Bits

Продолжение таблицы 1.775

Адрес	Регистр конфигурации	Описание регистра
3AD-3AC	DPA Status Register	Только для чтения Биты[15:9] – Зарезервированы Чтение, запись 1 для очистки Биты[8] – Substate Control Enabled Только для чтения Биты[7:0] – Substate Status
3EF-3AE	DPA Control Register	Только для чтения Биты[15:5] – Зарезервированы Чтение/запись Биты[4:0] – Substate Control
3CF-3B0	DPA Power Allocation Array	Только для чтения Биты[7:0] – Substate Power Allocation Register Адрес 3B0 для подсостояния 0 Адрес 3B1 для подсостояния 1, и т. д. до подсостояния Substate_Max

Таблица 1.776 - Расширенная надстройка подсостояний L1 PM

Адрес	Регистр конфигурации	Описание регистра
3D3-3D0	L1 PM Substates Capability Extended Capability Header	Начало расширенных настроек L1 PM Substates – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x001E == Capability ID
3D7-3D4	L1 PM Substates Capabilities Register	HwInit Биты[31:24] – Зарезервированы Биты[23:19] – Port TPOWER_ON Value Bits [18] - Зарезервированы Биты[17:16] – Port TPOWER_ON Scale Биты[15:8] – Port Common_Mode_Restore_Time (в μs) Биты[7:5] – Зарезервированы Бит[4] – L1 PM Substates Supported Бит[3] – ASPM L1.1 Supported Бит[2] – ASPM L1.2 Supported Бит[1] – PCI-PM L1.1 Supported Бит[0] – PCI-PM L1.2 Supported
3DB-3D8	L1 PM Substates Control 1 Register	Чтение/запись Биты[31:29] – LTR_L1.2_THRESHOLD_Scale Биты[28:26] – Зарезервированы Биты[25:16] – LTR_L1.2_THRESHOLD_Value Биты[15:8] – Common_Mode_Restore_Time Биты[7:4] – Зарезервированы Бит[3] – ASPM L1.1 Enable Бит[2] – ASPM L1.2 Enable Бит[1] – PCI-PM L1.1 Enable Бит[0] – PCI-PM L1.2 Enable
3DF-3DC	L1 PM Substates Control 2 Register	Чтение/запись Биты[31:8] – Зарезервированы Биты[7:3] – TPOWER_ON Value Бит[2] – Зарезервирован Биты[1:0] – TPOWER_ON Scale

Таблица 1.777 - Настройка механизма обмена значениями задержек (LTR)

Адрес	Регистр конфигурации	Описание регистра
3E3-3E0	LTR Capability Extended Capability Header	Начало расширенной настройки LTR – только для чтения Биты[31:20] - Указатель следующей расширенной настройки в списке. Биты[19:16] = 0x1 == Capability Version Биты[15:0] = 0x0018 == Capability ID
3E5-3E4	Max Snoop Latency Register	Чтение/запись Биты[15:13] - Зарезервированы Биты[12:10] – Max Snoop LatencyScale Биты[9:0] – Max Snoop LatencyValue
3E7-3E6	Max No-Snoop Latency Register	Чтение/запись Биты[15:13] = Зарезервированы Биты[12:10] – Max No-Snoop LatencyScale Биты[9:0] – Max No-Snoop LatencyValue

Таблица 1.778. Область расширенных настроек пользователя

Адрес	Регистр конфигурации	Описание регистра
FFF-400	User Region	Это область опционально реализуется пользователем. Чтение возвращает значение cfg_exp_rd_data.

1.3.10.2.1.8 Ядро Expresso DMA Bridge

Ядро Expresso DMA Bridge реализует высокопроизводительный прямой доступ в память и мост между PCI Express и AXI. Предназначено для таких применений PCI Express, как оконечная точка и корневой порт.

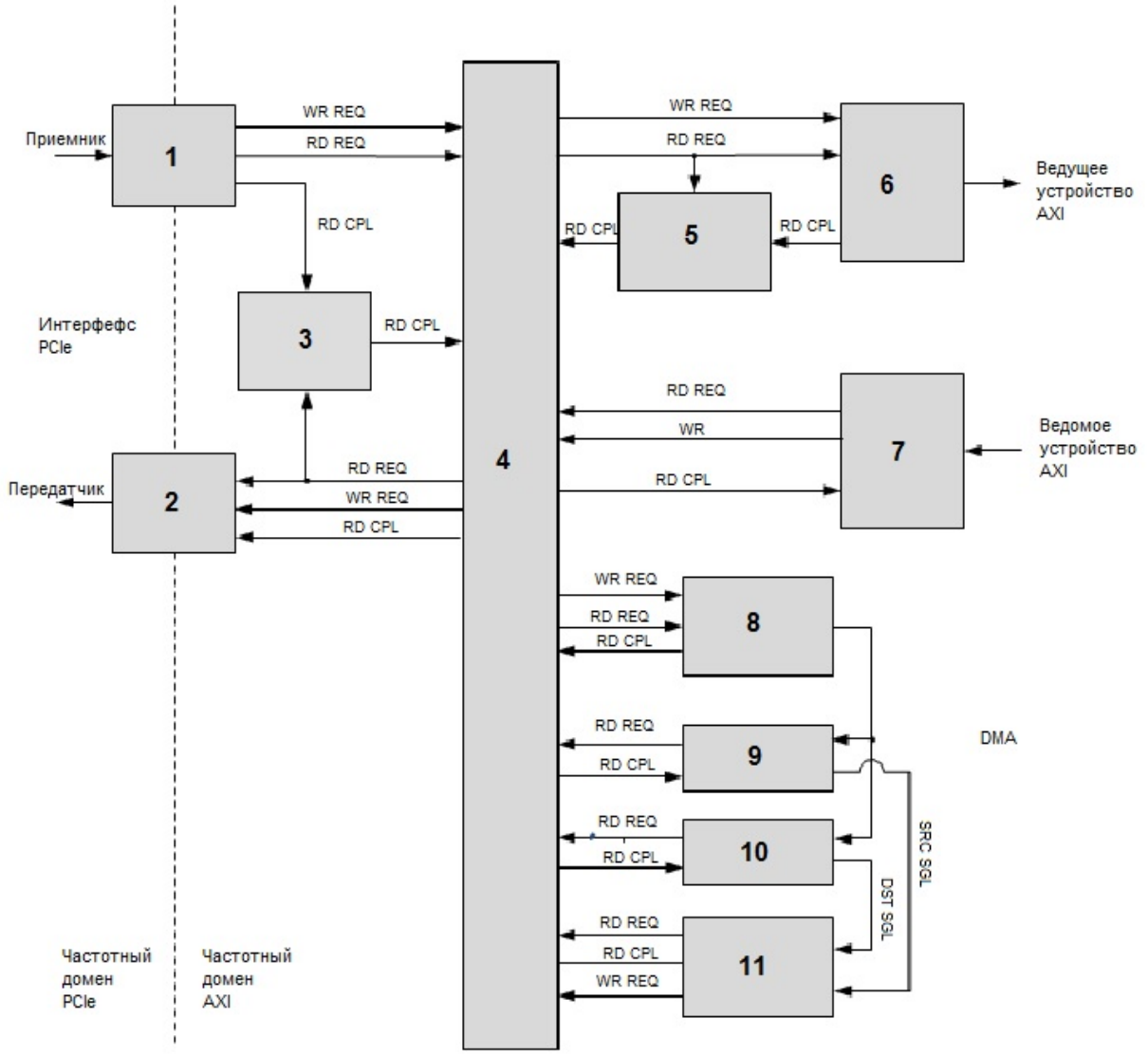


Рисунок 1.141 - Блок-схема Expresso DMA Bridge

1. Приемник PCIe
2. Передатчик PCIe
3. Регистратор очереди чтения PCIe
4. Внутреннее соединение
5. Регистратор очереди чтения AXI
6. Ведущее устройство AXI
7. Ведомое устройство AXI
8. Регистры и таблица MSI-X многоканального DMA
9. Кэш DMA SRC SGL
10. Кэш DMA DST SGL
11. Выполнение транзакции DMA

1.3.10.2.1.8.1 Характеристики Expresso DMA Bridge

Основные характеристики Expresso DMA Bridge:

- Интерфейс PCI Express

    - Высокопроизводительное подключение к многочисленным ядрам PCI Express 3.0, 2.1 и 1.1
  - Семейство Expresso ядер PCI Express компании Northwest Logic
  - Минимальные ядра PCI Express Xilinx и Altera
  - Поддержка ядер с разрядностью данных PCIe 32, 64, 128 и 256 бит
    - Поддержка возможностей PCIe Multi-Function и SR-IOV
  - Совместно с ядром Northwest Logic Expresso 3.0
    - Поддержка 64-битовой адресации
    - Тактовый сигнал PCIe может быьт независимым или синхронизироваться с интерфейсами AXI
- Интерфейс ведущего устройства AXI
    - Высокопроизводительное ведущее устройство AXI3/4 для выполнения транзакций моста PCIe -> AXI и DMA
  - Разработано для достижения полной пропускной способности
    - Разрядность данных интерфейса ведущего устройства AXI конфигурируется в зависимости от времени доставки данных
  - Поддержка 32, 64, 128 и 256-битовой разрядности AXI
  - Разрядность данных AXI должна быть >= разрядности данных PCIe
    - Поддержка 64-битовой адресации
    - Включает поддержку до 8 трансляций адреса PCIe -> AXI
    - Для всех источников транзакций (мост и ядро DMA) используется один общий интерфейс ведущего устройства AXI
- Интерфейс ведомого устройства AXI
    - Высокопроизводительное ведомое устройство AXI3 для выполнения транзакций моста PCIe->AXI (от AXI CPU или пользовательского контроллера DMA на основе AXI)
  - Предназначено для достижения полной пропускной способности
    - Разрядность данных интерфейса ведомого устройства AXI такая же, как у интерфейса ведущего устройства AXI
    - Поддержка 64-битовой адресации
    - Включает поддержку до 8 трансляций адреса AXI->PCIe
    - Поддержка механизма доступа к расширенной конфигурации PCIe (ECAM)
  - Используется приложениям корневого порта PCIe для конфигурирования иерархии PCI Express
    - Поддержка AXI CPU для генерации транзакций PCIe - конфигурационных, сообщений и ввода/вывода
    - Для всех источников транзакций (мост и пользователь DMA) используется один общий интерфейс ведомого устройства AXI
- Контроллер прерываний приема
    - Используется приложениями корневого порта PCIe для приема и агрегирования PCIe MSI-X, MSI, и унаследованных прерываний от иерархии PCI Express и предоставления их локальному AXI CPU.
- FIFO сообщений приемника PCI Express
- Высокопроизводительный многоканальный DMA
    - Высокопроизводительный DMA, предназначенный для достижения полной пропускной способности AXI и времени доставки PCIe, конфигурируется от 4 до 1024 каналов DMA
    - Время доставки конфигурируется для числа каналов DMA от 4 до 1024
  - Каналы DMA независимы и управляются программно

- Операции DMA канала DMA накладываются на границы 512 или1024 байт на циклической основе
- Реализация DMA выполнена на основе RAM с целью эффективной поддержки большого количества каналов при маленьком размере исполнения
- Типичные приложения обычно используют <= 32 каналов DMA >32 каналов DMA предназначены для Multi-Function/SR-IOV применений, так что каждая функция может иметь несколько назначенных каналов DMA
  - Каждый канал DMA поддерживает 4 направления пересылки данных
- PCIe->PCIe, PCIe->AXI, AXI->PCIe и AXI->AXI
  - Поддержка мультипроцессорного DMA
- Работа DMA управляется PCIe CPU, AXI CPU или обоими
- Списки фрагментации DMA могут быть расположены в памяти PCIe или AXI
- Состояния очереди завершения DMA могут быть расположены в памяти PCIe или AXI
- Двойные очереди статуса DMA позволяются каждому CPU в приложениях DMA с двумя CPU управлять собственной очередью статуса
  - Отдельные очереди фрагментации для источника и получателя
- Поддерживает независимую фрагментацию буферов DMA источника и получателя
- Ядро Expresso DMA Bridge объединяет SGL источника и получателя до создания соответствующих транзакций DMA, освобождая программу от этой задачи
  - Отдельное выполнение DMA и очередей статуса
- Выполнение DMA и статус могут эффективно обрабатываться разными процессами
  - Поддерживает применения в качестве корневого порта и оконечной точки (включая Multi-Function и SR-IOV)

- Опция ECC RAM Protection для высоконадежных применений (стандарт для ASIC)
- Содержит полный пример разработки
  - Иллюстрирует работу всех интерфейсов
  - Разрешает измерение производительности аппаратного DMA
  - Доступен исходный код
  - Доступны услуги индивидуальной настройки и интеграции

1.3.10.2.1.8.2 Генерация транзакций чтения и записи конфигурации PCI Express

Корневой порт PCI Express должен быть способен генерировать транзакции чтения и записи конфигурации, чтобы конфигурировать иерархию PCI Express. Стандарт AXI не обеспечивает средств поддержки для этой цели, и поэтому ядро Expresso DMA Bridge реализует механизм доступа к расширенной конфигурации (ECAM), чтобы транслировать запросы чтения и записи AXI в транзакции чтения и записи конфигурации PCI Express.

Оконечной точке PCI Express не разрешено генерировать транзакции чтения и записи конфигурации PCI Express, но ECAM в этом случае можно использовать для доступа к регистрам конфигурации ядра локальной PCI Express. При работе в качестве оконечной точки начальная установка и управление регистрами конфигурации ядра локальной PCI Express выполняются программой ведущего устройства и обычно не изменяются с помощью этого механизма.

Трансляция ECAM определяется следующими элементами управления:

- ecam_enable
  - 1 – Разрешено
  - 0 - Запрещено
- ecam_base[63:12]
  - Базовый адрес (байтовый адрес) для области ECAM {ecam_base[63:12], 12’h0}

- ecam_base[11:0], который не доступен для записи, соответствует минимальной области ECAM размером 4 KB.

- ecam_size[4:0]
  - 0 – 4 KB (минимум – только регистры конфигурации локального устройства PCIe)
  - 1 – 8 KB
  - – 16 KB
  - – 32 KB
  - – 64 KB
  - – 128 KB
  - – 256 KB
  - – 512 KB
  - – 1 MB (1 шина PCIe)
  - – 2 MB (2 шины PCIe)
  - – 4 MB (4 шины PCIe)
  - – 8 MB (8 шин PCIe)
  - – 16 MB (16 шин PCIe)
  - – 32 MB (32 шины PCIe)
  - – 64 MB (64 шины PCIe)
  - – 128 MB (128 шин PCIe)
  - – 256 MB (максимум 256 шин PCIe)
  - Минимальная область ECAM размером 4 KB - это минимальное адресное пространство, необходимое для поддержки только регистров конфигурации локального устройства PCI Express.
  - Максимальная область ECAM размером 256 MB - это максимальное адресное пространство, необходимое для поддержки наибольшей из возможных иерархий PCI Express (256 шин).

Трансляция ECAM имеет следующий статус:

- wr/rd_pending_ctr[8:0] (по одному счетчику для чтения и записи)
  - Количество транзакций, ожидающих трансляции (0-511). Количество незавершенных трансляций в каждый момент времени не может быть больше, чем количество незавершенных транзакций, поддерживаемых ядром Expresso DMA (всегда <512 транзакций), и поэтому ядро Expresso DMA не нуждается в отправке управляющих запросов из-за этого поля.

"Попадание" в регион ECAM имеет место, когда выполняются оба следующих условия:

- ecam_enable == 1
- ecam_base[63:12+ecam_size] == AXI Address[63:12+ecam_size]

Механизм ECAM PCI Express выделяет часть адресного пространства памяти AXI CPU под транзакции конфигурации PCI Express. Транзакции чтения/записи ведомого интерфейса AXI, целью которых являются эта транзакция, преобразуются в транзакцию чтения/записи конфигурации PCI Express при помощи следующей трансляции адресов:

- Адрес AXI [27:20]
  - Номер шины [7:0]
- Адрес AXI [19:12]
  - Для устройств PCI Express, реализующих ID альтернативной маршрутизации (ARI)
    - Адрес AXI [19:12] – Номер функции[7:0]
  - Для устройств PCI Express, не реализующих ID альтернативной маршрутизации (ARI)
    - Адрес AXI [19:15] – Номер функции[4:0]

- Адрес AXI [14:12] – Номер функции[2:0]
- Адрес AXI [11:2] – DWORD адрес регистра конфигурации [11:2]
- Адрес AXI [1:0] вместе с размером транзакции AXI используется для вычисления количества байт

Если трансляция ECAM сконфигурирована меньшей, чем 2^28 байт, то при выполнении транзакции конфигурации PCI Express старшие биты адреса, не входящие в окно трансляции, замещаются 0. Например, если для конкретного применения корневого порта требуются только 16 номеров шины PCI Express, то для доступа к необходимым 16 номерам шины PCI Express (номера шин 0-15) достаточно размера окна трансляции только 2^24.

Метод трансляции доступа ECAM к транзакциям конфигурации типа 0 или типа 1 различается в зависимости от используемого ядра PCI Express, поскольку в разных ядрах PCI Express по-разному реализована поддержка этой функции:

- Ядро Northwest Logic Expresso 3.0
  - Пакет TLP конфигурации типа 0 передается, если номер шины [7:0] совпадает с номером шины ядра Expresso 3.0 Core, в противном случае передается пакет TLP конфигурации типа 1.
  - Внутренние регистры конфигурации таргетируются посредством передачи пакета TLP конфигурации типа 0.
  - Ядро Expresso 3.0 направляет пакеты TLP конфигурации типа 1 на PCI Express, преобразовывая их в пакеты TLP конфигурации типа 0, когда номер шины совпадает с номером вторичной шины в регистре конфигурации, запрограммированном в ядре Expresso 3.0.
- Минимальное ядро Xilinx Gen 2 (GTX)
  - При работе в качестве корневого порта
    - Если номер шины[7:0] == номеру шины ядра Xilinx
    - Если номер устройства [4:0] == 0, и номер функции[2:0] == 0
    - Доступ к внутреннему регистру конфигурации передается на интерфейс конфигурации PCIe минимального ядра
    - В противном случае
    - Транзакция завершается со статусом DECERR
    - В противном случае, если номер шины [7:0] == номеру вторичной шины в регистре конфигурации, запрограммированном в ядре
    - Если номер устройства [4:0] == 0
    - Передается пакет TLP конфигурации типа 0.
    - В противном случае
    - Транзакция завершается со статусом DECERR.
    - В противном случае
    - Передается пакет TLP конфигурации типа 1.
  - При работе в качестве оконечной точки
    - Если номер шины[7:0] == 0, номер устройства[4:0] == 0, и номер функции[2:0] == 0
    - Доступ к внутреннему регистру конфигурации передается на интерфейс конфигурации PCIe минимального ядра
    - В противном случае
    - Транзакция завершается со статусом DECERR.
    - При работе в качестве оконечной точки программе трудно определить номер шины, который был назначен оконечной точке, и поэтому для доступа к конфигурационному пространству локального PCIe используется {Bus Number, Device Number, Function Number} == 0.
    - Оконечным точкам не разрешено передавать пакеты TLP запросов конфигурации, и поэтому пакеты TLP конфигурации типа 1/0 не передаются.

Следует учитывать, что транзакции, возвращающие статус DEC_ERR, преобразуются в OKAY с данными, состоящими из всех 1, когда регистра моста fg_rd_ur_is_ur_ok1s_n == 0. Некоторые CPU не воспринимают полученные отклики со статусом ошибки DEC_ERR, и им необходим статус OKAY со всеми 1. Более подробную информацию см. в разделе 0.

Транзакциям ECAM не разрешено пересекать границу адреса DWORD. Если транзакция, "попадающая" в регион ECAM, пересекает границу адреса DWORD, такая транзакция прерывается со статусом SLVERR.

Запись или чтение ECAM подчиняются тому же протоколу и правилам упорядочивания, что и другие транзакции AXI.

Транзакции записи ECAM записываются в FIFO транзакций записи с обязательным ожиданием ответного пакета, чтобы избежать взаимоблокировки. При тайм-ауте транзакции ECAM возвращается SLVERR.

1.3.10.2.1.8.2.1 Ответ AXI на транзакции чтения и записи конфигурации PCI Express

Когда иерархия PCIe пронумерована, программа должна выдать запросы чтения конфигурации туда, где функции могут не существовать, чтобы проверить их наличие. Чтение конфигурации не существующих функций возвращают завершение со статусом неподдеживаемого запроса (UR). Кроме того, устройства PCI Express могут завершать транзакции конфигурации со статусом повтора, указывая, что они временно не могут завершить транзакцию. Ядро Expresso DMA Bridge Core транслирует отклики PCI Express в ограниченные отклики AXI , используя регистры, чтобы определить опциональное поведение.

Ядро Expresso DMA Bridge Core транслирует отклики завершения транзакции чтения и записи конфигурации PCI Express Configuration в отклики AXI следующим образом:

- PCIe отклик Success

AXI отклик OKAY

- PCIe отклик Unsupported Request
  - При чтении конфигурации
    - cfg_rd_ur_is_ur_ok1s_n == 1
    - AXI отклик DECERR
    - cfg_rd_ur_is_ur_ok1s_n == 0
    - AXI отклик OKAY с данными чтения AXI все 1
  - При записи конфигурации
    - AXI отклик DECERR
- PCIe отклик Completer Abort
  - AXI отклик SLVERR
- PCIe отклик Configuration Request Retry Status (CRS)
  - Если разрешена CRS Software Visibility корневого порта PCI Express (p_cfg_crs_sw_vis_en == 1), AXI Address[11:0] == 0, AXI Size == 16-бит или 32-бит, и выполняется транзакция чтения, то
    - AXI выдает ответ OKAY с данными чтения AXI 0xFFFF0001.
    - Программное обеспечение, поддерживающее CRS Software Visibility, определит, что это специальное значение данных означает, что устройство PCI Express вернуло статус CRS. Затем инициатор AXI обычно переключается на другие задачи, повторяя транзакцию позже.

В противном случае ядро Expresso DMA Bridge будет продолжать попытки транзакции, пока не вернется статус, отличный от CRS. Однако транзакция будет прервана со статусом DECERR, если она не завершается успешно после попыток в течение > 1 секунды (самый большой период времени, в течение которого устройству PCIe разрешено возвращать статус CRS).

1.3.10.2.1.8.3 FIFO транзакций записи с обязательным ожиданием ответного пакета

Транзакции ECAM интерфейса ведомого устройства AXI и транзакции записи ввода/вывода являются транзакциями с обязательным ожиданием ответного пакета на PCI Express - они требуют арбитража для тега PCI Express и ресурсов для обработки завершения, управляемых очередью переупорядочивания PCI Express ядра Expresso DMA Bridge, и имеют правила упорядочивания, отличные от транзакций, не требующих ответного пакета (таких, как не-ECAM и запись ввода/вывода). Для PCI Express транзакциям с обязательным ожиданием ответного пакета не должно разрешаться останавливать задержанные транзакции, чтобы избежать взаимоблокировки. Поскольку транзакция записи с обязательным ожиданием ответного пакета, которая не может быть поставлена в очередь переупорядочивания PCI Express из-за недостатка области памяти, блокирует последующие транзакции записи AXI с обязательным ожиданием ответного пакета, то транзакции записи помещаются в очередь в дополнительное FIFO AXI для транзакций записи с обязательным ожиданием ответного пакета. FIFO транзакций записи с обязательным ожиданием ответного пакета должно иметь такой размер, чтобы вмещать все возможные транзакции записи с обязательным ожиданием ответного пакета, ожидающие своей очереди, поскольку если их количество превысит количество, поддерживаемое FIFO транзакций записи с обязательным ожиданием ответного пакета, то транзакции записи с обязательным ожиданием ответного пакета могут заблокировать транзакции записи, не требующие ответных пакетов.

Глубина FIFO транзакций записи с обязательным ожиданием ответного пакета должна выбираться для данного CPU, чтобы она была достаточной для хранения всех возможных ожидающих транзакций записи с обязательным ожиданием ответного пакета. По умолчанию глубина FIFO транзакций записи с обязательным ожиданием ответного пакета составляет 16 транзакций, но при необходимости это значение может быть увеличено индивидуальной настройкой. В каждый момент времени ожидающей может быть только одна транзакция ввода/вывода, и поэтому оставшийся объем FIFO транзакций записи с обязательным ожиданием ответного пакета доступен для буферизации транзакций записи конфигурации.

1.3.10.2.1.8.4 Запись во внутренние регистры

Доступ к внутренним регистрам моста осуществляется через интерфейс ведомого устройства AXI при помощи трансляции регистров моста. Доступ к внутренним регистрам DMA осуществляется через интерфейс ведомого устройства AXI при помощи трансляции регистров DMA.

Доступ к внутренней таблице MSI-X осуществляется через интерфейс ведомого устройства AXI при помощи трансляции таблицы MSI-X.

Доступ к внутреннему массиву бит статуса запросов прерываний (PBA) типа MSI-X осуществляется через интерфейс ведомого устройства AXI при помощи трансляции PBA MSI-X.

Трансляция регистров моста при сбросе конфигурируется таким образом, чтобы принимать все транзакции AXI как доступ к регистрам моста. После начальной установки AXI CPU должен первым делом реконфигурировать трансляцию регистров моста в конкретное (маленькое) окно, чтобы разрешить остальные трансляции адреса.

Трансляции регистра моста, регистра DMA, таблицы MSI-X и MSI-X PBA определяется следующими элементами управления:

- enable

1 – Разрешить

0 – Запретить

- size[2:0]
  - Размер области трансляции в байтах == 2^(12+size). Минимальный размер (2^(12+0)) == 4KB, а максимальный размер (2^(12+7)) == 512 KB.

- Например
  - 0 – 4 KB
  - 4 – 64 KB
  - 7 – 512 KB
- Если установленный размер меньше размера реализованной области регистров, то регистры, реализованные за границей этой области, недоступны.

- base[63:12]

"Попадание" в адресный регион регистров трансляции имеет место, когда выполняются оба условия:

- enable == 1
- base[63:(12+size)] == AXI Address[63:(12+size)]

При "попадании"

- Транзакция выполняется как соответствующая транзакция регистра DMA, таблицы MSI-X или MSI-X PBA.
- AXI адрес[(12+size)-1:0] используется для адресации конкретного регистра, к которому обращаются. Если размер области меньше, чем реализованное регистровое пространство, то старшие биты адреса регистра устанавливаются в 0.

Запись и чтение во внутренние ресурсы подчиняются тому же протоколу и правилам упорядочивания, что и другие транзакции AXI.

1.3.10.2.1.9 Трансляция адресов

Ядро Expresso DMA Bridge содержит поддержку трансляции адресов как для входных (к AXI) и выходных (от AXI) запросов.

- В режиме ведущего устройства AXI выполняются до 8 входных трансляций
  - Трансляция выполняется только для транзакций, исходящих от PCIe, причем тех, которые не были декодированы как прерывания MSI или MSI-X. Внутренние транзакции DMA не транслируются.
- В режиме ведомого устройства AXI выполняются до 8 выходных трансляций
  - Трансляция выполняется только для транзакций, исходящих от AXI и предназначенных для PCIe, причем тех, которые не относятся к ECAM (чтение/запись конфигурации)

Каждая трансляция адреса имеет следующие элементы управления:

- (ingress/egress)_enable - “tran_enable”
  - 1 – Разрешить
  - 0 – Запретить
- (ingress/egress)_security_enable – “tran_security_enable”
  - 1 – Разрешить
  - 0 - Запретить
- (ingress/egress)_attr_enable – “tran_attr_enable”
  - 1 – Разрешить
  - 0 – Запретить
- (ingress/egress)_invalid - “tran_invalid”
  - 1 – Недействительно
  - 0 – Действительно
- (ingress/egress)_size[4:0] – “tran_size”
  - Размер области трансляции в байтах == 2^(12+tran_size). Минимальный размер tran_size (2^(12+0))== 4KB, а максимальный размер (2^(12+31)) == 8 TB.

- Например
  - 0 – 4 KB
  - 4 – 64 KB
  - 8 – 1 MB
  - 12 – 16 MB
  - 18 – 1 GB
  - 31 – 8 TB
- (ingress/egress)_attr_r[3:0] – “tran_attr_r”
  - входная
    - Когда tran_attr_enable == 1, AXI порт транзакции m_arache устанавливается в tran_attr_r[3:0] при направлении в AXI транзакций чтения, попадающих в адресный регион этой трансляции. Когда tran_attr_enable == 0, атрибуты кэша по умолчанию определяются cfg_pcie_rx_arache
  - выходная
    - Когда tran_attr_enable == 1, атрибуты PCIe {ID Based Ordering, Relaxed Ordering, No Snoop} устанавливаются в tran_attr_r[2:0] при направлении в PCIe транзакций чтения, попадающих в адресный регион этой трансляции. Когда tran_attr_enable == 0, атрибуты PCIe все устанавливаются в 0 при направлении в PCIe транзакций чтения, попадающих в адресный регион этой трансляции. Атрибуты, которые не разрешено устанавливать из-за разрешенной функциональности регистра конфигурации PCI Express, будут очищены перед тем, как транзакции направляются в PCIe.
- (ingress/egress)_attr_w[3:0] – “tran_attr_w”
  - входная
    - Когда tran_attr_enable == 1, AXI порт транзакции m_awcache устанавливается в tran_attr_w[3:0] при направлении в AXI транзакций записи, попадающих в адресный регион этой трансляции. Когда tran_attr_enable == 0, атрибуты кэша по умолчанию определяются примененным cfg_pcie_rx_aqcache.
  - выходная
    - Когда tran_attr_enable == 1, атрибуты PCIe {ID Based Ordering, Relaxed Ordering, No Snoop} устанавливаются в tran_attr_w[2:0] при направлении транзакций записи, попадающих в адресный регион этой трансляции. Когда tran_attr_enable == 0, все атрибуты PCIe устанавливаются в 0 при направлении в PCIe транзакций записи, попадающих в адресный регион этой трансляции. Атрибуты, которые не разрешено устанавливать из-за разрешенной функциональности регистра конфигурации PCI Express, будут очищены перед тем, как транзакции направляются в PCIe.
- (ingress/egress)_src_base[63:12] – “tran_src_base”
- (ingress/egress)_dst_base[63:12] – “tran_dst_base”

Каждая трансляция адреса имеет следующий статус:

(ingress/egress)_pending_ctr[8:0]

- Количество транзакций, ожидающих трансляции (0-511). Количество ожидающих трансляций не может быть больше, чем количество одновременно ожидающих транзакций, поддерживаемых ядром Expresso DMA Bridge (всегда <512 транзакций), и поэтому ядро Expresso DMA Bridge не нуждается в отправке управляющих запросов из-за этого поля.

"Попадание" в адресный регион трансляции имеет место при выполнении двух условий:

- tran_enable == 1
- tran_src_base[63:(12+tran_size)] == Source Address[63:(12+tran_size)]

При попадании в несколько адресных регионов трансляции для транзакции используется трансляция с наименьшим индексом (наименьший сдвиг адреса регистра трансляции для направления вход/выход).

При "попадании"

- Старшие биты адреса источника заменяются соответствующими битами в tran_dst_base при направлении транзакции получателю:
  - Destination Address[63:(12+tran_size)] = tran_dst_base[63:(12+tran_size)]
  - Destination Address[(12+tran_size)-1:0] = Source Address[(12+tran_size)-1:0]
- Если tran_invalid == 1
  - Транзакция не направляется получателю и обрабатывается как ошибка:
    - Если источник транзакции AXI - возвращается ответ DECERR
    - Если источник транзакции PCIe - она обрабатывается как неподдерживаемый запрос
- Если tran_invalid == 0 и security_enable == 1
  - Если источник транзакции PCIe, а получатель транзакции AXI
    - При направлении транзакции в AXI порту m_arprot/m_awprot присваивается значение порта i_security_ingress[i], ассоциированного с трансляцией
  - Если источник транзакции AXI, а получатель транзакции PCIe
    - Если уровень защиты s_arprot/s_awprot транзакции AXI совпадает с уровнем защиты порта i_security_egress[i], ассоциированного с трансляцией, то транзакция направляется PCIe; в противном случае транзакция сбрасывается и возвращается ответ SLVERR.

Если транзакция не попадает во все адресные регионы трансляции (входа/выхода), то она обрабатывается посредством процесса “субтрактивного” декодирования.

Процесс субтрактивного декодирования управляется следующими регистрами:

- ingress_sub_enable - Разрешение субтрактивного декодирования для входных транзакций от PCIe к AXI
  - 1 – Разрешить
    - Транзакция, которая не попадает во все адресные регионы трансляции, передается нетранслируемой.
    - Если транзакция выполняется из источника PCIe к получателю AXI, то уровень защиты m_arprot/m_awprot устанавливается как незащищенный.
  - 0 – Запретить

Транзакция, которая не попадает во все адресные регионы трансляции, не направляется получателю и обрабатывается как ошибка. Возвращается неподдерживаемый запрос PCIe.

- egress_sub_enable - Разрешение субтрактивного декодирования для выходных транзакций от AXI к PCIe
  - 1 – Разрешить
    - Транзакция, которая не попадает во все адресные регионы трансляции, передается нетранслируемой.
  - 0 – Запретить
    - Транзакция, которая не попадает во все адресные регионы трансляции, не направляется получателю и обрабатывается как ошибка: Возвращается ответ AXI DECERR.

Если трансляциям адресов присвоены разные уровни защиты, важно выполнить начальную установку процесса субтрактивного декодирования таким образом, чтобы он не обходил требуемую защиту. В некоторых случаях может быть желательным запрет на субтрактивное декодирование.

1.3.10.2.1.10 Контроллер полученных прерываний и сообщений корневого порта

Ядро Expresso DMA Bridge реализует контроллер полученных прерываний и сообщений, предназначенный для сбора прерываний и сообщений от иерархии PCIe. Прием прерываний возможен только в применениях как корневого порта. При реализации применений в качестве оконечной точки все маски MSI-X, MSI и унаследованных прерываний должны быть очищены (прерывания запрещены).

Контроллер полученных прерываний и сообщений корневого порта поддерживает следующие прерывания:

- Полученные PCI Express прерывания MSI-X
  - Конфигурируемый диапазон адресов
  - Поддержка многовекторности
- Полученные PCI Express прерывания MSI
  - Конфигурируемый диапазон адресов
  - Поддержка многовекторности
- Полученные PCI Express унаследованные прерывания
  - Ядро Expresso DMA Bridge обрабатывает унаследованные сообщения о прерываниях деактивации и регенерирует унаследованные прерывания INTA, INTB, INTC и INTD
- Внутренние прерывания
  - Прерывания от каналов DMA
  - Другие прерывания
- Прерывания PCIe
  - Прерывания вследствие обнаруженных ядром событий (изменение пропускной способности канала, ошибка и т.д.)

Контроллер полученных прерываний и сообщений предоставляет следующие выходные порты прерываний, которые должны подключаться к одному или нескольким входам прерывания AXI CPU.

- s_int_msi[0]
  - Ожидает одно или несколько прерываний MSI с вектором прерываний MSI в диапазоне 31...0
- s_int_msi[1]
  - Ожидает одно или несколько прерываний MSI с вектором прерываний MSI в диапазоне 63...32
- s_int_legacy
  - Активно унаследованное прерывание PCI Express - INTA, INTB, INTC или INTD
    - Полученные сообщения INT[A, B,C,D] Interrupt Assert и Interrupt Deassert используются для реконструкции унаследованных прерываний PCIe - INT[A,B,C,D].
- s_int_dma
  - Активно одно или несколько прерываний AXI канала DMA.
- s_int_misc
  - Активно одно или несколько других прерываний.

Ядро Expresso DMA Bridge активизирует полученные прерывания MSI-X и MSI только после того, как предыдущие транзакции записи AXI завершились ответным пакетом.

Ядро Expresso DMA Bridge ждет того, чтобы предыдущие транзакции записи AXI завершились ответным пакетом, прежде чем сообщениям об унаследованных прерываниях Assert_INT[A,B,C,D] будет позволено активизировать регистры msgf_leg_int_status.

Ядро Ядро Expresso DMA Bridge реализует FIFO принятых сообщений с глубиной 128, чтобы хранить принятые сообщения, прерывания MSI-X и (опционально) прерывания MSI. Унаследованные прерывания и (опционально) прерывания MSI записываются в регистры статуса прерывания, а не в FIFO принятых сообщений.

Контроллер принятых прерываний и сообщений реализует следующие регистры для прерываний MSI-X:

- msix_enable
  - 1 - Разрешить - Разрешить запись прерываний MSI-X в FIFO принятых сообщений.
  - 0 – Запретить
- msix_size[4:0]
  - Размер области трансляции MSI-X в байтах == 2^(12+msix_size). Минимальный размер (2^(12+0)) == 4KB, а максимальный размер (2^(12+31)) == 8 TB
- msix_base[63:12]
  - Входная транзакция записи считается прерыванием MSI-X, когда одновременно msix_enabled == 1 и Ingress Address[63:12+ msix_size] = msix_base[63:12+msix_size]

Контроллер принятых прерываний и сообщений реализует следующие регистры для прерываний MSI:

- msii_enable
  - 1 - Разрешить декодирование прерывания MSI
  - 0 – Запретить
- msii_status_enable
  - 1 - Прерывания MSI записываются в регистры msgf_msi_status.
  - 0 - Прерывания MSI записываются в FIFO принятых сообщений.
- msii_size[4:0]
  - msii_status_enable == 1
    - Размер области трансляции MSI в байтах == 4 байта (поддерживается только 1 DWORD адрес с 64 разными векторами данных), а msii_size игнорируется.
  - msii_status_enable == 0
    - Размер области трансляции MSI в байтах == 2^(12+msii_size). Минимальный размер (2^(12+0))== 4KB, а максимальный размер (2^(12+31)) == 8 TB.
- msii_src_base[63:12]
  - Входная транзакция записи считается прерыванием MSI, когда одновременно msii_enable == 1 и Ingress Address[63:12+ msii_size] = msii_src_base[63:12+msii_size]. msii_status_enable определяет, записываются ли прерывания MSI в регистр msgf_msi_status или в FIFO принятых сообщений.
- msgf_msi_status[63:0]
  - Когда msii_status_enable == 1 и прерывание попадает в адресный регион MSI, содержание данных [5:0] прерывания MSI используется для определения вектора прерываний MSI. Устанавливается бит msgf_msi_status[i], соответствующий вектору i, чтобы зарегистрировать прерывание. msgf_msi_status[i] очищается, когда программа записывает 1 (после обслуживания прерывания). События прерывания, которые происходят, когда msgf_msi_status[i] активизирован, объединяются.
  - Когда msii_status_enable == 0, указывая, что прерывания MSI записаны в FIFO принятых сообщений, а не в msgf_msi_status, msgf_msi_status[63:0] возвращает 0.
- msgf_msi_mask[63:0]

- Когда msgf_msi_status[i] == 1 и msgf_msi_mask[i] == 1, прерывание подается на AXI CPU. Первые 32 вектора прерывания MSI логически складываются для генерации s_int_msi[0], а следующие 32 вектора прерывания MSI логически складываются для генерации s_int_msi[1]:
  - s_int_msi[i] = |(msgf_msi_status[(i*32)+31:(i*32)] & msgf_msi_mask[(i*32)+31:(i*32)])

Контроллер принятых прерываний и сообщений реализует следующие регистры для унаследованных прерываний PCIe:

- msgf_leg_status[3:0]
  - Принятые PCIe сообщения об унаследованных INT[D/C/B/A]Assert и Deassert обрабатываются так, чтобы регенерировать аппаратные линии унаследованных прерываний INTD/C/B/A.
  - msgf_leg_status[3] – INTD
  - msgf_leg_status[2] – INTC
  - msgf_leg_status[1] – INTB
  - msgf_leg_status[0] - INTA
- msgf_leg_mask[3:0]
  - Когда msgf_leg_status[i] == 1 and msgf_leg_mask[i] == 1, прерывание подается на AXI CPU. Все 4 унаследованных прерывания логически складываются для генерации s_int_legacy:
  - s_int_legacy = |(msgf_leg_status[3:0] & msgf_leg_mask[3:0])

Контроллер принятых прерываний и сообщений реализует следующие регистры для других прерываний:

- msgf_misc_status[31:0]
  - Устанавливает соответствующий msgf_misc_status [i] == 1 при появлении прерывания. msgf_misc_status[i] очищается, когда программа записывает 1 (после обслуживания прерывания). События прерывания, которые происходят, когда msgf_misc_status[i] активизирован, соединяются в один выход msgf_misc_status[i].
  - msgf_misc_status[31:0] распределен следующим образом:
  - [0] - Доступно FIFO принятых сообщений. Устанавливается в 1, когда FIFO принятых сообщений не пуст. Одно принятое сообщение извлекается из FIFO принятых сообщений каждый раз, когда в msgf_int_status[0] записывается 1. Если FIFO пуст, то при попытке извлечь из него сообщение ошибка не генерируется. Содержимое сообщение доступно на портах msgf_current_* и должно быть прочитано, прежде чем сообщение извлекается из стека.
  - [1] - Переполнение FIFO принятых сообщений. Устанавливается в 1 при любой попытке записи сообщения в FIFO принятых сообщений, однако при переполненном FIFO принятых сообщений сообщение будет сброшено. Поскольку объем FIFO принятых сообщений ограничен, программа должна обрабатывать принятые сообщения с высоким приоритетом, чтобы не допускать переполнения FIFO принятых сообщений и потери сообщений.
  - [3:2] – Зарезервировано.
  - [4] – Ошибка ведомого устройства AXI - Устанавливается в 1, когда запрос на интерфейсе ведомого устройства AXI завершается со статусом SLVERR или DECERR.
  - [5] – Ошибка ведущего устройства AXI - Устанавливается в 1, когда запрос на интерфейсе ведущего устройства AXI завершается со статусом SLVERR или DECERR.
  - [6] - Нарушение защиты трансляции входного адреса - Устанавливается в 1, когда происходит нарушение защиты при любой входной трансляции.

- [7] - Нарушение защиты трансляции выходного адреса - Устанавливается в 1, когда происходит нарушение защиты при любой выходной трансляции.
- [15:8] – Зарезервировано.
- [31:16] - Прерывание, сгенерированное ядром PCI Express [15:0]. Устанавливается, когда соответствующий входной порт прерывания i_pcie_interrupt[15:0] активирован в 1 в течение одного такта.

- msgf_misc_mask[31:0]
  - Когда msgf_misc_status[i] == 1 и msgf_misc_mask[i] == 1, прерывание подается на AXI CPU. Все другие прерывания логически складываются для генерации s_int_misc:
  - s_int_misc = |(msgf_misc_status[31:0] & msgf_misc_mask[31:0])
- msgf_axi_slave_id[7:0] – AXI ID первой обнаруженной ошибки SLVERR или DECERR интерфейса ведомого устройства AXI. Очищается, когда в msgf_misc_status[4] записывается 1, чтобы разрешить последующей ошибке записать новый ID ошибки AXI.
- msgf_axi_master_id[7:0] – AXI ID первой обнаруженной ошибки интерфейса ведущего устройства AXI. Очищается, когда в msgf_misc_status[5] записывается 1, чтобы разрешить последующей ошибке записать новый ID ошибки AXI.
- msgf_ingress_id[7:0] – AXI ID первой обнаруженной ошибки защиты трансляции входного адреса. Очищается, когда в msgf_misc_status[6] записывается 1, чтобы разрешить последующей ошибке записать новый ID ошибки AXI.
- msgf_egress_id[7:0] – AXI ID первой обнаруженной ошибки защиты трансляции выходного адреса. Очищается, когда в msgf_misc_status[7] записывается 1, чтобы разрешить последующей ошибке записать новый ID ошибки AXI.

Контроллер принятых прерываний и сообщений реализует следующие регистры для прерываний DMA:

- msgf_dma_status
  - Логическое сложение всех каналов прерывания DMA активно. msgf_dma_status очищается при очистке всех учитываемых регистров статуса прерывания канала DMA AXI в каждом наборе регистров канала DMA.
- msgf_dma_mask
  - Когда msgf_dma_status == 1 и msgf_dma_mask == 1, прерывание подается на AXI CPU.
  - s_int_dma = (msgf_dma_status & msgf_dma_mask)

Контроллер принятых прерываний и сообщений реализует следующие регистры для работы FIFO принятых сообщений:

- msgf_fifo_level[8:0]
  - Текущий уровень FIFO принятых сообщений. Если msgf_fifo_level > 0, то в FIFO принятых сообщений есть как минимум одно доступное сообщение.
- msgf_current_type[7:0]
  - Тип текущего сообщения - Определяет декодирование остальных полей
  - 255:3 – Зарезервировано
  - 2 – MSI-X
  - 1 – MSI
  - 0 – Другие
- msgf_current_code[7:0]
  - Тип текущего сообщения == MSI-X
  - Не используется
  - Тип текущего сообщения == MSI

        - Не используется
        - Тип текущего сообщения == Другой
        - PCIe TLP Message Tag[7:0]
    - msgf_current_tag[7:0]
        - Тип текущего сообщения == MSI-X
        - Не используется
        - Тип текущего сообщения == MSI
        - Не используется
        - Тип текущего сообщения == Другой
        - PCIe TLP Message Tag[7:0]
    - msgf_current_req_id[15:0]
        - PCIe TLP Requester ID[15:0]
        - {Bus[7:0], Function[7:0]} для функций, поддерживающих ID альтернативной маршрутизации (ARI)
        - {Bus[7:0], Device[4:0], Function[2:0]} для функций, не поддерживающих ARI
    - msgf_current_payload[31:0]
        - Тип текущего сообщения == MSI-X
        - Содержимое данных[31:0], полученных с прерыванием MSI-X
        - Тип текущего сообщения == MSI
        - Содержимое данных[31:0], полученных с прерыванием MSI (следует учитывать, что согласно спецификации MSI значение имеют только [15:0])
        - Тип текущего сообщения == Другой
        - Содержимое данных, полученных с сообщением (при наличии). Содержимое данных дополнительного сообщения (при наличии) не сохраняется.
    - msgf_current_address[63:0]
        - Тип текущего сообщения == MSI-X
        - Полный адрес[63:0], полученный с прерыванием MSI-X
        - Тип текущего сообщения == MSI
        - Полный адрес[63:0], полученный с прерыванием MSI
        - Тип текущего сообщения == Другой
        - Определяемые сообщением биты TLP {8, 9, 10, 11, 12, 13, 14, 15} (всего 64 бита). Следует учитывать, что определяемые сообщением байты размещаются в том же порядке, что получаются на шине PCI Express.

При реализации в качестве оконечной точки ядро Express DMA Bridge опционально может генерировать прерывания MSI/MSI-X, когда бит cfg_pcie_int_axi_pcie_n bit установлен в 0. Для этого применения весь набор логики msgf_misc errors/status логически складывается в один сигнал, который при активизации использует 0-ой элемент таблицы MSI-X для генерации прерывания MSI-X, если элемент 0 PVM равен 0 (не маскированный; если маскированный, то вместо этого устанавливается соответствующий бит PBA). Каждый канал DMA также может генерировать прерывание MSI-X, а в этом случае i-й элемент таблицы прерываний MSI-X используется для генерации прерывания MSI-X, когда i-й контроллер DMA запрашивает обслуживание прерывания. Кроме того, соответствующий бит msgf_dma/misc_mask (не маскированная функциональность, 1 разрешает/не маскирует прерывание) также должен быть установлен, чтобы имелась возможность генерировать прерывание MSI/MSI-X. Следует отметить, что существует перекрытие между прерыванием канала 0 DMA и прерыванием misc_status_combined. В случае 0 канала DMA и если некоторые или все ошибки misc_status разрешены и используются, то подпрограмма обработки прерывания ведущего устройства PCIe должна проверять оба потенциальных источника, чтобы полностью обработать прерывание.

Следует отметить, что новое сообщение MSI-X будет генерироваться каждый раз, когда появляется состояние статуса (обычно ошибки) (даже если подпрограмма обслуживания прерываний не очищает бит статуса после предыдущей активизации), если соответствующий вектору бит маски в таблице MSI-X очищен и при условии, что соответствующий бит msgf_dma/misc_mask также установлен, разрешая прерывание.

1.3.10.2.1.11 Обработка входной транзакции AXI

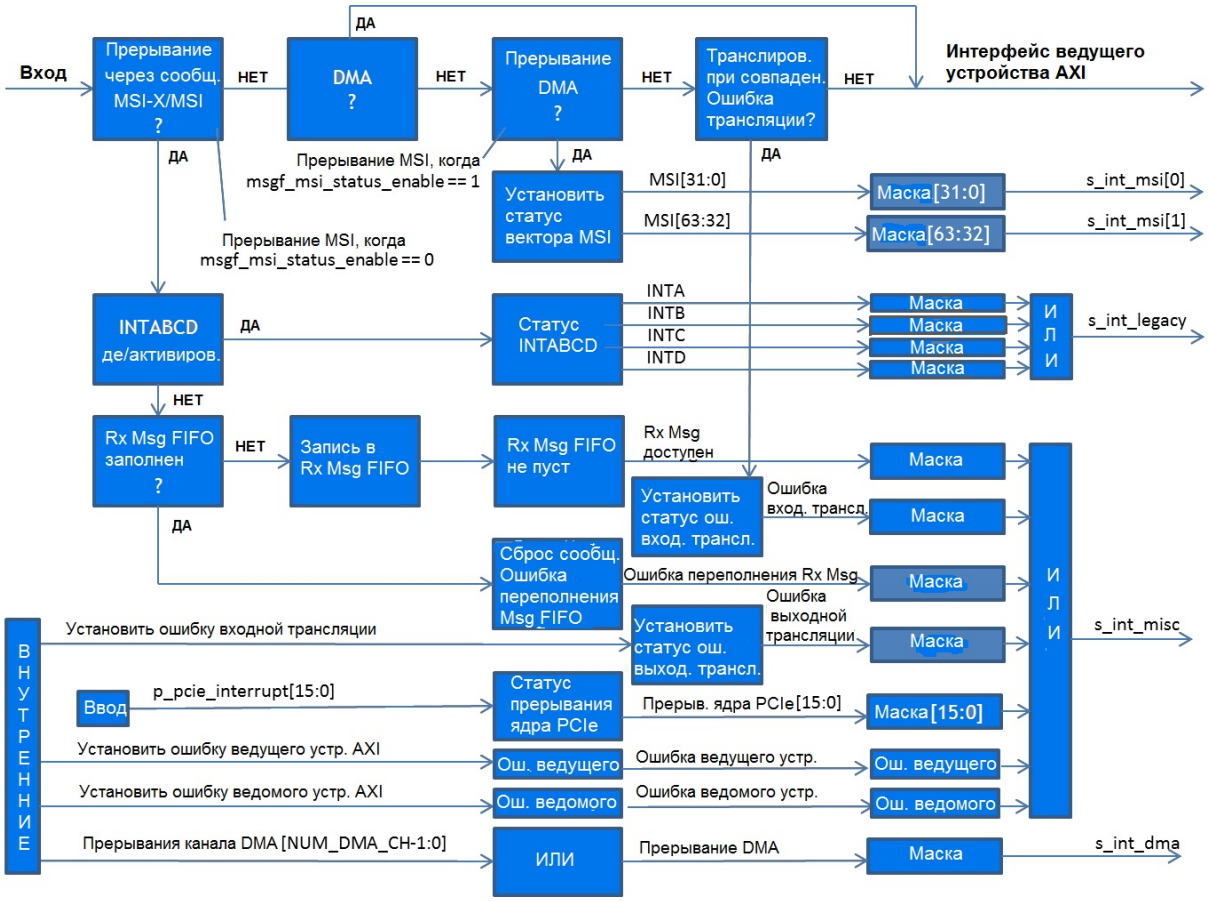


Рисунок 1.142. Блок-схема входной транзакции Expresso DMA Bridge

На рисунке 8 проиллюстрирована обработка входной транзакции AXI, включая генерацию прерывания для локального AXI CPU. Подробное описание обработки транзакции приведено в соответствующих разделах руководства пользователя.

1.3.10.2.1.12 Обработка выходной транзакции AXI

На рисунке 9 проиллюстрирована обработка выходной транзакции AXI. Подробное описание обработки транзакции приведено в соответствующих разделах руководства пользователя.

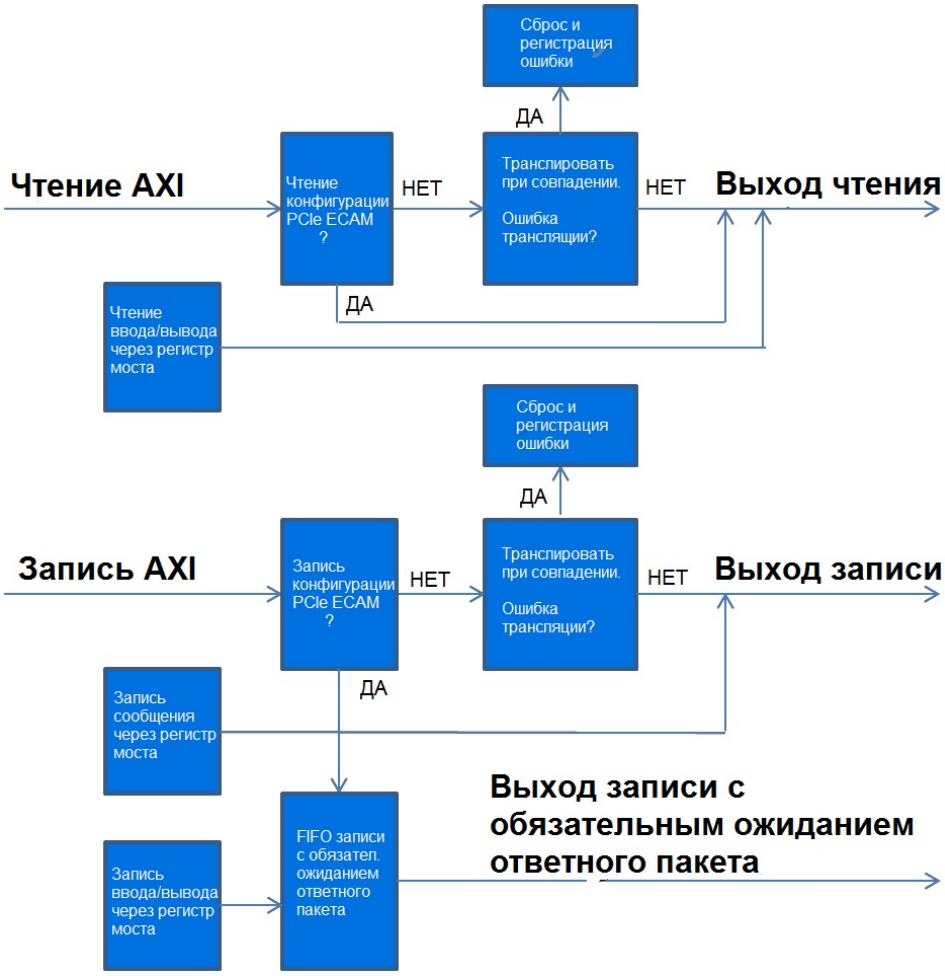


Рисунок 1.143 - Блок-схема выходной транзакции Expresso DMA Bridge

1.3.10.2.1.13 Регистры моста

Ядро Expresso DMA Bridge реализует множество внутренних регистров для функциональности моста и DMA.

Доступ к внутренним регистрам моста и DMA осуществляется при помощи разных механизмов, в зависимости от источника транзакции:

- AXI
  - Доступ к внутренним регистрам моста осуществляется при помощи трансляции регистров ведомого моста AXI.
  - Доступ к регистрам канала DMA осуществляется при помощи трансляции регистров ведомого DMA AXI.
- PCIe
  - Доступ к регистрам моста и DMA осуществляется от PCIe с фиксированными смещениями в регионе BAR ядра PCIe, который конфигурируется через cfg_dma_reg_bar.

- Доступ PCIe к регистрам моста и DMA можно по-отдельности запрещать в регистрах cfg_disable_pcie_bridge_reg_access и cfg_disable_pcie_dma_reg_access в том случае, если PCIe не должно иметь возможности записи в регистры моста и DMA (например, для обеспечения защиты).

1.3.10.2.1.14 Многоканальный DMA

1.3.10.2.1.14.1 Основные характеристики

- Высокопроизводительный DMA, предназначенный для достижения полной пропускной способности AXI и времени доставки PCIe, конфигурируется от 4 до 1024 каналов DMA
- Число каналов DMA конфигурируется от 4 до 1024
  - Каналы DMA независимы и управляются программно
  - Операции DMA канала DMA накладываются на границы 512 или 1024 байт
  - Реализация DMA на основе RAM поддерживает большое количество каналов малого размера
- Предназначено для поддержки таких применений, как оконечная точка, корневой порт и оконечная точка SR-IOV
  - Регистры DMA и механизмы статуса DMA совместимы с применениями SR-IOV
- Регистры DMA доступны и с PCIe, и AXI
  - DMA может запускаться/управляться с обоих интерфейсов
- Списки фрагментации DMA могут быть расположены в памяти PCIe и/или AXI
- Состояния очереди завершения DMA могут быть расположены в памяти PCIe и/или AXI
- Отдельные очереди фрагментации для источника и получателя
  - При выполнении операций DMA объединяется информация фрагментации источника и получателя, в результате чего программа освобождается от этой обязанности
- Отдельные очереди статуса завершения DMA для источника и получателя
  - Выполнение DMA и статус могут эффективно обрабатываться разными процессами
  - Для применений с двумя CPU каждое CPU может управлять собственной очередью статуса
  - Поддержка 4-х направлений потоков данных
  - PCIe->PCIe, PCIe->AXI, AXI->PCIe и AXI->AXI
  -

1.3.10.2.1.14.2 Регистры канала DMA

В каждом канале DMA реализован независимый набор регистров для управления работой данного канала DMA. Регистры предназначены для управления очередями статуса фрагментирования данных и завершения, а также управления и статуса DMA.

Регистры DMA реализованы в структурах, которые могут быть реализованы, в основном, в RAM, чтобы обеспечить эффективную поддержку большого количества каналов DMA.

При использовании многофункциональных/с поддержкой SR-IOV ядер PCI Express каналы DMA равномерно распределены между всеми функциями.

Очереди DMA должны занимать смежные области памяти, а адреса должны быть выравнены до 64 байт, чтобы обеспечить эффективное чтение/запись нескольких элементов очереди за одну операцию чтения/записи. Это обеспечивает более высокую пропускную способность для пересылок DMA малого размера, поскольку для нескольких пересылок DMA малого размера могут быть уменьшены издержки операций управления очередью.

Очереди фрагментирования данных чтения и записи описывают фрагментирование данных чтения и записи соответственно. Эти очереди независимы. Канал DMA объединяет информацию от очередей, чтобы выполнять операции DMA с учетом фрагментации каждой очереди. Это освобождает программу от этой задачи, требующей большого объема вычислений, и от необходимости передавать информацию о фрагментировании между доменами PCIe and AXI.

Прежде чем разрешить канал DMA, необходимо инициализировать очередь фрагментирования данных чтения, очередь статуса завершения чтения и очередь статуса завершения записи. Рекомендуется следующая процедура разрешения канала DMA. Последвотельность действий может быть любой, при условии, что запись 1 в DMA Enable будет последней.

- Убедиться, что канал DMA свободен
  - Считать DMA Running и проверить, что считывается 0x0.. В противном случае выполнить процедуру запрета активного канала DMA (раздел 13.9).
- Инициализировать базовый адрес и атрибуты очереди
  - Записать в SRC_Q_PTR_LO и SRC_Q_PTR_HI базовый адрес очереди
  - Записать в DST_Q_PTR_LO и DST_Q_PTR_HI базовый адрес очереди
  - Записать в STAS_Q_PTR_LO и STAS_Q_PTR_HI базовый адрес очереди
  - Записать в STAD_Q_PTR_LO и STAD_Q_PTR_HI базовый адрес очереди
- Инициализировать размер очереди
  - Записать в SRC_Q_SIZE размер очереди
  - Записать в DST_Q_SIZE размер очереди
  - Записать в STAS_Q_SIZE размер очереди
  - Записать в STAD_Q_SIZE размер очереди
- Инициализировать указатели Next, установив их на начало очереди
  - Записать SRC_Q_NEXT = 0x0
  - Записать DST_Q_NEXT = 0x0
  - Записать STAS_Q_NEXT = 0x0
  - Записать STAD_Q_NEXT = 0x0
- Инициализировать очереди фрагментирования в состояние "пусто" (отсутствуют операции DMA для выполнения)
  - Записать SRC_Q_LIMIT = 0x0
  - Записать DST_Q_LIMIT = 0x0
- Инициализировать очереди статуса в состояние "полный доступ" (доступны все элементы очереди статуса, за исключением одного, который необходим для программного управления потоком)
  - STAS_Q_LIMIT устанавливается в (STAS_Q_SIZE-1)
  - STAD_Q_LIMIT устанавливается в (STAD_Q_SIZE-1)
- Инициализировать все элементы очередей STAS и STAD, записав в них 0x0
  - Элементы очереди статуса завершения DMA должны быть установлены в 0x0, чтобы программа могла определить, когда элементы статуса завершены. Элементы статуса возвращают значение, отличное от 0, когда они завершены.
- Опционально инициализировать все элементы очередей фрагментирования данных SRC и DST
  - Элементы SGL записи и чтения могут инициализироваться или не инициализироваться по усмотрению программы. Выборка элементов SGL записи и чтения будет производиться только после того, как они будут заполнены инструкциями транзакции DMA, а соответствующий указатель очереди LIMIT перемещен, чтобы эти элементы были доступны для выполнения каналу DMA.
- Записать DMA Enable = 1, чтобы разрешить канал DMA.

Транзакции DMA передаются каналу DMA для выполнения записью SGL источника и SGL приемника в очереди SRC/DST SGL и инкрементированием регистров SRC/DST_Q_LIMIT.

Когда канал DMA завершает работу (завершает пересылку SGL источника с EOP ==1), очереди статусов завершения STAS/STAD обновляются статусом завершения DMA, и вырабатывается прерывание DMA (как определяется объединенным счетчиком прерываний каналов DMA и другими регистрами).

Регистры управления очередью *_PTR_LO, *_PTR_HI, *_SIZE и *_NEXT не должны записываться, когда канал DMA разрешен. Из всех регистров очереди допускается изменять содержимое, когда канал DMA разрешен, только у регистров SRC/DST/STAS/STAD_Q_LIMIT, которые инкрементируются для предоставления дополнительных элементов для работы канала DMA.

Минимальный размер очереди - 2. Причина заключается в том, что программа должна всегда иметь в своем распоряжении один элемент очереди для управления потоком. Канал DMA не будет выполнять/использовать элемент очереди, на который указывает указатель очереди LIMIT.

Минимальный размер очереди доложен быть достаточно большим, чтобы вмещать как минимум 1 полную транзакцию DMA максимального размера. Очереди статусов завершения DMA записываются только тогда, элемент SGL источника завершен с флагом EOP == 1 (окончание транзакции DMA). Если очередь слишком мала, чтобы вместить в очереди все SGL для одной транзакции DMA, то SGL с EOP == 1 не будет добавлена в очередь, и операция DMA не может быть завершена. S/W не сможет освободить элементы очереди, поскольку ни один SGL с EOP не может завершиться (потому что их нет в очереди), а новый SGL не может быть предоставлен каналу DMA, пока элементы очереди не освобождены.

Очередь размером N содержит N элементов: [0], [1], …, [N-1]. Например, очередь с размером 2 имеет элементы [0] и[1]. Очередь зацикливается после элемента N-1. Например, для очереди с размером 2 зацикливание происходит следующим образом: [0], [1], [0], …

Обычно начальная установка очередей DMA производится один раз, а затем они используются для множества операций DMA. Очереди DMA предназначены для выполнения транзакций со значительной степенью перекрытия. Программа может поместить новые операции DMA в очередь, пока канал DMA выполняет операции, которые программа поместила в очередь раньше.

При желании начальная установка очередей DMA может выполняться для каждой транзакции, но этот метод менее эффективен из-за необходимости повторной конфигурации очередей между транзакциями DMA. Повторная конфигурация очередей DMA может проводиться только когда канал DMA запрещен. Поэтому процесс должен иметь следующий вид: подождать, пока завершаться все ожидающие транзакции канала DMA, запретить канал, повторно конфигурировать очереди, затем снова разрешить канал DMA.

Поскольку операции, выполняемые DMA, определяются элементами фрагментирования чтения и записи, на которые указывают очереди, а не расположены в очередях, то очереди не содержат конкретную информацию пересылки DMA. Поэтому те же очереди могут использоваться для разных транзакций DMA.

Когда транзакция DMA завершается (программа считывает элемент очереди статуса завершения текущего DMA), программа обрабатывает результирующие данные DMA и сбрасывает элементы очереди SGL записи/чтения, а также соответствующие элементы очереди статуса завершения записи/чтения DMA. При сбросе в элементы очереди статуса должно быть записано значение 0x0, поскольку ненулевое значение, считанное программой из элемента статуса, воспринимается как указание на завершенную пересылку DMA. Сброшенные элементы очереди повторно используются, когда указатель очереди LIMIT возвращается к сброшенным элементам.

1.3.10.2.1.14.3 Карта адресов регистров канала DMA

В каждом канале DMA реализовано 128 (0x80) байт регистров DMA. Доступ к регистрам канала DMA может осуществляться через PCI Express и/или интерфейс ведомого устройства AXI, так что DMA можно инициализировать/управлять с помощью программного обеспечения PCIe или AXI , или обоих.

1.3.10.2.1.14.3.1 Карта регистров AXI канала DMA

Доступ к регистрам канала DMA осуществляется через интерфейс ведомого устройства AXI, когда транзакция AXI попадает в адресный регион этой трансляции регистров DMA. Для доступа со стороны интерфейса ведомого устройства AXI все регистры канала DMA расположены друг за другом.

Например, программное обеспечение AXI получает доступ к N 128-битным каналам DMA через трансляцию регистров DMA интерфейса ведомого устройства AXI.

Смещение адреса регистра DMA	DMA канал
0x0	0
0x80	1
0x100	2
…	…
(N-1)*0x80	N-1

1.3.10.2.1.14.3.2 Карта регистров PCI Express канала DMA

Доступ к регистрам канала DMA осуществляется через интерфейс PCI Express при помощи регистра базового адреса PCI Express, соответствующего регистрам канала DMA (как указано в fg_dma_reg_bar). При наличии нескольких функций PCIe (ядро PCI Express имеет поддержку SR-IOV и/или многофункциональности) каждой функции распределяется равное число каналов DMA, доступ к которым осуществляется через соответствующий функции регистр BAR. Регистры каналов DMA для одной функции расположены друг за другом.

Распределение каналов DMA по функциям PCI Express определяется входом ядра Expresso DMA Bridge p_rx_f, и поэтому зависит от поведения соответствующего ядра PCI Express. Ядро Expresso DMA Bridge не знает (и ему не нужно знать), является функция PCIe регулярной, или физической или виртуальной функцией SRIOV. Ядро Expresso DMA Bridge обращается со всеми функциями одинаково.

Следующие примеры описывают распределение регистров DMA, когда ядро Expresso DMA Bridge используется совместно с ядром Expresso 3.0 компании Northwest Logic (ядро PCI Express).

Например, при одной функции PCIe программа PCIe получает доступ к N каналам DMA по 128 байт через PCI Express следующим образом:

Смещение адреса	DMA канал
(DMA Reg BAR) + 0x00	0
(DMA Reg BAR) + 0x80	1
(DMA Reg BAR) + 0x100	2
…	…
(DMA Reg BAR) + (N- 1)*0x80	N-1

Например, при 4 функциях PCIe без SRIOV программа получает доступ к 16 каналам DMA по 64 байта через PCI Express следующим образом:

Смещение адреса	DMA канал	Функция
Function[0] DMA Reg BAR + 0x00	0	

Function[0] DMA Reg BAR + 0x40	1	0
Function[0] DMA Reg BAR + 0x80	2	
Function[0] DMA Reg BAR + 0xC0	3	
Function[1] DMA Reg BAR + 0x00	4	1
Function[1] DMA Reg BAR + 0x40	5	
Function[1] DMA Reg BAR + 0x80	6	
Function[1] DMA Reg BAR + 0xC0	7	
Function[2] DMA Reg BAR + 0x00	8	2
Function[2] DMA Reg BAR + 0x40	9	
Function[2] DMA Reg BAR + 0x80	10	
Function[2] DMA Reg BAR + 0xC0	11	
Function[3] DMA Reg BAR + 0x00	12	3
Function[3] DMA Reg BAR + 0x40	13	
Function[3] DMA Reg BAR + 0x80	14	
Function[3] DMA Reg BAR + 0xC0	15	

Например, при 1 физической функции и 3 виртуальных функциях (всего 4) программа PCIe получает доступ к 16 каналам DMA по 64 байта через PCI Express следующим образом:

Смещение адреса	DMA канал	Функция	Наименование функции
PF0 DMA Reg BAR + 0x00	0	0	PF0
PF0 DMA Reg BAR + 0x40	1		
PF0 DMA Reg BAR + 0x80	2		
PF0 DMA Reg BAR + 0xC0	3		
(PF0:VF DMA Reg BAR) + 0*(PF0:VF_BAR_SIZE) + 0x00	4		

Смещение адреса	DMA канал	Функция	Наименование функции
(PF0:VF DMA Reg BAR) + 0*(PF0:VF_BAR_SIZE) + 0x40	5		
(PF0:VF DMA Reg BAR) + 0*(PF0:VF_BAR_SIZE) + 0x80	6		PF0, VF1
(PF0:VF DMA Reg BAR) + 0*(PF0:VF_BAR_SIZE) + 0xC0	7		
(PF0:VF DMA Reg BAR) + 1*(PF0:VF_BAR_SIZE) + 0x00	8		
(PF0:VF DMA Reg BAR) + 1*(PF0:VF_BAR_SIZE) + 0x40	9	2	PF0, VF2
(PF0:VF DMA Reg BAR) + 1*(PF0:VF_BAR_SIZE) + 0x80	10		
(PF0:VF DMA Reg BAR) + 1*(PF0:VF_BAR_SIZE) + 0xC0	11		
(PF0:VF DMA Reg BAR) + 2*(PF0:VF_BAR_SIZE) + 0x00	12		
(PF0:VF DMA Reg BAR) + 2*(PF0:VF_BAR_SIZE) + 0x40	13	3	PF0, VF3
(PF0:VF DMA Reg BAR) + 2*(PF0:VF_BAR_SIZE) + 0x80	14		
(PF0:VF DMA Reg BAR) + 2*(PF0:VF_BAR_SIZE) + 0xC0	15		

Например, при SRIOV с 2 физическими функциями и 3 виртуальными функциями для каждой физической (всего 8) программа PCIe получает доступ к 32 каналам DMA по 64 байта через PCI Express следующим образом:

Смещение адреса	DMA канал	Функция	Наименование функции
(PF0 DMA Reg BAR + 0x00)	0		
PF0 DMA Reg BAR + 0x40	1		
(PF0 DMA Reg BAR + 0x80)	2		
(PF0 DMA Reg BAR + 0xC0)	3	0	PF0
(PF1 DMA Reg BAR + 0x00)	4		
(PF1 DMA Reg BAR + 0x40)	5		
(PF1 DMA Reg BAR + 0x80)	6		
(PF1 DMA Reg BAR + 0xC0)	7	1	PF1

Смещение адреса	DMA канал	Функция	Наименование функции
(PF0:VF DMA Reg BAR) + 0*(PF0:VF_BAR_SIZE) + 0x00	8		
(PF0:VF DMA Reg BAR) + 0*(PF0:VF_BAR_SIZE) + 0x40	9		
(PF0:VF DMA Reg BAR) + 0*(PF0:VF_BAR_SIZE) + 0x80	10		
(PF0:VF DMA Reg BAR) + 0*(PF0:VF_BAR_SIZE) + 0xC0	11	2	PF0, VF1
(PF0:VF DMA Reg BAR) + 1*(PF0:VF_BAR_SIZE) + 0x00	12		
(PF0:VF DMA Reg BAR) + 1*(PF0:VF_BAR_SIZE) + 0x40	13		
(PF0:VF DMA Reg BAR) + 1*(PF0:VF_BAR_SIZE) + 0x80	14		
(PF0:VF DMA Reg BAR) + 1*(PF0:VF_BAR_SIZE) + 0xC0	15	3	PF0, VF2
(PF0:VF DMA Reg BAR) + 2*(PF0:VF_BAR_SIZE) + 0x00	16		
(PF0:VF DMA Reg BAR) + 2*(PF0:VF_BAR_SIZE) + 0x40	17		
(PF0:VF DMA Reg BAR) + 2*(PF0:VF_BAR_SIZE) + 0x80	18		
(PF0:VF DMA Reg BAR) + 2*(PF0:VF_BAR_SIZE) + 0xC0	19	4	PF0, VF3
(PF1:VF DMA Reg BAR) + 0*(PF1:VF_BAR_SIZE) + 0x00	20		
(PF1:VF DMA Reg BAR) + 0*(PF1:VF_BAR_SIZE) + 0x40	21		
(PF1:VF DMA Reg BAR) + 0*(PF1:VF_BAR_SIZE) + 0x80	22		
(PF1:VF DMA Reg BAR) + 0*(PF1:VF_BAR_SIZE) + 0xC0	23	5	PF1, VF1
(PF1:VF DMA Reg BAR) + 1*(PF1:VF_BAR_SIZE) + 0x00	24	6	PF1, VF2
(PF1:VF DMA Reg BAR) + 1*(PF1:VF_BAR_SIZE) + 0x40	25		
(PF1:VF DMA Reg BAR) + 1*(PF1:VF_BAR_SIZE) + 0x80	26		
(PF1:VF DMA Reg BAR) + 1*(PF1:VF_BAR_SIZE) + 0xC0	27		

Смещение адреса	DMA канал	Функция	Наименование функции
(PF1:VF DMA Reg BAR) + 2*(PF1:VF_BAR_SIZE) + 0x00	28	7	PF1, VF3
(PF1:VF DMA Reg BAR) + 2*(PF1:VF_BAR_SIZE) + 0x40	29		
(PF1:VF DMA Reg BAR) + 2*(PF1:VF_BAR_SIZE) + 0x80	30		
(PF1:VF DMA Reg BAR) + 2*(PF1:VF_BAR_SIZE) + 0xC0	31		

1.3.10.2.1.14.4 Управление потоком канала DMA

Каждый канал DMA содержит три внутренних FIFO:

- Один FIFO для кэширования до 8 SGL источников
  - Заполняется при чтении очереди SGL источника, когда SRC_Q_LIMIT делает SGL доступной.
  - Очищается, когда транзакции DMA созданы и полностью соответствуют размеру текущего элемента SGL источника.
- Один FIFO для кэширования до 8 SGL приемников
  - Заполняется при чтении очереди SGL приемника, когда DST_Q_LIMIT делает SGL доступной.
  - Очищается, когда транзакции DMA созданы и полностью соответствуют размеру текущего элемента SGL приемника.
- Один FIFO для кэширования до 4 статусов завершения DMA
  - Заполняется при пересылке DMA с завершением SRC SGL EOP == 1
  - Очищается, когда статус завершения DMA записывается в очереди статуса завершения DMA (STAS/SATD).

Канал DMA осуществляет арбитраж с другими каналами DMA и функциями моста по циклическому алгоритму, чтобы выполнить операции DMA. Канал DMA будет осуществлять арбитраж, чтобы выполнить пересылку DMA, когда канал DMA разрешен, элемент SGL источника доступен во внутреннем кэше FIFO, элемент SGL приемника доступен во внутреннем кэше FIFO, а в FIFO статуса завершения канала DMA содержится хотя бы один элемент, доступный для приема статуса завершения DMA. Размер транзакции DMA, для которой выполняется арбитраж, может составлять до 512 байт для линий PCI Express от x1 до x4, и до 1024 байт для линий PCI Express x8 и x16. Когда SGL источника DMA с EOP == 1 завершается, статус завершения DMA записывается внутреннее FIFO статуса завершения канала DMA, пока не появляется возможность его записи во внешнюю очередь STA(S/D) статуса завершения DMA.

1.3.10.2.1.14.5 Очереди фрагментирования и статуса

Каналы DMA способны выполнять фрагментирование отдельно для источника и приемника DMA. Очереди фрагментирования источника и приемника управляются программой, и канал DMA должен описать программе желаемые операции DMA. Регистры очереди DMA указывают начальные смещения адресов очередей, размер очередей и организуют передачу управления потоком данных между программой и аппаратной частью канала DMA.

Программа снабжает очереди источника и приемника элементами, позволяющими каналу DMA выполнить операцию DMA. Когда канал DMA разрешен, доступные элементы очередей источника и приемника помещаются в маленький внутренний буфер, необходимый для завершения операций DMA.

Элементы очереди имеют фиксированный размер. Адрес конкретного элемента очереди определяется как начальный адрес + номер элемента, умноженный на размер элемента.

Элементы очереди фрагментирования источника имеют размер 128 бит:

- [63:0] - Адрес источника [63:0]
- [87:64] - ByteCount[23:0]
  - 0 == 2^24 bytes
- [95:88] - Flags[7:0]
  - [7:4] – Атрибуты чтения данных DMA
- Если источник AXI, то [7:4] используется для m_arcache[3:0]
- Если источник PCIe, то [6:4] используется для Attr[2:0]
  - [3] - Зарезервировано
  - [2] – Прерывание
    - 1 - Генерировать событие прерывания, когда в очередь статуса записывается статус завершения DMA; действительно только при EOP == 1; прерывание будет сгенерировано на PCIe или AXI регистре прерывания канала DMA, или на обоих.
    - 0 – Не генерировать прерывание
  - [1] – EOP
    - 1 - окончание пакета; записывается элемент очереди статуса, содержащий статус окончания пакета DMA, когда окончание пакета передается к приемнику DMA.
    - 0 - пакет не закончился; пакеты могут занимать несколько дескрипторов
  - [0] – Местоположение
    - 1 – Источник данных DMA - AXI (чтение из AXI посредством ведущего устройства AXI)
    - 0 – Источник данных DMA - PCIe (чтение из PCIe посредством ведущего устройства PCIe)
- [111:96] - UserHandle[15:0]
  - UserHandle копируется из SRC SGL с EOP==1 в соответствующее поле UserHandle элемента очереди статуса завершения DMA источника.
- [127:112] – UserID[15:0]
  - UserID копируется из SRC SGL с EOP==1 в соответствующее поле UserHandle элементов очереди статуса завершения DMA источника и приемника.

Элементы очереди фрагментирования приемника имеют размер 128 бит:

- [63:0] - Destination Address[63:0]
- [87:64] - ByteCount[23:0]
  - 0 == 2^24 bytes
- [95:88] - Flags[7:0]
  - [7:4] – Атрибуты записи данных DMA
    - Если приемник AXI, то [7:4] используется для m_awcache[3:0]
    - Если приемник PCIe, то [6:4] используется для Attr[2:0]
      - [3:2] – Зарезервировано
      - [1] – Разрешить один пакет на SGL приемника
    - SGL текущего источника и SGL текущего приемника объединяются для одной пересылки DMA. Когда элемент SGL текущего источника занят полностью, а его поле EOP установлено в 1, указывая на конец пакета, следующий пакет может быть записан либо начиная с оставшейся части (при наличии) SGL текущего приемника, либо оставшаяся часть (при наличии) SGL текущего приемника

пропускается, и следующий пакет записывается с начала следующего элемента SGL приемника. Этот бит определяет выбор поведения.

- 1 – Перейти к следующей SGL приемника при окончании пакета
- При пересылке последнего пакета (что определяется полем EOP SGL источника) оставшаяся часть (при наличии) элемента SGL текущего приемника не используется. Следующий пакет начинает записываться в начало области памяти, описанной следующим элементом SGL приемника.
- 0 - Поместить пакеты вплотную друг за другом в SGL приемника.
- Когда пакет заканчивается, не заняв всю SGL приемника, следующий пакет начинает записываться в оставшуюся часть SGL приемника с бита, следующего за концом предыдущего пакета.
  - [0] – Местоположение
- 1 – Приемник данных DMA - AXI (запись в AXI посредством ведущего устройства AXI)
- 0 – Приемник данных DMA - PCIe (запись в AXI посредством ведущего устройства PCIe)

- [111:96] - UserHandle[15:0]
  - UserHandle копируется из последнего элемента DST SGL, использованного при пересылке пакета, в соответствующее поле UserHandle элемента очереди статуса завершения DMA приемника.
- [127:112] - Зарезервировано[15:0]

Элементы очереди статуса имеют размер 32 бита или 64 бита, что определяется при программировании регистров каналов DMA. 32-битные элементы содержат первые 32 бита информации о статусе - ie [31:0]. 64-битные элементы содержат все 64 бита информации о статусе - ie [63:0].

- [63:48] – UserID[15:0]

UserID копируется из SRC SGL с EOP==1 в соответствующие поля UserHandle элементов очереди статуса завершения DMA источника и приемника.

UserID определяется применением. Канал DMA копирует, но сам не использует эту информацию.

- [47:32] – UserHandle[15:0]
  - Очередь статуса завершения источника DMA - UserHandle копируется из SRC SGL с EOP==1.
  - Очередь статуса завершения приемника DMA - UserHandle копируется из последней SGL приемника, использованной для пакета.
  - Использование UserHandle определяется применением. Канал DMA копирует, но сам не использует эту информацию.
  - Старшие разряды статуса ненулевые
  - Для 64-битных элементов статуса Upper Status Is Non Zero == 1 когда биты элемента статуса [63:32] != 0. В некоторых случаях CPU способны выполнять только 32-битные атомарные транзакции. Для таких CPU программа считывает биты статуса [31:0]. Если бит[31] == 1, то программа знает, что Status[63:32] != 0 и считывает биты статуса [63:32]
  - Программа должна очистить все элементы очереди статуса, прежде чем передать их в канал DMA, так чтобы старшие биты статуса оставались нулевыми, пока не изменятся по завершении DMA.
  - Для 32-битных элементов статуса это поле не используется и всегда считывается 0.
- [30:4] – Completed Byte Count[26:0]
  - Диапазон от 0 до (2^27)-1 байт (от 0 до 128МБ-1). Если DMA превышает (2^27)-1 байт, Completed Byte Count ограничивается максимальным значением. Если программе требуется пересылать пакеты большие, чем 128МБ-1, то

программа должна разделить такие пакеты на пакеты меньшего размера, < 128 MB.

- [3] – Во время операции DMA обнаружена внутренняя ошибка
- [2] – Во время операции DMA обнаружена ошибка приемника
- [1] – Во время операции DMA обнаружена ошибка источника
- [0] - Завершено

Элементы очереди статуса содержат статус для одного пересылаемого пакета. CPU должны считывать элементы очереди статуса при помощи атомарных транзакций (которые должны быть не меньше 32 бит), чтобы гарантировать, что вся информация, считанная из очереди статуса, была обновлена и принадлежит одному завершению DMA. 64-битные элементы очереди статуса содержат метод, позволяющий CPU, которые допускают только 32-битную атомарность, правильно ассоциировать все 64 бита элементов очереди статуса. См. выше, поле ненулевых старших бит статуса.

Если во время операции DMA произошла ошибка, данным доверять нельзя. Ошибки возникают редко и указывают на серьезную проблему, требующую устранения. После того, как произошла ошибка, канал DMA должен быть запрещен, а любым активным операциям должно быть предоставлено время для завершения. В зависимости от серьезности ошибки канал DMA может или не может быть использован повторно без сброса ядра Expresso DMA Bridge.

Следует отметить, что ошибки DMA передаются в элементы очереди статуса только для пакетов, которые могут быть завершены. Ошибки, которые не позволяют пакетам завершиться, передаются только в регистры статуса канала PCIe DMA Error и AXI DMA Error. Для длинных пересылок DMA более вероятно, что ошибки будут зафиксированы только в регистрах ошибок канала DMA.

1.3.10.2.1.14.6 Назначение вектора прерываний PCI Express канала DMA

Прерывания MSI-X и MSI канала DMA, в том числе программные прерывания PCIe, инициируются при помощи вектора прерываний MSI-X/MSI, соответствующего номеру канала DMA. Например, прерывания канала DMA [0] инициируются на MSI-X/MSI векторе 0, прерывания канала DMA [1] инициируются на MSI-X/MSI векторе 1, и т.д.

1.3.10.2.1.14.7 Ограничения на переключение направления потока данных канала DMA

Поскольку элементы фрагментирования содержат поля, которые определяют направление потока данных DMA, то представляется возможным поручить одному каналу DMA одновременно поддерживать пересылки с разным направлением потока данных, однако эта возможность НЕ поддерживается.

Один канал DMA может поддерживать только одно направление потока данных. Например, канал DMA 0 можно использовать только для передачи в одном из следующих направлений: PCIe->AXI, AXI->PCIe, PCIe->PCIe, или AXI->AXI. Канал DMA 0 не может использоваться для поддержки потоков данных PCIe->AXI и AXI->PCIe одновременно. Перед сменой направления потока данных DMA на одном канале все операции предыдущего потока данных должны быть полностью завершены. Ограничений на направление потоков данных между разными каналами DMA нет. Например, канал DMA 0 может поддерживать направление PCIe->AXI, канал DMA 1 - AXI->PCIe, канал DMA 2 - PCIe->PCIe, а канал DMA 3 - AXI-AXI.

1.3.10.2.1.14.8 Запрещение активного канала DMA

Каждый канал DMA может иметь несколько ожидающих транзакций в разных частях системы. Операции, которые могут быть в процессе выполнения:

- Чтение очередей SGL источника и приемника
- Запись очереди статуса завершения DMA
- Транзакции чтения данных DMA
- Транзакции записи данных DMA

Кроме того, в каждом канале DMA реализованы небольшие FIFOs для хранения элементов SGL источника и приемника, а также статуса завершения DMA, чтобы обеспечить высокую пропускную способность для одновременно выполняющихся операций DMA.

Транзакции DMA для каждого канала DMA и функции моста DMA используют общее внутреннее соединение и поток данных.

Из-за высокой степени перекрытия операций канала DMA и общего потока данных необходима возможность запрещения операций канала DMA, чтобы минимизировать воздействие на работу других каналов DMA, а также на трафик моста.

Рекомендуемый процесс запрещения действующего канала DMA:

- Программа перестает добавлять новые пересылки DMA к очередям SGL источника и приемника
  - Последние элементы очередей SGL источника и приемника, направленные для выполнения каналу DMA (через регистры RC/DST_Q_LIMIT) должны быть полностью использованы последней пересылкой DMA
- Программа ждет завершения всех ожидающих операций DMA и обрабатывает статус завершения DMA из очереди статуса завершения DMA. Программа должна предусматривать тайм-аут на тот случай, если операции DMA не будет завершены. Чаще всего это происходит из-за того, что программа не предоставила соответствующих элементов SGL источника и приемника, которые могут быть полностью использованы.
- Программа записывает DMA_Enable == 0 для данного канала DMA, чтобы запретить канал DMA
- Программа считывает DMA_Running для данного канала DMA. Если DMA_Running == 0, то канал DMA закончил все ожидающие транзакции, и программа может опционально переходить к шагу 6.
- Если DMA_Running == 1, то канал DMA не смог завершить все ожидающие транзакции. Одна из внутренних очередей SGL источника, SGL приемника или статуса завершения DMA не пуста. Программа записывает MA_Reset == 1, ждет >= 256 nS, и записывает DMA_Reset ==
- 0. Установка DMA_Reset == 1 очищает внутренние FIFO SGL источника, SGL приемника и FIFO статуса завершения DMA.
- Теперь канал DMA готов к повторному использованию.

1.3.10.2.1.14.9 Работа DMA с маленькими пакетами

Ядро Expresso DMA Bridge способно максимизировать пропускную способность, когда таблица фрагментирования DMA (дескрипторы) и соответствующие пакеты DMA относительно велики. Граничный порог размера SGL/пакета, ниже которого пропускная способность начинает падать, в первую очередь зависит от количества и размера запросов чтения/выборок SGL которые ядро может одновременно иметь в ожидании, и задержки, с которой возвращаются запрошенные данные.

По умолчанию конфигурация ядра Expresso DMA Bridge выбирается таким образом, чтобы получить высокую пропускную способность PCIe чтения DMA для типичных задержек чтения PCIe, когда DMA SGL/пакеты имеют размер 4КБ или больше. Для систем с большей, чем обычно, задержкой чтения PCIe или меньшим размером DMA SGL/пакетов пропускная способность может уменьшиться.

По умолчанию конфигурация ядра Expresso DMA Bridge выбирается таким образом, чтобы получить высокую пропускную способность AXI чтения DMA для типичных задержек чтения AXI, когда DMA SGL/пакеты имеют размер 512 байт или больше. Для систем с большей, чем обычно, задержкой чтения AXI или меньшим размером DMA SGL/пакетов пропускная способность может уменьшиться.

Если пользовательское применение DMA требует DMA SGL/пакетов размером меньше 4 КБ, пользователю рекомендуется обратиться к Northwest Logic, прежде чем лицензировать ядро. Во многих случаях компания Northwest Logic сможет настроить конфигурацию ядра (повысить количество ожидающих транзакций чтения, повысить количество одновременно выбираемых SGL и т.д.), и поставить ядро с высокой пропускной способностью для размеров SGL/пакетов пользователя. Такие изменения должны вноситься Northwest Logic до лицензирования ядра и могут быть организованы Northwest Logic до лицензирования ядра. При поддержке меньших по размеру SGL/пакетов увеличиваются размеры логики и RAM ядра, поскольку для достижения большой пропускной способности при меньших размерах SGL/Packet необходим большой объем дополнительной информации для их отслеживания/ожидания.

Существует два главных фактора, ограничивающих пропускную способность DMA с маленьким размером пакетов:

- Задержка чтения данных источника DMA
  - Задержка чтения из PCI Express может быть значительной - обычно несколько сотен нс.
  - Задержка чтения из AXI меньше, но тоже значительная - обычно больше ста нс.
  - Для максимизации пропускной способности DMA необходимо ожидание достаточного количества запросов чтения одновременно, чтобы компенсировать задержку чтения.
    - Например: Канал x8 PCIe со скоростью 8G и задержкой чтения 768 nS потребует 6КБ ожидающих запросов чтения (768ns*8bytes/nS == 6KBytes), чтобы компенсировать задержку чтения и поддержать максимальную пропускную способность.
    - 6КБ == 12 запросов чтения по 512 байт
    - 6КБ == 96 запросов чтения по 64 байт
- Эффективность DMA
  - Для каждого пакета DMA обычно требуется 32 байта информации SGL, чтобы описать операцию DMA, и 4 байта статуса обычно требуется для передачи статуса замещения DMA. Для пакета DMA маленького размера, например, 64 байта, служебная информация сравнима с размером содержимого пакета DMA, что существенно снижает возможную пропускную способность DMA.
- Эффективность PCIe
  - Каждый пакет TLP PCI Express требует 20-28 байт служебной информации для заголовка TLP, фрейминга, LCRC, и ECRC. Высокая пропускная способность требует использования содержимого большего размера (от 128 до 512 байт), чтобы уменьшить служебную информацию TLP по отношению к данным содержимого.
  - AXI использует боковую полосу частот для управляющих сигналов (адрес, счетчик байтов и т. д.), и поэтому его пропускная способность не уменьшается. как у PCIe.

Для повышения эффективности передачи пакетов малого размера ядро Expresso DMA Bridge содержит следующие функции:

- Ядро Expresso DMA Bridge при каждом чтении по возможности делает выборку нескольких SGL.
- Ядро Expresso DMA Bridge выполняет несколько ожидающих операций чтения.
- Все операции DMA в значительной степени перекрываются.

1.3.10.2.1.14.10 Обработка ошибок DMA

Обнаруживаются и обрабатываются следующие ошибки DMA:

- Ошибки, появляющиеся при чтении SGL из SRC или DST QUEUE, или при чтении данных DMA
  - Ошибки, обнаруженные и сообщенные исполнителем операции чтения
    - Чтение PCIe:
    - Операция чтения возвращает статус неподдерживаемого запроса или Completer Abort
    - p_rx_err == 1 (ошибка ECC/четности или ECRC, обнаруженная ядром PCIe)
    - Искаженный пакет TLP (EP == 1 в принятом пакете TLP, который содержит данные)
    - Чтение AXI
    - Операция чтения возвращает ошибку AXI (m_rresp != OKAY)
  - Не получен отклик (тайм-аут завершения)
  - Ошибка ECC внутреннего RAM при обработке SGL или данных DMA.
- Ошибки, возникающие во время записи статуса завершения DMA в STA QUEUE или во время записи данных DMA.
  - Ошибка, возвращенная для операции записи STA(S/D) QUEUE или DMA.
    - PCIe не отвечает на запись в память
    - Операция записи возвращает ошибку AXI (m_wresp != OKAY)
  - Внутренняя ошибка ECC при выполнении записи STA(S/D) QUEUE или DMA.
- Указатель LIMIT за границей диапазона при разрешенном DMA.
  - SRC_LIMIT >= SRC_SIZE
  - DST_LIMIT >= DST_SIZE
  - STA_LIMIT >= STA_SIZE
- Указатель NEXT за границей диапазона при разрешенном DMA.
  - SRC_NEXT >= SRC_SIZE
  - DST_NEXT >= DST_SIZE
  - STA_NEXT >= STA_SIZE
- Недопустимый размер очереди SIZE при разрешенном DMA
  - SRC_SIZE < 2
  - DST_SIZE < 2
  - STA_SIZE < 2

Ядро The Expresso DMA Bridge ассоциирует ошибки DMA с каналом DMA источника и сообщает об ошибках для каждого канала отдельно.

При появлении ошибки DMA она обрабатывается каналом DMA следующим образом:

- DMA Enable устанавливается в 0 (если был 1)
  - Поскольку DMA Enable == 0, никакие операции DMA больше не планируются.
  - Операции DMA, начатые когда DMA был разрешен, будут завершаться.
    - Следует ожидать, что после обнаружения ошибки активность DMA на короткое время сохранится, а уже исполняемые транзакции завершатся
- Регистры PCIe DMA Error и AXI DMA Error устанавливаются в 1, чтобы зарегистрировать ошибку
- Событие прерывания PCIe и AXI планируется и отражается в статусе прерывания PCIe и в статусе прерывания AXI.
  - Прерывания ошибок обрабатываются так же, как обычные прерывания
    - Один и тот же вектор прерывания используется для обычных прерываний завершения DMA
    - Прерывание будет генерироваться только тогда, когда прерывания разрешены

- PCIe Interrupt Enable == 1
- AXI Interrupt Enable == 1
- Программа может считывать PCIe DMA Error и/или AXI DMA Error, если прерывание было вызвано ошибкой

Если транзакция DMA полностью завершается, так что ее элемент STA QUEUE записывается обычным образом, то любые ошибки, обнаруженные во время DMA, будут отражены в элементе STA_QUEUE, записанном во время завершения транзакции. Следует отметить, что такое часто случается при транзакциях DMA маленького объема и редко большого объема, поскольку после обнаружения ошибки DMA все операции DMA останавливаются, и во многих случаях это означает, что транзакция DMA с ошибкой не завершится, и STA QUEUE не будет записан.

При возникновении ошибки программа должна обрабатывать ошибку так, как считает необходимым. Если программа после обнаружения ошибки намерена продолжать работу без системного сброса, то канал DMA должен быть сброшен, прежде чем его можно будет использовать снова.

1.3.10.2.1.15 Максимальный размер транзакции и число незавершенных транзакций

Ядро Expresso DMA Bridge может быть сконфигурировано на разное количество незавершенных PCIe и AXI транзакций, чтобы минимизировать размер разработки, одновременно обеспечивая максимальную пропускную способность.

Чтобы скомпенсировать задержки чтения и записи в системе и поддержать максимальную пропускную способность, ядро Expresso DMA Bridge (а также внутреннее соединение AXI и исполнение пользователя) должно быть сконфигурировано для того, чтобы допускать ожидаемую задержку без приостановки транзакций из-за недостатка ресурсов. С повышением разрядности данных и скорости ядра за каждый цикл синхронизации передается все больший объем данных, и поэтому для поддержки данной системной задержки чтения/записи требуется больший объем буферов.

По умолчанию ядро Expresso DMA Bridge конфигурируется так, чтобы компенсировать следующие задержки при работе с максимальной разрядностью и скоростью. Ядро Expresso DMA Bridge может конфигурироваться компанией Northwest Logic для компенсации больших задержек. При необходимости свяжитесь с Northwest Logic, чтобы получить более подробную информацию о компенсации задержек.

- Задержка ответа чтения ведущего устройства PCIe
  - 1uS
- Задержка ответа записи ведущего устройства PCIe
  - Не применяется - необходимо компенсировать только внутреннюю задержку, поскольку операции запись в память по шине PCIe не требует ответа
- Задержка ответа чтения ведущего устройства AXI
  - 100 nS
- Задержка ответа записи ведущего устройства AXI
  - 100 nS

У разных интерфейсов разные требования к размеру транзакций моста и DMA. Например, интерфейс PCIe ограничен максимальным поддерживаемым размером содержимого и минимальным поддерживаемым размером запроса чтения, а разраядность интерфейсов ведущего и ведомого устройства AXI ограничены 16 словами (AXI3) данных. Программа управляет максимально допустимым размером содержимого посредством конфигурации Max Payload Size и Max Read Request Size для PCIe, а также cfg_m_max_rd_req_size и cfg_m_max_wr_req_size для AXI.

При прохождении через ядро Expresso DMA Bridge транзакции фрагментируются, чтобы соответствовать максимальному размеру пересылки соответствующих интерфейсов и внутренней логики. Транзакции также фрагментируются, когда они содержат неразрывный

набор сигналов валидности байт в транзакциях, содержащих только набор сигналов валидности байт с разрывами. Транзакции никогда не объединяются.

1.3.10.2.1.16 Максимальный размер транзакции интерфейса ведущего устройства AXI

Интерфейс ведущего устройства AXI поддерживает 4-битовое поле m_awlen/m_arlen AXI3, которое ограничивает максимальный размер транзакций AXI до разрядности 16*:

- AXI 32 бит
  - Максимальный размер пересылки AXI == 64 бита
- AXI 64 бит
  - Максимальный размер пересылки AXI == 128 бита
- AXI 128 бит
  - Максимальный размер пересылки AXI == 256 бита
- AXI 256 бит
  - Максимальный размер пересылки AXI == 512 бита

Интерфейс ведущего устройства AXI поддерживает максимум 8 незавершенных транзакций чтения одновременно. Внутри может ожидать завершения и большее количество транзакций чтения, но только 8 запросам чтения разрешено ожидать завершения на интерфейсе ведущего устройства AXI одновременно. Дополнительная фрагментация, которая выполняется из-за ограничения cfg_m_max_rd_req_size на размер запроса чтения AXI, не считается в 8 незавершенных транзакциях. Например, 8 запросов чтения по 512 байт каждый могут ожидать завершения на интерфейсе ведущего устройства AXI одновременно (для 256-битного AXI), когда cfg_m_max_rd_req_size был установлен на >= 512 байт. Те же самые 8 запросов чтения по 512 байт при cfg_m_max_rd_req_size запрограммированном на 64 байта, на интерфейсе ведущего устройства AXI появятся как 64 запросов чтения по 64 байта, одновременно ожидающих завершения.

Интерфейс ведущего устройства AXI поддерживает максимум 32 или 64 незавершенных транзакций чтения одновременно.

- PCIe x1, x2, и x4 линий – 32 незавершенные транзакции записи
- PCIe x8 и x16 линий – 64 незавершенные транзакции записи

Внутри может ожидать завершения и большее количество транзакций записи, но только 32/64 запросам записи разрешено ожидать завершения на интерфейсе ведущего устройства AXI одновременно. В отличие от cfg_m_max_rd_req_size, фрагментация транзакций из-за cfg_m_max_wr_req_size влияет на количество незавершенных транзакций записи.

Например, запросы чтения на 32 256 байт при программировании cfg_m_max_wr_req_size на 128 байт будут выданы как запросы записи на 64 128 байт, но только 32 этих запроса будут одновременно ожидать завершения на интерфейсе ведущего устройства AXI.



Рисунок 1.144 - Максимальный размер запроса чтения ведущего устройства AXI для ядра Expresso DMA Bridge

1.3.10.2.1.16.1 Максимальный размер транзакции интерфейса ведомого устройства AXI

Интерфейс ведомого устройства AXI поддерживает 4-битовое поле s_awlen/s_arlen AXI3, которое ограничивает максимальный размер транзакций AXI до разрядности 16*:

AXI 32 бит

Максимальный размер пересылки AXI == 64 бита

AXI 64 бит

Максимальный размер пересылки AXI == 128 бита

AXI 128 бит

Максимальный размер пересылки AXI == 256 бита

AXI 256 бит

Максимальный размер пересылки AXI == 512 бита

Транзакции, полученные на интерфейсе ведомого устройства AXI, не объединяются, и поэтому размер начальной транзакции - это наибольший размер, который будет обрабатываться. Транзакции интерфейса ведомого устройства AXI могут фрагментироваться, если этого требует максимальный размер транзакции приемника.

Время доставки интерфейса ведомого устройства AXI конфигурируется так, чтобы поддерживать одновременно 16 или 32 незавершенных запроса чтения. Поддержка запросов 32 чтения обеспечивается в том случае, когда очередь переупорядочивания чтения PCIe обеспечивает поддержку 32 незавершенных транзакций чтения; в противном случае поддерживаются 16 запросов чтения (очередь переупорядочивания чтения PCIe <= 16 запросов). Поддержка 32 запросов чтения может понадобиться для ядер >= x8 8G с разрядностью данных == 128, чтобы можно было обеспечить ожидание достаточного количества данных запросов чтения для компенсации задержки чтения PCIe. Ядра с меньшей разрядностью и меньшей скоростью должны обеспечивать компенсацию системных задержек, поддерживая 16 запросов чтения.

Интерфейс ведомого устройства AXI во всех конфигурациях поддерживает 16 незавершенных запросов чтения одновременно.

1.3.10.2.1.16.2 Максимальный размер транзакции ведущего устройства PCI Express

Максимальный размер инициируемого запроса чтения интерфейса PCI Express ограничен меньшим из программируемых размеров максимального размера запроса чтения PCIe и максимальным размером, поддерживаемым очередью переупорядочивания чтения PCIe. Очередь переупорядочивания чтения PCIe при необходимости переупорядочивает завершения чтения PCIe, чтобы они соответствовали требованиям упорядочивания AXI и DMA и имеет достаточно места для хранения каждого незавершенного запроса чтения, выданного ядром Expresso DMA Bridge. Максимальный размер запроса чтения очереди переупорядочивания чтения PCIe устанавливается в соответствии с максимальным размером транзакции интерфейса ведомого устройства AXI, поскольку функции ведомого устройства AXI и DMA используют общую очередь переупорядочивания чтения PCIe, и пропускная способность оптимизирована для обеих функций.

AXI 32 бит – максимальный размер транзакции очереди переупорядочивания чтения PCIe == 64 байт

AXI 64 бит – максимальный размер транзакции очереди переупорядочивания чтения PCIe == 128 байт

AXI 128 бит – максимальный размер транзакции очереди переупорядочивания чтения PCIe == 256 байт

AXI 256 бит – максимальный размер транзакции очереди переупорядочивания чтения PCIe == 512 байт

Задержка чтения при чтении от PCIe может быть очень велика. В результате может потребоваться ожидание большого количества транзакций чтения PCIe одновременно, чтобы скомпенсировать задержку PCIe. Например, для типичной системы PCIe 3.0 (задержка чтения PCIe <= 1uS) исполнению x8 8G требуется 8КБ незавершенных транзакций чтения, чтобы скомпенсировать системную задержку чтения PCIe и поддержать максимальную скорость пересылки данных. 8КБ == 16 512-байтовые или 32 256 -байтовые запросы.

Очередь переупорядочивания PCIe (и, следовательно, количество запросов чтения, одновременно ожидающих завершения) определяется временем доставки, сконфигурированным для 16 или 32 запросов чтения, одновременно ожидающих завершения. Для ядер с >= x8 8G и разрядностью данных AXI == 128 может потребоваться поддержка 32 запросов чтения для компенсации задержки чтения PCIe. Ядра с меньшей разрядностью и меньшей скоростью должны обеспечивать компенсацию системных задержек, поддерживая 16 запросов чтения.

Поскольку запросы записи PCIe не предоставляют отклика, и следовательно данные сопровождения хранить не требуется, количество незавершенных запросов записей PCIe ограничивается только ограничениями на запросы записи различных источников записи. Запросы записи PCIe фрагментируются до меньшего из максимально поддерживаемого размера содержимого записи PCIe и максимального размера содержимого, на которого программа программирует соответствующее ядро PCIe.

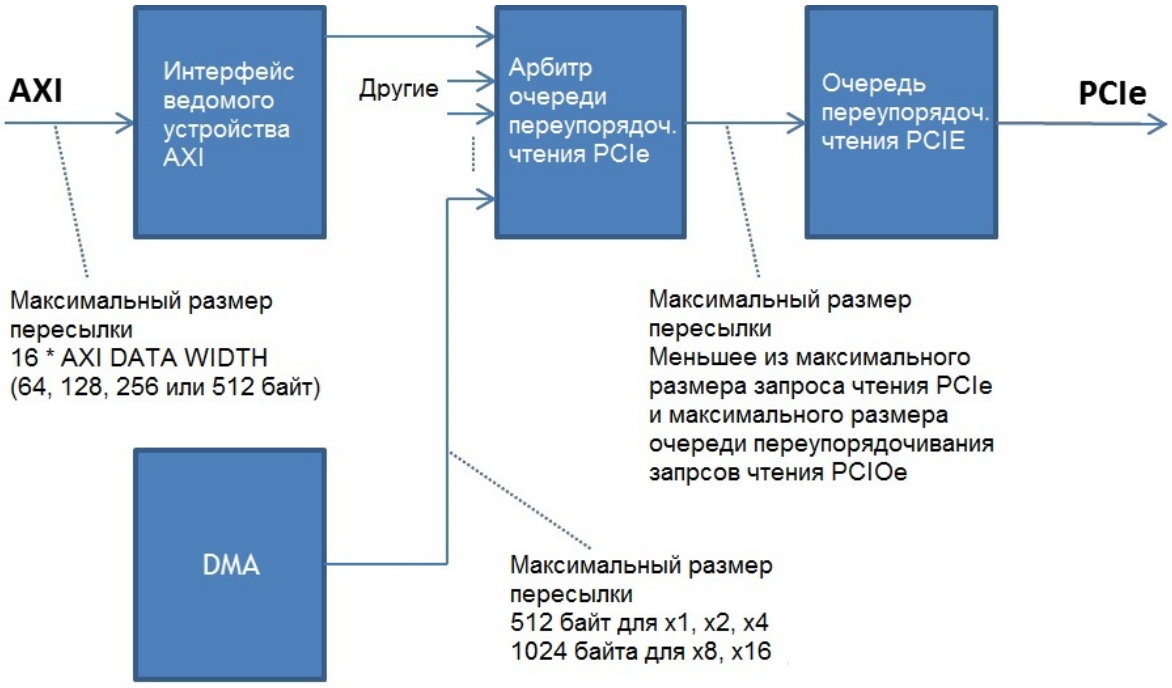


Рисунок 1.145 - Максимальный размер транзакции запроса чтения PCIe Express ядра Expresso DMA Bridge

1.3.10.2.1.16.3 Максимальный размер транзакции данных в канале DMA

Максимальный размер пересылки DMA, которая получается путем объединения элементов SGL источника и приемника DMA (и размер наложения операций с данными канала DMA), зависит от максимальной поддерживаемой ширины канала PCIe:

- Максимальная поддерживаемая ширина канала x1, x2 и x4
  - Максимальный размер пересылки DMA == 512 бит
- Максимальная поддерживаемая ширина канала x8 и x16
  - Максимальный размер пересылки DMA == 1024 бит

Транзакции DMA зачастую больше, чем максимальный размер содержимого ассоциированных интерфейсов, и поэтому транзакции DMA часто разбиваются на несколько транзакций меньшего размера, чтобы соответствовать требованиям к максимальному размеру транзакций интерфейса.

1.3.10.2.1.17 Обработка ошибок моста

Ядро Expresso DMA Bridge предоставляет возможность надежного обнаружения и обработки ошибок.

Общий подход к транзакциям, содержащим ошибки, заключается в том, что транзакция с известной ошибкой сбрасывается без выполнения. Это ограничивает потенциальное распространение воздействия того факта, что транзакция с ошибкой разрешена к обработке. Транзакции с ошибками, которые возвращают статус (все транзакции AXI, внутренние транзакции DMA и транзакции PCIe с обязательным ожиданием ответного пакета) будут возвращать ошибочное завершение, чтобы сообщить об источнике ошибки. Кроме того, ошибки моста записываются в MSGF_MISC: регистра STATUS.

1.3.10.2.1.18 Внутренние регистры моста/DMA

В следующих разделах приводится подробное описание регистров моста и DMA. Приведенные адреса представляют собой смещения относительно PCIe_MASTER и AXI_Master.

- Для ведущего устройства PCIe к смещению прибавляется BAR[cfg_dma_reg_bar], чтобы получить полный адрес PCI Express.
- Для ведущего устройства AXI смещение представляет собой адрес регистра внутри трансляции регистра моста для регистров моста и внутри трансляции регистров DMA для регистров DMA.

1.3.10.2.1.18.1 BRIDGE_CORE

PCIe_MASTER	BRIDGE_CORE_BASE	0x8000
AXI_MASTER	BRIDGE_CORE_BASE	0x0

Регистры конфигурации ядра, которые влияют на основные операции. Эти регистры должны быть статическими во время работы.

1.3.10.2.1.18.1.1 Регистр CFG_PCIE_RX0 0x0

Доступ с PCI Express и конфигурация BAR

Поле	Наименование	Доступ	Разрядност	Сброс
[31:18]	зарезервировано	только для чтения	14	0x0
[17]	cfg_disable_pcie_dma_reg_access	чтение-запись	1	0x0
[16]	cfg_disable_pcie_bridge_reg_access	чтение-запись	1	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2:0]	cfg_dma_reg_bar	чтение-запись	3	0x0

1.3.10.2.1.18.1.1.1 cfg_disable_pcie_dma_reg_access

Определяет, возможен ли доступ к регистрам DMA с PCI Express.

- 0 - Регистры DMA доступны с PCI Express через BAR, конфигурируемый cfg_dma_reg_bar и начинающийся со смещения 0x0.
- 1 - Регистры DMA НЕ доступны с PCI Express.

1.3.10.2.1.18.1.1.2 cfg_disable_pcie_bridge_reg_access

Определяет, возможен ли доступ к регистрам моста с PCI Express.

- 0 - Регистры моста доступны с PCI Express через BAR, конфигурируемый cfg_dma_reg_bar и начинающийся со смещения 0x8000.
- 1 - Регистры моста НЕ доступны с PCI Express.

1.3.10.2.1.18.1.1.3 cfg_dma_reg_bar

Определяет, какая область базовых адресов PCI Express (BAR) используется для доступа с PCI Express к регистрам DMA и моста/ Если информация о BAR принятых запросов чтения/записи PCI Express равна cfg_dma_reg_bar, то транзакция завершается внутренними регистрами DMA/моста; в противном случае транзакция направляется на интерфейс ведущего устройства AXI. cfg_dma_reg_bar предназначен только для полученных запросов чтения и записи PCI Express и не оказывает влияния на принятые запросы интерфейса ведомого устройства AXI.

- 0 - BAR0
- 1 - BAR1
- 2 - BAR2
- 3 - BAR3
- 4 - BAR4
- 5 - BAR5
- 6 - Зарезервировано
- 7 - Запрещено

1.3.10.2.1.18.1.2 Регистр CFG_PCIE_RX1 0x4

Обработка атрибутов принятой транзакции PCI Express.

Поле	Наименование	Доступ	Разрядност	Сброс
[31:9]	зарезервирова	только для чтения	23	0x0
[8]	cfg_rd_ur_is_ur_ok1s_n	чтение-запись	1	0x0
[7:4]	cfg_pcie_rx_arcache	чтение-запись	4	0x0
[3:0]	cfg_pcie_rx_awcache	чтение-запись	4	0x0

1.3.10.2.1.18.1.2.1 cfg_rd_ur_is_ur_ok1s_n

Определяет отклик AXI на инициированные интерфейсом ведомого устройства AXI транзакции чтения/записи конфигурации PCI Express, которые завершаются со статусом неподдерживаемого запроса. Во время нумерации иерархии корневого порта ПО PCI Express обычно должно направлять транзакции чтения и записи конфигурации PCI Express к номерам функций PCI Express, которые могут не существовать, чтобы определить, у каких номеров функций присутствуют устройства. На запросы чтения/записи конфигурации PCI Express к несуществующим функциям вернутся завершения со статусом неподдерживаемых запросов, что считается серьезной ошибкой для других типов транзакций интерфейса ведомого устройства AXI. Поскольку во время нумерации иерархии PCI Express ожидаются ответы в виде неподтвержденных запросов, этот регистр позволяет по выбору уменьшать сереьезноть таких ошибок. Во многих случаях программа конфигурации корневого комплекса PCI Express требует действий, определяемых значением cfg_rd_ur_is_ur_ok1s_n == 0.

- 0 - Возвращать AXI ответ DECERR.
- 1 - Возвращать AXI ответ OKAY с данными чтения AXI из всех 1.

1.3.10.2.1.18.1.2.2 cfg_pcie_rx_arcache

Значение используется портом m_arcache интерфейса ведущего устройства AXI для принятых запросов чтения памяти PCI Express с атрибутом номера петли = 0, которые направлены интерфейсу ведущего устройства AXI для завершения.

Если полученный запрос чтения PCI Express, направляемый на интерфейс ведущего устройства AXI, попадает в адресный регион входной трансляции с разрешенным атрибутом замены кэша, то приоритет имеет значение кэша входной трансляции.

cfg_pcie_rx_awcache

Значение используется портом m_awcache интерфейса ведущего устройства AXI для принятых запросов записи памяти PCI Express с атрибутом номера петли = 0, которые направлены интерфейсу ведущего устройства AXI для завершения.

Если полученный запрос записи PCI Express, направляемый на интерфейс ведущего устройства AXI попадает в адресный регион входной трансляции с разрешенным атрибутом замены кэша, то приоритет имеет значение кэша входной трансляции.

1.3.10.2.1.18.1.3 Регистр CFG_AXI_MASTER 0x8

Конфигурация максимального размера содержимого транзакции для интерфейса ведущего устройства AXI

Поле	Наименование	Доступ	Разрядност	Сброс
[31:7]	зарезервирован	только для чтения	25	0x0
[6:4]	cfg_m_max_rd_req_size	чтение-запись	3	0x0
[3]	зарезервирован	только для чтения	1	0x0
[2:0]	cfg_m_max_wr_req_size	чтение-запись	3	0x0

1.3.10.2.1.18.1.3.1 cfg_m_max_rd_req_size

Максимальный размер запроса чтения, разрешенный для транзакций чтения интерфейса ведущего устройства AXI. Запросы чтения интерфейса ведущего устройства AXI, превышающие установленный максимальный размер запроса чтения, разбиваются на несколько транзакций AXI, чтобы удовлетворить ограничению на максимальный размер запроса чтения.

- 0 - 64 Байта
- 1 - 128 Байт
- 2 - 256 Байт
- 3 - 512 Байт
- 4 - 1024 Байта
- 5 - 2048 Байт
- 6 - 4096 Байт
- 7 - Зарезервировано

1.3.10.2.1.18.1.3.2 cfg_m_max_wr_req_size

Максимальный размер запроса записи, разрешенный для транзакций записи интерфейса ведущего устройства AXI. Запросы записи интерфейса ведущего устройства AXI, превышающие установленный максимальный размер запроса записи, разбиваются на несколько транзакций AXI, чтобы удовлетворить ограничению на максимальный размер запроса записи.

- 0 - 64 Байт
- 1 - 128 Байт
- 2 - 256 Байт
- 3 - 512 Байт
- 4 - 1024 Байт
- 5 - 2048 Байт
- 6 - 4096 Байт
- 7 - Зарезервировано

1.3.10.2.1.18.1.3.3 cfg_m_max_wr_req_size

Максимальный размер запроса записи, разрешенный для транзакций записи интерфейса ведущего устройства AXI. Запросы записи интерфейса ведущего устройства AXI, превышающие установленный максимальный размер запроса записи, разбиваются на несколько транзакций AXI, чтобы удовлетворить ограничению на максимальный размер запроса записи.

- 0 - 64 Байт
- 1 - 128 Байт
- 2 - 256 Байт
- 3 - 512 Байт
- 4 - 1024 Байт
- 5 - 2048 Байт
- 6 - 4096 Байт
- 7 - Зарезервировано

1.3.10.2.1.18.1.4 Регистр CFG_PCIE_TX 0xc

Конфигурация сквозной пересылки PCI Express

Поле	Наименование	Доступ	Разрядност	Сброс
[31:1]	зарезервирован	только для чтения	31	0x0
[0]	cfg_pcie_tx_cut_through	чтение-запись	1	0x0

1.3.10.2.1.18.1.4.1 cfg_pcie_tx_cut_through

Разрешает/запрещает сквозную маршрутизацию для пересылки TLP пакетов PCIe. Разрешение сквозной маршрутизации уменьшает задержку, но может быть разрешено только при условии, что пропускная способность AXI >= пропускной способности PCI Express при всех возможных условиях.

- 0 - Сквозная маршрутизация запрещена (сохранить, затем переправить); пересылка TLP пакета PCIe начинается только после того, как весь пакет TLP был сохранен.
- 1 - Сквозная маршрутизация разрешена; пересылка TLP пакета PCIe начинается сразу же после того, как стало доступно первое слово TLP.

1.3.10.2.1.18.1.5 Регистр CFG_INTERRUPT 0x10

Конфигурация маршрутизации прерываний ядра PCI Express.

Поле	Наименование	Доступ	Разрядност	Сброс
[31:1]	зарезервирова	только для чтения	31	0x0
[0]	cfg_pcie_int_axi_pcie_n	чтение-запись	1	0x0

1.3.10.2.1.18.1.5.1 cfg_pcie_int_axi_pcie_n

Определяет, куда направляются прерывания, генерируемые ядром PCI Express, в PCI Express или AXI. Обычно прерывания ядра PCI Express должны направятся в корневой комплекс, если программное обеспечение локального CPU не предназначено для их обработки.

- 0 - PCIe
- 1 - AXI

1.3.10.2.1.18.1.6 Регистр CFG_RAM_DISABLE0 0x14

Разрешение/запрещение исправления 1-битовых ошибок ECC RAM (только для исполнений с поддержкой ECC)

Поле	Наименование	Доступ	Разрядность	Сброс
[31:15]	зарезервировано	только для чтения	17	0x0
[14]	cfg_ram_dma_sgl_dst_dis_cor	чтение-запись	1	0x0
[13]	cfg_ram_dma_sgl_src_dis_cor	чтение-запись	1	0x0
[12]	cfg_ram_dma_ch_reg_dis_cor	чтение-запись	1	0x0
[11]	cfg_ram_dma_msix_tab_dis_cor	чтение-запись	1	0x0
[10:9]	зарезервировано	только для чтения	2	0x0
[8]	cfg_ram_dma_axi_s_w_dis_cor	чтение-запись	1	0x0
[7]	зарезервировано	только для чтения	1	0x0
[6]	cfg_ram_dma_axi_m_r_dis_cor	чтение-запись	1	0x0
[5]	cfg_ram_dma_pcie_s_cd_dis_cor	чтение-запись	1	0x0
[4]	cfg_ram_dma_pcie_s_ra_dis_cor	чтение-запись	1	0x0
[3]	cfg_ram_dma_pcie_s_w_dis_cor	чтение-запись	1	0x0
[2]	cfg_ram_dma_pcie_s_wa_dis_cor	чтение-запись	1	0x0
[1]	cfg_ram_dma_pcie_tx_w_dis_cor	чтение-запись	1	0x0
[0]	cfg_ram_dma_pcie_m_r_dis_cor	чтение-запись	1	0x0

1.3.10.2.1.18.1.6.1 cfg_ram_dma_sgl_dst_dis_cor

Буфер SGL приемника канала DMA (только DMA).

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.2 cfg_ram_dma_sgl_src_dis_cor

Буфер SGL источника канала DMA (только DMA).

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.3 cfg_ram_dma_ch_reg_dis_cor

Регистры канала DMA (только DMA)

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.4 cfg_ram_dma_msix_tab_dis_cor

Таблица MSI-X

- 0 – Разрешено
- 1 – Запрещено

1.3.10.2.1.18.1.6.5 cfg_ram_dma_axi_s_w_dis_cor

Буфер данных записи ведомого устройства AXI

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.6 cfg_ram_dma_axi_m_r_dis_cor

Очередь переупорядочивания чтения ведущего устройства AXI

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.7 cfg_ram_dma_pcie_s_cd_dis_cor

Буфер данных завершения чтения ведомого устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.8 cfg_ram_dma_pcie_s_ra_dis_cor

Буфер адреса чтения ведомого устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.9 cfg_ram_dma_pcie_s_w_dis_cor

Буфер данных записи ведомого устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.10 cfg_ram_dma_pcie_s_wa_dis_cor

Буфер адреса записи ведомого устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.11 cfg_ram_dma_pcie_tx_w_dis_cor

Буфер TLP ведущего устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.6.12 cfg_ram_dma_pcie_m_r_dis_cor

Очередь переупорядочивания чтения ведущего устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7 Регистр CFG_RAM_DISABLE1 0x18

Разрешение/запрещение обработки 2-битовых ошибок ECC RAM (только для исполнений с поддержкой ECC)

Поле	Наименование	Доступ	Разрядность	Сброс
[31:15]	зарезервировано	только для чтения	17	0x0
[14]	cfg_ram_dma_sgl_dst_dis_err	чтение-запись	1	0x0
[13]	cfg_ram_dma_sgl_src_dis_err	чтение-запись	1	0x0
[12]	cfg_ram_dma_ch_reg_dis_err	чтение-запись	1	0x0
[11]	cfg_ram_dma_msix_tab_dis_err	чтение-запись	1	0x0
[10:9]	зарезервировано	только для чтения	2	0x0
[8]	cfg_ram_dma_axi_s_w_dis_err	чтение-запись	1	0x0
[7]	зарезервировано	только для чтения	1	0x0
[6]	cfg_ram_dma_axi_m_r_dis_err	чтение-запись	1	0x0
[5]	cfg_ram_dma_pcie_s_cd_dis_err	чтение-запись	1	0x0
[4]	cfg_ram_dma_pcie_s_ra_dis_err	чтение-запись	1	0x0
[3]	cfg_ram_dma_pcie_s_w_dis_err	чтение-запись	1	0x0
[2]	cfg_ram_dma_pcie_s_wa_dis_err	чтение-запись	1	0x0
[1]	cfg_ram_dma_pcie_tx_w_dis_err	чтение-запись	1	0x0
[0]	cfg_ram_dma_pcie_m_r_dis_err	чтение-запись	1	0x0

1.3.10.2.1.18.1.7.1 cfg_ram_dma_sgl_dst_dis_err

Буфер SGL приемника канала DMA (только DMA).

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.2 cfg_ram_dma_sgl_src_dis_err

Буфер SGL источника канала DMA (только DMA)

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.3 cfg_ram_dma_ch_reg_dis_err

Регистры канала DMA (только DMA)

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.4 cfg_ram_dma_msix_tab_dis_err

Таблица MSI-X

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.5 cfg_ram_dma_axi_s_w_dis_err

Буфер данных записи ведомого устройства AXI

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.6 cfg_ram_dma_axi_m_r_dis_err

Очередь переупорядочивания чтения ведущего устройства AXI

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.7 cfg_ram_dma_pcie_s_cd_dis_err

Буфер данных завершения чтения ведомого устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.8 cfg_ram_dma_pcie_s_ra_dis_err

Буфер адреса чтения ведомого устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.9 cfg_ram_dma_pcie_s_w_dis_err

Буфер данных записи ведомого устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.10 cfg_ram_dma_pcie_s_wa_dis_err

Буфер адреса записи ведомого устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.11 cfg_ram_dma_pcie_tx_w_dis_err

Буфер TLP ведущего устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.7.12 cfg_ram_dma_pcie_m_r_dis_err

Очередь переупорядочивания чтения ведущего устройства PCIe

- 0 – Разрешено
- 1 - Запрещено

1.3.10.2.1.18.1.8 Регистр CFG_PCIE_RELAXED_ORDER 0x1c

Конфигурация упорядочивания завершения приема PCI Express

Поле	Наименование	Доступ	Разрядност	Сброс
[31:2]	зарезервирова	только для чтения	30	0x0
[1]	cfg_enable_cfgio_wr_ro	чтение-запись	1	0x1
[0]	cfg_enable_dma_ro	чтение-запись	1	0x1

1.3.10.2.1.18.1.8.1 cfg_enable_cfgio_wr_ro

Конфигурация упорядочивания принятых завершения записи и записи ввода/вывода

cfg_enable_cfgio_wr_ro == 1 и == 0 соответствуют правилам упорядочивания PCIe, пока прохождение принятых транзакций записи разрешено, но не требуются для операций записи конфигурации и ввода/вывода.

- 0 - Запрет принятым завершениям PCI Express от запросов записи конфигурации и запросов записи ввода/вывода проходить раньше принятых запросов чтения PCIe.
- 1 - Разрешение принятым завершениям PCI Express от запросов записи конфигурации и запросов записи ввода/вывода проходить раньше принятых запросов чтения PCIe.

1.3.10.2.1.18.1.8.2 cfg_enable_dma_ro

Конфигурация упорядочивания завершения принятых PCI Express запросов чтения DMA

cfg_enable_dma_ro == 1 не соответствует правилам PCIe, однако такая оптимизация обычно допустима для большинства применений и обеспечивает более высокую производительность DMA вследствие устранения ненужной блокировки транзакций.

- 0 - Принудительно установить правила упорядочивания для принятых PCIe завершений от запросов чтения DMA. Принятым PCIe завершениям от запросов чтения DMA будет разрешено проходить раньше принятых запросов записи PCIe только в том случае, если для завершения чтения DMA установлен атрибут ослабленного упорядочивания.
- 1 - Упорядочивать все принятые от DMA завершения PCIe так, как будтор у них установлен атрибут ослабленного упорядочивания. Таким образом, всем завершениям DMA будет позволено проходить до принятых запросов записи DMA

1.3.10.2.1.18.1.9 Регистр CFG_PCIE_RX_MSG_FILTER 0x20

Конфигурация фильтрования принятых сообщений PCI Express

Поле	Наименование	Доступ	Разрядность	Сброс
[31:16]	cfg_desired_ven_msg_ven_id	чтение-запись	16	0x0
[15]	cfg_desired_ven_msg_ven_inv	чтение-запись	1	0x0
[14]	cfg_desired_ven_msg_en	чтение-запись	1	0x0
[13]	cfg_enable_oth_msg_fwd	чтение-запись	1	0x0
[12:8]	зарезерви ровано	только для чтения	5	0x0
[7]	cfg_enable_ven_msg_fwd	чтение-запись	1	0x0
[6]	зарезерви ровано	только для чтения	1	0x0
[5]	cfg_enable_slt_msg_fwd	чтение-запись	1	0x0
[4]	зарезерви ровано	только для чтения	1	0x0
[3]	cfg_enable_err_msg_fwd	чтение-запись	1	0x0
[2]	cfg_enable_int_msg_fwd	чтение-запись	1	0x0
[1]	cfg_enable_pm_msg_fwd	чтение-запись	1	0x0
[0]	зарезерви ровано	только для чтения	1	0x0

1.3.10.2.1.18.1.9.1 cfg_desired_ven_msg_ven_id

Определяемый производителем ID типа сообщения 0/1 (код сообщения 0x7e & 0x7f) используется для специальной фильтрации сообщений, зависящей от производителя

1.3.10.2.1.18.1.9.2 cfg_desired_ven_msg_ven_inv

Выбирает, как применять фильтрацию, зависящую от производителя.

1.3.10.2.1.18.1.9.3 cfg_desired_ven_msg_en

- 0 - Запретить
- 1 - Разрешить зависящую от производителя фильтрацию для сообщений, определяемых производителем.

1.3.10.2.1.18.1.9.4 cfg_enable_oth_msg_fwd

При 1 полученные PCIe сообщения с msg_code[7:4], не совпадающим с x1, 0x2, 0x3, 0x5 или 0x7 (другие типы сообщений), будут направлены на интерфейс ведущего устройства AXI. При 0 такие сообщения будут сбрасываться и не передаваться дальше.

1.3.10.2.1.18.1.9.5 cfg_enable_ven_msg_fwd

При 1 полученные PCIe сообщения с msg_code[7:4], не совпадающим с x1, 0x2, 0x3, 0x5 или 0x7 (другие типы сообщений), будут направлены на интерфейс ведущего устройства AXI. При 0 такие сообщения будут сбрасываться и не передаваться дальше.

1.3.10.2.1.18.1.9.6 cfg_enable_slt_msg_fwd

При 1 полученные PCIe сообщения с msg_code[7:4] == 0x5 (такие как Set Slot Power Limit), будут направлены на интерфейс ведущего устройства AXI. При 0 такие сообщения будут сбрасываться и не передаваться дальше.

1.3.10.2.1.18.1.9.7 cfg_enable_err_msg_fwd

При 1 полученные PCIe сообщения с msg_code[7:4] == 0x3 (такие, как ERR_COR, ERR_NONFATAL, ERR_FATAL), будут направлены на интерфейс ведущего устройства AXI. При 0 такие сообщения будут сбрасываться и не передаваться дальше.

1.3.10.2.1.18.1.9.8 cfg_enable_int_msg_fwd

При 1 полученные PCIe сообщения с msg_code[7:4] == 0x2 (Assert_INT[A, B,C,D], Deassert_INT[A,B,C,D]), будут направлены на интерфейс ведущего устройства AXI. При 0 такие сообщения будут сбрасываться и не передаваться дальше.

1.3.10.2.1.18.1.9.9 cfg_enable_pm_msg_fwd

При 1 полученные PCIe сообщения с msg_code[7:4] == 0x1 (сообщения управления питанием, такие, как PME_Turn_Off, PME_TO_ACK), будут направлены на интерфейс ведущего устройства AXI. При 0 такие сообщения будут сбрасываться и не передаваться дальше.

1.3.10.2.1.18.1.10 Регистр CFG_RQ_REQ_ORDER 0x24

Конфигурация упорядочивания завершения очереди переупорядочивания чтения PCI Express и AXI

Поле	Наименование	Доступ	Разрядность	Сброс
[31]	cfg_pcie_req_order_strict	чтение-запись	1	0x0
[30]	cfg_axi_req_order_strict	чтение-запись	1	0x0
[29:16]	зарезерв ировано	только для чтения	14	0x0
[15:0]	cfg_axi_req_order_id_mask	чтение-запись	16	0x0

1.3.10.2.1.18.1.10.1 cfg_pcie_req_order_strict

Для запросов чтения ведущего устройства AXI. Оба значения, 0 и 1, удовлетворяют требованиям упорядочивания PCIe и AXI. Значение 1 увеличит задержку завершения чтения, когда завершения чтения поступают неупорядоченно. По умолчанию рекомендуется значение 0, которое обеспечит максимальную производительность.

- 0 - Разрешить завершениям запросов чтения ведущего устройства AXI от разных источников (PCIe Rx, DMA, и т. д.) возвращаться в порядке, отличном, от изначального порядка запросов ведущего устройства AXI. Кроме того, для принятых запросов чтения PCIe с разными ID инициаторов и тегами завершениям чтения PCIe разрешается возвращаться в порядке, отличном, от изначального порядка запросов.
- 1 - Направлять завершения запросов чтения к источникам чтения в том же порядке, в котором запросы чтения были выставлены на интерфейсе ведущего устройства AXI. Если на шине AXI запросы чтения выполняются в другом порядке, завершения запросов чтения будут приведены в оригинальный порядок, прежде чем их передадут к источнику запроса чтения.

1.3.10.2.1.18.1.10.2 cfg_axi_req_order_strict

Для запросов чтения (переданных) ведущего устройства PCIe. Оба значения, 0 и 1, удовлетворяют требованиям упорядочивания PCIe и AXI. Значение 1 увеличит задержку завершения чтения, когда завершения чтения поступают неупорядоченно. По умолчанию рекомендуется значение 0, которое обеспечит максимальную производительность.

- 0 - Разрешить завершениям запросов чтения PCIe от разных источников (ведомое устройство AXI, DMA, и т. д.) возвращаться в порядке, отличном, от изначального порядка запросов PCIe. Кроме того, для запросов чтения интерфейса ведомого устройства AXI у каждного интрефейса ведомого устройства со своим ID имеется cfg_axi_req_order_id_mask для управления неупорядоченными завершениями чтения.
- 1 - Направлять завершения запросов чтения к источникам чтения в том же порядке, в котором запросы чтения были выставлены на интерфейсе PCIe. Если на шине PCIe запросы чтения выполняются в другом порядке, завершения запросов чтения будут приведены в оригинальный порядок, прежде чем их передадут к источнику запроса чтения.

1.3.10.2.1.18.1.10.3 cfg_axi_req_order_id_mask

Определяет, какие биты s_arid[SID-1:0] интерфейса ведомого устройства учитываются, когда fg_axi_rq_req_order_strict = 0.

*cfg_axi_req_order_id_mask[i] == 1. Учитывать s_arid[i] при определении разрешенного порядка завершений чтения на запросы чтения ведомого устройства AXI

*cfg_axi_req_order_id_mask[i] == 0. Не учитывать s_arid[i] при определении разрешенного порядка завершений чтения AXI

1.3.10.2.1.18.1.11 Регистр CFG_PCIE_CREDIT 0x28

Конфигурация заголовка завершения пересылки и Data Credit Metering PCI Express.

Поле	Наименование	Доступ	Разрядность	Сброс
[31]	cfg_pcie_credit_en	чтение-запись	1	0x0
[30]	cfg_pcie_credit_ch_inf	чтение-запись	1	0x0
[29]	cfg_pcie_credit_cd_inf	чтение-запись	1	0x0
[28:24]	зарезервировано	только для чтения	5	0x0
[23:16]	cfg_pcie_credit_ch_val	чтение-запись	8	0x20
[15:12]	зарезервировано	только для чтения	4	0x0
[11:0]	cfg_pcie_credit_cd_val	чтение-запись	12	0x200

1.3.10.2.1.18.1.11.1 cfg_pcie_credit_en

Незавершенные запросы чтения PCIe ограничены областью приемного буфера (CH и CD разрешения на пересылку), выделенной в приемном буфере ядра PCI Express.

- 0 - Взять для этой цели границы CH и CD разрешений из ядра PCIe.
- 1 - Взять для этой цели границы CH и CD разрешений из регистра CFG_PCIE_CREDIT. Конфигурирование моста с большим количеством CH и CD разрешений, чем уже присутствует в ядре PCIe, может значительно повысить производительность чтения из PCIe для ядер PCIe с малым числом разрешений, однако в данном случае открывается возможность серьезных ошибок переполнения завершения, и это следует делать только при полной уверенности, что в применении пользователя такого переполнения не случится.

1.3.10.2.1.18.1.11.2 cfg_pcie_credit_ch_inf

Неограниченное число разрешений CH

- 0 - Конечное число разрешений CH, равное cfg_pcie_credit_ch_val.
- 1 - Неограниченное число разрешений CH. Не проверяются разрешения CH.

1.3.10.2.1.18.1.11.3 cfg_pcie_credit_cd_inf

Неограниченное число разрешений CD

- 0 - Конечное число разрешений CD, равное cfg_pcie_credit_cd_val.
- 1 - Неограниченное число разрешений CD. Не проверяются разрешения CD.

1.3.10.2.1.18.1.11.4 cfg_pcie_credit_ch_val

Количество доступных разрешений CH. Одно разрешение CH == пространство для 1 заголовка пакета TLP завершения.

1.3.10.2.1.18.1.11.5 cfg_pcie_credit_cd_val

Количество доступных разрешений CD. Одно разрешение CD == пространство для 16 бит содержимого пакета TLP завершения.

1.3.10.2.1.18.1.12 Регистр CFG_AXI_M_W_TICK_COUNT 0x2c

Конфигурация тайм-аута завершения записи ведущего устройства AXI

Поле	Наименование	Доступ	Разрядность	Сброс
[31:16]	зарезервировано	только для чтения	16	0x0
[15:0]	axi_m_w_tick_count	чтение-запись	16	0xbea

1.3.10.2.1.18.1.12.1 axi_m_w_tick_count

Определяет период тайм-аута запроса записи ведущего устройства AXI. Тайм-аут произойдет после 14 или 15 w_ticks, где w_ticks имеет шаг (256*axi_m_w_tick_count) периодов тактового сигнала AXI. Значение 0 запрещает механизм тайм-аута. Например, axi_m_w_tick_count==7 приводит к тайм-ауту в диапазоне от 50,176uS до 53,760uS при тактовой частоте AXI 500 MHz.

1.3.10.2.1.18.1.13 Регистр CFG_AXI_M_R_TICK_COUNT 0x30

Конфигурация тайм-аута завершения чтения ведущего устройства AXI

Поле	Наименование	Доступ	Разрядност	Сброс
[31:16]	зарезервировано	только для чтения	16	0x0
[15:0]	axi_m_r_tick_count	чтение-запись	16	0xbea

1.3.10.2.1.18.1.13.1 axi_m_r_tick_count

Определяет период тайм-аута запроса чтения ведущего устройства AXI. Тайм-аут произойдет после 14 или 15 r_ticks, где r_ticks имеет шаг (256*axi_m_r_tick_count) периодов тактового сигнала AXI. Значение 0 запрещает механизм тайм-аута. Например, axi_m_r_tick_count==7000 приводит к тайм-ауту в диапазоне от 50,176mS до 53,760mS при тактовой частоте AXI 500 MHz.

1.3.10.2.1.18.1.14 Регистр CFG_CRS_RPL_TICK_COUNT 0x34

Конфигурация тайм-аута повтора CRS чтения/записи конфигурации PCIe

Поле	Наименование	Доступ	Разрядност	Сброс
[31:16]	зарезервировано	только для чтения	16	0x0
[15:0]	crs_rpl_tick_count	чтение-запись	16	0x2210

1.3.10.2.1.18.1.14.1 crs_rpl_tick_count

Определяет период тайм-аута повтора CRS. Интерфейс ведомого устройства AXI инициирует запросы записи и чтения конфигурации PCI Express (ECAM), которые возвращают статус повтора запроса конфигурации (CRS) и которые не обрабатываются с помощью CRS Software Visbility, будут повторяться до истечения периода тайм-аута. После истечения времени тайм-аута следующий ответ CRS, полученный для данной транзакции, преобразуется в ответ Slave Error (ошибка ведомого устройства), и повторные запросы больше не отправляются. Тайм-аут произойдет после 14 или 15 l_ticks, где l_ticks имеет шаг (4096*crs_rpl_tick_count) периодов тактового сигнала AXI. Значение 0 запрещает механизм тайм-аута. Например, crs_rpl_tick_count==8720 приводит к тайм-ауту в диапазоне от 1,000S до 1,017S при тактовой частоте AXI 500 MHz.

1.3.10.2.1.18.2 E_BREG

PCIe_MASTER	E_BREG_BASE	0x8200
AXI_MASTER	E_BREG_BASE	0x200

Выходная трансляция регистра моста.

Данная трансляция используется для чтения/записи внутренних регистров моста, в число которых входят эти регистры трансляции, с интерфейса ведомого устройства AXI. В результате эта трансляция разрешена по умолчанию при сбросе и по умолчанию попадает во все адресные регионы доступа интерфейса ведомого устройства AXI. Одно из первых действий CPU для AXI - начальная установка регистров моста, в том числе помещение выходной трансляции регистра моста в конкретную область памяти, а затем запрещение выходной трансляции регистра моста полностью захватывать все транзакции интерфейса ведомого устройства AXI, чтобы обеспечить доступ к другим ресурсам.

1.3.10.2.1.18.2.1 Регистр CAPABILITIES 0x0

Выходная трансляция регистра моста – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	breg_size_max	только для чтения	8	0x3
[23:16]	breg_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	breg_present	только для чтения	1	0x1

1.3.10.2.1.18.2.1.1 breg_size_max

breg_size поддерживает значения между 0 и breg_size_max. Максимальный размер трансляции - 2

1.3.10.2.1.18.2.1.2 breg_size_offset

Минимальный размер трансляции - 2breg_size_offset.

1.3.10.2.1.18.2.1.3 breg_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.2.2 Регистр STATUS 0x4

Выходная трансляция регистра моста – статус

Поле	Наименование	Доступ	Разрядност	Сброс
[31:25]	зарезервировано	только для	7	0x0
[24:16]	wr_pending_ctr	только для	9	0x0
[15:9]	зарезервировано	только для	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.2.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.2.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.2.3 Регистр CONTROL 0x8

Выходная трансляция регистра моста – управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:18]	зарезервировано	только для чтения	14	0x0
[17:16]	breg_size	чтение-запись	2	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2]	breg_security_enable	чтение-запись	1	0x0
[1]	breg_enable_force	чтение-запись	1	0x1
[0]	breg_enable	чтение-запись	1	0x0

1.3.10.2.1.18.2.3.1 breg_size

Размер этого окна трансляции выражается как 2breg_size_offset+breg_size. Ограничена полями breg_size_max и breg_size_offset

1.3.10.2.1.18.2.3.2 breg_security_enable

Разрешение/запрещение защиты трансляции.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.2.3.3 breg_enable_force

Принудить все транзакции интерфейса ведомого устройства AXI быть затребованными входной трансляцией регистра моста

- 0 - Запрещено
- 1 – Разрешено. Разрешено по умолчанию, так что после сброса AXI CPU получает доступ к регистрам моста, чтобы выполнить начальную установку этой трансляции и других регистров моста. После завершения начальной конфигурации регистров моста в breg_enable_force должен быть записан 0, чтобы разрешить действие других трансляций.

1.3.10.2.1.18.2.3.4 breg_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*breg_enable == 1

*breg_src_base[63:(12+breg_size)] == AXI Address[63:(12+breg_size)]

1.3.10.2.1.18.2.4 Регистр BASE_LO 0x10

Выходная трансляция регистра моста - младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	breg_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.2.4.1 breg_base_lo

breg_src_base[31:12]

1.3.10.2.1.18.2.5 Регистр BASE_HI 0x14

Выходная трансляция регистра моста - старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	breg_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.2.5.1 breg_base_hi

breg_src_base[63:32]

1.3.10.2.1.18.3 E_ECAM

PCIe_MASTER	E_ECAM_BASE	0x8220
AXI_MASTER	E_ECAM_BASE	0x220

Выходная трансляция механизма доступа к расширенной конфигурации (ECAM) PCI Express.

Эта трансляция разрешает чтение/запись регистров конфигурации PCI Express из интерфейса ведомого устройства

Стандарт AXI не обеспечивает средств поддержки для генерации транзакций конфигурации PCI Express, и поэтому мост реализует механизм доступа к расширенной конфигурации (ECAM) через эту трансляцию, чтобы транслировать запросы чтения и записи интерфейса ведомого устройства AXI в транзакции чтения и записи конфигурации PCI Express.

Применение в качестве корневого порта (нисходящий порт PCI Express) должно генерировать транзакции чтения и записи конфигурации PCI Express, чтобы конфигурировать и отслеживать функции PCI Express в иерархии PCI Express.

Применение в качестве оконечной точки (восходящий порт PCI Express) не требует, и ему не разрешено, генерировать транзакции чтения и записи конфигурации PCI Express к другим функциям PCI Express, но эта трансляция по-прежнему может использоваться для доступа к регистрам конфигурации локального ядра PCI Express. Однако при работе в качестве оконечной точки начальная установка и управление регистрами конфигурации ядра локальной PCI Express выполняются программой ведущего устройства и обычно не изменяются с помощью этого механизма.

Механизм ECAM PCI Express выделяет часть адресного пространства памяти AXI CPU под транзакции конфигурации PCI Express. Транзакции чтения/записи ведомого интерфейса AXI, целью которых являются эта транзакция, преобразуются в транзакцию чтения/записи конфигурации PCI Express при помощи следующей трансляции адресов:

A[27:20] Номер шины

A[19:15] Номер устройства

A[14:12] Номер функции

A[11:8] Номер расширенного регистра

A[7:2] Номер регистра

A[1:0] Байты внутри каждого 32-битового регистра

ECAM транзакции PCI Express поддерживают максимум 32-битный размер транзакций, и им запрещено превышать границу 32-битного адреса. Максимальный 32-битный размер AXI разрешен при A[1:0]==00.

Если трансляция ECAM сконфигурирована меньшей, чем 2^28 байт, то старшие биты адреса, не входящие в окно трансляции, замещаются 0 при осуществлении транзакции конфигурации PCI Express.

Например, если для конкретного применения корневого порта требуются только 16 номеров шин 16 PCI, то для доступа к необходимым 16 номерам шин PCI Express (номера шин 0-15) достаточно размера окна трансляции только 2^24 .

1.3.10.2.1.18.3.1 Регистр CAPABILITIES 0x0

Выходная трансляция ECAM – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	ecam_size_max	только для чтения	8	0x10
[23:16]	ecam_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	ecam_present	только для чтения	1	0x1

1.3.10.2.1.18.3.1.1 ecam_size_max

ecam_size поддерживает значения между 0 и ecam_size_max. Максимальный размер трансляции - 2ecam_size_offset+ecam_size_max.

1.3.10.2.1.18.3.1.2 ecam_size_offset

Минимальный размер трансляции - 2ecam_size_offset.

1.3.10.2.1.18.3.1.3 ecam_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.3.2 Регистр STATUS 0x4

Выходная трансляция ECAM – статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.3.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.3.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.3.3 Регистр CONTROL 0x8

Выходная трансляция ECAM - управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:21]	зарезервировано	только для чтения	11	0x0
[20:16]	ecam_size	чтение-запись	5	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2]	ecam_security_enable	чтение-запись	1	0x0
[1]	зарезервировано	только для чтения	1	0x0
[0]	ecam_enable	чтение-запись	1	0x0

1.3.10.2.1.18.3.3.1 ecam_size

Размер этого окна трансляции выражается как 2ecam_size_offset+ecam_size. Ограничена полями ecam_size_max и ecam_size_offset

1.3.10.2.1.18.3.3.2 ecam_security_enable

Разрешение/запрещение защиты трансляции.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.3.3.3 ecam_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*ecam_enable == 1

*ecam_src_base[63:(12+ecam_size)] == AXI Address[63:(12+ecam_size)]

1.3.10.2.1.18.3.4 Регистр BASE_LO 0x10

Выходная трансляция ECAM - младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	ecam_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.3.4.1 ecam_base_lo

ecam_src_base[31:12].

1.3.10.2.1.18.3.5 Регистр BASE_HI 0x14

Выходная трансляция ECAM - старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	ecam_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.3.5.1 ecam_base_hi

ecam_src_base[63:32].

1.3.10.2.1.18.4 E_MSXT

PCIe_MASTER	E_MSXT_BASE	0x8240
AXI_MASTER	E_MSXT_BASE	0x240

Выходная трансляция таблицы MSI-X.

Эта трансляция позволяет внутренней таблице MSI-X, реализованной мостом, считываться/записываться из интерфейса ведомого устройства AXI.

Таблица MSI-X должна записываться только BIOS/операционной системой корневого порта как составная часть процесса реализации функциональности прерываний MSI-X в PCI Express. Когда мост работает в качестве оконечной точки, таблица MSI-X должна записываться только программой ведущего устройства PCIe. Когда мост работает в качестве корневого порта, таблица MSI-X должна записываться только программой ведущего устройства AXI. Таблица MSI-X используется только в том случае, если прерывания MSI-X разрешены в возможностях MSI-X подключенного к мосту ядра PCI Express.

1.3.10.2.1.18.4.1 Регистр CAPABILITIES 0x0

Выходная трансляция таблицы MSI-X - возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	msxt_size_max	только для чтения	8	0x3
[23:16]	msxt_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	msxt_present	только для чтения	1	0x1

1.3.10.2.1.18.4.1.1 msxt_size_max

msxt_size поддерживает значения между 0 и msxt_size_max. Максимальный размер трансляции - 2msxt_size_offset+msxt_size_max.

1.3.10.2.1.18.4.1.2 msxt_size_offset

Минимальный размер трансляции - 2msxt_size_offset.

1.3.10.2.1.18.4.1.3 msxt_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.4.2 Регистр STATUS 0x4

Выходная трансляция таблицы MSI-X – статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.4.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.4.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.4.3 Регистр CONTROL 0x8

Выходная трансляция таблицы MSI-X – управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:18]	зарезервировано	только для чтения	14	0x0
[17:16]	msxt_size	чтение-запись	2	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2]	msxt_security_enable	чтение-запись	1	0x0
[1]	зарезервировано	только для чтения	1	0x0
[0]	msxt_enable	чтение-запись	1	0x0

1.3.10.2.1.18.4.3.1 msxt_size

Размер этого окна трансляции выражается как 2msxt_size_offset+msxt_size. Ограничена полями msxt_size_max и msxt_size_offset

1.3.10.2.1.18.4.3.2 msxt_security_enable

Разрешение/запрещение защиты трансляции.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.4.3.3 msxt_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*msxt_enable == 1

*msxt_src_base[63:(12+msxt_size)] == AXI Address[63:(12+msxt_size)]

1.3.10.2.1.18.4.4 Регистр BASE_LO 0x10

Выходная трансляция таблицы MSI-X - младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	msxt_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.4.4.1 msxt_base_lo

msxt_src_base[31:12]

1.3.10.2.1.18.4.5 Регистр BASE_HI 0x14

Выходная трансляция таблицы MSI-X - старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	msxt_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.4.5.1 msxt_base_hi

msxt_src_base[63:32].

1.3.10.2.1.18.5 E_MSXP

PCIe_MASTER	E_MSXP_BASE	0x8260
AXI_MASTER	E_MSXP_BASE	0x260

Выходная трансляция массива бит статуса запросов прерываний (PBA) типа MSI-X.

Эта трансляция позволяет внутреннему массиву бит статуса запросов прерываний типа MSI-X, реализованному мостом, считываться/записываться из интерфейса ведомого устройства AXI.

MSI-X PBA должна записываться только BIOS/операционной системой корневого порта как составная часть процесса реализации функциональности прерываний MSI-X в PCI Express. Когда мост работает в качестве оконечной точки, MSI-X PBA должна записываться только программой ведущего устройства PCIe. Когда мост работает в качестве корневого порта, MSI-X PBA должна записываться только программой ведущего устройства AXI. MSI-X PBA используется только в том случае, если прерывания MSI-X разрешены в возможностях MSI-X подключенного к мосту ядра PCI Express.

1.3.10.2.1.18.5.1 Регистр CAPABILITIES 0x0

Выходная трансляция MSI-X PBA – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	msxp_size_max	только для чтения	8	0x3
[23:16]	msxp_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	msxp_present	только для чтения	1	0x1

1.3.10.2.1.18.5.1.1 msxp_size_max

msxp_size поддерживает значения между 0 и msxp_size_max. Максимальный размер трансляции - 2msxp_size_offset+msxp_size_max.

1.3.10.2.1.18.5.1.2 msxp_size_offset

Минимальный размер трансляции - 2msxp_size_offset.

1.3.10.2.1.18.5.1.3 msxp_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.5.2 Регистр STATUS 0x4

Выходная трансляция MSI-X PBA – статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.5.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.5.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.5.3 Регистр CONTROL 0x8

Выходная трансляция таблицы MSI-X - управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:18]	зарезервировано	только для чтения	14	0x0
[17:16]	msxt_size	чтение-запись	2	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2]	msxt_security_enable	чтение-запись	1	0x0
[1]	зарезервировано	только для чтения	1	0x0
[0]	msxt_enable	чтение-запись	1	0x0

1.3.10.2.1.18.5.3.1 msxt_size

Размер этого окна трансляции выражается как 2msxt_size_offset+msxt_size. Ограничена полями msxt_size_max и msxt_size_offset

1.3.10.2.1.18.5.3.2 msxt_security_enable

Разрешение/запрещение защиты трансляции.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.5.3.3 msxt_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*msxt_enable == 1

*msxt_src_base[63:(12+msxt_size)] == AXI Address[63:(12+msxt_size)]

1.3.10.2.1.18.5.4 Регистр BASE_LO 0x10

Выходная трансляция таблицы MSI-X - младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	msxt_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.5.4.1 msxt_base_lo

msxt_src_base[31:12]

1.3.10.2.1.18.5.5 Регистр BASE_HI 0x14

Выходная трансляция таблицы MSI-X - старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	msxt_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.5.5.1 msxt_base_hi

msxt_src_base[63:32].

1.3.10.2.1.18.6 E_MSXP

PCIe_MASTER	E_MSXP_BASE	0x8260
AXI_MASTER	E_MSXP_BASE	0x260

Выходная трансляция массива бит статуса запросов прерываний (PBA) типа MSI-X.

Эта трансляция позволяет внутреннему массиву бит статуса запросов прерываний типа MSI-X, реализованному мостом, считываться/записываться из интерфейса ведомого устройства AXI.

MSI-X PBA должна записываться только BIOS/операционной системой корневого порта как составная часть процесса реализации функциональности прерываний MSI-X в PCI Express. Когда мост работает в качестве оконечной точки, MSI-X PBA должна записываться только программой ведущего устройства PCIe. Когда мост работает в качестве корневого порта, MSI-X PBA должна записываться только программой ведущего устройства AXI. MSI-X PBA используется только в том случае, если прерывания MSI-X разрешены в возможностях MSI-X подключенного к мосту ядра PCI Express.

1.3.10.2.1.18.6.1 Регистр CAPABILITIES 0x0

Выходная трансляция MSI-X PBA – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	msxp_size_max	только для чтения	8	0x3
[23:16]	msxp_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	msxp_present	только для чтения	1	0x1

1.3.10.2.1.18.6.1.1 msxp_size_max

msxp_size поддерживает значения между 0 и msxp_size_max. Максимальный размер трансляции - 2msxp_size_offset+msxp_size_max.

1.3.10.2.1.18.6.1.2 msxp_size_offset

Минимальный размер трансляции - 2msxp_size_offset.

1.3.10.2.1.18.6.1.3 msxp_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.6.2 Регистр STATUS 0x4

Выходная трансляция MSI-X PBA - статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.6.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.6.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.6.3 Регистр CONTROL 0x8

Выходная трансляция MSI-X PBA – управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:18]	зарезервировано	только для чтения	14	0x0
[17:16]	msxp_size	чтение-запись	2	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2]	msxp_security_enable	чтение-запись	1	0x0
[1]	зарезервировано	только для чтения	1	0x0
[0]	msxp_enable	чтение-запись	1	0x0

1.3.10.2.1.18.6.3.1 msxp_size

Размер этого окна трансляции выражается как 2msxp_size_offset+msxp_size. Ограничена полями msxp_size_max и msxp_size_offset

1.3.10.2.1.18.6.3.2 msxp_security_enable

Разрешение/запрещение защиты трансляции.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.6.3.3 msxp_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*msxp_enable == 1

*msxp_src_base[63:(12+msxp_size)] == AXI Address[63:(12+msxp_size)]

1.3.10.2.1.18.6.4 Регистр BASE_LO 0x10

Выходная трансляция MSI-XPBA - младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	msxp_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.6.4.1 msxp_base_lo

msxp_src_base[31:12].

1.3.10.2.1.18.6.5 Регистр BASE_HI 0x14

Выходная трансляция MSI-XPBA - старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	msxp_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.6.5.1 msxp_base_hi

msxp_src_base[63:32].

1.3.10.2.1.18.7 E_DREG

PCIe_MASTER	E_DREG_BASE	0x8280
AXI_MASTER	E_DREG_BASE	0x280

Выходная трансляция регистра DMA.

Эта трансляция позволяет внутренним регистрам DMA, реализованным мостом, считываться/записываться из интерфейса ведомого устройства AXI. Эти регистры не присутствуют, и из них считывается 0x0, когда ядро Expresso DMA конфигурируется с поддержкой DMA.

1.3.10.2.1.18.7.1 Регистр CAPABILITIES 0x0

Выходная трансляция регистра DMA – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	dma_size_max	только для чтения	8	0x3
[23:16]	dma_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	dma_present	только для чтения	1	0x1

1.3.10.2.1.18.7.1.1 dma_size_max

dma_size поддерживает значения между 0 и dma_size_max. Максимальный размер трансляции - 2dma_size_offset+dma_size_max.

1.3.10.2.1.18.7.1.2 dma_size_offset

Минимальный размер трансляции - 2dma_size_offset.

1.3.10.2.1.18.7.1.3 dma_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.7.2 Регистр STATUS 0x4

Выходная трансляция регистра DMA – статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.7.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.7.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.7.3 Регистр CONTROL 0x8

Выходная трансляция регистра DMA – управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:18]	зарезервировано	только для чтения	14	0x0
[17:16]	dma_size	чтение-запись	2	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2]	dma_security_enable	чтение-запись	1	0x0
[1]	зарезервировано	только для чтения	1	0x0
[0]	dma_enable	чтение-запись	1	0x0

1.3.10.2.1.18.7.3.1 dma_size

Размер этого окна трансляции выражается как 2dma_size_offset+dma_size. Ограничена полями dma_size_max и dma_size_offset

1.3.10.2.1.18.7.3.2 dma_security_enable

Разрешение/запрещение защиты трансляции.

- 0 - Запрещено
- 1 - Разрешено

1.3.10.2.1.18.7.3.3 dma_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*dma_enable == 1

*dma_src_base[63:(12+dma_size)] == AXI Address[63:(12+dma_size)]

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.7.4 Регистр BASE_LO 0x10

Выходная трансляция регистра DMA- младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	dma_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.7.4.1 dma_base_lo

dma_src_base[31:12].

1.3.10.2.1.18.7.5 Регистр BASE_HI 0x14

Выходная трансляция регистра DMA- старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	dma_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.7.5.1 dma_base_hi

dma_src_base[63:32].

1.3.10.2.1.18.8 E_ESUB

PCIe_MASTER	E_ESUB_BASE	0x82e0
AXI_MASTER	E_ESUB_BASE	0x2e0

Выходная трансляция субтрактивного декодирования.

1.3.10.2.1.18.8.1 Запросы чтения и записи интерфейса ведомого устройства AXI, которые не попадают во все остальные адресные регионы выходных трансляций, обрабатываются как ошибки или передаются в PCI Express, в зависимости от конфигурации этой трансляции. Регистр CAPABILITIES 0x0

Выходная трансляция субтрактивного декодирования – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:1]	зарезервировано	только для чтения	31	0x0
[0]	subtractive_decode_present	только для чтения	1	0x1

1.3.10.2.1.18.8.1.1 subtractive_decode_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.8.2 Регистр STATUS 0x4

Выходная трансляция субтрактивного декодирования – статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.8.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.8.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.8.3 Регистр CONTROL 0x8

Выходная трансляция субтрактивного декодирования - управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:1]	зарезервировано	только для чтения	31	0x0
[0]	egress_sub_enable	чтение-запись	1	0x1

1.3.10.2.1.18.8.3.1 egress_sub_enable

Разрешение субтрактивного декодирования.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.9 I_MSII

PCIe_MASTER	I_MSII_BASE	0x8300
AXI_MASTER	I_MSII_BASE	0x300

Входная трансляция прерывания MSI, принятого PCI Express.

Определяет область адресов, в которой принятые запросы записи в память PCI Express считаются прерываниями MSI.

Запись в память, попадающая в этот адресный регион трансляции, передается контроллеру принятых прерываний для обработки в качестве прерываний MSI.

1.3.10.2.1.18.9.1 Регистр CAPABILITIES 0x0

Входная трансляция прерывания MSI, принятого PCI Express – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	i_msii_size_max	только для чтения	8	0x1f
[23:16]	i_msii_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	i_msii_present	только для чтения	1	0x1

1.3.10.2.1.18.9.1.1 _msii_size

Размер этого окна трансляции выражается как 2msii_size_offset+msii_size. Ограничена полями i_msii_size_max и i_msii_size_offset

1.3.10.2.1.18.9.1.2 i_msii_status_enable

0 - Прерывания MSI записываются в IFO принятых сообщений.
1 - Прерывания MSI записываются в регистры msgf_msi_status.

1.3.10.2.1.18.9.1.3 i_msii_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*i_msii_enable == 1

*i_msii_src_base[63:(12+i_msii_size)] == AXI Address[63:(12+i_msii_size)]

1.3.10.2.1.18.9.2 Регистр BASE_LO 0x10

Входная трансляция прерывания MSI, принятого PCI Express - младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	i_msii_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.9.2.1 i_msii_base_lo

i_msii_src_base[31:12]

1.3.10.2.1.18.9.3 Регистр BASE_HI 0x14

Входная трансляция прерывания MSI, принятого PCI Express - старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	i_msii_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.9.3.1 i_msii_base_hi

i_msii_src_base[63:32]

1.3.10.2.1.18.10 I_MSIX

PCIe_MASTER	I_MSIX_BASE	0x8320
AXI_MASTER	I_MSIX_BASE	0x320

Входная трансляция прерывания MSI-X, принятого PCI Express.

Определяет область адресов, в которой запросы чтения памяти PCI Express считаются прерываниями MSI-X.

Запись в память, попадающая в адресный регион этой трансляции, передается контроллеру принятых прерываний для обработки в качестве прерываний MSI-X.

1.3.10.2.1.18.10.1 Регистр CAPABILITIES 0x0

Входная трансляция прерывания MSI-X, принятого PCI Express – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	i_msix_size_max	только для чтения	8	0x1f
[23:16]	i_msix_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	i_msix_present	только для чтения	1	0x1

1.3.10.2.1.18.10.1.1 i_msix_size_max

i_msix_size поддерживает значения между 0 и i_msix_size_max. Максимальный размер трансляции - 2i_msix_size_offset+i_msix_size_max.

1.3.10.2.1.18.10.1.2 i_msix_size_offset

Минимальный размер трансляции - 2i_msix_size_offset.

1.3.10.2.1.18.10.1.3 i_msix_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.10.2 Регистр CONTROL 0x8

Входная трансляция прерывания MSI-X, принятого PCI Express - управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:21]	зарезервировано	только для чтения	11	0x0
[20:16]	i_msix_size	чтение-запись	5	0x0
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	i_msix_enable	чтение-запись	1	0x0

1.3.10.2.1.18.10.2.1 i_msix_size

Размер этого окна трансляции выражается как 2msix_size_offset+msix_size. Ограничена полями i_msix_size_max и i_msix_size_offset

1.3.10.2.1.18.10.2.2 i_msix_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*i_msix_enable == 1

*i_msix_src_base[63:(12+i_msix_size)] == AXI Address[63:(12+i_msix_size)]

1.3.10.2.1.18.10.3 Регистр BASE_LO 0x10

Входная трансляция прерывания MSI-X, принятого PCI Express - младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	i_msix_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.10.3.1 i_msix_base_lo

i_msix_src_base[31:12]

1.3.10.2.1.18.10.4 Регистр BASE_HI 0x14

Входная трансляция прерывания MSI-X, принятого PCI Express - старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	i_msix_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.10.4.1 i_msix_base_hi

i_msix_src_base[63:32].

1.3.10.2.1.18.11 I_ISUB

PCIe_MASTER	I_ISUB_BASE	0x83e0
AXI_MASTER	I_ISUB_BASE	0x3e0

Входная трансляция субтрактивного декодирования.

Чтение и запись от PCIe к AXI, не попадающая во все остальные адресный регион входных ресурсов и трансляций, обрабатываются как ошибки или передаются в AXI, в зависимости от конфигурации этой трансляции.

1.3.10.2.1.18.11.1 Регистр CAPABILITIES 0x0

Входная трансляция субтрактивного декодирования – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:1]	зарезервировано	только для чтения	31	0x0
[0]	subtractive_decode_present	только для чтения	1	0x1

1.3.10.2.1.18.11.1.1 subtractive_decode_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.11.2 Регистр STATUS 0x4

Входная трансляция субтрактивного декодирования – статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.11.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.11.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.11.3 Регистр CONTROL 0x8

Входная трансляция субтрактивного декодирования – управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:1]	зарезервировано	только для чтения	31	0x0
[0]	ingress_sub_enable	чтение-запись	1	0x1

1.3.10.2.1.18.11.3.1 ingress_sub_enable

Разрешение субтрактивного декодирования.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.12 MSGF_MISC

PCIe_MASTER	MSGF_MISC_BASE	0x8400
AXI_MASTER	MSGF_MISC_BASE	0x400

Контроллер полученных прерываний и сообщений - статус прочих прерываний.

1.3.10.2.1.18.12.1 Регистр STATUS 0x0

Контроллер полученных прерываний и сообщений - статус прочих прерываний.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:16]	pcie_core_event	только для чтения; запись 1 для очистки	16	0x0
[15:8]	зарезервировано	только для чтения	8	0x0
[7]	egress_address_translation_error	только для чтения; запись 1 для очистки	1	0x0
[6]	ingress_address_translation_error	только для чтения; запись 1 для очистки	1	0x0
[5]	master_error	только для чтения; запись 1 для очистки	1	0x0
[4]	slave_error	только для чтения; запись 1 для очистки	1	0x0
[3]	uncorrectable_write_error	только для чтения; запись 1 для очистки	1	0x0
[2]	зарезервировано	только для чтения	1	0x0
[1]	rx_msg_overflow	только для чтения; запись 1 для очистки	1	0x0
[0]	rx_msg_avail	только для чтения; запись 1 для очистки	1	0x0

1.3.10.2.1.18.12.1.1 pcie_core_event

PCI Express Core Event[15:0]. Бит[i] устанавливается в 1, когда соответствующий входной порт p_set_interrupt[i] активизирован == 1 Исполнение ядра PCIe определяет, какие события передаются на p_set_interrupt.

1.3.10.2.1.18.12.1.2 egress_address_translation_error

Нарушение защиты трансляции выходного адреса. Устанавливается в 1 при нарушении защиты любой выходной трансляции.

1.3.10.2.1.18.12.1.3 ingress_address_translation_error

Нарушение защиты трансляции входного адреса. Устанавливается в 1 при нарушении защиты любой входной трансляции.

1.3.10.2.1.18.12.1.4 master_error

Ошибка ведущего устройства AXI. Устанавливается в 1, когда запрос на интерфейсе ведущего устройства AXI завершается со статусом SLVERR или DECERR.

1.3.10.2.1.18.12.1.5 slave_error

Ошибка ведомого устройства AXI. Устанавливается в 1, когда запрос на интерфейсе ведомого устройства AXI завершается со статусом SLVERR или DECERR.

1.3.10.2.1.18.12.1.6 uncorrectable_write_error

Неисправляемая аппаратная/программная ошибка записи. Устанавливается в 1 при обнаружении ошибки для запроса записи в регистр, не относящийся к DMA (от AXI илиPCIe).

1.3.10.2.1.18.12.1.7 rx_msg_overflow

Переполнение FIFO принятых сообщений. Устанавливается в 1 при любой попытке записи сообщения в FIFO принятых сообщений, однако при переполненном FIFO принятых сообщений сообщение будет сброшено. Поскольку объем FIFO принятых сообщений ограничен, программа должна обрабатывать принятые сообщения с высоким приоритетом, чтобы не допускать переполнения FIFO принятых сообщений и потери сообщений.

1.3.10.2.1.18.12.1.8 rx_msg_avail

Доступность FIFO принятых сообщений. Устанавливается в 1, когда FIFO принятых сообщений не пусто. Содержимое сообщение доступно на портах msgf_rx_fifo_type/msg/address_lo/address_hi/data ports.

1.3.10.2.1.18.12.2 Регистр MASK 0x4

Контроллер полученных прерываний и сообщений - статус прочих прерываний.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:16]	pcie_core_ event _mask	чтение-запись	16	0x0
[15:8]	зарезервировано	только для чтения	8	0x0
[7]	egress_address_translation_error_mask	чтение-запись	1	0x0
[6]	ingress_address_translation_error_mask	чтение-запись	1	0x0
[5]	master_error_mask	чтение-запись	1	0x0
[4]	slave_error_mask	чтение-запись	1	0x0
[3]	uncorrectable_write_error_mask	чтение-запись	1	0x0
[2]	зарезервировано	только для чтения	1	0x0
[1]	rx_msg_overflow_mask	чтение-запись	1	0x0
[0]	rx_msg_avail_mask	чтение-запись	1	0x0

1.3.10.2.1.18.12.2.1 pcie_core_event_mask

PCI Express Core Event[15:0]. Установить бит[i] в 1, чтобы разрешить генерацию прерываний для события[i] ядра PCI Express.

1.3.10.2.1.18.12.2.2 egress_address_translation_error_mask

Маска прерываний egress_address_translation_error. Устанавливается в 1, чтобы разрешить генерацию прерываний при egress_address_translation_error == 1.

1.3.10.2.1.18.12.2.3 ingress_address_translation_error_mask

Маска прерываний ingress_address_translation_error. Устанавливается в 1, чтобы разрешить генерацию прерываний при ingress_address_translation_error == 1.

1.3.10.2.1.18.12.2.4 master_error_mask

Маска прерываний master_error. Устанавливается в 1, чтобы разрешить генерацию прерываний при master_error == 1.

1.3.10.2.1.18.12.2.5 slave_error_mask

Маска прерываний slave_error. Устанавливается в 1, чтобы разрешить генерацию прерываний при slave_error == 1.

1.3.10.2.1.18.12.2.6 uncorrectable_write_error_mask

Маска прерываний uncorrectable_write_error. Устанавливается в 1, чтобы разрешить генерацию прерываний при uncorrectable_write_error == 1.

1.3.10.2.1.18.12.2.7 rx_msg_overflow_mask

Маска прерываний rx_msg_overflow. Устанавливается в 1, чтобы разрешить генерацию прерываний при rx_msg_overflow == 1.

1.3.10.2.1.18.12.2.8 rx_msg_avail_mask

Маска прерываний x_msg_avail. Устанавливается в 1, чтобы разрешить генерацию прерываний при x_msg_avail == 1.

1.3.10.2.1.18.12.3 Регистр SLAVE_ID 0x8

AXI ID ошибки ведомого устройства.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:8]	зарезервировано	только для чтения	24	0x0
[7:0]	slave_error_id	только для чтения	8	0x0

1.3.10.2.1.18.12.3.1 slave_error_id

AXI ID первой обнаруженной ошибки SLVERR или DECERR интерфейса ведомого устройства AXI. Очищается, когда в msgf_misc_status[4] записывается 1, чтобы разрешить последующей ошибке записать новый ID ошибки AXI.

1.3.10.2.1.18.12.4 Регистр MASTER_ID 0xc

AXI ID ошибки ведущего устройства.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:8]	зарезервировано	только для чтения	24	0x0
[7:0]	master_error_id	только для чтения	8	0x0

1.3.10.2.1.18.12.4.1 master_error_id

AXI ID первой обнаруженной ошибки интерфейса ведущего устройства AXI. Очищается, когда в msgf_misc_status[5] записывается 1, чтобы разрешить последующей ошибке записать новый ID ошибки AXI.

1.3.10.2.1.18.12.5 Регистр INGRESS_ID 0x10

AXI ID ошибки входа.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:8]	зарезервировано	только для чтения	24	0x0
[7:0]	ingress_error_id	только для чтения	8	0x0

1.3.10.2.1.18.12.5.1 ingress_error_id

AXI ID первой обнаруженной ошибки защиты трансляции входного адреса. Очищается, когда в msgf_misc_status[6] записывается 1, чтобы разрешить последующей ошибке записать новый ID ошибки AXI.

1.3.10.2.1.18.12.6 EGRESS_ID Register 0x14

AXI ID ошибки выхода.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:8]	зарезервировано	только для чтения	24	0x0
[7:0]	egress_error_id	только для чтения	8	0x0

1.3.10.2.1.18.12.6.1 egress_error_id

AXI ID первой обнаруженной ошибки защиты трансляции выходного адреса. Очищается, когда в msgf_misc_status[7] записывается 1, чтобы разрешить последующей ошибке записать новый ID ошибки AXI.

1.3.10.2.1.18.13 MSGF_LEG

PCIe_MASTER	MSGF_LEG_BASE	0x8420
AXI_MASTER	MSGF_LEG_BASE	0x420

Контроллер принятых прерываний - статус/маска унаследованных прерываний

1.3.10.2.1.18.13.1 Регистр STATUS 0x0

Legacy Interrupt Status.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:4]	зарезервировано	только для чтения	28	0x0
[3]	msgf_leg_status_intd	только для чтения	1	0x0
[2]	msgf_leg_status_intc	только для чтения	1	0x0
[1]	msgf_leg_status_intb	только для чтения	1	0x0
[0]	msgf_leg_status_inta	только для чтения	1	0x0

1.3.10.2.1.18.13.1.1 msgf_leg_status_intd

msgf_leg_status_intd генерируется из принятых сообщений PCIe, Assert_INTD и Deassert_INTD. msgf_leg_status_intd будет очищен только после того. как будут обслужены все источники прерывания INTD, а их регистры прерывания очищены.

- 0 - Прерывание неактивно
- 1 - Прерывание активно

1.3.10.2.1.18.13.1.2 msgf_leg_status_intc

msgf_leg_status_intc генерируется из принятых сообщений PCIe, Assert_INTC и Deassert_INTC. msgf_leg_status_intc будет очищен только после того. как будут обслужены все источники прерывания INTC, а их регистры прерывания очищены.

- 0 - Прерывание неактивно
- 1 - Прерывание активно

1.3.10.2.1.18.13.1.3 msgf_leg_status_intb

msgf_leg_status_intb генерируется из принятых сообщений PCIe, Assert_INTB и Deassert_INTB. msgf_leg_status_intb будет очищен только после того. как будут обслужены все источники прерывания INTB, а их регистры прерывания очищены.

- 0 - Прерывание неактивно
- 1 - Прерывание активно

1.3.10.2.1.18.13.1.4 msgf_leg_status_inta

- 1 - Прерывание разрешено

1.3.10.2.1.18.14 MSGF_MSI

PCIe_MASTER	MSGF_MSI_BASE	0x8440
AXI_MASTER	MSGF_MSI_BASE	0x440

Контроллер принятых прерываний - статус/маска прерываний MSI

1.3.10.2.1.18.14.1 Регистр STATUS_LO 0x0

Статус прерываний MSI [31:0].

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	msgf_msi_status	только для чтения; запись 1 для очистки	32	0x0

1.3.10.2.1.18.14.1.1 msgf_msi_status

Статус вектора прерываний MSI. msgf_msi_status[i] устанавливается в 1, когда получено прерывание MSI с вектором i.

- 0 - Прерывание неактивно
- 1 - Прерывание активно

1.3.10.2.1.18.14.2 Регистр STATUS_HI 0x4

MSI Interrupt Status[63:32].

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	msgf_msi_status	только для чтения; запись 1 для очистки	32	0x0

1.3.10.2.1.18.14.2.1 msgf_msi_status

Статус вектора прерываний MSI. msgf_msi_status[i] устанавливается в 1, когда получено прерывание MSI с вектором i.

- 0 - Прерывание неактивно
- 1 - Прерывание активно

1.3.10.2.1.18.14.3 Регистр MASK_LO 0x8

Маска прерываний MSI [31:0].

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	msgf_msi_mask	чтение-запись	32	0x0

1.3.10.2.1.18.14.3.1 msgf_msi_mask

Прерывание генерируется на выходном порте int_msi[0], когда (msgf_msi_status[31:0] & msgf_msi_mask[31:0]) != 0.

- 0 - Прерывание запрещено
- 1 - Прерывание разрешено

1.3.10.2.1.18.14.4 Регистр MASK_HI 0xc

Маска прерываний MSI [63:32].

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	msgf_msi_mask	чтение-запись	32	0x0

1.3.10.2.1.18.14.4.1 msgf_msi_mask

Маска вектора прерывания MSI. Прерывание генерируется на выходном порте int_msi[1], когда (msgf_msi_status[63:32] & msgf_msi_mask[63:32]) != 0.

- 0 - Прерывание запрещено
- 1 - Прерывание разрешено

1.3.10.2.1.18.15 MSGF_DMA

PCIe_MASTER	MSGF_DMA_BASE	0x8460
AXI_MASTER	MSGF_DMA_BASE	0x460

Контроллер принятых прерываний - статус/маска прерываний DMA

1.3.10.2.1.18.15.1 Регистр STATUS 0x0

Статус прерываний DMA.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:1]	зарезервировано	только для чтения	31	0x0
[0]	msgf_dma_status	только для чтения	1	0x0

1.3.10.2.1.18.15.1.1 msgf_dma_status

Статус прерываний DMA. msgf_dma_status == 1, когда активно одно или более прерываний DMA. msgf_dma_status очищается при очистке регистров статуса прерывания AXI соответствующего канала DMA.

- 0 - Прерывание неактивно
- 1 - Прерывание активно

1.3.10.2.1.18.15.2 Регистр MASK 0x4

Маска прерываний DMA.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:1]	зарезервировано	только для чтения	31	0x0
[0]	msgf_dma_mask	чтение-запись	1	0x0

1.3.10.2.1.18.15.2.1 msgf_dma_mask

Прерывание генерируется на выходном порте int_dma, когда (msgf_dma_status & msgf_dma_mask) != 0.

- 0 - Прерывание запрещено
- 1 - Прерывание разрешено

1.3.10.2.1.18.16 MSGF_RX_FIFO

PCIe_MASTER	MSGF_RX_FIFO_BASE	0x8480
AXI_MASTER	MSGF_RX_FIFO_BASE	0x480

Контроллер принятых прерываний - FIFO принятых прерываний и сообщений

1.3.10.2.1.18.16.1 Регистр LEVEL 0x0

FIFO принятых прерываний и сообщений – уровень

Поле	Наименование	Доступ	Разрядность	Сброс
[31:8]	зарезервировано	только для чтения	24	0x0
[7:0]	level	только для чтения	8	0x0

1.3.10.2.1.18.16.1.1 level

FIFO принятых прерываний и сообщений - уровень. Если не 0, то имеются ожидающие прерывания или сообщения. Содержимое самого первого по времени прерывания/сообщения доступно посредством чтения регистров msgf_rx_fifo_type/msg/address_lo/address_hi/data.

По окончании считывания текущего прерывания/сообщения текущий элемент удаляется из FIFO при помощи записи в регистр msgf_rx_fifo_pop.

1.3.10.2.1.18.16.2 Регистр POP 0x4

FIFO принятых прерываний и сообщений – удаление элемента

Поле	Наименование	Доступ	Разрядность	Сброс
[31:1]	зарезервировано	только для чтения	31	0x0
[0]	pop	только для чтения	1	0x0

1.3.10.2.1.18.16.2.1 pop

Записать pop == 1, чтобы удалить верхний элемент прерывания/сообщения из FIFO.

1.3.10.2.1.18.16.3 Регистр TYPE 0x8

FIFO полученных прерываний и сообщений - тип сообщения/прерывания

Поле	Наименование	Доступ	Разрядность	Сброс
[31:16]	requester_id	только для чтения	16	0x0
[15:2]	зарезервировано	только для чтения	14	0x0
[1:0]	intr_type	только для чтения	2	0x0

1.3.10.2.1.18.16.3.1 requester_id

ID инициатора PCI Express, считанный из входного пакета TLP. Формат {Bus[7:0], Function[7:0]} для источников, поддерживающих ID альтернативной маршрутизации (ARI) PCI Express, и {Bus[7:0], Device[4:0], Function[2:0]} для источников PCIe, не поддерживающих ARI.

1.3.10.2.1.18.16.3.2 intr_type

Тип прерывания/сообщения

- 0 - Получено сообщение
- 1 - Получено прерывание MSI
- 2 - Получено прерывание MSI-X
- 3 - Зарезервировано

1.3.10.2.1.18.16.4 Регистр MSG 0xc

Заголовок принятого сообщения

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	зарезервировано	только для чтения	8	0x0
[23:16]	message_tag	только для чтения	8	0x0
[15:8]	message_code	только для чтения	8	0x0
[7:4]	зарезервировано	только для чтения	4	0x0
[3]	message_payload_present	только для чтения	1	0x0
[2:0]	message_routing	только для чтения	3	0x0

1.3.10.2.1.18.16.4.1 message_tag

Байт 6 (тег) принятого TLP сообщения. Действителен только для intr_type == Received Message.

1.3.10.2.1.18.16.4.2 message_code

Байт 7 (код сообщения) принятого TLP сообщения. Действителен только для intr_type == Received Message.

1.3.10.2.1.18.16.4.3 message_payload_present

Присутствие данных содержимого сообщения. Действителен только для intr_type == Received Message.

- 0 - Нет содержимого
- 1 - Есть содержимое

1.3.10.2.1.18.16.4.4 message_routing

Байт0[2:0] (маршрутизация сообщения) принятого TLP сообщения. Действителен только для intr_type == Received Message.

1.3.10.2.1.18.16.5 Регистр ADDRESS_LO 0x10

Адрес[31:0] принятого сообщения/прерывания.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	address_lo	только для чтения	32	0x0

1.3.10.2.1.18.16.5.1 address_lo

Когда intr_type == Received MSI-X/MSI Interrupt
[31:0] - адрес[31:0] полученного TLP записи MSI-X/MSI.
Когда intr_type == Received Message
[31:24] - байт 12 TLP сообщения.
[23:16] - байт 13 TLP сообщения.
[15:8] - байт 14 TLP сообщения.
[7:0] - байт 15 TLP сообщения.

1.3.10.2.1.18.16.6 Регистр ADDRESS_HI 0x14

Адрес[63:32] принятого сообщения/прерывания.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	address_hi	только для чтения	32	0x0

1.3.10.2.1.18.16.6.1 address_hi

Когда intr_type == Received MSI-X/MSI Interrupt
[31:0] - адрес[63:32] полученного TLP записи MSI-X/MSI.
Когда intr_type == Received Message
[31:24] - байт 8 TLP сообщения.
[23:16] - байт 9 TLP сообщения.
[15:8] - байт 10 TLP сообщения.
[7:0] - байт 11 TLP сообщения.

1.3.10.2.1.18.16.7 Регистр DATA 0x18

Содержимое данных[31:0] принятого сообщения/прерывания.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	data	только для чтения	32	0x0

1.3.10.2.1.18.16.7.1 data

Когда intr_type == Received MSI-X/MSI Interrupt
[31:0] - содержимое данных[31:0] полученного TLP записи MSI-X/MSI.
Когда intr_type == Received Message
[31:0] - содержимое данных[31:0] полученного TLP сообщения. Действителен только для intr_type == Received Message cmessage_payload_present == 1.

1.3.10.2.1.18.17 TX_PCIE_IO

PCIe_MASTER	TX_PCIE_IO_BASE	0x8600
AXI_MASTER	TX_PCIE_IO_BASE	0x600

Генерация запроса чтения/записи ввода-вывода PCIe.

1.3.10.2.1.18.17.1 Регистр EXECUTE 0x0

Выполнение запроса чтения/записи ввода-вывода PCIe.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:26]	зарезервировано	только для чтения	6	0x0
[25:24]	io_done_status	только для чтения	2	0x0
[23:17]	зарезервировано	только для чтения	7	0x0
[16]	io_done	только для чтения	1	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8]	io_busy	только для чтения	1	0x0
[7:1]	зарезервировано	только для чтения	7	0x0
[0]	io_execute	только для чтения	1	0x0

1.3.10.2.1.18.17.1.1 io_done_status

Статус выполненной транзакции ввода-вывода. Очищается при записи 1 в io_execute. Устанавливается, когда io_done устанавливается в 1.

- 0 - OK
- 1 - Зарезервировано
- 2 - SLVERR

- 3 – DECERR

1.3.10.2.1.18.17.1.2 io_done

Выполненная транзакция ввода-вывода. Очищается при записи 1 в o_execute, чтобы начать новую транзакцию ввода-вывода. Устанавливается в 1, когда транзакция завершена.

- 0 - Не выполнена
- 1 - Выполнена

1.3.10.2.1.18.17.1.3 io_busy

Транзакция ввода-вывода занята. Устанавливается в 1 при записи 1 в io_execute. Очищается, когда io_done устанавливается в 1, указывая, что транзакция выполнена.

- 0 – Готова
- 1 – Занята

1.3.10.2.1.18.17.1.4 io_execute

Выполнение транзакции ввода-вывода. Записать 1, чтобы выполнить транзакцию ввода-вывода. Транзакция создается с использованием значений регистров CONTROL, ADDRESS и DATA.

1.3.10.2.1.18.17.2 Регистр CONTROL 0x4

Выполнение запроса чтения/записи ввода-вывода PCIe – управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	зарезервировано	только для чтения	20	0x0
[11:8]	io_be	чтение-запись	4	0x0
[7:1]	зарезервировано	только для чтения	7	0x0
[0]	io_type	чтение-запись	1	0x0

1.3.10.2.1.18.17.2.1 io_be

Разрешение байт данных транзакции ввода-вывода [3:0]. [3]==Data[31:24]..[0]==Data[7:0]. Разрешаться должны смежные байты. Разрешение несмежных байт (0x5, 0x9, 0xa, 0xb и 0xd) приводит к прерыванию транзакции со статусом ошибки.

1.3.10.2.1.18.17.2.2 io_type

Тип транзакции ввода/вывода.

- 0 - Чтение ввода-вывода
- 1 - Запись ввода-вывода

1.3.10.2.1.18.17.3 Регистр ADDRESS 0x8

Выполнение запроса чтения/записи ввода-вывода PCIe – адрес

Поле	Наименование	Доступ	Разрядность	Сброс
[31:2]	io_address	чтение-запись	30	0x0
[1:0]	зарезервировано	только для чтения	2	0x0

1.3.10.2.1.18.17.3.1 io_address

Адрес ввода-вывода[31:2] Адрес ввода-вывода[1:0] всегда 00.

1.3.10.2.1.18.17.4 Регистр DATA 0xc

Выполнение запроса чтения/записи ввода-вывода PCIe - записываемые данные

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	io_data	чтение-запись	32	0x0

1.3.10.2.1.18.17.4.1 io_data

Записываемые данные ввода-вывода[31:0]

1.3.10.2.1.18.17.5 Регистр DONE_DATA 0x10

Выполнение запроса чтения/записи ввода-вывода PCIe - считываемые данные

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	io_done_data	только для чтения	32	0x0

1.3.10.2.1.18.17.5.1 io_done_data

Данные завершения чтения ввода-вывода[31:0].

1.3.10.2.1.18.18 16.17 TX_PCIE_MSG

PCIe_MASTER	TX_PCIE_MSG_BASE	0x8620
AXI_MASTER	TX_PCIE_MSG_BASE	0x620

Генерация запроса сообщения PCIe.

1.3.10.2.1.18.18.1 Регистр EXECUTE 0x0

Выполнение запроса сообщения PCIe.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:26]	зарезервировано	только для чтения	6	0x0
[25:24]	msg_done_status	только для чтения	2	0x0
[23:17]	зарезервировано	только для чтения	7	0x0
[16]	msg_done	только для чтения	1	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8]	msg_busy	только для чтения	1	0x0
[7:1]	зарезервировано	только для чтения	7	0x0
[0]	msg_execute	только для чтения	1	0x0

1.3.10.2.1.18.18.1.1 msg_done_status

Статус выполненной транзакции сообщения. Очищается при записи 1 в msg_execute. Устанавливается, когда msg_done устанавливается в 1.

- 0 - OK
- 1 - Зарезервировано
- 2 - SLVERR
- 3 – DECERR

1.3.10.2.1.18.18.1.2 msg_done

Выполненная транзакция сообщения. Очищается при записи 1 в msg_execute, чтобы начать новую транзакцию сообщения. Устанавливается в 1, когда транзакция завершена.

- 0 - Не выполнена.
- 1 - Выполнена.

1.3.10.2.1.18.18.1.3 msg_busy

Транзакция сообщения занята. Устанавливается в 1 при записи 1 в msg_execute. Очищается, когда msg_done устанавливается в 1, указывая, что транзакция выполнена.

- 0 – Готова
- 1 – Занята

1.3.10.2.1.18.18.1.4 msg_execute

Выполнение транзакции сообщения. Записать 1, чтобы выполнить транзакцию сообщения. Транзакция создается с использованием значений регистров CONTROL, ADDRESS и DATA.

1.3.10.2.1.18.18.2 CONTROL Register 0x4

PCIe Message Request Execution – Control

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24]	msg_has_data	чтение-запись	1	0x0
[23:16]	msg_tag	чтение-запись	8	0x0
[15:8]	msg_code	чтение-запись	8	0x0
[7:0]	msg_fmt_type	чтение-запись	8	0x0

1.3.10.2.1.18.18.2.1 msg_has_data

Присутствие содержимого сообщения. Когда msg_has_data == 1, msg_data - это данные содержимого, передаваемого в сообщении. msg_has_data должен соответствовать msg_code.

- 0 - Не содержит данные содержимого.
- 1 - Содержит 1 DWORD данных содержимого.

1.3.10.2.1.18.18.2.2 msg_tag

Тег сообщения[7:0] Байт 6 заголовка TLP сообщения. Устанавливается в 0x0, если сообщение не использует поле тега.

1.3.10.2.1.18.18.2.3 msg_code

Код сообщения[7:0] Байт 7 заголовка TLP сообщения. Определяет тип сообщения. Коды сообщений определяются спецификацией PCI Express и соответствующими ECN.

1.3.10.2.1.18.18.2.4 msg_fmt_type

Формат и тип сообщения. Байт 0 заголовка TLP сообщения. Должен соответствовать коду сообщения.

1.3.10.2.1.18.18.3 Регистр SPECIFIC_LO 0x8

Выполнение запроса сообщения PCIe - Message Specific[31:0].

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	msg_tlp_hdr11	чтение-запись	8	0x0
[23:16]	msg_tlp_hdr10	чтение-запись	8	0x0
[15:8]	msg_tlp_hdr9	чтение-запись	8	0x0
[7:0]	msg_tlp_hdr8	чтение-запись	8	0x0

1.3.10.2.1.18.18.3.1 msg_tlp_hdr11

Байт 11 заголовка TLP сообщения.

1.3.10.2.1.18.18.3.2 msg_tlp_hdr10

Байт 10 заголовка TLP сообщения.

1.3.10.2.1.18.18.3.3 msg_tlp_hdr9

Байт 9 заголовка TLP сообщения.

1.3.10.2.1.18.18.3.4 msg_tlp_hdr8

Байт 8 заголовка TLP сообщения.

1.3.10.2.1.18.18.4 Регистр SPECIFIC_HI 0xc

Выполнение запроса сообщения PCIe - Message Specific[63:32].

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	msg_tlp_hdr15	чтение-запись	8	0x0
[23:16]	msg_tlp_hdr14	чтение-запись	8	0x0
[15:8]	msg_tlp_hdr13	чтение-запись	8	0x0
[7:0]	msg_tlp_hdr12	чтение-запись	8	0x0

1.3.10.2.1.18.18.4.1 msg_tlp_hdr15

Байт 15 заголовка TLP сообщения.

1.3.10.2.1.18.18.4.2 msg_tlp_hdr14

Байт 14 заголовка TLP сообщения.

1.3.10.2.1.18.18.4.3 msg_tlp_hdr13

Байт 13 заголовка TLP сообщения.

1.3.10.2.1.18.18.4.4 msg_tlp_hdr12

Байт 12 заголовка TLP сообщения.

1.3.10.2.1.18.18.5 Регистр DATA 0x10

Выполнение запроса сообщения PCIe - Message Data Payload[31:0].

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	msg_data	чтение-запись	32	0x0

1.3.10.2.1.18.18.5.1 msg_data

Данные содержимого сообщения.
Используется только при msg_has_data, установленном в 1.

1.3.10.2.1.18.19 TRAN_INGRESS

PCIe_MASTER	TRAN_INGRESS_0_BASE	0x8800
PCIe_MASTER	TRAN_INGRESS_1_BASE	0x8820
PCIe_MASTER	TRAN_INGRESS_2_BASE	0x8840
PCIe_MASTER	TRAN_INGRESS_3_BASE	0x8860
PCIe_MASTER	TRAN_INGRESS_4_BASE	0x8880
PCIe_MASTER	TRAN_INGRESS_5_BASE	0x88a0
PCIe_MASTER	TRAN_INGRESS_6_BASE	0x88c0
PCIe_MASTER	TRAN_INGRESS_7_BASE	0x88e0
AXI_MASTER	TRAN_INGRESS_0_BASE	0x800
AXI_MASTER	TRAN_INGRESS_1_BASE	0x820
AXI_MASTER	TRAN_INGRESS_2_BASE	0x840
AXI_MASTER	TRAN_INGRESS_3_BASE	0x860
AXI_MASTER	TRAN_INGRESS_4_BASE	0x880
AXI_MASTER	TRAN_INGRESS_5_BASE	0x8a0
AXI_MASTER	TRAN_INGRESS_6_BASE	0x8c0
AXI_MASTER	TRAN_INGRESS_7_BASE	0x8e0

Входная трансляция AXI. Только при наличии поддержки входной трансляции.

1.3.10.2.1.18.19.1 Регистр CAPABILITIES 0x0

Входная трансляция AXI – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	ingress_size_max	только для чтения	8	0x1f
[23:16]	ingress_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	ingress_present	только для чтения	1	0x1

1.3.10.2.1.18.19.1.1 ingress_size_max

Максимальный размер трансляции.

1.3.10.2.1.18.19.1.2 ingress_size_offset

Минимальный размер трансляции.

1.3.10.2.1.18.19.1.3 ingress_present

Индикатор присутствия трансляции.

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.19.2 Регистр STATUS 0x4

Входная трансляция AXI – статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.19.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции.

1.3.10.2.1.18.19.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции.

1.3.10.2.1.18.19.3 Регистр CONTROL 0x8

Входная трансляция AXI – управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:28]	ingress_attr_w	чтение-запись	4	0x0
[27:24]	ingress_attr_r	чтение-запись	4	0x0
[23]	ingress_attr_enable	чтение-запись	1	0x0
[22:21]	зарезервировано	только для чтения	2	0x0
[20:16]	ingress_size	чтение-запись	5	0x0
[15:14]	зарезервировано	только для чтения	2	0x0
[13:8]	зарезервировано	только для чтения	6	0x0
[7:5]	зарезервировано	только для чтения	3	0x0
[4]	зарезервировано	только для чтения	1	0x0
[3]	ingress_invalid	чтение-запись	1	0x0
[2]	ingress_security_enable	чтение-запись	1	0x0
[1]	зарезервировано	только для чтения	1	0x0
[0]	ingress_enable	чтение-запись	1	0x0

1.3.10.2.1.18.19.3.1 ingress_attr_w

Перекрытие входного кэша записи. Когда ingress_attr_enable == 1, AXI порт транзакции m_awcache устанавливается в это значение при направлении в AXI транзакций записи, попадающих в адресный регион этой трансляции. Когда ingress_attr_enable == 0, атрибуты кэша по умолчанию определяются примененным cfg_pcie_rx_awcache.

1.3.10.2.1.18.19.3.2 ingress_attr_r

Перекрытие входного кэша чтения. Когда ingress_attr_enable == 1, AXI порт транзакции m_arcache устанавливается в это значение при направлении в AXI транзакций чтения, попадающих в адресный регион этой трансляции. Когда ingress_attr_enable == 0, атрибуты кэша по умолчанию определяются примененным cfg_pcie_rx_arcache.

1.3.10.2.1.18.19.3.3 ingress_attr_enable

Перекрытие входного кэша записи/чтения.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.19.3.4 ingress_size

Размер трансляции. Размер окна трансляции в байтах конфигурируется как 2ingress_size_offset+ingress_size. ingress_size должно быть <= ingress_size_max.

1.3.10.2.1.18.19.3.5 ingress_invalid

Разрешение отмены трансляции.

- 0 - Принять транзакции, попадающие в адресный регион этой трансляции.
- 1 - Отменить транзакции, попадающих в адресный регион этой трансляции. Транзакция не направляется получателю и обрабатывается как ошибка неподдерживаемого запроса.

1.3.10.2.1.18.19.3.6 ingress_security_enable

Разрешение защиты трансляции.

- 0 - Защита запрещена. При направлении транзакции к AXI порту m_arprot[1]/m_awprot[1] присваивается значение незащищенного (Non-Secure).
- 1 - Защита разрешена. При направлении транзакции к AXI порту m_arprot[1]/m_awprot[1] присваивается значение порта i_security_ingress[i], ассоциированного с трансляцией.

1.3.10.2.1.18.19.3.7 ingress_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:
*ingress_enable == 1
*ingress_src_base[63:(12+ingress_size)] == AXI Address[63:(12+ingress_size)]

1.3.10.2.1.18.19.4 Регистр SRC_BASE_LO 0x10

Входная трансляция AXI- младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	ingress_src_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.19.4.1 ingress_src_base_lo

ingress_src_base[31:12].

1.3.10.2.1.18.19.5 Регистр SRC_BASE_HI 0x14

Входная трансляция AXI- старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	ingress_src_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.19.5.1 ingress_src_base_hi

ingress_src_base[63:32].

1.3.10.2.1.18.19.6 Регистр DST_BASE_LO 0x18

Входная трансляция AXI- младшие разряды адреса приемника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	ingress_dst_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.19.6.1 ingress_dst_base_lo

ingress_dst_base[31:12]

1.3.10.2.1.18.19.7 Регистр DST_BASE_HI 0x1c

Входная трансляция AXI- старшие разряды адреса приемника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	ingress_dst_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.19.7.1 ingress_dst_base_hi

ingress_dst_base[63:32].

1.3.10.2.1.18.20 TRAN_EGRESS

PCIe_MASTER	TRAN_EGRESS_0_BASE	0x8c00
PCIe_MASTER	TRAN_EGRESS_1_BASE	0x8c20
PCIe_MASTER	TRAN_EGRESS_2_BASE	0x8c40
PCIe_MASTER	TRAN_EGRESS_3_BASE	0x8c60
PCIe_MASTER	TRAN_EGRESS_4_BASE	0x8c80
PCIe_MASTER	TRAN_EGRESS_5_BASE	0x8ca0
PCIe_MASTER	TRAN_EGRESS_6_BASE	0x8cc0
PCIe_MASTER	TRAN_EGRESS_7_BASE	0x8ce0
AXI_MASTER	TRAN_EGRESS_0_BASE	0xc00
AXI_MASTER	TRAN_EGRESS_1_BASE	0xc20
AXI_MASTER	TRAN_EGRESS_2_BASE	0xc40
AXI_MASTER	TRAN_EGRESS_3_BASE	0xc60
AXI_MASTER	TRAN_EGRESS_4_BASE	0xc80
AXI_MASTER	TRAN_EGRESS_5_BASE	0xca0
AXI_MASTER	TRAN_EGRESS_6_BASE	0xcc0
AXI_MASTER	TRAN_EGRESS_7_BASE	0xce0

Выходная трансляция AXI. Только при наличии поддержки выходной трансляции.

1.3.10.2.1.18.20.1 Регистр CAPABILITIES 0x0

Выходная трансляция AXI – возможности

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	egress_size_max	только для чтения	8	0x1f
[23:16]	egress_size_offset	только для чтения	8	0xc
[15:1]	зарезервировано	только для чтения	15	0x0
[0]	egress_present	только для чтения	1	0x1

1.3.10.2.1.18.20.1.1 egress_size_max

egress_size поддерживает значения между 0 и egress_size_max. * Максимальный размер трансляции - 2egress_size_offset+egress_size_max.

1.3.10.2.1.18.20.1.2 egress_size_offset

Минимальный размер трансляции - 2egress_size_offset.

1.3.10.2.1.18.20.1.3 egress_present

Индикатор присутствия трансляции

- 0 - Не присутствует
- 1 - Присутствует

1.3.10.2.1.18.20.2 Регистр STATUS 0x4

Выходная трансляция AXI – статус

Поле	Наименование	Доступ	Разрядность	Сброс
[31:25]	зарезервировано	только для чтения	7	0x0
[24:16]	wr_pending_ctr	только для чтения	9	0x0
[15:9]	зарезервировано	только для чтения	7	0x0
[8:0]	rd_pending_ctr	только для чтения	9	0x0

1.3.10.2.1.18.20.2.1 wr_pending_ctr

Количество незавершенных транзакций записи для этой трансляции. Количество ожидающих трансляций не может быть больше, чем количество одновременно ожидающих транзакций, поддерживаемых ядром Expresso DMA (всегда <512 транзакций), и поэтому ядро Expresso DMA не нуждается в отправке управляющих запросов из-за этого поля.

1.3.10.2.1.18.20.2.2 rd_pending_ctr

Количество незавершенных транзакций чтения для этой трансляции. Количество ожидающих трансляций не может быть больше, чем количество одновременно ожидающих транзакций, поддерживаемых ядром Expresso DMA (всегда <512 транзакций), и поэтому ядро Expresso DMA не нуждается в отправке управляющих запросов из-за этого поля.

1.3.10.2.1.18.20.3 Регистр CONTROL 0x8

Выходная трансляция AXI – управление

Поле	Наименование	Доступ	Разрядность	Сброс
[31:28]	egress_attr_w	чтение-запись	4	0x0
[27:24]	egress_attr_r	чтение-запись	4	0x0
[23]	egress_attr_enable	чтение-запись	1	0x0
[22:21]	зарезервировано	только для чтения	2	0x0
[20:16]	egress_size	чтение-запись	5	0x0
[15:14]	зарезервировано	только для чтения	2	0x0
[13:8]	зарезервировано	только для чтения	6	0x0
[7:5]	зарезервировано	только для чтения	3	0x0
[4]	зарезервировано	только для чтения	1	0x0
[3]	egress_invalid	чтение-запись	1	0x0
[2]	egress_security_enable	чтение-запись	1	0x0
[1]	зарезервировано	только для чтения	1	0x0
[0]	egress_enable	чтение-запись	1	0x0

1.3.10.2.1.18.20.3.1 egress_attr_w

Перекрытие выходных атрибутов записи. Когда egress_attr_enable == 1, атрибуты PCIe {ID Based Ordering, Relaxed Ordering, No Snoop} устанавливаются в egress_attr_w[2:0] при направлении в PCIe транзакций записи, попадающих в адресный регион этой трансляции. Когда egress_attr_enable == 0, атрибуты PCIe все устанавливаются в 0 при направлении в PCIe транзакций записи, попадающих в адресный регион этой трансляции. Атрибуты, которые не разрешено устанавливать из-за разрешенной функциональности регистра конфигурации PCI Express, будут очищены перед тем, как транзакция направляется в PCIe.

1.3.10.2.1.18.20.3.2 egress_attr_r

Перекрытие выходных атрибутов чтения. Когда egress_attr_enable == 1, атрибуты PCIe {ID Based Ordering, Relaxed Ordering, No Snoop} устанавливаются в egress_attr_r[2:0] при направлении в PCIe транзакций чтения, попадающих в адресный регион этой трансляции. Когда egress_attr_enable == 0, атрибуты PCIe все устанавливаются в 0 при направлении в PCIe транзакций чтения, попадающих в адресный регион этой трансляции. Атрибуты, которые не разрешено устанавливать из-за разрешенной функциональности регистра конфигурации PCI Express, будут очищены перед тем, как транзакция направляется в PCIe.

1.3.10.2.1.18.20.3.3 egress_attr_enable

Перекрытие выходных атрибутов записи/чтения.

- 0 - Запрещено
- 1 – Разрешено

1.3.10.2.1.18.20.3.4 egress_size

Размер трансляции. Размер окна трансляции в байтах конфигурируется как 2egress_size_offset+egress_size. egress_size должно быть <= egress_size_max.

1.3.10.2.1.18.20.3.5 egress_invalid

Разрешение отмены трансляции.

- 0 - Принять транзакции, попадающие в адресный регион этой трансляции.
- 1 - Отменить транзакции, попадающие в адресный регион этой трансляции. Транзакция не направляется получателю, и AXI возвращает ответ DECERR.

1.3.10.2.1.18.20.3.6 egress_security_enable

Разрешение защиты трансляции.

- 0 - Защита запрещена. Уровень защиты AXI s_arprot[1]/s_awprot[1] игнорируется, и он не влияет на трансляцию транзакции.
- 1 - Защита разрешена. Если уровень защиты s_arprot[1]/s_awprot[1] транзакции AXI совпадает с уровнем защиты порта i_security_egress[i], ассоциированного с трансляцией, то транзакция направляется PCIe; в противном случае транзакция сбрасывается и возвращается ответ SLVERR.

1.3.10.2.1.18.20.3.7 egress_enable

Разрешение трансляции. "Попадание" в адресный регион трансляции имеет место, когда выполняются оба условия:

*egress_enable == 1

*egress_src_base[63:(12+egress_size)] == AXI Address[63:(12+egress_size)]

1.3.10.2.1.18.20.4 Регистр SRC_BASE_LO 0x10

Выходная трансляция AXI- младшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	egress_src_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.20.4.1 egress_src_base_lo

egress_src_base[31:12]

1.3.10.2.1.18.20.5 Регистр SRC_BASE_HI 0x14

Выходная трансляция AXI- старшие разряды адреса источника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	egress_src_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.20.5.1 egress_src_base_hi

egress_src_base[63:32].

1.3.10.2.1.18.20.6 Регистр DST_BASE_LO 0x18

Выходная трансляция AXI- младшие разряды адреса приемника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:12]	egress_dst_base_lo	чтение-запись	20	0x0
[11:0]	зарезервировано	только для чтения	12	0x0

1.3.10.2.1.18.20.6.1 egress_dst_base_lo

egress_dst_base[31:12].

1.3.10.2.1.18.20.7 Регистр DST_BASE_HI 0x1c

Выходная трансляция AXI- старшие разряды адреса приемника

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	egress_dst_base_hi	чтение-запись	32	0x0

1.3.10.2.1.18.20.7.1 egress_dst_base_hi

egress_dst_base[63:32].

1.3.10.2.1.18.21 DMA_CHANNEL

PCIe_MASTER	DMA_CHANNEL_0_BASE	0x0
PCIe_MASTER	DMA_CHANNEL_1_BASE	0x80
PCIe_MASTER	DMA_CHANNEL_2_BASE	0x100
PCIe_MASTER	DMA_CHANNEL_3_BASE	0x180
AXI_MASTER	DMA_CHANNEL_0_BASE	0x0
AXI_MASTER	DMA_CHANNEL_1_BASE	0x80
AXI_MASTER	DMA_CHANNEL_2_BASE	0x100
AXI_MASTER	DMA_CHANNEL_3_BASE	0x180

1.3.10.2.1.18.21.1 Регистр SRC_Q_PTR_LO 0x0

Младшие разряды базового адреса очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:6]	start_addr_lo	чтение-запись	26	0x0
[5:2]	read_attr	чтение-запись	4	0x0
[1]	queue_enable	чтение-запись	1	0x0
[0]	queue_location	чтение-запись	1	0x0

1.3.10.2.1.18.21.1.1 start_addr_lo

Базовый адрес очереди Очереди должны быть выровнены с шагом 64 байта (start_addr_lo[5:0] == 00000).

1.3.10.2.1.18.21.1.2 read_attr

Атрибуты чтения очереди. Атрибуты транзакции, используемые для чтения очереди. Если местоположение очереди == AXI, то read_attr[3:0] используется для m_arcache[3:0]. Если местоположение очереди == PCIe, то read_attr[2:0] используется для PCIe Attr[2:0].

1.3.10.2.1.18.21.1.3 queue_enable

Разрешение очереди

- 0 - Запрещено - Канал DMA не будет считывать из очереди.
- 1 - Разрешено - Канал DMA при необходимости будет осуществлять выборку элементов из очереди, пока Q_LIMIT != Q_NEXT.

1.3.10.2.1.18.21.1.4 queue_location

Местоположение очереди

- 0 - Очередь расположена в памяти PCIe.
- 1 - Очередь расположена в памяти AXI.

1.3.10.2.1.18.21.2 Регистр SRC_Q_PTR_HI 0x4

Старшие разряды базового адреса очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	start_addr_hi	чтение-запись	32	0x0

1.3.10.2.1.18.21.2.1 start_addr_hi

Базовый адрес очереди[63:32]. Должен быть установлен в 0x0, если очередь расположена в 32-битовом адресном пространстве.

1.3.10.2.1.18.21.3 Регистр SRC_Q_SIZE 0x8

Размер очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	queue_size	чтение-запись	32	0x0

1.3.10.2.1.18.21.3.1 queue_size

Размер очереди. Количество элементов в очереди. queue_size должно быть >= 2. Для поддержки программной/аппаратной принадлежности очереди контроля потока данных требуется минимум 2 элемента.

1.3.10.2.1.18.21.4 Регистр SRC_Q_LIMIT 0xc

Указатель границ очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	limit	чтение-запись	32	0x0

1.3.10.2.1.18.21.4.1 limit

Очередь управления потоком данных - указатель границ. Указатель первого элемента очереди "принадлежит" программе. Инкрементируется программой, чтобы предоставить каналу DMA дополнительные элементы для управления. По достижении Q_LIMIT аппаратная часть канала DMA приостановит работу и не будет использовать элементы очереди до тех пор, пока Q_LIMIT не будет передвинут, чтобы предоставить для выполнения дополнительные элементы.

1.3.10.2.1.18.21.5 Регистр DST_Q_PTR_LO 0x10

Младшие разряды	Наименование	Доступ	Разрядность	Сброс
[31:6]	start_addr_lo	чтение-запись	26	0x0
[5:2]	read_attr	чтение-запись	4	0x0
[1]	queue_enable	чтение-запись	1	0x0
[0]	queue_location	чтение-запись	1	0x0

1.3.10.2.1.18.21.5.1 start_addr_lo

Базовый адрес очереди[31:6]. Очереди должны быть выровнены с шагом 64 байта (start_addr_lo[5:0] == 00000).

1.3.10.2.1.18.21.5.2 read_attr

Атрибуты чтения очереди. Атрибуты транзакции, используемые для чтения очереди. Если местоположение очереди == AXI, то read_attr[3:0] используется для m_arcache[3:0]. Если местоположение очереди == PCIe, то read_attr[2:0] используется для PCIe Attr[2:0].

1.3.10.2.1.18.21.5.3 queue_enable

Разрешение очереди

- 0 - Запрещено - Канал DMA не будет считывать из очереди.
- 1 - Разрешено - Канал DMA при необходимости будет осуществлять выборку элементов из очереди, пока Q_LIMIT != Q_NEXT.

1.3.10.2.1.18.21.5.4 queue_location

Местоположение очереди
0 - Очередь расположена в памяти PCIe.
1 - Очередь расположена в памяти AXI.

1.3.10.2.1.18.21.6 Регистр DST_Q_PTR_HI 0x14

Старшие разряды базового адреса очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	start_addr_hi	чтение-запись	32	0x0

1.3.10.2.1.18.21.7 start_addr_hi

Базовый адрес очереди[63:32]. Должен быть установлен в 0x0, если очередь расположена в 32-битовом адресном пространстве.

1.3.10.2.1.18.21.8 Регистр DST_Q_SIZE 0x18

Размер очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	queue_size	чтение-запись	32	0x0

1.3.10.2.1.18.21.8.1 queue_size

Размер очереди. Количество элементов в очереди. queue_size должен быть >= 2. Для поддержки программной/аппаратной принадлежности очереди контроля потока данных требуется минимум 2 элемента.

1.3.10.2.1.18.21.9 Регистр DST_Q_LIMIT 0x1c

Указатель границ очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	limit	чтение-запись	32	0x0

1.3.10.2.1.18.21.9.1 limit

Очередь управления потоком данных - указатель границ. Указатель первого элемента очереди "принадлежит" программе. Инкрементируется программой, чтобы предоставить каналу DMA дополнительные элементы для управления. По достижении Q_LIMIT аппаратная часть канала DMA приостановит работу и не будет использовать элементы очереди до тех пор, пока Q_LIMIT не будет передвинут, чтобы предоставить для выполнения дополнительные элементы.

1.3.10.2.1.18.21.10 Регистр STAS_Q_PTR_LO 0x20

Младшие разряды базового адреса очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:6]	start_addr_lo	чтение-запись	26	0x0
[5:2]	read_attr	чтение-запись	4	0x0
[1]	queue_enable	чтение-запись	1	0x0
[0]	queue_location	чтение-запись	1	0x0

1.3.10.2.1.18.21.10.1 start_addr_lo

Базовый адрес очереди[31:6]. Очереди должны быть выровнены с шагом 64 байта (start_addr_lo[5:0] == 00000).

1.3.10.2.1.18.21.10.2 read_attr

Атрибуты чтения очереди. Атрибуты транзакции, используемые для чтения очереди. Если местоположение очереди == AXI, то read_attr[3:0] используется для m_arcache[3:0]. Если местоположение очереди == PCIe, то read_attr[2:0] используется для PCIe Attr[2:0].

1.3.10.2.1.18.21.10.3 queue_enable

Разрешение очереди

- 0 - Запрещено - Канал DMA не будет считывать из очереди.
- 1 - Разрешено - Канал DMA при необходимости будет осуществлять выборку элементов из очереди, пока Q_LIMIT != Q_NEXT.

1.3.10.2.1.18.21.10.4 queue_location

Местоположение очереди
0 - Очередь расположена в памяти PCIe.
1 - Очередь расположена в памяти AXI.

1.3.10.2.1.18.21.11 Регистр STAS_Q_PTR_HI 0x24

Старшие разряды базового адреса очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	start_addr_hi	чтение-запись	32	0x0

1.3.10.2.1.18.21.11.1 start_addr_hi

Базовый адрес очереди[63:32]. Должен быть установлен в 0x0, если очередь расположена в 32-битовом адресном пространстве.

1.3.10.2.1.18.21.12 Регистр STAS_Q_SIZE 0x28

Размер очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	queue_size	чтение-запись	32	0x0

1.3.10.2.1.18.21.12.1 queue_size

Размер очереди. Количество элементов в очереди. queue_size должно быть >= 2. Для поддержки программной/аппаратной принадлежности очереди контроля потока данных требуется минимум 2 элемента.

1.3.10.2.1.18.21.13 Регистр STAS_Q_LIMIT 0x2c

Указатель границ очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	limit	чтение-запись	32	0x0

1.3.10.2.1.18.21.13.1 limit

Очередь управления потоком данных - указатель границ. Указатель первого элемента очереди "принадлежит" программе. Инкрементируется программой, чтобы предоставить каналу DMA дополнительные элементы для управления. По достижении Q_LIMIT аппаратная часть канала DMA приостановит работу и не будет использовать элементы очереди до тех пор, пока Q_LIMIT не будет передвинут, чтобы предоставить для выполнения дополнительные элементы.

1.3.10.2.1.18.21.14 Регистр STAD_Q_PTR_LO 0x30

Младшие разряды базового адреса очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:6]	start_addr_lo	чтение-запись	26	0x0
[5:2]	read_attr	чтение-запись	4	0x0
[1]	queue_enable	чтение-запись	1	0x0
[0]	queue_location	чтение-запись	1	0x0

1.3.10.2.1.18.21.14.1 start_addr_lo

Базовый адрес очереди[31:6]. Очереди должны быть выровнены с шагом 64 байта (start_addr_lo[5:0] == 00000).

1.3.10.2.1.18.21.14.2 read_attr

Атрибуты чтения очереди. Атрибуты транзакции, используемые для чтения очереди. Если местоположение очереди == AXI, то read_attr[3:0] используется для m_arcache[3:0]. Если местоположение очереди == PCIe, то read_attr[2:0] используется для PCIe Attr[2:0].

1.3.10.2.1.18.21.14.3 queue_enable

Разрешение очереди

- 0 - Запрещено - Канал DMA не будет считывать из очереди.
- 1 - Разрешено - Канал DMA при необходимости будет осуществлять выборку элементов из очереди, пока Q_LIMIT != Q_NEXT.

1.3.10.2.1.18.21.14.4 queue_location

Местоположение очереди

- 0 - Очередь расположена в памяти PCIe.
- 1 - Очередь расположена в памяти AXI.

1.3.10.2.1.18.21.15 Регистр STAD_Q_PTR_HI 0x34

Старшие разряды базового адреса очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	start_addr_hi	чтение-запись	32	0x0

1.3.10.2.1.18.21.15.1 start_addr_hi

Базовый адрес очереди[63:32]. Должен быть установлен в 0x0, если очередь расположена в 32-битовом адресном пространстве.

1.3.10.2.1.18.21.16 Регистр STAD_Q_SIZE 0x38

Размер очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	queue_size	чтение-запись	32	0x0

1.3.10.2.1.18.21.16.1 queue_size

Размер очереди. Количество элементов в очереди. queue_size должен быть >= 2. Для поддержки программной/аппаратной принадлежности очереди контроля потока данных требуется минимум 2 элемента.

1.3.10.2.1.18.21.17 Регистр STAD_Q_LIMIT 0x3c

Указатель границ очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	limit	чтение-запись	32	0x0

1.3.10.2.1.18.21.17.1 limit

Очередь управления потоком данных - указатель границ. Указатель первого элемента очереди "принадлежит" программе. Инкрементируется программой, чтобы предоставить каналу DMA дополнительные элементы для управления. По достижении Q_LIMIT аппаратная часть канала DMA приостановит работу и не будет использовать элементы очереди до тех пор, пока Q_LIMIT не будет передвинут, чтобы предоставить для выполнения дополнительные элементы.

1.3.10.2.1.18.21.18 Регистр SRC_Q_NEXT 0x40

Указатель следующего элемента очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	next	чтение-запись	32	0x0

1.3.10.2.1.18.21.18.1 next

Очередь управления потоком данных - указатель следующего элемента. Указатель следующего элемента очереди, который будет считываться аппаратной частью канала DMA. Инкрементируется аппаратной частью канала DMA по мере генерации запросов чтения очереди. Количество элементов очереди, доступных для канала DMA == SRC_Q_LIMIT - SRC_Q_NEXT (с учетом закольцовывания очереди). SRC_Q_NEXT не указывает, что элементы очереди были выполнены - только то, что канал DMA начал обработку элементов очереди. Программа должна записывать в этот регистр 0x0, чтобы инициализировать очередь, до разрешения канала DMA. Программе запрещено выполнять запись в этот регистр, пока канал DMA разрешен. SRC_Q_NEXT используется аппаратной частью канала DMA, чтобы отслеживать положение элемента в очереди, и не должно использоваться программным обеспечением DMA.

1.3.10.2.1.18.21.19 Регистр DST_Q_NEXT 0x44

Указатель следующего элемента очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	next	чтение-запись	32	0x0

1.3.10.2.1.18.21.19.1 next

Очередь управления потоком данных - указатель следующего элемента. Указатель следующего элемента очереди, который будет считываться аппаратной частью канала DMA. Инкрементируется аппаратной частью канала DMA по мере генерации запросов чтения очереди. Количество элементов очереди, доступных для канала DMA == DST_Q_LIMIT - DST_Q_NEXT (с учетом закольцовывания очереди). DST_Q_NEXT не указывает, что элементы очереди были выполнены - только то, что канал DMA начал обработку элементов очереди. Программа должна записывать в этот регистр 0x0, чтобы инициализировать очередь, до разрешения канала DMA. Программе запрещено выполнять запись в этот регистр, пока канал DMA разрешен. DST_Q_NEXT используется аппаратной частью канала DMA, чтобы отслеживать положение элемента в очереди, и не должно использоваться программным обеспечением DMA.

1.3.10.2.1.18.21.20 Регистр STAS_Q_NEXT 0x48

Указатель следующего элемента очереди

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	next	чтение-запись	32	0x0

1.3.10.2.1.18.21.20.1 next

Очередь управления потоком данных - указатель следующего элемента. Указатель следующего элемента очереди, который будет записываться аппаратной частью канала DMA со статусом завершения DMA. Инкрементируется программой канала DMA, по мере того, как элементы резервируются для получения статуса завершения DMA. Количество элементов очереди, доступных для канала DMA == STAS_Q_LIMIT - STAS_Q_NEXT (с учетом закольцовывания очереди). STAS_Q_NEXT не указывает, что элементы очереди были записаны - только то, что канал DMA зарезервировал элементы для получения статуса завершения DMA. Программа должна записывать в этот регистр 0x0, чтобы инициализировать очередь, до разрешения канала DMA. Программе запрещено выполнять запись в этот регистр, пока канал DMA разрешен. STAS_Q_NEXT используется аппаратной частью канала DMA, чтобы отслеживать положение элемента в очереди, и не должно использоваться программным обеспечением DMA.

1.3.10.2.1.18.21.21 Регистр STAD_Q_NEXT 0x4c

Только запись для инициализации канала DMA.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	next	чтение-запись	32	0x0

1.3.10.2.1.18.21.21.1 next

Очередь управления потоком данных - указатель следующего элемента. Указатель следующего элемента очереди, который будет записываться аппаратной частью канала DMA со статусом завершения DMA. Инкрементируется программой канала DMA, по мере того, как элементы резервируются для получения статуса завершения DMA. Количество элементов

очереди, доступных для канала DMA == STAD_Q_LIMIT - STAD_Q_NEXT (с учетом закольцовывания очереди). STAD_Q_NEXT не указывает, что элементы очереди были записаны - только то, что канал DMA зарезервировал элементы для получения статуса завершения DMA. Программа должна записывать в этот регистр 0x0, чтобы инициализировать очередь, до разрешения канала DMA. Программе запрещено выполнять запись в этот регистр, пока канал DMA разрешен. STAD_Q_NEXT используется аппаратной частью канала DMA, чтобы отслеживать положение элемента в очереди, и не должно использоваться программным обеспечением DMA.

1.3.10.2.1.18.21.22 Регистр SCRATCH0 0x50

Регистр обмена данными общего назначения

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	scratch0	чтение-запись	32	0x0

1.3.10.2.1.18.21.22.1 scratch0

Регистр обмена данными общего назначения. Предназначен для разрешения передачи информации внутри программы. Например, приложения с AXI CPU и PCIe CPU могут использовать этот регистр для передачи информации между CPU. Реализация канала DMA не использует и не изменяет эту информацию.

1.3.10.2.1.18.21.23 Регистр SCRATCH1 0x54

Регистр обмена данными общего назначения

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	scratch1	чтение-запись	32	0x0

1.3.10.2.1.18.21.23.1 scratch1

Регистр обмена данными общего назначения. Предназначен для разрешения передачи информации внутри программы. Например, приложения с AXI CPU и PCIe CPU могут использовать этот регистр для передачи информации между CPU. Реализация канала DMA не использует и не изменяет эту информацию.

1.3.10.2.1.18.21.24 Регистр SCRATCH2 0x58

Регистр обмена данными общего назначения

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	scratch2	чтение-запись	32	0x0

1.3.10.2.1.18.21.24.1 scratch2

Регистр обмена данными общего назначения. Предназначен для разрешения передачи информации внутри программы. Например, приложения с AXI CPU и PCIe CPU могут использовать этот регистр для передачи информации между CPU. Реализация канала DMA не использует и не изменяет эту информацию.

1.3.10.2.1.18.21.25 Регистр SCRATCH3 0x5c

Регистр обмена данными общего назначения

Поле	Наименование	Доступ	Разрядность	Сброс
[31:0]	scratch3	чтение-запись	32	0x0

1.3.10.2.1.18.21.25.1 scratch3

Регистр обмена данными общего назначения. Предназначен для разрешения передачи информации внутри программы. Например, приложения с AXI CPU и PCIe CPU могут использовать этот регистр для передачи информации между CPU. Реализация канала DMA не использует и не изменяет эту информацию.

1.3.10.2.1.18.21.26 Регистр PCIE_INTERRUPT_CONTROL 0x60

Управление прерываниями PCI Express

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	зарезервировано	только для чтения	8	0x0
[23:16]	coalesce_count	чтение-запись	8	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2]	sgl_int_enable	чтение-запись	1	0x0
[1]	dma_err_int_enable	чтение-запись	1	0x0
[0]	interrupt_mask	чтение-запись	1	0x0

1.3.10.2.1.18.21.26.1 coalesce_count

Счетчик объединения прерываний DMA SGL для PCIe. Управляет частотой, с которой события прерывания DMA SGL, завершение элемента SGL источника, у которых установлены в 1 оба бита управления SGL "EOP" и "Interrupt", вызывают прерывания PCIe. Поддерживается внутренний, отдельный для каждого канала DMA счетчик объединения DMA SGL для PCIe. Для каждого события прерывания DMA SGL: если счетчик объединения DMA SGL для PCIe равен счетчику объединения прерываний, то генерируется прерывание PCIe, и если счетчик объединения DMA SGL очищается; в противном случае счетчик объединения DMA SGL инкрементируется. Счетчик объединения DMA SGL также очищается, когда DMA Enable == 0. Этот механизм позволяет программе конфигурировать канал DMA, чтобы канал DMA выставлял прерывание PCIe после каждых 1-256 событий прерывания DMA SGL. Когда в счетчик объединения прерываний DMA SGL для PCIe программируется отличное от 0 значение, программа должна предполагать, что в конце всех пакетных пересылок DMA в счетчике объединения DMA SGL для PCIe могут ожидать прерывания, который не вызвали прерывание PCIe, поскольку не пришли новые прерывания, и счетчик объединения DMA SGL для PCIe не достиг порога объединения прерываний DMA SGL для PCIe.

1.3.10.2.1.18.21.26.2 sgl_int_enable

Разрешение события прерывания DMA SGL для PCIe

- 0 - Запрещено. События прерывания DMA SGL игнорируются и не будут вызывать генерацию прерываний PCI Express.
- 1 – Разрешено. События прерывания DMA SGL разрешены для данного канала DMA.

1.3.10.2.1.18.21.26.3 dma_err_int_enable

Разрешение события прерывания ошибки DMA для PCIe

- 0 - Запрещено. События прерывания ошибки DMA игнорируются и не будут вызывать генерацию прерываний PCI Express.
- 1 – Разрешено. События прерывания ошибки DMA разрешены для данного канала DMA.

1.3.10.2.1.18.21.26.4 interrupt_mask

Разрешение прерываний PCIe

- 0 - Запретить прерывания. Удерживать прерывания PCIe для данного канала DMA и активизировать их, когда разрешение прерываний снова будет установлен в 1. Ожидающие прерывания сбрасываются, когда MA_Enable == 0.
- 1 - Разрешить прерывания. Разрешить прерываниям PCIe быть активизированными данным каналом DMA.

1.3.10.2.1.18.21.27 Регистр PCIE_INTERRUPT_STATUS 0x64

Legacy Interrupt Status.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:4]	зарезервировано	только для чтения	28	0x0
[3]	software_int	только для чтения; запись 1 для очистки	1	0x0
[2]	dma_sgl_int	только для чтения; запись 1 для очистки	1	0x0
[1]	dma_error_int	только для чтения; запись 1 для очистки	1	0x0
[0]	зарезервировано	только для чтения	1	0x0

1.3.10.2.1.18.21.27.1 software_int

Статус программного прерывания PCIe.

- 0 - Нет ожидающего прерывания software_int.
- 1 - Было сгенерировано программное прерывание.

1.3.10.2.1.18.21.27.2 dma_sgl_int

Статус прерывания DMA SGL для PCIe

- 0 - Нет ожидающего прерывания dma_sgl_int.
- 1 - События прерывания DMA SGL вызвали генерацию прерывания.

1.3.10.2.1.18.21.27.3 dma_error_int

Статус прерывания PCIe ошибки DMA.

- 0 - Нет ожидающего прерывания dma_error_int.
- 1 - События прерывания ошибки DMA вызвали генерацию прерывания.

1.3.10.2.1.18.21.28 Регистр AXI_INTERRUPT_CONTROL 0x68

Управление прерываниями PCI Express

Поле	Наименование	Доступ	Разрядность	Сброс
[31:24]	зарезервировано	только для чтения	8	0x0
[23:16]	coalesce_count	чтение-запись	8	0x0
[15:3]	зарезервировано	только для чтения	13	0x0
[2]	sgl_int_enable	чтение-запись	1	0x0
[1]	dma_err_int_enable	чтение-запись	1	0x0
[0]	interrupt_enable	чтение-запись	1	0x0

1.3.10.2.1.18.21.28.1 coalesce_count

Счетчик объединения прерываний DMA SGL для AXI. Управляет частотой, с которой события прерывания DMA SGL, завершение элемента SGL источника, у которых установлены в

1 оба бита управления SGL "EOP" и "Interrupt", вызывают прерывания AXI. Поддерживается внутренний, отдельный для каждого канала DMA счетчик объединения DMA SGL для AXI. Для каждого события прерывания DMA SGL: если счетчик объединения DMA SGL для AXI равен счетчику объединения прерываний, то генерируется прерывание AXI, и если счетчик объединения DMA SGL очищается; в противном случае счетчик объединения DMA SGL инкрементируется. Счетчик объединения DMA SGL для AXI также очищается, когда DMA Enable == 0. Этот механизм позволяет программе конфигурировать канал DMA, чтобы канал DMA выставлял прерывание AXI после каждых 1-256 событий прерывания DMA SGL. Когда в счетчик объединения прерываний DMA SGL для AXI программируется отличное от 0 значение, программа должна предполагать, что в конце всех пакетных пересылок DMA в счетчике объединения DMA SGL для AXI могут ожидать прерывания, который не вызвали прерывание AXI, поскольку не пришли новые прерывания, и счетчик объединения DMA SGL для AXI не достиг порога объединения прерываний DMA SGL для AXI.

1.3.10.2.1.18.21.28.2 sgl_int_enable

Разрешение события прерывания DMA SGL для AXI

- 0 - Запрещено. События прерывания DMA SGL игнорируются и не будут вызывать генерацию прерываний AXI.
- 1 – Разрешено. События прерывания DMA SGL разрешены для данного канала DMA.

1.3.10.2.1.18.21.28.3 dma_err_int_enable

Разрешение события прерывания ошибки DMA для AXI

- 0 - Запрещено. События прерывания ошибки DMA игнорируются и не будут вызывать генерацию прерываний AXI.
- 1 – Разрешено. События прерывания ошибки DMA разрешены для данного канала DMA.

1.3.10.2.1.18.21.28.4 interrupt_enable

Разрешение прерываний AXI

- 0 - Запретить прерывания. Удерживать прерывания AXI для данного канала DMA и активизировать их, когда разрешение прерываний AXI снова будет установлено в 1. Ожидающие прерывания сбрасываются, когда MA_Enable == 0.
- 1 - Разрешить прерывания. Разрешить прерываниям AXI быть активизированными данным каналом DMA

1.3.10.2.1.18.21.29 Регистр AXI_INTERRUPT_STATUS 0x6c

Статус прерываний AXI.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:4]	зарезервировано	только для чтения	28	0x0
[3]	software_int	только для чтения; запись 1 для очистки	1	0x0
[2]	dma_sgl_int	только для чтения; запись 1 для очистки	1	0x0
[1]	dma_error_int	только для чтения; запись 1 для очистки	1	0x0
[0]	зарезервировано	только для чтения	1	0x0

1.3.10.2.1.18.21.29.1 software_int

Статус программного прерывания AXI.

- 0 - Нет ожидающего прерывания software_int.
- 1 - Было сгенерировано программное прерывание.

1.3.10.2.1.18.21.29.2 dma_sgl_int

Статус прерывания DMA SGL для AXI

- 0 - Нет ожидающего прерывания dma_sgl_int.
- 1 - События прерывания DMA SGL вызвали генерацию прерывания.

1.3.10.2.1.18.21.29.3 dma_error_int

Статус прерывания ошибки DMA для AXI

- 0 - Нет ожидающего прерывания dma_error_int.
- 1 - События прерывания ошибки DMA вызвали генерацию прерывания.

1.3.10.2.1.18.21.30 Регистр PCIE_INTERRUPT_ASSERT 0x70

Активизация прерываний PCIe.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:4]	зарезервировано	только для чтения	28	0x0
[3]	pcie_software_interrupt	только для чтения	1	0x0
[2:0]	зарезервировано	только для чтения	3	0x0

1.3.10.2.1.18.21.30.1 pcie_software_interrupt

Программное прерывание PCIe. Записать 1 в этот регистр, чтобы генерировать программное прерывание PCIe. Прерывание генерируется и передается через ядро PCI Express Core к PCIe точно так же, как прерывание DMA (с помощью того же вектора прерываний). Прерывание генерируются записью в этот регистр независимо от того, разрешен ли канал DMA.

1.3.10.2.1.18.21.31 Регистр AXI_INTERRUPT_ASSERT 0x74

Активизация прерываний AXI.

Поле	Наименование	Доступ	Разрядность	Сброс
[31:4]	зарезервировано	только для чтения	28	0x0
[3]	axi_software_interrupt	только для чтения	1	0x0
[2:0]	зарезервировано	только для чтения	3	0x0

1.3.10.2.1.18.21.31.1 axi_software_interrupt

Программное прерывание AXI. Записать 1 в этот регистр, чтобы генерировать программное прерывание AXI. Прерывание генерируется и передается к AXI точно так же, как прерывание DMA (с помощью того же вектора прерываний). Прерывание генерируются записью в этот регистр независимо от того, разрешен ли канал DMA.

1.3.10.2.1.18.21.32 Регистр DMA_CONTROL 0x78

Управление каналом DMA

Поле	Наименование	Доступ	Разрядность	Сброс
[31:3]	зарезервировано	только для чтения	29	0x0
[2]	cmpl_stat_q_elem_size	чтение-запись	1	0x0
[1]	dma_reset	чтение-запись	1	0x0
[0]	dma_enable	чтение-запись	1	0x0

1.3.10.2.1.18.21.32.1 cmpl_stat_q_elem_size

Размер элемента очереди статуса завершения канала DMA. Если для данного приложения требуется информация UserID и/или UserHandle, то при начальной установке приложение должно указать для канала DMA 64-битные элементы очереди статуса. Если эта информация не нужна, то при начальной установке канала можно уменьшить число разрядов, необходимых для записи элементов очереди статуса.

- 0 - 32 бита - Элементы очереди статуса содержат только основную информацию DMA.
- 0 - 64 бита - Элементы очереди статуса включают дополнительную информацию UserID и/или UserHandle, которая может использоваться в некоторых приложениях.

1.3.10.2.1.18.21.32.2 dma_reset

Сброс канала DMA. Каждый канал DMA имеет небольшие буфера FIFO: FIFO SGL источника, FIFO SGL приемника, FIFO статуса завершения источника DMA и FIFO статуса завершения приемника DMA, чтобы разрешить перекрытие транзакций DMA для увеличения пропускной способности. При запрете канала DMA эти FIFO могут оказаться не полностью пустыми, и их, возможно, потребуется очистить, прежде чем канал DMA будет использован для новой операции.

- 0 - Обычный режим работы.
- 1 - Сбросить буфера FIFO: FIFO SGL источника, FIFO SGL приемника, FIFO статуса завершения источника DMA и FIFO статуса завершения приемника DMA.

1.3.10.2.1.18.21.32.3 dma_enable

Разрешение канала DMA

- 0 - Запретить канал DMA. Очереди канала DMA должны конфигурироваться только когда канал DMA запрещен. После запрещения канала DMA программа должна дождаться завершения всех незавершенных транзакций DMA, прежде чем снова разрешать канал DMA; в противном случае запрещенная, но не завершенная операция DMA может завершиться неудачно, что приведет к фатальной ошибке. Чтобы определить, завершилась ли предыдущая операция DMA, можно считать DMA Running. Информацию о методе остановки и возобновления операций DMA см. в разделе 14.6.
- 1 - Разрешить канал DMA. Канал DMA осуществляет выборку элементов очередей SGL источника и приемника, выполняет их и записывает статус завершения пакета DMA в очереди статуса завершения источника и приемника DMA. Канал DMA приостанавливает транзакции DMA, если в его распоряжении нет хотя бы одного доступного для выполнения элемента во всех очередях. Значение регистров Queue LIMIT увеличивается, чтобы предоставить дополнительные элементы для выполнения.

1.3.10.2.1.18.21.33 Регистр DMA_STATUS 0x7c

Статус канала DMA

Поле	Наименование	Доступ	Разрядность	Сброс
[31:16]	зарезервировано	только для чтения	16	0x0
[15]	channel_present	только для чтения	1	0x1
[14]	зарезервировано	только для чтения	1	0x0
[13:4]	channel_number	только для чтения	10	0x0
[3:1]	зарезервировано	только для чтения	3	0x0
[0]	dma_running	только для чтения	1	0x0

1.3.10.2.1.18.21.33.1 channel_present

Присутствие канала DMA. Во время инициализации драйвер DMA может считывать этот регистр во всех возможных местоположениях регистров канала DMA, чтобы определить количество реализованных каналов DMA.

- 0 – Нет регистров.
- 1 - В данном местоположении присутствует набор регистров канала DMA.

1.3.10.2.1.18.21.33.2 channel_number

Номер канала DMA[9:0]. Уникальный номер канала DMA, присвоенный данному каналу DMA. Регистр номера канала DMA предназначен для информации и не нужен для работы DMA. Номер канала DMA уникален для каждого канала DMA даже для многофункциональных применений SR-IOV.

1.3.10.2.1.18.21.33.3 dma_running

DMA работает. Прежде чем переводить DMA Enable из состояния 0 в состояние 1 или изменять содержимое регистров управление очередью канала DMA, программа должна считать DMA Running == 0, чтобы проверить, что операция DMA завершена, и что можно повторно инициализировать и повторно запускать канал DMA. Очереди источника и приемника канала DMA могут быть использованы для новых операций DMA (путем изменения содержимого очереди SGL) без необходимости запрещать канал DMA. В общем случае программа должна выполнять начальную установку SGL источника, SGL приемника, статуса завершения SGL источника и статуса завершения SGL приемника при инициализации драйвера, а также поддерживать разрешение канала DMA все время, пока загружается драйвер. При получении нового задания драйвер делает доступными для выполнения соответствующие элементы очереди, обновляя указатели Queue LIMIT.

- 0 - Канал DMA простаивает.
- 1 - Канал DMA работает.

1.3.10.2.1.19 Шина APB

1.3.10.2.1.19.1 Устройство CFG_CONSTANTS

APB_MASTER CFG_CONSTANTS_BASE 0x0

Регистры конфигурации ядра влияют на основную работу ядра и должны быть статическими во время работы.

1.3.10.2.1.19.1.1 Регистр REG0 0x0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..6	device_id	rw	сохраненное	сохранить	16	e00ch
15..	revision_id	rw	сохраненное	сохранить	16	19aah

Значение device_id возвращается при чтении регистра конфигурации Device ID (ID устройства).

Значение vendor_id возвращается при чтении регистра конфигурации Vendor ID (ID производителя).

1.3.10.2.1.19.1.2 1.1.2 Регистр REG1 0x4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..8	class_code	rw	сохраненное	сохранить	24	118000h
7..0	revision_id	rw	сохраненное	сохранить	8	4h

Значение class_code возвращается при чтении регистра конфигурации Class Code (код класса). Должно быть установлено корректное значение для реализуемого типа устройства; подробности установки кода класса см. в спецификации локальной шины PCI, версия 2.3, приложение D.

Значение revision_id возвращается при чтении регистра конфигурации Revision ID (ID версии).

1.3.10.2.1.19.1.3 1.1.3 Регистр REG2 0x8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	subsystem_id	rw	сохраненное	сохранить	16	e00ch
15.. 0	subsystem_vendor_i	rw	сохраненное	сохранить	16	19aah

Значение subsystem_id возвращается при чтении регистра конфигурации Subsystem ID (ID подсистемы).

Значение subsystem_vendor_id возвращается при чтении регистра конфигурации Subsystem Vendor ID (ID производителя подсистемы).

1.3.10.2.1.19.1.4 1.1.4 Регистр REG3 0xc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	Cardbus_cis_pointer	rw	сохраненное	сохранить	32	0h

Значение cardbus_cis_pointer возвращается при чтении регистра конфигурации Cardbus CIS Pointer (указатель Cardbus CIS). Устанавливается 0x00000000; если структура Cardbus CIS не реализована в памяти (что бывает редко), то устанавливается адрес структуры CIS.

1.3.10.2.1.19.1.5 1.1.5 Регистр REG4 0x10

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	base_address_cfgO_LO	rw	сохраненное	сохранить	16	0h
15.. 8	expanded_capabilities_pointer	rw	сохраненное	сохранить	8	0h
7	unadjusted_ack_tx_latency_limit_divider	rw	сохраненное	сохранить	1	0h
6	t1_bypass_addr_dec	rw	сохраненное	сохранить	1	0h
5	t1_tx_bypass_msg_dec	rw	сохраненное	сохранить	1	0h
4.. 3	interrupt_pin	rw	сохраненное	сохранить	2	0h
2	target_only	rw	сохраненное	сохранить	1	0h
1	expanded_capabilities_enable	rw	сохраненное	сохранить	1	0h
0	interrupt_enable	rw	сохраненное	сохранить	1	1h

base_address_cfgO_LO BAR0

expanded_capabilities_pointer Если установлен бит Expanded Capabilities Enable (разрешение расширенных настроек), то Expanded Capabilities Pointer (указатель расширенных настроек) должен быть установлен на адрес пункта реализованного пользователем списка настроек.

unadjusted_ack_tx_latency_limit_divider1 В сочетании с DIVIDER0, позволяет уменьшить задержку ядрас Rx TLP до Tx DLLP ACK.

0 Использование Unadjusted ACK TX Latency Limits (нескорректированных границ задержки ACK TX)

1 Использование Unadjusted ACK TX Latency Limits/2

2 Использование Unadjusted ACK TX Latency Limits/4

3 Использование Unadjusted ACK TX Latency Limits/8

t1_bypass_addr_dec Bypass address decode.

0 Выполнена нормальная, совместимая с PCI Express проверка достоверности адреса

1 Пропущена проверка достоверности адреса для памяти ввода/вывода и памяти с предвыборкой данных

t1_tx_bypass_msg_dec Релевантен только тогда, когда в ядре реализуется тип 1 регистров конфигурации (корневой
порт/коммутатор); релевантен только для переданных TLP сообщений.
0 Обычный режим работы
1 Все TLP сообщений переданы.
interrupt_pin Выбирает, какое из унаследованных прерываний используется.

- 0 - INTA
- 1 - INTB
- 2 - INTC
- 3 - INTD

target_only Устанавливается для исполнений пользователя, которые реализуют режим с поддержкой функционала только ведомого устройства.
expanded_capabilities_enable При установке разрешает поддержку пользовательской реализации дополнительных настроек.
interrupt_enable При установке разрешает поддержку прерываний.

1.3.10.2.1.19.1.6 Регистр REG5 0x14

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	base_address_cfgl_LO	rw	сохраненное	сохранить	16	8h
15.. 0	base_address_cfgO_HI	rw	сохраненное	сохранить	16	fffeh

base_address_cfg1_LO BAR1
base_address_cfgO_HI BAR0

1.3.10.2.1.19.1.7 1.1.7 Регистр REG560x18

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	base_address_cfg2_LO	rw	сохраненное	сохранить	16	ch
15.. 0	base_address_cfg1_HI	rw	сохраненное	сохранить	16	ffffh

base_address_cfg2_LO BAR2
base_address_cfg1_HI BAR1

1.3.10.2.1.19.1.8 Регистр REG7 0x1c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	base_address_cfg3_LO	rw	сохраненное	сохранить	16	ffffh
15.. 0	base_address_cfg2_HI	rw	сохраненное	сохранить	16	ffffh

base_address_cfg3_LO BAR3
base_address_cfg2_HI BAR2

1.3.10.2.1.19.1.9 Регистр REG4 0x10

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	base_address_cfg4_LO	rw	сохраненное	сохранить	16	0h
15.. 0	base_address_cfg3_HI	rw	сохраненное	сохранить	16	ffffh

base_address_cfg4_LO BAR4
base_address_cfg3_HI BAR3

1.3.10.2.1.19.1.10 Регистр REG9 0x24

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	base_address_cfg5_LO	rw	сохраненное	сохранить	16	0h
15.. 0	base_address_cfg4_HI	rw	сохраненное	сохранить	16	0h

base_address_cfg5_LO BAR5
base_address_cfg4_HI BAR4

1.3.10.2.1.19.1.11 Регистр REG10 0x28

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	expansion_rom_cfg_LO	rw	сохраненное	сохранить	16	0h
15.. 0	base_address_cfg5_HI	rw	сохраненное	сохранить	16	0h

expansion_rom_cfg_LO Определяет, реализован ли регистр базового адреса дополнительного ROM, и если реализован, то его размер.

base_address_cfg5_HI BAR5

1.3.10.2.1.19.1.12 Регистр REG11 0x2c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	quick_simulationjreduce_timeouts	rw	сохраненное	сохранить	1	0h
30	disable_automatic_pme_to_ack_message_generation	rw	сохраненное	сохранить	1	0h
29	bypass jrx_idle	rw	сохраненное	сохранить	1	0h
28	bypassjreceiver_detection	rw	сохраненное	сохранить	1	0h
27	bypass_power_state_change_indication	rw	сохраненное	сохранить	1	0h
26	enable_lOs_power_mgmt	rw	сохраненное	сохранить	1	1h
25	enable_l1_power_mgmt	rw	сохраненное	сохранить	1	1h
24	enable_l2_power_mgmt	rw	сохраненное	сохранить	1	1h
23	mask_skp_during_electrical_idle	rw	сохраненное	сохранить	1	0h
22	ignore_physical_errors	rw	сохраненное	сохранить	1	0h
21	direct_to_hot_reset	rw	сохраненное	сохранить	1	0h
20	direct_to_disable	rw	сохраненное	сохранить	1	0h
19	direct_to_loopback	rw	сохраненное	сохранить	1	0h
18	force_compliance_pattern	rw	сохраненное	сохранить	1	0h
17	force_scrambling_off	rw	сохраненное	сохранить	1	0h
16	quick_simulationjreduce_ts1	rw	сохраненное	сохранить	1	0h
15..0	expansion_rom_cfg_HI	rw	сохраненное	сохранить	16	0h

quick_simulation_reduce_timeouts Этот бит устанавливается (только для симуляции), чтобы изменить нормальную настроечную последовательность с целью сокращения времени настройки ядра Expresso 3.0.

disable_automatic_pme_to_ack_message_generation Устанавливается для того, чтобы определить, когда передается сообщение PME_TO_Ack.

0 Ядро передает сообщение PME_TO_Ack непосредственно после получения сообщения PME_Turn_Off без какого-либо вмешательства пользователя (передача планируется одновременно с активизацией pm_enter_l2)

1 Ядро передает сообщение PME_TO_Ack после активизации порта ядра pm_enter_l2.

bypass_rx_idle Bypass RX Idle: Устанавливается, чтобы все линии считали, что передатчик активен на всех линиях приема.

bypass_receiver_detection Bypass Receiver Detection: Устанавливается, чтобы обойти обнаружение приемника и всегда предполагать, что на всех линиях присутствует удаленный приемник.

bypass_power_state_change_indication Bypass Power State Change Indication: Устанавливается, чтобы считать, что PHY завершает изменения состояния питания сразу же, а не ждать, когда PHY сообщит об успешных изменениях состояния питания.

enable_lOs_power_mgmt Enable L0s Power Mgmt: Устанавливается для того, чтобы разрешить ядру функции управления питанием L0s.

enable_l1_power_mgmt Enable L1 Power Mgmt: Устанавливается для того, чтобы разрешить ядру функции управления питанием L1.

enable_l2_power_mgmt Enable L2 Power Mgmt: Устанавливается для того, чтобы разрешить ядру функции управления питанием L2.

mask_skp_during_electrical_idle Mask SKP During Electrical Idle. Если этот бит установлен, ядро не будет планировать передачу SKP стандартной последовательности символов во время электрического простоя.

ignore_physical_errors Ignore Physical Errors. Установка этого бита заставляет ядро игнорировать ошибки физического уровня, обнаруженные PHY, и пытаться обрабатывать пакеты стандартных последовательностей символов, DLLP и TLP в предположении, что входные данные не содержат ошибок.

direct_to_hotpreset Direct to Hot Reset. При установке переводит ядро в состояние горячего сброса.

direct_to_disable Direct to Disable. При установке переводит ядро в состояние DISABLED LTSSM.

direct_to_loopback Direct to Loopback. При установке переводит ядро в состояние LOOPBACK LTSSM и режим управляющего устройства петли.

force_compliance_pattern При установке этого бита на всех линиях ядра PCIe появляется последовательность согласования.

force_scrambling_off Заставляет ядро PCIe запретить весь требуемый скремблинг как передаваемых, так и принимаемых данных. Этот бит используется только для диагностики физического уровня и должен сбрасываться в 0, когда требуется обычное поведение PCIe. Этот бит зарезервирован только для Northwest Logic.

quick_simulation_reduce_ts1 Ускоряет настройку для симуляции

Expansion_rom_cfg_HI Определяет, реализован ли регистр базового адреса дополнительного ROM, и если реализован, то его размер.

1.3.10.2.1.19.1.13 Регистр REG12 0x30

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..26	replay _timerjrx_lOs_adjustment_LO	rw	сохраненное	сохранить	6	0h
25	enable_lls_power_mgmt	rw	сохраненное	сохранить	1	0h
24	unadjusted_replay_timer_limit_multiplier	rw	сохраненное	сохранить	1	0h
23..16	nfts	rw	сохраненное	сохранить	8	ffh
15.. 0	aspm_lOs_tx_entry_time	rw	сохраненное	сохранить	16	0h

replay_timer_rx_lOs_adjustment_LO Replay Timer Rx_L0s_Adjustment. Прибавляется (опционально с множителем) к не скорректированному значению Replay Timer, чтобы получить окончательное значение тайм-аута Replay Timer, в условных единицах.

enable_l1s_power_mgmt Enable ASPM L1 Power Mgmt: Устанавливается для разрешения функций управления питанием ASPM L1 ядра.

unadjustedjreplay_timer_limit_multiplier Unadjusted Replay Timer Limit Multiplier. Предназначен для поддержки PHY с большими задержками, а также кабельных исполнений PCI Express с большими задержками.

0 Рекомендованное значение для не скорректированного значения Replay Timer..

1 Используется удвоенное рекомендованное значение.

nfts NFTS устанавливается, чтобы запрос удаленного устройства посылался по истечение l0s; 0 == использование значения по умолчанию/

aspm_lOs_tx_entry_time ASPM L0s TX Entry Time; 0 == Использовать значение ядра по умолчанию 6.9uS

1.3.10.2.1.19.1.14 Регистр REG13 0x34

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..24	root_port_id_LO	rw	сохраненное	сохранить	8	0h
23..10	ack_timer_tx_lOs_adjustment	rw	сохраненное	сохранить	14	0h
9	include_tx_lOs_adjustment	rw	сохраненное	сохранить	1	0h
8	unadjusted_ack_tx_latency_limit_dividerO	rw	сохраненное	сохранить	1	0h
7..0	replay _timer_rx_lOs_adjustment_HI	rw	сохраненное	сохранить	8	0h

root_port_id_LO Root Port ID. Это 16-битовое поле предназначено для определения ID, используемого в PCIe для ID инициатора и ID исполнителя,

когда ядро работает в режиме нисходящего порта.

ack_timer_tx_lOs_adjustment ACK Timer TX L0s Adjustment. Устанавливается для указания времени, требуемого передающим цепям PHY

для перехода из состояния L0s в L0.

include_tx_lOs_adjustment При установке включает ACK Timer Tx_L0s_Adjustment в расчет значения the Ack TX Latency Limit.

unadjusted_ack_tx_latency_limit_dividerO Делитель ACK TX Latency Limit Divider[0]. Для получения дополнительной информации см. Divider[1] .

replay_timer_rx_lOs_adjustment_HI Replay Timer Rx_L0s_Adjustment. Прибавляется (опционально с множителем) к не скорректированному значению Replay Timer, чтобы получить окончательное значение тайм-аута Replay Timer, в условных единицах.

1.3.10.2.1.19.1.15 Регистр REG14 0x38

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	ignore_ecrc	rw	сохраненное	сохранить	1	0h
30	strip_rx_ecrc_enable	rw	сохраненное	сохранить	1	0h
29	ecrc_td_add_has_n	rw	сохраненное	сохранить	1	0h
28	rx_c_priority	rw	сохраненное	сохранить	1	0h
27	rx_c_forcejro	rw	сохраненное	сохранить	1	0h
26	rx_cut_thru	rw	сохраненное	сохранить	1	0h
25	ecrc_capable_disable	rw	сохраненное	сохранить	1	0h
24	tOjrx_bypass_msg_dec	rw	сохраненное	сохранить	1	1h
23	t1 jrx_bypass_msg_dec	rw	сохраненное	сохранить	1	0h
22	crs_enable	rw	сохраненное	сохранить	1	0h
21	mgmt_gen1_compatibility_mode_enable	rw	сохраненное	сохранить	1	0h
20	mgmt_force_5g_deemphais_3_5db_6db_n	rw	сохраненное	сохранить	1	0h
19	mgmt_force_5g_deemphais_en	rw	сохраненное	сохранить	1	0h
18	selectable_deemphasis_for_5g_operation	rw	сохраненное	сохранить	1	0h
17	direct_to_5gts	rw	сохраненное	сохранить	1	0h
16	support_5gts	rw	сохраненное	сохранить	1	1h
15	direct_to_8gts	rw	сохраненное	сохранить	1	0h
14	support_8gts	rw	сохраненное	сохранить	1	0h
13	inhibit jrx_idle	rw	сохраненное	сохранить	1	0h
12	inhibit jrx_det	rw	сохраненное	сохранить	1	0h
11	switch_port_mode	rw	сохраненное	сохранить	1	0h
10	pm_force_rp_mode	rw	сохраненное	сохранить	1	0h
9	advertise_infinite_ch_cd_credits_as_root_port	rw	сохраненное	сохранить	1	0h
8	root_port_mode	rw	сохраненное	сохранить	1	0h
7.. 0	root_port_id_HI	rw	сохраненное	сохранить	8	0h

ignore_ecrc Игнорировать ECRC для пакетов TLP, передаваемых пользователю на интерфейс приемника.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

stripjrx_ecrc_enable Если бит установлен, то ECRC удаляется из финального пакета RX TLP, если ECRC присутствует.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

ecrc_td_add_has_n Определяет значение бита TD (заголовок[23]) в переданных пакетах TLP.

0 (TD==1) указывает, что ECRC уже присутствует

1 (TD==1) указывает добавить ECRC

rx_c_priority Устанавливает наивысший приоритет для ожидающих завершений приема.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

rx_c_force_ro Принудительно устанавливает ослабленное упорядочивание на всех принятых завершениях, как будто все они имеют разрешенный атрибут ослабленного упорядочивания.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

rx_cut_thru Разрешает сквозную коммутацию, когда входной буфер приемника PCIe не ждет окончания пакета до начала передачи пакета на следующий уровень.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

ecrc_capable_disable Запрещает возможность предложения генерации/проверки ECRC.

0 ECRC предложено.

1 Возможность генерации AER ECRC и возможность проверки ECRC устанавливаются в 0 (не поддерживаются) во всех функциях.

t0_rx_bypass_msg_dec При реализации регистров конфигурации типа 0:

0 Обычный режим работы

1 Все действительные пакеты TLP сообщений, полученные по PCI (за исключением тех, которые маршрутизируются по ID и по адресу, поскольку их маршрутизация определяется типом), направляются в vc0_rx

t1_rx_bypass_msg_dec При реализации регистров конфигурации типа 1:

0 Обычный режим работы

1 Все действительные пакеты TLP сообщений, полученные по PCI (за исключением тех, которые маршрутизируются по ID и по адресу, поскольку их маршрутизация определяется типом), направляются в vc0_rx

crs_enable Разрешение статуса повтора запроса конфигурации.

0 Обычный режим работы; разрешены чтение и запись конфигурации типа 0

1 Возвращает чтение и запись статуса повтора запроса конфигурации.

mgmt_gen1_compatibility_mode_enable Режим совместимости с PCIe 1.1.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

mgmt_force_5g_deemphais_3_5db_6db_n Устанавливает значение величины коррекции предыскажений.

0 - 6dB

1 - 3.5dB

mgmt_force_5g_deemphais_en Принудительно устанавливает коррекцию предыскажений PHY приемника на скоростях 5g в соответствие с mgmt_force_5g_deemphais_3_5db_6db_n.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

selectable_deemphasis_for_5g_operation Устанавливает значение коррекции предыскажений по умолчанию для режима 5G во время определения состояния LTSSM.

0 - 3.5dB

1 - 6dB

direct_to_5gts Только для ядер PCIe Gen 2/3.

0 Разрешает удаленному устройству определять скорость канала.

1 При возможности устанавливает скорость канала 5Gbps.

support_5gts Advertise 5gt/s. Только для ядер PCIe Gen 2/3.

0 Предложение только 2.5gbps

1 Предложение поддержки для 5Gbps.

direct_to_8gts Только для ядер PCIe Gen 3.

0 Разрешает удаленному устройству определять скорость канала.

1 При возможности устанавливает скорость канала 8Gbps.

support_8gts Adversite 8gt/s. Только для ядер PCIe Gen 3.

0 не поддерживает 8gt/s

1 Предложение поддержки скорости передачи данных PCI Express 8gt/s, 5gt/s и 2.5 gt/s

inhibit_rx_idle Inhibit Receiver Electrical Idle: Устанавливается в 1, чтобы заставить все линии указать, что они находятся в свободном состоянии

inhibit_rx_det Inhibit Receiver Detection: Устанавливается в 1, чтобы блокировать определение приемника и предполагать, что на всех линиях есть удаленные приемники

switch_port_mode Switch Port Mode: Реализовать корневой порт как нисходящий порт коммутатора (режим порта коммутатора, режим корневого порта) ==

2'b10

pm_forcejrp_mode Устаревшая опция конфигурации, использовавшаяся NW Logic в более ранних функциональных моделях шины PCIe . Этот бит ДОЛЖЕН быть установлен в 0.

advertise_infinite_ch_cd_credits_asjroot_port Предоставить условно-бесконечный размер буферов для приема данных и заголовков (CD и CH). Если ядро сконфигурировано как оконечная точка, оно всегда устанавливает принудительное предоставление информации о размере буферов для приема данных и заголовков.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

root_port_mode Root Port Mode: Реализовать корневой порт как порт нисходящего коммутатора (режим порта коммутатора, режим корневого порта) == 2'b10

root_port_id_HI Root Port ID Это 16-битовое поле предназначено для определения ID, используемого для PCIe Requester ID и Completer ID, когда ядро работает в режиме нисходящего порта.

1.3.10.2.1.19.1.16 1.1.16 Регистр REG15 0x3c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..20	extended_capabilities_pointer	rw	сохраненное	сохранить	12	0h
19	extended_capabilities_enable	rw	сохраненное	сохранить	1	0h
18	disable_vc_tx_np_fifo	rw	сохраненное	сохранить	1	0h
17	mgmt_l2_d3hot_tx_beacon_enable	rw	сохраненное	сохранить	1	0h
16	modified_compliance_exit_direct_to_detect	rw	сохраненное	сохранить	1	0h
15.. 8	nfts_timeout_extend	rw	сохраненное	сохранить	8	7fh
7.. 5	force_tx_margin_value	rw	сохраненное	сохранить	3	0h
4	force_tx_margin	rw	сохраненное	сохранить	1	0h
3	зарезервировано	rw	сохраненное	сохранить	1	0h
2	clr_ep_on_ep_set	rw	сохраненное	сохранить	1	0h
1	set_ecrc_err_on_ep_set	rw	сохраненное	сохранить	1	0h
0	ignore_poison	rw	сохраненное	сохранить	1	1h

extended_capabilities_pointer Extended Capabilities Pointer. Если Expanded Capabilities Enable (разрешение расширенных настроек) == 1, то Expanded Capabilities Pointer указывает адрес пункта реализованного пользователем списка настроек.

extended_capabilities_enable Extended Capabilities Enable. (Разрешение расширенных настроек).

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

disable_vc_tx_np_fifo Для отладки; используется для того, чтобы запретить FIFO дополнительного канала, который позволяет трафику ответного пакета и транзакции, не требующей ответных пакетов, пропускать трафик транзакции с обязательным ожиданием ответного пакета (NP), который был остановлен из-за недоступных разрешений на пересылку транзакции с обязательным ожиданием ответного пакета или заголовка.

0 FIFO_ENABLED

1 ACTIVATE_DISABLE

mgmt_l2_d3hot_tx_beacon_enable При конфигурации оконечной точки и в состоянии канала L2 (D3hot), с включенным тактовым сигналом и питанием, определяет, должно ли ядро сигнализировать PHY о необходимости передачи сигнала готовности Beacon, получив указание выйти из L2.

0 Не сигнализировать о передаче сигнала готовности Beacon.

1 Сигнализировать PHY о необходимости передачи сигнала готовности Beacon.

modified_compliance_exit_direct_to_detect При окончании передачи модифицированной последовательности согласования и при Configuration Register Enter Compliance == 0, спецификация PCIe позволяет перейти либо в состояние обнаружения (Detect), либо в состояние активного опроса (Polling.Active). Этот бит позволяет пользователю решить, какая опция будет реализована.

0 Polling.Active

1 Detect

nfts_timeout_extend NFTS устанавливается для увеличения времени тайм-аута на; 0 == Использовать значение по умолчанию.

force_tx_margin_value Значение замены для tx_margin, разрешенное битом force_tx_margin.

force_tx_margin Принудительная установка диапазона рабочих напряжений передатчика (TX margin) в значение tx_margin_value.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

clr_ep_on_ep_set Только для тестов NW Logic.

0 Обычный режим работы

1 При установке очищает бит EP.

set_ecrc_err_on_ep_set Только для теста NW Logic.

0 Обычный режим работы

1 Устанавливает ошибку ECRC, если установлен бит EP

ignore_poison Игнорировать искажение - Устанавливается в 1, чтобы ядро игнорировало индикатор искажения” EP” для принятых пакетов TLP с содержимым, которое не отбрасывается в ядре. При установке в 1 ядро передаст все искаженные пакеты пользователю точно так же, как передает неискаженный пакет. Следует учесть, что бит игнорирования искажений принудительно устанавливается ы 1 ядром Expresso, когда ядро конфигурируется как корневой порт или коммутатор. Следует также учесть, что некоторые типы пакетов TLP игнорируют установку этого бита.

(1) Искаженные пакеты записи конфигурации типа 0 будут отбрасываться в ядре в любом случае (независимо от установки бита игнорирования искажений) при следующих условиях: генерируется завершение со статусом UR и генерируется соответствующее сообщение об

ошибке (ERR_COR [ANF] или ERR_FATAL), если оно не маскировано. Следует помнить, что пакеты записи конфигурации чтения 0 всегда обрабатываются так, как будто у них нет искажений: чтение завершается с установленным статусом успешного завершения и опциональным статусом Advisory Non-Fatal Error (предположительно нефатальная ошибка), если степень серьезности ошибки установлена как NON FATAL.

(2) Искаженные пакеты без содержимого будут в любом случае передаваться пользователю, поскольку индикатор "EP" не устанавливается для пакетов без содержимого, и эти пакеты обычно обрабатываются так, как будто в них нет искажений, или логика пользователя обрабатывает их как пакеты с предположительно нефатальной ошибкой.

(3) Определенные производителем искаженные сообщения типа 1 всегда передаются пользователю и, если бит mgmt_ignore_poison сброшен в 0, дополнительно устанавливается статус предположительной нефатальной ошибки), если степень серьезности ошибки установлена как NON FATAL.

Когда бит игнорирования искажений установлен в 0, ядро обрабатывает остальные искаженные пакеты TLP с содержимым следующим образом:

Искаженные запросы записи и завершения чтения с пакетами TLP с данными принимаются ядром и обрабатываются как ошибки искажения TLP, генерирующие соответствующее сообщение об ошибке искажения (ERR_NON_FATAL или ERR_FATAL в зависимости от значения регистра серьезности ошибки).

Искаженное сообщение с содержимым (отличное от типа 1, определенное производителем) принимается ядром и обрабатывается как ошибка искажения TLP, генерирующая соответствующее сообщение об ошибке искажения (ERR_NON_FATAL или ERR_FATAL в зависимости от значения регистра серьезности ошибки).

Рекомендуемые действия по умолчанию для оконечных точек с поддержкой функционала только ведомого устройства - установить Ignore Poison == 0 и указать пользовательской логике игнорировать бит заголовка” EP” у принимаемых сообщений. В этом случае пакеты TLP с содержимым (за исключением пакетов записи конфигурации типа 0 и определенных производителем сообщений типа 1) будут генерировать сообщение ERR_NON_FATAL и будут сбрасываться ядром. Искаженные сообщения без содержимого (для которых не применяется индикатор” EP”) будут обрабатываться так, как будто в них нет искажений.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

1.3.10.2.1.19.1.17 Регистр REG16 0x40

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	rx_detect_override	rw	сохраненное	сохранить	1	0h
30	rx_polarity_override	rw	сохраненное	сохранить	1	0h
29..12	tx_coef	rw	сохраненное	сохранить	18	0h
11.. 8	lptx_preset	rw	сохраненное	сохранить	4	0h
7	eq_coef_hardcoded	rw	сохраненное	сохранить	1	0h
6	eq_preset_hardcoded	rw	сохраненное	сохранить	1	0h
5.. 1	ds_link_number	rw	сохраненное	сохранить	5	0h
0	selfcrosslink	rw	сохраненное	сохранить	1	0h

rx_detect_override PHY RX Detect Override. Если разрешено, то результат обнаружения приемника на линии PHY логически складывается с маской mgmt_rx_detect_mask, прежде чем этот результат будет использован. Таким образом, линию можно принудительно заставить не обнаруживать приемник, когда соответствующий бит маски mgmt_rx_detect_mask[i] равен 0.

rx_polarity_override PHY RX Polarity Overrid. Если разрешено, принудительно устанавливает phy_rx_polarity в значение mgmt_rx_polarity[L-1:0].

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

tx_coef TX Coefficients. Используемые при скорости 8G коэффициенты выравнивания для всех линий передатчика локального PHY, когда mgmt_eq_coef_hardcoded == 1. В

mgmt_tx_coef должен быть установлен допустимый коэффициент для передатчика локального PHY, или определено поведение.

lptx_preset TX Preset. Предустановка выравнивания требуется от ответной части соединения, когда mgmt_eq_preset_hardcoded == 1. Разрешенная, эта предустановка используется для всех линий канала. В mgmt_tx_preset должна быть установлена допустимая предустановка, или определено поведение.

eq_coef_hardcoded Enables Local Tx Coefficients Hardcoded. При скорости 8G устанавливает для порта передатчика PHY phy_tx_deemph значения mgmt_tx_coef[17:0], независимо от коэффициентов/предустановок, запрошенных ответной частью соединения во время выравнивания.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

eq_preset_hardcoded Разрешает жестко заданную предустановку удаленного выравнивания передатчиков. Отменяет стандартный выбор коэффициентов выравнивания для 8G и указывает ядру использовать предустановленный метод выравнивания с только одним предустановленным значением mgmt_tx_preset[3:0].

1.3.10.2.1.19.1.18 Регистр REG16 0x40

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	rx_detect_override	rw	сохраненное	сохранить	1	0h
30	rx_polarity_override	rw	сохраненное	сохранить	1	0h
29..12	tx_coef	rw	сохраненное	сохранить	18	0h
11.. 8	lptx_preset	rw	сохраненное	сохранить	4	0h
7	eq_coef_hardcoded	rw	сохраненное	сохранить	1	0h
6	eq_preset_hardcoded	rw	сохраненное	сохранить	1	0h
5.. 1	ds_link_number	rw	сохраненное	сохранить	5	0h
0	selfcrosslink	rw	сохраненное	сохранить	1	0h

rx_detect_override PHY RX Detect Override. Если разрешено, то результат обнаружения приемника на линии PHY логически складывается с маской mgmt_rx_detect_mask, прежде чем этот результат будет использован. Таким образом, линию можно принудительно заставить не обнаруживать приемник, когда соответствующий бит маски mgmt_rx_detect_mask[i] равен 0.

rx_polarity_override PHY RX Polarity Overrid. Если разрешено, принудительно устанавливает phy_rx_polarity в значение mgmt_rx_polarity[L-1:0].

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

tx_coef TX Coefficients. Используемые при скорости 8G коэффициенты выравнивания для всех линий передатчика локального PHY, когда mgmt_eq_coef_hardcoded == 1. В mgmt_tx_coef должен быть установлен допустимый коэффициент для передатчика локального PHY, или определено поведение.

lptx_preset TX Preset. Предустановка выравнивания требуется от ответной части соединения, когда mgmt_eq_preset_hardcoded == 1. Разрешенная, эта предустановка используется для всех линий канала. В mgmt_tx_preset должна быть установлена допустимая предустановка, или определено поведение.

eq_coef_hardcoded Enables Local Tx Coefficients Hardcoded. При скорости 8G устанавливает для порта передатчика PHY phy_tx_deemph значения mgmt_tx_coef[17:0], независимо от коэффициентов/предустановок, запрошенных ответной частью соединения во время выравнивания.

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

eq_preset_hardcoded Разрешает жестко заданную предустановку удаленного выравнивания передатчиков. Отменяет стандартный выбор коэффициентов выравнивания для 8G и указывает ядру использовать предустановленный метод выравнивания с только одним предустановленным значением mgmt_tx_preset[3:0].

0 ЗАПРЕЩЕНО
1 РАЗРЕШЕНО

ds_link_number Downstream Port Link Number. Номер канала используется нисходящими портами в последовательностях TS при настройке канала. Рекомендуется нулевое значение, за исключением коммутаторов, которые могут реализовывать ненулевые значения, так чтобы каждый канал коммутатора имел отдельный номер.

self crosslink Разрешает внутреннюю петлю. Установка этого бита изменяет LTSSM настройку ядра, чтобы разрешить ядру устанавливать связь с самим собой в режиме внешней петли (последовательный передатчик с последовательным приемником) . Этот порт предназначен только для тестирования и не должен быть установлен в обычном режиме работы.

0 ЗАПРЕЩЕНО
1 РАЗРЕШЕНО

1.3.10.2.1.19.1.19 1.1.18 Регистр REG17 0x44

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	rx_detect_mask	rw	сохраненное	сохранить	16	0h
15.. 0	rx_polarity	rw	сохраненное	сохранить	16	0h

rx_detect_mask RX Detect Mask. Когда mgmt_rx_detect_override == 1, результаты обнаружения приемника линии PHY логически складываются с маской mgmt_rx_detect_mask, прежде чем использоваться. Таким образом, линию можно принудительно заставить не обнаруживать приемник, когда соответствующий бит маски mgmt_rx_detect_mask[i] равен 0.

rx_polarity Forced PHY RX Polarity. Значение, принудительно записанное в phy_rx_polarity, когда mgmt_rx_polarity.override == 1. Каждой линии (до 16 линий) соответствует один бит; значение 1 указывает, что линия данных должна быть инвертирована, - 0 - не должна быть инвертирована.

1.3.10.2.1.19.1.20 Регистр REG18 0x48

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	insert.link_err	rw	сохраненное	сохранить	1	0h
30	insert_malf_err	rw	сохраненное	сохранить	1	0h
29	rx_inhibit_tlp	rw	сохраненное	сохранить	1	0h
28	rx_inhibit_ack_nakjreception	rw	сохраненное	сохранить	1	0h
27	tx_parity_dl_errorjreport_ disable	rw	сохраненное	сохранить	1	0h
26	tx_parity_dl_error_handle_disable	rw	сохраненное	сохранить	1	0h
25	tx_parity_dl_error_inject_ enable	rw	сохраненное	сохранить	1	0h
24	tx_parity_tl_errorjreport.disable	rw	сохраненное	сохранить	1	0h
23	tx_parity_tl_ error_handle_disable	rw	сохраненное	сохранить	1	0h
22	tx_parity_tl_error_inject_ enable	rw	сохраненное	сохранить	1	0h
21	replay_ecc_ucor_error.report.disable	rw	сохраненное	сохранить	1	0h
20	replay_ecc_ucor_handle_disable	rw	сохраненное	сохранить	1	0h
19	replay_ecc_2bit_error_inject_enable	rw	сохраненное	сохранить	1	0h
18	replay_ecc_corrjreport_disable	rw	сохраненное	сохранить	1	0h
17	replay_ecc_corr_handle_disable	rw	сохраненное	сохранить	1	0h
16	replay_ecc_lbit_error_inject_ enable	rw	сохраненное	сохранить	1	0h
15	rx_dyn_credit_status	rw	сохраненное	сохранить	1	0h
14	force_tx_swing_low_at_8g	rw	сохраненное	сохранить	1	0h
13	tx_swing	rw	сохраненное	сохранить	1	0h
12..11	detect.quiet_min_time	rw	сохраненное	сохранить	2	0h
10.. 6	rx_idle_delay	rw	сохраненное	сохранить	5	0h
5	disable_2_5g_infer	rw	сохраненное	сохранить	1	1h
4	disable_5g_infer	rw	сохраненное	сохранить	1	1h
3	disable_8g_infer	rw	сохраненное	сохранить	1	1h
2.. 0	max_link_width_override	rw	сохраненное	сохранить	3	0h

rx_inhibit_ack_nakjreception (1) Устанавливается в 1 для одного такта, чтобы сбросить полученные пакеты DLLP типа ACK и NAK без указания буферу повтора повторить задержанные пакеты TLP в результате тайм-аута повтора; после начала повтора пакеты DLLP типа ACK и NAK обрабатываются как обычно; для обычного поведения Rx_Inhibit_Ack_Nak_Reception должен быть установлен в (0).

tx_parity_dl_error jreport_disable (1) запрещает сообщение о неисправляемых ошибках четности в интерфейсе

канального уровня/физического уровня; (0) разрешает сообщение (рекомендуемая установка по умолчанию)

tx_parity_dl_error_handle_disable (1) запрещает обнуление TLP, содержащих неисправляемые ошибки четности, обнаруженные в интерфейсе канального уровня/физического уровня; (0) разрешает обнуление (рекомендуемая установка по умолчанию для исполнений с большим количеством передаваемых разрешений на передачу виртуальных каналов)

tx_parity_dl_error_inject_enable Устанавливается в 1 для одного такта, чтобы вставить указание на неисправляемую ошибку четности в следующий пакет TLP, передаваемый по интерфейсу канального уровня/физического уровня (ошибки требующие коррекции номера последовательности Sequence Number); для обычной работы должен быть установлен в 0

tx_parity_tl_error jreport_disable (1) запрещает сообщение о не исправляемых ошибках четности, обнаруженных в интерфейсе уровня транзакций/канального уровня;

(0) разрешает сообщение (рекомендуемая установка по умолчанию)

tx_parity_tl_error_handle_disable (1) запрещает обнуление TLP, содержащих неисправляемые ошибки четности, обнаруженные в интерфейсе уровня транзакций/канального уровня; (0) разрешает обнуление (рекомендуемая установка по умолчанию для исполнений с большим количеством передаваемых разрешений на передачу виртуальных каналов)

tx_parity_tl_error_inject_enable Устанавливается в 1 для одного такта, чтобы вставить указание на неисправляемую ошибку четности в следующий пакет TLP, передаваемый по интерфейсу уровня транзакций/канального уровня; для обычной работы должен быть установлен в 0

eplay_ecc_ucor_error_report_disable (1) запрещает сообщение о не исправляемых ошибках ECC, обнаруженных в буфере повтора; (0) разрешает сообщение (рекомендуемая установка по умолчанию)

replay_ecc_ucor_handle_disable (1) запрещает обнуление TLP, содержащих неисправляемые ошибки ECC, обнаруженные в буфере повтора; (0) разрешает обнуление (рекомендуемая установка по умолчанию для исполнений с большим количеством передаваемых разрешений на передачу виртуальных каналов)

replay_ecc_2bit_error_inject_enable Устанавливается в 1 для одного такта, чтобы вставить указание на неисправляемую 2-битовую ошибку ECC в следующий пакет TLP, записываемый в буфер повтора; чтобы соответствующая ошибка была обнаружена, и о ней сообщено, буфер повтора должен повторять пакет(ы) TLP с ошибкой; для обычной работы должен быть установлен в 0

replay_ecc_corr_report_disable (1) запрещает сообщение об исправляемых ошибках ECC, обнаруженных в буфере повтора; (0) разрешает сообщение (рекомендуемое значение по умолчанию)

replay_ecc_corr_handle_disable (1) запрещает исправление 1-битовых ошибок ECC, обнаруженных в буфере повтора; (0) разрешает сообщение (рекомендуемое значение по умолчанию)

replay_ecc_1bit_error_inject_enable Устанавливается в 1 для одного такта, чтобы вставить указание на 1-битовую (исправляемую) ошибку ECC в следующий пакет TLP, записываемый в буфер повтора; чтобы соответствующая ошибка была обнаружена, и о ней сообщено, буфер повтора должен повторять пакет(ы) TLP с ошибкой; для обычной работы должен быть установлен в 0 rx_dyn_credit_status Разрешает сообщение статуса динамического разрешения на передачу для буфера приема в поле mgmt_pcie_status.core_ph.

force_tx_swing_low_at_8g Компенсация для PHY, которые должным образом не игнорируют tx_swing, tx_margin при скорости 8G. Принудительно устанавливает низкое значение phy_tx_swing, tx_margin при 8G.

0 Outputs_Per_PIPE_Spec

1 Override_Outputs_At_8G

tx_swing Непосредственно управляет выходным сигналом phy_tx_swing PIPE Ifc.

0 Full_Swing

1 Reduced_Swing

detect.quiet_min_time Минимально необходимое время в состоянии ожидания обнаружения (Detect.Quiet), прежде чем проверять состояние электрического простоя приемника PHY.

0 Нет

1 1ms

2 2ms

3 12ms

rx_idle_delay PHY RX Electrical Idle Dela. Задержка сигнала PHY phy_rx_elec_idle на mgmt_rx_idle_delay[4:0] (допустимый диапазон от 0 до16) тактов для улучшения согласования сигнала PHY phy_rx_elecJdle с соответствующими данными associated phy_rx_data.

disable_2_5g_infer Запрет вывода электрического простоя при 2.5G.

0 Цифровая фильтрация аналогового электрического простоя PHY при 2.5G

1 Запрет фильтрации. (предпочтительная установка по умолчанию)

disable_5g_infer Запрет вывода электрического простоя при 5G.

0 Цифровая фильтрация аналогового электрического простоя PHY при 5G

1 Запрет фильтрации. (предпочтительная установка по умолчанию)

disable_8g_infer Запрет вывода электрического простоя при 8G.

0 Цифровая фильтрация аналогового электрического простоя PHY при 8G

1 Запрет фильтрации. (предпочтительная установка по умолчанию)

max_link_width_override Перекрытие максимальной ширины канала. Эта установка, отличная от нуля, отменяет значение максимальной ширины канала в регистре функциональностей канала PCIe .

0 Ширина канала, определяемая ядром

1 1 линия

2 2 линии

3 4 линии

4 8 линий

5 16 линий

6 Зарезервировано6

7 Зарезервировано7

1.3.10.2.1.19.1.21 Регистр REG19 0x4c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	aspm_lls_tx_entry_time	rw	сохраненное	сохранить	16	0h
15..9	transaction_layer_error.control	rw	сохраненное	сохранить	7	0h
8	ctrl_parl_report_disable	rw	сохраненное	сохранить	1	0h
7	ctrl_parl_ handle_disable	rw	сохраненное	сохранить	1	0h
6	ctrl_parl_inject_en	rw	сохраненное	сохранить	1	0h
5	ctrl_ecc2_report.disable	rw	сохраненное	сохранить	1	0h
4	ctrl_ecc2_handle_disable	rw	сохраненное	сохранить	1	0h
3	ctrl_ ecc2_inject_ en	rw	сохраненное	сохранить	1	0h
2	ctrl_eccl_report_disable	rw	сохраненное	сохранить	1	0h
1	ctrl_eccl_ handle_disable	rw	сохраненное	сохранить	1	0h
0	ctrl_ eccL inject_ en	rw	сохраненное	сохранить	1	0h

aspm_l1s_tx_entry_time ASPM L1 Entry Time; количество микросекунд ожидания до запроса перехода в ASPM L1 (используется только восходящими портами - оконечной

точкой/восходящим коммутатором - Endpoint/Upstream Switch). Используется только при установленном L1s Power Mgmt. 0 - особый случай, устанавливает 1000 uS (0x3e8).

ctrl_ecc1_inject_en Запрещает исправление ECC для улучшения синхронизации

1.3.10.2.1.19.1.22 1.1.21 Регистр REG20 0x50

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..22	direct_to_loopback_slave_8g_coef _ LO	rw	сохраненное	сохранить	10	0h
21..18	direct_to_loopback_slave_8g_preset	rw	сохраненное	сохранить	4	0h
17	direct_to_loopback_slave_8g_coef _ en	rw	сохраненное	сохранить	1	0h
16	direct_to_loopback_slave_8g_preset_en	rw	сохраненное	сохранить	1	0h
15..13	direct_to_loopback_slave_8g_hint	rw	сохраненное	сохранить		0h
12	direct_to_loopback_error_countpreset	rw	сохраненное	сохранить	1	0h
11	direct_to_loopback_inject_rx_2bit_data_err	rw	сохраненное	сохранить	1	0h
10	direct_to_loopback_inject_rx_1bit_data_err	rw	сохраненное	сохранить	1	0h
9	direct_to_loopback_inject_rx_valid_err	rw	сохраненное	сохранить	1	0h
8	direct_to_loopback_inject_rx_skp_err	rw	сохраненное	сохранить	1	0h
7.. 5	direct_to_loopback_pattern	rw	сохраненное	сохранить		0h
4	direct_to_loopback_master_5g_deemph	rw	сохраненное	сохранить	1	0h
3	direct_to_loopback_slave_5g_deemph	rw	сохраненное	сохранить	1	0h
2	direct_to_loopback_tx_compliance_receive	rw	сохраненное	сохранить	1	0h
1.. 0	direct_to_loopback_speed	rw	сохраненное	сохранить	2	0h

direct_to_loopback_slave_8g_coef_LO Loopback Master: См. CFG_CONSTANTS_DIRECT_TO_LOOPBACK_SLAVE

direct_to_loopback_slave_8g_preset Loopback Master: См. CFG_CONSTANTS_DIRECT_TO_LOOPBACK_SLAVE

direct_to_loopback_slave_8g_coef_en Loopback Master: Если разрешено, то указывает передатчику ведомого устройства использовать коэффициент == direct_to_loopback_slave_coef при помощи последовательностей TS, передаваемых при направлении ведомого устройства к петле.

direct_to_loopback_slave_8g_preset_en Loopback Master: Если разрешено, то указывает передатчику ведомого устройства использовать предустановку == direct_to_loopback_slave_preset при помощи последовательностей TS, передаваемых при направлении ведомого устройства к петле.

direct_to_loopback_slave_8g_hint Loopback Master: В последовательностях EQ TS1 sets передается Rx Hint direct_to_loopback_error_count_reset Loopback Master: Если разрешено, то поддерживает на всех линиях счетчик ошибок ведущего устройства петли в состоянии сброса (0x0).

direct_to_loopback_inject_rx_2bit_data_err Loopback Master: Если разрешено в режиме петли, то вставляет две последовательных ошибки тактового сигнала в rx_data.

direct_to_loopback_inject_rx_1bit_data_err Loopback Master: Если разрешено в режиме петли, то вставляет одну ошибку тактового сигнала в rx_data.

direct.to_loopback_inject_rx_valid_err Loopback Master: Если разрешено в режиме петли, то вставляет одну ошибку тактового сигнала в rx_valid.

direct.to_loopback_inject_rx_skp_err Loopback Master: Если разрешено в режиме петли, то вставляет одну ошибку тактового сигнала в символ Skip приемника.

direct.to_loopback_pattern Loopback Master: Схема, используемая в режиме петли; 0==TS1 с декрементированием

канала/линии, остальное зарезервировано

direct.to_loopback_master_5g_deemph Loopback Master: Выбор значения коррекции предыскажений для использования ведущим устройством в режиме петли. При 5G

0 - 6dB

1 - 3.5dB

direct.to_loopback_slave_5g_deemph Loopback Master: Выбор значения коррекции предыскажений для направления ведомому устройству для использования в режиме петли. При 5G

0 - 6dB

1 - 3.5dB

direct_to_loopback_tx_compliancejreceive Loopback Master: Устанавливает получение согласования в передаваемых последовательностях TS

0 ЗАПРЕЩЕНО

1 РАЗРЕШЕНО

direct.to_loopback_speed Loopback Master: Скорость в канале для петли.

0 2.5G

1 5G

2 8G

1.3.10.2.1.19.1.23 Регистр REG21 0x54

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..14	direct.to_loopback_master_8g_coef	rw	сохраненное	сохранить	18	0h
13..10	direct_to_loopback_master_8g_preset	rw	сохраненное	сохранить	4	0h
9	direct_to_loopback_master_8g_coef_en	rw	сохраненное	сохранить	1	0h
8	direct.to_loopback_master_8g_preset_ en	rw	сохраненное	сохранить	1	0h
7.. 0	direct_to_loopback_slave_8g_coef_HI	rw	сохраненное	сохранить	8	0h

direct.to_loopback_master_8g_coef Loopback Master: См. CFG_CONSTANTS_DIRECT_TO_LOOPBACK_MASTER

direct.to_loopback_master_8g_preset Loopback Master: См. CFG_CONSTANTS_DIRECT_TO_LOOPBACK_MASTER'

direct.to_loopback_master_8g_coef_ en Loopback Master: Если разрешено, то указывает передатчику ядра использовать коэффициент == direct_to_loopback_master_coef при работе в качестве ведущего устройства в активной петле.

direct_to_loopback_master_8g_preset_en Loopback Master: Если разрешено, то указывает передатчику ядра использовать предустановку == direct_to_loopback_master_preset при работе в качестве ведущего устройства в активной петле.

direct_to_loopback_slave_8g_coef_HI Loopback Master: См. CFG_CONSTANTS_DIRECT_TO_LOOPBACK_SLAVE

1.3.10.2.1.19.1.24 Регистр REG22 0x58

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	зарезервировано	rw	сохраненное	сохранить	16	0h
15	tph_completer_enable	rw	сохраненное	сохранить	1	0h
14	ari_disable	rw	сохраненное	сохранить	1	0h
13	vsec_enable	rw	сохраненное	сохранить	1	0h
12.. 9	filter^rx_elec_idle_value	rw	сохраненное	сохранить	4	0h
8	vpd_enable	rw	сохраненное	сохранить	1	0h
7	dsn_enable	rw	сохраненное	сохранить	1	0h
6.. 1	tph_st_table_size	rw	сохраненное	сохранить	6	3fh
0	tph_enable	rw	сохраненное	сохранить	1	0h

tph_completer_enable TPH Completer Enable

ari_disable ARI Capability Disable; должно быть разрешено при разрешенном SR-IOV; должно быть запрещено для нисходящих портов, встроенных оконечных точек корневого комплекса и коллекторов событий корневого комплекса

0 ARI РАЗРЕШЕНО

1 ARI ЗАПРЕЩЕНО

vsec_enable VSEC Capability Enable

filter_rx_elec_idle_value Значение фильтра rx_elecJdle (шаг 128ns)

vpd_enable VPD Capability Enable

dsn_enable DSN Capability Enable
tph_st_table_size TPH ST Table Size (по умолчанию: 6’d63 == 64 записи)
tph_enable TPH Capability Enable

1.3.10.2.1.19.1.25 Регистр REG23 0x5c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	pwr_budget_sys_alloc	rw	сохраненное	сохранить	1	0h
30	pwr_budget_enable	rw	сохраненное	сохранить	1	0h
29	ltr_enable	rw	сохраненное	сохранить	1	0h
28	l1pmss_enable	rw	сохраненное	сохранить	1	0h
27	dpa_enable	rw	сохраненное	сохранить	1	0h
26..17	зарезервировано	rw	сохраненное	сохранить	10	0h
16	sris_enable	rw	сохраненное	сохранить	1	0h
15	rbar_enable	rw	сохраненное	сохранить	1	0h
14	ats_pasid_global_inval_support	rw	сохраненное	сохранить	1	1h
13.. 9	ats_inval_q_depth	rw	сохраненное	сохранить	5	0h
8	ats_enable	rw	сохраненное	сохранить	1	0h
7.. 3	pasid_max_width	rw	сохраненное	сохранить	5	14h
2	pasid_privilege_support	rw	сохраненное	сохранить	1	1h
1	pasid_execute_support	rw	сохраненное	сохранить	1	1h
0	pasid_enable	rw	сохраненное	сохранить	1	0h

pwr_budget_sys_alloc Power Budgeting System Allocated.
0 Распределение питания должно использовать значения Power Budgeting Capability
1 Распределение питания определяется системой
pwr_budget_enable Power Budgeting Capability Configuration Capability Enable.
ltr_enable Latency Tolerance Reporting (LTR) Enable.
l1pmss_enable L1 PM Substates Capability Enable.
dpa_enable Dynamic Power Allocation (DPA) Capability Enable.
sris_enable Разрешена SRIS
rbar.enable Resizable BAR Capability Enable
ats_pasid_global_inval_support ATS/PASID Global Invalidate Support. Если установлена, то функция поддерживает запросы денонсации с установленным битом Global Invalidate.
ats_inval_q_depth ATS Invalidate Queue Depth. Количество денонсированных запросов, которые могут быть поставлены в очередь в ядре. Устанавливается в 0, чтобы указать глубину очереди 32.
ats_enable ATS Capability Enable
pasid_max_width PASID Max ID Width
pasid_privilege_support PASID Privileged Mode Supported
pasid_execute_support PASID Execute Permission Supported
pasid_enable PASID Capability Enable

1.3.10.2.1.19.2 Устройство CFG_CONTROL

APB_MASTER CFG_CONTROL_BASE 0x200
Регистры конфигурации ядра влияют на основную работу ядра и должны быть статическими во время работы.

1.3.10.2.1.19.2.1 Регистр REG0 0x0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..22	aer_optional_error_en	rw	сохраненное	сохранить	10	0h
21..20	зарезервировано	rc	константа	сброс	2	0h
19..12	зарезервировано	rc	константа	сброс	8	0h
11..8	зарезервировано	rc	константа	сброс	4	0h
7..0	зарезервировано	rc	константа	сброс	8	0h

1.3.10.2.1.19.2.2 Регистр REG1 0x4

Поле	Наименование		Чтение	Запись	Разрядност	Сброс
31..29	зарезервировано	rc	константа	сброс	3	0h
28	pcie_dev_cap_disable_flr_capability	rw	сохранен	сохран	1	0h
27..12	зарезервировано	rc	константа	сброс	16	0h
11..9	pcie_dev_cap_endpoint_ll_acceptable_latency	rw	сохранен	сохран	3	0h
8.. 6	pcie_dev_cap_endpoint_l0s_acceptable_latency	rw	сохранен	сохран	3	0h
5	pcie_dev_cap_extended_tag_Поле_supported	rw	сохранен	сохран	1	1h
4.. 3	pcie_dev_cap_phantom_functions_supported	rw	сохранен	сохран	2	0h
2.. 0	pcie_dev_cap_max_payload_size_supported	rw	сохранен	сохран	3	2h

1.3.10.2.1.19.2.3 Регистр REG2 0x8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	pcie_link_cap_ll_exit_	rw	сохраненное	сохранить	1	1h
30..28	pcie_link_cap_l0s_exit.latency	rw	сохраненное	сохранить	3	7h
27..26	pcie_link_cap_aspm_support	rw	сохраненное	сохранить	2	1h
25..16	зарезервировано	rc	константа	сброс	10	0h
15.. 6	зарезервировано	rc	константа	сброс	10	0h
5	зарезервировано	rc	константа	сброс	1	0h
4	зарезервировано	rc	константа	сброс	1	0h
3.. 0	зарезервировано	rc	константа	сброс	4	0h

1.3.10.2.1.19.2.4 Регистр REG3 0xc

Поле	Наименование		Чтение	Запись	Разрядность	Сбро
31..29	зарезервировано	rc	константа	сброс	3	0h
28	pcie_link_stat_slot_clock_configuratio	r	сохраненное	сохрани	1	1h
27..16	зарезервировано	rc	константа	сброс	12	0h
15..8	pcie_link_cap_port.number	r	сохраненное	сохрани	8	0h
7.. 2	зарезервировано	rc	константа	сброс	6	0h
1..0	pcie_link_cap_l1_exit_latency_HI	r	сохраненное	сохрани	2	3h

1.3.10.2.1.19.2.5 Регистр REG4 0x10

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	зарезервировано	rc	константа	сброс	1	0h
30..29	pm_ctrl_stat_data_scale	rw	сохраненное	сохранить	2	0h
28..25	pm_ctrl_stat_data_select	rw	сохраненное	сохранить	4	0h
24..16	зарезервировано	rc	константа	сброс	9	0h
15..11	pm_cap_pme_support	rw	сохраненное	сохранить	5	0h
10	pm_cap_d2_support	rw	сохраненное	сохранить	1	1h
9	pm_cap_d1.support	rw	сохраненное	сохранить	1	1h
8.. 6	pm_cap_aux_current	rw	сохраненное	сохранить	3	0h
5	pm_cap_dsi	rw	сохраненное	сохранить	1	0h
4	зарезервировано	rc	константа	сброс	1	0h
3	pm_cap_pme_clock	rw	сохраненное	сохранить	1	0h
2.. 0	pm_cap.version	rw	сохраненное	сохранить	3	3h

1.3.10.2.1.19.2.6 Регистр REG5 0x14

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..27	pcie_cap_interrupt_message_number	rw	сохраненное	сохранить	5	0h
26	pcie_cap_slot_implemented	rw	сохраненное	сохранить	1	0h
25..22	pcie_cap_device_port_type	rw	сохраненное	сохранить	4	0h
21..18	pcie_cap_capability_version	rw	сохраненное	сохранить	4	2h
17	root_port jread_completion_boundary	rw	сохраненное	сохранить	1	0h
16	requires_aux_power	rw	сохраненное	сохранить	1	0h
15.. 8	pm_data_register.control	rw	сохраненное	сохранить	8	0h
7	pm_pmcsr_bse_bus_power_clk_ctrl_en	rw	сохраненное	сохранить	1	0h
6	pm_pmcsr_bse_b2_b3_support_for_d3hot	rw	сохраненное	сохранить	1	0h
5.. 0	зарезервировано	rc	константа	сброс	6	0h

1.3.10.2.1.19.2.7 Регистр REG560x18

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
21	pcie_dev_cap2_end_end_prefixes_supported	rw	сохраненное	сохранить	1	0h
20..5	зарезервировано	rc	константа	сброс	16	0h
4	pcie_dev_cap2_cpl_timeout_disable_supported	rw	сохраненное	сохранить	1	1h
3..0	pcie_dev_cap2_cpl_timeout_ranges_supported	rw	сохраненное	сохранить	4	0h

1.3.10.2.1.19.2.8 Регистр REG7 0x1c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..0	зарезервировано	rc	константа	сброс	32	0h

1.3.10.2.1.19.2.9 Регистр REG8 0x20

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..0	зарезервировано_0	rc	константа	сброс	32	0h

1.3.10.2.1.19.2.10 Регистр REG9 0x2

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_1	rc	константа	сброс	32	0h

1.3.10.2.1.19.2.11 Регистр REG10 0x28

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_2	rc	константа	сброс	32	0h

1.3.10.2.1.19.2.12 Регистр REG11 0x2c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_3	rc	константа	сброс	32	0h

1.3.10.2.1.19.2.13 Регистр REG12 0x30

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..0	зарезервировано	rc	константа	сброс	32	0h

1.3.10.2.1.19.2.14 Регистр REG13 0x34

Поле	Наименование	Чтение	Запись	Разрядность	Сброс	
31..21	msi_x_table_size	rw	сохраненное	сохранить	11	1fh
20..18	msi_x_table_size	rw	сохраненное	сохранить	3	1fh
20..17	msi_x_table_size	rw	сохраненное	сохранить	1	1fh
20..16	msi_x_table_size	rw	сохраненное	сохранить	1	1fh
20..15	msi_x_table_size	rw	сохраненное	сохранить	1	1fh
14..1	зарезервировано	rc	константа	сброс	14	0h
0	aer_version_0x2_enable	rw	сохраненное	сохранить	1	0h

1.3.10.2.1.19.2.15 Регистр REG14 0x38

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 3	msi_x_table_offset	rw	сохраненное	сохранить	29	3000h
2.. 0	msi_x_table_bir	rw	сохраненное	сохранить	3	0h

1.3.10.2.1.19.2.16 Регистр REG15 0x3c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 3	msi_x_pba_offset	rw	сохраненное	сохранить	29	3000h
2.. 0	msi_x_pba_bir	rw	сохраненное	сохранить	3	0h

1.3.10.2.1.19.2.17 Регистр REG16 0x40

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31 .19	pcie_slot_ cap_physical_ slot_number	rw	сохраненное	сохранить	13	1h
18	pcie_slot_cap_no_command_completed_support	rw	сохраненное	сохранить	1	0h
17	pcie_slot_cap .electromechanical _interlock_present	rw	сохраненное	сохранить	1	0h
16 .15	pcie_slot_cap_slot_power_limit_ scale	rw	сохраненное	сохранить	2	0h
14 .7	pcie_slot_cap_slot_power_limit_value	rw	сохраненное	сохранить	8	ah
6	pcie_slot_cap_hot_plug_capable	rw	сохраненное	сохранить	1	0h
5	pcie_slot_cap_hot_plug_surprise	rw	сохраненное	сохранить	1	0h
4	pcie_slot_cap_power_indicator_present	rw	сохраненное	сохранить	1	0h
3	pcie_slot_ cap_attention_indicator_present	rw	сохраненное	сохранить	1	0h
2	pcie_slot_ ca^_mrL sensor_present	rw	сохраненное	сохранить	1	0h
1	pcie_slot_cap_power_controller_present	rw	сохраненное	сохранить	1	0h
0	pcie_slot_ cap_attention_button_present	rw	сохраненное	сохранить	1	0h

1.3.10.2.1.19.2.18 Регистр REG17 0x44

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31 .16	vendor_specific_extended_capability_vsec_id	rw	сохраненное	сохранить	16	1h
15.. 8	зарезервировано	rc	константа	сброс	8	0h
7	зарезервировано	rc	константа	сброс	1	0h
6	зарезервировано	rc	константа	сброс	1	0h
5	зарезервировано	rc	константа	сброс	1	0h
4	зарезервировано	rc	константа	сброс	1	0h
3..2	зарезервировано	rc	константа	сброс	2	0h
1	зарезервировано	rc	константа	сброс	1	0h
0	зарезервировано	rc	константа	сброс	1	0h

1.3.10.2.1.19.2.19 Регистр REG18 0x48

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31 .16	vsec_id	rw	сохраненное	сохранить	16	0h
15.. 14	зарезервировано	rc	константа	сброс	2	0h
13..8	vsec_input_bits	rw	сохраненное	сохранить	6	8h
7..0	status_func_num	rw	сохраненное	сохранить	8	0h

1.3.10.2.1.19.2.20 Регистр REG19 0x4c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dsn_serial_number_LO	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.21 Регистр REG20 0x50

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dsn_serial_number_HI	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.22 Регистр REG21 0x54

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..29	зарезервировано	rc	константа	сброс	3	0h
28..24	rbar_default_0	rw	сохраненное	сохранить	5	0h
23.. 4	rbar_sizes_0	rw	сохраненное	сохранить	20	fh
3	зарезервировано	rc	константа	сброс	1	0h
2.. 0	rbar_bar_0	rw	сохраненное	сохранить	3	0h

1.3.10.2.1.19.2.23 Регистр REG22 0x58

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..29	зарезервировано	rc	константа	сброс	3	0h
28..24	rbar_default_1	rw	сохраненное	сохранить	5	0h
23.. 4	rbar_sizes_1	rw	сохраненное	сохранить	20	0h
3	зарезервировано	rc	константа	сброс	1	0h
2.. 0	rbar_bar_1	rw	сохраненное	сохранить	3	1h

1.3.10.2.1.19.2.24 Регистр REG23 0x5c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..29	зарезервировано	rc	константа	сброс	3	0h
28..24	rbar_default_2	rw	сохраненное	сохранить	5	0h
23.. 4	rbar_sizes_2	rw	сохраненное	сохранить	20	0h
3	зарезервировано	rc	константа	сброс	1	0h
2.. 0	rbar_bar_2	rw	сохраненное	сохранить	3	2h

1.3.10.2.1.19.2.25 Регистр REG24 0x60

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..29	зарезервировано	rc	константа	сброс	3	0h
28..24	rbar_default_3	rw	сохраненное	сохранить	5	0h
23.. 4	rbar_sizes_3	rw	сохраненное	сохранить	20	0h
3	зарезервировано	rc	константа	сброс	1	0h
2.. 0	rbar_bar_3	rw	сохраненное	сохранить	3	3h

1.3.10.2.1.19.2.26 Регистр REG25 0x64

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..29	зарезервировано	rc	константа	сброс	3	0h
28..24	rbar_default_4	rw	сохраненное	сохранить	5	0h
23.. 4	rbar_sizes_4	rw	сохраненное	сохранить	20	0h
3	зарезервировано	rc	константа	сброс	1	0h
2.. 0	rbar_bar_4	rw	сохраненное	сохранить	3	4h

1.3.10.2.1.19.2.27 Регистр REG26 0x68

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..29	зарезервировано	rc	константа	сброс	3	0h
28..24	rbar_default_5	rw	сохраненное	сохранить	5	0h
23.. 4	rbar_sizes_5	rw	сохраненное	сохранить	20	0h
3	зарезервировано	rc	константа	сброс	1	0h
2.. 0	rbar_bar_5	rw	сохраненное	сохранить	3	5h

1.3.10.2.1.19.2.28 Регистр REG27 0x6c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..24	l1pmss_cm_restor_ time	rw	сохраненное	сохранить	8	0h
23..19	l1pmss_port_tpower_on_value	rw	сохраненное	сохранить	5	0h
18	зарезервировано	rc	константа	сброс	1	0h
17..16	l1pmss_port_tpower_on_scale	rw	сохраненное	сохранить	2	0h
15..8	l1pmss_port_cm_restore_time	rw	сохраненное	сохранить	8	0h
7.. 5	зарезервировано	rc	константа	сброс	3	0h
4	l1pmss_l1pm_supported	rw	сохраненное	сохранить	1	1h
3	l1pmss_aspm_l_1_supported	rw	сохраненное	сохранить	1	1h
2	l1pmss_aspm_l1_2_supported	rw	сохраненное	сохранить	1	1h
1	l1pmss_ pcipm_l1_1_supported	rw	сохраненное	сохранить	1	1h
0	l1pmss_pcipm_l1_2_supported	rw	сохраненное	сохранить	1	1h

1.3.10.2.1.19.2.29 Регистр REG28 0x70

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..24	dpa^xlcyl	rw	сохраненное	сохранить	8	0h
23..16	dpa_xlcy0	rw	сохраненное	сохранить	8	0h
15..14	зарезервировано	rc	константа	сброс	2	0h
13..12	dpa_pas	rw	сохраненное	сохранить	2	0h
11..10	зарезервировано	rc	константа	сброс	2	0h
9.. 8	dpa_tlunit	rw	сохраненное	сохранить	2	0h
7.. 5	зарезервировано	rc	константа	сброс	3	0h
4.. 0	dpa_substate_max	rw	сохраненное	сохранить	5	0h

1.3.10.2.1.19.2.30 Регистр REG29 0x74

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_xlcy_indicator	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.31 Регистр REG30 0x78

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_alloc_array_0	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.32 Регистр REG31 0x7c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_alloc_array_1	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.33 Регистр REG32 0x80

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_alloc_array_2	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.34 Регистр REG33 0x84

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_alloc_array_3	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.35 Регистр REG34 0x88

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_alloc_array_4	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.36 Регистр REG35 0x8c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_alloc_array_5	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.37 Регистр REG36 0x90

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_alloc_array_6	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.2.38 REG37 Register 0x94

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	dpa_alloc_array_7	rw	сохраненное	сохранить	32	0h

1.3.10.2.1.19.3 Устройство RP-STATUS

APB_MASTER KP_STATUS_BASE 0x300

Регистры конфигурации ядра влияют на основную работу ядра и должны быть статическими во время работы.

1.3.10.2.1.19.3.1 Регистр REG0 0x0

Поле	Наименование		Чтение	Запись	Разрядно сти	Сброс
31..28	зарезервировано	rc	константа	сброс	4	0h
27	info_tlp_received	rs	статус	сброс	1	0h
26	pm_jrx_pme_to_ack_msg	rs	статус	сброс	1	1h
25	pm^rx_pme_turn_off _msg	rs	статус	сброс	1	0h
24	зарезервировано	rc	константа	сброс	1	0h
23	зарезервировано	rc	константа	сброс	1	0h
22	зарезервировано	rc	константа	сброс	1	0h
21	зарезервировано	rc	константа	сброс	1	0h
20	зарезервировано	rc	константа	сброс	1	0h
19	rp_intd	rs	статус	сброс	1	0h
18	rp_intc	rs	статус	сброс	1	0h
17	rp_intb	rs	статус	сброс	1	0h
16	rp_inta	rs	статус	сброс	1	0h
15	зарезервировано	rc	константа	сброс	1	0h
14	зарезервировано	rc	константа	сброс	1	0h
13	зарезервировано	rc	константа	сброс	1	0h
12	rp_err_fat	rs	статус	сброс	1	0h
11	rp_err_nfat	rs	статус	сброс	1	0h
10	rp_err_cor	rs	статус	сброс	1	0h
9	rp_set_pcie_unsupportedjreq	rs	статус	сброс	1	0h
8	rp_set_pcie_fatal_err	rs	статус	сброс	1	0h
7	rp_set_pcie_non_fatal_err	rs	статус	сброс	1	0h
6	rp_set_pcie_correctable_err	rs	статус	сброс	1	0h
5	rp_set_detected_parity_error	rs	статус	сброс	1	0h
4	rp_set_signaled_system_error	rs	статус	сброс	1	0h
3	rp_setjreceived_master_abort	rs	статус	сброс	1	0h
2	rp_setjreceived_target_abort	rs	статус	сброс	1	0h
1	rp_set_signaled_target_abort	rs	статус	сброс	1	0h
0	rp_set_master_data_parity_error	rs	статус	сброс	1	0h

1.3.10.2.1.19.3.2 Регистр REG1 0x4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 6	зарезервировано	rc	константа	сброс	26	0h
5	rp_sec_detected_parity_error	rs	статус	сброс	1	0h
4	rp_sec_received_system_error	rs	статус	сброс	1	0h
3	rp_sec_received_master_abort	rs	статус	сброс	1	0h
2	rp_sec_received_target_abort	rs	статус	сброс	1	0h
1	rp_sec_signaled_target_abort	rs	статус	сброс	1	0h
0	rp_sec_master_data_parity_error	rs	статус	сброс	1	0h

1.3.10.2.1.19.3.3

1.3.10.2.1.19.3.4 Регистр REG2 0x8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_barO_LO	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.5 Регистр REG3 0xc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_bar0_HI	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.6 Регистр REG4 0x10

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_barl_LO	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.7 Регистр REG5 0x14

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_barl_HI	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.8 Регистр REG6 0x18

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_exp_rom	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.9 Регистр REG7 0x1c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_bar0_mask_L0	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.10 Регистр REG8 0x20

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_bar0_mask_HI	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.11 Регистр REG9 0x24

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_barl_mask_L0	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.12 Регистр REG10 0x28

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_barl_mask_HI	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.13 Регистр REG11 0x2c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	cfg_expjrom_mask	rs	статус	сброс	32	0h

1.3.10.2.1.19.3.14 Регистр REG12 0x30

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 1	зарезервировано_0	rc	константа	сброс	31	0h
0	direct_ds_to_hot_reset	rs	статус	сброс	1	0h

1.3.10.2.1.19.3.15 Регистр REG13 0x34

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_1	rc	константа	сброс	32	0h

1.3.10.2.1.19.3.16 Регистр REG14 0x38

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_2	rc	константа	сброс	32	0h

1.3.10.2.1.19.3.17 Регистр REG15 0x3c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_3	rc	константа	сброс	32	0h

1.3.10.2.1.19.4 Устройство CFGSTATUS

1.3.10.2.1.19.4.1 APB_MASTER CFG_STATUS_BASE 0x400

Этот набор регистров предоставляет отладочные доступ только для чтения к локальным PCIe Configuration Space Byte Offsets 0x00-0xFF. Номер функции устанавливается в поле cfg_control_status_func_num

1.3.10.2.1.19.4.2 Регистр REG0 0x0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x00	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.3 Регистр REG1 0x4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x01	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.4 Регистр REG2 0x8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x02	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.5 Регистр REG3 0xc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x03	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.6 Регистр REG4 0x10

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x04	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.7 Регистр REG5 0x14

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x05	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.8 1.4.7 Регистр REG6 0x18

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x06	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.9 Регистр REG7 0x1c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x07	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.10 Регистр REG8 0x20

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x08	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.11 Регистр REG9 0x24

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x09	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.12 Регистр REG10 0x28

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x0a	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.13 Регистр REG11 0x2c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x0b	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.14 Регистр REG12 0x30

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x0c	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.15 Регистр REG13 0x34

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x0d	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.16 Регистр REG14 0x38

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x0e	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.17 Регистр REG15 0x3c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x0f	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.18 Регистр REG16 0x40

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x10	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.19 Регистр REG17 0x44

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x11	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.20 Регистр REG18 0x48

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x12	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.21 Регистр REG19 0x4c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x13	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.22 Регистр REG20 0x50

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x14	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.23 Регистр REG21 0x54

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xl5	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.24 Регистр REG22 0x58

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xl6	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.25 Регистр REG23 0x5c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xl7	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.26 Регистр REG24 0x60

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xl8	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.27 Регистр REG25 0x64

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xl9	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.28 Регистр REG26 0x68

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xla	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.29 Регистр REG27 0x6c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xlb	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.30 Регистр REG28 0x70

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xlc	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.31 Регистр REG29 0x74

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xld	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.32 Регистр REG30 0x78

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xle	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.33 Регистр REG31 0x7c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0xlf	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.34 Регистр REG32 0x80

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x20	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.35 Регистр REG33 0x84

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x21	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.36 Регистр REG34 0x88

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x22	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.37 Регистр REG35 0x8c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x23	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.38 Регистр REG36 0x90

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x24	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.39 Регистр REG37 0x94

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x25	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.40 Регистр REG38 0x98

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x26	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.41 Регистр REG39 0x9c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x27	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.42 Регистр REG40 0xa0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x28	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.43 Регистр REG41 0xa4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x29	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.44 Регистр REG42 0xa8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x2a	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.45 Регистр REG41 0xac

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x2b	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.46 Регистр REG44 0xb0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x2c	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.47 Регистр REG45 0xb4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x2d	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.48 Регистр REG46 0xb8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x2e	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.49 Регистр REG47 0xbc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x2f	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.50 Регистр REG48 0xc0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x30	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.51 Регистр REG49 0xc4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x3l	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.52 Регистр REG50 0xc8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x32	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.53 Регистр REG51 0xcc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x33	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.54 Регистр REG52 0xd0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x34	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.55 Регистр REG53 0xd4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x35	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.56 Регистр REG54 0xd8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x36	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.57 Регистр REG55 0xdc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x37	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.58 Регистр REG56 0xe0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x38	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.59 Регистр REG57 0xe4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x39	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.60 Регистр REG58 0xe8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x3a	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.61 Регистр REG59 0xec

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x3b	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.62 Регистр REG60 0xf0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x3c	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.63 Регистр REG61 0xf4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x3d	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.64 Регистр REG62 0xf8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x3e	rs	статус	сброс	32	0h

1.3.10.2.1.19.4.65 Регистр REG63 0xfc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x3f	rs	статус	сброс	32	0h

1.3.10.2.1.19.5 Устройство CFGESTATUS

1.3.10.2.1.19.6 *APB_MASTER CFG_ESTATUS_BASE 0x500*

Этот набор регистров предоставляет отладочные доступ только для чтения к локальным PCIe Configuration Space Byte Offsets 0x100-0x1FF. Номер функции устанавливается в поле cfg_controlj3tatus_func_num.

1.3.10.2.1.19.6.1 Регистр REG0 0x0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x40	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.2 Регистр REG1 0x4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..0	config_space_dword_0x4l	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.3 Регистр REG2 0x8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x42	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.4 Регистр REG3 0xc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x43	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.5 Регистр REG4 0x10

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x44	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.6 Регистр REG5 0x14

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x45	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.7 Регистр REG6 0x18

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x46	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.8 Регистр REG7 0x1c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x47	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.9 Регистр REG8 0x20

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x48	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.10 Регистр REG9 0x24

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x49	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.11 Регистр REG10 0x28

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..0	config_space_dword_0x4a	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.12 Регистр REG11 0x2c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x4b	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.13 Регистр REG12 0x30

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x4c	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.14 Регистр REG13 0x34

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x4d	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.15 Регистр REG14 0x38

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x4e	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.16 Регистр REG15 0x3c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x4f	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.17 Регистр REG16 0x40

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x50	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.18 Регистр REG17 0x44

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x51	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.19 Регистр REG18 0x48

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x52	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.20 Регистр REG19 0x4c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x53	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.21 Регистр REG20 0x50

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x54	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.22 Регистр REG21 0x54

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x55	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.23 Регистр REG22 0x58

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x56	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.24 Регистр REG23 0x5c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x57	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.25 Регистр REG24 0x60

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x58	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.26 Регистр REG25 0x64

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x59	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.27 Регистр REG26 0x68

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x5a	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.28 Регистр REG27 0x6c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x5b	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.29 Регистр REG28 0x70

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x5c	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.30 Регистр REG29 0x74

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x5d	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.31 Регистр REG30 0x78

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x5e	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.32 Регистр REG31 0x7c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x5f	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.33 Регистр REG32 0x80

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x60	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.34 Регистр REG33 0x84

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x6l	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.35 Регистр REG34 0x88

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x62	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.36 Регистр REG35 0x8c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x63	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.37 Регистр REG36 0x90

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x64	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.38 Регистр REG37 0x94

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x65	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.39 Регистр REG38 0x98

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x66	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.40 Регистр REG39 0x9c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x67	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.41 Регистр REG40 0xa0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x68	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.42 Регистр REG41 0xa4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x69	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.43 Регистр REG42 0xa8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x6a	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.44 Регистр REG43 0xac

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x6b	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.45 Регистр REG44 0xb0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x6c	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.46 Регистр REG45 0xb4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x6d	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.47 Регистр REG46 0xb8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x6e	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.48 Регистр REG47 0xbc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x6f	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.49 Регистр REG48 0xc0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x70	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.50 Регистр REG49 0xc4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x7l	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.51 Регистр REG50 0xc8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x72	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.52 Регистр REG51 0xcc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x73	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.53 Регистр REG52 0xd0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x74	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.54 Регистр REG53 0xd4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x75	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.55 Регистр REG54 0xd8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x76	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.56 Регистр REG55 0xdc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x77	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.57 Регистр REG56 0xe0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x78	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.58 Регистр REG57 0xe4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x79	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.59 Регистр REG58 0xe8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x7a	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.60 Регистр REG59 0xec

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x7b	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.61 Регистр REG60 0xf0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x7c	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.62 Регистр REG61 0xf4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x7d	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.63 Регистр REG62 0xf8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x7e	rs	статус	сброс	32	0h

1.3.10.2.1.19.6.64 Регистр REG63 0xfc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	config_space_dword_0x7f	rs	статус	сброс	32	0h

1.3.10.2.1.19.7 Устройство PCIE_STATUS

1.3.10.2.1.19.7.1 APB_MASTER PCIE_STATUS_BASE 0x600

Регистры конфигурации ядра влияют на основную работу ядра и должны быть статическими во время работы.

1.3.10.2.1.19.7.2 Регистр REG0 0x0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..26	cfg_neg_link_width	rs	статус	сброс	6	0h
25..24	link_speed	rs	статус	сброс	2	0h
23..12	зарезервировано	rs	константа	сброс	12	0h
11..8	lane_lock_status	rs	статус	сброс	4	0h
7..2	mgmt_lts_state	rs	статус	сброс	6	0h
1	mgmt_dl_link_up	rs	статус	сброс	1	0h
0	mgmt_pl_ link_up	rs	статус	сброс	1	0h

1.3.10.2.1.19.7.3 Регистр REG1 0x4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..28	зарезервировано	rc	константа	сброс	4	0h
27	mgmt_tx_err_fat	rs	статус	сброс	1	0h
26	mgmt_tx_err_nfat	rs	статус	сброс	1	0h
25	mgmt_tx_err_cor	rs	статус	сброс	1	0h
24	err_ucorr_internal_error	rs	статус	сброс	1	0h
23	err_ur_error	rs	статус	сброс	1	0h
22	er_ecr_error	rs	статус	сброс	1	0h
21	errjreceiver_overflow	rs	статус	сброс	1	0h
20	err_unexpected_completion	rs	статус	сброс	1	0h
19	err_completer_abort	rs	статус	сброс	1	0h
18	err_completion_timeout	rs	статус	сброс	1	0h
17	err_fc_protocol_error	rs	статус	сброс	1	0h
16	err_poisoned_tlp	rs	статус	сброс	1	0h
15	err_surprise_down_error	rs	статус	сброс	1	0h
14	err_header_log_overflow	rs	статус	сброс	1	0h
13	err_corrected_internal_error	rs	статус	сброс	1	0h
12	err_advisory_non_fatal_error	rs	статус	сброс	1	0h
11	зарезервировано	rc	константа	сброс	1	0h

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
10	info_schedule_dupl_ack	rs	статус	сброс	1	0h
9	info_nakreceived	rs	статус	сброс	1	0h
8	err_dl_protocol_error	rs	статус	сброс	1	0h
7	err jreplay_num_rollover	rs	статус	сброс	1	0h
6	errjreplay_timer_timeout	rs	статус	сброс	1	0h
5	err_malformed_tlp	rs	статус	сброс	1	0h
4	info_bad_tlp_crc_n_err	rs	статус	сброс	1	0h
3	info_bad_tlp_seq_err	rs	статус	сброс	1	0h
2	info_bad_tlp_crc_err	rs	статус	сброс	1	0h
1	err_bad_tlp	rs	статус	сброс	1	0h
0	err_bad_dllp	rs	статус	сброс	1	0h

1.3.10.2.1.19.7.4 Регистр REG2 0x8

Поле	Наименование		Чтение	Запись	Разрядно	Сброс
31..27	зарезервировано	rc	константа	сброс	5	0h
26	info_vc^rx_assertions_8	rs	статус	сброс	1	0h
25	info_vc_rx_assertions_7	rs	статус	сброс	1	0h
24	info_vc_rx_assertions_6	rs	статус	сброс	1	0h
23	info_vc_rx_assertions_5	rs	статус	сброс	1	0h
22	info_vc_rx_assertions_4	rs	статус	сброс	1	0h
21	info_vc_rx_assertions_3	rs	статус	сброс	1	0h
20	info_vc_rx_assertions_2	rs	статус	сброс	1	0h
19	info_vc_rx_assertions_1	rs	статус	сброс	1	0h
18	info_vc_rx_assertions_0	rs	статус	сброс	1	0h
17	зарезервировано	rc	константа	сброс	1	0h
16	info_vc_tx_assertions_8	rs	статус	сброс	1	0h
15	info_vc_tx_assertions_7	rs	статус	сброс	1	0h
14	info_vc_tx_assertions_6	rs	статус	сброс	1	0h
13	info_vc_tx_assertions_5	rs	статус	сброс	1	0h
12	info_vc_tx_assertions_4	rs	статус	сброс	1	0h
11	info_vc_tx_assertions_3	rs	статус	сброс	1	0h
10	info_vc_tx_assertions_2	rs	статус	сброс	1	0h
9	info_vc_tx_assertions_1	rs	статус	сброс	1	0h
8	info_vc_tx_assertions_0	rs	статус	сброс	1	0h
7	зарезервировано	rc	константа	сброс	1	0h
6	зарезервировано	rc	константа	сброс	1	0h
5	mgmt_speed_change_fail	rs	статус	сброс	1	0h
4	err_receiver_error	rs	статус	сброс	1	0h
3	зарезервировано	rc	константа	сброс	1	0h
2	зарезервировано	rc	константа	сброс	1	0h
1	зарезервировано	rc	константа	сброс	1	0h
0	зарезервировано	rc	константа	сброс	1	0h

1.3.10.2.1.19.7.5 Регистр REG3 0xc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	зарезервировано	rc	константа	сброс	8	0h
15..8	зарезервировано	rc	константа	сброс	1	0h
7	зарезервировано	rc	константа	сброс	1	0h
6	зарезервировано	rc	константа	сброс	1	0h
5	infojrx_poisoned_tlp	rs	статус	сброс	1	0h
4	info_tx_poisoned_tlp	rs	статус	сброс	1	0h
3	infojrx_unsupported_request	rs	статус	сброс	1	0h
2	infojrx_completer_abort	rs	статус	сброс	1	0h
1	info_tx_unsupported_request	rs	статус	сброс	1	0h
0	info_tx_completer_abort	rs	статус	сброс	1	0h

1.3.10.2.1.19.7.6 Регистр REG4 0x10

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..24	remote_pd_LO	rs	статус	сброс	8	0h
23..16	remote_ch	rs	статус	сброс	8	0h
15.. 8	remote_nh	rs	статус	сброс	8	0h
7.. 0	remote_ph	rs	статус	сброс	8	0h

1.3.10.2.1.19.7.7 Регистр REG5 0x14

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..28	core_ph_LO	rs	статус	сброс	4	не определено
27..16	remote_cd	rs	статус	сброс	12	0h
15.. 4	remote_nd	rs	статус	сброс	12	0h
3.. 0	remote_pd_HI	rs	статус	сброс	4	0h

1.3.10.2.1.19.7.8 Регистр REG6 0x18

Поле	Наименование	Чтение	Запись	Разрядность	Сброс	
31..20	core_pd	rs	статус	сброс	12	не определено
19..12	core_ch	rs	статус	сброс	8	не определено
11.. 4	core_nh	rs	статус	сброс	8	не определено
3.. 0	core_ph_HI	rs	статус	сброс	4	не определено

1.3.10.2.1.19.7.9 Регистр REG7 0x1c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31	remote_credit_block	rs	статус	сброс	1	0h
30..24	зарезервировано	rc	константа	сброс	7	0h
23..12	core_cd	rs	статус	сброс	12	не определено
11.. 0	core_nd	rs	статус	сброс	12	не определено

1.3.10.2.1.19.7.10 Регистр REG8 0x20

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_lane_status_0	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.11 Регистр REG9 0x24

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_lane_status_1	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.12 Регистр REG10 0x28

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_lane_status_2	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.13 Регистр REG11 0x2c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_lane_status_3	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.14 Регистр REG12 0x30

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_0	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.15 Регистр REG13 0x34

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_1	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.16 Регистр REG14 0x38

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_2	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.17 Регистр REG15 0x3c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_3	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.18 Регистр REG16 0x40

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_0	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.19 Регистр REG17 0x44

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_1	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.20 Регистр REG18 0x48

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_2	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.21 Регистр REG19 0x4c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_3	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.22 Регистр REG20 0x50

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_4	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.23 Регистр REG21 0x54

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_5	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.24 Регистр REG22 0x58

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_6	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.25 Регистр REG23 0x5c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_7	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.26 Регистр REG24 0x60

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_pl_tx_data_0	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.27 Регистр REG25 0x64

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_pl_tx_data_1	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.28 Регистр REG26 0x68

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_pl_tx_data_2	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.29 Регистр REG27 0x6c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_pl_tx_data_3	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.30 Регистр REG28 0x70

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..30	cfg_aspm_control	rs	статус	сброс	2	0h
29..28	mgmt_power_state	rs	статус	сброс	2	0h
27..23	mgmt_pm_state	rs	статус	сброс	5	не определено
22..16	clk_period_in_ns	rs	статус	сброс	7	не определено
15.. 0	mgmt_pl_tx_en	rs	статус	сброс	16	0h

1.3.10.2.1.19.7.31 Регистр REG29 0x74

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	mgmt_lts_sub_state	rs	статус	сброс	16	не определено
15..12	mgmt_lane^rev_status	rs	статус	сброс	4	0h
11.. 3	mgmt_configured_link_num	rs	статус	сброс	9	1ffh
2.. 0	mgmt_ltsjrx_l0s_state	rs	статус	сброс	3	0h

1.3.10.2.1.19.7.32 Регистр REG30 0x78

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	mgmt_link_used_lanes	rs	статус	сброс	16	0h
15.. 0	mgmt_rx_detect_lanes	rs	статус	сброс	16	0h

1.3.10.2.1.19.7.33 Регистр REG31 0x7c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..22	зарезервировано	rc	константа	сброс	10	0h
21	mgmt_el_idle_inferred	rs	статус	сброс	1	0h
20	mgmt_loopback_cfg_entry	rs	статус	сброс	1	0h
19.. 4	mgmt_loopback_sync	rs	статус	сброс	16	0h
3	mgmt_crosslink_ds_us_n	rs	статус	сброс	1	0h
2	mgmt_crosslink_active	rs	статус	сброс	1	0h
1	зарезервировано	rc	константа	сброс	1	0h
0	mgmt_speed_change_fail	rs	статус	сброс	1	0h

1.3.10.2.1.19.7.34 Регистр REG32 0x80

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_pl_rx_data_0	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.35 Регистр REG33 0x84

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_pl_rx_data_1	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.36 Регистр REG34 0x88

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_pl_rx_data_2	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.37 Регистр REG35 0x8c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_pl_rx_data_3	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.38 Регистр REG36 0x90

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31..16	зарезервировано_0	rc	константа	сброс	16	0h
15.. 0	mgm^pLr^en	rs	статус	сброс	16	0h

1.3.10.2.1.19.7.39 Регистр REG37 0x94

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_1	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.40 Регистр REG38 0x98

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_2	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.41 Регистр REG39 0x9c

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	зарезервировано_3	rc	константа	сброс	32	0h

1.3.10.2.1.19.7.42 Регистр REG40 0xa0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_loopback_err_count_0	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.43 Регистр REG41 0xa4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_loopback_err_count_1	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.44 Регистр REG42 0xa8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_loopback_err_count_2	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.45 Регистр REG43 0xac

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_loopback_err_count_3	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.46 Регистр REG44 0xb0

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_loopback_err_count_4	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.47 Регистр REG45 0xb4

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_loopback_err_count_5	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.48 Регистр REG46 0xb8

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_loopback_err_count_6	rs	статус	сброс	32	0h

1.3.10.2.1.19.7.49 Регистр REG47 0xbc

Поле	Наименование		Чтение	Запись	Разрядность	Сброс
31.. 0	mgmt_loopback_err_count_7	rs	статус	сброс	32	0h

1.3.10.2.2 Контроллер интерфейса ARINC-818

1.3.10.2.2.1 Общие сведения

Блок “Контроллер интерфейса ARINC-818” предназначен для подключения системы к внешней коммутационной среде, функционирующей в соответствии со спецификацией ARINC-818.

Блок взаимодействует с системой посредством интерфейса AMBA AXI.

Блок взаимодействует с внешней коммутационной средой посредством интерфейса с внешним сериализатором-десериализатором (SERDES).

Управление блоком осуществляется посредством интерфейса AMBA APB.

Структурная схема блока представлена на рисунке.

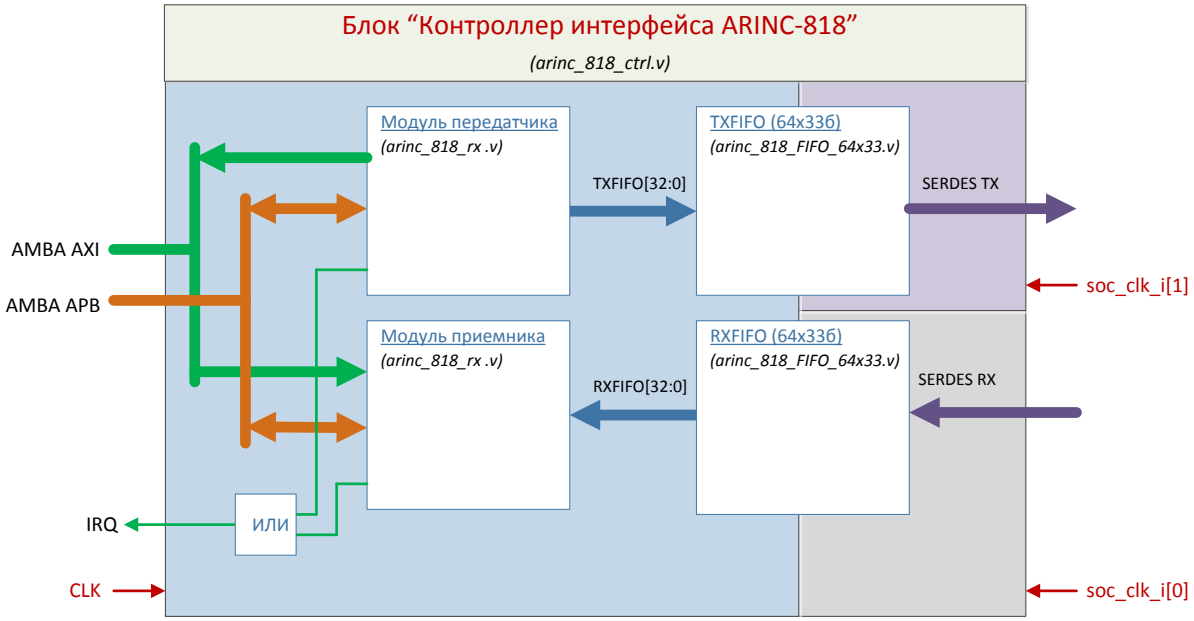


Рисунок 1.146 - Структурная схема Блока “Контроллер интерфейса ARINC-818”

Блок состоит из двух независимых трактов: приемного и передающего.

Передающий тракт включает Модуль передатчика и буфер типа FIFO.

Приемный тракт включает Модуль приемника и буфер типа FIFO.

Передающий тракт взаимодействует с шиной AXI только по чтению, в то время как приемный тракт взаимодействует с шиной AXI только по записи. Таким образом приемный и передающий тракты могут быть подключены к разным абонентам AXI.

Управляющий интерфейс AMBA APB является общим для приемного и передающего тракта.

Прерывания от приемного и передающего трактов объединены по схеме логического “ИЛИ” для формирования сигнала прерывания Блока.

Передающий (SERDES TX) и приемный (SERDES RX) интерфейсы с блоком SERDES являются независимыми и могут быть подключены к разным блокам SERDES.

Блок использует 3 сигнала синхронизации:

CLK – системный тактовый сигнал;

soc_clk_i[0] – сигнал синхронизации выходных данных (RX DATA) блока SERDES;

soc_clk_i[1] – сигнал синхронизации входных данных (TX DATA) блока SERDES.

Области действия данных сигналов синхронизации условно обозначены на рисунке в виде цветовых зон.

Блок также использует сигнал синхронизации шины AMBA APB для операций чтения/записи конфигурационных и информационных регистров. Данный сигнал синхронизации рекомендуется объединять с сигналом CLK.

Перечень внешних сигналов Блока представлен в таблице.

Таблица 1.779 - Список внешних сигналов контроллер интерфейса ARINC-818

Наименование	Направление	Примечание
Системные сигналы		
CLK	IN	Системный тактовый сигнал Также тактирует интерфейс AXI
nRST	IN	Асинхронный сброс. Активный 0 Объединен с PRESETn_i
IRQ_o	OUT	Прерывание блока
Управляющий интерфейс AMBA APB		
PCLK_i	IN	Рекомендуется объединять с CLK
PRESETn_i	IN	Объединен с nRST
PADDR_i[31:0]	IN	
PSEL_i	IN	
PENABLE_i	IN	
PWRITE_i	IN	
PRDATA_o[31:0]	OUT	
PWDATA_i[31:0]	IN	
PREADY_o	OUT	Всегда равно 0x1
PSLVERR_o	OUT	Всегда равно 0x0
AXI Master Read Address		
ARID_o[1:0]	OUT	
ARADDR_o[31:0]	OUT	
ARLEN_o[3:0]	OUT	
ARSIZE_o[2:0]	OUT	
ARBURST_o[1:0]	OUT	
ARLOCK_o[1:0]	OUT	
ARCACHE_o[3:0]	OUT	
ARPROT_o[2:0]	OUT	
ARVALID_o	OUT	
ARREADY_i	IN	
AXI Master Read Data		
RID_i[1:0]	IN	

Продолжение таблицы1.780

Наименование	Направление	Примечание
RDATA_i[63:0]	IN	
RRESP_i[1:0]	IN	
RLAST_i	IN	
RVALID_i	IN	
RREADY_o	OUT	
AXI Master Write Address		
AWID_o[1:0]	OUT	
AWADDR_o[31:0]	OUT	
AWLEN_o[3:0]	OUT	
AWSIZE_o[2:0]	OUT	
AWBURST_o[1:0]	OUT	
AWVALID_o	OUT	
AWREADY_i	IN	
AXI Master Write Data		
WID_o[1:0]	OUT	
WDATA_o[63:0]	OUT	
WSTRB_o[7:0]	OUT	
WLAST_o	OUT	
WVALID_o	OUT	
WREADY_i	IN	
AXI Master Write Response		
BID_i[1:0]	IN	
BRESP_i[1:0]	IN	
BVALID_i	IN	
BREADY_o	OUT	
SERDES TX		
serdes_pipe_txdatak_o[3:0]	OUT	[3] – метка К-символа выходных данных [2:0] – не используются, равны 0x0
serdes_pipe_txdata_o[31:0]	OUT	Шина входных данных SERDES
soc_clk_i[1]	IN	сигнал синхронизации входных данных (TX DATA) блока SERDES
SERDES RX		
serdes_pipe_rxdatak_i[3:0]	IN	[3:1] – не используется [0] – метка К-символа входных данных

Продолжение таблицы1.781

Наименование	Направление	Примечание
serdes_pipe_rxdata_i[31:0]	IN	Шина выходных данных SERDES
serdes_pipe_rxstatus_i[2:0]	IN	Шина состояния SERDES. Не используется.
serdes_pipe_rxvalid_i	IN	Строб валидности входных данных.
soc_clk_i[0]	IN	сигнал синхронизации выходных данных (RX DATA) блока SERDES

Перечень параметров синтеза Блока представлен в таблице.

Наименование	Значение по умолчанию	Описание
MEMTECH	0	Тип используемой памяти: 0 – симуляция 1 – ASIC 2 - FPGA
AXI_AWBURST	1	Значение сигнала AWBURST_o[1:0]
TRANSMITTER_ENABLE	1	1 - Включить передатчик в ядро
RECEIVER_ENABLE	1	1 - Включить приемник в ядро
RECEIVER_DUMMY	0	1 – Модуль приемника работает только в режиме RAW Data.
RX_SUPPORT_24BIT	1	1 – Поддержка приведения входных видеоданных к формату RGB 8:8:8
TRANSMITTER_DUMMY	0	1 – Модуль передатчика работает только в режиме RAW Data.
TX_SUPPORT_24BIT	1	1 – Поддержка выходных видеоданных формата RGB 8:8:8
TX_OVERLAY_ENABLE	1	1 - Включить функции совмещения в модуль передатчика
RX_HISTOGRAM_ENABLE	1	1 - Включить функции сбора гистограммы яркости в модуль приемника
RX_GAMMA_CORRECTION_ENABLE	1	1 - Включить функции коррекции яркости в модуль приемника
RX_Monochrome_SUPPORT	1	Поддержка приемником Монохромного режима
RX_RGB_SUPPORT	1	Поддержка приемником режима RGB
RX_RGGB_SUPPORT	1	Поддержка приемником режима RGGB
RX_RGBA_SUPPORT	1	Поддержка приемником режима RGBA
RX_YIQ_SUPPORT	1	Поддержка приемником режима YIQ
RX_Packed_RGB_SUPPORT	1	Поддержка приемником режима Packed RGB
RX_Packed_RGGB_SUPPORT	1	Поддержка приемником режима Packed RGGB
RX_Packed_RGBA_SUPPORT	1	Поддержка приемником режима Packed RGBA
RX_Packed_YIQ_SUPPORT	1	Поддержка приемником режима Packed YIQ
RX_PTN_8bit_SUPPORT	1	Поддержка приемником упаковки пикселов в 8-битные компоненты
RX_PTN_10bit_SUPPORT	1	Поддержка приемником упаковки пикселов в 10-битные компоненты

Наименование	Значение по умолчанию	Описание
RX_PTN_12bit_SUPPORT	1	Поддержка приемником упаковки пикселов в 12-битные компоненты
RX_PTN_16bit_SUPPORT	1	Поддержка приемником упаковки пикселов в 16-битные компоненты
RX_PTN_20bit_SUPPORT	1	Поддержка приемником упаковки пикселов в 20-битные компоненты
RX_PTN_24bit_SUPPORT	1	Поддержка приемником упаковки пикселов в 24-битные компоненты
RX_PTN_32bit_SUPPORT	1	Поддержка приемником упаковки пикселов в 32-битные компоненты

1.3.10.2.2.2 Модуль передатчика

1.3.10.2.2.2.1 Общие сведения

Модуль передатчика осуществляет загрузку видеоданных из ОЗУ посредством интерфейса AXI, формирование из них ADVB-контейнеров в соответствии со спецификацией ARINC-818 и их выдачу.

Основные характеристики модуля передатчика представлены в таблице.

Характеристика	Значение
Поддерживаемые типы Объектов	Объект 0; Объект 2.
Выходные форматы данных	Raw Data – “сырые данные”, данные из ОЗУ без изменения и обработки; Монохромный (CI = 0x0), 8 битов; Упакованный RGB (CI = 0xB), 16 битов – 5:6:5; Упакованный RGB (CI = 0xB), 24 бита – 8:8:8.
Максимальный размер выходного кадра	2047x2047 пикселов
Поддержка внешней палитры	Нет
Поддержка параметров Объекта 0	Нет
Тип выходной синхронизации	D1 (попиксельно, разбивка по видеострокам)

Структурная схема модуля передатчика представлена на рисунке 1.147.

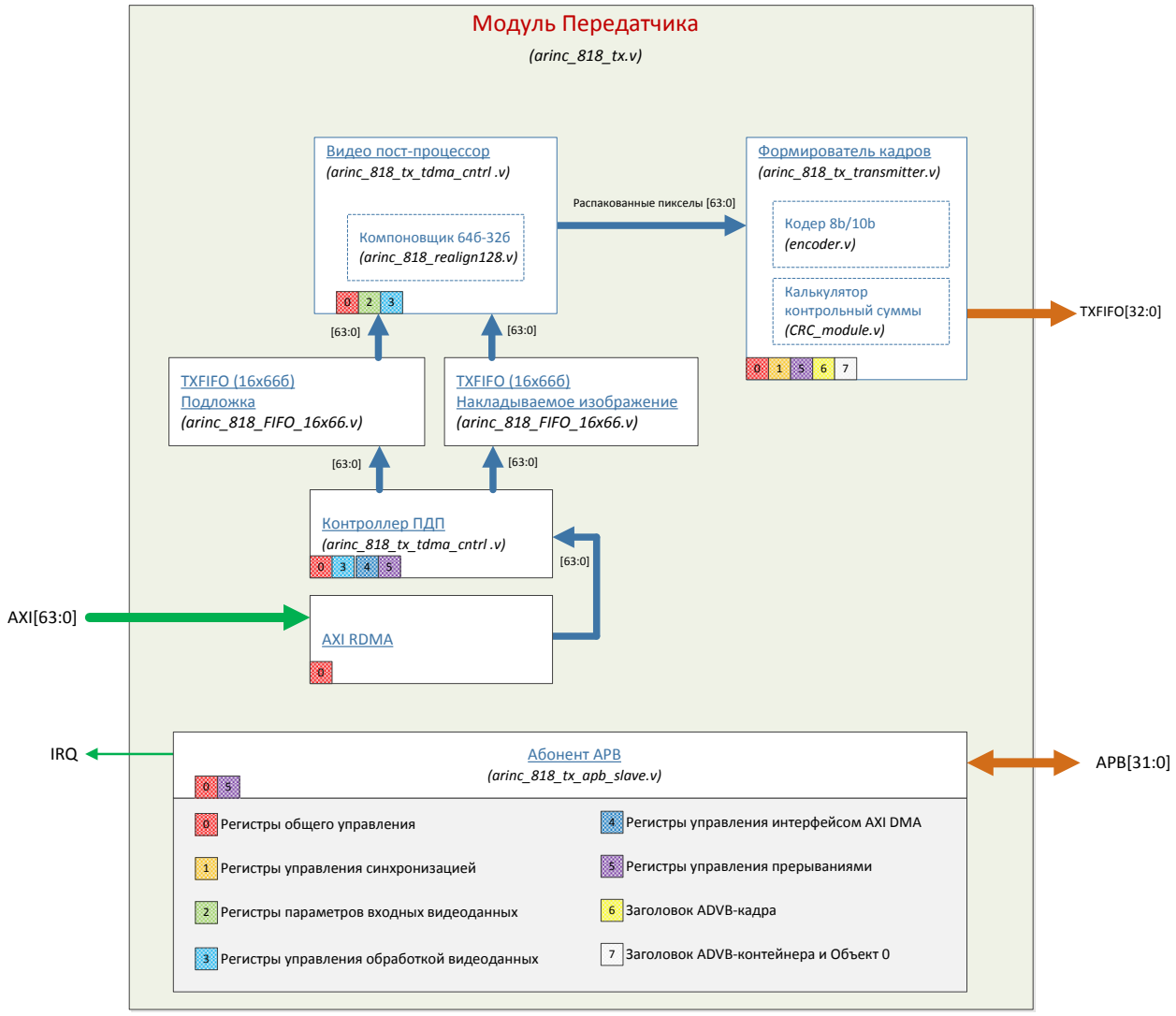


Рисунок 1.147 - Структурная схема модуля передатчика

1.3.10.2.2.2.1.1 Контроллер ПДП

“Контроллер ПДП" (Прямого Доступа к Памяти) обеспечивает взаимодействие с примененным ядром AXI RDMA, которое в свою очередь обеспечивает обмен по шине AXI.

Для формирования выходного изображения система должна зарезервировать в оперативной памяти два видеобуфера одинакового размера для хранения двух последовательных видеокадров. “Контроллер ПДП” поочередно загружает видеокадры из данных видеобуферов и передает их в Модуль “Видео пост-процессор”.

Загруженные из ВОЗУ видеоданные размещаются в FIFO буфере “Подложка”.

При использовании функции совмещения система должна дополнительно зарезервировать ещё два видеобуфера для хранения двух последовательных видеокадров накладываемого изображения. Размер данных видеобуферов должен быть аналогичен размеру видеобуферов в которых хранится подложка.

Если в регистре TX_VIDEO_PROCESSING_ENS активирована функция совмещения, контроллер ПДП дополнительно загружает второе, накладываемое изображение и размещает его в FIFO-буфере “Накладываемое изображение”. Загрузка накладываемого изображения из видеобуферов происходит аналогично загрузке подложки – поочередно из двух соответствующих видеобуферов.

Регистры управления интерфейсом AXI DMA задают параметры размещения изображения в видеопамяти.

“Контроллер ПДП” обеспечивает генерацию прерываний по началу и по окончанию чтения очередного видеокадра из видеопамяти. Регистры управления прерываниями содержат вектора прерываний, а также логику управления прерываниями.

1.3.10.2.2.2.1.2 Видео пост-процессор

Модуль “Видео пост-процессор” осуществляет совмещение двух изображений, представляющее собой суммирование их соответствующих цветовых компонент.

Сформированное изображение поступает в под-модуль “Компоновщик 64б-32б”, представляющий собой буфер типа FIFO на 4 128-разрядных слова с байтовым доступом и обеспечивающий приведение потока данных к разрядности шины SERDES.

Скомпонованное таким образом изображение поступает на вход модуля “Формирователь кадров”.

1.3.10.2.2.2.1.3 Формирователь кадров

“Формирователь кадров” осуществляет следующий набор действий:

упаковку изображения в ADVB-контейнер;

сегментацию ADVB-контейнера на ADVB-кадры по правилам, соответствующим типу синхронизации D1 в соответствии со спецификацией ARINC-818.

расчет контрольных сумм ADVB-кадров;

“Формирователь кадров” обеспечивает генерацию прерываний по началу и по окончанию выдачи очередного ADVB-контейнера. Регистры управления прерываниями содержат вектора прерываний, а также логику управления прерываниями.

Шаблоны заголовков ADVB-кадра, ADVB-контейнера, а также полей объекта 0 хранятся в регистрах заголовка ADVB-кадра, ADVB-контейнера и Объекта 0. Ряд полей заголовков и Объекта 0 формируются автоматически в процессе формирования кадра. Данные поля доступны в регистрах только для чтения.

Регистры управления синхронизацией задают временные параметры выдачи ADVB-контейнера.

1.3.10.2.2.2.1.4 Абонент APB

“Абонент APB” обеспечивает взаимодействие модуля передатчика с управляющим процессором по интерфейсу APB.

Также в “Абоненте APB” находится логика формирования внешнего прерывания модуля передатчика.

1.3.10.2.2.2.2 Карта регистров передатчика

Адрес	Наименование	Описание	Доступ
0x0 - 0x7F		Резерв	
Регистры общего управления			
0x80	TX_IP_CONTROL	Регистр общего управления	Чт/Зп
0x84	TX_SERDES_CONTROL	Регистр управления SERDES	Чт/Зп
0x88 - 0x27F		Резерв	
Регистры управления синхронизацией			
0x280	TX_SYNC_TLT	Регистр длительности видеостроки	Чт
0x284	TX_SYNC_MIN_TFT	Регистр минимальной длительности видеокадра	Чт

Адрес	Наименование	Описание	Доступ
0x288	TX_SYNC_T0	Регистр интервала между Объектом 0 и Объектом 2 ADVB контейнера	Чт
0x28C	TX_SYNC_T3_T4	Регистр интервалов между кадрами внутри ADVB контейнера	Чт/Зп
0x290	TX_SYNC_T5	Регистр интервалов между ADVB контейнерами	Чт/Зп
0x294	TX_SYNC_MIN_REQ_TFT	Регистр длительности кадра	Чт/Зп
0x298 - 0x37F		Резерв	
Регистры параметров входных видеоданных			
0x380	TX_COLOR_CONVERSION_OUTFORMAT	Регистр режима цветности изображения в памяти	Чт/Зп
0x384 - 0x3FF		Резерв	
Регистры управления обработкой видеоданных			
0x400	TX_VIDEO_PROCESSING_ENS	Регистр активации функций видеообработки	Чт/Зп
0x404 - 0x47F		Резерв	
Регистры управления интерфейсом AXI DMA			
0x480	TX_ADMA_MNG	Регистр управления состояния AXI RDMA	Чт
0x484	TX_ADMA_BASEADDR_IM0_SIDE0	Регистр базового адреса буфера 0	Чт/Зп
0x488	TX_ADMA_BASEADDR_IM0_SIDE1	Регистр базового адреса буфера 1	Чт/Зп
0x48C	TX_ADMA_BASEADDR_IM1_SIDE0	Регистр базового адреса буфера 0 второго изображения при совмещении	Чт/Зп
0x490	TX_ADMA_BASEADDR_IM1_SIDE1	Регистр базового адреса буфера 1 второго изображения при совмещении	Чт/Зп
0x494	TX_ADMA_BUFFER_STRING_SIZE	Размер видеостроки буфера	Чт/Зп
0x498	TX_RAW_ZERO_FRAME_SIZE	Размер 0 фрейма в режиме RAW data	Чт/Зп
0x49C	TX_RAW_QTY	Количество фреймов в режиме RAW data (без учета 0 фрейма)	Чт/Зп
0x4A0	TX_RAW_VIDEO_LINE_SEG12_SIZE	Размер сегментов 0/1 фреймов в режиме RAW data	Чт/Зп
0x4A4	TX_RAW_VIDEO_LINE_SEG34_SIZE	Размер сегментов 2/3 фреймов в режиме RAW data	Чт/Зп
0x4A8	TX_RAW_CONTROL	Регистр общего управления режимом RAW data	Чт/Зп
0x4AC - 0x4FF		Резерв	
Регистры управления прерываниями			
0x500	TX_IRQ_MNG	Регистр управления доступом	Чт/Зп
0x504 - 0x50F		Резерв	
0x510	TX_IRQ_REQ	Регистр запросов прерываний	Чт
0x514	TX_IRQ_MASK	Регистр масок прерываний	Чт/Зп

Адрес	Наименование	Описание	Доступ
0x518	TX_IRQ_VECTOR	Регистр вектора прерываний	Чт
0x51C	TX_IRQ_COLLISION	Регистр вектора коллизий	Чт
0x520	TX_IRQ_SET	Регистр программной установки прерываний	Зп
0x524	TX_IRQ_RESET	Регистр программного сброса прерываний	Зп
0x528 - 0x5FF		Резерв	
Регистры заголовка ADVB кадра			
0x600 : 0x614	TX_F_HEADER_WORD0 : TX_F_HEADER_WORD5	Заголовок ADVB кадра	Чт/Зп
0x618 - 0x6FF		Резерв	
Регистры заголовка ADVB контейнера и Объекта 0			
0x700 : 0x754	TX_C_HEADER_WORD0 : TX_C_HEADER_WORD21	Заголовок ADVB контейнера	Чт/Зп
0x758 : 0x764	TX_OBJECT0_WORD0 : TX_OBJECT0_WORD3	Объект 0	Чт/Зп

1.3.10.2.2.2.3 Базовая настройка передатчика

Для начала выдачи видеоданных достаточно сделать следующее:

Сформировать видеокадры в памяти.

Задать базовые адреса буферов видеокадров в регистрах TX_ADMA_BASEADDR_IM0_SIDE0 и TX_ADMA_BASEADDR_IM0_SIDE1

Задать размер видеостроки в видеобуфере в регистре TX_ADMA_BUFFER_STRING_SIZE. Размер видеостроки в видеобуфере должен быть не менее размера видеостроки изображения. Если используется формат изображения по умолчанию, то в регистр необходимо записать значение не менее 320.

Задать Формат изображения посредством регистра TX_COLOR_CONVERSION_OUTFORMAT и полей “Number of Rows” и “Number of Columns” объекта 0.

Записать значение 0x1 в регистр TX_IP_CONTROL.

После этого начнется выдача видеокадров 320x200x8бит с частотой 25 Гц (при скорости 1GFC).

Частота формирования видеоданных определяется содержимым регистра TX_SYNC_MIN_REQ_TFT.

1.3.10.2.2.2.4 Регистры общего управления

1.3.10.2.2.2.4.1 TX_IP_CONTROL (0x80)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:2			Резерв	
1	tx_soft_reset	0x1	Программный сброс 0x0: Программный сброс завершен 0x1: Программный сброс в процессе Автоматически принимает значение 0x0 по завершении процедуры сброса	Чт/Зп
0	tx_on	0x0	Включение передатчика 0x0: Передатчик выключен 0x1: Передатчик включен	Чт/Зп

1.3.10.2.2.2.4.2 TX_SERDES_CONTROL (0x84)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:2			Резерв	
1	tx_serdes_k_swap_en	0x0	Положение метки K-символа в шине serdes_pipe_txdatak_o[3:0] 0x0: Метка в бите 3 0x1: Метка в бите 0	Чт/Зп
0	tx_serdes_swap_en	0x0	Порядок следования байт в шине serdes_pipe_txdata_o[31:0] 0x0: big-endian 0x1: little-endian Предполагается, что K-символы всегда располагаются в старшем байте шины	Чт/Зп

1.3.10.2.2.2.5 Регистры управления синхронизацией

Расчетные значения временных интервалов определяются в зависимости от заданных геометрии и режима цветности видеокадра в памяти, а также от заданных временных параметров выходного формирователя.

Единицей измерения временных интервалов является величина TLW (Long Word Time), равная времени, необходимому для передачи 40 битов данных по последовательному каналу (длительность одного 32-разрядного слова до кодировки 8b/10b). В частности для скорости 1GFC TLW = 26.5625нс.

1.3.10.2.2.2.5.1 TX_SYNC_TLT (0x280)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:27			Резерв	
26:16	tx_TLT_IDLE	—	Расчетная суммарная длительность (количество) всех последовательностей IDLE внутри интервала tx_TLT	Чт
15:11			Резерв	
10:0	tx_TLT	—	Расчетная длительность одной видеостроки. Включает в себя длительность всех ADVB кадров в одной видеостроке, сумму интервалов между ними и интервал до следующей видеостроки	Чт

Величина tx_TLT_IDLE позволяет определить сколько TLW от общей длительности видеостроки приходится на последовательности заполнения IDLE. Данный параметр нужен для точной подстройки частоты формирования видеокадров путем варьирования различных временных параметров выходного формирователя.

1.3.10.2.2.2.5.2 TX_SYNC_MIN_TFT (0x284)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:23			Резерв	
22:0	tx_min_TFT	—	Расчетная минимально возможная длительность видеокадра. Включает в себя длительность всех ADVB кадров видеокадра, интервалы между ними, а также заданный минимальный интервал между видеокадрами	Чт

1.3.10.2.2.2.5.3 TX_SYNC_T0 (0x288)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:13			Резерв	
12:0	tx_t0	—	Расчетный интервал между Объектом 0 и Объектом 2 ADVB контейнера. Включает в себя длительность нулевого ADVB кадра и интервал между нулевым и первым ADVB кадрами	Чт

1.3.10.2.2.2.5.4 TX_SYNC_T3_T4 (0x28C)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:24			Резерв	
23:16	tx_t4	0xB (11)	Интервал (количество последовательностей IDLE) между ADVB кадрами разных видеострок При попытке записать значение менее 6 – запись не происходит	Чт/Зп
15:4			Резерв	
3:0	tx_t3	0xB (11)	Интервал (количество последовательностей IDLE) между ADVB кадрами внутри одной видеостроки При попытке записать значение менее 6 – запись не происходит	Чт/Зп

1.3.10.2.2.2.5.5 TX_SYNC_T5 (0x290)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:18			Резерв	
17:0	tx_t5	0x1F4 (500)	Минимальный интервал (количество последовательностей IDLE) между видеокадрами При попытке записать значение менее 500 – запись не происходит	Чт/Зп

1.3.10.2.2.2.5.6 TX_SYNC_MIN_REQ_TFT (0x294)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:23			Резерв	
22:0	tx_min_req_TFT	0x103664 (1062500)	Заданная минимальная длительность видеокадра. Включает в себя длительность всех ADVB кадров видеокадра, интервалы между ними, а также заданный интервал между видеокадрами При попытке записать значение менее 4 – запись не происходит	Чт/Зп

Заданное значение по умолчанию соответствует частоте формирования видеокадров 25Гц при скорости 1 GFC.

Фактический период формирования видеокадров может превышать заданный максимум на расчетную длительность одной видеостроки без одного TLW (tx_TLT — 1). Точное соответствие фактического периода заданному будет обеспечено только в случае кратности заданной длительности видеокадра расчетной длительности видеостроки. При условии сохранения геометрии видеокадра, параметр tx_TLT может варьироваться путем изменения интервалов между ADVB кадрами (параметры tx_t3 и tx_t4).

1.3.10.2.2.2.6 Регистры параметров входных видеоданных

1.3.10.2.2.2.6.1 TX_COLOR_CONVERSION_OUTFORMAT (0x380)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:9			Резерв	
8	tx_outformat24_unpack	0x0	Данное поле имеет значение только при значении выходного формата 0x3 0x0: Режим Packed RGB 24-разрядный 0x1: Режим Unpacked RGB 24-разрядный Данное поле влияет только на содержимое Объекта 0. Содержимое Объекта 2 в обоих режимах идентично в соответствии со спецификацией ARINC-818	Чт/Зп
7:2			Резерв	
1:0	tx_outformat	0x1	Режим цветности изображения в памяти: 0x0: RAW data 0x1: 8-разрядный серый 0x2: 16-разрядный цветной в кодировке 5(R):6(G):5(B) 0x3: 24-разрядный цветной в кодировке 8(R):8(G):8(B)	Чт/Зп

1.3.10.2.2.2.7 Регистры управления обработкой видеоданных

1.3.10.2.2.2.7.1 TX_VIDEO_PROCESSING_ENS (0x400)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:9			Резерв	
8	tx_overlay_en	0x0	Активация совмещения 0x0: совмещение не осуществляется 0x1: совмещение осуществляется	Чт/Зп
7:0			Резерв	

В режиме совмещения происходит суммирование между собой соответствующих цветовых компонент двух изображений.

1.3.10.2.2.2.8 Регистры управления интерфейсом AXI DMA

1.3.10.2.2.2.8.1 TX_ADMA_MNG (0x480)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:2			Резерв	
1:0	tx_buffer_side	0x2	Активный видеобуфер 0x1: Передатчик загружает изображение из буфера 1 0x2: Передатчик загружает изображение из буфера 0	Чт

Управляющему процессору следует формировать новое изображение в буфере, отличном от того, с которым взаимодействует передатчик. По окончанию загрузки изображения из памяти – номер активного буфера автоматически изменяется.

1.3.10.2.2.2.8.2 TX_ADMA_BASEADDR_IM0_SIDE0 (0x484)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:0	tx_im0_baseaddr_side0	0x0	Регистр базового адреса буфера 0	Чт/Зп

1.3.10.2.2.2.8.3 TX_ADMA_BASEADDR_IM0_SIDE1 (0x488)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:0	tx_im0_baseaddr_side1	0x0	Регистр базового адреса буфера 1	Чт/Зп

1.3.10.2.2.2.8.4 TX_ADMA_BASEADDR_IM1_SIDE0 (0x48C)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:0	tx_im1_baseaddr_side0	0x0	Регистр базового адреса буфера 0 второго изображения при совмещении	Чт/Зп

1.3.10.2.2.2.8.5 TX_ADMA_BASEADDR_IM1_SIDE1 (0x490)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:0	tx_im1_baseaddr_side1	0x0	Регистр базового адреса буфера 1 второго изображения при совмещении	Чт/Зп

1.3.10.2.2.2.8.6 TX_ADMA_BUFFER_STRING_SIZE (0x494)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:0	tx_buffer_string_size	0x140 (320)	Размер видеостроки буфера	Чт/Зп

Размер видеостроки буфера должен быть больше или равен фактическому размеру видеостроки изображения в памяти. Про достижении конца видеостроки изображения в памяти указатель адреса чтения из памяти переключается на начало новой видеостроки буфера.

1.3.10.2.2.2.8.7 TX_RAW_ZERO_FRAME_SIZE (0x498)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:12			Резерв	
11:0	tx_raw_zero_frame_size	0x0	Размер 0 фрейма в режиме RAW data	Чт/Зп

Регистр определяет размер 0 (сервисного) фрейма в режиме RAW data. Размер должен быть кратен 8 байтам. Для начала работы в режиме RAW не должен иметь нулевого значения.

1.3.10.2.2.2.8.8 TX_RAW_QTY (0x49C)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:19			Резерв	
18:16	tx_raw_video_line_segs_qty	0x0	Количество сегментов в одном фрейме.	Чт/Зп
15:11			Резерв	
10:0	tx_raw_video_lines_qty	0x0	Количество фреймов в режиме RAW data (без учета 0 фрейма)	Чт/Зп

Поле tx_raw_video_lines_qty определяет количество фреймов, за исключением 0 фрейма, в режиме RAW data. Для начала работы в режиме RAW не должен иметь нулевого значения.

Поле tx_raw_video_line_segs_qty определяет порядок разбиения фреймов на сегменты. Корректные значения: 0x1 (1 сегмент), 0x2 (2 сегмента) 0x4 (4 сегмента). Запись значения 0x3 интерпретируется как 2 сегмента. Для начала работы в режиме RAW не должен иметь нулевого значения.

1.3.10.2.2.2.8.9 TX_RAW_VIDEO_LINE_SEG12_SIZE (0x4A0)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:28			Резерв	
27:16	tx_raw_video_line_seg2_size	0x0	Размер сегмента 1 в режиме RAW data	Чт/Зп
15:12			Резерв	
11:0	tx_raw_video_line_seg1_size	0x0	Размер сегмента 0 в режиме RAW data	Чт/Зп

Регистр определяет размеры сегментов 1 и 2. Оба поля регистра должны иметь значения кратные 8 байтам. Не зависимо от реального количество сегментов, оба поля должны быть проинициализированы (не должны иметь значения 0x0).

1.3.10.2.2.2.8.10 TX_RAW_VIDEO_LINE_SEG34_SIZE (0x4A4)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:28			Резерв	
27:16	tx_raw_video_line_seg4_size	0x0	Размер сегмента 4 в режиме RAW data	Чт/Зп
15:12			Резерв	
11:0	tx_raw_video_line_seg3_size	0x0	Размер сегмента 3 в режиме RAW data	Чт/Зп

Регистр определяет размеры сегментов 3 и 4. Оба поля регистра должны иметь значения кратные 8 байтам. Не зависимо от реального количество сегментов, оба поля должны быть проинициализированы (не должны иметь значения 0x0).

1.3.10.2.2.2.8.11 TX_RAW_CONTROL (0x4A8)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:1			Резерв	
0	tx_raw_word32_swap	0x0	Порядок следования байт в режиме Raw data 0x0: big-endian 0x1: little-endian Предполагается, что K-символы всегда располагаются в старшем байте	Чт/Зп

1.3.10.2.2.2.9 Регистры управления прерываниями

1.3.10.2.2.2.9.1 TX_IRQ_MNG (0x500)

Биты	Наименование	Значение по сбросу	Описание	Доступ
15:1		0x0	Резерв	
0	tx_irq_lock_override	0x0	0x0: программная имитация прерываний - запрещена 0x1: программная имитация прерываний - разрешена	Чт/Зп

Данный регистр используется в случае осуществления программной имитации прерываний путем записи в регистры TX_IRQ_SET и TX_IRQ_RESET. По умолчанию программная имитация прерываний – запрещена.

1.3.10.2.2.2.9.2 TX_IRQ_REQ (0x510)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:16		0x0	Резерв	Чт
15	tx_irq_collision	0x0	Прерывание по повторному возникновению события при наличии запроса в регистре запросов прерываний	Чт
14:6		0x0	Резерв	Чт
5	tx_txfifo_empty	0x0	Прерывание по опустошению выходного буфера	Чт
4	tx_transmitter_fsm_error	0x0	Прерывание по сбою внутреннего автомата состояния передатчика При возникновении данного события требуется программный сброс модуля передатчика	Чт
3	tx_rdma_fsm_error	0x0	Прерывание по сбою внутреннего автомата состояния контроллера RDMA При возникновении данного события требуется программный сброс модуля передатчика	Чт
2	tx_rdma_frame_done	0x0	Прерывание по окончанию выдачи текущего ADVB-контейнера	Чт
1	tx_rdma_memory_done	0x0	Прерывание по окончанию работы RDMA	Чт
0	tx_rdma_frame_start	0x0	Прерывание по началу работы RDMA Фактически индицирует начало выдачи текущего ADVB контейнера	Чт

Поля регистра запросов прерываний взводятся при наступлении соответствующих событий, указанных в таблице.

Регистр запросов прерываний побитно соответствует регистру масок прерываний TX_IRQ_MASK, вектору прерываний TX_IRQ_VECTOR и регистру коллизий TX_IRQ_COLLISION.

Сброс полей регистра запросов прерываний (перезапись значений в поля вектора прерываний TX_IRQ_VECTOR) происходит в следующих случаях:

чтение вектора прерываний TX_IRQ_VECTOR при наличии значения 0x1 в соответствующем поле регистра масок прерываний TX_IRQ_MASK;

наличие значения 0x0 в соответствующем поле TX_IRQ_VECTOR вектора прерываний при наличии значения 0x1 в соответствующем поле регистра масок прерываний TX_IRQ_MASK.

Программная установка и сброс полей регистра запросов прерываний возможны при помощи регистров TX_IRQ_SET и TX_IRQ_RESET при наличии значения 0x1 в поле tx_irq_lock_override регистра TX_IRQ_MNG.

1.3.10.2.2.2.9.3 TX_IRQ_MASK (0x514)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:16		0x0	Резерв	
15:0	tx_irq_mask	0xFFFF	Значение 0x1 разрешает перезапись соответствующего прерывания из регистра запросов прерываний TX_IRQ_REQ в регистр вектора прерываний TX_IRQ_VECTOR	Чт/Зп

Данный регистр побитно соответствует регистру запроса прерываний TX_IRQ_REQ.

1.3.10.2.2.2.9.4 TX_IRQ_VECTOR (0x518)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:16		0x0	Резерв	
15:0	tx_irq_vector	0x0	Вектор прерываний	Чт

Внешнее прерывание блока передатчика формируется при наличии значения 0x1 хотя бы в одном из полей регистра вектора прерываний.

Данный регистр побитно соответствует регистру запросов прерываний TX_IRQ_REQ.

1.3.10.2.2.2.9.5 TX_IRQ_COLLISION (0x51C)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:16		0x0	Резерв	
14:0	tx_irq_collision	0x0	Коллизии прерываний	Чт

Поля регистра запросов прерываний взводятся при наступлении соответствующих событий в случае когда соответствующие прерывания уже присутствуют в регистре запросов прерываний TX_IRQ_REQ и ещё не были обработаны.

Бит 15 регистра запросов прерываний TX_IRQ_REQ не проверяется на наличие коллизий, так как в нем содержится само прерывание по фиксации коллизии.

Данный регистр побитно соответствует регистру запросов прерываний TX_IRQ_REQ.

Регистр коллизий сбрасывается чтением.

1.3.10.2.2.2.9.6 TX_IRQ_SET (0x520)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:4			Резерв	
3:0	tx_req_field_number_set		Номер поля в регистре запросов прерываний TX_IRQ_REQ	Зп

Данный регистр служит для программной установки прерываний в регистре запросов прерываний TX_IRQ_REQ.

Программная установка полей регистра запросов прерываний возможна при условии наличия значения 0x1 в поле tx_irq_lock_override регистра TX_IRQ_MNG.

Для программной установки прерывания в каком либо поле регистра запросов прерываний TX_IRQ_REQ, необходимо осуществить запись номера этого поля в поле tx_req_field_number_set данного регистра.

1.3.10.2.2.2.9.7 TX_IRQ_RESET (0x524)

Биты	Наименование	Значение по сбросу	Описание	Доступ
31:5			Резерв	
4	tx_irq_soft_reset		Программный сброс контроллера прерываний	Зп
3:0	tx_req_field_number_reset		Номер поля в регистре запросов прерываний TX_IRQ_REQ	Зп

Данный регистр служит для программного сброса прерываний в регистре запросов прерываний TX_IRQ_REQ.

Программный сброс полей регистра запросов прерываний возможен при условии наличия значения 0x1 в поле tx_irq_lock_override регистра TX_IRQ_MNG.

Для программного сброса прерывания в каком либо поле регистра запросов прерываний TX_IRQ_REQ, необходимо осуществить запись номера этого поля в поле tx_req_field_number_reset данного регистра.

Запись значения 0x1 в поле tx_irq_soft_reset данного регистра приводит к программному сбросу контроллера прерываний.

Регистры заголовка ADVB кадра (0x600 : 0x614)

Содержимое регистров заголовка ADVB кадра переписывается в заголовки ADVB кадров формируемого изображения. Не все поля регистров заголовка ADVB кадра доступны для записи.

В таблице приведены значения полей регистров заголовка ADVB кадра после сброса. Поля, доступные только для чтения обозначены цветом.

Слово	Байт 3	Байт 2	Байт 1	Байт 0
0	0100 0100 (R_CTL)	0000 0000 (Dest._ID)	0000 0000 (Dest._ID)	0000 0000 (Dest._ID)
1	0000 0000 (Reserved) (CS_CTL)	0000 0000 (Source_ID)	0000 0000 (Source_ID)	0000 0000 (Source_ID)
2	0110 0000 0x60=FC-AV	0011 0000 (bit 19=END_SEQ) (F_CTL)	0000 0000 (F_CTL)	0000 0000 (F_CTL)
3	0000 0000 (SEQ_ID)	0000 0000 (DF_CTL)	0000 0000 (SEQ_CNT)	0000 0000 (SEQ_CNT)
4	1111 1111 (OX_ID)	1111 1111 (OX_ID)	1111 1111 (RX_ID)	1111 1111 (RX_ID)
5	0000 0000 (PARAMETER)	0000 0000 (PARAMETER)	0000 0000 (PARAMETER)	0000 0000 (PARAMETER)

1.3.10.2.2.2.10 Регистры заголовка ADVB контейнера и Объекта 0

1.3.10.2.2.2.10.1 Регистры заголовка ADVB контейнера (0x700:0x754)

Содержимое регистров заголовка ADVB контейнера переписывается в заголовки ADVB контейнеров формируемого изображения. Не все поля регистров заголовка ADVB контейнера доступны для записи.

В таблице приведены значения полей регистров заголовка ADVB контейнера после сброса. Поля, доступные только для чтения выделены серым фоном.

Слово	Идентификатор	Байт 3	Байт 2	Байт 1	Байт 0
0	Container Count	0000 0000	0000 0000	0000 0000	0000 0000
1	Clip ID	0000 0000	0000 0000	0000 0000	0000 0000
2	Container Time Stamp	0000 0000	0000 0000	0000 0000	0000 0000
3	Container Time Stamp	0000 0000	0000 0000	0000 0000	0000 0000
4	Transmission Type	0100 0100 (Video Frame Rate)	0000 0001 (Transmission Rate)	0000 0000 (Reserved)	0000 0000 (Reserved)
	Container Type	0000 0000 (Mode)	0000 0100 (Number of Objects)	0000 0000 (Reserved)	0000 0000 (Size of Ext. Header)
	Object 0 Class	0101 0000 (Type Ancillary)	0000 0000 (Link Pointer)	1101 0000 (SPDV Index)	0000 0000 (SPDV Index)

Слово	Идентификатор	Байт 3	Байт 2	Байт 1	Байт 0
	Object 0 Size	0000 0000	0000 0000	0000 0000	0001 0000
	Object 0 Offset	0000 0000	0000 0000	0000 0000	0101 1000
	Object 0 Object Type Defined	0000 0000	0000 0000	0000 0000	0000 0000
0	Object 1 Class	0100 0000 (Type Audio)	0000 0000 (Link Pointer)	1101 0000 (SPDV Index)	0000 0000 (SPDV Index)
1	Object 1 Size	0000 0000	0000 0000	0000 0000	0000 0000
2	Object 1 Offset	0000 0000	0000 0000	0000 0000	0110 1000
3	Object 1 Object Type Defined	0000 0000	0000 0000	0000 0000	0000 0000
4	Object 2 Class	0100 0000 (Type Video)	0000 0000 (Link Pointer)	1101 0000 (SPDV Index)	0000 0000 (SPDV Index)
5	Object 2 Size	0000 0000	0000 0000	1111 1010	0000 0000
6	Object 2 Offset	0000 0000	0000 0000	0000 0000	0110 1000
7	Object 2 Object Type Defined	0000 0000	0000 0000	0000 0000	0000 0000
8	Object 3 Class	0100 0000 (Type Video)	0000 0000 (Link Pointer)	1101 0000 (SPDV Index)	0000 0000 (SPDV Index)
9	Object 3 Size	0000 0000	0000 0000	0000 0000	0000 0000
0	Object 3 Offset	0000 0000	0000 0000	1111 1010	0110 1000
1	Object 3 Object Type Defined	0000 0000	0000 0000	0000 0000	0000 0000

1.3.10.2.2.2.10.2 Регистры Объекта 0 (0x758:0x764)

Содержимое регистров Объекта 0 переписывается в Объект 0 формируемого изображения. Не все поля регистров заголовка ADVB контейнера доступны для записи.

Поля “Number of Rows” и “Number of Columns” определяют геометрию видеокадра и непосредственно влияют на функционирование передатчика. Запись значения 0x0 в данные поля невозможна. Формат видеокадра по умолчанию – 320x200x8бит (цветовой режим определяется регистром TX_COLOR_CONVERSION_OUTFORMAT)

В таблице приведены значения полей регистров Объекта 0 после сброса. Поля, доступные только для чтения выделены серым фоном.

Слово 0																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Number of Rows 0xC8 (200)														Number of Columns 0x140 (320)														Frame / Field 0x0			
0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
Слово 1																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Слово 2																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Prior Image CRC 0x0																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Слово 3																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Parameter 2 Type 0x0						Parameter 2 Data 0x0										Parameter 1 Type 0x0						Parameter 1 Data 0x0									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1.3.10.2.3 Контроллер интерфейса Fiber Channel

1.3.10.2.3.1 Общие сведения

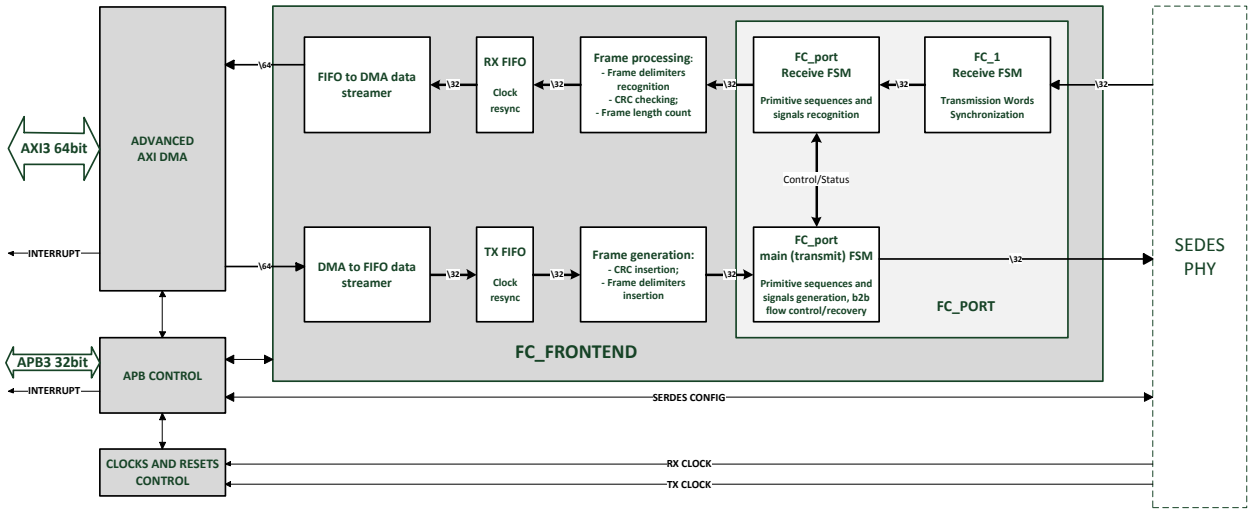


Рисунок 1.148 - Структура контроллера Fiber Channel

1.3.10.2.3.2 Аппаратный интерфейс

Сигнал	I/O	Описание
Системные сигналы		
fcrcm_core_irq_o	O	Уровневое прерывание подсистему FC_FRONTEND, активный уровень высокий
fcrcm_dma_irq_o	O	Уровневое прерывание подсистемы ADVANCED AXI DMA, активный уровень высокий
sys_clk_i	I	Основной системный синхросигнал
sys_rst_n_i	I	Основной системный сигнал асинхронного сброса (активный уровень – низкий)
Интерфейс APB 3.0 Slave		
Стандартные сигналы интерфейса APB 3 Slave		
Интерфейс AXI 3.0 Master		
Стандартные сигналы интерфейса AXI 3 Master		
Интерфейс к SERDES PHY		
fc_rx_clk_i	I	Тактовый сигнал для приема данных
fc_rx_data_i[31:0]	I	Входные данные
fc_rx_data_signal_ok_i	I	Сигнал валидности поступающих данных
fc_rx_data_sync_error_i	I	Сигнал индикации ошибки синхронизации
fc_rx_data_z_i	I	Сигнал метки К-символа входных данных
fc_tx_clk_i	I	Тактовый сигнал для передачи данных
fc_tx_data_o[31:0]	O	Выходные данные
fc_tx_data_z_o	O	Сигнал метки К-символа выходных данных
fc_tx_data_en_o	O	Сигнал валидности выходных данных
fc_phy_control_o[31:0]	O	Шина управления внешним SERDES PHY (опционально)
fc_phy_status_i[31:0]	I	Шина статуса внешнего SERDES PHY (опционально)

1.3.10.2.3.3 Описание регистров

Название	Адрес	Описание	Доступ
Регистры подсистемы FC_FRONTEND			
CONTROL	000	Основной регистр управления подсистемы FC_FRONTEND	RW
STATUS	004	Регистр состояния подсистемы FC_FRONTEND	R
IRQMASKS	008	Маски прерываний подсистемы FC_FRONTEND	RW
B2BCONTROL1	00C	Управление механизмом b2b flow control	RW
B2BCONTROL2	010	Управление механизмом b2b flow control	RW
SOFTRST	014	Программный сброс подсистемы FC_FRONTEND	RW
TIMER5VAL	018	Таймер 5мс для подсистемы FC_PORT	RW
RTTOVVAL	01c	Таймер R_T_TOV	RW
EDTOVVAL	020	Таймер E_D_TOV	RW
EDTOV	024	Таймаут E_D_TOV	RW
LSYNCLIMIT	028	Предельное количество слов с потерянной синхронизацией	RW
ER_LFAC	02c	Значение счетчика Link Failure Count	R
ER_LSYC	030	Значение счетчика Loss-of-Synchronization Count	R
ER_LSIC	034	Значение счетчика Loss-of-Signal Count	R
ER_PSEC	038	Значение счетчика Primitive Sequence Protocol Error Count	R
ER_TWERR	03c	Значение счетчика Invalid Transmission Word Counter	R
ER_CRCERR	040	Значение счетчика Invalid CRC Count	R

Название	Адрес	Описание	Доступ
DBG_SOFTRST	044	Отладочный регистр: программные сбросы подсистем	RW
DBG_CLKCTRL	048	Отладочный регистр: управление тактовыми сигналами	RW
DBG_BB1	04c	Отладочный регистр: механизм b2b flow control	R
DBG_BB2	050	Отладочный регистр: механизм b2b flow control	R
DBG_BB3	054	Отладочный регистр: механизм b2b flow control	R
DBG_BBFC	058	Отладочный регистр: механизм b2b flow control	R
MAXFLEN	05c	Максимально допустимая длина принимаемых пакетов	RW
PHYCTRL	060	Опциональный регистр: управление SERDES PHY	RW
PHYSTATUS	064	Опциональный регистр: статус SERDES PHY	R
Регистры подсистемы ADVANCED AXI DMA			
ADMA_ID	800	Идентификатор подсистемы AXI DMA	R
DMA_REVISION	804	Номер ревизии и количество каналов AXI DMA	R
DMA_SW_RESET	808	Программный сброс ADVANCED AXI DMA	RW
CH_STATUS	80c	Регистр состояния каналов AXI DMA	R
ARLEN	810	Значение ARLEN для транзакций по AXI	RW
AWLEN	814	Значение AWLEN для транзакций по AXI	RW
BRESP	818	Значение BRESP интерфейса AXI канала RX	R
RAXI_ERR_ADDR	81c	Адрес AXI-транзакции чтения канала TX, вызвавшей ошибку интерфейса AXI (RRESP != 00)	R
RDMA_SETTINGS	900	Настройки канала TX AXI DMA	
RDMA_STATUS	904	Регистр состояния канала TX AXI DMA	
RDMA_SYS_ADDR	908	Адрес таблицы дескрипторов канала TX	RW
RDMA_TBL_SIZE	90c	Размер таблицы дескрипторов канала TX (по умолчанию 2 Кб)	RW
RDMA_ADDR	910	Отладочный регистр: текущий адрес записи данных для канала TX	RW
RDMA_LEN	914	Отладочный регистр: размер блока данных для канала TX	RW
RDMA_DESC_ADDR	918	Отладочный регистр: адрес текущего дескриптора	RW
WDMA_SETTINGS	a00	Настройки канала RX AXI DMA	
WDMA_STATUS	a04	Регистр состояния канала RX AXI DMA	
WDMA_SYS_ADDR	a08	Адрес таблицы дескрипторов канала RX	RW
WDMA_TBL_SIZE	a0c	Размер таблицы дескрипторов канала RX (по умолчанию 2 Кб)	RW
WDMA_ADDR	a10	Отладочный регистр: текущий адрес чтения данных для канала RX	RW
WDMA_LEN	a14	Отладочный регистр: размер блока данных для канала RX	RW
WDMA_DESC_ADDR	a18	Отладочный регистр: адрес текущего дескриптора	RW

1.3.10.2.3.3.1.1 CONTROL – Основной регистр управления подсистемы FC_FRONTEND, адрес 000h, RW

Разряды	Значение по умолчанию	Описание
[31]	0	Изменение порядка байт передаваемых (TX) по каналу FC слов данных: 1 – порядок байт меняется (0123 -> 3210) 0 – порядок байт остается неизменным
[30]	0	Изменение порядка байт принимаемых (RX) по каналу FC слов данных: 1 – порядок байт меняется (0123 -> 3210) 0 – порядок байт остается неизменным
[29:13]	-	
[12:5]	0	Количество RRDY, BBSCS или BBSCR для пересылки (используется совместно с полем [4:2])

Разряды	Значение по умолчанию	Описание
[4:2]	0	Тип последовательности Primitive Sequence, передающейся в отсутствие данных в режиме pass-through (бит [2] данного регистра): b000 – Idle b001 – NOS (Not Operational) b010 – OLS (Offline) b011 – LR (Link Reset) b100 – LRR (Link Reset Response) b101 – Idle + посылка RRDY в количестве, указанном в поле [12:5] b110 – Idle + посылка BBSCS в количестве, указанном в поле [12:5] b111 – Idle + посылка BBSCR в количестве, указанном в поле [12:5]
[1]	0	1 - Включение режима pass-through (прием и передача данных без установления соединения типа FC_PORT; тип последовательности Primitive Sequence, передающейся в отсутствие данных, определяется полем [4:0] данного регистра)
[0]	0	1 - Включение подсистемы FC_PORT

1.3.10.2.3.3.1.2 STATUS – Регистр состояния подсистемы FC_FRONTEND, адрес 004h, R

При чтении данного регистра автоматически сбрасывается прерывание на линии fcrcm_core_irq_o.

Разряды	Значение по умолчанию	Описание
[31:30]	-	
[29]	0	Состояние принимаемого (RX) сигнала по каналу FC: 1 – сигнал присутствует и стабилен 0 – сигнала нет
[28:24]	0	Тип принимаемой по каналу FC последовательности Primitive Sequence: b00000 – не распознается ни одна последовательность, рекомендуется проверить наличие сигнала b00001 – принимается Idle b00010 – принимается LR (Link Reset) b00100 – принимается LRR (Link Reset Response) b01000 – принимается OLS (Offline) b10000 – принимается NOS (Not Operational)
[23:4]	-	
[3]	0	1 - Критическая ошибка механизма b2b flow control. В случае возникновения желателен перезапуск подсистемы FC_PORT.
[2]	0	1 - Опустошение буфера FIFO канала передачи данных FC TX во время передачи.
[1]	0	1 - Переполнение буфера FIFO канала приема данных FC RX во время приема.
[0]	0	1 - Соединение типа FC_PORT успешно установлено. 0 - Соединение типа FC_PORT не установлено.

1.3.10.2.3.3.1.3 IRQMASKS – Маски прерываний подсистемы FC_FRONTEND, адрес 008h, RW

Линия прерывания fcrcm_core_irq_o . 1 – прерывание активно, 0 – прерывание замаскировано.

Разряды	Значение по умолчанию	Описание
[31:4]	-	
[3]	0	Прерывание при критической ошибке механизма b2b flow control. В случае возникновения желателен перезапуск подсистемы FC_PORT.

Разряды	Значение по умолчанию	Описание
[2]	0	Прерывание по опустошению буфера FIFO канала приема данных FC RX во время приема.
[1]	0	Прерывание по переполнению буфера FIFO канала передачи данных FC TX во время передачи.
[0]	0	Прерывание по любому изменению состояния соединения типа FC_PORT.

1.3.10.2.3.3.1.4 B2BCONTROL1 – Управление механизмом b2b flow control, адрес 00ch, RW

Разряды	Значение по умолчанию	Описание
[31:25]	-	
[24]	0	1 – Отправка сигналов BBRDY контролируется программно (регистр B2BCONTROL2). 0 – Отправка сигнала BBRDY производится автоматически после каждого полученного пакета.
[23:20]	0	Значение BB_SC_N ответной части, подключенной к данному контроллеру Fiber Channel.
[19:16]	0	Значение BB_SC_N данного контроллера Fiber Channel.
[15:0]	1	Значение BB_Credit ответной части, подключенной к данному контроллеру Fiber Channel.

1.3.10.2.3.3.1.5 B2BCONTROL2 – Управление механизмом b2b flow control, адрес 010h, RW

Разряды	Значение по умолчанию	Описание
[31:16]	-	
[15:0]	0	Количество сигналов BBRDY, которое необходимо отправить (работает только при B2BCONTROL1[24] = 1) .

1.3.10.2.3.3.1.6 SOFTRST – Программный сброс подсистемы FC_FRONTEND, адрес 014h, RW

Разряды	Значение по умолчанию	Описание
[31:1]	-	
[0]	0	1 - Программный сброс инициирован. 0 - Программный сброс завершен.

1.3.10.2.3.3.1.7 TIMER5VAL – Таймер 5мс для подсистемы FC_PORT, адрес 018h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0xfffff	Таймаут 5мс для подсистемы FC_PORT, необходим для установления соединения. Значение по умолчанию для системной частоты 200МГц. Формула: <период системной частоты> * TIMER5VAL[31:0] ~= 5мс

1.3.10.2.3.3.1.8 RTTOVVAL – Таймер R_T_TOV, адрес 01ch, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0x1312f00	Таймаут R_T_TOV для подсистемы FC_PORT. Значение по умолчанию в 100мс для системной частоты 200МГц. Формула: <период системной частоты> * RTTOVVAL[31:0] ~= 100мс

1.3.10.2.3.3.1.9 EDTOVVAL – Таймер E_D_TOV, адрес 020h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0x17d78f00	Таймаут E_D_TOV для подсистемы FC_PORT. Значение по умолчанию в 2 секунды для системной частоты 200МГц. Формула: <период системной частоты> * EDTOVVAL[31:0] ~= 2с

1.3.10.2.3.3.1.10 EDTOV – Таймаут E_D_TOV, адрес 024h, RW

Разряды	Значение по умолчанию	Описание
[31:1]	-	
[0]	0	Запись 1 сообщает контроллеру о программном событии таймаута E_D_TOV.

1.3.10.2.3.3.1.11 LSYNCLIMIT – Предельное количество слов с потерянной синхронизацией, адрес 028h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0x fffff	Предельное количество слов с потерянной синхронизацией, используется в алгоритме FC_PORT для определения события Loss of Sync > Limit

1.3.10.2.3.3.1.12 ER_LFAC – Значение счетчика Link Failure Count, адрес 02ch, R

Разряды	Значение по умолчанию	Описание
[31:0]	0	Значение счетчика Link Failure Count из состава Link Error Status Block (приложение E стандарта FC-FS-2).

1.3.10.2.3.3.1.13 ER_LSYC – Значение счетчика Loss-of-Synchronization Count, адрес 030h, R

Разряды	Значение по умолчанию	Описание
[31:0]	0	Значение счетчика Loss-of-Synchronization Count из состава Link Error Status Block (приложение E стандарта FC-FS-2).

1.3.10.2.3.3.1.14 ER_LSIC – Значение счетчика Loss-of-Signal Count, адрес 034h, R

Разряды	Значение по умолчанию	Описание
[31:0]	0	Значение счетчика Loss-of-Signal Count из состава Link Error Status Block (приложение E стандарта FC-FS-2).

1.3.10.2.3.3.1.15 ER_PSEC – Значение счетчика Primitive Sequence Protocol Error Count, адрес 038h, R

Разряды	Значение по умолчанию	Описание
[31:0]	0	Значение счетчика Primitive Sequence Protocol Error Count из состава Link Error Status Block (приложение E стандарта FC-FS-2).

1.3.10.2.3.3.1.16 ER_TWERR – Значение счетчика Invalid Transmission Word Counter, адрес 03ch, R

Разряды	Значение по умолчанию	Описание
[31:0]	0	Значение счетчика Invalid Transmission Word Counter из состава Link Error Status Block (приложение E стандарта FC-FS-2).

1.3.10.2.3.3.1.17 ER_CRCERR – Значение счетчика Invalid CRC Count, адрес 040h, R

Разряды	Значение по умолчанию	Описание
[31:0]	0	Значение счетчика Invalid CRC Count из состава Link Error Status Block (приложение E стандарта FC-FS-2).

1.3.10.2.3.3.1.18 DBG_SOFTRST – Отладочный регистр: программные сбросы подсистем, адрес 044h, RW

Разряды	Значение по умолчанию	Описание
[31:3]	-	
[2]	0	1 - Программный сброс подсистемы, тактируемой sys_clk инициирован. 0 - Программный сброс подсистемы, тактируемой sys_clk завершен.
[1]	0	1 - Программный сброс подсистемы, тактируемой tx_clk инициирован. 0 - Программный сброс подсистемы, тактируемой tx_clk завершен.
[0]	0	1 - Программный сброс подсистемы, тактируемой rx_clk инициирован. 0 - Программный сброс подсистемы, тактируемой rx_clk завершен.

1.3.10.2.3.3.1.19 DBG_CLKCTRL – Отладочный регистр: управление тактовыми сигналами, адрес 048h, RW

Разряды	Значение по умолчанию	Описание
[31:3]	-	
[2]	1	1 - Разрешена подача тактового сигнала sys_clk. 0 - Запрещена подача тактового сигнала sys_clk.
[1]	0	1 - Принудительна разрешена подача тактового сигнала tx_clk. 0 - Нормальная работа.
[0]	0	1 - Принудительна разрешена подача тактового сигнала rx_clk. 0 - Нормальная работа.

1.3.10.2.3.3.1.20 DBG_BB1 – Отладочный регистр: механизм b2b flow control, адрес 04ch, R

Разряды	Значение по умолчанию	Описание
[31:16]	0	Текущее значение счетчика bb_credit (сколько отправлено пакетов, но не получено RDY)
[15:0]	0	Текущее значение счетчика bb_rrdy_gen (сколько RDY ждут отправки)

1.3.10.2.3.3.1.21 DBG_BB2 – Отладочный регистр: механизм b2b flow control, адрес 050h, R

Разряды	Значение по умолчанию	Описание
[31:16]	0	Текущее значение счетчика bb_rdy_n (количество принятых RDY)
[15:0]	0	Текущее значение счетчика bb_frm_n (количество принятых пакетов)

1.3.10.2.3.3.1.22 DBG_BB3 – Отладочный регистр: механизм b2b flow control, адрес 054h, R

Разряды	Значение по умолчанию	Описание
[31:16]	0	Текущее значение счетчика bb_frame_sent (сколько отправлено RDY)
[15:0]	0	Текущее значение счетчика bb_rdy_sent (сколько отправленных пакетов)

1.3.10.2.3.3.1.23 DBG_BBFC – Отладочный регистр: механизмы b2b flow control и FC_PORT, адрес 058h, R

Разряды	Значение по умолчанию	Описание
[31:8]	-	
[7:4]	0	Текущее значение счетчика bb_sc_n()
[3:0]	?	Текущее состояние FC_PORT FSM

1.3.10.2.3.3.1.24 MAXFLEN - Максимально допустимая длина принимаемых пакетов, адрес 05ch, RW

Разряды	Значение по умолчанию	Описание
[31:14]	-	
[13:0]	0x858	Максимально допустимая длина принимаемых пакетов, по умолчанию это 2136 (в регистр пишется на 1 меньшее значение: 2136-1=2135) байт в соответствии со стандартом FC-FS-2 (при условии отсутствия заголовка Extended Header и не включая frame delimiters)

1.3.10.2.3.3.1.25 PHYCTRL – Опциональный регистр: управление SERDES PHY, адрес 060h, RW

Управляет выходной шиной fc_phy_control_o[31:0]. Использование данного регистра регламентируется в зависимости от типа SERDES PHY.

1.3.10.2.3.3.1.26 PHYSTATUS – Опциональный регистр: статус SERDES PHY, адрес 064h, R

Позволяет считать состояние входной шины fc_phy_status_i[31:0]. Использование данного регистра регламентируется в зависимости от типа SERDES PHY.

1.3.10.2.3.3.1.27 ADMA_ID – Идентификатор подсистемы AXI DMA, адрес 800h, R

Разряды	Значение по умолчанию	Описание
[31:0]	0x414d4441	Идентификатор подсистемы AXI DMA

1.3.10.2.3.3.1.28 DMA_REVISION – Номер ревизии и количество каналов AXI DMA, адрес 804h, R

Разряды	Значение по умолчанию	Описание
[31:8]	-	Идентификатор подсистемы AXI DMA
[7:4]	0x1	Количество каналов записи в устройстве
[3:0]	0x1	Количество каналов чтения в устройстве

1.3.10.2.3.3.1.29 DMA_SW_RESET – Программный сброс подсистемы ADVANCED AXI DMA, адрес 808h, RW

Разряды	Значение по умолчанию	Описание
[31:1]	-	
[0]	0	1 - Программный сброс инициирован 0 - Программный сброс завершен

1.3.10.2.3.3.1.30 CH_STATUS – Регистр состояния каналов AXI DMA, адрес 80ch, R

При чтении данного регистра автоматически сбрасывается прерывание на линии fcrcm_dma_irq_o.

Разряды	Значение по умолчанию	Описание
[31:17]	-	
[16]	0x0	1 – Прерывание канала RX, подробности в регистре WDMA_STATUS Аппаратно сбрасывается в 0 при чтении данного регистра.
[15:1]	-	
[0]	0x0	1 – Прерывание канала TX, подробности в регистре RDMA_STATUS Аппаратно сбрасывается в 0 при чтении данного регистра.

1.3.10.2.3.3.1.31 ARLEN – Значение ARLEN для транзакций по AXI, адрес 810h, RW

Разряды	Значение по умолчанию	Описание
[31:4]	-	
[3:0]	0xf	Значение ARLEN для транзакций по AXI

1.3.10.2.3.3.1.32 AWLEN – Значение AWLEN для транзакций по AXI, адрес 814h, RW

Разряды	Значение по умолчанию	Описание
[31:4]	-	
[3:0]	0xf	Значение AWLEN для транзакций по AXI

1.3.10.2.3.3.1.33 BRESP – Значение BRESP интерфейса AXI канала RX, адрес 818h, R

Разряды	Значение по умолчанию	Описание
[31:2]	-	
[1:0]	0	Значение BRESP интерфейса AXI, значение в данном регистре актуально после возникновения ошибки (BRESP != 00) канала записи интерфейса AXI (бит 9 регистра DMA_STATUS)

1.3.10.2.3.3.1.34 RAXI_ERR_ADDR – Адрес AXI-транзакции канала TX, вызвавшей ошибку интерфейса AXI, адрес 81ch, R

Разряды	Значение по умолчанию	Описание
[31:0]	0	Адрес AXI-транзакции чтения, вызвавшей ошибку интерфейса AXI, значение в данном регистре актуально после возникновения ошибки (RRESP != 00) канала записи интерфейса AXI (бит 8 регистра DMA_STATUS)

1.3.10.2.3.3.1.35 RDMA_SETTINGS – Настройки канала TX AXI DMA, адрес 900h, RW

Биты [5:0] – маски прерываний линии fcrcm_dma_irq_o, соответствующих битам регистра RDMA_STATUS[5:0]. 1 – прерывание активно, 0 – прерывание замаскировано.

Разряды	Значение по умолчанию	Описание
[31]	0	1 - Остановка работы канала TX (чтение данных из системной памяти и их отправка по интерфейсу Fiber Channel)
[30]	-	Не используется, необходимо всегда записывать 0
[29]	0	1 - Работа с поддержкой внешней таблицы дескрипторов (основной режим) 0 - Работа без использования внешней таблицы дескрипторов (отладочный режим, не рекомендуется использовать)
[28]	0	1 - Включение канала TX (чтение данных из системной памяти и их отправка по интерфейсу Fiber Channel)
[27:6]	-	Не используется, необходимо всегда записывать 0
[5]	0x0	1 - Активно прерывание по ошибке канала TX интерфейса AXI (бит 5 регистра RDMA_STATUS) – RRESP != 00
[4]	-	
[3]	0x0	1 - Активно прерывание по достижению последнего адрес блока данных канала TX при работе без поддержки внешней таблицы дескрипторов (бит 3 регистра RDMA_STATUS)
[2]	0x0	1 - Активно прерывание по ошибке считывания дескриптора канала TX (бит 2 регистра RDMA_STATUS) – считанный дескриптор помечен как выполненный
[1]	0x0	1 - Закончено выполнение остановки RDMA, инициированная битом RDMA_SETTINGS[31]
[0]	0x0	1 - Активно прерывание окончанию выполнения дескриптора канала TX с 1 в поле Int (бит 0 регистра RDMA_STATUS)

1.3.10.2.3.3.1.36 RDMA_STATUS – Регистр состояния канала TX AXI DMA, адрес 904h, RW

Разряды	Значение по умолчанию	Описание
[31:23]	-	
[22:16]	0x0	Количество слов в буфер AXI канала TX (на момент чтения данного регистра)
[15:13]	-	
[12]	0x0	1 - Признак наличия (на момент чтения данного регистра) незавершенных запросов на чтение данных по шине AXI канала TX
[11:9]	-	
[8]	0x0	1 - Признак наличия (на момент чтения данного регистра) активной транзакции на шине AXI канала TX
[7]	0x1	Буфер AXI канала TX пуст

Разряды	Значение по умолчанию	Описание
[6]	0x0	Буфер AXI канала TX полон
[5]	0x0	1 - Произошла ошибка канала TX интерфейса AXI – получен RRESP != 00
[4]	-	
[3]	0x0	1 - Достигнут последний адрес блока данных канала TX при работе без поддержки внешней таблицы дескрипторов
[2]	0x0	1 - Считанный дескриптор помечен как выполненный
[1]	0x0	1 - Завершена остановка канала TX, инициированная битом RDMA_SETTINGS[31]
[0]	0x0	1 - Окончено выполнение дескриптора канала TX с 1 в поле Int

1.3.10.2.3.3.1.37 RDMA_SYS_ADDR – Адрес таблицы дескрипторов канала TX, адрес 908h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0	Начальный адрес таблицы дескрипторов канала TX, должен быть выровнен по 8 байт

1.3.10.2.3.3.1.38 RDMA_TBL_SIZE – Размер таблицы дескрипторов канала TX, адрес 90ch, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0x800	В начале работы в этот регистр программно записывается размер таблицы дескрипторов в байтах. Размер должен быть выровнен по 8 байт (длина одного дескриптора). По достижении конца таблицы AXI-DMA автоматически возвращается к адресу первого дескриптора таблицы, и работа продолжается в нормальном режиме. Текущее значение регистра не изменяется. Запись в данный регистр возможна только при выключенном RDMA. Попытка записи в данный регистр во время работы RDMA игнорируется.

1.3.10.2.3.3.1.39 RDMA_ADDR – Отладочный регистр: текущий адрес чтения данных для канала TX, адрес 910h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0	В этот регистр автоматически переписывается текущий адрес из дескриптора.

1.3.10.2.3.3.1.40 RDMA_LEN – Отладочный регистр: размер блока данных для канала TX, адрес 914h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0	В этот регистр автоматически переписывается размер блока данных из дескриптора.

1.3.10.2.3.3.1.41 RDMA_DESC_ADDR – Отладочный регистр: адрес текущего дескриптора канала TX, адрес 918h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0	В начале работы в этот регистр аппаратно переписывается адрес таблицы дескрипторов (если она используется). При переходе к новой строке таблицы, значение этого регистра меняется – в нём устанавливается адрес текущего дескриптора. При переходе к новой строке, значение RDMA_SYS_ADDR увеличивается на 8. Во время работы RDMA считанное значение регистра может отличаться от реального. Запись в данный регистр позволяет начать работу на с первого дескриптора в таблице, а с заданного в данном регистра адреса. Запись возможна только при выключенном RDMA (en_r(w)dma == 0). Попытка записи в данный регистр во время работы RDMA игнорируется. При записи в данный регистр RDMA начинает движение по таблице с дескриптора с заданным в регистре адресом. При чтении дескрипторов RDMA из таблицы не анализируется граница в 4 Кбайта. Установка не выровненного по 8 байт адреса таблицы дескрипторов может привести к ошибке.

1.3.10.2.3.3.1.42 WDMA_SETTINGS – Настройки канала RX AXI DMA, адрес a00h, RW

Биты [5:0] – маски прерываний линии fcrcm_dma_irq_o, соответствующих битам регистра WDMA_STATUS[5:0]. 1 – прерывание активно, 0 – прерывание замаскировано.

Разряды	Значение по умолчанию	Описание
[31]	0	1 - Остановка работы канала RX (приема данных по интерфейсу Fiber Channel и их записи в системную память)
[30]	-	Не используется, необходимо всегда записывать 0
[29]	0	1 - Работа с поддержкой внешней таблицы дескрипторов (основной режим) 0 - Работа без использования внешней таблицы дескрипторов (отладочный режим, не рекомендуется использовать)
[28]	0	1 - Включение канала RX (приема данных по интерфейсу Fiber Channel и их записи в системную память)
[27:6]	-	Не используется, необходимо всегда записывать 0
[5]	0x0	1 - Активно прерывание по ошибке канала RX интерфейса AXI (бит 5 регистра WDMA_STATUS) – BRESP != 00
[4]	-	
[3]	0x0	1 - Активно прерывание по достижению последнего адрес блока данных канала RX при работе без поддержки внешней таблицы дескрипторов (бит 3 регистра WDMA_STATUS)
[2]	0x0	1 - Активно прерывание по ошибке считывания дескриптора канала RX (бит 2 регистра WDMA_STATUS) – считанный дескриптор помечен как выполненный
[1]	0x0	1 - Активно прерывание по завершению остановки канала RX (бит 4 регистра WDMA_STATUS), инициированной битом WDMA_SETTINGS[31]
[0]	0x0	1 - Активно прерывание окончанию выполнения дескриптора канала RX с 1 в поле Int (бит 0 регистра WDMA_STATUS)

1.3.10.2.3.3.1.43 WDMA_STATUS – Регистр состояния канала RX AXI DMA, адрес a04h, RW

Разряды	Значение по умолчанию	Описание
[31:8]	-	
[7]	0x1	Буфер AXI канала RX пуст
[6]	0x0	Буфер AXI канала RX полон
[5]	0x0	1 - Произошла ошибка канала RX интерфейса AXI – получен BRESP != 00
[4]	-	
[3]	0x0	1 - Достигнут последний адрес блока данных канала RX при работе без поддержки внешней таблицы дескрипторов
[2]	0x0	1 - Считанный дескриптор помечен как выполненный
[1]	0x0	1 - Закончено выполнение остановки WDMA
[0]	0x0	1 - Окончено выполнение дескриптора канала RX с 1 в поле Int

1.3.10.2.3.3.1.44 WDMA_SYS_ADDR – Адрес таблицы дескрипторов канала RX, адрес a08h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0	Начальный адрес таблицы дескрипторов канала RX, должен быть выровнен по 8 байт

1.3.10.2.3.3.1.45 WDMA_TBL_SIZE – Размер таблицы дескрипторов канала RX, адрес a0ch, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0x800	В начале работы в этот регистр программно записывается размер таблицы дескрипторов в байтах. Размер должен быть выровнен по 8 байт (длина одного дескриптора). По достижении конца таблицы AXI-DMA автоматически возвращается к адресу первого дескриптора таблицы, и работа продолжается в нормальном режиме. Текущее значение регистра не изменяется. Запись в данный регистр возможна только при выключенном WDMA. Попытка записи в данный регистр во время работы WDMA игнорируется.

1.3.10.2.3.3.1.46 WDMA_ADDR – Отладочный регистр: адрес записи данных для канала RX, адрес a10h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0	В этот регистр автоматически переписывается текущий адрес из дескриптора.

1.3.10.2.3.3.1.47 WDMA_LEN – Отладочный регистр: размер блока данных для канала RX при работе без таблицы дескрипторов, адрес a14h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0	В этот регистр автоматически переписывается размер блока данных из дескриптора.

1.3.10.2.3.3.1.48 WDMA_DESC_ADDR – Отладочный регистр: адрес текущего дескриптора канала RX, адрес a18h, RW

Разряды	Значение по умолчанию	Описание
[31:0]	0	В начале работы в этот регистр аппаратно переписывается адрес таблицы дескрипторов (если она используется). При переходе к новой строке таблицы, значение этого регистра меняется – в нём устанавливается адрес текущего дескриптора. При переходе к новой строке, значение WDMA_SYS_ADDR увеличивается на 8. Во время работы R(W)DMA считанное значение регистра может отличаться от реального. Запись в данный регистр позволяет начать работу на с первого дескриптора в таблице, а с заданного в данном регистра адреса. Запись возможна только при выключенном WDMA (en_r(w)dma == 0). Попытка записи в данный регистр во время работы WDMA игнорируется. При записи в данный регистр WDMA начинает движение по таблице с дескриптора с заданным в регистре адресом. При чтении дескрипторов WDMA из таблицы не анализируется граница в 4 Кбайта. Установка не выровненного по 8 байт адреса таблицы дескрипторов может привести к ошибке.

1.3.10.2.3.4 Формат дескрипторов ADVANCED AXI DMA

Каждый дескриптор DMA управляет передачей(TX)/приемом(RX) одного пакета FC Frame. Размер одного дескриптора 8 байт. В таблице в системной памяти дескрипторы должны располагаться один за другим (по адресу 0 первый дескриптор, по адресу 8 второй и так далее). После достижения конца таблицы будет снова считан дескриптор по начальному адресу таблицы.

1.3.10.2.3.4.1 Дескриптор канала TX (при записи дескриптора):

Поле	Разряды	Описание
Адрес	[63:32]	32-битный стартовый адрес для чтения пакета, сформированного в системной памяти
Тип SOF пакета	[31:29]	0x0 : SOFc1 0x1 : SOFi1 0x2 : SOFn1 0x3 : SOFi2 0x4 : SOFn2 0x5 : SOFi3 0x6 : SOFn3 0x7 : SOFf
Тип EOF пакета	[28:26]	0x0, 0x6, 0x7 : EOFt 0x1 : EOFdt 0x2 : EOFa 0x3 : EOFn 0x4 : EOFni 0x5 : EOFdti
Длина пакета	[25:6]	Длина пакета данных (Frame) в байтах, должна быть кратна 4б
Act2	[5]	Тип дескриптора DMA 01,00,11 compl Дескриптор выполнен (отработан контроллером FC) 10 tran Дескриптор активен (в очереди на исполнение контроллером FC)
Act1	[4]	
-	[3]	Не используется, необходимо всегда записывать 0
Int	[2]	1 - Выработка прерывания по завершении выполнения
-	[1:0]	Не используется, необходимо всегда записывать 0

1.3.10.2.3.4.2 Дескриптор канала TX (при чтении отработанного дескриптора):

Поле	Разряды	Описание
Адрес	[63:32]	32-битный стартовый адрес для чтения пакета, сформированного в системной памяти (не изменяется при обработке дескриптора)
-	[31:6]	Не используется
Статус	[5:4]	01 - Дескриптор отработан. Любое другое значение говорит о том, что дескриптор еще не отработан.
-	[3:2]	Не используется
Ошибка соединения	[1]	1 - Произошел обрыв соединения FC_PORT при передаче данного пакета 0 - Передача пакета завершилась успешно
FIFO underrun	[0]	1 - Произошло опустошения буфера при передаче данного пакета 0 - Передача пакета завершилась успешно

1.3.10.2.3.4.3 Дескриптор канала RX (при записи дескриптора):

Поле	Разряды	Описание
Адрес	[63:32]	32-битный стартовый адрес для записи пакета, полученного по каналу Fiber Channel
-	[31:26]	Не используется, необходимо всегда записывать 0
Длина пакета	[25:6]	Максимальная длина принимаемого пакета данных (Frame) – размер выделенной области в памяти для записи пакета (пакет может быть меньше, но не больше этого значения)
Act2	[5]	Тип дескриптора DMA 01,00,11 compl Дескриптор выполнен (отработан контроллером FC) 10 tran Дескриптор активен (в очереди на исполнение контроллером FC)
Act1	[4]	
-	[3]	Не используется, необходимо всегда записывать 0
Int	[2]	1 - Выработка прерывания по завершении выполнения
-	[1:0]	Не используется, необходимо всегда записывать 0

1.3.10.2.3.4.4 Дескриптор канала RX (при чтении отработанного дескриптора):

Поле	Разряды	Описание
Адрес	[63:32]	32-битный стартовый адрес для записи пакета, полученного по каналу Fiber Channel (не изменяется при обработке дескриптора)
Тип SOF принятого пакета	[31:29]	0x0 : SOFc1 0x1 : SOFi1 0x2 : SOFn1 0x3 : SOFi2 0x4 : SOFn2 0x5 : SOFi3 0x6 : SOFn3 0x7 : SOFf
Тип EOF принятого пакета	[28:26]	0x0 : EOFt 0x1 : EOFdt 0x2 : EOFa 0x3 : EOFn 0x4 : EOFni 0x5 : EOFdti 0x6,0x7 : Ошибочный EOF
Длина пакета	[25:6]	Длина принятого пакета данных (Frame)
Статус	[5:4]	01 - Дескриптор отработан. Любое другое значение говорит о том, что дескриптор еще не отработан контроллером FC.
Превышение максимальной длины пакета	[3]	1 - Принят пакет с превышением максимально допустимой длины, установленной в MAXFLEN[13:0] 0 - Принят пакет без превышения максимально допустимой длины

Поле	Разряды	Описание
Ошибка синронизации	[2]	1 - При приеме пакета была зафиксирована ошибка синхронизации алгоритма 8B/10B 0 - Принят пакет без ошибок синхронизации
Ошибка соединения	[1]	1 - При приеме пакета была зафиксирована ошибка соединения FC_PORT 0 - Принят пакет без ошибок соединения
Ошибка CRC	[0]	1 - Принят пакет с ошибочной контрольной суммой CRC 0 - Принят пакет без ошибок CRC

Поле	Разряды	Описание
Адрес	[63:32]	32-битный стартовый адрес для чтения пакета, сформированного в системной памяти
Тип SOF пакета	[31:29]	0x0 : SOFc1 0x1 : SOFi1 0x2 : SOFn1 0x3 : SOFi2 0x4 : SOFn2 0x5 : SOFi3 0x6 : SOFn3 0x7 : SOFf
Тип EOF пакета	[28:26]	0x0, 0x6, 0x7 : EOFt 0x1 : EOFdt 0x2 : EOFa 0x3 : EOFn 0x4 : EOFni 0x5 : EOFdti
Длина пакета	[25:6]	Длина пакета данных (Frame) в байтах, должна быть кратна 4б
Act2	[5]	Тип дескриптора DMA 01,00,11 compl Дескриптор выполнен (отработан контроллером FC) 10 tran Дескриптор активен (в очереди на исполнение контроллером FC)
Act1	[4]	
-	[3]	Не используется, необходимо всегда записывать 0
Int	[2]	1 - Выработка прерывания по завершении выполнения
-	[1:0]	Не используется, необходимо всегда записывать 0

1.3.10.2.3.5 Работа с контроллером Fiber Channel

Основной алгоритм работы с контроллером Fiber Channel

- Настроить SERDES PHY. В общем случае: снять все сигналы сброса, выставить скорость работы (делитель опорной частоты), включить SERDES PHY, убедиться, что стабильны все синхросигналы (PLL lock).
- Сформировать в системной памяти таблицу дескрипторов для канала приема данных RX; настроить необходимые параметры ADVANCED AXI DMA; указать начальный адрес таблицы дескрипторов и ее размер (регистры RDMA_SYS_ADDR, RDMA_TBL_SIZE); включить канал приема данных RX с поддержкой внешней таблицы дескрипторов (RDMA _SETTINGS[29:28] =0x3).
- Прописать значения счетчиков и параметров, если не устраивают те, что по умолчанию. Значения следующих параметров могут быть установлены только при выключенной подсистеме FC_PORT: TIMER5VAL, RTTOVVAL, EDTOVVAL, LSYNCLIMIT, MAXFLEN.

- Включить подсистему FC_PORT контроллера Fiber Channel (CONTROL[0]=1) и дождаться статуса FC_PORT_OK (STATUS[0]=1, возможна выработка прерывания по этому событию). После этого контроллер готов к приему и передаче пакетов (Frames) Fiber Channel. Также возможно программное управление соединением в режиме pass-through (бит CONTROL[1] включает режим pass-through, биты CONTROL[12:2] позволяют управлять соединением, отслеживая состояние соединения с помощью регистра STATUS).
- Контроллер находится в рабочем режиме. Можно передавать и принимать пакеты. Для передачи пакетов необходимо сформировать в системной памяти таблицу дескрипторов для канала передачи данных TX, указать начальный адрес этой таблицы и ее размер (регистры WDMA_SYS_ADDR, WDMA_TBL_SIZE), и включить канал передачи TX (WDMA_SETTINGS[29:28] =0x3). Для обслуживания таблиц дескрипторов и отслеживания состояния контроллера Fiber Channel можно использовать прерывания (предварительно настроив маски в регистрах IRQMASKS, RDMA _SETTINGS и WDMA_SETTINGS).
- Для отслеживания состояния таблиц дескрипторов каналов RX и TX возможно периодически устанавливать в 1 поля Int дескрипторов и разрешить выработку необходимых прерываний в регистре RDMA _SETTINGS и WDMA_SETTINGS.
- После прохождения процедуры Fabric Login можно установить требуемый режим и параметры работы механизма b2b flow control.

1.3.10.2.3.5.1 Работа с механизмом buffer to buffer flow control

В контроллере Fiber Channel предусмотрено 2 режима работы механизма b2b flow control: программный (B2BCONTROL1[24]=1) и аппаратный (B2BCONTROL1[24]=0). В аппаратном режиме сигнал BBRDY всегда отправляется сразу после получения пакета – предполагается, что программный буфер условно бесконечный. Программный режим работы предполагает инициацию отправки сигналов BBRDY записью в регистр B2BCONTROL2, количество отправляемых сигналов BBRDY при этом устанавливается в поле B2BCONTROL2[15:0]. При использовании любого из режимов значение BB_Credit ответной части (по умолчанию после сброса равно 1), подключенной к данному контроллеру Fiber Channel, необходимо записать в регистр B2BCONTROL1[15:0].

Также контроллер Fiber Channel поддерживает механизм b2b credit recovery в соответствии со стандартом FC-FS-2. Для включения механизма b2b credit recovery необходимо записать отличные от 0 значения BB_SC_N данного контроллера и ответной части в поля B2BCONTROL1[19:16] и B2BCONTROL1[23:20] соответственно.

1.3.10.2.3.5.2 Обработка исключений

В случае возникновения исключений (событий регистра STATUS[3:1], а также изменения состояния из 1 в 0 STATUS[0]) рекомендуется отключить канал передачи TX ADVANCED AXI DMA и перезапустить подсистему FC_PORT: CONTROL[0]=0, затем CONTROL[0]=1.

При возникновении ошибок подсистемы ADVANCED AXI DMA (бит 5 регистров RDMA_STATUS и WDMA_STATUS) рекомендуется выполнить полный программный сброс контроллера Fiber Channel.

1.3.10.2.3.5.3 Программный сброс

Для полного программного сброса контроллера Fiber Channel необходимо отдельно сбросить подсистемы FC_FRONTEND (регистр SOFTRST) и ADVANCED AXI DMA (регистр DMA_SW_RESET).

1.3.11 Подсистема низкоскоростных интерфейсов LSIF0

1.3.11.1 Описание подсистемы LSIF0

1.3.11.1.1 Структурная схема и состав блоков подсистемы LSIF0

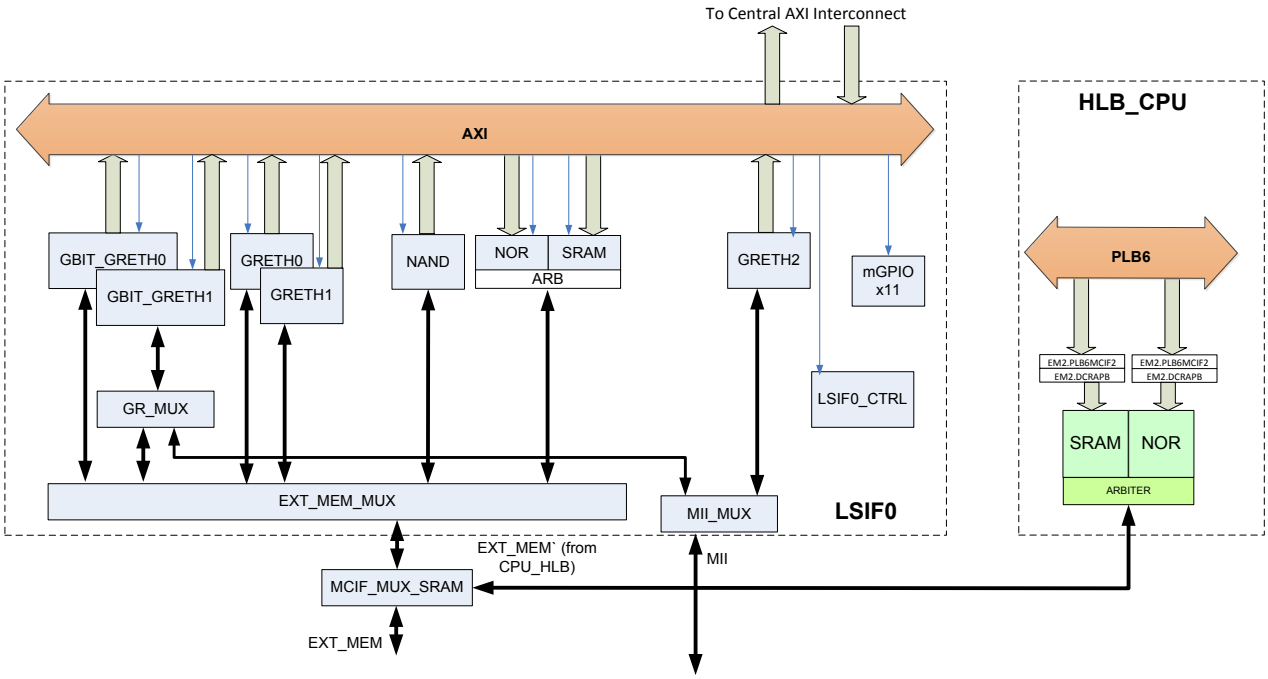


Рисунок 1.149 - Структурная схема подсистемы LSIF0

Подсистема LSIF0 состоит из следующих блоков:

- Два контроллера Ethernet 10/100/1000 (GBIT_GRETH0, GBIT_GRETH1)
- Три контроллера Ethernet 10/10 (GRETH0, GRETH1, GRETH2)
- Контроллер NAND флеш-памяти с поддержкой ECC (описание приведено в разделе 1.3.11.2.3).
- Два контроллера SRAM/NOR-памяти с интерфейсом AXI (SRAM_NOR_1 и SRAM_NOR_2, описание представлено в разделе 1.3.11.2.4).
- Одиннадцать контроллеров мультиплексируемых портов программируемого ввода-вывода (mGPIO)
- Схемы мультиплексирования GR_MUX, EXT_MEM_MUX и MII_MUX (описание приведено в разделе 0).
- Блок управляющих регистров LSIF0_CTRL (описание приведено в разделе 1.3.12.2.1)

Дополнительно с портами подсистемы LSIF0 мультиплексируются порты контроллера SRAM/NOR-памяти с интерфейсом MCIF, расположенного в подсистеме CPU (SRAM_NOR_MCIF_1 и SRAM_NOR_MCIF_2, описание приведено в разделе 1.3.11.2.5). Описание мультиплексирования (MCIF_MUX_SRAM) приведено в разделе 1.3.11.1.5.

1.3.11.1.2 Адресация блоков подсистемы LSIF0 и SRAM/NOR контроллеров подсистемы CPU

Таблица 1.782 - Карта адресов подсистемы LSIF0.

Блок	Адрес на шине APB
SRAM/NOR 1 AXI	0x3C030000 – 0x3C030FFF
SRAM/NOR 2 AXI	0x3C031000 – 0x3C031FFF
NAND	0x3C032000 – 0x3C032FFF
GBIT_GRETH0	0x3C033000 – 0x3C033FFF
GBIT_GRETH1	0x3C034000 – 0x3C034FFF
GRETH0	0x3C035000 – 0x3C035FFF
GRETH1	0x3C036000 – 0x3C036FFF
GRETH2	0x3C037000 – 0x3C037FFF
LSIF0_CTRL	0x3C03F000 – 0x3C03FFFF
mGPIO x 11	0x3C040000 – 0x3C04AFFF (APB x 11)

Таблица 1.783 - Карта адресов контроллеров SRAM/NOR памяти с системным интерфейсом MCIF в подсистеме CPU.

Блок	Адрес на шине DCR
EM2.PLB6MCIF2	0x80160000 - 0x8016FFFF
SRAM/NOR 1 MCIF	0x80170000 - 0x8017FFFF
EM3.PLB6MCIF2	0x80180000 - 0x8018FFFF
SRAM/NOR 2 MCIF	0x80190000 - 0x8019FFFF

1.3.11.1.3 Блок управляющих регистров LSIF0_CTRL

Блок управляющих регистров LSIF0_CTRL подсистемы LSIF0 микросхемы предназначен для настройки работы подсистемы. Блок позволяет управлять следующими функциями:

- Выбор режима работы контроллеров:
  - SRAM/NOR-памяти;
  - NAND-памяти;
  - Gigabit Ethernet (GBIT_GRETH - имеющим выход на GMII-интерфейс);
  - Ethernet (GRETH - имеющим выход на интерфейсы RMII).
- Управление доступом двух контроллеров SRAM/NOR-памяти к внешних портам.
- Расширение адресного пространства для контроллеров:
  - NAND флеш-памяти;
  - GBIT_GRETH0 и 1;
  - GRETH0,1,2.

Таблица 1.784 – Карта регистров блока LSIF0_CTRL

Название	Адрес	Доступ	Значение после сброса	Описание
ID	0x0	RO	0x3046494c	Регистр идентификации
CE_MANAGE	0x04	RW	0x0	Распределение линий CEn между контроллерами SRAM_NOR_1 и SRAM_NOR_2
GBIT_GRETH0_ADDR_MSB	0x08	RW	0x0	Расширение адреса для 36-разрядной адресации контроллера GBIT_GRETH0: Биты [3:0] - соответствуют 4 старшим разряда адреса
-	0x0C	-	-	-

Продолжение таблицы 1.785

Название	Адрес	Доступ	Значение после сброса	Описание
GBIT_GRETH1_ADDR_MSB	0x10	RW	0x0	Расширение адреса для 36-разрядной адресации контроллера GBIT_GRETH1: Биты [3:0] - соответствуют 4 старшим разряда адреса
GRETH0_ADDR_MSB	0x14	RW	0x0	Расширение адреса для 36-разрядной адресации контроллера GRETH0: Биты [3:0] - соответствуют 4 старшим разряда адреса
GRETH1_ADDR_MSB	0x18	RW	0x0	Расширение адреса для 36-разрядной адресации контроллера GRETH1: Биты [3:0] - соответствуют 4 старшим разряда адреса
-	0x1C	-	-	-
GRETH2_ADDR_MSB	0x20	RW	0x0	Расширение адреса для 36-разрядной адресации контроллера GRETH2: Биты [3:0] - соответствуют 4 старшим разряда адреса
NAND_RADDR_EXTEND	0x24	RW	0x0	Расширение адреса для 36-разрядной адресации канала чтения из NAND-памяти: Биты [3:0] - соответствуют 4 старшим разряда адреса
NAND_WADDR_EXTEND	0x28	RW	0x0	Расширение адреса для 36-разрядной адресации канала записи в NAND-память: Биты [3:0] - соответствуют 4 старшим разряда адреса
-	0x2C	-	-	-
EXT_MEM_MUX_MODE	0x30	RW	0x0	Выбор типа используемых контроллеров для выводов EXT_MEM
MII_MUX_MODE	0x34	RW	0x0	Выбор типа используемых контроллеров для выводов MII

1.3.11.1.3.1 Поля регистра CE_MANAGE

Определяет количество линий CEn доступных контролеру SRAM_NOR_2, остальные линии доступны контроллеру SRAM_NOR_1.

Таблица 1.786 – Поля регистра CE_MANAGE

Биты	Поле	Знач. по умолч.	Описание
2:0	manage_1	0x0	Доступность портов для контроллера 1
31:3	-	-	reserve

В поле manage_1 записывается количество выходов CEn_o доступных для SRAM_NOR_1. Для SRAM_NOR_2 будут доступны все остальные выходы CEn_o.

Значение 7 запрещает доступ к внешним микросхемам памяти.

Таблица 1.787 – Доступность выводов EXT_MEM_CEx для контроллеров SRAM_NOR_1 и SRAM_NOR_2

Значение CE_ MANAGE	EXT_MEM_ CE5	EXT_MEM_ CE4	EXT_MEM_ CE3	EXT_MEM_ CE2	EXT_MEM_ CE1	EXT_MEM_ CE0
0	1*	1	1	1	1	1
1	2**	1	1	1	1	1
2	2	2	1	1	1	1
3	2	2	2	1	1	1
4	2	2	2	2	1	1
5	2	2	2	2	2	1
6	2	2	2	2	2	2
7	нет	нет	нет	нет	нет	нет

* - Контроллер 1
** - Контроллер 2

1.3.11.1.3.2 Поля регистра EXT_MEM_MUX_MODE

Регистр определяет, какие контроллеры будут использовать внешние выводы EXT_MEM

Таблица 1.788 – Поля регистра EXT_MEM_MUX_MODE

Биты	Поле	Знач. по умолч.	Описание
1:0	ext_mem_mux_mode	0x0	Выбор контроллеров, использующих внешние выводы EXT_MEM: 2'b00 - SRAM_NOR; 2'b01 - NAND, GBIT_GRETH0, GBIT_GRETH1 (при mii_mux_mode = 1, см таблицу Таблица 1.789); 2'b10 - GRETH0, GRETH1, NAND; 2'b11 - reserved. Более подробная схема мультиплексирования выводов приведена в разделе 0
31:2	-	-	reserve

1.3.11.1.3.3 Поля регистра MII_MUX_MODE

Определяет, какие контроллеры будут использовать внешние выводы MII

Таблица 1.789 – Поля регистра MII_MUX_MODE

Биты	Поле	Знач. по умолч.	Описание
0	mii_mux_mode	0x0	Выбор контроллеров, использующих внешние выводы MII: 0 - GBIT_GRETH1; 1 - GRETH2. Более подробная схема мультиплексирования выводов приведена в разделе 0
31:1	-	-	reserve

1.3.11.1.4 Схема мультиплексирования портов интерфейсных контроллеров подсистемы LSIF0

Таблица 1.790 – Схема мультиплексирования портов интерфейсных контроллеров подсистемы LSIF0.

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_D0	in	SRAM_NOR/D0_i	GBIT_GRETH0/rx_dv	-	MGPIO0/GPIN[0]
	out	SRAM_NOR/D0_o	-	-	MGPIO0/GPOUT[0]
EXT_MEM_D1	in	SRAM_NOR/D1_i	GBIT_GRETH0/rx_er	GRETH1/rx_er	MGPIO0/GPIN[1]
	out	SRAM_NOR/D1_o	-	-	MGPIO0/GPOUT[1]
EXT_MEM_D2	in	SRAM_NOR/D2_i	GBIT_GRETH0/rx_col	-	MGPIO0/GPIN[2]
	out	SRAM_NOR/D2_o	-	-	MGPIO0/GPOUT[2]
EXT_MEM_D3	in	SRAM_NOR/D3_i	GBIT_GRETH0/rx_crs	GRETH1/rx_crs	MGPIO0/GPIN[3]
	out	SRAM_NOR/D3_o	-	-	MGPIO0/GPOUT[3]
EXT_MEM_D4	in	SRAM_NOR/D4_i	GBIT_GRETH0/rxd[0]	GRETH1/rxd[0]	MGPIO0/GPIN[4]
	out	SRAM_NOR/D4_o	-	-	MGPIO0/GPOUT[4]
EXT_MEM_D5	in	SRAM_NOR/D5_i	GBIT_GRETH0/rxd[1]	GRETH1/rxd[1]	MGPIO0/GPIN[5]
	out	SRAM_NOR/D5_o	-	-	MGPIO0/GPOUT[5]
EXT_MEM_D6	in	SRAM_NOR/D6_i	GBIT_GRETH0/rxd[2]	-	MGPIO0/GPIN[6]
	out	SRAM_NOR/D6_o	-	-	MGPIO0/GPOUT[6]
EXT_MEM_D7	in	SRAM_NOR/D7_i	GBIT_GRETH0/rxd[3]	-	MGPIO0/GPIN[7]
	out	SRAM_NOR/D7_o	-	-	MGPIO0/GPOUT[7]
EXT_MEM_D8	in	SRAM_NOR/D8_i	GBIT_GRETH0/rxd[4]	-	MGPIO1/GPIN[0]
	out	SRAM_NOR/D8_o	-	-	MGPIO1/GPOUT[0]
EXT_MEM_D9	in	SRAM_NOR/D9_i	GBIT_GRETH0/rxd[5]	-	MGPIO1/GPIN[1]
	out	SRAM_NOR/D9_o	-	-	MGPIO1/GPOUT[1]
EXT_MEM_D10	in	SRAM_NOR/D10_i	GBIT_GRETH0/rxd[6]	-	MGPIO1/GPIN[2]
	out	SRAM_NOR/D10_o	-	-	MGPIO1/GPOUT[2]
EXT_MEM_D11	in	SRAM_NOR/D11_i	GBIT_GRETH0/rxd[7]	-	MGPIO1/GPIN[3]
	out	SRAM_NOR/D11_o	-	-	MGPIO1/GPOUT[3]
EXT_MEM_D12	in	SRAM_NOR/D12_i	GBIT_GRETH0/tx_clk	GRETH1/rmii_clk	-
	out	SRAM_NOR/D12_o	-	-	-
EXT_MEM_D13	in	SRAM_NOR/D13_i	GBIT_GRETH0/rx_clk	-	-
	out	SRAM_NOR/D13_o	-	-	-
EXT_MEM_D14	in	SRAM_NOR/D14_i	GBIT_GRETH1/rx_dv	-	MGPIO1/GPIN[4]
	out	SRAM_NOR/D14_o	-	-	MGPIO1/GPOUT[4]
EXT_MEM_D15	in	SRAM_NOR/D15_i	GBIT_GRETH1/rx_er	GRETH0/rx_er	MGPIO1/GPIN[5]
	out	SRAM_NOR/D15_o	-	-	MGPIO1/GPOUT[5]
EXT_MEM_D16	in	SRAM_NOR/D16_i	GBIT_GRETH1/rx_col	-	MGPIO1/GPIN[6]
	out	SRAM_NOR/D16_o	-	-	MGPIO1/GPOUT[6]
EXT_MEM_D17	in	SRAM_NOR/D17_i	GBIT_GRETH1/rx_crs	GRETH0/rx_crs	MGPIO1/GPIN[7]
	out	SRAM_NOR/D17_o	-	-	MGPIO1/GPOUT[7]
EXT_MEM_D18	in	SRAM_NOR/D18_i	GBIT_GRETH1/rxd[0]	GRETH0/rxd[0]	MGPIO2/GPIN[0]
	out	SRAM_NOR/D18_o	-	-	MGPIO2/GPOUT[0]
EXT_MEM_D19	in	SRAM_NOR/D19_i	GBIT_GRETH1/rxd[1]	GRETH0/rxd[1]	MGPIO2/GPIN[1]
	out	SRAM_NOR/D19_o	-	-	MGPIO2/GPOUT[1]
EXT_MEM_D20	in	SRAM_NOR/D20_i	GBIT_GRETH1/rxd[2]	-	MGPIO2/GPIN[2]
	out	SRAM_NOR/D20_o	-	-	MGPIO2/GPOUT[2]
EXT_MEM_D21	in	SRAM_NOR/D21_i	GBIT_GRETH1/rxd[3]	-	MGPIO2/GPIN[3]
	out	SRAM_NOR/D21_o	-	-	MGPIO2/GPOUT[3]
EXT_MEM_D22	in	SRAM_NOR/D22_i	GBIT_GRETH1/rxd[4]	-	MGPIO2/GPIN[4]
	out	SRAM_NOR/D22_o	-	-	MGPIO2/GPOUT[4]
EXT_MEM_D23	in	SRAM_NOR/D23_i	GBIT_GRETH1/rxd[5]	-	MGPIO2/GPIN[5]
	out	SRAM_NOR/D23_o	-	-	MGPIO2/GPOUT[5]
EXT_MEM_D24	in	SRAM_NOR/D24_i	GBIT_GRETH1/rxd[6]	-	MGPIO2/GPIN[6]
	out	SRAM_NOR/D24_o	-	-	MGPIO2/GPOUT[6]

Продолжение таблицы 1.791

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_D25	in	SRAM_NOR/D25_i	GBIT_GRETH1/rxd[7]	-	MGPIO2/GPIN[7]
	out	SRAM_NOR/D25_o	-	-	MGPIO2/GPOUT[7]
EXT_MEM_D26	in	SRAM_NOR/D26_i	GBIT_GRETH1/tx_clk	GRETH0/rmii_clk	-
	out	SRAM_NOR/D26_o	-	-	-
EXT_MEM_D27	in	SRAM_NOR/D27_i	GBIT_GRETH1/rx_clk	-	-
	out	SRAM_NOR/D27_o	-	-	-
EXT_MEM_D28	in	SRAM_NOR/D28_i	NAND/din0	NAND/din0	MGPIO3/GPIN[0]
	out	SRAM_NOR/D28_o	NAND/dout0	NAND/dout0	MGPIO3/GPOUT[0]
EXT_MEM_D29	in	SRAM_NOR/D29_i	NAND/din1	NAND/din1	MGPIO3/GPIN[1]
	out	SRAM_NOR/D29_o	NAND/dout1	NAND/dout1	MGPIO3/GPOUT[1]
EXT_MEM_D30	in	SRAM_NOR/D30_i	NAND/din2	NAND/din2	MGPIO3/GPIN[2]
	out	SRAM_NOR/D30_o	NAND/dout2	NAND/dout2	MGPIO3/GPOUT[2]
EXT_MEM_D31	in	SRAM_NOR/D31_i	NAND/din3	NAND/din3	MGPIO3/GPIN[3]
	out	SRAM_NOR/D31_o	NAND/dout3	NAND/dout3	MGPIO3/GPOUT[3]
EXT_MEM_D32	in	SRAM_NOR/D32_i	NAND/din4	NAND/din4	MGPIO3/GPIN[4]
	out	SRAM_NOR/D32_o	NAND/dout4	NAND/dout4	MGPIO3/GPOUT[4]
EXT_MEM_D33	in	SRAM_NOR/D33_i	NAND/din5	NAND/din5	MGPIO3/GPIN[5]
	out	SRAM_NOR/D33_o	NAND/dout5	NAND/dout5	MGPIO3/GPOUT[5]
EXT_MEM_D34	in	SRAM_NOR/D34_i	NAND/din6	NAND/din6	MGPIO3/GPIN[6]
	out	SRAM_NOR/D34_o	NAND/dout6	NAND/dout6	MGPIO3/GPOUT[6]
EXT_MEM_D35	in	SRAM_NOR/D35_i	NAND/din7	NAND/din7	MGPIO3/GPIN[7]
	out	SRAM_NOR/D35_o	NAND/dout7	NAND/dout7	MGPIO3/GPOUT[7]
EXT_MEM_D36	in	SRAM_NOR/D36_i	NAND/ready1n	NAND/ready1n	MGPIO4/GPIN[0]
	out	SRAM_NOR/D36_o	-	-	MGPIO4/GPOUT[0]
EXT_MEM_D37	in	SRAM_NOR/D37_i	NAND/ready2n	NAND/ready2n	MGPIO4/GPIN[1]
	out	SRAM_NOR/D37_o	-	-	MGPIO4/GPOUT[1]
EXT_MEM_D38	in	SRAM_NOR/D38_i	-	-	MGPIO4/GPIN[2]
	out	SRAM_NOR/D38_o	GBIT_GRETH0/tx_en	GRETH1/tx_en	MGPIO4/GPOUT[2]
EXT_MEM_D39	in	SRAM_NOR/D39_i	-	-	MGPIO4/GPIN[3]
	out	SRAM_NOR/D39_o	GBIT_GRETH1/tx_en	GRETH0/tx_en	MGPIO4/GPOUT[3]
EXT_MEM_A0	in	-	-	-	MGPIO4/GPIN[4]
	out	SRAM_NOR/A0_o	-	-	MGPIO4/GPOUT[4]
EXT_MEM_A1	in	-	-	-	MGPIO4/GPIN[5]
	out	SRAM_NOR/A1_o	GBIT_GRETH0/tx_er	-	MGPIO4/GPOUT[5]
EXT_MEM_A2	in	-	-	-	MGPIO4/GPIN[6]
	out	SRAM_NOR/A2_o	GBIT_GRETH0/txd[0]	GRETH1/txd[0]	MGPIO4/GPOUT[6]
EXT_MEM_A3	in	-	-	-	MGPIO4/GPIN[7]
	out	SRAM_NOR/A3_o	GBIT_GRETH0/txd[1]	GRETH1/txd[1]	MGPIO4/GPOUT[7]
EXT_MEM_A4	in	-	-	-	MGPIO5/GPIN[0]
	out	SRAM_NOR/A4_o	GBIT_GRETH0/txd[2]	-	MGPIO5/GPOUT[0]
EXT_MEM_A5	in	-	-	-	MGPIO5/GPIN[1]
	out	SRAM_NOR/A5_o	GBIT_GRETH0/txd[3]	-	MGPIO5/GPOUT[1]
EXT_MEM_A6	in	-	-	-	MGPIO5/GPIN[2]
	out	SRAM_NOR/A6_o	GBIT_GRETH0/txd[4]	-	MGPIO5/GPOUT[2]
EXT_MEM_A7	in	-	-	-	MGPIO5/GPIN[3]
	out	SRAM_NOR/A7_o	GBIT_GRETH0/txd[5]	-	MGPIO5/GPOUT[3]
EXT_MEM_A8	in	-	-	-	MGPIO5/GPIN[4]
	out	SRAM_NOR/A8_o	GBIT_GRETH0/txd[6]	-	MGPIO5/GPOUT[4]
EXT_MEM_A9	in	-	-	-	MGPIO5/GPIN[5]
	out	SRAM_NOR/A9_o	GBIT_GRETH0/txd[7]	-	MGPIO5/GPOUT[5]

Продолжение таблицы 1.792

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_A10	in	-	-	-	-
	out	SRAM_NOR/A10_o	GBIT_GRETH0/gtx_clk	-	-
EXT_MEM_A11	in	-	-	-	MGPIO5/GPIN[6]
	out	SRAM_NOR/A11_o	GBIT_GRETH0/mdc	GRETH1/mdc	MGPIO5/GPOUT[6]
EXT_MEM_A12	in	-	GBIT_GRETH0/mdio_i	GRETH1/mdio_i	MGPIO5/GPIN[7]
	out	SRAM_NOR/A12_o	GBIT_GRETH0/mdio_o	GRETH1/mdio_o	MGPIO5/GPOUT[7]
EXT_MEM_A13	in	-	-	-	MGPIO6/GPIN[0]
	out	SRAM_NOR/A13_o	-	-	MGPIO6/GPOUT[0]
EXT_MEM_A14	in	-	-	-	MGPIO6/GPIN[1]
	out	SRAM_NOR/A14_o	GBIT_GRETH1/tx_er	-	MGPIO6/GPOUT[1]
EXT_MEM_A15	in	-	-	-	MGPIO6/GPIN[2]
	out	SRAM_NOR/A15_o	GBIT_GRETH1/txd[0]	GRETH0/txd[0]	MGPIO6/GPOUT[2]
EXT_MEM_A16	in	-	-	-	MGPIO6/GPIN[3]
	out	SRAM_NOR/A16_o	GBIT_GRETH1/txd[1]	GRETH0/txd[1]	MGPIO6/GPOUT[3]
EXT_MEM_A17	in	-	-	-	MGPIO6/GPIN[4]
	out	SRAM_NOR/A17_o	GBIT_GRETH1/txd[2]	-	MGPIO6/GPOUT[4]
EXT_MEM_A18	in	-	-	-	MGPIO6/GPIN[5]
	out	SRAM_NOR/A18_o	GBIT_GRETH1/txd[3]	-	MGPIO6/GPOUT[5]
EXT_MEM_A19	in	-	-	-	MGPIO6/GPIN[6]
	out	SRAM_NOR/A19_o	GBIT_GRETH1/txd[4]	-	MGPIO6/GPOUT[6]
EXT_MEM_A20	in	-	-	-	MGPIO6/GPIN[7]
	out	SRAM_NOR/A20_o	GBIT_GRETH1/txd[5]	-	MGPIO6/GPOUT[7]
EXT_MEM_A21	in	-	-	-	MGPIO7/GPIN[0]
	out	SRAM_NOR/A21_o	GBIT_GRETH1/txd[6]	-	MGPIO7/GPOUT[0]
EXT_MEM_A22	in	-	-	-	MGPIO7/GPIN[1]
	out	SRAM_NOR/A22_o	GBIT_GRETH1/txd[7]	-	MGPIO7/GPOUT[1]
EXT_MEM_A23	in	-	-	-	-
	out	SRAM_NOR/A23_o	GBIT_GRETH1/gtx_clk	-	-
EXT_MEM_A24	in	-	-	-	MGPIO7/GPIN[2]
	out	SRAM_NOR/A24_o	GBIT_GRETH1/mdc	GRETH0/mdc	MGPIO7/GPOUT[2]
EXT_MEM_A25	in	-	GBIT_GRETH1/mdio_i	GRETH0/mdio_i	MGPIO7/GPIN[3]
	out	SRAM_NOR/A25_o	GBIT_GRETH1/mdio_o	GRETH0/mdio_o	MGPIO7/GPOUT[3]
EXT_MEM_CE0	in	-	-	-	MGPIO7/GPIN[4]
	out	SRAM_NOR/CE0	NAND/ren	NAND/ren	MGPIO7/GPOUT[4]
EXT_MEM_CE1	in	-	-	-	MGPIO7/GPIN[5]
	out	SRAM_NOR/CE1	NAND/cle	NAND/cle	MGPIO7/GPOUT[5]
EXT_MEM_CE2	in	-	-	-	MGPIO7/GPIN[6]
	out	SRAM_NOR/CE2	NAND/ce2n	NAND/ce2n	MGPIO7/GPOUT[6]
EXT_MEM_CE3	in	-	-	-	MGPIO7/GPIN[7]
	out	SRAM_NOR/CE3	NAND/ce1n	NAND/ce1n	MGPIO7/GPOUT[7]
EXT_MEM_CE4	in	-	-	-	MGPIO8/GPIN[0]
	out	SRAM_NOR/CE4	NAND/ale	NAND/ale	MGPIO8/GPOUT[0]
EXT_MEM_CE5	in	-	-	-	MGPIO8/GPIN[1]
	out	SRAM_NOR/CE5	NAND/wen	NAND/wen	MGPIO8/GPOUT[1]
EXT_MEM_BHE_H	in	-	-	-	MGPIO8/GPIN[2]
	out	SRAM_NOR/ BHE_H	-	-	MGPIO8/GPOUT[2]
EXT_MEM_BHE_L	in	-	-	-	MGPIO8/GPIN[3]
	out	SRAM_NOR/ BHE_L	-	-	MGPIO8/GPOUT[3]
EXT_MEM_BLE_H	in	-	-	-	MGPIO8/GPIN[4]
	out	SRAM_NOR/ BLE_H	-	-	MGPIO8/GPOUT[4]
EXT_MEM_BLE_L	in	-	-	-	MGPIO8/GPIN[5]
	out	SRAM_NOR/ BLE_L	-	-	MGPIO8/GPOUT[5]

Продолжение таблицы 1.793

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_OE	in	-	-	-	MGPIO8/GPIN[6]
	out	SRAM_NOR/OE	-	-	MGPIO8/GPOUT[6]
EXT_MEM_WE	in	-	-	-	MGPIO8/GPIN[7]
	out	SRAM_NOR/WE	NAND/wpn	NAND/wpn	MGPIO8/GPOUT[7]

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме MII_MODE:		Подключение в GPIO режиме
		1'b0	1'b1	
MII_RX_CLK	in	GBIT_GRETH1/rx_clk	GRETH2/rmii_clk	-
	out	-	-	-
MII_RX_COL	in	GBIT_GRETH1/rx_col	-	MGPIO9/GPIN[0]
	out	-	-	MGPIO9/GPOUT[0]
MII_RX_CRS	in	GBIT_GRETH1/rx_crs	GRETH2/rx_crs	MGPIO9/GPIN[1]
	out	-	-	MGPIO9/GPOUT[1]
MII_RXD0	in	GBIT_GRETH1/rxd[0]	GRETH2/rxd[0]	MGPIO9/GPIN[2]
	out	-	-	MGPIO9/GPOUT[2]
MII_RXD1	in	GBIT_GRETH1/rxd[1]	GRETH2/rxd[1]	MGPIO9/GPIN[3]
	out	-	-	MGPIO9/GPOUT[3]
MII_RXD2	in	GBIT_GRETH1/rxd[2]	-	MGPIO9/GPIN[4]
	out	-	-	MGPIO9/GPOUT[4]
MII_RXD3	in	GBIT_GRETH1/rxd[3]	-	MGPIO9/GPIN[5]
	out	-	-	MGPIO9/GPOUT[5]
MII_RX_ER	in	GBIT_GRETH1/rx_er	GRETH2/rx_er	MGPIO9/GPIN[6]
	out	-	-	MGPIO9/GPOUT[6]
MII_RX_DV	in	GBIT_GRETH1/rx_dv	-	MGPIO9/GPIN[7]
	out	-	-	MGPIO9/GPOUT[7]
MII_TX_CLK	in	GBIT_GRETH1/tx_clk	-	-
	out	-	-	-
MII_TX_EN	in	-	-	MGPIO10/GPIN[0]
	out	GBIT_GRETH1/tx_en	GRETH2/tx_en	MGPIO10/GPOUT[0]
MII_TX_ER	in	-	-	MGPIO10/GPIN[1]
	out	GBIT_GRETH1/tx_er	-	MGPIO10/GPOUT[1]
MII_TXD0	in	-	-	MGPIO10/GPIN[2]
	out	GBIT_GRETH1/txd[0]	GRETH2/txd[0]	MGPIO10/GPOUT[2]
MII_TXD1	in	-	-	MGPIO10/GPIN[3]
	out	GBIT_GRETH1/txd[1]	GRETH2/txd[1]	MGPIO10/GPOUT[3]
MII_TXD2	in	-	-	MGPIO10/GPIN[4]
	out	GBIT_GRETH1/txd[2]	-	MGPIO10/GPOUT[4]
MII_TXD3	in	-	-	MGPIO10/GPIN[5]
	out	GBIT_GRETH1/txd[3]	-	MGPIO10/GPOUT[5]
MII_MDC	in	-	-	MGPIO10/GPIN[6]
	out	GBIT_GRETH1/mdc	GRETH2/mdc	MGPIO10/GPOUT[6]
MII_MDIO	in	GBIT_GRETH1/mdio_i	GRETH2/mdio_i	MGPIO10/GPIN[7]
	out	GBIT_GRETH1/mdio_o	GRETH2/mdio_o	MGPIO10/GPOUT[7]

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_D0	in	SRAM_NOR/D0_i	GBIT_GRETH/rx_dv	-	MGPIO0/GPIN[0]
	out	SRAM_NOR/D0_o	-	-	MGPIO0/GPOUT[0]
EXT_MEM_D1	in	SRAM_NOR/D1_i	GBIT_GRETH/rx_er	GRETH1/rx_er	MGPIO0/GPIN[1]
	out	SRAM_NOR/D1_o	-	-	MGPIO0/GPOUT[1]

Продолжение таблицы 1.794

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_D2	in	SRAM_NOR/D2_i	GBIT_GRETH/rx_col	-	MGPIO0/GPIN[2]
	out	SRAM_NOR/D2_o	-	-	MGPIO0/GPOUT[2]
EXT_MEM_D3	in	SRAM_NOR/D3_i	GBIT_GRETH/rx_crs	GRETH1/rx_crs	MGPIO0/GPIN[3]
	out	SRAM_NOR/D3_o	-	-	MGPIO0/GPOUT[3]
EXT_MEM_D4	in	SRAM_NOR/D4_i	GBIT_GRETH/rxd[0]	GRETH1/rxd[0]	MGPIO0/GPIN[4]
	out	SRAM_NOR/D4_o	-	-	MGPIO0/GPOUT[4]
EXT_MEM_D5	in	SRAM_NOR/D5_i	GBIT_GRETH/rxd[1]	GRETH1/rxd[1]	MGPIO0/GPIN[5]
	out	SRAM_NOR/D5_o	-	-	MGPIO0/GPOUT[5]
EXT_MEM_D6	in	SRAM_NOR/D6_i	GBIT_GRETH/rxd[2]	-	MGPIO0/GPIN[6]
	out	SRAM_NOR/D6_o	-	-	MGPIO0/GPOUT[6]
EXT_MEM_D7	in	SRAM_NOR/D7_i	GBIT_GRETH/rxd[3]	-	MGPIO0/GPIN[7]
	out	SRAM_NOR/D7_o	-	-	MGPIO0/GPOUT[7]
EXT_MEM_D8	in	SRAM_NOR/D8_i	GBIT_GRETH/rxd[4]	-	MGPIO1/GPIN[0]
	out	SRAM_NOR/D8_o	-	-	MGPIO1/GPOUT[0]
EXT_MEM_D9	in	SRAM_NOR/D9_i	GBIT_GRETH/rxd[5]	-	MGPIO1/GPIN[1]
	out	SRAM_NOR/D9_o	-	-	MGPIO1/GPOUT[1]
EXT_MEM_D10	in	SRAM_NOR/D10_i	GBIT_GRETH/rxd[6]	-	MGPIO1/GPIN[2]
	out	SRAM_NOR/D10_o	-	-	MGPIO1/GPOUT[2]
EXT_MEM_D11	in	SRAM_NOR/D11_i	GBIT_GRETH/rxd[7]	-	MGPIO1/GPIN[3]
	out	SRAM_NOR/D11_o	-	-	MGPIO1/GPOUT[3]
EXT_MEM_D12	in	SRAM_NOR/D12_i	GBIT_GRETH/tx_clk	GRETH1/rmii_clk	-
	out	SRAM_NOR/D12_o	-	-	-
EXT_MEM_D13	in	SRAM_NOR/D13_i	GBIT_GRETH/rx_clk	-	-
	out	SRAM_NOR/D13_o	-	-	-
EXT_MEM_D14	in	SRAM_NOR/D14_i	GBIT_GRETH/rx_dv	-	MGPIO1/GPIN[4]
	out	SRAM_NOR/D14_o	-	-	MGPIO1/GPOUT[4]
EXT_MEM_D15	in	SRAM_NOR/D15_i	GBIT_GRETH/rx_er	GRETH0/rx_er	MGPIO1/GPIN[5]
	out	SRAM_NOR/D15_o	-	-	MGPIO1/GPOUT[5]
EXT_MEM_D16	in	SRAM_NOR/D16_i	GBIT_GRETH/rx_col	-	MGPIO1/GPIN[6]
	out	SRAM_NOR/D16_o	-	-	MGPIO1/GPOUT[6]
EXT_MEM_D17	in	SRAM_NOR/D17_i	GBIT_GRETH/rx_crs	GRETH0/rx_crs	MGPIO1/GPIN[7]
	out	SRAM_NOR/D17_o	-	-	MGPIO1/GPOUT[7]
EXT_MEM_D18	in	SRAM_NOR/D18_i	GBIT_GRETH/rxd[0]	GRETH0/rxd[0]	MGPIO2/GPIN[0]
	out	SRAM_NOR/D18_o	-	-	MGPIO2/GPOUT[0]
EXT_MEM_D19	in	SRAM_NOR/D19_i	GBIT_GRETH/rxd[1]	GRETH0/rxd[1]	MGPIO2/GPIN[1]
	out	SRAM_NOR/D19_o	-	-	MGPIO2/GPOUT[1]
EXT_MEM_D20	in	SRAM_NOR/D20_i	GBIT_GRETH/rxd[2]	-	MGPIO2/GPIN[2]
	out	SRAM_NOR/D20_o	-	-	MGPIO2/GPOUT[2]
EXT_MEM_D21	in	SRAM_NOR/D21_i	GBIT_GRETH/rxd[3]	-	MGPIO2/GPIN[3]
	out	SRAM_NOR/D21_o	-	-	MGPIO2/GPOUT[3]
EXT_MEM_D22	in	SRAM_NOR/D22_i	GBIT_GRETH/rxd[4]	-	MGPIO2/GPIN[4]
	out	SRAM_NOR/D22_o	-	-	MGPIO2/GPOUT[4]
EXT_MEM_D23	in	SRAM_NOR/D23_i	GBIT_GRETH/rxd[5]	-	MGPIO2/GPIN[5]
	out	SRAM_NOR/D23_o	-	-	MGPIO2/GPOUT[5]
EXT_MEM_D24	in	SRAM_NOR/D24_i	GBIT_GRETH/rxd[6]	-	MGPIO2/GPIN[6]
	out	SRAM_NOR/D24_o	-	-	MGPIO2/GPOUT[6]
EXT_MEM_D25	in	SRAM_NOR/D25_i	GBIT_GRETH/rxd[7]	-	MGPIO2/GPIN[7]
	out	SRAM_NOR/D25_o	-	-	MGPIO2/GPOUT[7]
EXT_MEM_D26	in	SRAM_NOR/D26_i	GBIT_GRETH/tx_clk	GRETH0/rmii_clk	-
	out	SRAM_NOR/D26_o	-	-	-
EXT_MEM_D27	in	SRAM_NOR/D27_i	GBIT_GRETH/rx_clk	-	-
	out	SRAM_NOR/D27_o	-	-	-

Продолжение таблицы 1.795

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_D28	in	SRAM_NOR/D28_i	NAND/din0	NAND/din0	MGPIO3/GPIN[0]
	out	SRAM_NOR/D28_o	NAND/dout0	NAND/dout0	MGPIO3/GPOUT[0]
EXT_MEM_D29	in	SRAM_NOR/D29_i	NAND/din1	NAND/din1	MGPIO3/GPIN[1]
	out	SRAM_NOR/D29_o	NAND/dout1	NAND/dout1	MGPIO3/GPOUT[1]
EXT_MEM_D30	in	SRAM_NOR/D30_i	NAND/din2	NAND/din2	MGPIO3/GPIN[2]
	out	SRAM_NOR/D30_o	NAND/dout2	NAND/dout2	MGPIO3/GPOUT[2]
EXT_MEM_D31	in	SRAM_NOR/D31_i	NAND/din3	NAND/din3	MGPIO3/GPIN[3]
	out	SRAM_NOR/D31_o	NAND/dout3	NAND/dout3	MGPIO3/GPOUT[3]
EXT_MEM_D32	in	SRAM_NOR/D32_i	NAND/din4	NAND/din4	MGPIO3/GPIN[4]
	out	SRAM_NOR/D32_o	NAND/dout4	NAND/dout4	MGPIO3/GPOUT[4]
EXT_MEM_D33	in	SRAM_NOR/D33_i	NAND/din5	NAND/din5	MGPIO3/GPIN[5]
	out	SRAM_NOR/D33_o	NAND/dout5	NAND/dout5	MGPIO3/GPOUT[5]
EXT_MEM_D34	in	SRAM_NOR/D34_i	NAND/din6	NAND/din6	MGPIO3/GPIN[6]
	out	SRAM_NOR/D34_o	NAND/dout6	NAND/dout6	MGPIO3/GPOUT[6]
EXT_MEM_D35	in	SRAM_NOR/D35_i	NAND/din7	NAND/din7	MGPIO3/GPIN[7]
	out	SRAM_NOR/D35_o	NAND/dout7	NAND/dout7	MGPIO3/GPOUT[7]
EXT_MEM_D36	in	SRAM_NOR/D36_i	NAND/ready1n	NAND/ready1n	MGPIO4/GPIN[0]
	out	SRAM_NOR/D36_o	-	-	MGPIO4/GPOUT[0]
EXT_MEM_D37	in	SRAM_NOR/D37_i	NAND/ready2n	NAND/ready2n	MGPIO4/GPIN[1]
	out	SRAM_NOR/D37_o	-	-	MGPIO4/GPOUT[1]
EXT_MEM_D38	in	SRAM_NOR/D38_i	-	-	MGPIO4/GPIN[2]
	out	SRAM_NOR/D38_o	GBIT_GRETH/tx_en	GRETH1/tx_en	MGPIO4/GPOUT[2]
EXT_MEM_D39	in	SRAM_NOR/D39_i	-	-	MGPIO4/GPIN[3]
	out	SRAM_NOR/D39_o	GBIT_GRETH/tx_en	GRETH0/tx_en	MGPIO4/GPOUT[3]
EXT_MEM_A0	in	-	-	-	MGPIO4/GPIN[4]
	out	SRAM_NOR/A0_o	-	-	MGPIO4/GPOUT[4]
EXT_MEM_A1	in	-	-	-	MGPIO4/GPIN[5]
	out	SRAM_NOR/A1_o	GBIT_GRETH0/tx_er	-	MGPIO4/GPOUT[5]
EXT_MEM_A2	in	-	-	-	MGPIO4/GPIN[6]
	out	SRAM_NOR/A2_o	GBIT_GRETH0/txd[0]	GBIT_GRETH1/txd[0]	MGPIO4/GPOUT[6]
EXT_MEM_A3	in	-	-	-	MGPIO4/GPIN[7]
	out	SRAM_NOR/A3_o	GBIT_GRETH0/txd[1]	GBIT_GRETH1/txd[1]	MGPIO4/GPOUT[7]
EXT_MEM_A4	in	-	-	-	MGPIO5/GPIN[0]
	out	SRAM_NOR/A4_o	GBIT_GRETH0/txd[2]	-	MGPIO5/GPOUT[0]
EXT_MEM_A5	in	-	-	-	MGPIO5/GPIN[1]
	out	SRAM_NOR/A5_o	GBIT_GRETH0/txd[3]	-	MGPIO5/GPOUT[1]
EXT_MEM_A6	in	-	-	-	MGPIO5/GPIN[2]
	out	SRAM_NOR/A6_o	GBIT_GRETH0/txd[4]	-	MGPIO5/GPOUT[2]
EXT_MEM_A7	in	-	-	-	MGPIO5/GPIN[3]
	out	SRAM_NOR/A7_o	GBIT_GRETH0/txd[5]	-	MGPIO5/GPOUT[3]
EXT_MEM_A8	in	-	-	-	MGPIO5/GPIN[4]
	out	SRAM_NOR/A8_o	GBIT_GRETH0/txd[6]	-	MGPIO5/GPOUT[4]
EXT_MEM_A9	in	-	-	-	MGPIO5/GPIN[5]
	out	SRAM_NOR/A9_o	GBIT_GRETH0/txd[7]	-	MGPIO5/GPOUT[5]
EXT_MEM_A10	in	-	-	-	-
	out	SRAM_NOR/A10_o	GBIT_GRETH0/gtx_clk	-	-
EXT_MEM_A11	in	-	-	-	MGPIO5/GPIN[6]
	out	SRAM_NOR/A11_o	GBIT_GRETH0/mdc	GBIT_GRETH1/mdc	MGPIO5/GPOUT[6]

Продолжение таблицы 1.796

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_A12	in	-	GBIT_GRETH0/mdio_i	GBIT_GRETH1/mdio_i	MGPIO5/GPIN[7]
	out	SRAM_NOR/A12_o	GBIT_GRETH0/mdio_o	GBIT_GRETH1/mdio_o	MGPIO5/GPOUT[7]
EXT_MEM_A13	in	-	-	-	MGPIO6/GPIN[0]
	out	SRAM_NOR/A13_o	-	-	MGPIO6/GPOUT[0]
EXT_MEM_A14	in	-	-	-	MGPIO6/GPIN[1]
	out	SRAM_NOR/A14_o	GBIT_GRETH1/tx_er	-	MGPIO6/GPOUT[1]
EXT_MEM_A15	in	-	-	-	MGPIO6/GPIN[2]
	out	SRAM_NOR/A15_o	GBIT_GRETH1/txd[0]	GBIT_GRETH0/txd[0]	MGPIO6/GPOUT[2]
EXT_MEM_A16	in	-	-	-	MGPIO6/GPIN[3]
	out	SRAM_NOR/A16_o	GBIT_GRETH1/txd[1]	GBIT_GRETH0/txd[1]	MGPIO6/GPOUT[3]
EXT_MEM_A17	in	-	-	-	MGPIO6/GPIN[4]
	out	SRAM_NOR/A17_o	GBIT_GRETH1/txd[2]	-	MGPIO6/GPOUT[4]
EXT_MEM_A18	in	-	-	-	MGPIO6/GPIN[5]
	out	SRAM_NOR/A18_o	GBIT_GRETH1/txd[3]	-	MGPIO6/GPOUT[5]
EXT_MEM_A19	in	-	-	-	MGPIO6/GPIN[6]
	out	SRAM_NOR/A19_o	GBIT_GRETH1/txd[4]	-	MGPIO6/GPOUT[6]
EXT_MEM_A20	in	-	-	-	MGPIO6/GPIN[7]
	out	SRAM_NOR/A20_o	GBIT_GRETH1/txd[5]	-	MGPIO6/GPOUT[7]
EXT_MEM_A21	in	-	-	-	MGPIO7/GPIN[0]
	out	SRAM_NOR/A21_o	GBIT_GRETH1/txd[6]	-	MGPIO7/GPOUT[0]
EXT_MEM_A22	in	-	-	-	MGPIO7/GPIN[1]
	out	SRAM_NOR/A22_o	GBIT_GRETH1/txd[7]	-	MGPIO7/GPOUT[1]
EXT_MEM_A23	in	-	-	-	-
	out	SRAM_NOR/A23_o	GBIT_GRETH1/gtx_clk	-	-
EXT_MEM_A24	in	-	-	-	MGPIO7/GPIN[2]
	out	SRAM_NOR/A24_o	GBIT_GRETH1/mdc	GBIT_GRETH0/mdc	MGPIO7/GPOUT[2]
EXT_MEM_A25	in	-	GBIT_GRETH1/mdio_i	GBIT_GRETH0/mdio_i	MGPIO7/GPIN[3]
	out	SRAM_NOR/A25_o	GBIT_GRETH1/mdio_o	GBIT_GRETH0/mdio_o	MGPIO7/GPOUT[3]
EXT_MEM_CE0	in	-	-	-	MGPIO7/GPIN[4]
	out	SRAM_NOR/CE0	NAND/ren	NAND/ren	MGPIO7/GPOUT[4]
EXT_MEM_CE1	in	-	-	-	MGPIO7/GPIN[5]
	out	SRAM_NOR/CE1	NAND/cle	NAND/cle	MGPIO7/GPOUT[5]
EXT_MEM_CE2	in	-	-	-	MGPIO7/GPIN[6]
	out	SRAM_NOR/CE2	NAND/ce2n	NAND/ce2n	MGPIO7/GPOUT[6]
EXT_MEM_CE3	in	-	-	-	MGPIO7/GPIN[7]
	out	SRAM_NOR/CE3	NAND/ce1n	NAND/ce1n	MGPIO7/GPOUT[7]
EXT_MEM_CE4	in	-	-	-	MGPIO8/GPIN[0]
	out	SRAM_NOR/CE4	NAND/ale	NAND/ale	MGPIO8/GPOUT[0]
EXT_MEM_CE5	in	-	-	-	MGPIO8/GPIN[1]
	out	SRAM_NOR/CE5	NAND/wen	NAND/wen	MGPIO8/GPOUT[1]
EXT_MEM_BHE_H	in	-	-	-	MGPIO8/GPIN[2]
	out	SRAM_NOR/BHE_H	-	-	MGPIO8/GPOUT[2]
EXT_MEM_BHE_L	in	-	-	-	MGPIO8/GPIN[3]
	out	SRAM_NOR/BHE_L	-	-	MGPIO8/GPOUT[3]

Продолжение таблицы 1.797

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме EXT_MEM_MUX_MODE:			Подключение в GPIO режиме
		2'b00	2'b01	2'b10	
EXT_MEM_BLE_H	in	-	-	-	MGPIO8/GPIN[4]
	out	SRAM_NOR/ BLE_H	-	-	MGPIO8/GPOUT[4]
EXT_MEM_BLE_L	in	-	-	-	MGPIO8/GPIN[5]
	out	SRAM_NOR/ BLE_L	-	-	MGPIO8/GPOUT[5]
EXT_MEM_OE	in	-	-	-	MGPIO8/GPIN[6]
	out	SRAM_NOR/OE	-	-	MGPIO8/GPOUT[6]
EXT_MEM_WE	in	-	-	-	MGPIO8/GPIN[7]
	out	SRAM_NOR/WE	NAND/wpn	NAND/wpn	MGPIO8/GPOUT[7]

Наименование вывода микросхемы	I/O	Подключение в функциональном режиме MII_MODE:		Подключение в GPIO режиме
		1'b0	1'b1	
MII_RX_CLK	in	GBIT_GRETH1/rx_clk	GRETH2/rmii_clk	-
	out	-	-	-
MII_RX_COL	in	GBIT_GRETH1/rx_col	-	MGPIO9/GPIN[0]
	out	-	-	MGPIO9/GPOUT[0]
MII_RX_CRS	in	GBIT_GRETH1/rx_crs	GRETH2/rx_crs	MGPIO9/GPIN[1]
	out	-	-	MGPIO9/GPOUT[1]
MII_RXD0	in	GBIT_GRETH1/rxd[0]	GRETH2/rxd[0]	MGPIO9/GPIN[2]
	out	-	-	MGPIO9/GPOUT[2]
MII_RXD1	in	GBIT_GRETH1/rxd[1]	GRETH2/rxd[1]	MGPIO9/GPIN[3]
	out	-	-	MGPIO9/GPOUT[3]
MII_RXD2	in	GBIT_GRETH1/rxd[2]	-	MGPIO9/GPIN[4]
	out	-	-	MGPIO9/GPOUT[4]
MII_RXD3	in	GBIT_GRETH1/rxd[3]	-	MGPIO9/GPIN[5]
	out	-	-	MGPIO9/GPOUT[5]
MII_RX_ER	in	GBIT_GRETH1/rx_er	GRETH2/rx_er	MGPIO9/GPIN[6]
	out	-	-	MGPIO9/GPOUT[6]
MII_RX_DV	in	GBIT_GRETH1/rx_dv	-	MGPIO9/GPIN[7]
	out	-	-	MGPIO9/GPOUT[7]
MII_TX_CLK	in	GBIT_GRETH1/tx_clk	-	-
	out	-	-	-
MII_TX_EN	in	-	-	MGPIO10/GPIN[0]
	out	GBIT_GRETH1/tx_en	GRETH2/tx_en	MGPIO10/GPOUT[0]
MII_TX_ER	in	-	-	MGPIO10/GPIN[1]
	out	GBIT_GRETH1/tx_er	-	MGPIO10/GPOUT[1]
MII_TXD0	in	-	-	MGPIO10/GPIN[2]
	out	GBIT_GRETH1/txd[0]	GRETH2/txd[0]	MGPIO10/GPOUT[2]
MII_TXD1	in	-	-	MGPIO10/GPIN[3]
	out	GBIT_GRETH1/txd[1]	GRETH2/txd[1]	MGPIO10/GPOUT[3]
MII_TXD2	in	-	-	MGPIO10/GPIN[4]
	out	GBIT_GRETH1/txd[2]	-	MGPIO10/GPOUT[4]
MII_TXD3	in	-	-	MGPIO10/GPIN[5]
	out	GBIT_GRETH1/txd[3]	-	MGPIO10/GPOUT[5]
MII_MDC	in	-	-	MGPIO10/GPIN[6]
	out	GBIT_GRETH1/mdc	GRETH2/mdc	MGPIO10/GPOUT[6]
MII_MDIO	in	GBIT_GRETH1/mdio_i	GRETH2/mdio_i	MGPIO10/GPIN[7]
	out	GBIT_GRETH1/mdio_o	GRETH2/mdio_o	MGPIO10/GPOUT[7]

1.3.11.1.5 Мультиплексирования портов EXT_MEM подсистемы LSIF0 и подсистемы HLB_CPU

Порты EXT_MEM могут использоваться контроллерами подсистемы LSIF0 или контроллерами SRAM_NOR_MCIF_1(2). Это реализуется мультиплексором MCIF_MUX_SRAM и определяется значением регистра SCTL: SRAM_NOR_CTRL адрес 0x3800_001C.

Таблица 1.798 – Поля регистра SRAM_NOR_CTRL

Биты	Поле	Знач. по умолч.	Описание
0	sram_nor_mux_ctrl	0x0	Выбор использования портов EXT_MEM: 0 – используются контроллерами подсистемы LSIF0; 1 – используются контроллерами SRAM_NOR_MCIF_1(2) подсистемы HLB_CPU.
3:1	mcif_ce_manage[2:0]	0x0	Управление CE_MANAGE для контроллеров SRAM_NOR MCIF_1(2). Поле аналогично регистру CE_MANAGE см. 1.3.11.1.3.1
31:4	-	-	reserve

1.3.11.2 Описание блоков подсистемы LSIF0

1.3.11.2.1 Контроллер Ethernet 10/100/1000 (GBIT_GRETH)

Гигабитный Ethernet медиа контроллер доступа (GRETH_GBIT) образует интерфейс между шиной AMBA AHB и сетью Ethernet. Он поддерживает 10/100/1000 Мбит скорость в полнодуплексном и полудуплексном режимах. Интерфейс AMBA состоит из интерфейса APB для конфигурации и управления и мастер-интерфейса AHB, который обрабатывает поток данных. Передача данных осуществляется через DMA каналы. Существует один DMA контроллер для передатчика и один для приемника. И одни и те же мастер AHB интерфейс.

Интерфейс Ethernet поддерживает интерфейсы MII и GMII, которые должны быть подключены ко внешнему PHY. GRETH также обеспечивает доступ к интерфейсу MII Management, который используется для настройки PHY. Также присутствует дополнительная аппаратная поддержка для протокола Ethernet отладочной линии связи (EDCL). Это протокол на основе UDP / IP используется для удаленной отладки.

Поддерживаются функции Точечная Сборка (Scatter Gather) I / O и TCP / UDP для каналов DMA через контрольную сумму IPv4, разгружая приемник и передатчик.

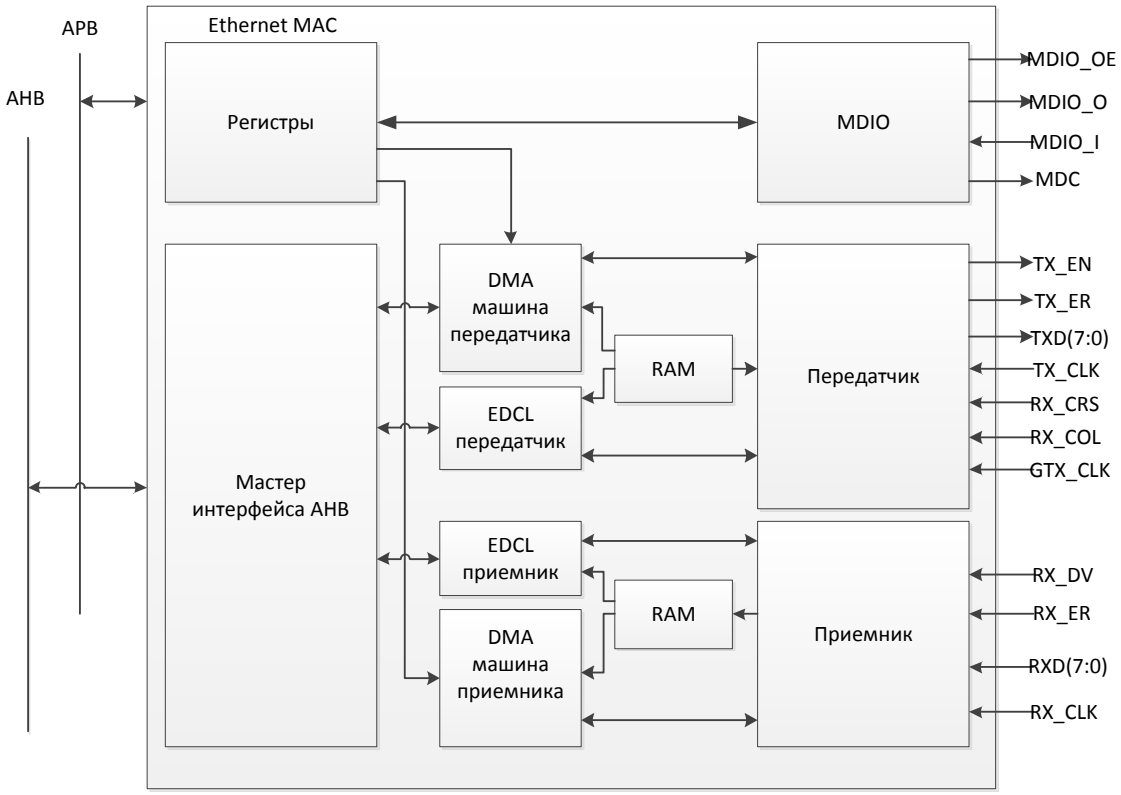


Рисунок 1.150 - Блок-схема внутренней структуры GRETH_GBIT

1.3.11.2.1.1 Управление

1.3.11.2.1.1.1 Обзор системы

GRETH состоит из 3 функциональных блоков: каналы DMA, MDIO интерфейс и дополнительная Ethernet отладочная линия связи (EDCL).

Основной функционал заложен в DMA каналы, которые используются для передачи данных между шиной AHB и сетью Ethernet. Существует один передатчик DMA канала и один канал приемника DMA. Функционирование DMA каналов управляется посредством регистров, доступных через интерфейс APB.

Интерфейс MDIO используется для доступа к конфигурационному и статусному регистрам в одном или нескольких подключенных PHY к MAC. Работа этого интерфейса также контролируется через интерфейс APB.

Дополнительный EDCL обеспечивает доступ на чтение и запись в шину AHB через Ethernet. Он использует UDP, IP и ARP протоколы вместе с пользовательским протоколом прикладного уровня для достижения этой цели. EDCL содержит недоступные пользователю регистры и всегда работает параллельно DMA каналам.

MII и GMII используются для связи с PHY. Остальная информация может быть найдена в разделе 1.3.11.2.1.6.2

EDCL и DMA каналы вместе используют Ethernet приемник и передатчик. Остальная информация может быть найдена в разделах 1.3.11.2.1.3. – 1.3.11.2.1.6

1.3.11.2.1.1.2 Поддержка протокола

GRETH_GBIT реализуется согласно стандарту IEEE 802.3-2002. Не поддерживается опциональный контроль подуровня. Это означает что пакеты типа 0x8808 (единственные в настоящий момент определенные контрольные пакеты) не будут обрабатываться.

1.3.11.2.1.2 Tx DMA интерфейс

Передающий DMA интерфейс используется для передачи данных в Ethernet сеть. Передача осуществляется с помощью дескрипторов, расположенных в памяти.

1.3.11.2.1.3 Настройки дескриптора

Единичный дескриптор показан в таблицах Таблица 1.799 и Таблица 1.800. Число посылаемых байт должно быть установлено в поле длины, и адресное поле должно указывать на данные. На выравнивание поля адреса ограничений не накладывается. Если установлен бит на разрешение прерывания (IE), то прерывание будет генерироваться, если пакет был отправлен (также требуется, чтобы бит прерывания передатчика в регистре управления также был установлен). Прерывание будет сгенерировано независимо от того, будет пакет передан успешно или нет.

Таблица 1.799 - Слово 0 дескриптора передатчика GRETH_GBIT (адрес смещения 0x0).

31 21	20	19	18	17	16	15	14	13 12	11	10 0
РЕЗЕРВ	UC	TC	IC	MO	LC	AL	UE	IE	WR	EN	ДЛИНА

31:21 РЕЗЕРВ

20 UDP контрольная сумма (UC) – Считает и вставляет UDP контрольную сумму для этого пакета. Контрольная сумма вставляется только в случае, если UDP пакет обнаружен.

19 TCP контрольная сумма (TC) – Считает и вставляет TCP контрольную сумму для этого пакета. Контрольная сумма вставляется только в случае, если TCP пакет обнаружен.

18 IP контрольная сумма (IC) – Считает и вставляет IP контрольную сумму заголовка для этого пакета. Контрольная сумма вставляется только в случае, если TCP пакет обнаружен.

17 Больше (MO) – Для формирования пакета должно быть получено больше дескрипторов (Точечная сборка I/O).

16 Поздняя коллизия (LC) - Коллизия произошла в конце во время передачи.

15 Ограничение лимита ошибок (AL) – Пакет не был передан из-за того, что было достигнуто максимальное количество попыток на передачу.

14 Ошибка опустошения (UE) – Пакет был передан неправильно из-за ошибки опустошения FIFO.

13 Разрешение прерывания (IE) – Разрешение прерываний. Прерывание будет сгенерировано, когда из дескриптора будет отправлен пакет, при условии, что установлен бит, разрешающий прерывание на передачу в регистре управления. Прерывание генерируется независимо от того, пакет был передан успешно, или передача была прекращена с ошибкой.

12 Сворачивать(WR) – Установка 1 заставляет указатель дескриптора возвращаться в 0 после того, как дескриптор был использован. Если этот бит не установлен, указатель будет увеличен на 8. Указатель автоматически устанавливается в 0, когда достигается граница таблицы дескрипторов в 1 Кбайт.

11 Включение(EN) – Установка в 1 включает дескриптор. Должно всегда устанавливаться последним из всех полей дескриптора.

10:0 Длина – Количества байт для передачи.

Таблица 1.800 - Слово 1 дескриптора передатчика GRETH_GBIT (адрес смещения 0x4).

31	0
АДРЕС	

31:0 Адрес(ADDRESS) – Указатель на область буфера, из которого должны быть загружены данные.

Чтобы включить дескриптор необходимо установить бит включения (EN) и после того, как это сделано, дескриптор не следует трогать, пока бит включения не будет очищен GRETH_GBIT. Остальные поля в области дескрипторов объяснены позже в этом разделе.

1.3.11.2.1.3.1 Начало передачи

Включение дескриптора не достаточно для начала передачи. Указатель на область памяти, содержащую дескриптор, должен быть установлен первым в GRETH_GBIT. Это осуществляется в регистре указателя дескриптора передачи. Адрес должен быть выровнен по границе 1 Кбайт. Биты с 31 по 10 содержат базовый адрес области дескриптора, в то время как биты с 9 по 3 формируют указатель на определенный дескриптор. Первый дескриптор должен находиться по базовому адресу, и когда он будет использован GRETH_GBIT , указатель области будет увеличен на 8, чтобы указывать на следующий дескриптор. Указатель будет автоматически возвращаться в 0, когда следующая граница в 1 Кбайт будет достигнута (используется адрес дескриптора со смещением 0x3F8). WR бит в дескрипторе может быть установлен, чтобы возвращать назад в 0 указатель, не достигший границы в 1 Кбайт.

Поле указателя, к тому же, сделано доступным для записи для максимальной гибкости, но нужно быть аккуратным при записи в регистр указателя дескриптора. Никогда не следует трогать регистр, когда передача запущена.

Последний шаг для запуска передачи установить бит включения передачи в регистре управления. Это сообщает GRETH_GBIT, что есть еще активные дескрипторы в таблице дескрипторов. Этот бит всегда должен быть установлен, когда новый дескриптор доступен, даже если передача уже запущена. Дескрипторы должны всегда быть доступны до того, как бит, включающий передачу, установлен.

1.3.11.2.1.3.2 Обработка дескриптора после передачи

Когда передача пакета окончена, статус записан в первое слово соответствующего дескриптора. Бит Ошибки Опустошения устанавливается, если RAM передатчик был не в состоянии предоставить данные с достаточной скоростью. Это означает, что, скорее всего, проблема в синхронизации, вызванная низкой тактовой частоте на AHB клоке. Весь пакет хранится в RAM передатчика перед передачей так, чтобы недогрузки не могли бы вызвать перегрузку шины. Бит Лимита Ограничения Ошибок устанавливается, если коллизий произошло

больше, чем разрешено. При работе в режиме 1000 Мбит бит Поздней Коллизии указывает, что коллизия произошла после того, как граница временного интервала была пройдена.

Пакет был успешно передан только, если эти три бита равны нулю. Другие биты в первом слове дескриптора устанавливаются в ноль после передачи, а второе слово остается нетронутым.

Бит включения должен быть использован в качестве индикатора того, когда дескриптор может быть использован повторно после того, как он был очищен с помощью GRETH_GBIT. Есть три бита в статусном регистре GRETH_GBIT, которые отображают статус передачи. Бит ошибки передачи (TE) устанавливается каждый раз, когда передача завершилась с ошибкой (если был установлен по крайней мере один из трех статусных битов в дескрипторе передачи). Передача Успешна (TI) устанавливается каждый раз, когда передача закончилась успешно.

Бит ошибки AHB Передачи (TA) устанавливается, если была обнаружена ошибка AHB, либо при чтении дескриптора, чтение пакета данных или при записи в статусный дескриптор. Любые активные передачи прерываются и передатчик отключается. Передатчик может быть активирован снова настройкой регистра передатчика.

1.3.11.2.1.3.3 Настройка данных для передачи

Данные, подлежащие передаче, должны быть помещены, начиная с адреса, на который указывает поле дескриптора адреса. GRETH_GBIT не добавляет адреса Ethernet и поля типа, так что они должны присутствовать в буфере данных. 4 байта Ethernet CRC автоматически добавляется в конце каждого пакета. Каждый дескриптор будет отправлен как единичный пакет Ethernet. Если размер поля в дескрипторе больше 1514 байт, пакет не будет отправлен.

1.3.11.2.1.3.4 Точечная сборка I/O

Пакет может быть сформирован из данных, извлекаемых из нескольких дескрипторов. Это называется Точечная Сборка I /O. Бит Больше (MO) должен быть установлен в 1, чтобы указать, что больше дескрипторов должен быть использовано для создания текущего пакета. Когда данные из текущего дескриптора были прочитаны в память, выбирается следующий дескриптор и новые данные добавляются к предыдущим данным. Это продолжается до тех пор, пока не встречается дескриптор, с установленным 0 в бите MO. Пакет будет передан.

Статус мгновенно записывается после считывания данных в оперативную память для дескрипторов с включенной MO. Биты состояния всегда установлены в 0 до тех пор, пока не происходит передач. Биты состояния будут записаны в последний дескриптор пакета (в котором MO установлен в 0)после завершения передачи.

Прерывания не генерируются для дескрипторов с включенной MO, так что на бит IE не обращаем внимания в этом случае.

Биты контрольной суммы разгрузки управления (объясняется в разделе 1.3.11.2.1.3.5) должны быть установлены на одни и те же значения для всех дескрипторов, используемых для одного пакета.

1.3.11.2.1.3.5 Контрольная сумма разгрузки

Поддержка предоставляется для правильных расчетов контрольной суммы в оборудовании для TCP и UDP через IPv4 для нефрагментированных пакетов с длиной данных меньше или равной максимальному передаваемому размеру. Расчеты контрольной суммы включены в каждом дескрипторе и применяется только к этому пакету (если установлен бит MO, все дескрипторы, используемые для одного пакета, должны иметь такие же установки битов управления контрольной суммы).

Бит контрольной суммы IP (IC) разрешает расчет контрольной суммы IP заголовка. Если IPv4 пакет обнаружен при передаче пакета, связанного с дескриптором, вычисляется и вставляется контрольная сумма заголовка. Если устанавливается TCP контрольная сумма (TC), контрольная сумма TCP рассчитывается и вставляется, если обнаруживается / пакет IPv4 TCP. Наконец, если устанавливается бит контрольной суммы UDP, контрольная сумма UDP вычисляется и вставляется, если обнаружен пакет UDP / IPv4. В случае фрагментированных IP

пакетов, неправильная контрольная сумма для TCP и UDP вставляется в первый фрагмент (который содержит TCP или UDP заголовка).

1.3.11.2.1.4 Rx DMA интерфейс

Принимающий DMA интерфейс применяется для приема данных через Ethernet сеть. Прием осуществляется с помощью дескрипторов, расположенных в памяти.

1.3.11.2.1.4.1 Настройка дескрипторов

Единичный дескриптор показан в таблице Таблица 1.801 и Таблица 1.802. Поле адреса указывает на место, где полученные данные будут храниться. Там нет ограничения на выравнивание. GRETH_GBIT никогда не хранит более 1518 байт в буфере (что является максимальным размером кадра без учета CRC). Поле CRC (4 байта) никогда не сохраняются в памяти, поэтому ине входит в это количество. Если установлен бит разрешения прерывания (IE), то прерывание будет генерироваться после того, как пакет был принят в этот буфер (для этого требуется, чтобы бит прерывания по приему был также установлен в регистре управления). Прерывание будет генерироваться независимо от того, был получен пакет успешно или нет.

Установка бита включения сообщает, что дескриптор действует, то есть он может быть использован для хранения пакета. После того, как установка произошла, дескриптор не следует трогать, пока бит EN не будет очищен самим GRETH_GBIT.

Остальные поля дескриптора объясняются далее в этом разделе.

Таблица 1.801 - Слово 0 дескриптора приемника GRETH_GBIT (адрес смещения 0x0)

31 27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11 10	0
РЕЗЕРВ	MC	IF	TR	TD	UR	UD	IR	ID	LE	OE	CE	FT	AE	IE	WR	EN	ДЛИНА

31:27 РЕЗЕРВ

26 Широковещательная рассылка (MC) - Адресом назначения пакета будет группа адресов (не транслируется).

25 IP-фрагмент (IF) – Обнаружен фрагментированный IP-пакет.

24 TCP ошибка (TR) – Обнаружена ошибка контрольной суммы TCP.

23 TCP обнаружен (TD) – Обнаружен пакет TCP пакет.

22 UDP ошибка (UR) – Обнаружена ошибка контрольной суммы UDP.

21 UDP обнаружен (UD) – Обнаружен UDP пакет.

20 IP ошибка (IR) – Обнаружена ошибка контрольной суммы IP.

19 IP обнаружен (ID) – Обнаружен IP пакет.

18 Ошибка длины (LE) – Поле длины / типа пакета не совпадает с фактическим числом полученных байт.

17 Ошибка переполнения (OE) – Из-за переполнения получен некорректный кадр.

16 Ошибка CRC (CE) – В данном кадре обнаружена CRC ошибка.

15 Слишком длинный кадр(FT) – Был получен кадр больше максимального размера. Часть, превосходящая максимальный размер, была усечена.

14 Ошибка выравнивания (AE) - Было получено нечетное число полубайт.

13 Разрешение прерывания (IE) - Включение прерываний. Прерывание будет генерироваться после того, как пакет будет принят в дескриптор при условии, что установлен бит, разрешающий прерывание по приему, в контрольном регистре. Прерывание генерируется независимо от того, пакет был принят успешно, или он завершен с ошибкой.

12 Сворачивать (WR) – Установка 1 заставляет указатель дескриптора указывать в 0 после того, как дескриптор был использован. Если этот бит не установлен, указатель будет увеличен на 8. Указатель автоматически устанавливается в 0, когда граница таблицы дескрипторов в 1 Кбайт достигнута.

11 Включение(EN) – Установка в 1 включает дескриптор. Должно всегда устанавливаться последним из всех полей дескриптора.

10:0 Длина – Количества байтов для передачи.

Таблица 1.802 - Слово 1 дескриптора приемника GRETH_GBIT (адрес смещения 0x4)

31	0
АДРЕС	

31:0 Адрес(ADDRESS) – Указатель на область буфера, куда должны быть загружены данные.

1.3.11.2.1.4.2 Начало приема

Включение дескриптора не достаточно, чтобы начать прием. Указатель на область памяти, содержащую дескрипторы, должен быть первым установлен в GRETH_GBIT. Это осуществляется в регистре указателя дескриптора приемника. Адрес должен быть выровнен по границе 1 Кбайт. Биты с 31 по 10 содержат базовый адрес области дескриптора, в то время как биты с 9 по 3 формируют указатель на определенный дескриптор. Первый дескриптор должен находиться по базовому адресу, и когда он будет использован GRETH_GBIT , указатель области будет увеличен на 8, чтобы указывать на следующий дескриптор. Указатель будет автоматически возвращаться в 0, когда следующая граница в 1 Кбайт будет достигнута (используется адрес дескриптора со смещением 0x3F8). WR бит в дескрипторе может быть установлен, чтобы возвращать назад в 0 указатель, не достигший границы в 1 Кбайт.

Поле указателя к тому же сделано доступным для записи для максимальной гибкости, но нужно быть аккуратным при записи в регистр указателя дескриптора. Никогда не следует трогать регистр, когда прием запущен.

Последний шаг для запуска передачи установить бит включения приема в контрольном регистре. Это заставит GRETH_GBIT прочитать первый дескриптор и ждать входящий пакет.

1.3.11.2.1.4.3 Обработка дескриптора после приема

Когда передача пакета окончена, статус записан в первое слово соответствующего дескриптора. Бит Ошибки Опустошения устанавливается, если RAM передатчик был не в состоянии предоставить данные с достаточной скоростью. Это означает, что, скорее всего, проблема в синхронизации, вызванная низкой тактовой частоте на AHB клоке. Весь пакет

хранится в RAM передатчика перед передачей так, чтобы недогрузки не смогли бы вызвать перегрузку шины. Бит Лимита Ограничения Ошибок устанавливается, если коллизий произошло больше, чем разрешено. При работе в режиме 1000 Мбит бит Поздней Коллизии указывает, что коллизия произошла после того, как граница временного интервала была пройдена.

GRETH сообщает о завершении приема по очистке бита включения дескриптора. Остальные биты управления (WR, IE) также обнуляются. Количество полученных байт показано в поле длины. Части Ethernet кадра содержат поле адреса назначения, адреса источника, поле типа и данных. Статусные биты 24-14 в первом слове дескриптора сообщают о различных ошибках приема. При приеме без ошибок канального уровня биты 18 - 14 равны нулю. Биты состояния описаны в таблице Таблица 1.801 (за исключением битов контрольной суммы разгрузки, которые также описаны в разделе 1.3.11.2.1.3.5).

Поступающие пакеты меньше, чем минимальный размер Ethernet кадра в 64 байта, не принимаются и отбрасываются. Текущий дескриптор приема останется нетронутым для первого поступившего пакета с корректным размером. Бит TS в регистре состояния устанавливается в 1 каждый раз, когда это случается.

Если пакет получен с адреса, не разрешенного MAC, бит IA в статусном регистре будет установлен в 1. Пакеты, размер которых превышает максимальный размер, влекут за собой установку в 1 бита FT в дескрипторе приема. Поле длины не гарантированно содержит правильное значение принятых байт. Подсчет прекращается после того, как в память был записан последний байт разрешенный лимитом максимального размера.

GRETH никогда не трогает слово адреса дескриптора.

1.3.11.2.1.4.4 Прием с AHB ошибками

Если AHB ошибка случилась во время того, как дескриптор читает или сохраняет данные, бит AHB ошибки при приеме в статусном регистре будет установлен в 1, и приемник отключится. Текущий прием прекратится. Приемник может быть включен снова установкой бита Разрешения Приема в регистре управления.

1.3.11.2.1.4.5 Принимаемые MAC-адреса

В стандартной конфигурации ядро получает пакеты либо с одиночным адресом, установленным в адресном регистре MAC, или с широковещательным адресом. Поддержка многоадресной передачи также может быть включена, и в этом случае хэш-функция используется для фильтрации полученных широковещательных пакетов. 64-разрядный регистр, который доступен через интерфейс APB, определяет, какие адреса должны быть получены. Каждый адрес сопоставляется с одним из 64 битами, используемыми хэш-функцией и если этот бит установлен в 1, то пакет будет принят. Адрес сопоставляется с таблицей, путем взятия 6 младших битов 32 разрядного Ethernet CRC, вычисленного с использованием MAC адреса назначения кадра. Бит в дескрипторе приема устанавливается, если был получен пакет с широковещательным адресом.

1.3.11.2.1.4.6 Контрольная сумма разгрузки

Поддержка предоставляется для правильных расчетов контрольной суммы в оборудовании для TCP и UDP через IPv4. Логика расчета контрольных сумм всегда активна и фиксирует IPv4 пакеты с TCP или UDP информационной нагрузки. Если IPv4 обнаруживает установленный ID бит, UD устанавливается, если информационная нагрузка UDP обнаружено в пакете IP и TD устанавливается, если информационная нагрузка TCP определена в пакете IP (TD и UD никогда не устанавливается, если пакет IPv4 не обнаружен). Когда один или несколько из этих типов пакетов обнаружены, соответствующая контрольная сумма вычисляется, и если обнаружена ошибка, устанавливается бит ошибки контрольной суммы для этого типа пакета. Биты ошибок никогда не устанавливается, если соответствующий тип пакета не обнаружен. Ядро не

поддерживает расчеты контрольных сумм для TCP и UDP, когда пакет IP фрагментирован. Это условие сигнализируется битом IF в дескрипторе приема, который действителен, когда нет ошибок контрольных сумм TCP и UDP.

1.3.11.2.1.5 MDIO интерфейс

Интерфейс MDIO обеспечивает доступ к конфигурационным и статусным регистрам PHY через двухпроводной интерфейс, который входит в интерфейс MII. GRETH_GBIT обеспечивает полную поддержку интерфейса MDIO.

Интерфейс MDIO может быть использован для доступа к от 1 до 32 PHY, содержащих от 1 до 32 16-битных регистров. Процесс чтения осуществляется путем записи PHY и регистра адресов в регистр управления MDIO и установкой бита чтения. Это влечет за собой установку в 1 бита Занят, и операция считается законченной, когда бит Занят обнуляется. Если операция прошла успешно, бит Сбой связи равен нулю, а поле данных содержит считанные данные. Неудачная операция сигнализируется установкой бита Сбой связи в 1. Поле данных не определено в этом случае.

Операция записи начинается с записи 16-битных данных, адреса PHY и адреса регистра в регистр управления MDIO, и установкой бита записи в 1. Операция закончена, когда бит Занят сбрасывается, и считается успешной, если бит Сбой связи равен нулю.

1.3.11.2.1.5.1 Прерывания PHY

Ядро также поддерживает изменение статуса прерывания от PHY. Бит изменения состояния PHY в статусном регистре устанавливается в 1 каждый раз, когда в этом сигнале обнаруживается событие. Если бит, разрешающий статус прерывания PHY, установлен в 1, в этом случае ядро также будет генерировать прерывание на шину AHB.

1.3.11.2.1.6 Ethernet Отладка линии связи (EDCL)

EDCL обеспечивает доступ к шине AHB на кристалле через Ethernet. Он использует UDP, IP и ARP протоколы вместе с пользовательским протоколом прикладного уровня. Протокол прикладного уровня использует алгоритм ARQ, чтобы обеспечить надежность перевода инструкций AHB. Через это соединение может быть создана передача на чтение или запись на любой адрес на шине AHB.

1.3.11.2.1.6.1 Эксплуатация

EDCL принимает пакеты параллельно с MAC каналом DMA приемника. Он использует отдельный MAC-адрес, который используется для распознавания EDCL пакета из пакетов, предназначенных для MAC-канала DMA. EDCL также имеет IP-адрес, который устанавливается в настройках rtl уровня. Так как ARP пакеты используют широковещательный Ethernet-адрес, IP-адрес необходимо использовать в данном случае для отличия между EDCL ARP пакетами и теми пакетами, которые должны идти в DMA-канал. Пакеты, определенные как EDCL пакетами, не обрабатываются каналом DMA приемника.

Когда пакеты проверяются на корректность осуществляется работа AHB. Операция выполняется тем же мастером интерфейса AHB, что и используют DMA машины. Ответы будут автоматически отправлены передатчиком EDCL после завершения операции. EDCL делит передатчик Ethernet с передатчиком DMA машин, но имеет более высокий приоритет.

1.3.11.2.1.6.2 EDCL протоколы

EDCL принимает Ethernet кадры, содержащие данные IP или ARP. ARP обрабатывается в соответствии со спецификацией протокола без каких-либо исключений.

IP пакеты несут текущие AHB команды. EDCL ожидает кадр Ethernet, содержащий IP, UDP и EDCL специфические части прикладного уровня. Таблица 549 показывает пакет IP, требуемый для EDCL. Содержание различных заголовков протокола можно найти в TCP / IP литературе.

Таблица 1.803 - IP пакет, необходимый для работы с EDCL.

Ethernet заголовок	IP заголовок	UDP заголовок	2 байта смещение	4 байта Контрольное слово	4 байта Адрес	Данные 0 – 242 Слова по 4 байта	Ethernet CRC

Следующие требования для успешной коммутации с EDCL: правильный MAC-адрес назначения, как установлен в настройках rtl уровня, поле типа Ethernet, содержащее 0x0806 (ARP) или 0x0800 (IP). Затем IP адрес сравнивается на соответствие со значением, заданным в настройках rtl уровня. Контрольная сумма и идентификационные поля IP-заголовка не проверяются. Есть несколько ограничений для полей IP-заголовка. Версия должна быть равна четырем и размер заголовка должен быть 5 байт (без дополнительных опций). Поле протокола должен быть всегда 0x11, что свидетельствует о UDP пакете. Длина и контрольная сумма являются единственными IP полями изменяемыми в ответе.

EDCL обеспечивает только одну операцию за раз, и поэтому не требуется проверять номер UDP порта. Ответ будет иметь первоначальный номер порта источника в обоих полях источника и назначения. Контрольная сумма UDP не используются, и поле контрольной суммы в ответах устанавливается в ноль.

Таблица 1.804 - Поля прикладного уровня EDCL в принимаемых кадрах

16 бит Смещение	14 бит Порядковый номер	1 бит R / W	10 бит Длина	7 бит Неиспользуемые

Поле данных UDP содержит поля EDCL протокола приложения. Таблица Таблица 1.804 показывает поля протокола прикладного уровня (поле данных исключено) в пакетах, полученных через EDCL. 16-битное смещение используется для выравнивания остальных данных прикладного уровня в границах слов в памяти и, таким образом, может быть установлено в любое значение. Поле R / W определяет, следует ли выполнять чтение (0) или запись (1). Поле длины содержит количество байт для чтения или записи. Если поле R / W равно единице, то поле данных, показанное в таблицеТаблица 1.803, содержит данные, которые будут записаны. Если поле R / W равно нулю, то поле данных пустое в принятых пакетах. ТаблицаТаблица 1.805 показывает поля прикладного уровня ответов EDCL. Поле длины ответов всегда равно нулю в случае, если ответы на запросы на запись. Для запросов на чтение, поле содержит количество байт, содержащихся в поле данных.

Таблица 1.805 - Поля прикладного уровня EDCL в принимаемых кадрах

16 бит Смещение	14 бит Порядковый номер	1 бит R / W	10 бит Длина	7 бит Неиспользуемые

EDCL реализует алгоритм Go-Back-N, обеспечивающий надежные переводы. 14-битный порядковый номер принятых пакетов сверяется с внутренним счетчиком на соответствие. Если они не совпадают, операция не выполняется, и поле ACK / NAK устанавливается в 1 в ответном кадре. Ответный кадр содержит внутреннее значение счетчика в поле порядкового номера. Если порядковый номер совпадает, то операция выполняется, внутренний счетчик увеличивается, значение внутреннего счетчика сохраняется в поле порядкового номера, и поле ACK / NAK

устанавливается в 0 в ответном сообщении. Поле длины всегда равно 0 для кадров, где ACK / NAK равно 1. Поле неиспользуемых 7 бит не проверяются и копируются в ответ. Таким образом, их можно установить, если необходимо ввести, например, некоторые дополнительные идентификационные биты.

1.3.11.2.1.7 Регистры

Ядро, программируемое через регистры, соответствует в адресном пространстве APB.

Таблица 1.806 - Регистры GRETH_GBIT

Смещение адреса на шине APB	Регистр
0x0	Регистр управления
0x4	Статусный регистр
0x8	MAC адрес MSB
0xC	MAC адрес LSB
0x10	MDIO статусный / управления
0x14	Указатель дескриптора передатчика
0x18	Указатель дескриптора приемника
0x1C	EDCL IP
0x20	Таблица хешей msb
0x24	Таблица хешей lsb
0x28	EDCL MAC адрес MSB
0x2C	EDCL MAC адрес LSB
0x10000 – 0x107FC	Доступ к отладке буфера памяти передатчика
0x20000 – 0x207FC	Доступ к отладке буфера памяти приемника
0x30000 – 0x3FFFC	Доступ к буферу отладки EDCL

Таблица 1.807 - Регистр контроля GRETH

31	30	28	27	26	25	24	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ED	BS		GA	MA	MC	РЕЗЕРВ		ED	RD	DD	ME	PI	BM	GB	SP	RS	PR	FD	RI	TI	RE	TE	

31 EDCL доступен (ED) – Устанавливается в 1, чтобы сделать EDCL доступным.

30:28 Размер буффера EDCL (BS) – Показывает количество памяти, используемое для буффера EDCL. 0 = 1 Кбайт, 1 = 2 Кбайт,….., 6 = 64 Кбайт.

27 Гигабитный MAC доступен (GA) – Этот бит всегда читается как 1, и сообщает о том, что у

MAC есть возможность работы на скорости 1000 Мбит.

26 Включение MDIO прерываний (ME) – Установлено в 1, когда ядро поддерживает MDIO прерывания. Только для чтения.

25 Широковещательная рассылка доступна (MC) – Установлен в 1, когда ядро поддерживает многоадресный прием. Только для чтения.

24:15 РЕЗЕРВ

14 Выключение EDCL (ED) – Установить в 1, чтобы отключить EDCL и 0, чтобы включить. Значение при сбросе берется из настроек rtl уровня. Доступно только, если аппаратное обеспечение EDCL присутствует в ядре.

13 Включение отладки RAM (RD) – Установить в 1 для включения режима отладки RAM. Значение при сбросе: '0'. Доступно только если в настройках rtl уровня ramdebug VHDL отличен от нуля.

12 Отключение дуплексного обнаружения (DD) – Отключение EDCL скоростного/дуплексного FSM обнаружения. Если FSM не может завершить обнаружение, тогда интерфейс MDIO будет заблокирован в режиме Занято. Если программное обеспечение должно получить доступ к MDIO, FSM может быть отключен здесь, и как только бит Занято установится в 0 MDIO интерфейс будет доступен. Следует отметить, что FSM не может быть заново переключен в рабочее состояние.

11 Широковещательная рассылка включена (ME) - Включить прием групповых адресов. Значение при сбросе: '0'.

10 Разрешение изменения статуса прерывания PHY (PI) - Разрешает прерывания для отслеживания изменения состояния PHY.

9 Пакетный режим (BM) - Когда установлен в 1, передача производится с использованием пакетного режима при 1000 Мбит в полудуплексном режиме (GB = 1, FD = 0). При 0 в скоростном режиме при обычной передаче всегда используются с расширением вставлен. Операция не определена, когда бит установлен в 1 в других скоростных режимах. Значение при сбросе: '0'.

8 Гигабитный (GB) – При 1 устанавливает текущий скоростной режим в 1000 Мбит, а при 0, режим скорости выбирается согласно 7му биту (SP). Значение при сбросе: '0'.

7 Скорость (SP) - Устанавливает текущий режим скорости. 0 = 10 Мбит, 1 = 100 Мбит. Не должен быть установлен в 1 одновременно с 8м битом (GB). Значение при сбросе: '0'.

6 Сброс (RS) – Записанная единица в этот бит производит сброс ядра GRETH_GBIT. Сам обнуляет бит сброса. Никакие другие доступы не должно осуществляться к подчиненному интерфейсу, кроме опроса этот бита, пока он не очищен.

5 Беспорядочный режим (PM) – Если бит установлен, GRETH_GBIT работает в беспорядочном режиме, который означает, что он будет получать все пакеты, независимо от адреса назначения. Значение при сбросе: '0'.

4 Полный дуплекс (FD) - Если бит установлен, GRETH_GBIT работает в режиме полного дуплекса, в противном случае работает в полудуплексном режиме. Значение при сбросе: '0'.

3 Прерывание по приему (RI) - Включает прерывание по приему. Прерывание будет генерироваться каждый раз при приеме пакета, если этот бит установлен. Прерывание генерируется независимо от того, был пакет принят успешно или с ошибкой. Значение при сбросе: '0'.

2 Прерывание по передаче (TI) – Включает прерывание по передаче. Прерывание будет генерироваться каждый при передаче пакета, если этот бит установлен. Прерывание

генерируется независимо от того, был пакет передан успешно или с ошибкой. Значение при сбросе: '0'.

1 Включение приемника (RE) - Должен быть установлен в 1 каждый раз, когда новые дескрипторы включены. Пока этот бит установлен в 1, GRETH_GBIT будет читать новые дескрипторы, и как только он сталкивается с выключенным дескриптором, приемник останавливается до тех пор, пока RE снова не будет установлен в 1. Этот бит должен быть установлен в 1 после того, как новые дескрипторы были включены. Значение при сбросе сброса: '0'.

0 Включение передатчика (TE) - Должен быть установлен в 1 каждый раз, когда новые дескрипторы включены. Пока этот бит установлен в 1, GRETH_GBIT будет читать новые дескрипторы, и как только он сталкивается с выключенным дескриптором, передатчик останавливается до тех пор, пока TE снова не будет установлен в 1. Этот бит должен быть установлен в 1 после того, как новые дескрипторы были включены. Значение при сбросе сброса: '0'.

Таблица 1.808 - Статусный регистр GRETH_GBIT

31	9	8	7	6	5	4	3	2	1	0
РЕЗЕРВ	PS		IA	TS	TA	RA	TI	RI	TE	RE

31:9 РЕЗЕРВ

8 Изменения статуса PHY (PS) – Устанавливается каждый раз, когда фиксируется изменение состояния PHY.

7 Неверный адрес (IA) – Получен пакет, с недопустимым MAC адресом. Сбрасывается при записи 1. Значение при сбросе: '0'.

6 Слишком маленький (TC) – Размер полученного пакета меньше, чем возможный минимальный размер пакета. Сбрасывается при записи 1. Значение при сбросе: '0'.

5 Ошибка передатчика AHB (TA) - Ошибка AHB была обнаружена в передатчике DMA машины. Сбрасывается при записи 1. Не сбрасывается при аппаратном сбросе.

4 Ошибка приемника AHB (RA) - Ошибка AHB была обнаружена в приемнике DMA машины. Сбрасывается при записи 1. Не сбрасывается при аппаратном сбросе.

3 Успешная передача (TI) - Пакет был передан без ошибок. Сбрасывается при записи 1. Не сбрасывается при аппаратном сбросе.

2 Успешный прием (RI) - Пакет был принят без ошибок. Сбрасывается при записи 1. Не сбрасывается при аппаратном сбросе.

1 Ошибка передатчика (TE) – Передача пакета была завершена с ошибкой. Сбрасывается при записи 1. Не сбрасывается при аппаратном сбросе.

0 Ошибка приема (RE) – Прием пакета был завершен с ошибкой. Сбрасывается при записи 1. Не сбрасывается при аппаратном сбросе.

Таблица 1.809 - GRETH_GBIT MAC адрес MSB

31 16	15 0
РЕЗЕРВ	Биты с 47 по 32 MAC адреса

31: 16 РЕЗЕРВ

15: 0 Два наиболее значимых байта MAC адреса. Не сбрасывается при аппаратном сбросе.

Таблица 1.810 - GRETH_GBIT MAC адрес LSB

31 0
Биты с 31 по 0 MAC адреса

31: 0 4 младших байта MAC адреса. Не сбрасывается при аппаратном сбросе.

Таблица 1.811 - GRETH_GBIT MDIO управления / статусный регистр

31 16	15 11	10 6	5	4	3	2	1	0
ДАННЫЕ	PHY АДРЕС	АДРЕС РЕГИСТРА		NV	BU	LF	RD	WR

31: 16 Данные(DATA) – Содержит данные, считанные с помощью операции чтения, и при передаче данные берутся из этой области. Значение при сбросе: 0x0000.

15: 11 PHY адрес (PHYADDR) – Это поле содержит адрес PHY, который должен быть доступен во время записи или чтения. Значение сброса: ‘00000’.

10: 6 Регистр адреса (REGADDR) - это поле содержит адрес регистра, который должен быть доступ во время записи или чтения. Значение при сбросе: ‘00000’.

5 РЕЗЕРВ

4 Недействительна (NV) - Когда операция завершена (ЗАНЯТО = 0), этот бит указывает, были ли полученные данные, хранящиеся в поле данных, достоверными данными. Значение при сбросе: '0'.

3 Занят (BU) - Когда операция выполняется, этот бит устанавливается в единицу. Как только операция завершается, и управление находится в режиме ожидания, этот бит очищается. Значение при сбросе: '0'.

2 Сбой Связи (LF) - После завершения операции (ЗАНЯТО = 0) этот бит устанавливается в 1, если не было обнаружено функциональное управление. Значение при сбросе: '1'.

1 Чтение (RD) - Начать операцию чтения на интерфейсе управления. Данные сохраняются в поле данных. Значение при сбросе: '0'.

0 Запись (WR) - Начать операцию записи на интерфейсе управления. Данные берутся из поля данных. Значение при сбросе: '0'.

Таблица 1.812 - Таблица базового адреса дескриптора передатчика GRETH_GBIT

31 — 10	9 — 3	2 — 0
БАЗОВЫЙ АДРЕС	УКАЗАТЕЛЬ	РЕЗЕРВ

31: 10 Таблица базового адреса дескриптора передатчика (BASEADDR) - Базовый адрес в таблицу дескрипторов передатчика. Не сбрасывается при аппаратном сбросе.

9: 3 Указатель дескриптора (DESCPNT) - Указатель на отдельные дескрипторы. Автоматически увеличивается Ethernet MAC контроллером.

2: 0 РЕЗЕРВ

Таблица 1.813 - Таблица базового адреса дескриптора приемника GRETH_GBIT.

31 — 10	9 — 3	2 — 0
БАЗОВЫЙ АДРЕС	УКАЗАТЕЛЬ	РЕЗЕРВ

31: 10 Таблица базового адреса дескриптора приемника (BASEADDR) - Базовый адрес в таблицу дескрипторов приемника. Не сбрасывается при аппаратном сбросе.

9: 3 Указатель дескриптора (DESCPNT) - Указатель на отдельные дескрипторы. Автоматически увеличивается Ethernet MAC контроллером.

2: 0 РЕЗЕРВ

Таблица 1.814 - Регистр GRETH_GBIT EDCL IP

31 — 0
EDCL IP АДРЕС

31: 0 IP адрес EDCL. Значение при сбросе устанавливается настройками rtl уровня.

Таблица 1.815 - GRETH_GBIT Таблица хешей msb

31 — 0
Таблица хешей (64 : 32)

31: 0 Хэш таблица MSB. Биты от 64 по 32 хеш таблицы.

Таблица 1.816 - GRETH_GBIT Таблица хешей lsb

31 — 0
Таблица хешей (31 : 0)

31: 0 Хэш таблица LSB. Биты от 31 по 0 хеш таблицы.

Таблица 1.817 - GRETH_GBIT EDCL MAC адрес MSB

31 — 16	15 — 0
РЕЗЕРВ	Биты с 47 по 32 EDCL MAC адреса

31: 16 РЕЗЕРВ

15: 0 Два наиболее значимых байта EDCL MAC адреса. Значения программного сброса устанавливается согласно настройкам rtl уровня.

Таблица 1.818 - GRETH_GBIT EDCL MAC адрес LSB

31	0
Биты с 31 по 0 EDCL MAC адреса	

31: 0 4 младших байта EDCL MAC адреса. Значения программного сброса устанавливается согласно настройкам rtl уровня.

1.3.11.2.2 Контроллер Ethernet 10/100 (GRETH)

Gaisler Research's Ethernet MAC (GRETH) контроллер обеспечивает интерфейс между AMBA- AHB шиной и сетью Ethernet. Он поддерживает скорость передачи 10/100 Мбит/сек в полудуплексном и дуплексном режимах.. AMBA интерфейс состоит из APB интерфейса для конфигурирования и управления и интерфейса AHB-мастера для управления потоком данных. Поток данных обрабатывается DMA-каналами. Один DMA– канал для передатчика и один - для приемника. Оба канала делят один и тот же AHB-мастер-интерфейс.Интерфейс Ethernet поддерживает оба MII и RMII интерфейсы, которые должны быть соединены с внешним PHY блоком. GRETH также обеспечивает доступ к MII Management интерфейсу, который используется для конфигурирования PHY блока

Ethernet Debug Communication Link (EDCL) протокол также поддерживается аппаратно в контроллере. Это протокол, базирующийся на UDP/IP протоколе, для удаленной отладки.

Блок-схема внутренней структуры GRETH приведена на рисунке Рисунок 1.151

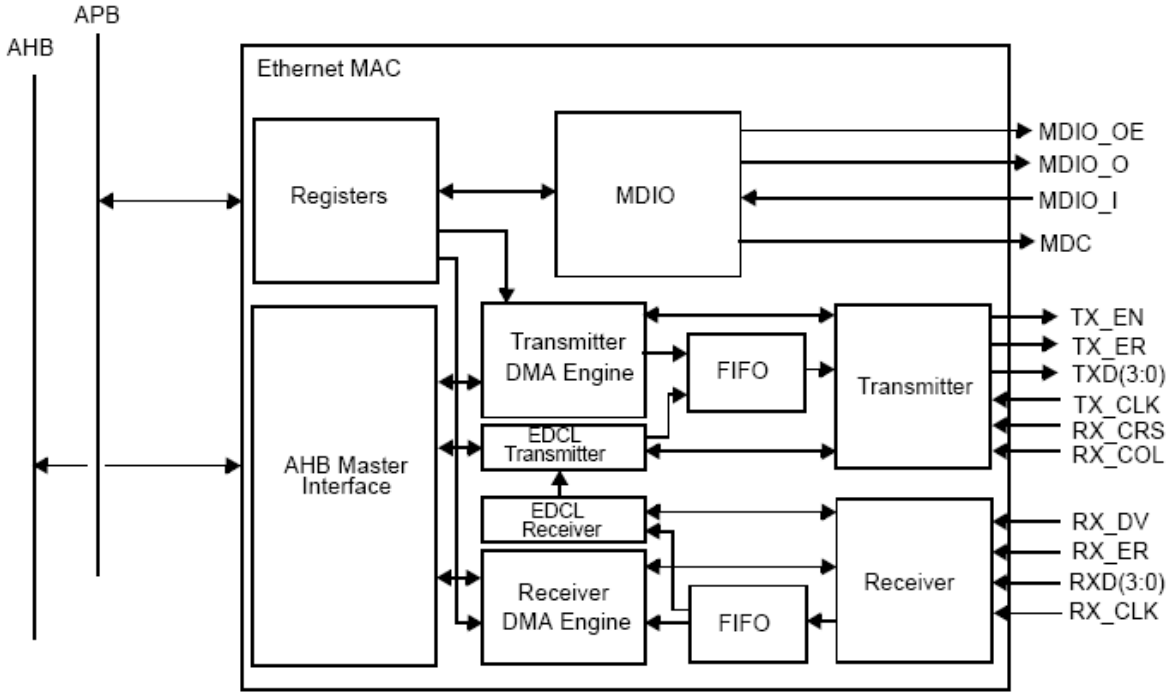


Рисунок 1.151 - Блок-схема внутренней структуры GRETH

1.3.11.2.2.1 Операции

1.3.11.2.2.1.1 Обзор системы

Контроллер GRETH состоит из 3-х функциональных устройств: DMA-каналов, MDIO-интерфейса и Ethernet Debug Communication Link (EDCL) блока.

Основная функциональность состоит из DMA-каналов, которые используются для передачи данных между AHB-шиной и сетью Ethernet. Используется один DMA-канал передатчика и один DMA-канал приемника. Операция с DMA-каналами управляется через регистры, доступные через APB-интерфейс.

MDIO-интерфейс используется для доступа к конфигурационным и статус-регистрам в один или более PHY-блоков, подключенных к MAC. Операция этого интерфейса также управляется через APB-интерфейс.

EDCL обеспечивает доступ по чтению и записи к AHB-шине через Ethernet . Он использует UDP, IP, ARP протоколы вместе с протоколом пользовательского уровня (custom application layer), чтобы выполнить это. EDCL не содержит регистров, доступных пользователю, и всегда работает параллельно с DMA-каналами.

Media Independent Interface (MII) используется для связс с PHY. Ethernet -передатчик посылает все данные из AHB домена в Ethernet, используя MII интерфейс. Соответственно, Ethernet - приемник сохраняет все данные из Ethernet на AHB-шине. Оба этих интерфейса используют FIFO памяти, когда передают потоки данных. GRETH также использует RMII, который использует подмножество этих сигналов.

EDCL и DMA-каналы делят Ethernet - приемник и передатчик.

1.3.11.2.2.1.2 Поддержка протоколов

GRETH–контроллер изготовлен согласно IEEE стандарту 802.3-2002. Не поддерживается используемое по выбору управление подуровнем и групповые (multicast) адреса не могут быть назначены для MAC. Это означает , что пакеты с типом 0x8808 (единственные в настоящее время определенные управляющие пакеты) отбрасываются.

1.3.11.2.2.1.3 Синхронизация

GRETH–контроллер имеет 3 домена синхроимпульсов. AHB синхроимпульс, синхроимпульс

Ethernet-приемника, синхроимпульс Ethernet-передатчика. Синхроимпульсы Ethernet-приемника и Ethernet-передатчика генерируются внешним Ethernet PHY и заходят в ядро (core) через MII интерфейс. 3 домена синхроимпульсов не связаны друг с другом и все сигналы пересекаюшие области синхроимпульсов полностью синхронизованы внутри ядра.

Оба рабочих режима – дуплексный и полудуплексный поддерживаются и оба могут выполняться со скоростью или в 10 или 100 Мбит. Минимальный AHB синхроимпульс для 10 Мбит – 2,5 МГц., в то время как 18 МГц необходимо для 100 Мбит. Использование более медленного AHB синхроимпульса приведет к завышенной потере пакетов.

1.3.11.2.2.1.4 Интерфейс передатчика Tx DMA

Интерфейс DMA-передатчика используется для передачи данных в Ethernet сеть.Передача сделана с помощью дескрипторов, помещенных в память.

1.3.11.2.2.1.4.1 Установка дескриптора

Один дескриптор показан на рисунке Рисунок 1.152 и рисунке Рисунок 1.153. Количество байтов, которое должно быть передано, устанавливается в поле длины и поле адреса указывает на данные. Адрес должен быть выровнен с точностью до слова. Если бит разрешения прерывания interrupt enable (IE) установлен, прерывание будет сгенерировано, когда пакет был послан (это требует , чтобы бит прерывания передатчика в управляющем регистре также был

установлен). Прерывание будет сгенерировано независимо от того, был ли пакет передан успешно или нет. Бит Свертки Wrap(WR) , также управляющий бит, который будет установлен перед передачей, и он будет объяснен позже в этом разделе.

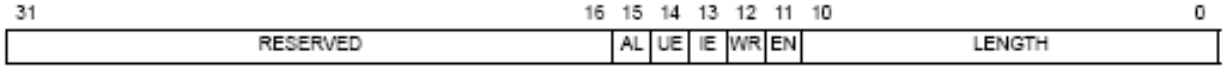


Рисунок 1.152 – Дескриптор передатчика GRETH слово 0 (смещение адреса (address offset) 0x0)

31:16 не используется.

15 Attempt Limit Error (AL) – Пакет не был передан, потому было достигнуто максимальное количество попыток.

14 Underrun Error (UE) – Пакет был некорректно передан из-за ошибки переполнения FIFO.

13 Interrupt enable (IE) – Разрешение прерываний. Прерывание было сгенерировано, когда пакет от этого дескриптора был послан, обеспеченный тем, что Interrupt enable бит в управляющем регистре установлен. Прерывание сгенерировано независимо от того, успешно был передан пакет или он был завершен с ошибкой

12 Wrap(WR) –Устанавливается в 1, чтобы заставить указатель дескриптора свернуться в 0 после того, как дескриптор был использован. Если этот бит не установлен, указатель будет увеличен на 8 . Указатель автоматически сворачивается в 0, когда граница 1 кВ дескриптора достигнута.

11 Enable(EN) - Устанавливается в 1, чтобы сделать дескриптор значимым. Будет всегда устанавливаться последним из всех дескрипторных полей.

10:0 LENGTH – количество байтов, которое должно быть передано.

Дескриптор передатчика GRETH слово 1 (смещение адреса (address offset) 0x4)

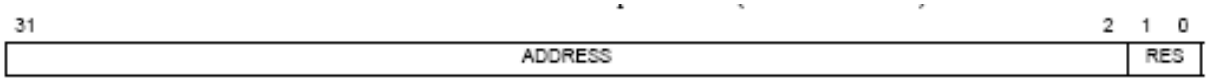


Рисунок 1.153 – Дескриптор передатчика GRETH слово 1 (смещение адреса (address offset) 0x0)

31:2 Address (ADDRESS) – Указатель на область буфера, откуда данные будут загружены..

1:0 не используется.

1.3.11.2.2.1.4.2 Начало передачи

Значимый дескриптор не достаточен, чтобы начать передачу. Сначала в GRETH должен быть установлен указатель области памяти, содержащей дескрипторы. Это сделано в регистре указателя дескриптора передатчика.. Адрес должен быть выровнен к 1кБ границе. Разряды с 31 по 10 содержат базовый адрес области дескрипторов, в то время как разряды с 9 по 3 формируют указатель на индивидуальный дескриптор. Первый дескриптор будет помещен по базовому адресу и, когда он будет использован GRETH-контроллером, поле указателя будет увеличено на 8, чтобы указать на следующий дескриптор. Указатель будет автоматически свернут обратно в 0, когда следующая 1кБ граница будет достигнута (дескриптор со смещением адреса 0x3F8 был использован). Разряд WR в дескрипторах может быть установлен , чтобы сделать свертку указателя к 0 обратно до 1кБ границы.

Поле указателя также сделано доступным по записи для максимальной гибкости, но нужно соблюдать осторожность при записи указателя дескриптора в регистр. Его никогда нельзя трогать, когда передача активна.

Последний шаг для активизации передачи - это установка разряда разрешение передачи (transmit enable) в управляющем регистре. Это говорит GRETH- контроллеру , что еще есть активные дескрипторы в таблице дескрипторов. Этот разряд будет всегда установлен, когда новые дескрипторы значимы., даже если передачи уже активны. Дескрипторы должны всегда быть значимы до того, как разряд разрешения передачи (transmit enable bit) установлен.

1.3.11.2.2.1.4.3 Обработка дескриптора после передачи

Когда передача пакета закончена, статус записывается в первое слово соответствующе-го дескриптора. Разряд Ошибки Неполного Завершения (Underrun Error) устанавливается, если FIFO стало пустым до того, как пакет был полностью передан, когда разряд Ошибки исчерпания Предела Попыток (Attempt Limit Error) установлен , если произошло больше столкновений, чем разрешено. Пакет был успешно передан, если оба этих разряда равны 0. Другие разряды первого слова дескриптора устанавливаются в 0 после передачи, в то время как второе слово остается нетронутым.

Разряд разрешения (enable bit) будет использован как индикатор, когда дескриптор может быть использован снова, тогда, когда он был обнулен контроллером GRETH. В GRETH 3 разряда, которые содержат статус передачи. Ошибка передатчика Transmission Error (TE) устанавливается каждый раз, когда передача заканчивается с ошибкой (когда по крайней мере один из 2 разрядов статуса в передаваемом дескрипторе был установлен). Прерывание передатчика Transmitter Interrupt (TI) устанавливается каждый раз, когда передача закончилась успешно.

Ошибка передатчика AHB Transmitter AHB Error (TA) , когда ошибка AHB встретилась или во время чтения дескриптора или во время чтения пакета данных. Любые активные передачи будут прекращены и передатчик будет незначимым. Передатчик может быть активирован снова установкой регистра разрешения передачи (transmit enable register).

1.3.11.2.2.1.4.4 Установка данных передачи

Данные для передачи будут помещены, начиная с адреса, указанного в поле адреса дескриптора. Контроллер GRETH не прибавляет Ethernet адрес и поля типа так, чтобы они также должны были храниться в буфере данных. 4 байта Ethernet CRC автоматически добавляются в конце каждого пакета. Каждый дескриптор будет посылаться как единственный Ethernet пакет. Если поле размера в дескрипторе больше, чем 1514 байт, пакет не будет передан.

1.3.11.2.2.1.4.5 Интерфейс приемника Rx DMA

Интерфейс DMA-приемника используется для приема данных из сети Ethernet. Прием делается с использованием дескрипторов, помещенных в память.

1.3.11.2.2.1.4.5.1 Установка дескрипторов

Один дескриптор показан на рисунке Рисунок 1.154 и рисунке Рисунок 1.155. Поле адреса будет указывать на выровненный до слова буфер, где принятые данные будут храниться. GRETH никогда не будет хранить в буфере больше, чем 1514 байт. Если разряд разрешения прерывания interrupt enable (IE) установлен, прерывание будет сгенерировано, когда пакет будет принят в этот буфер (это требует, чтобы разряд прерывания приемника в управляющем регистре также был установлен) Разряд свертки Wrap (WR) также управляющий разряд, который будет установлен до того как дескриптор значимый и будет объяснен позже в этом разделе.

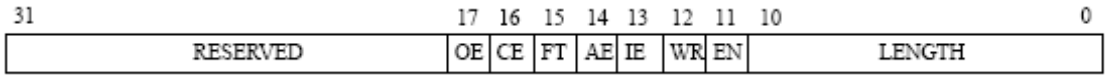


Рисунок 1.154 – Дескриптор приемника GRETH слово 0 (смещение адреса (address offset) 0x0)

31:16 не используется.

18 Length error (LE) - Поле длины.типа пакета не совпадает с действительным числом принятых байтов.

17 Overrun error (OE) - Кадр (frame) был некорректно принят из-за переполнения FIFO.

16 CRC error (CE) - Ошибка CRC была определена в этом кадре (frame).

15 Frame too long (FT) – Кадр больший, чем максимальный размер был принят. Лишняя часть будет отброшена.

14 Alignment error (AE) – Нечетное число полубайтов было принято.

13 Interrupt enable (IE) – Разрешение прерываний. Прерывание было сгенерировано, когда пакет от этого дескриптора был принят, обеспеченный тем, что Interrupt enable бит в управляющем регистре установлен. Прерывание сгенерировано независимо от того, успешно был принят пакет или он был завершен с ошибкой

12 Wrap(WR) –Устанавливается в 1, чтобы заставить указатель дескриптора свернуться в 0 после того, как дескриптор был использован. Если этот бит не установлен, указатель будет увеличен на 8 . Указатель автоматически сворачивается в 0, когда граница 1 кВ дескриптора достигнута.

11 Enable(EN) - Устанавливается в 1, чтобы сделать дескриптор значимым. Будет всегда устанавливаться последним из всех дескрипторных полей.

10:0 LENGTH – количество байтов, которое должно быть передано.

Дескриптор приемника GRETH слово 1 (смещение адреса (address offset) 0x4)

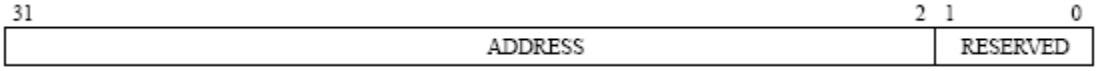


Рисунок 1.155 – Дескриптор приемника GRETH слово 1 (смещение адреса (address offset) 0x4)

31:2 Address (ADDRESS) – Указатель на область буфера, куда данные будут загружены..

1:0 не используется.

1.3.11.2.2.1.4.5.2 Начало приема

Значимый дескриптор не достаточен, чтобы начать прием. Сначала в GRETH должен быть установлен указатель области памяти, содержащей дескрипторы. Это сделано в регистре указателя дескриптора приемника. Адрес должен быть выровнен к 1кБ границе. Разряды с 31 по 10 содержат базовый адрес области дескрипторов, в то время как разряды с 9 по 3 формируют указатель на индивидуальный дескриптор. Первый дескриптор будет помещен по базовому адресу и, когда он будет использован GRETH-контроллером, поле указателя будет увеличено на 8, чтобы указать на следующий дескриптор. Указатель будет автоматически свернут обратно в 0, когда следующая 1кБ граница будет достигнута (дескриптор со смещением адреса 0x3F8 был использован). Разряд WR в дескрипторах может быть установлен , чтобы сделать свертку указателя к 0 обратно до 1 кБ границы.

Поле указателя также сделано доступным по записи для максимальной гибкости, но нужно соблюдать осторожность при записи указателя дескриптора в регистр. Его никогда нельзя трогать, когда прием активен.

Последний шаг для активизации приема - это установка разряда разрешение приема (receive enable) в управляющем регистре. Это заставляет GRETH- контроллер считывать первый дескриптор и ожидать входной пакет.

1.3.11.2.2.1.4.5.3 Обработка дескриптора после приема

GRETH- контроллер указывает на окончание приема сбросом разряда разрешения дескриптора (descriptor enable bit). Другие управляющие разряды (WR,IE) также сбрасываются. Количество принятых байтов показывается в поле длина. Сохраняемые части Ethernet-кадра - это адрес назначения, адрес источника, тип и поля данных. Разряды 17-14 в первом слове дескриптора – статус разряды , указывающие различные ошибки приема. Все четыре разряда – нули после приема без ошибок. Разряды статуса описаны на рисунке Рисунок 1.154.

Если пакет принят с адресом, не принятым MAC, регистр статуса IA, будет установлен.

Пакеты размера больше максимального размера вызывают установку FT разряда. Не гарантируется, что поле длина содержит правильное значение принятых байтов. Счетчик останавливается, когда слово, содержащее последний байт вплоть до предельно максимального размера , было записано в память.

Адресное слово дескриптора никогда не трогается контроллером GRETH.

1.3.11.2.2.1.4.5.4 Прием с AHB ошибками

Если ошибка AHB происходит во время чтения дескриптора или сохранения данных, разряд AHB ошибки приемника (Receiver AHB Error) будет установлен и приемник отключен. Текущий прием будет отброшен. Приемник будет включен снова установкой разряда разрешения приема (Receive Enable) в управляющем регистре.

1.3.11.2.2.1.4.6 MDIO интерфейс

MDIO интерфейс обеспечивает доступ к PHY конфигурационным и статус – регистрам через двухпроводной интерфейс, который включен через MII интерфейс . GRETH обеспечил полную поддержку для MDIO интерфейса. Если это не нужно в устройстве, он может быть удален в VHDL generic .

MDIO интерфейс может быть использован для доступа к 1 до 32 PHY, содержащих от 1 до 32-х 16-разрядных регистров. Установка передачи по чтению i производится записью PHY и регистровых адресов в MDIO управляющий регистр и установкой разряда чтения. Это вызывает установку разряда Занятости (Busy bit) и операция заканчивается, когда разряд Занятости (Busy bit). сбрасывается. Если операция была успешной, Linkfail разряд равен 0, и поле данных содержит считанные данные. Неуспешная операция указывается установкой Linkfail разряда. В этом случае поле данных не определено.

Операция записи начинается записью 16- разрядных данных, PHY адреса и регистрового адреса в MDIO Управляющий регистр установкой разряда записи. Операция заканчивается, когда разряд занятости (busy bit) сбрасывается и она была успешной, если Linkfail разряд равен 0.

1.3.11.2.2.1.4.7 Ethernet Debug Communication Link (EDCL) интерфейс для удаленной отладки

EDCL обеспечивает доступ к AHB-шине,находящейся в микросхеме, через Ethernet. Он использует UDP, IP и ARP протоколы вместе с протоколом прикладного пользовательского уровня . Протокол прикладного уровня использует ARQ алгоритм для обеспечения передачи AHB команд. Через эту связь передача по чтению или записи может быть сгенерирована по любому адресу на AHB-шине. EDCL необязательный и может быть разрешен в generic.

1.3.11.2.2.1.4.7.1 Операция

EDCL получает пакеты параллельно с приемным MAC DMA каналом. Он использует отдельный MAC адрес, который используется для различения EDCL-пакетов от пакетов, предназначенных MAC DMA каналу. EDCL также имеет IP адрес, который устанавливается через generics. Так как ARP пакеты используют Ethernet широковещательный адрес, IP адрес должен быть использован в этом случае, чтобы различить EDCL ARP пакеты от тех, которые идут в DMA канал. Пакеты, которые определены быть EDCL-пакетами, не обрабатываются приемным DMA каналом.

Когда пакеты проверены на правильность, AHB операция выполнена. Операция выполняется тем же самым AHB-мастер интерфейсом, который использует DMA-обработчик(engine). Ответы автоматически посылаются EDCL-передатчиком, когда операция закончена. Он разделяет Ethernet- передатчик с передатчиком DMA-обработчика(engine), но имеет более высокий приоритет.

1.3.11.2.2.1.4.7.2 EDCL протоколы

EDCL получает Ethernet-кадры, содержащие IP или ARP данные. ARP обрабатываются согласно спецификации протокола без всяких исключений.

IP пакеты несут действительные AHB команды. EDCL ожидает Ethernet-кадр, содержвщий IP, UDP и EDCL специфические части прикладного уровня. На рисунке Рисунок 1.156 показан IP пакет, требуемый EDCL. Содержание различных протокольных заголовков может быть найдено в TCP/IP литературе.

Ethernet Header	IP Header	UDP Header	2 B Offset	4 B Control word	4 B Address	Data 0 - 242 4B Words	Ethernet CRC

Рисунок 1.156 – IP пакет, требуемый EDCL

Для успешной связи по EDCL требуется следующее: правильный MAC адрес назначения, как указано в generics, Ethernet поле типа, содержащее 0x806 (ARP) или 0x800 (IP) . IP– адрес тогда сравнивается со значением, определенном в generics на совпадение. Контрольная сумма IP– заголовка (IP-header) и идентификационные поля не проверяются. Существует несколько ограничений полей IP-header. Версия должна быть равна 4 и размер заголовка должен быть 5 B(без выбора). Поле протокола (protocol field) должно быть всегда 0x11 , указывающее на UDP пакет. Длина и контрольная сумма – только IP–поля, изменяемые в ответе.

EDCL обспечивает только одно обслуживание в данный момент и поэтому не требуется проверять номер UDP–порта.Ответ будет иметь исходный номер порта источника в обоих полях источника и назначения. UDP контрольная сумма не используется и поле контрольной суммы равно 0 в ответах.

UDP поле данных содержит EDCL поля прикладного уровня. На рисунке Рисунок 1.157 показаны поля протокола прикладного уровня (поля данных исключены) в пакетах, принятых EDCL. 16-разрядное смещение используется для того, чтобы выровнять оставшуюся часть данных прикладного уровня к словным границам в памяти и может быть установлено таким образом в любое значение. Поле R/W , определяет будет выполняться чтение или запись Поле длины

16-bit Offset	14-bit Sequence number	1-bit R/W	10-bit Length	7-bit Unused

Рисунок 1.157 – Поля прикладного уровня в принимаемых кадрах

содержит количество байтов, которое нужно считать или записать. Если R/W равен 1, поле данных, содержит данные, которые должны быть записаны. Если R/W равен 0, поле данных пусто в принимаемых пакетах. На Рисунок 1.158 показаны поля прикладного уровня ответов от EDCL. Поле длины всегда 0 для ответов на запросы записи. Для запросов чтения оно содержит количество байтов данных, содержащихся в поле данных.

16-bit Offset	14-bit sequence number	1-bit ACK/NAK	10-bit Length	7-bit Unused

Рисунок 1.158 – Поля прикладного уровня в передаваемых кадрах

EDCL использует Go-Back-N алгоритм, обеспечивающий надежные передачи. 14-разрядный номер последовательности в принимаемых пакетах проверяется на совпадение с внутренним счетчиком. Если они не совпадают, никакая операция не выполняется и ACK/NAK поле устанавливается в 1 в ответном кадре. Ответный кадр содержит значение внутреннего счетчика в поле номер последовательности. Если номер последовательности совпадает, внутренний счетчик увеличивается, значение внутреннего счетчика запоминается в поле номера последовательномти и ACK/NAK поле устанавливается в 0 в ответе. Поле длины всегда устанавливается в 0 для кадров с ACK/NAK = 1.Неиспользуемые поля не проверяются и записываются в ответ. Это могут быть установлены, взятые для примера, некоторые дополнительные разряды идентификатора, если необходимо.

1.3.11.2.2.1.4.7.3 Установка EDCL IP и Ethernet адреса

Значение по умолчанию EDCL IP и MAC-адресов установлено в ipaddrh, ipaddrl, macaddrh и macaddrl generics. IP адрес может быть позже изменен программно, но MAC адрес зафиксирован. Чтобы обеспечить несколько GRETH-котроллеров с разрешенным EDCL в одной и той же подсети, 4 LSB разряда IP и MAC адреса могут быть выборочно установлены входным сигналом. Это достигается установкой edcl generics =2 и выдачей 4-х разряд-ного значения LSB на ethi.edcladr..

1.3.11.2.2.1.4.7.4 EDCL ограничения

EDCL разработан для работы без программного вмешательства , требуя таким образом автоматическую конфигурацию аппаратуры. Для того, чтобы упростить аппаратуру предполагается , что EDCL имеет способность обрабатывать все рабочие режимы PHY и только считывает авто-переговорный режим и конфигурирует MAC. Это предохраняет его от работы с гигабитными PHY, так как они они могут быть в гигабитном режиме и это не может быть управляемо с помощью MAC.

1.3.11.2.2.1.4.8 Не зависящие от среды (Media Independent) интерфейсы

Имеется несколько интефейсов, определяемых между подуровнем MAC и физическим уровнем. GRETH поддерживает два из них Media Independent интерфейс(MII) и Reduced Media Independent интерфейс (RMII) .

MII был определен в 802.3 стандарте и наиболее часто поддерживаемый. Ethernet интерфейс был изготовлен в соответствии с этой спецификацией. Он использует 16 сигналов. RMII был развит столкнувшись с необходимостью интерфейса для Ethernet – контроллеров с малым количеством выводов. Он использует 6(7) сигналов, которые являются подмножеством

MII сигналов. Таблица 1.819 показывает соответствие между RMII сигналами и GRLIB MII интерфейсом.

Таблица 1.819 – Соответствие (Mapping) сигналов между RMII и GRLIB MII интерфейсом

RMII	MII
txd[1:0]	txd[1:0]
tx_en	tx_en
crs_dv	rx_crs
rxd[1:0]	rxd[1:0]
ref_clk	rmii_clk
rx_er	not used

1.3.11.2.2.1.5 Программные драйверы

Драйверы для MAC поддержаны следующими операционными системами: RTEMS, eCos, uClinux и Linux-2.6 . Драйверы находятся в свободном доступе в виде полного source -кода по GPL лицензии от Gaisler Research"s web site (http://gaisler.com)

1.3.11.2.2.1.6 Регистры

Ядро (core) программируется через регистры, отображаемые (mapped) в адресное пространство APB .(см.таблицу Таблица 1.820)

Таблица 1.820 – GRETH регистры

Register	APB Address offset
Control register	0x0
Status/Interrupt-source register	0x4
MAC Address MSB	0x8
MAC Address LSB	0xC
MDIO Control/Status	0x10
Transmit descriptor pointer	0x14
Receiver descriptor pointer	0x18
EDCL IP	0x1C

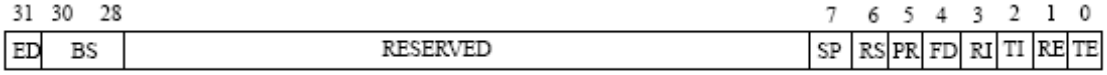


Рисунок 1.159 – GRETH управляющий регистр

EDCL available (ED) - Устанавливается, если EDCL разрешен.
30:28 EDCL buffer size (BS) – Показывает количество памяти, используемое в EDCL буферах. 0=1кВ, 1=2кВ, …6=64кВ.
27:11 Не используется.

PHY status change interrupt enable (PI) – Разрешить прерывания для определенных изменений PHY- статуса.

9:8 Не используется.

7 Speed (SP) - Устанавливает текущий режим скорости. 0= 10 Mbit, 1=100 Mbit.

Используется только в RMII режиме(RMII=1). Значение по умолчанию автоматически считывается из PHY после обнуления. Reset -значение “1”.

6 Reset (RS) - 1, записанная в этот разряд, обнуляет ядро (core) GRETH. Сам себя сбрасывает.

5 Promiscuous mode (PM) – Если установлен, GRETH работает в смешанном режиме, который означает, что он будет получать все пакеты независимо от адреса назначения.

4 Full duplex (FD) - Если установлен, GRETH работает в дуплексном режиме, иначе он работает в полудуплексе. Значение по Reset “0”.

3 Receiver Interrupt (RI) – Разрешает прерывания приемника. Прерывание будет сгенерировано каждый раз, когда пакет передан, когда разряд установлен в 1. Прерывание генерируется независимо от того, был ли пакет принят успешно или закончен с ошибкой. Значение по Reset “0”.

2 Transmitter Interrupt (TI) - Разрешает прерывания передатчика. Прерывание будет сгенерировано каждый раз, когда пакет принят, когда разряд установлен в 1. Прерывание генерируется независимо от того, был ли пакет передан успешно или закончен с ошибкой . Значение по Reset “0”.

1 Receive enable (RE) – Будет установлен всегда, когда новые дескрипторы разрешены. До тех пор, пока этот разряд 1, GRETH будет считывать новые дескрипторы и как только он встретит неразрешенный дескриптор, он остановится до тер пор, пока RE не установится снова. В этот разряд будет записана 1, после того, как новый дескриптор разрешен. Значение по Reset “0”.

0 Transmit enable (TE) - Будет установлен всегда, когда новые дескрипторы разрешены. До тех пор, пока этот разряд 1, GRETH будет считывать новые дескрипторы и как только он встретит неразрешенный дескриптор, он остановится до тер пор, пока TE не установится снова. В этот разряд будет записана 1, после того, как новый дескриптор разрешен. Значение по Reset “0”.

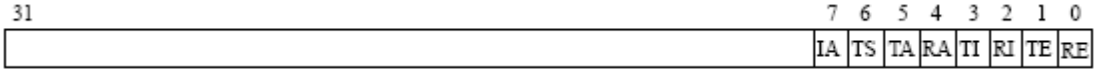


Рисунок 1.160 – GRETH управляющий регистр GRETH статус - регистр

31:8 Не используется.

Invalid address (IA) – Пакет с адресом, не принятым MAC, был принят. Сбрасывается, когда записывается 1. Значение по Reset “0”.

Too small (TS) – был принят пакет меньше минимального размера. Сбрасывается, когда записывается 1. Значение по Reset “0”.

Transmitter AHB error (TA) – AHB- ошибка встретилась при работе DMA-передатчика. Сбрасывается, когда записывается 1. Значение не обнуляется.

4 Receiver AHB error (RA) - AHB- ошибка встретилась при работе DMA-приемника. Сбрасывается, когда записывается 1. Значение не обнуляется.

3 Transmitter Interrupt (TI) – Пакет был передан без ошибок. Сбрасывается, когда записывается 1. Значение не обнуляется

2 Receiver interrupt (RI) – Пакет был принят без ошибок. Сбрасывается, когда записывается 1. Значение не обнуляется

1 Transmit error (TE) - Пакет был передан с ошибкой. Сбрасывается, когда записывается 1. Значение не обнуляется

0 Receiver error (RE) - Пакет был принят с ошибкой. Сбрасывается, когда записывается 1. Значение не обнуляется

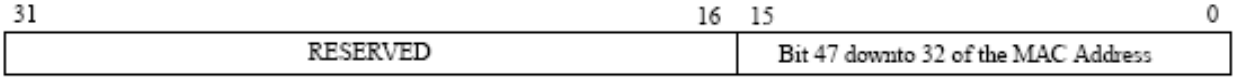


Рисунок 1.161 – GRETH MAC – адрес MSB

31:16 Не используется.
15:0 Два старших значащих байта MAC – адреса. Не обнуляются

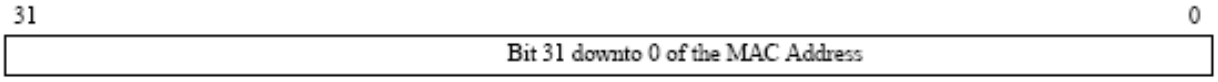


Рисунок 1.162 – GRETH MAC – адрес LSB

31:0 Четыре младших значащих байта MAC – адреса. Не обнуляются

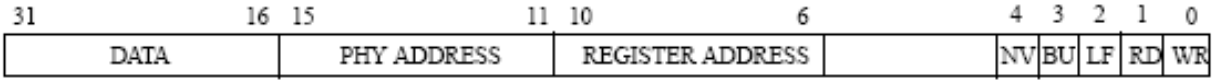


Рисунок 1.163 – GRETH MDIO управляющий/статус регистр

31:16 Data (DATA) – Содержит данные, считываемые во время операции чтения и данные, которые передаются, берутся из этого поля. Значение по Reset -0x0000.
15:11 PHY address (PHYADDR) – Это поле содержит адрес PHY, который будет выбираться во время операции чтения или записи.. Значение по Reset -0x0000.
10:6 Register address (REGADDR) - Это поле содержит адрес регистра, который будет выбираться во время операции чтения или записи.. Значение по Reset -0x0000
Не используется.
Not valid (NV) – Когда операция закончена (BUSY =0), этот разряд указывает были ли приняты значащие данные, что означает , что поле данных содержит правильные данные. Значение по Reset “0”.
3 Busy (BU) - Когда операция выполняется, этот разряд установлен в 1. Как только операция закончена и управляющая связь (management link) находится в ожидании , этот разряд сбрасывается. Значение по Reset “0”.
2 Linkfail (LF) – Когда операция завершается (BUSY =0) этот разряд установлен, если функциональная управляющая связь (management link) была не определена. Значение по Reset “1”.
1 Read (RD) – Начать операцию чтения в управляющем интерфейсе. Данные хранятся в поле данных. Значение по Reset “0”.
0 Write (WR) - Начать операцию записи в управляющем интерфейсе. Данные берутсяя из поля данных. Значение по Reset “0”.

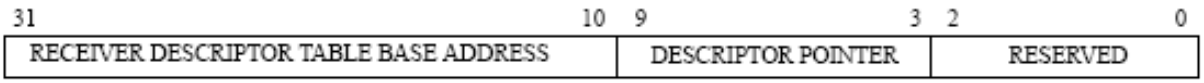


Рисунок 1.164 – GRETH регистр базового адреса таблицы дескрипторов передатчика

31:10 Transmitter descriptor table base address (BASEADDR) – Базовый адрес таблицы дескрипторов передатчика. Не обнуляется.

9:3 Descriptor pointer (DESCPNT) – Указатель на индивидуальные дескрипторы. Автоматически увеличивается контроллером Ethernet MAC.

2:0 Не используется

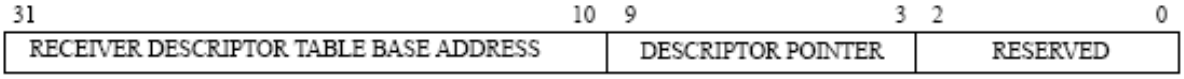


Рисунок 1.165 – GRETH регистр базового адреса таблицы дескрипторов приемника

31:10 Receiver descriptor table base address (BASEADDR) – Базовый адрес таблицы дескрипторов приемника. Не обнуляется.

9:3 Descriptor pointer (DESCPNT) – Указатель на индивидуальные дескрипторы. Автоматически увеличивается контроллером Ethernet MAC.

2:0 Не используется

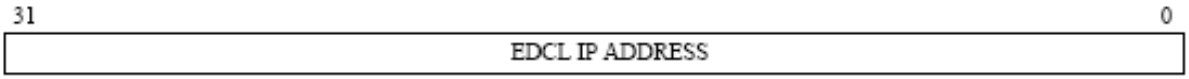


Рисунок 1.166 – GRETH EDCL IP регистр

31 -0 EDCL IP адрес. Значение при обнулении 0xC0A80101.

1.3.11.2.3 Контроллер NAND флеш-памяти с поддержкой ECC

Контроллер NAND флэш-памяти используется в качестве моста между внешней микросхемой флэш-памяти и ведущей главной системой. Блок обеспечивает широкую поддержку микросхем флэш-памяти благодаря аппаратным алгоритмам коррекции ошибок, поддерживающим режим до 24 ошибок/1 КБ, и программируемым размерам страницы и блока. Блок обеспечивает полную поддержку асинхронного режима ONFI 2.2 (Open NAND Flash Interface – открытый интерфейс памяти NAND). Встроенный контроллер ПДП уменьшает нагрузку на ЦП.

Основные характеристики:

- Настраиваемые значения задержек
- Поддержка набора команд стандарта ONFI 2.2
- Поддержка только асинхронного интерфейса
- Внутренний буфер FIFO на 2 КБ
- Конфигурируемый размер страницы
- 8-битная шина данных
- Аппаратная коррекция ошибок - 2 алгоритма: 4 ошибки на 512 байт и 24 ошибки на 1024 байт
- Поддержка работы по прерыванию
- Аппаратная защита от стирания
- Поддержка чипов SLC и MLC флэш-памяти

- Конфигурируемая плотность кристалла
- Конфигурируемый размер блока

- Конфигурируемый размер страницы от 0,5 КБ до более 128 КБ
- ПДП с настройкой порядка следования байтов (прямой и обратный)

1.3.11.2.3.1 Обзор функциональных характеристик

На рисунке Рисунок 1.167 - Внутренняя структура контроллера NAND продемонстрирована внутренняя структура контроллера NAND. Контроллер содержит 5 блоков.

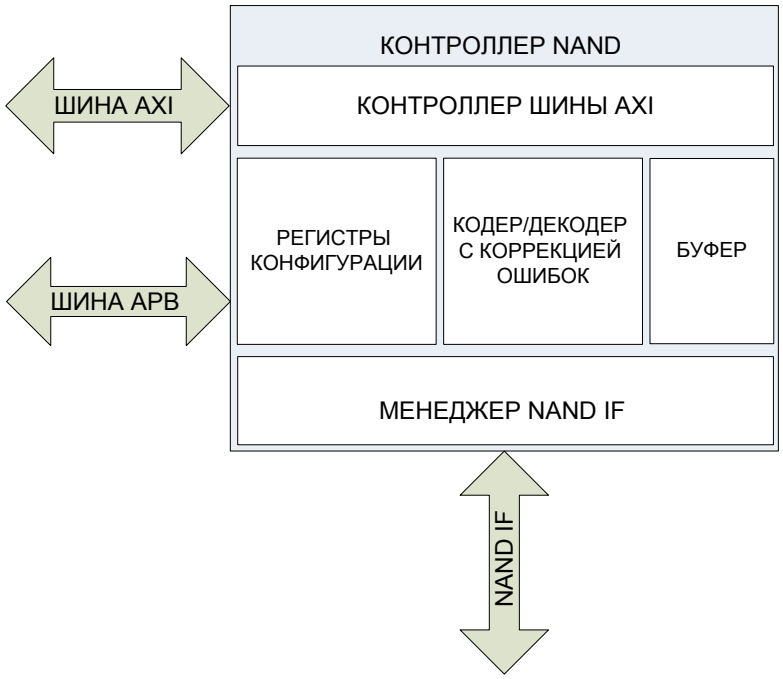


Рисунок 1.167 - Внутренняя структура контроллера NAND

КОНТРОЛЛЕР ШИНЫ AXI

Функциональный блок контроллера шины AXI используется для приема и отправки данных с/на системную шину

РЕГИСТРЫ КОНФИГУРАЦИИ

Блок регистров конфигурации используется для приема через интерфейс шины APB и последующего сохранения параметров конфигурации

КОДЕР/ДЕКОДЕР С КОРРЕКЦИЕЙ ОШИБОК

Функциональный блок кодера/декодера с коррекцией ошибок производит оценку кодов коррекции ошибок во время операция считывания и записи во флэш-память NAND

БУФЕР

Внутренний буфер контроллера NAND используется для хранения данных, считанных из флэш-памяти NAND, до передачи их по системной шине

МЕНЕДЖЕР NAND IF

МЕНЕДЖЕР NAND IF осуществляет все операции, связанные с интерфейсом флэш-памяти NAND, отвечающим за уровень контакта

1.3.11.2.3.2 Программный интерфейс

1.3.11.2.3.2.1 Карта регистров

Таблица 1.698 содержит карту регистров контроллера NAND с описанием функционирования.

Таблица 1.821 - Карта регистров

Адрес	Доступ	Наименование	Описание
0x00	чтение	nand_status	Состояние контроллера и флэш-памяти
0x04	чтение/запись	nand_control	Рабочие параметры контроллера
0x08	чтение/запись	nand_command	Задание типа команды
0x0C	чтение/запись	nand_col_addr	Адрес столбца
0x10	чтение/запись	nand_row_addr_read_d	Адрес и данные строки
0x14	чтение/запись	irq_mask_nand	Задание масок прерываний
0x18	чтение/запись	cdb_buffer_enable	Разрешение чтения для ведущего интерфейса AXI
0x1C	чтение/запись	cdb_buffer_type	Конфигурация буфера
0x20	чтение	cdb_ch0_buffer_status	Состояние буфера ведущего интерфейса чтения AXI
0x24	чтение/запись	axi_arlen	Конфигурация ARLEN шины AXI
0x28	чтение/запись	sw_rstn_r	Программный сброс ведущего интерфейса чтения AXI
0x2C		reserved	
0x30	чтение	axi_if_err_addr	Адрес ошибочной транзакции по шине AXI
0x34	чтение	axi_if_status	Регистр состояния ведущего интерфейса чтения AXI
0x38	чтение/запись	msb_lsbr	Конфигурация порядка следования байтов ведущего интерфейса чтения AXI
0x3C		reserved	
0x40	чтение/запись	start_dma_r	Адрес первого байта данных, подлежащих считыванию с памяти
0x44	чтение/запись	end_dma_r	Адрес последнего байта данных, подлежащих считыванию с памяти
0x48	чтение	current_dma_r	Адрес текущих считываемых данных
0x4C	чтение/запись	cntrl_dma_r	Регистр конфигурации для управления адресами для ведущего интерфейса чтения AXI
0x50	чтение/запись	nand_cntrl_sw_rst	Программный сброс контроллера NAND
0x54	чтение	nand_id	Регистр идентификации контроллера NAND
0x58	чтение	nand_version	Версия контроллера NAND
0x5C		reserved	
0x60		reserved	
0x64		reserved	
0x68		reserved	
0x6C	чтение/запись	awlen_max	Конфигурация AWLEN шины AXI
0x70	чтение/запись	sw_rst	Программный сброс ведущего интерфейса записи AXI
0x74	чтение/запись	msb_lsbw	Конфигурация порядка следования байтов ведущего интерфейса записи AXI
0x78	чтение	respw_status	Состояние ошибки транзакции по шине AXI
0x7C	чтение	status_full	Регистр состояния внутреннего буфера ведущего интерфейса записи AXI
0x80		reserved	
0x84		reserved	
0x88	чтение/запись	start_dma_w	Адрес первого байта данных, подлежащих записи в память
0x8C	чтение/запись	end_dma_w	Адрес последнего байта данных, подлежащих записи в память
0x90	чтение/запись	current_dma_w	Адрес текущих записываемых данных
0x94	чтение/запись	cntrl_dma_w	Регистр конфигурации для управления адресами для ведущего интерфейса записи AXI
0x9C		reserved	
0xA0		reserved	

Адрес	Доступ	Наименование	Описание
0xA4		reserved	
0xA8		reserved	

Продолжение таблицы 1.822

Адрес	Доступ	Наименование	Описание
0xAC		reserved	
0xB0		reserved	
0xB4	чтение	irq_status	Состояния контроллера
0xB8	чтение/запись	nand_timing_0	Величины синхронизации для tADL, tALH, tALS, tCH
0xBC	чтение/запись	nand_timing_1	Величины синхронизации для tCLH, tCLS, tCS, tDH
0xC0	чтение/запись	nand_timing_2	Величины синхронизации для tDS, tWC, tWH, tWP
0xC4	чтение/запись	nand_timing_3	Величины синхронизации для tWW, tAR, tCLR
0xC8	чтение/запись	nand_timing_4	Величины синхронизации для tCOH, tIR, tRC, tREH
0xCC	чтение/запись	nand_timing_5	Величины синхронизации для tRHOH, tRHW, tRLOH, tRP
0xD0	чтение/запись	nand_timing_6	Величины синхронизации для tRR, tWHR, tCEA, tCHZ
0xD4	чтение/запись	nand_timing_7	Величины синхронизации для tREA, tRHZ, tWB

1.3.11.2.3.2.2 Подробное описание регистров

1.3.11.2.3.2.2.1 nand_status

Адрес: 0x0. Состояние контроллера NAND и флэш-памяти

Таблица 1.823 - Поля регистра nand_status.

Биты	Описание	Значение по умолчанию
[7:0]	После выполнения команд контроллера NAND флэш-памяти данное поле содержит значение регистра состояния флэш-памяти. Описание приведено в документации на контроллер NAND флэш-памяти.	0
[8]	Резерв Ошибка при попытке коррекции. 1 – возникновение некорректируемой ошибки 0 – отсутствие ошибок или успешное исправление всех возникших ошибок	0
[9]	Готовность контроллера NAND к приему следующей команды 1 – Контроллер NAND готов 0 – Контроллер NAND не готов	1
[10]	Готовность флэш-памяти NAND к выполнению операций 1 – флэш-память готова 0 – флэш-память не готова	1
[31:11]	Резерв	0

1.3.11.2.3.2.2.2 nand_control

Адрес: 0x4. Рабочие параметры контроллера NAND

Таблица 1.824 - Поля регистра nand_control

Биты	Описание	Значение по умолчанию
[0]	Флаг начала операции 1: немедленное выполнение команды 0: не начинать выполнение команды немедленно	0

Продолжение таблицы 1.825

<table>
<tr><th>Биты</th><th>Описание</th><th>Значение по умолчанию</th></tr>
<tr><td>[3:1]</td><td><table><tr><td>Размер страницы*</td><td>Размер страницы, байты</td></tr><tr><td>0x0</td><td>512</td></tr><tr><td>0x1</td><td>1024</td></tr><tr><td>0x2</td><td>2048</td></tr><tr><td>0x3</td><td>4096</td></tr><tr><td>0x4</td><td>8192</td></tr><tr><td>0x5</td><td>16384</td></tr><tr><td>0x6</td><td>32768</td></tr><tr><td>0x7</td><td>65536</td></tr></table>* - только для команд относительно страницы. Не относится к другим командам</td><td>0</td></tr>
<tr><td>[6:4]</td><td>Количество байтов (циклов) адреса в текущей команде</td><td>0</td></tr>
<tr><td>[8:7]</td><td><table><tr><td>Режим коррекции ошибок</td><td>Опция коррекции ошибок</td></tr><tr><td>0x0</td><td>Без коррекции ошибок/неприменимо для данной команды</td></tr><tr><td>0x1</td><td>Алгоритм коррекции: 4 ошибки на 512 байт</td></tr><tr><td>0x2</td><td>Алгоритм коррекции: 24 ошибки на 1024 байт</td></tr><tr><td>0x3</td><td>Резерв</td></tr></table></td><td>0</td></tr>
<tr><td>[9]</td><td>Номер разрешенного кристалла для текущей операции
0 – используется CE1n
1 – используется CE2n</td><td>0</td></tr>
<tr><td>[10]</td><td>Резерв</td><td>0</td></tr>
<tr><td>[11]</td><td>Аппаратная защита от записи (подлежит деактивации для команд записи/стирания)
0 – разрешена
1 – запрещена</td><td>0</td></tr>
<tr><td>[12]</td><td>Разрешение оценки кодов коррекции ошибок для данных в поле OOB (рекомендуется присвоить значение данному биту для обеспечения защиты всех данных страницы). Дополнительная информация приведена в разделе 0
0 – запрещена
1 – разрешена</td><td>0</td></tr>
<tr><td>[23:13]</td><td>Команды в отношении страницы – размер поля OOB
Для остальных команд – количество байтов данных, подлежащих считыванию</td><td>0</td></tr>
<tr><td>[31:24]</td><td>Для команд чтения идентификационного номера, страницы, команд программирования – первая команда, подлежащая отправке во флэш-память
К остальным командам не относится.</td><td>0</td></tr>
</table>

1.3.11.2.3.2.2.3 nand_command

Адрес: 0x08. Задание типа команды

Таблица 1.826 - Поля регистра nand_command

Биты	Описание	Значение по умолчанию
[7:0]	Цифровой код команды для контроллера NAND: 0x01 - считывание с NAND 0x03 – считывание идентификационного номера 0x04 – программирование NAND 0x06 – стирание блока 0x07 – сброс флэш-памяти NAND	0
[31:8]	Резерв	0

1.3.11.2.3.2.2.4 nand_col_addr

Адрес: 0x0C. Адрес столбца

Таблица 1.827 - Поля регистра nand_col_addr

Биты	Описание	Значение по умолчанию
[15:0]	Для команды считывания идентификационного номера – адрес для считывания идентификационного номера Для команд чтения и программирования – адрес столбца К остальным командам не относится.	0
[31:16]	Резерв	0

1.3.11.2.3.2.2.5 nand_row_addr_read_d

Адрес: 0x10. Адрес и данные строки

Таблица 1.828 - Поля регистра nand_row_addr_read_d

Биты	Описание	Значение по умолчанию
[23:0]	Для команд чтения, программирования и стирания блока – адрес троки + адрес блока.	0
[31:0]	Для команд чтения или программирования, используемых в операциях, не связанных со страницами (control_reg_i[31:24] != 0x80; control_reg_i[31:24] != 0x0), при объеме данных (control_reg_i[23:13]) менее или равном 4 байта значение этих данных подлежит считыванию или программированию. Необходимо помнить, что для данного типа операций размер адреса не должен превышать 2 байтов.	0

1.3.11.2.3.2.2.6 irq_mask_nand

Адрес: 0x14. Задание масок прерываний

Таблица 1.829 - Поля регистра irq_mask_nand

Биты	Описание	Значение по умолчанию
[0]	Mask irq_status[0] При присвоении значения данному биту генерируется прерывание по завершении выполнения команды чтения. Рекомендуемое значение: 0.	0
[1]	Mask irq_status[1] При присвоении значения данному биту генерируется прерывание по завершении выполнения команды считывания идентификационного номера. Рекомендуемое значение: 0.	0
[2]	Mask irq_status[2] При присвоении значения данному биту генерируется прерывание по завершении выполнения команды программирования. Рекомендуемое значение: 1.	0
[3]	Mask irq_status[3] При присвоении значения данному биту генерируется прерывание по завершении выполнения команды стирания блока. Рекомендуемое значение: 1.	0
[4]	Mask irq_status[4] При присвоении значения данному биту генерируется прерывание по завершении выполнения команды сброса флэш-памяти NAND. Рекомендуемое значение: 1.	0
[5]	Mask irq_status[5] При присвоении значения данному биту генерируется прерывание при возникновении некорректируемой ошибки. Рекомендуемое значение: 0.	0

Продолжение таблицы 1.704

Биты	Описание	Значение по умолчанию
[6]	Mask irq_status[6] При присвоении значения данному биту генерируется прерывание при возникновении ошибочной транзакции по интерфейсу записи AXI (неверное значение BRESP) Дополнительная информация приведена в разделе, посвященном регистру respw_status. Рекомендуемое значение: 1.	0
[7]	Mask irq_status[7] При присвоении значения данному биту генерируется прерывание по достижению последнего адреса ведущего интерфейса записи AXI. Рекомендуемое значение: 1.	0
[8]	Mask irq_status[8] При присвоении значения данному биту генерируется прерывание по достижению последнего сегмента адреса ведущего интерфейса записи AXI. Рекомендуемое значение: 0.	0
[9]	Mask irq_status[9] При присвоении значения данному биту генерируется прерывание при возникновении ошибочной транзакции по шине чтения AXI (неверное значение RRESP). Дополнительная информация приведена в разделе, посвященном регистру axi_if_err_addr. Рекомендуемое значение: 1.	
[10]	Mask irq_status[10] При присвоении значения данному биту генерируется прерывание по достижению последнего адреса ведущего интерфейса чтения AXI. Рекомендуемое значение: 0.	0
[11]	Mask irq_status[8] При присвоении значения данному биту генерируется прерывание по достижению последнего сегмента адреса ведущего интерфейса чтения AXI. Рекомендуемое значение: 0.	0
[31-12]	Резерв	0

1.3.11.2.3.2.2.7 cdb_buffer_enable

Адрес: 0x18. Разрешение чтения для ведущего интерфейса AXI

Таблица 1.830 - Поля регистра cdb_buffer_enable

Биты	Описание	Значение по умолчанию
[0]	Флаг разрешения режима ведущего интерфейса чтения AXI	0
[31:1]	Резерв	0

1.3.11.2.3.2.2.8 cdb_buffer_type

Адрес: 0x1C. Конфигурация буфера

Таблица 1.831 - Поля регистра cdb_buffer_type

Биты	Описание	Значение по умолчанию
[0]	Флаг возврата адреса 0 – при обработке последнего адреса происходит возврат к первому адресу 1 – при обработке последнего адреса производится остановка ведущего интерфейса чтения AXI	0
[31:1]	Резерв	0

1.3.11.2.3.2.2.9 cdb_ch0_buffer_status

Адрес: 0x20. Состояние буфера ведущего интерфейса чтения AXI

Таблица 1.832 - Поля регистра cdb_ch0_buffer_status

Биты	Описание	Значение по умолчанию
[7:0]	Количество слов в буфере ведущего интерфейса чтения AXI	0
[8]	Флаг запроса активных данных из ведущего интерфейса чтения AXI. Программный сброс ведущего интерфейса чтения AXI может быть произведен только при нулевом значении	0
[9]	Флаг заполнения буфера ведущего интерфейса чтения AXI	0
[10]	Флаг запроса данных из пустого буфера ведущего интерфейса чтения AXI	0
[11]	Флаг опустошения буфера ведущего интерфейса чтения AXI	1
[31:12]	Резерв	0

1.3.11.2.3.2.2.10 axi_arlen

Адрес: 0x24. Конфигурация ARLEN шины AXI

Таблица 1.833 - Поля регистра axi_arlen

Биты	Описание	Значение по умолчанию
[3:0]	Максимальное значение ARLEN для транзакций по ведущему интерфейсу чтения AXI. Рекомендуемое значение: 0xF.	0
[31:4]	Reserved	0

1.3.11.2.3.2.2.11 sw_rstn_r

Адрес: 0x28. Программный сброс ведущего интерфейса чтения AXI

Таблица 1.834 - Поля регистра sw_rstn_r

Биты	Описание	Значение по умолчанию
[0]	Установить значение в «1» для программного сброса ведущего интерфейса чтения AXI. Значение бита автоматически обнуляется по завершении операции. См. также регистры sw_rst и nand_cntrl_sw_rst	0
[31:1]	Резерв	0

1.3.11.2.3.2.2.12 axi_if_err_addr

Адрес: 0x30. Адрес ошибочной транзакции по шине AXI

Таблица 1.835 - Поля регистра axi_if_err_addr

Биты	Описание	Значение по умолчанию
[31:0]	Адрес транзакции с неверным значением RRESP	0

1.3.11.2.3.2.2.13 axi_if_status

Адрес: 0x34. Регистр состояния ведущего интерфейса чтения AXI

Таблица 1.836 - Поля регистра axi_if_status

Биты	Описание	Значение по умолчанию
[0]	Флаг активного режима транзакций по ведущему интерфейсу чтения AXI Программный сброс ведущего интерфейса чтения AXI может быть произведен только при нулевом значении	0
[4:1]	Число текущих активных транзакций по ведущему интерфейсу чтения AXI	0
[31:5]	Резерв	0

1.3.11.2.3.2.2.14 msb_lsbr

Адрес: 0x38. Конфигурация порядка следования байтов ведущего интерфейса чтения AXI

Таблица 1.837 - Поля регистра msb_lsbr

Биты	Описание	Значение по умолчанию
[0]	Порядок следования байтов ведущего интерфейса чтения AXI 0 - [СТАРШИЙ ЗНАЧАЩИЙ БИТ:МЛАДШИЙ ЗНАЧАЩИЙ БИТ] 1 - [МЛАДШИЙ ЗНАЧАЩИЙ БИТ:СТАРШИЙ ЗНАЧАЩИЙ БИТ]	0
[31:1]	Резерв	0

1.3.11.2.3.2.2.15 start_dma_r

Адрес: 0x40. Адрес первого байта данных, подлежащих считыванию с памяти

Таблица 1.838 - Поля регистра start_dma_r

Биты	Описание	Значение по умолчанию
[31:0]	Адрес первого байта данных, подлежащих считыванию с памяти	0

1.3.11.2.3.2.2.16 end_dma_r

Адрес: 0x44. Адрес последнего байта данных, подлежащих считыванию с памяти

Таблица 1.839 - Поля регистра end_dma_r

Биты	Описание	Значение по умолчанию
[31:0]	Адрес последнего байта данных, подлежащих считыванию с памяти	0

1.3.11.2.3.2.2.17 current_dma_r

Адрес: 0x48. Адрес текущих считываемых данных

Таблица 1.840 - Поля регистра current_dma_r

Биты	Описание	Значение по умолчанию
[31:0]	Адрес текущих считываемых данных. Перед чтением данного регистра необходимо задать регистр cntrl_dma_r как «cntrl_dma_r [30]»	0

1.3.11.2.3.2.2.18 cntrl_dma_r

Адрес: 0x4C. Регистр конфигурации для управления адресами для ведущего интерфейса чтения AXI

Таблица 1.841 - Поля регистра cntrl_dma_r

Биты	Описание	Значение по умолчанию
[4:0]	Должен быть всегда установлен в «0»	0
[28:5]	Размер сегмента (байты – 1) По достижении конца сегмента должно быть сгенерировано прерывание	0
[29]	Резерв	0
[30]	После задания значения данного бита может быть произведено чтение регистра current_dma_r	0
[31]	После задания значения начать генерировать адреса с начального по конечный	0

1.3.11.2.3.2.2.19 nand_cntrl_sw_rst

Адрес: 0x50. Программный сброс контроллера NAND

Таблица 1.842 - Поля регистра nand_cntrl_sw_rst

Биты	Описание	Значение по умолчанию
[0]	Запись «0x0» в данный регистр инициирует программный сброс контроллера NAND (за исключением ведущих интерфейсов AXI) См. также регистры sw_rstn_r и sw_rst	1
[31:1]	Резерв	0

1.3.11.2.3.2.2.20 nand_id

Адрес: 0x54. Регистр идентификации контроллера NAND

Таблица 1.843 - Поля регистра nand_id

Биты	Описание	Значение по умолчанию
[31:0]	Регистр идентификации контроллера NAND	444E414E

1.3.11.2.3.2.2.21 nand_version

Адрес: 0x58. Регистр версий контроллера NAND

Таблица 1.844 - Поля регистра nand_version

Биты	Описание	Значение по умолчанию
[31:0]	Регистр версий контроллера NAND	8B4

1.3.11.2.3.2.2.22 awlen_max

Адрес: 0x6C. Конфигурация AWLEN шины AXI

Таблица 1.845 - Поля регистра awlen_max

Биты	Описание	Значение по умолчанию
[3:0]	Максимальное значение AWLEN для транзакций по ведущему интерфейсу записи AXI. Рекомендуемое значение: 0xF.	0
[31:4]	Резерв	0

1.3.11.2.3.2.2.23 sw_rst

Адрес: 0x70 . Программный сброс ведущего интерфейса записи AXI

Таблица 1.846 - Поля регистра sw_rst

Биты	Описание	Значение по умолчанию
[0]	Запись «0x0» в данный регистр инициирует программный сброс ведущего интерфейса записи AXI После этого для продолжения нормальной работы необходимо записать «0x1» См. также регистры sw_rstn_r и nand_cntrl_sw_rst	1
[31:1]	Резерв	0

1.3.11.2.3.2.2.24 msb_lsbw

Адрес: 0x74. Конфигурация порядка следования байтов ведущего интерфейса записи AXI

Таблица 1.847 - Поля регистра msb_lsbw

Биты	Описание	Значение по умолчанию
[0]	Порядок следования байтов ведущего интерфейса записи AXI 0 - [СТАРШИЙ ЗНАЧАЩИЙ БИТ:МЛАДШИЙ ЗНАЧАЩИЙ БИТ] 1 - [МЛАДШИЙ ЗНАЧАЩИЙ БИТ:СТАРШИЙ ЗНАЧАЩИЙ БИТ]	0
[31:1]	Резерв	0

1.3.11.2.3.2.2.25 respw_status

Адрес: 0x78. Состояние ошибки транзакции по шине AXI

Таблица 1.848 - Поля регистра respw_status

Биты	Описание	Значение по умолчанию
[4:0]	Резерв (должен быть всегда установлен в «0»)	0
[6:5]	Значение BRESP	0
[31:7]	Резерв	0

1.3.11.2.3.2.2.26 status_full

Адрес: 0x7C. Регистр состояния внутреннего буфера ведущего интерфейса записи AXI

Таблица 1.849 - Поля регистра status_full

Биты	Описание	Значение по умолчанию
[0]	Буфер ведущего интерфейса записи AXI пуст	0
[1]	Буфер ведущего интерфейса записи AXI заполнен	0
[31:2]	Резерв	0

1.3.11.2.3.2.2.27 start_dma_w

Адрес: 0x88. Адрес первого байта данных, подлежащих записи в память

Таблица 1.850 - Поля регистра start_dma_w

Биты	Описание	Значение по умолчанию
[31:0]	Адрес первого байта данных, подлежащих записи в память	0

1.3.11.2.3.2.2.28 end_dma_w

Адрес: 0x8C. Адрес последнего байта данных, подлежащих записи в память

Таблица 1.851 - Поля регистра end_dma_w

Биты	Описание	Значение по умолчанию
[31:0]	Адрес последнего байта данных, подлежащих записи в память	0

1.3.11.2.3.2.2.29 current_dma_w

Адрес: 0x90.Адрес текущих записываемых данных

Таблица 1.852 - Поля регистра current_dma_w

Биты	Описание	Значение по умолчанию
[31:0]	Адрес первого байта данных, подлежащих считыванию с памяти	0

1.3.11.2.3.2.2.30 cntrl_dma_w

Адрес: 0x94. Регистр конфигурации для управления адресами для ведущего интерфейса записи AXI

Таблица 1.853 - Поля регистра cntrl_dma_w

Биты	Описание	Значение по умолчанию
[4:0]	Должен быть всегда установлен в «0»	0
[28:5]	Размер сегмента (байты – 1) По достижении конца сегмента должно быть сгенерировано прерывание	0
[29]	Резерв	0
[30]	После задания значения данного бита может быть произведено чтение регистра current_dma_w	0
[31]	После задания значения начать генерировать адреса с начального по конечный	0

1.3.11.2.3.2.2.31 irq_status

Адрес: 0xB4. Состояния контроллера

Таблица 1.854 - Поля регистра irq_status

Биты	Описание	Значение по умолчанию
[0]	Состояние завершения выполнения команды чтения	0
[1]	Состояние завершения выполнения команды считывания идентификационного номера	0
[2]	Состояние завершения выполнения команды программирования	0
[3]	Состояние завершения выполнения команды стирания блока	0
[4]	Состояние завершения выполнения команды сброса	0
[5]	Состояние возникновения некорректируемой ошибки	0
[6]	Состояние возникновения ошибки транзакции по ведущему интерфейсу записи AXI	0
[7]	Последний адрес состояния ведущего интерфейса записи AXI	0
[8]	Последний сегмент адреса состояния ведущего интерфейса записи AXI	0
[9]	Состояние возникновения ошибки транзакции по интерфейсу чтения AXI	
[10]	Последний адрес состояния ведущего интерфейса чтения AXI	0
[11]	Последний сегмент адреса состояния ведущего интерфейса чтения AXI	0
[12]	Состояние возникновения корректируемой ошибки	0
[13]	Состояние «читаемая страница пуста» 1 – страница не пуста 0 – страница пуста	0
[31-12]	Резерв	0

1.3.11.2.3.2.2.32 nand_timing_0

Адрес: 0xB8. Величины синхронизации для tADL, tALH, tALS, tCH

Таблица 1.855 - Поля регистра nand_timing_0

Биты	Описание	Значение по умолчанию
[7:0]	Значение tADL по системным часам clk_i	0
[15:8]	Значение tALH по системным часам clk_i	0
[23:16]	Значение tALS по системным часам clk_i	0
[31:24]	Значение tCH по системным часам clk_i	0

1.3.11.2.3.2.2.33 nand_timing_1

Адрес: 0xBC. Величины синхронизации для tCLH, tCLS, tCS, tDH

Таблица 1.856 - Поля регистра nand_timing_1

Биты	Описание	Значение по умолчанию
[7:0]	Значение tADL по системным часам clk_i	0
[15:8]	Значение tALH по системным часам clk_i	0
[23:16]	Значение tALS по системным часам clk_i	0
[31:24]	Значение tCH по системным часам clk_i	0

1.3.11.2.3.2.2.34 nand_timing_2

Адрес: 0xC0. Величины синхронизации для tDS, tWC, tWH, tWP

Таблица 1.857 - Поля регистра nand_timing_2

Биты	Описание	Значение по умолчанию
[7:0]	Значение tDS по системным часам clk_i	0
[15:8]	Значение tWC по системным часам clk_i	0
[23:16]	Значение tWH по системным часам clk_i	0
[31:24]	Значение tWP по системным часам clk_i	0

1.3.11.2.3.2.2.35 nand_timing_3

Адрес: 0xC4. Величины синхронизации для tWW, tAR, tCLR

Таблица 1.858 - Поля регистра nand_timing_3

Биты	Описание	Значение по умолчанию
[7:0]	Значение tWW по системным часам clk_i	0
[15:8]	Значение tAR по системным часам clk_i	0
[23:16]	Значение tCLR по системным часам clk_i	0
[31:24]	Резерв	0

1.3.11.2.3.2.2.36 nand_timing_4

Адрес: 0xC8. Величины синхронизации для tCOH, tIR, tRC, tREH

Таблица 1.859 - Поля регистра nand_timing_4

Биты	Описание	Значение по умолчанию
[7:0]	Значение tCOH по системным часам clk_i	0
[15:8]	Значение tIR по системным часам clk_i	0
[23:16]	Значение tRC по системным часам clk_i	0
[31:24]	Значение tREH по системным часам clk_i	0

1.3.11.2.3.2.2.37 nand_timing_5

Адрес: 0xCC. Величины синхронизации для tRHOH, tRHW, tRLOH, tRP

Таблица 1.860 - Поля регистра nand_timing_5

Биты	Описание	Значение по умолчанию
[7:0]	Значение tRHOH по системным часам clk_i	0
[15:8]	Значение tRHW по системным часам clk_i	0
[23:16]	Значение tRLOH по системным часам clk_i	0
[31:24]	Значение tRP по системным часам clk_i	0

1.3.11.2.3.2.2.38 nand_timing_6

Адрес: 0xD0. Величины синхронизации для tRR, tWHR, tCEA, tCHZ

Таблица 1.861 - Поля регистра nand_timing_6

Биты	Описание	Значение по умолчанию
[7:0]	Значение tRR по системным часам clk_i	0
[15:8]	Значение tWHR по системным часам clk_i	0
[23:16]	Значение tCEA по системным часам clk_i	0
[31:24]	Значение tCHZ по системным часам clk_i	0

1.3.11.2.3.2.2.39 nand_timing_7

Адрес: 0xD4. Величины синхронизации для tREA, tRHZ, tWB

Таблица 1.862 - Поля регистра nand_timing_7

Биты	Описание	Значение по умолчанию
[7:0]	Значение tREA по системным часам clk_i	0
[15:8]	Значение tRHZ по системным часам clk_i	0
[23:16]	Значение tWB по системным часам clk_i	0
[31:24]	Резерв	0

1.3.11.2.3.3 Основные операции

1.3.11.2.3.3.1 Введение

Данная глава приводит описание правильных последовательностей записей в регистры для выполнения основных операций с контроллером NAND. В данной главе приводится код Verilog и краткие описания, а также прочие примечания по вышеупомянутым операциям.

Используемые функции:

- запись (REG,VAL) – означает запись в регистр (REG) значения (VAL)
- чтение (REG, READ) – означает чтение регистра (или адреса) (REG) и сохранение результатов в READ
- ожидание прерывания (4) означает ожидание прерывания и проверку бита 4 регистра irq_status

1.3.11.2.3.3.2 Программный сброс контроллера NAND

Ниже представлена последовательность программного сброса.

```
write(nand_cntrl_sw_rst,0x0 );
write(sw_rst, 0x0);
write(sw_rst, 0x1);
write(sw_rstn_r, 0x1);
axi_read_sw_rst = 1
while (axi_read_sw_rst)
  read (sw_rstn_r, axi_read_sw_rst)
```

1.3.11.2.3.3.3 Конфигурация генерирования прерываний

Рекомендуемые значения масок для генерации запроса на прерывание (IRQ) приведены ниже

```
write(irq_mask_nand ,0x2DC);
```

1.3.11.2.3.3.4 Конфигурирование величин синхронизации

Необходимо задать величины синхронизации. Они зависят от тактовой частоты и могут быть оценены следующим образом:

```
NtXX = ceiling((tXX+1) * Fclk /1000)
```

где

- NtXX – значение, подлежащее записи в соответствующее поле регистра синхронизации
- tXX – величина синхронизации в наносекундах
- Fclk – тактовая частота в мегагерцах
- ceiling(VAL) – функция CEILING для преобразование значений VAL в двухбайтовые целые значения не меньше значений VAL

Ниже приведены некоторые примеры предварительной оценки.

Задание режима ONFI 0 с тактовой частотой F = 50 МГц

```
write(nand_timing_0,0x0203020B)
write(nand_timing_1,0x02040302)
write(nand_timing_2,0x03020603)
write(nand_timing_3,0x00020206)
write(nand_timing_4,0x02060101)
write(nand_timing_5,0x03010B01)
write(nand_timing_6,0x06060703)
write(nand_timing_7,0x000B0B03)
```

Задание режима ONFI 5 с тактовой частотой F = 100 МГц

```
write(nand_timing_0,0x01020108)
write(nand_timing_1,0x01020201)
write(nand_timing_2,0x02010301)
write(nand_timing_3,0x0002020B)
write(nand_timing_4,0x01030102)
write(nand_timing_5,0x02010B02)
write(nand_timing_6,0x04030903)
write(nand_timing_7,0x000B0B02)
```

1.3.11.2.3.3.5 Выполнение команды сброса (RESET)

```
write (irq_mask_nand, 0x10); //unmask reset command IRQ
write (nand_command, 0x7 ); //set nand reset command
write (nand_control, 0x1 ); //start nand reset command
 wait for interrupt (4)
read  (nand_status, flash_status);  //check nand interface status
  if (read_apb[7:0] != 8'h60)
     $display ("ERROR! Reset command: command fail");
```

1.3.11.2.3.3.6 Выполнение команды считывания идентификационного номера (READ ID)

1.3.11.2.3.3.6.1 Чтение 4 байтов идентификатора JEDEC

```
write (awlen_max, 0xf); //set write axi channel AWLEN=16
write (msb_lsbw, 0x1); //set write axi channel msb_lsb
write (start_dma_w, 0x100); //DMAW start address
write (end_dma_w, 0x200); //DMAW end address
write (cntrl_dma_w, 0x80000800); //DMAW set descriptor
write (irq_mask_nand, 0x2); //unmask read ID command IRQ
write (nand_command, 0x3); //set nand read ID command
write (nand_col_addr, 0x0); //set command address
write (nand_control, 0x90008011); //start nand read ID command
wait for interrupt (1);
read (0x100, read_axi); //read 4 bytes from 0x100 address
```

1.3.11.2.3.3.6.2 Чтение 4 байтов идентификатора ONFI

```
write (awlen_max, 0xf); //set write axi channel AWLEN=16
write (msb_lsbw, 0x1); //set write axi channel msb_lsb
write (start_dma_w, 0x100); //DMAW start address
write (end_dma_w, 0x200); //DMAW end address
write (cntrl_dma_w, 0x80000800); //DMAW set descriptor
write (irq_mask_nand, 0x2); //unmask read ID command IRQ
write (nand_command, 0x3); //set nand read ID command
write (nand_col_addr, 0x20); //set command address
write (nand_control, 0x90008011); //start nand 4 bytes read ID command (0x90)
wait for interrupt (1);
read (0x100, read_axi); //read 4 bytes from 0x100 address
```

1.3.11.2.3.3.7 Выполнение команды установки свойств (SET FEATURES)

Задание режима синхронизации 2 через команду SET FEATURES

```
read_apb = 0;
read (nand_status, read_apb);
while (read_apb[10:9] != 2'h3)  //Check Flash and Controller ready
  begin
    $display ("ERROR! NAND flash is not ready");
    $stop;
  end
write (irq_mask_nand, 0x4); //unmask write IRQ
write (nand_command, 0x4); //set nand write command
write (nand_col_addr, 0x1); //set Timing Mode feature address
write (nand_row_addr_read_d, 2);  //set Timing Mode 2
write (nand_control, 0xEF008811); //SET 0xEF as command,  4 bytes of data, disable
Write Protect, 1 byte of address, enable execution
wait for interrupt (2);
```

1.3.11.2.3.3.8 Выполнение команды получения свойств (GET FEATURES)

Проверка значений свойств режима синхронизации

```
read_apb = 0;
read (nand_status, read_apb);
while (read_apb[10:9] != 2'h3)  //Check Flash and Controller ready
  begin
    $display ("ERROR! NAND flash is not ready");
    $stop;
  end
write (irq_mask_nand, 0x4); //unmask write IRQ
write (nand_command, 0x1); //set nand read command
write (nand_col_addr, 0x1); //set Timing Mode feature address
write (nand_control, 0xEE008011); //SET 0xEE as command,  4 bytes of data, enable
Write Protect, 1 byte of address, enable execution
wait for interrupt (0);
read (nand_row_addr_read_d, read_apb);  //read Timing Mode feature value
```

1.3.11.2.3.3.9 Выполнение команды чтения страницы (READ PAGE)

Выполнение команды READ PAGE для 15-ой страницы (4 КБ каждая) с коррекцией ошибок с использованием алгоритма 24 ошибки/1024 байт

```
read_apb = 0;
read (nand_status, read_apb);
while (read_apb[10:9] != 2'h3)  //Check Flash and Controller ready
  begin
    $display ("ERROR! NAND flash is not ready");
    $stop;
  end
write (irq_mask_nand, 0x80); //unmask read AXI WRITE Master last address interrupt
write (awlen_max, 0xf); //set write axi channel AWLEN=16
write (msb_lsbw, 0x0); //set write axi channel endiannes
write (start_dma_w, 0x10000); //DMAW start address
write (end_dma_w, 0x1101F); //DMAW end address = start address+page size+OOB user data size1-1
write (cntrl_dma_w, 0x80000000); //DMAW set descriptor
write (nand_command, 0x1); //set nand read command
write (nand_col_addr, 0x0); //set NAND column address
write (nand_row_addr_read_d,0x270F); //set NAND row address
write (nand_control, 0x001C1957); //SET 0x00 as command,  224 bytes of OOB data2, Enable ECC for OOB, enable Write Protect,ECC type 2, 5 byte of address, 4KB page, enable execution
irq_handler (7);
```

1) – оценка и толкование размера пользовательских OOB-данных приведены в разделе 1.3.11.2.3.3.6.

2) – размер OOB-данных является параметром, используемым контроллером NAND флэш-памяти

1.3.11.2.3.3.10 Выполнение команды программирования страницы (PROGRAM PAGE)

Программирование первой страницы (2 КБ) первого блока второго адресата (CE2n) без коррекции ошибок

```
read_apb = 0;
read (nand_status, read_apb);
while (read_apb[10:9] != 2'h3)  //Check Flash and Controller ready
  begin
    $display ("ERROR! NAND flash is not ready");
    $stop;
  end
write (irq_mask_nand, 0x4); //unmask write NAND command interrupt
write (cdb_buffer_type, 0x1); //set read axi buffer type – finite (not circled)
write (axi_arlen, 0xf); //set read axi channel ARLEN=16
write (msb_lsbr, 0x0); //set READ axi channel endiannes
write (start_dma_r, 0x10000); //DMAR start address
write (end_dma_r, 0x1083F); //DMAR end address = start address+page size+OOB user data size[1]-1
write (cntrl_dma_r, 0x80000000); //DMAW set descriptor
write (nand_command, 0x4); //set NAND write command
write (nand_col_addr, 0x0); //set NAND column address
write (nand_row_addr_read_d,0x0); //set NAND row address
write (nand_control, 0x80080A55); //SET 0x80 as command,  40 bytes of OOB data[2], disable ECC for OOB, disable Write Protect, CE2n in use, ECC type 0, 5 byte of address, 2KB page, enable execution
irq_handler (2);
```

1) – оценка и толкование размера пользовательских OOB-данных приведены в разделе 1.3.11.2.3.3.6.

2) – размер OOB-данных является параметром, используемым контроллером NAND флэш-памяти

1.3.11.2.3.3.11 Выполнение команды стирания блока (ERASE BLOCK)

Стирание 1019-го блока

```
read_apb = 0;
read (nand_status, read_apb);
while (read_apb[10:9] != 2'h3)  //Check Flash and Controller ready
  begin
    $display ("ERROR! NAND flash is not ready");
    $stop;
  end
write (irq_mask_nand, 0x8); //unmask erase block IRQ
write (0x08, 0x6); //set erase block command
write (0x10, 0x3FB); //set block address
write (0x04, 0x001c0857); //1   40 bytes of OOB data2, disable ECC for OOB, disable
Write Protect, ECC type 0, 5 byte of address(3 ROW address), , enable execution
wait for interrupt (3)
```

1) – поля, которые не были упомянуты, не имеют значения, и при данной команде могут принимать любое значение

1.3.11.2.3.4 Примечания относительно кода коррекции ошибок

1.3.11.2.3.4.1 Расположение кода коррекции ошибок в памяти

Способ записи данных во флэш-память NAND строго фиксирован. Рисунок Рисунок 1.168 демонстрирует пример способа записи для страниц объемом 2 КБ с алгоритмом коррекции ошибок 0x2 и разрешенной коррекцией ошибок для OOB-данных. Здесь:

DATA FIELD X – пользовательские данные, подлежащие записи во флэш-память NAND

ECC DATA FIELD X – код коррекции ошибок для пользовательских данных, автоматически анализируемый и записываемый во флэш-память NAND

OOB DATA FIELD X – дополнительные данные, используемые флэш-памятью (хранение информации о поврежденных блоках, чистке памяти, нивелировании износа и т.д.)

ECC OOB DATA FIELD X – код коррекции ошибок для дополнительных данных, автоматически анализируемый и записываемый во флэш-память NAND

Рисунок Рисунок 1.168 демонстрирует предполагаемую конфигурацию данных в системной памяти во время выполнения команды записи в страницу и после чтения страницы. Предоставляются 4 поля пользовательских данных (2 КБ/512 Б) с пятью дополнительными 4-байтовыми полями данных. Требуется дополнительный буфер в системной памяти как для пользовательских данных, так и для OOB-данных. Примечание 1: Можно использовать функцию автоматической оценки кода коррекции ошибок для пользовательских данных, после чего все остальное пространство памяти будет доступно для дополнительных данных. Использование данной функции не обязательно.

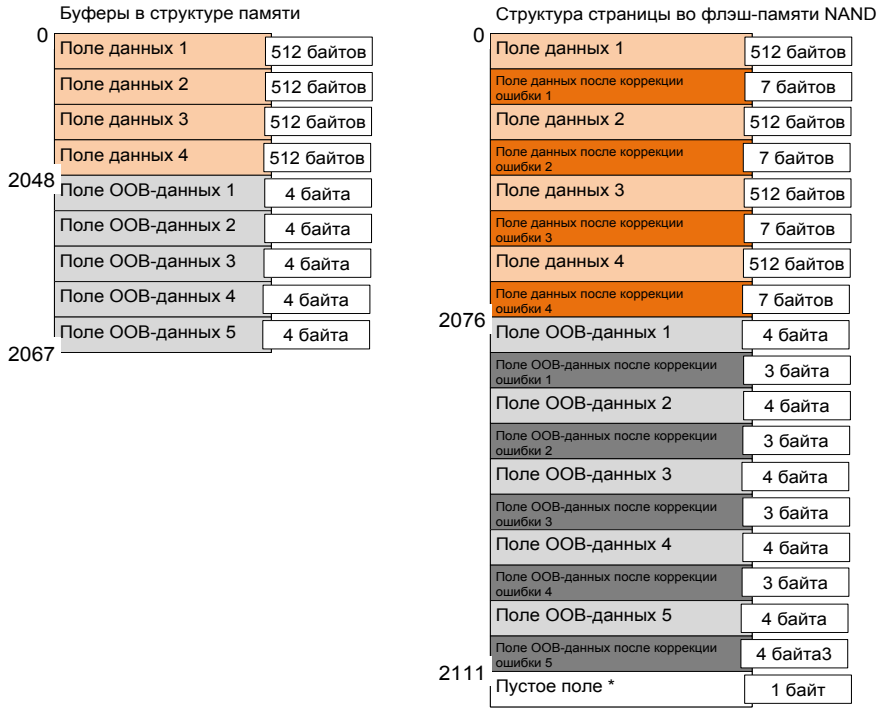


Рисунок 1.168 – Структура данных для системной памяти и флэш-памяти NAND

* – Необходимо помнить, что в данном случае (размер страницы и требования к коду коррекции ошибок) невозможно сохранить какие-либо данные в последнем байте страницы NAND.

1.3.11.2.3.4.2 Дополнительная память

Контроллер NAND использует три алгоритма коррекции ошибок, два из которых используются для поля пользовательских данных и один – для поля дополнительных данных с

дополнительной опцией, позволяющей деактивацию аппаратной коррекции ошибок пользовательских и дополнительных данных (см. раздел 1.3.11.2.3.2.2.2). Необходимо помнить, что запрещение коррекции ошибок пользовательских данных при разрешенной коррекции ошибок дополнительных данных не допускается. Функция коррекции ошибок контроллера NAND представляет собой аппаратную реализацию кодов БЧХ и является прозрачной для внешней системы. Тем не менее, она занимает место флэш-памяти. В таблице Таблица 1.863 представлена информация о пространстве памяти, необходимом для размещения обоих видов данных и кода коррекции ошибок.

Таблица 1.863 - Дополнительное пространство, используемое различными алгоритмами коррекции ошибок

Алгоритм коррекции ошибок	Размер данных, байты	Размер кода коррекции ошибок, байты
4 ошибки на 512 байтов	512	7
24 ошибки на 1024 байта	1024	42
Код коррекции ошибок для OOB-данных (4 ошибки/4 байта)	4	3

1.3.11.2.3.4.3 Конфигурация регистров для работы с кодом коррекции ошибок

Контроллер имеет ряд регистров, зависящих от использования кода коррекции ошибок. Значения регистров end_dma_w и end_dma_r могут быть оценены следующим образом:

$$ADDRend = ADDRstart + NDuser + NDoob$$

где

- ADDRend – значение, подлежащее записи в регистр конечного адреса
- ADDRstart – значение, подлежащее записи в регистры начального адреса (адреса первого байта буфера системной памяти)
- NDuser – размер страницы флэш-памяти без памяти OOB-данных, обычно пользовательских данных
- NDoob – количество байтов дополнительных данных, подлежащих записи после пользовательских данных, которое должно быть кратным 4 и удовлетворять следующему условию:

$$NDoob \leq ((Npage - NDuser\_ecc) \setminus 7) * 4$$

где

- N_page – размер страницы флэш-памяти NAND из спецификации
- NDuser_ecc – из таблицы
- Таблица 1.864 занимаемое пространство с пользовательскими данными и кодом коррекции ошибок
- \ – оператор целочисленного деления
- * – оператор умножения
- Регистр nand_control
- Задание значений регистра nand_control [7:8] необходимо производить в соответствии с конкретными требованиями к флэш-памяти NAND.
- Если необходимо обеспечить защиту всех данных страницы, значение регистра nand_control [12] должно быть установлено в «1». Если для поля OOB-данных требуется большее пространство, необходимо обнулить значение регистра nand_control [12] (перед этим убедитесь, что Вам действительно нужна данная опция).

nand_control [23:13] – Значение Noob задается следующим образом:

$$Noob = NDuser\_ecc - NDuser + NDoob * 7 \setminus 4$$

Таблица 1.864 – Пространство, используемое для данных и кода коррекции ошибок на страницах различного размера

Используемый алгоритм коррекции ошибок	Используемые байты		
	Страница 2 КБ	Страница 4 КБ	Страница 8 КБ
Без коррекции ошибок	2048	4096	8192
4 ошибки на 512 байтов	2076	4152	8308
24 ошибки на 1024 байта	2132	4264	8528

1.3.11.2.4 Контроллер SRAM/NOR-памяти с интерфейсом AXI (в LSIF0)

Блок контроллера SRAM/NOR-памяти предназначен для подключения к СнК до 6 внешних банков SRAM-памяти или NOR-памяти с шинами адреса шириной от 10 до 26 разрядов, шириной шины данных 32 разряда и шириной шины проверочных данных 8 разрядов.

Структурная схема блока контроллера SRAM/NOR-памяти с примером подключения к нему двух банков SRAM-памяти приведена на рисунке ниже.

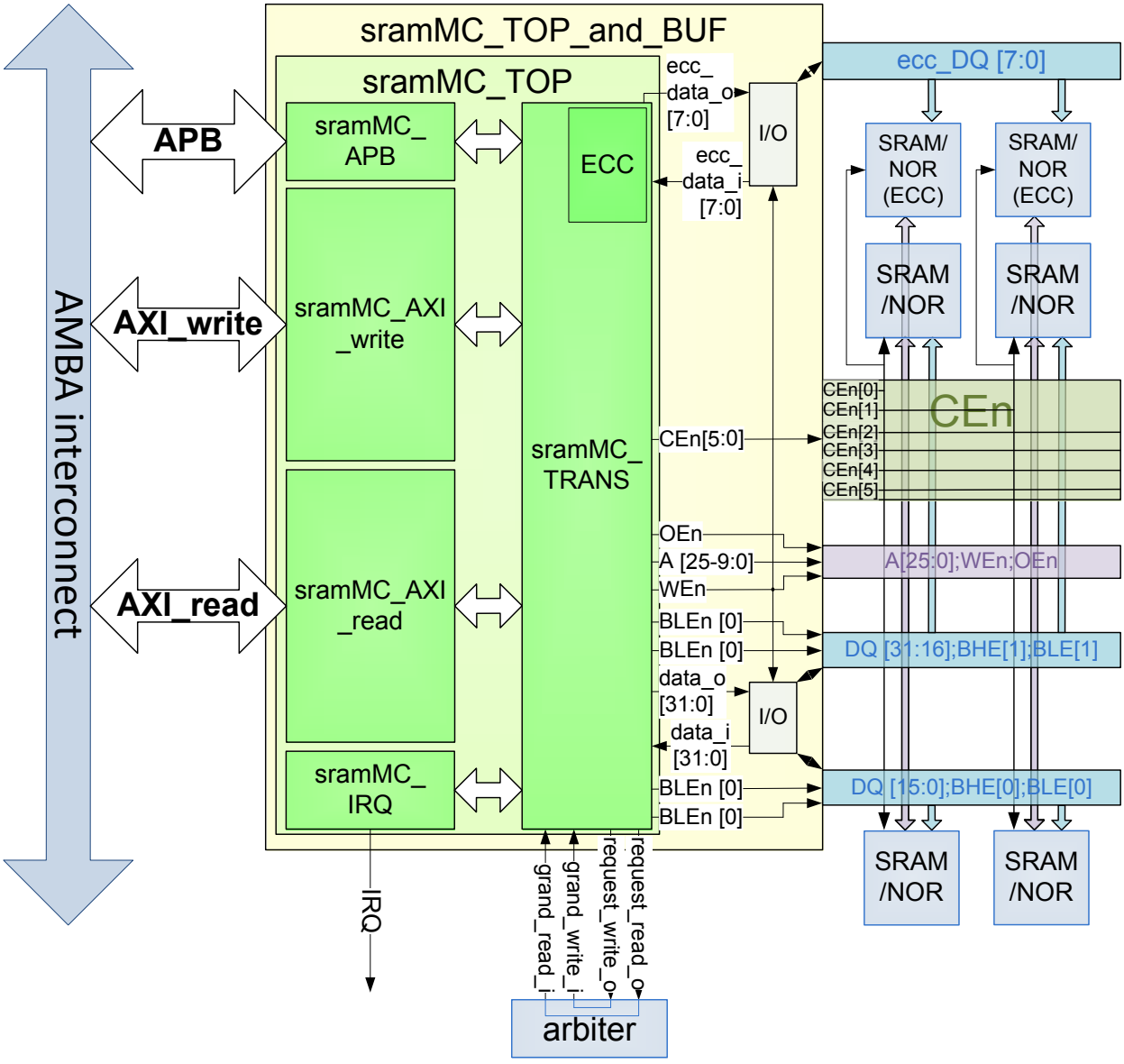


Рисунок 1.169 - Структурная схема блока контроллера SRAM/NOR-памяти

Каждый банк памяти управляется своим сигналом CEn[i] для перевода памяти из режима хранения в режим чтения/записи. Для SRAM-памяти сигналы BHEn[1:0] и BLEn[1:0] позволяют записывать только нужные байты из 4 байтового слова, для NOR-памяти они не используются.

Для возможности совместного использования нескольких контроллеров с общими шинами сигналов к памяти блок имеет интерфейс для организации арбитра.

1.3.11.2.4.1 Описание назначения и функций блоков

Контроллер SRAM/NOR-памяти (блок sramMC_TOP) состоит из блоков sramMC_AXI_write, sramMC_AXI_read, sramMC_APB, sramMC_IRQ и sramMC_TRANS.

1.3.11.2.4.1.1 Блоки sramMC_AXI_write и sramMC_AXI_read

Блоки sramMC_AXI_write и sramMC_AXI_read реализуют 32-разрядный системный интерфейс AXI_slave для подключения к коммутационной среде СнК и отвечют следующим требованиям:

- Список и назначение сигналов системного интерфейса AXI-slave соответствуют спецификации AMBA AXI Protocol v 1.0.
- Системный интерфейс AXI-slave поддерживает следующие функциональные свойства протокола AXI:
  - Пакетные транзакции (Burst transfer) типа INCR и WRAP.
  - Чтение/запись данных с разрядностью 8, 16 и 32 разряда (Narrow transfer).
- Системный интерфейс AXI-slave не поддерживает следующие функциональные свойства протокола AXI:
  - Чередования данных от транзакций записи с разными ID (write interleaving).
  - Эксклюзивный доступ к ячейки памяти (Exclusive access).

Блок sramMC_AXI_write реализует часть протокола AXI необходимую для запись данных во внешнюю память (через транслятор).

Блок sramMC_AXI_read реализует часть протокола AXI необходимую для чтения данных из внешней памяти (через транслятор).

1.3.11.2.4.1.2 Блок sramMC_APB

Блок sramMC_APB предназначен для программирования регистров контроллера.

Список и назначение сигналов системного интерфейса APB-slave соответствуют спецификации AMBA3 APB.

1.3.11.2.4.1.3 Блок sramMC_IRQ

Блок sramMC_IRQ генерирует уровневые прерывание, для сброса прерывания используется регистр STATUS_IRQ чтение которого осуществляется по APB-интерфейсу.

1.3.11.2.4.1.4 Блок sramMC_TRANS

Блок sramMC_TRANS контроллера (транслятор) реализует преобразование AXI-транзакции в класс временных диаграмм интерфейса для подключения микросхем SRAM/NOR-памяти, а также реализует работу с кодами обнаружения и коррекции ошибок (ECC).

1.3.11.2.4.1.5 ECC

Контроллер SRAM и NOR-памяти поддерживает возможность записи и чтения не только данных, но и проверочных данных. Проверочные данные можно передавать параллельно с основными данными. Для этого интерфейс подключения микросхем SRAM- и NOR-памяти к совмещенному контроллеру SRAM- и NOR-памяти имеет дополнительные выводы для передачи проверочных данных.

Для возможности исправить все однократные и одновременно обнаружить все двукратные ошибки, используется расширенный код Хэмминга. При кодировании 32 бит данных требуется 7 проверочных бит.

Для SRAM поддерживается возможность записи в режиме с ECC не полных слов за счет реализации режима read-modify-write, когда для сохранения корректного кода ECC сначала прочитывается 32-битное слово данных, хранящееся по нужному адресу, из него берутся те байты которые надо оставить без изменений, добавляются байты для записи, считается ECC код слова, и в память записываются данные и ECC-код.

Для корректной записи не полных слов в режиме с ECC необходимо чтобы перед началом работы вся используемая память была проинициализирована.

Чтение не полных слов в режиме с ECC поддерживается за счет того что из памяти вычитывается полное 32-х разрядное слово с проверочными битами, а далее уже AXI-slave выбирает нужные байты для передачи.

Для взаимодействия контроллера (с поддержкой кодов коррекции ошибок) и СнК в случаях возникновения ошибок необходимо генерировать прерывания.

1.3.11.2.4.2 Регистровая карта

Контроллер SRAM/NOR-памяти имеет набор регистров (табл. Таблица 1.865), программируемых по APB интерфейсу, позволяющих задавать конфигурации и параметры временных диаграмм интерфейса для подключения различного количества и различных моделей микросхем SRAM- или NOR-памяти.

Таблица 1.865 Регистровая карта

Адрес	Название	Значение после сброса	Доступ	Описание
0x00	ID	Определяется при сборке системы	RO	Идентификатор устройства
0x04	VERSION	0x00000101*	RO	Версия устройства
0x08	CONFIG	0x0116000F*	RW	Регистр настройки конфигурации
0x0C	TIMINGS	0x06060100*	RW	Регистр настройки временных параметров
0x10	MASK_IRQ	0x00000002	RW	Маска прерываний
0x14	STATUS_IRQ	0x0	RO	Регистр статуса прерываний, чтением данного регистра осуществляется сброс прерывания и значения данного регистра
0x18	ECC_ERR_ADDR	0x0	RO	Регистр адреса слова в котором была обнаружена ошибка
0x1c	-	-	-	reserve
0x20	MANAGEMENT_ECC	0x0	RW	Регистр управления работой с памятью для ECC
0x24	DATA_ECC_WRITE_MEM	0x0	RW	Регистр данных для передачи по шине ECC_data[7:0] при записи в память
0x28	DATA_ECC_READ_MEM	0x0	RO	Регистр данных принимаемых по шине ECC_data[7:0] при чтении из память

* значения могут уточняться при сборке системы

1.3.11.2.4.2.1 CONFIG

Таблица 1.866 - CONFIG

Биты	Поле	Описание
5:0	chip_num	Активация подключенных микросхем памяти 0 – не активно, 1 – активно.
7:6	-	reserve
8	ce_mode	Режим активации подключенных микросхем 0 – CEn устанавливается в соответствии со старшими разрядами адреса 1 – CEn устанавливается в соответствии со значением поля chip_num
15:9	-	reserve
20:16	addr_size	Абсолютное значение ширины шины адреса (поддерживаемые значения от 10 до 26)
23:21	-	reserve
24	ECC_ACTIVE	0 – коррекцию ошибок отключена 1 – коррекцию ошибок включена
31:25	-	reserve

1.3.11.2.4.2.2 TIMINGS

Таблица 1.867 - TIMINGS

Биты	Поле	Описание
7:0	TRG	Минимальная длительность неактивной фазы между циклами чтения
15:8	TWG	Минимальная длительность неактивной фазы между циклами записи
23:16	TRC	Длительность цикла чтения
31:24	TWC	Длительность цикла записи

1.3.11.2.4.2.3 MASK_IRQ

Таблица 1.868 - MASK_IRQ

Биты	Поле	Описание
0	MASK_S_ERR	При присвоении лог. 1 данному биту генерируется прерывание при обнаружении однократной ошибки. Рекомендуемое значение: 0.
1	MASK_D_ERR	При присвоении лог. 1 данному биту генерируется прерывание при обнаружении двукратной ошибки. Рекомендуемое значение: 1.
31:2	-	reserve

1.3.11.2.4.2.4 STATUS_IRQ

Таблица 1.869 - STATUS_IRQ

Биты	Поле	Описание
0	ECC_S_ERR	Лог. 1 если в слове вызвавшем прерывание была однократная ошибка
1	ECC_D_ERR	Лог. 1 если в слове вызвавшем прерывание была двукратная ошибка.
3:2	-	reserve
14:4	COUNT_S_ERR	Значение счетчика соответствует числу однократных ошибок произошедших до чтения регистра
15	OVERFLOW_S_ERR	Переполнение счетчика однократных ошибок, выставляется в лог. 1 если число однократных ошибок превышает 2047
26:16	COUNT_D_ERR	Значение счетчика соответствует числу двукратных ошибок произошедших до чтения регистра
27	OVERFLOW_D_ERR	Переполнение счетчика двукратных ошибок, выставляется в лог. 1 если число двукратных ошибок превышает 2047
31:28	-	reserve

1.3.11.2.4.2.5 ECC_ERR_ADDR

Таблица 1.870 - ECC_ERR_ADDR

Биты	Поле	Описание
31:0	ECC_ERR_ADDR	Адрес слова, ошибка в котором вызвала прерывание.

1.3.11.2.4.2.6 MANAGEMENT_ECC

Таблица 1.871 MANAGEMENT_ECC

Биты	Поле	Описание
1:0	MODE_ECC_DATA	0 – передача по шине ECC_data[7:0], управляющих сигналов (таких же как на шине data[7:0]), для NUMBER_COMAND слов данных, далее передача кода обнаружение и коррекции ошибок (ECC_data[7] – не используется), при этом счетчик сбрасывается в изначально записанное значение; чтение в данном режиме с ECC (также коды прочитанные из ECC-памяти записываются в регистр DATA_ECC_READ_MEM); при записи данных в SRAM NUMBER_COMAND должен быть равен 0. 1 – передача по шине ECC_data[7:0], управляющих сигналов из регистра DATA_ECC_WRITE_MEM при записи управляющих команд в NOR-память; чтение в данном режиме без ECC, данные прочитанные из ECC-памяти записываются в регистр DATA_ECC_READ_MEM. 2 – передача по шине ECC_data[7:0], управляющих сигналов (таких же как на шине data[7:0]) при записи управляющих команд в NOR-память; чтение в данном режиме без ECC, данные прочитанные из ECC-памяти записываются в регистр DATA_ECC_READ_MEM. 3 – reservd
3:2	-	reserve
7:4	NUMBER_COMMAND	Число управляющих сигналов до передачи в память кодов обнаружение и коррекции ошибок
31:8	-	reserve

1.3.11.2.4.2.7 DATA_ECC_WRITE_MEM

Таблица 1.872 - DATA_ECC_WRITE_MEM

Биты	Поле	Описание
7:0	DATA_ECC_WRITE_MEM	Данные для передачи по шине ECC_data[7:0] при записи в память
31:8	-	reserve

1.3.11.2.4.2.8 DATA_ECC_READ_MEM

Таблица 1.873 - DATA_ECC_READ_MEM

Биты	Поле	Описание
7:0	DATA_ECC_READ_MEM	Данные принимаемые по шине ECC_data[7:0] при чтении из память
31:8	-	reserve

1.3.11.2.4.3 Адресация SRAM/NOR-памяти

К контроллеру SRAM/NOR-памяти можно подключить до 6 внешних банков SRAM/NOR-памяти с шинами адреса шириной до 26 разрядов и шириной шины данных 32 разряда. К каждому банку подведены общие шина адреса (A), шина данных (I/O[31:0]), управляющие сигналы (CEn,OEn,WEn) и сигналы значимости битов ячеек памяти (BHEn[1:0],BLEn[1:0]; используются только для SRAM-памяти). Описание сигналов значимости битов ячеек памяти приведено в таблице ниже.

Таблица 1.874 - Описание сигналов значимости данных

Сигнал	Значимые биты данных
BLEn[0]=1	7:0
BHEn[0]=1	15:8
BLEn[1]=1	23:16
BHEn[1]=1	31:24

Схема примера подключения двух банков памяти, каждый из которых состоит из двух микросхем SRAM-памяти с шириной шины данных 16 разрядов, приведена на рисунке Рисунок 1.169.

Каждому адресу в банке SRAM/NOR-памяти соответствует 32 бита данных. При обращении к SRAM/NOR-памяти по AXI интерфейсу адресация производится побайтно, описание представлено в таблице ниже.

Таблица 1.875 - Описание адресации памяти

Биты адреса поступающего на AXI при ширине шины адреса равной значению поля addr_size регистра CONFIG	Значение поля ce_mode регистра CONFIG	Описание	Сигналы к SRAM/NOR-памяти
1:0	x	адрес байта в слове	BHEn[1:0],BLEn[1:0]*
(size_addr_m_r+1):2	x	адрес слова в банке SRAM/NOR-памяти	A[size_addr_m_r-1:0]
(size_addr_m_r+4): (size_addr_m_r+2)	0	адрес банка SRAM/NOR-памяти (описание ниже)	CEn[5:0]

x – любое значение

*Шины BLEn и BHEn к NOR-памяти не подключены.

Старшие биты адреса контроллером не проверяются.

В случае если в поле ce_mode регистра CONFIG записан 0 поле chip_num преобразуется в зависимости от адреса в сигналы CEn для микросхем памяти в соответствии с таблицей 1.751.

Таблица 1.876 Описание выбора пары микросхем SRAM/NOR-памяти при ce_mode=0

Подключение микросхем SRAM-памяти	Значения трех старших используемых битов шины адреса AXI AxADDR[(size_addr_m_r+4):(size_addr_m_r+2)]	Доступ к микросхемам SRAM-памяти
chip_num [0]=1	3'b000	CEn[0]=0
chip_num [1]=1	3'b001	CEn[1]=0
chip_num [2]=1	3'b010	CEn[2]=0
chip_num [3]=1	3'b011	CEn[3]=0
chip_num [4]=1	3'b100	CEn[4]=0
chip_num [5]=1	3'b101	CEn[5]=0

В случае если в поле ce_mode регистра CONFIG записана 1 на шину CEn[5:0] контроллера передаётся инвертированное значение поля chip_num[5:0]. Для корректной работы контроллера не допускается одновременное подключение нескольких микросхем памяти в этом режиме (поле chip_num[5:0] должно содержать только одну 1).

1.3.11.2.4.4 Описание работы контроллера

1.3.11.2.4.4.1 Состояние после сброса

Каналы записи и чтения sramMC_AXI находятся в состоянии ожидания прихода транзакций.

sramMC_APB находится в состоянии ожидания прихода транзакций на запись или чтение регистров.

sramMC_IRQ устанавливает выход внешнего прерывания в не активное состояние.

Значения внутренних программируемых регистров в трансляторе соответствует значениям таблицы 2.

Транслятор свободен для программирования или обращений на чтение/запись данных.

1.3.11.2.4.4.2 Состояние ожидания

Блок контроллера памяти находится в состоянии ожидания если:

- по шине APB не приходят сигналы на запись регистров;
- по шине AXI не приходят сигналы на запись или чтение данных;
- отработаны все предыдущие обращения;
- внешнее прерывание выставлено в не активное состояние.

Когда блок контроллера памяти находится в состояние ожидания, на его выходах будут выставлены следующие сигналы:

- к SRAM/NOR-памяти:
  - Все микросхемы памяти находятся в режиме хранения (CEn = 1);
- sramMC_APB:
  - Готов принять транзакцию (PREADY_o = 1);
- sramMC_AXI_write:
  - Готов принять адрес (AWREADY_o = 1) транзакции записи;
- sramMC_AXI_read:
  - Готов принять адрес (ARREADY_o = 1) транзакции чтения;
- sramMC_IRQ:
  - Прерывание в не активном состоянии (IRQ = 0).

1.3.11.2.4.4.3 Перепрограммирование внутренних регистров

Для программирования внутренних регистров контроллера используется интерфейс APB. При поступлении транзакции записи по шине APB во внутренние регистры APB записываются поступающие данные. Когда транслятор находится в не занятом состоянии (нет запросов на работу с памятью), то значения данных регистров APB переписываются на регистры транслятора. Когда транслятор занят (ожидает разрешения арбитра на работу с памятью или производит запись(чтение) данных), то значения данных регистров APB переписываются на регистры транслятора только когда транслятор освободится.

При поступлении транзакции чтения в sramMC_APB производится чтение внутренних регистров APB. Это может осуществляться в любой момент времени.

1.3.11.2.4.4.4 Чтение данных из памяти

При захвате в блок sramMC_AXI_read транзакции чтения из памяти, блок sramMC_AXI_read выставляет на выход к транслятору sramMC_TRANS адрес читаемой ячейки и выдает сигнал о захвате адреса для чтения и далее ожидает от транслятора сигнала о готовности прочитанных данных.

Когда в транслятор sramMC_TRANS поступает сигнал о захвате адреса для чтения, если транслятор свободен, то начинается обработка запроса на чтение. Если транслятор занят, то обработка запроса на чтение начинается после завершения обработки ранее пришедшего запроса на запись.

Обработка запроса на чтение:

- транслятор делает запрос (request_read_o=1) к ARB_SRAM/NOR.

Когда арбитр дает доступ к памяти (grant_read_i=1), транслятор начинает чтение, выставляя следующие сигналы на выходы к памяти:

- WEn = 1;
- BLEn = 2'b00 (BLEn переключается в неактивное состояние (BLEn = 2'b11) при не активном CEn);*
- BHEn = 2'b00 (BHEn переключается в неактивное состояние (BHEn = 2'b11) при не активном CEn);*
- A = addr_read_w[27:2];
- На шине CEn в ноль устанавливается только один из шести битов соответствующий нужному банку памяти (описание в разделе адресация SRAM/NOR-памяти);
- Сигналы OEn имеет значение противоположное WEn (OEn переключается в неактивное состояние (OEn =1) при не активном CEn).

*Шины BLEn и BHEn к NOR-памяти не подключены.

Как только сигналы на выходах выставлены они удерживаются в течение времени прописанного в поле TRC регистра TIMING, после чего данные поступающие от памяти (в том числе и проверочные данные, если включен режим поддержки кодов коррекции ошибок) считаются истинными и защелкиваются на внутренний регистр блока sramMC_TRANS и транслятор выдает для блока чтения sramMC_AXI_read сигнал о готовности данных от памяти (mvalid_read_w=1, сигнал держится в 1 до того как sramMC_AXI_read не прочитает данные), если включен режим поддержки кодов коррекции ошибок, то блок sramMC_TRANS проверяет данные на наличие ошибок и при обнаружении ошибок передает об этом данные контроллеру прерываний (sramMC_IRQ), а также при возможности исправления передает в sramMC_AXI_read уже исправленные данные. В следующем такте блок чтения sramMC_AXI_read выставляет данные на выход RDATA_o и выдает сигнал об их готовности RVALID_o.

В случае если это последнее слово данных пакетной транзакции, транслятор переводит сигнал request_read_o к арбитру ARB_SRAM/NOR в неактивное состояние (request_read_o=0) и приступает к обработке следующих обращений или переходит в режим ожидания следующего обращения.

В случае если это не последние данные пакетной транзакции и минимальная длительность неактивной фазы между циклами чтения равна 0, то транслятор, захватывает адрес следующей читаемой ячейки и повторяется чтение.

В случае если минимальная длительность неактивной фазы между циклами чтения не равна 0, то транслятор, одновременно с выдачей данных и сигнала об их готовности, переводит шину CEn в не активное состояние и удерживает это состояние в течение времени прописанного в поле TRG регистра TIMING, по прошествии этого времени транслятор начинает чтение из следующего адреса памяти.

Если по адресу, пришедшему от sramMC_AXI_read, отсутствует банк памяти, то транслятор sramMC_TRANS возвращает блоку sramMC_AXI_read сигнал об ошибке.

Временная диаграмма примера цикла чтения пакетной транзакции состоящей из 2 слов представлена на рисунке ниже.

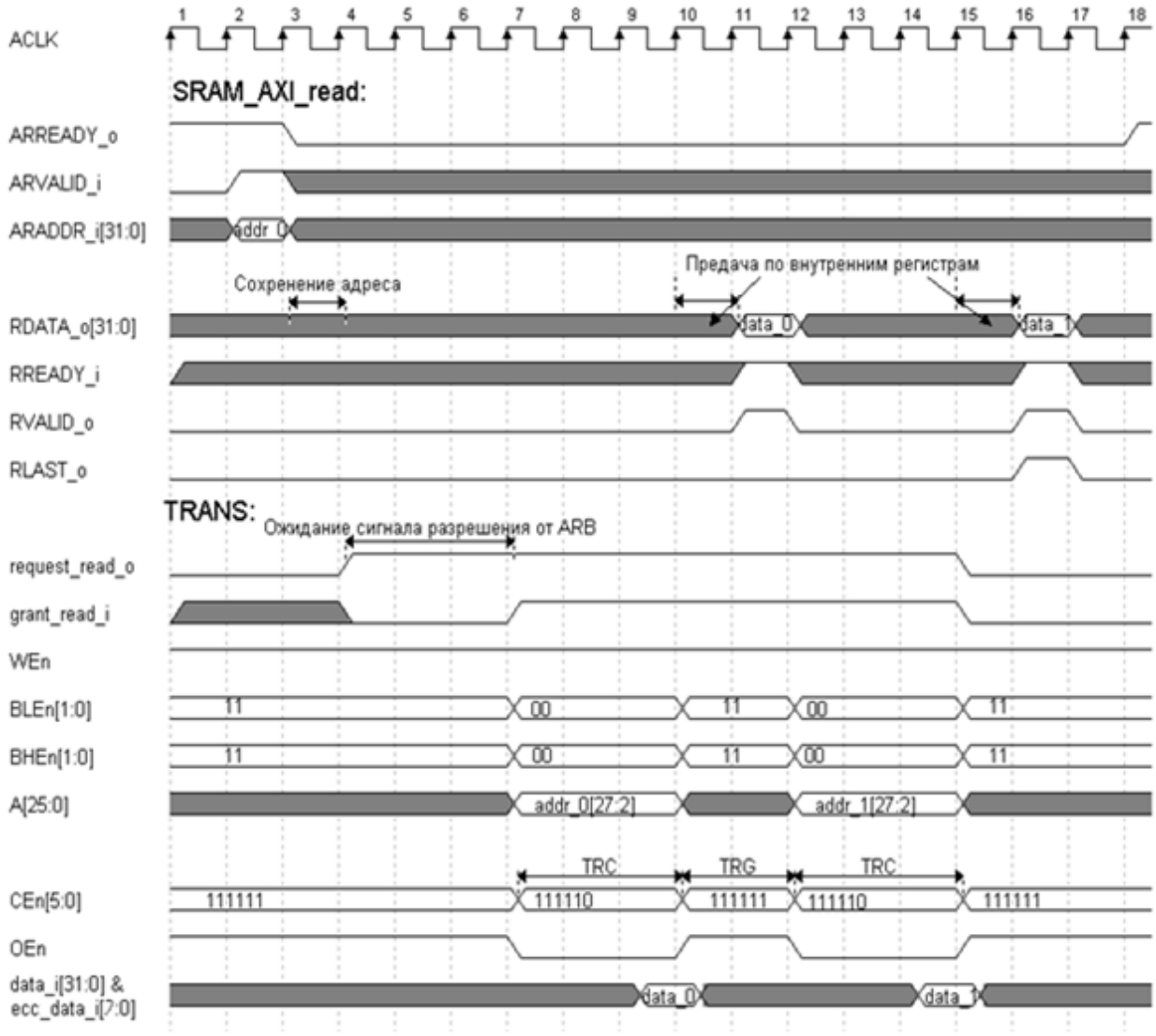


Рисунок 1.170 - Временная диаграмма цикла чтения при постоянном доступе к SRAM/NOR-памяти и при TRG=0 (шины BLEn и BHEn к NOR-памяти не подключены)

Длительность цикла чтения: TRC = TACLK*TRC(TIMING) ;
TACLK – период тактового сигнала;
Длительность неактивной фазы между циклами чтения: TRG = TACLK*TRG(TIMING) ;
TWC(TIMING), TWG(TIMING), - значения соответствующих полей регистра TIMING.

1.3.11.2.4.4.5 Запись данных в память

При поступлении в sramMC_AXI_write адреса транзакции записи, sramMC_AXI_write AXI выставляет сигнал о готовности принять данные (WREADY_o=1). Когда от AXI приходят данные для записи в память, они сохраняются на регистр и sramMC_AXI_write выдает:

- в sramMC_TRANS строб записи (strb_write_w!=0)
- на шину AXI сигнал о готовности принять следующие данные (WREADY_o=1) или, если это последние данные транзакции, ждет сигнал о готовности ответа от sramMC_TRANS (resp_mem_write_w=1).

При поступлении строба записи sramMC_TRANS сохраняет во внутренний регистр: значение строба записи, адрес, данные для записи и переводится в неактивное состояние сигнал о готовности принять от sramMC_AXI_write данные на запись (mready_write_w=0).

Если при поступлении строба записи транслятор свободен и не происходит одновременного обращения на чтение данных, то начинается обработка запроса на запись. Если, блок SRAM_TRANS занят или происходит одновременное обращения на чтение данных, то обработка запроса на запись начинается после завершения обработки ранее (или одновременно) пришедшего запроса на чтение.

Обработка запроса на запись:

- транслятор делает запрос (request_write_o=1) к арбитру ARB_SRAM/NOR.

Когда арбитр дает доступ к памяти (grant_write_i=1), транслятор начинает запись, выставляя следующие сигналы на выходы к памяти:

- WEn = 0;
- BLEn[0] = ~strb_r[0]*;
- BLEn[1] = ~strb_r[2]*;
- BHEn[0] = ~strb_r[1]*;
- BHEn[1] = ~strb_r[3]*;
- data_o = data_write_r;
- ecc_data_o
  - равняется 0 если ECC_ACTIVE=0;
  - рассчитывается по расширенному коду Хемминга если ECC_ACTIVE=1, MODE_ECC_DATA=0 и NUMBER_COMMAND=0;
  - ecc_data_o[7:0] = data_o[7:0](используется для программирования NOR-памяти) если ECC_ACTIVE=1,MODE_ECC_DATA=0 и NUMBER_COMMAND!=0, или MODE_ECC_DATA=2;
  - ecc_data_o[7:0] = DATA_ECC_WRITE_MEM[7:0] (используется для программирования NOR-памяти или для внесения ошибок в ECC память для тестирования) если ECC_ACTIVE=1, MODE_ECC_DATA=1;
- A = addr_write_r[27:2];
- На шине CEn в ноль устанавливается только один из шести битов соответствующий нужному банку памяти (описание в разделе адресация SRAM/NOR-памяти);
- Сигналы OEn имеет значение противоположное WEn (OEn переключается в неактивное состояние (OEn =1) при не активном CEn).

*BLEn, BHEn переключается в неактивное состояние (BLEn = 2'b11, BHEn = 2'b11) при не активном CEn, к NOR-памяти данные шины не подключены.

Как только сигналы на выходах выставлены они удерживаются в течение времени прописанного в поле TWC регистра TIMING, после чего данные поступающие в память считаются записанными, и транслятор готов для записи следующих данных.

Если записаны не последние данные пакетной транзакции, то блок sramMC_TRANS ждет от sramMC_AXI_write следующего строба записи.

По окончании записи последних данных пакетной транзакции транслятор переводит сигнал request_write_o к арбитру ARB_SRAM/NOR в неактивное состояние (request_write_o=0) и приступает к обработке следующих обращений или переходит в режим ожидания следующего обращения.

Если по адресу, пришедшему от sramMC_AXI_write, отсутствует банк памяти, то транслятор sramMC_TRANS возвращает блоку sramMC_AXI_write сигнал об ошибке.

Временная диаграмма примера цикла записи пакетной транзакции состоящей из 2 слов представлена на рисунке ниже, для того чтобы диаграмма имела такой вид должна быть или отключена коррекция ошибок или слова должны быть полными, иначе будет работать механизм read-modify-write.

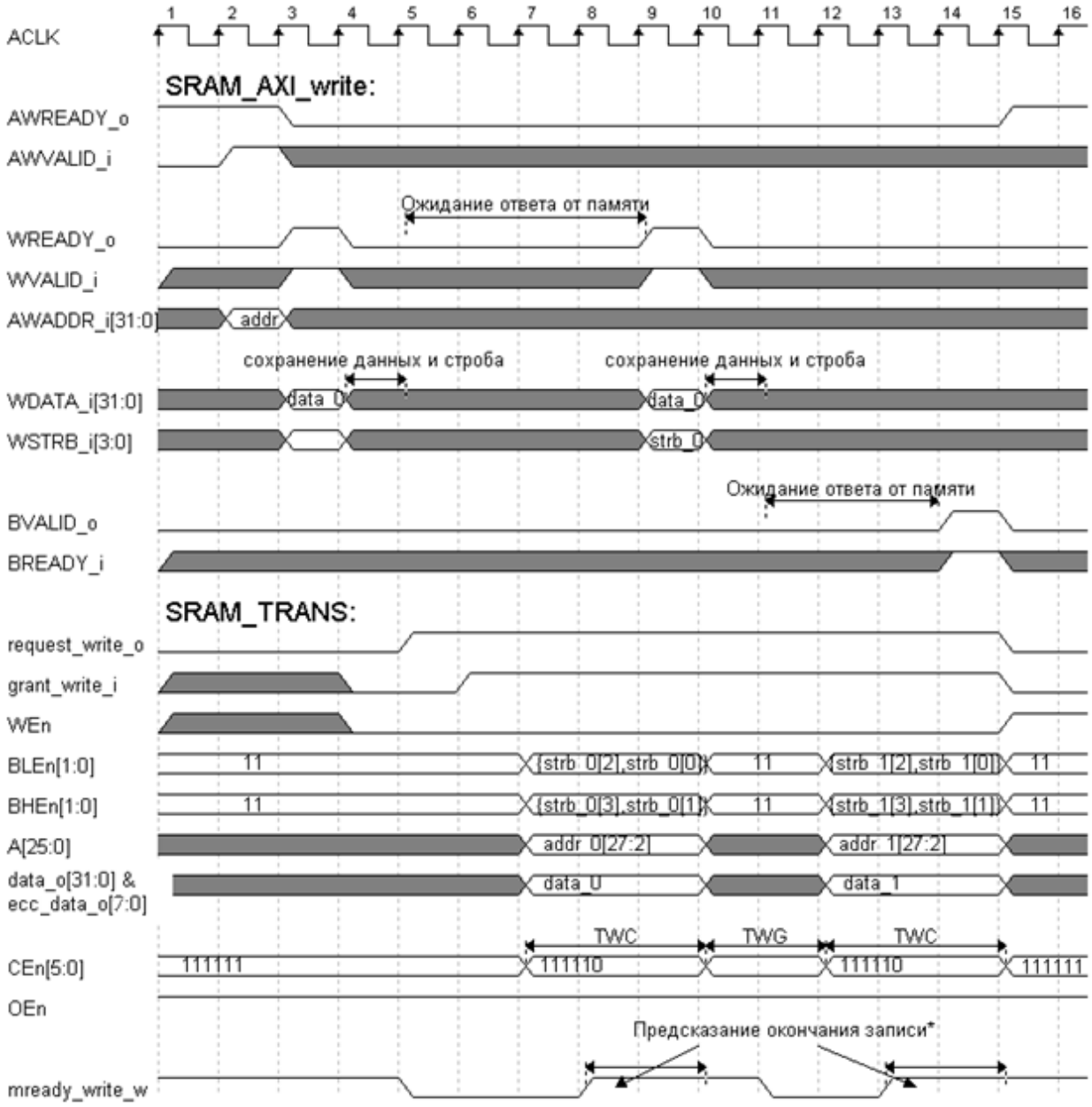


Рисунок 1.171 - Временная диаграмма цикла записи в SRAM/NOR-память (шины BLEn и BHEn к NOR-памяти не подключены)

Длительность цикла записи: TWC = TACLK*TWC(TIMING)
Минимальная длительность неактивной фазы между циклами записи:
TWG = TACLK*TWG(TIMING)
TACLK – период тактового сигнала;
TWC(TIMING), TWG(TIMING), - значения соответствующих полей регистра TIMING.

1.3.11.2.4.5 Арбитраж обращений на чтение и запись

При одновременном поступлении в блок SRAM_AXI_read транзакции чтения из SRAM/NOR-памяти и в блок SRAM_AXI_write транзакции записи в SRAM/NOR-память, сначала производится чтение пакета данных, а затем запись пакета данных.

Если транзакция записи (чтения) поступает в блок SRAM_AXI_write (SRAM_AXI_read), когда транзакция чтения (записи) уже обрабатывается в блоке SRAM_TRANS, то запись (чтение) производится после окончания чтения (записи) пакета данных.

1.3.11.2.4.6 Ошибки обращений

Блоки SRAM_AXI_read и SRAM_AXI_write выдают xRESP = SLVERR в следующих случаях:

- отсутствует банк памяти соответствующий адресу пришедшей AXI транзакции;
- AXI транзакция имеет не поддерживаемые параметры:
  - AxSIZE > 3'b010;
  - AxBURST не равен 2’b01(INCR) или 2’b10(WRAP);
  - AxLOCK не равен 2’b00(Normal access) или 2’b10(Locked access);
  - AxCACHE не равен 4'b000(normal access, secure access, data access);
- транзакция с AxBURST = 2’b10(WRAP) имеет значение длины AxLEN отличное от 1,3,7,15;
- при записи в память, если строб значимости (WSTRB) обрезается размером сообщения (AWSIZE) так, что строб (strb_write_w) идущий от SRAM_AXI_write в транслятор SRAM_TRANS имеет нулевое значение;
- при записи в память, если ID адреса и ID данных различаются awid_r!=rid_r.

Для корректной работы блока необходимо чтобы запись в NOR-память проводилась только при следующих параметрах AXI транзакций:

- AWSIZE = 3'b010;
- WSTRB = 4’b1111.

1.3.11.2.5 Контроллер SRAM/NOR-памяти с интерфейсом MCIF (в CPU)

Блок контроллера SRAM/NOR-памяти предназначен для подключения к СнК до 6 внешних банков SRAM-памяти или NOR-памяти с шинами адреса шириной от 10 до 26 разрядов, шириной шины данных 32 разряда и шириной шины проверочных данных 8 разрядов.

Структурная схема блока контроллера SRAM/NOR-памяти с примером подключения к нему двух банков SRAM-памяти приведена на рисунке 1.173.

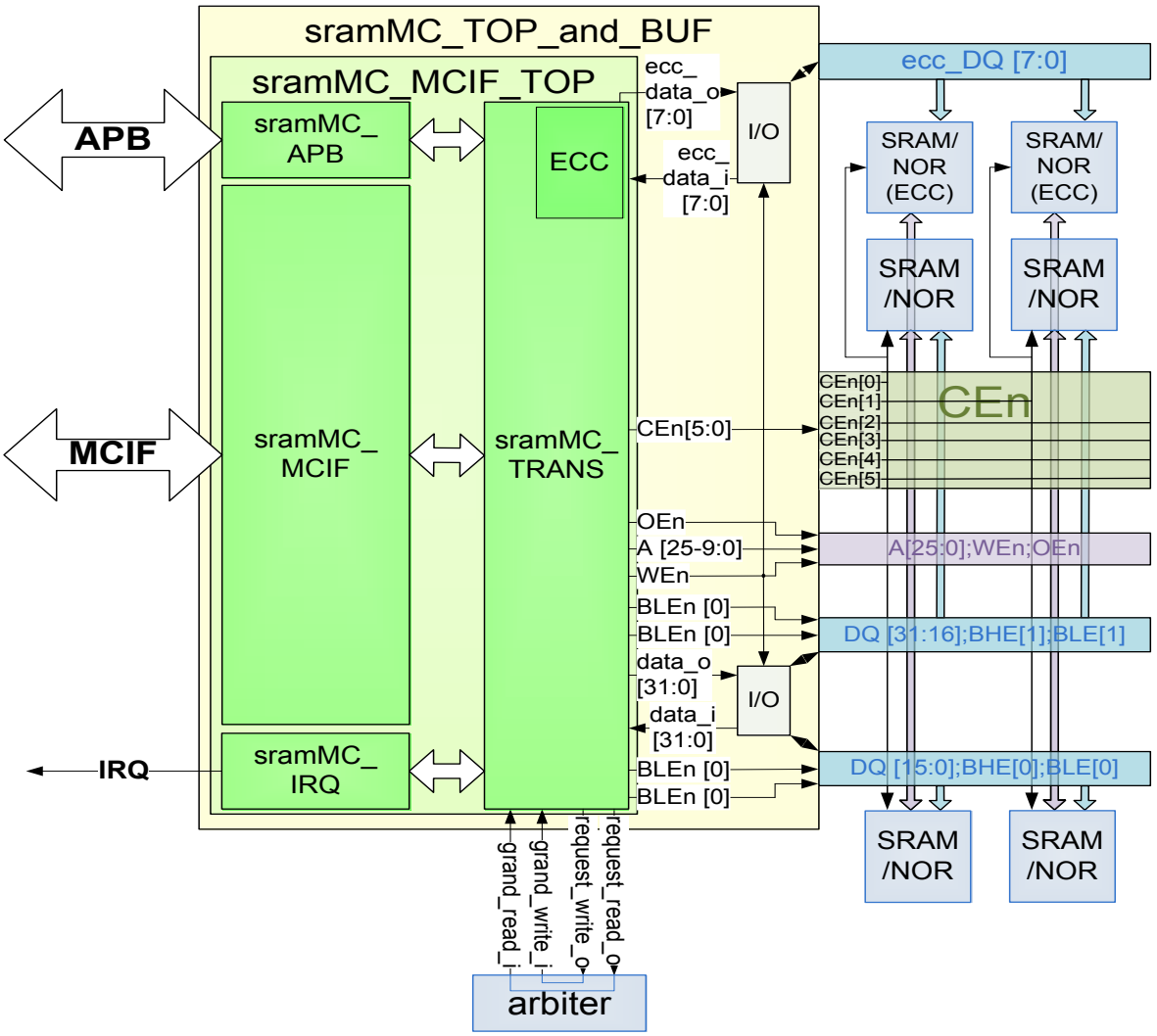


Рисунок 1.172- Структурная схема блока контроллера SRAM/NOR-памяти

Каждый банк памяти управляется своим сигналом CEn[i] для перевода памяти из режима хранения в режим чтения/записи. Для SRAM-памяти сигналы BHEn[1:0] и BLEn[1:0] позволяют записывать только нужные байты из 4 байтового слова, для NOR-памяти они не используются.

Для возможности совместного использования нескольких контроллеров с общими шинами сигналов к памяти блок имеет интерфейс для организации арбитра.

1.3.11.2.5.1 Описание назначения и функций блоков

Контроллер SRAM/NOR-памяти (блок sramMC_MCIF_TOP) состоит из блоков sramMC_MCIF, sramMC_APB, sramMC_IRQ и sramMC_TRANS.

1.3.11.2.5.1.1 Блок sramMC_MCIF

Блок sramMC_MCIF системный интерфейс MCIF-slave для подключения к коммутационной среде СнК и отвечют следующим требованиям:

- Список и назначение сигналов системного интерфейса MCIF-slave соответствуют спецификации MCIF II revision 03.
- Системный интерфейс MCIF-slave имеет следующие функциональные особенности:
  - глубину очереди равную 1 (Queue Depth = 4'b0001);
  - размер пакетной транзакции до 8ми 128-разрядных слов (максимальное значение M_TSIZE = 00111, 8 Beats);

-
- Twr_resp_lat = 2.

- Системный интерфейс MCIF-slave не поддерживает следующие функциональные свойства протокола MCIF:
  - Отмену транзакций (Cancel), игнорирует данный сигнал;
  - Out-of-Order порядок выдачи данных, игнорирует данный сигнал;
  - ECC Pass-through Operation.

1.3.11.2.5.1.2 Блок sramMC_APB

Блок sramMC_APB предназначен для программирования регистров контроллера.

Список и назначение сигналов системного интерфейса APB-slave соответствуют спецификации AMBA3 APB.

1.3.11.2.5.1.3 Блок sramMC_IRQ

Блок sramMC_IRQ генерирует уровневые прерывание, для сброса прерывания используется регистр STATUS_IRQ чтение которого осуществляется по APB-интерфейсу.

1.3.11.2.5.1.4 Блок sramMC_TRANS

Блок sramMC_TRANS контроллера (транслятор) реализует преобразование MCIF-транзакции в класс временных диаграмм интерфейса для подключения микросхем SRAM/NOR-памяти, а также реализует работу с кодами обнаружения и коррекции ошибок (ECC).

1.3.11.2.5.1.5 ECC

Контроллер SRAM и NOR-памяти поддерживает возможность записи и чтения не только данных, но и проверочных данных. Проверочные данные можно передавать параллельно с основными данными. Для этого интерфейс подключения микросхем SRAM- и NOR-памяти к совмещенному контроллеру SRAM- и NOR-памяти имеет дополнительные выводы для передачи проверочных данных.

Для возможности исправить все однократные и одновременно обнаружить все двукратные ошибки, используется расширенный код Хэмминга. При кодировании 32 бит данных требуется 7 проверочных бит.

Для SRAM поддерживается возможность записи в режиме с ECC не полных слов за счет реализации режима read-modify-write, когда для сохранения корректного кода ECC сначала прочитывается 32-битное слово данных хранящееся по нужному адресу, из него берутся те байты которые надо оставить без изменений, добавляются байты для записи, считается ECC код слова, и в память записываются данные и ECC-код.

Для корректной записи не полных слов в режиме с ECC необходимо чтобы перед началом работы вся используемая память была проинициализирована.

Чтение не полных слов в режиме с ECC поддерживается за счет того что из памяти вычитывается полное 32-х разрядное слово с проверочными битами, а далее уже MCIF-slave выбирает нужные байты для передачи.

Для взаимодействия контроллера (с поддержкой кодов коррекции ошибок) и СнК в случаях возникновения ошибок необходимо генерировать прерывания.

1.3.11.2.5.2 Регистровая карта

Контроллер SRAM/NOR-памяти имеет набор регистров (см. Таблица 1.877), программируемых по APB интерфейсу, позволяющих задавать конфигурации и параметры временных диаграмм интерфейса для подключения различного количества и различных моделей микросхем SRAM- или NOR-памяти.

Таблица 1.877 - Регистровая карта

Адрес	Название	Значение после сброса	Доступ	Описание
0x00	ID	0x524F4E53*	RO	Идентификатор устройства
0x04	VERSION	0x00000201*	RO	Версия устройства
0x08	CONFIG	0x0116000F*	RW	Регистр настройки конфигурации
0x0C	TIMINGS	0x06060100*	RW	Регистр настройки временных параметров
0x10	MASK_IRQ	0x0000003E*	RW	Маска прерываний
0x14	STATUS_IRQ	0x0	RO	Регистр статуса прерываний, чтением данного регистра осуществляется сброс прерывания и значения данного регистра
0x18	-	-	-	reserve
0x1c	HIGH_ADDR	0x0	RW	Старшие биты адреса контроллера
0x20	MANAGEMENT_ECC	0x0	RW	Регистр управления работой с памятью для ECC
0x24	DATA_ECC_WRITE_MEM	0x0	RW	Регистр данных для передачи по шине ECC_data[7:0] при записи в память
0x28	DATA_ECC_READ_MEM	0x0	RO	Регистр данных принимаемых по шине ECC_data[7:0] при чтении из память
0x2c	-	-	-	reserve
0x30	ECC_ERR_ADDR	0x0	RO	Регистр адреса 32-битного слова в котором была обнаружена ECC ошибка, младшие 32 бита.
0x34	TR_ERR_ADDR	0x0	RO	Регистр адреса транзакции в которой была обнаружена ошибка, младшие 32 бита.

* значения могут уточняться при сборке системы

1.3.11.2.5.2.1 CONFIG

Таблица 1.878 - CONFIG

Биты	Поле	Описание
5:0	chip_num	Активация подключенных микросхем памяти 0 – не активно, 1 – активно.
7:6	-	reserve
8	ce_mode	Режим активации подключенных микросхем 0 – CEn устанавливается в соответствии со старшими разрядами адреса 1 – CEn устанавливается в соответствии со значением поля chip_num
15:9	-	reserve
20:16	addr_size	Абсолютное значение ширины шины адреса (поддерживаемые значения от 10 до 26)

Продолжение таблицы

Биты	Поле	Описание
23:21	-	reserve
24	ECC_ACTIVE	0 – коррекцию ошибок отключена 1 – коррекцию ошибок включена
31:25	-	reserve

1.3.11.2.5.2.2 TIMINGS

Таблица 1.879 - TIMINGS

Биты	Поле	Описание
7:0	TRG	Минимальная длительность неактивной фазы между циклами чтения
15:8	TWG	Минимальная длительность неактивной фазы между циклами записи
23:16	TRC	Длительность цикла чтения
31:24	TWC	Длительность цикла записи

1.3.11.2.5.2.3 MASK_IRQ

Таблица 1.880 - MASK_IRQ

Биты	Поле	Описание
0	MASK_S_ERR	При присвоении лог. 1 данному биту генерируется прерывание при обнаружении однократной ошибки. Рекомендуемое значение: 0
1	MASK_D_ERR	При присвоении лог. 1 данному биту генерируется прерывание при обнаружении двукратной ошибки. Рекомендуемое значение: 1
2	MASK_ERR_ADDR	При присвоении лог. 1 данному биту генерируется прерывание при обнаружении ошибки адреса транзакции. Рекомендуемое значение: 1
3	MASK_ERR_ADR_RS_PTY	При присвоении лог. 1 данному биту генерируется прерывание при обнаружении ошибки передачи адреса транзакции. Рекомендуемое значение: 1
4	MASK_ERR_BE_PTY	При присвоении лог. 1 данному биту генерируется прерывание при обнаружении ошибки передачи BE. Рекомендуемое значение: 1
5	MASK_ERR_CMD_PTY	При присвоении лог. 1 данному биту генерируется прерывание при обнаружении ошибки передаче команд. Рекомендуемое значение: 1
31:6	-	reserve

1.3.11.2.5.2.4 STATUS_IRQ

Таблица 1.881 - STATUS_IRQ

Биты	Поле	Описание
0	ECC_S_ERR	Лог. 1 устанавливается при обнаружении однократной ошибки
1	ECC_D_ERR	Лог. 1 устанавливается при обнаружении двукратной ошибки
2	ERR_ADDR	Лог. 1 устанавливается при обнаружении ошибки адреса транзакции
3	ERR_ADR_RS_PTY	Лог. 1 устанавливается если в MCIF транзакции сигнал ADR_RS_PTY не совпал с ожидаемым
4	ERR_BE_PTY	Лог. 1 устанавливается если в MCIF транзакции сигнал BE_PTY не совпал с ожидаемым
5	ERR_CMD_PTY	Лог. 1 устанавливается если в MCIF транзакции сигнал CMD_PTY не совпал с ожидаемым
7:6	-	reserve
18:8	COUNT_S_ERR	Значение счетчика соответствует числу однократных ошибок произошедших до чтения регистра

Продолжение таблицы 1.882

Биты	Поле	Описание
19	OVERFLOW_S_ERR	Переполнение счетчика однократных ошибок, выставляется в лог. 1 если число однократных ошибок превышает 2047
30:20	COUNT_D_ERR	Значение счетчика соответствует числу двукратных ошибок произошедших до чтения регистра
31	OVERFLOW_D_ERR	Переполнение счетчика двукратных ошибок, выставляется в лог. 1 если число двукратных ошибок превышает 2047

При возникновении прерывания блок может продолжать работу.
При чтении данного регистра все его поля сбрасываются в лог. 0.

1.3.11.2.5.2.5 MANAGEMENT_ECC

Таблица 1.883 - MANAGEMENT_ECC

Биты	Поле	Описание
1:0	MODE_ECC_DATA	0 – передача по шине ECC_data[7:0], управляющих сигналов (таких же как на шине data[7:0]), для NUMBER_COMAND слов данных, далее передача кода обнаружение и коррекции ошибок (ECC_data[7] – не используется), при этом счетчик сбрасывается в изначально записанное значение; чтение в данном режиме с ECC (также коды прочитанные из ECC-памяти записываются в регистр DATA_ECC_READ_MEM); при записи данных в SRAM NUMBER_COMAND должен быть равен 0. 1 – передача по шине ECC_data[7:0], управляющих сигналов из регистра DATA_ECC_WRITE_MEM при записи управляющих команд в NOR-память; чтение в данном режиме без ECC, данные прочитанные из ECC-памяти записываются в регистр DATA_ECC_READ_MEM. 2 – передача по шине ECC_data[7:0], управляющих сигналов (таких же как на шине data[7:0]) при записи управляющих команд в NOR-память; чтение в данном режиме без ECC, данные прочитанные из ECC-памяти записываются в регистр DATA_ECC_READ_MEM. 3 – reservd
3:2	-	reserve
7:4	NUMBER_COMMAND	Число управляющих сигналов до передачи в память кодов обнаружение и коррекции ошибок
31:8	-	reserve

1.3.11.2.5.2.6 HIGH_ADDR

Таблица 1.884 - HIGH_ADDR

Биты	Поле	Описание
31:0	HIGH_ADDR	Старшие биты адреса контроллера [0:31], для проверки правильности обращения. В соответствии со значением данного регистра проверяются биты шины адреса большие чем addr_size + 5. (более подробное описание адресации памяти приведено ниже)

1.3.11.2.5.2.7 DATA_ECC_WRITE_MEM

Таблица 1.885 - DATA_ECC_WRITE_MEM

Биты	Поле	Описание
7:0	DATA_ECC_WRITE_MEM	Данные для передачи по шине ECC_data[7:0] при записи в память
31:8	-	reserve

1.3.11.2.5.2.8 DATA_ECC_READ_MEM

Таблица 1.886 - DATA_ECC_READ_MEM

Биты	Поле	Описание
7:0	DATA_ECC_READ_MEM	Данные принимаемые по шине ECC_data[7:0] при чтении из память
31:8	-	reserve

1.3.11.2.5.2.9 ECC_ERR_ADDR

Таблица 1.887 - ECC_ERR_ADDR

Биты	Поле	Описание
31:0	ECC_ERR_ADDR	Адрес слова (младшие 32 бита), ошибка ECC в котором вызвала прерывание.

1.3.11.2.5.2.10 TR_ERR_ADDR

Таблица 1.888 - TR_ERR_ADDR

Биты	Поле	Описание
31:0	TR_ERR_ADDR	Адрес транзакции (младшие 32 бита), ошибка (адреса или битов четности) в которой вызвала прерывание (данные ошибки соответствуют битам статуса [2:5]).

1.3.11.2.5.3 Адресация SRAM/NOR-памяти

К контроллеру SRAM/NOR-памяти можно подключить до 6 внешних банков SRAM/NOR-памяти с шинами адреса шириной до 26 разрядов и шириной шины данных 32 разряда. К каждому банку подведены общие шина адреса (A), шина данных (I/O[31:0]), управляющие сигналы (CEn,OEn,WEn) и сигналы значимости битов ячеек памяти (BHEn[1:0],BLEn[1:0]; используются только для SRAM-памяти). Описание сигналов значимости битов ячеек памяти приведено в таблице ниже.

Таблица 1.889 - Описание сигналов значимости данных

Сигнал	Значимые биты данных
BLEn[0]=1	7:0
BHEn[0]=1	15:8
BLEn[1]=1	23:16
BHEn[1]=1	31:24

Схема примера подключения двух банков памяти, каждый из которых состоит из двух микросхем SRAM-памяти с шириной шины данных 16 разрядов, приведена на рисунке 1.

Каждому адресу в банке SRAM/NOR-памяти соответствует 32 бита данных. При обращении к SRAM/NOR-памяти по MCIF интерфейсу адресация производится побайтно, описание представлено в таблице 1.764.

Таблица 1.890 - Описание адресации памяти

Биты адреса поступающего на MCIF при ширине шины адреса равной значению поля addr_size регистра CONFIG	Значение поля ce_mode регистра CONFIG	Описание	Сигналы к SRAM/NOR-памяти
34:35	x	адрес байта в слове	BHEn[1:0],BLEn[1:0]*
35-(size_addr_m_r+1):33	x	адрес слова в банке SRAM/NOR-памяти	A[size_addr_m_r-1:0]
35-(size_addr_m_r+4): 35-(size_addr_m_r+2)	0	адрес банка SRAM/NOR-памяти (описание ниже)	CEn[5:0]

x – любое значение

*Шины BLEn и BHEn к NOR-памяти не подключены.

Старшие биты адреса контроллером не проверяются.

В случае если в поле ce_mode регистра CONFIG записан 0 поле chip_num преобразуется в зависимости от адреса в сигналы CEn для микросхем памяти в соответствии с таблицей 1.765.

Таблица 1.891 -Описание выбора пары микросхем SRAM/NOR-памяти при ce_mode=0

Подключение микросхем SRAM-памяти	Значения трех старших используемых битов шины адреса MCIF I_M_ADR [35-(size_addr_m_r+4): 35-(size_addr_m_r+2)]	Доступ к микросхемам SRAM-памяти
chip_num [0]=1	3'b000	CEn[0]=0
chip_num [1]=1	3'b001	CEn[1]=0
chip_num [2]=1	3'b010	CEn[2]=0
chip_num [3]=1	3'b011	CEn[3]=0
chip_num [4]=1	3'b100	CEn[4]=0
chip_num [5]=1	3'b101	CEn[5]=0

В случае если в поле ce_mode регистра CONFIG записана 1 на шину CEn[5:0] контроллера передаётся инвертированное значение поля chip_num[5:0]. Для корректной работы контроллера не допускается одновременное подключение нескольких микросхем памяти в этом режиме (поле chip_num[5:0] должно содержать только одну 1).

1.3.11.2.5.4 Описание работы контроллера

1.3.11.2.5.4.1 Состояние после сброса

sramMC_MCIF находятся в состоянии ожидания прихода транзакций на запись или чтение памяти.

sramMC_APB находится в состоянии ожидания прихода транзакций на запись или чтение регистров.

sramMC_IRQ устанавливает выход внешнего прерывания в не активное состояние.

Значения внутренних программируемых регистров в трансляторе соответствует значениям таблицы 2.

Транслятор свободен для программирования или обращений на чтение/запись данны.

1.3.11.2.5.4.2 Состояние ожидания

Блок контроллера памяти находится в состоянии ожидания если:

- по шине APB не приходят сигналы на запись регистров;
- по шине MCIF не приходят сигналы на запись или чтение данных;
- отработаны все предыдущие обращения;
- внешнее прерывание выставлено в не активное состояние.

Когда блок контроллера памяти находится в состояние ожидания, на его выходах будут выставлены следующие сигналы:

- к SRAM/NOR-памяти:
  - Все микросхемы памяти находятся в режиме хранения (CEn = 1);

- sramMC_APB:
  - Готов принять транзакцию (PREADY_o = 1);
- sramMC_MCIF:
  - Готов принять транзакцию (O_M_IDLE = 1);
- sramMC_IRQ:
  - Прерывание в не активном состоянии (IRQ = 0).

1.3.11.2.5.4.3 Перепрограммирование внутренних регистров

Для программирования внутренних регистров контроллера используется интерфейс APB. При поступлении транзакции записи по шине APB во внутренние регистры APB записываются поступающие данные. Когда транслятор находится в не занятом состоянии (нет запросов на работу с памятью), то значения данных регистров APB переписываются на регистры транслятора. Когда транслятор занят (ожидает разрешения арбитра на работу с памятью или производит запись(чтение) данных), то значения данных регистров APB переписываются на регистры транслятора только когда транслятор освободится.

При поступлении транзакции чтения в sramMC_APB производится чтение внутренних регистров APB. Это может осуществляться в любой момент времени.

1.3.11.2.5.4.4 Чтение данных из памяти

Обработка транзакции чтения из памяти:

- контроллер делает запрос (request_read_o=1) к ARB_SRAM/NOR.

Когда арбитр дает доступ к памяти (grant_read_i=1), контроллер начинает чтение, выставляя следующие сигналы на выходы к памяти:

- WEn = 1;
- BLEn = 2'b00 (BLEn переключается в неактивное состояние (BLEn = 2'b11) при не активном CEn);*
- BHEn = 2'b00 (BHEn переключается в неактивное состояние (BHEn = 2'b11) при не активном CEn);*
- A = addr_read_w[27:2];
- На шине CEn в ноль устанавливается только один из шести битов соответствующий нужному банку памяти (описание в разделе адресация SRAM/NOR-памяти);
- OEn =0 (OEn переключается в неактивное состояние (OEn =1) при не активном CEn).

*Шины BLEn и BHEn к NOR-памяти не подключены.

Как только сигналы на выходах выставлены они удерживаются в течение времени прописанного в поле TRC регистра TIMING, после чего данные поступающие от памяти (в том числе и проверочные данные, если включен режим поддержки кодов коррекции ошибок) считаются истинными и защелкиваются на внутренний регистр блока sramMC_TRANS, если включен режим поддержки кодов коррекции ошибок, то блок sramMC_TRANS проверяет данные на наличие ошибок и при обнаружении ошибок передает об этом данные контроллеру прерываний (sramMC_IRQ), а также при возможности исправляет данные и выдает в блока sramMC_MCIF 32 бита данных.

В случае если это последние 32 бита данных транзакции, транслятор переводит сигнал request_read_o к арбитру ARB_SRAM/NOR в неактивное состояние (request_read_o=0) и приступает к обработке следующих обращений или переходит в режим ожидания следующего обращения.

В случае если это не последние 32 бита данных транзакции и минимальная длительность неактивной фазы между циклами чтения равна 0, то транслятор повторяется чтение.

В случае если минимальная длительность неактивной фазы между циклами чтения не равна 0, то транслятор после защелкивания данных переводит шину CEn в не активное состояние и

удерживает это состояние в течение времени прописанного в поле TRG регистра TIMING, по прошествии этого времени транслятор начинает чтение из следующего адреса памяти.

Если по адресу, пришедшему от sramMC_MCIF, отсутствует банк памяти, то транслятор sramMC_TRANS возвращает блоку sramMC_MCIF сигнал об ошибке.

Временная диаграмма примера цикла чтения пакетной транзакции состоящей из 2 слов представлена на рисунке 1.174.

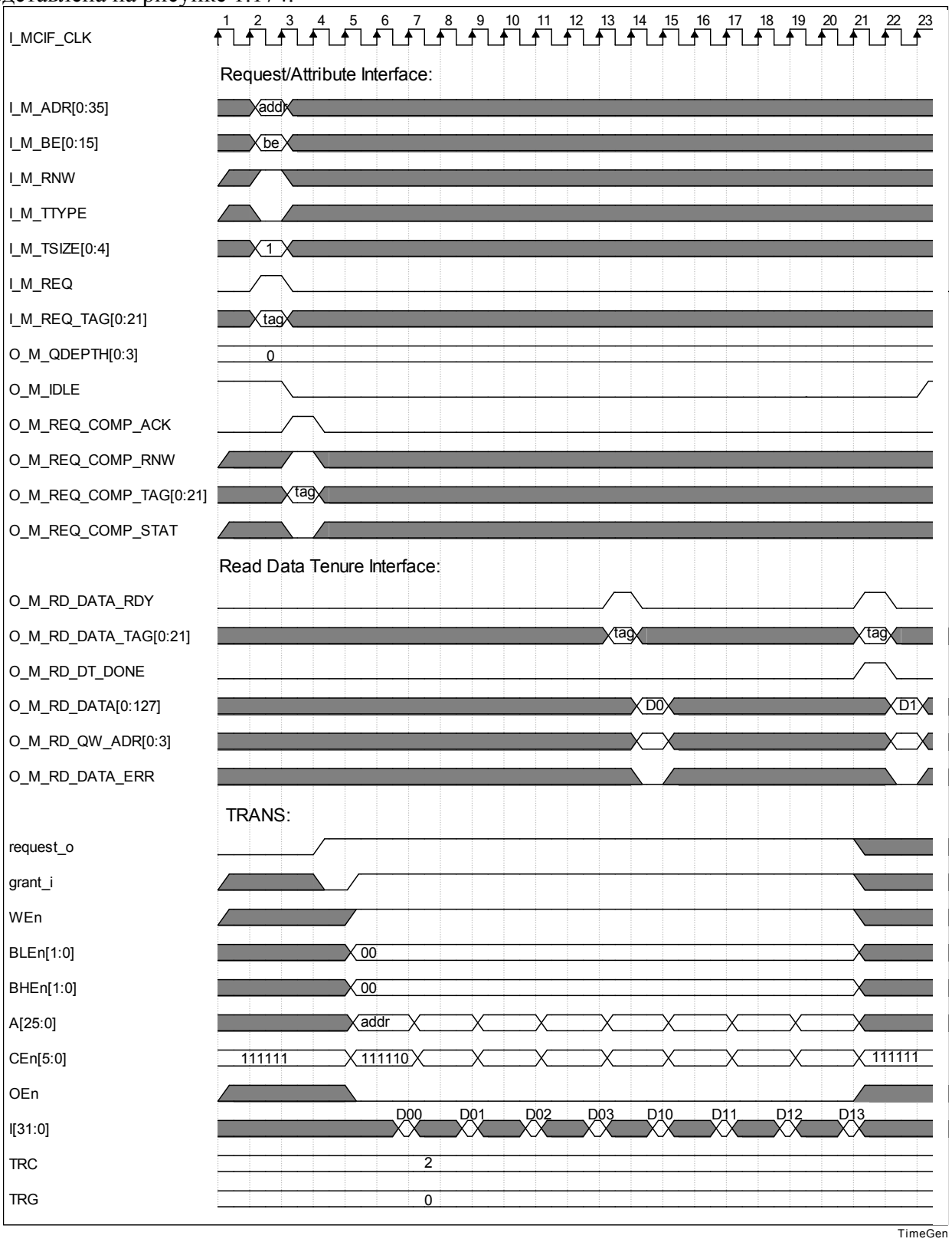


Рисунок 1.173 Временная диаграмма цикла чтения при постоянном доступе к SRAM/NOR-памяти и при TRG=0 (шины BLEn и BHEn к NOR-памяти не подключены)

Длительность цикла чтения: TRC = TACLK*TRC(TIMING) ;
TACLK – период тактового сигнала;
Длительность неактивной фазы между циклами чтения: TRG = TACLK*TRG(TIMING) ;
TWC(TIMING), TWG(TIMING), - значения соответствующих полей регистра TIMING.

1.3.11.2.5.4.5 Запись данных в память

Обработка транзакции запись в память:

- контроллер делает запрос (request_write_o=1) к арбитру ARB_SRAM/NOR.

Когда арбитр дает доступ к памяти (grant_write_i=1), транслятор начинает запись, выставляя следующие сигналы на выходы к памяти:

- WEn = 0;
- BLEn[0] = ~strb_r[0]*;
- BLEn[1] = ~strb_r[2]*;
- BHEn[0] = ~strb_r[1]*;
- BHEn[1] = ~strb_r[3]*;
- data_o = data_write_r;
- ecc_data_o
  - равняется 0 если ECC_ACTIVE=0;
  - рассчитывается по расширенному коду Хемминга если ECC_ACTIVE=1, MODE_ECC_DATA=0 и NUMBER_COMMAND=0;
  - ecc_data_o[7:0] = data_o[7:0](используется для программирования NOR-памяти) если ECC_ACTIVE=1,MODE_ECC_DATA=0 и NUMBER_COMMAND!=0, или MODE_ECC_DATA=2;
  - ecc_data_o[7:0] = DATA_ECC_WRITE_MEM[7:0] (используется для программирования NOR-памяти или для внесения ошибок в ECC память для тестирования) если ECC_ACTIVE=1, MODE_ECC_DATA=1;
- A = addr_write_r[27:2];
- На шине CEn в ноль устанавливается только один из шести битов соответствующий нужному банку памяти (описание в разделе адресация SRAM/NOR-памяти);
- Сигналы OEn имеет значение противоположное WEn (OEn переключается в неактивное состояние (OEn =1) при не активном CEn).

*BLEn, BHEn переключается в неактивное состояние (BLEn = 2'b11, BHEn = 2'b11) при не активном CEn, к NOR-памяти данные шины не подключены;

Как только сигналы на выходах выставлены они удерживаются в течение времени прописанного в поле TWC регистра TIMING, после чего данные поступающие в память считаются записанными, и транслятор готов для записи следующих данных.

Если записаны не последние 32 бита данных транзакции, контроллер переводит шину CEn в не активное состояние и удерживает это состояние в течение времени прописанного в поле TWG регистра TIMING, по прошествии этого времени транслятор начинает запись следующих данных.

По окончании записи последних данных пакетной транзакции транслятор переводит сигнал request_write_o к арбитру ARB_SRAM/NOR в неактивное состояние (request_write_o=0) и приступает к обработке следующих обращений или переходит в режим ожидания следующего обращения.

Если по адресу, пришедшему по MCIF, отсутствует банк памяти, то контроллер возвращает сигнал об ошибке.

Временная диаграмма примера цикла записи транзакции состоящей из 1 слова представлена на рисунке Рисунок 1.174, для того чтобы диаграмма имела такой вид должна быть или отключена коррекция ошибок или слова должны быть полными, иначе будет работать механизм read-modify-write.

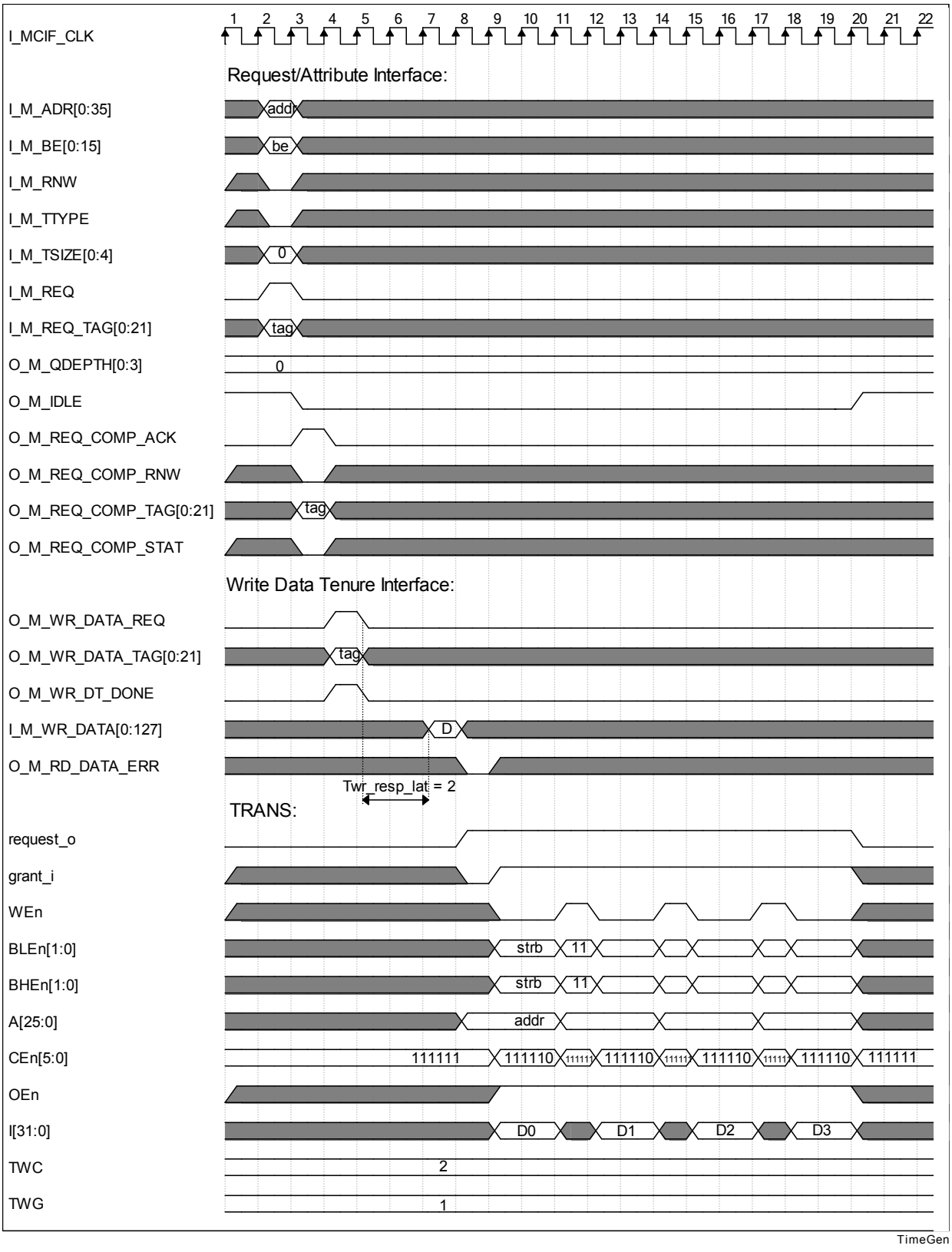


Рисунок 1.174 Временная диаграмма цикла записи в SRAM/NOR-память (шины BLEn и BHEn к NOR-памяти не подключены)

*Длительность цикла записи: TWC = TACLK*TWC(TIMING)*

Минимальная длительность неактивной фазы между циклами записи:

*TWG = TACLK*TWG(TIMING)*
TACLK – период тактового сигнала;
TWC(TIMING), TWG(TIMING), - значения соответствующих полей регистра TIMING.

1.3.11.2.5.4.6 Отработка ошибок

1.3.11.2.5.4.6.1 С возможностью выставления прерывания:

1.3.11.2.5.4.6.1.1 ECC_S_ERR

Если в слове читаемых данных (в том числе и в операции чтения модификация запись(RMW)) была однократная ошибка, то эта ошибка исправляется и в MCIF интерфейс (или при RMW в память) выдаются уже достоверные данные. Если в регистре MASK_IRQ прерывание по данному событию активно (1), то адрес данной ошибки (с точностью до 32 битовых слов) сохраняется в регистр ECC_ERR_ADDR.

1.3.11.2.5.4.6.1.2 ECC_D_ERR

Если в слове читаемых данных (в том числе и в операции чтения модификация запись(RMW)) была двукратная ошибка, то эта ошибка не исправляется и в MCIF интерфейс (или при RMW в память) выдаются уже данные с ошибками. Если в регистре MASK_IRQ прерывание по данному событию активно (1), то адрес данной ошибки (с точностью до 32 битовых слов) сохраняется в регистр ECC_ERR_ADDR.

1.3.11.2.5.4.6.1.3 ERR_ADR_RS_PTY

Если в контроллер пришла транзакция с ошибкой в битах четности адреса, то контроллер завершает эту транзакцию (не читая/записывая данные из/в памяти) выставляя O_M_RD(WR)_DATA_ERR. Если в регистре MASK_IRQ прерывание по данному событию активно (1), то адрес данной транзакции сохраняется в регистр TR_ERR_ADDR.

1.3.11.2.5.4.6.1.4 ERR_CMD_PTY

Если в контроллер пришла транзакция с ошибкой в битах четности команды, то контроллер завершает эту транзакцию (не читая/записывая данные из/в памяти) выставляя O_M_RD(WR)_DATA_ERR. Если в регистре MASK_IRQ прерывание по данному событию активно (1), то адрес данной транзакции сохраняется в регистр TR_ERR_ADDR.

1.3.11.2.5.4.6.1.5 ERR_BE_PTY

Если в контроллер пришла транзакция с TTYPE=0 с ошибкой в битах четности BE, то контроллер завершает эту транзакцию (не читая/записывая данные из/в памяти) выставляя O_M_RD(WR)_DATA_ERR. Если в регистре MASK_IRQ прерывание по данному событию активно (1), то адрес данной транзакции сохраняется в регистр TR_ERR_ADDR.
Если TTYPE=1 то BE_PTY не анализируются.

1.3.11.2.5.4.6.2 Без возможности выставления прерывания:

1.3.11.2.5.4.6.2.1 Ошибка битов четности данных

Если в транзакции записи в контроллер пришли данные с ошибкой в битах четности данных, то данные записываются в память с ошибкой и выставляется O_M_WR_DATA_ERR.

1.3.12 Подсистема низкоскоростных интерфейсов LSIF1

1.3.12.1 Описание подсистемы LSIF1

1.3.12.1.1 Структурная схема и состав блоков подсистемы LSIF1

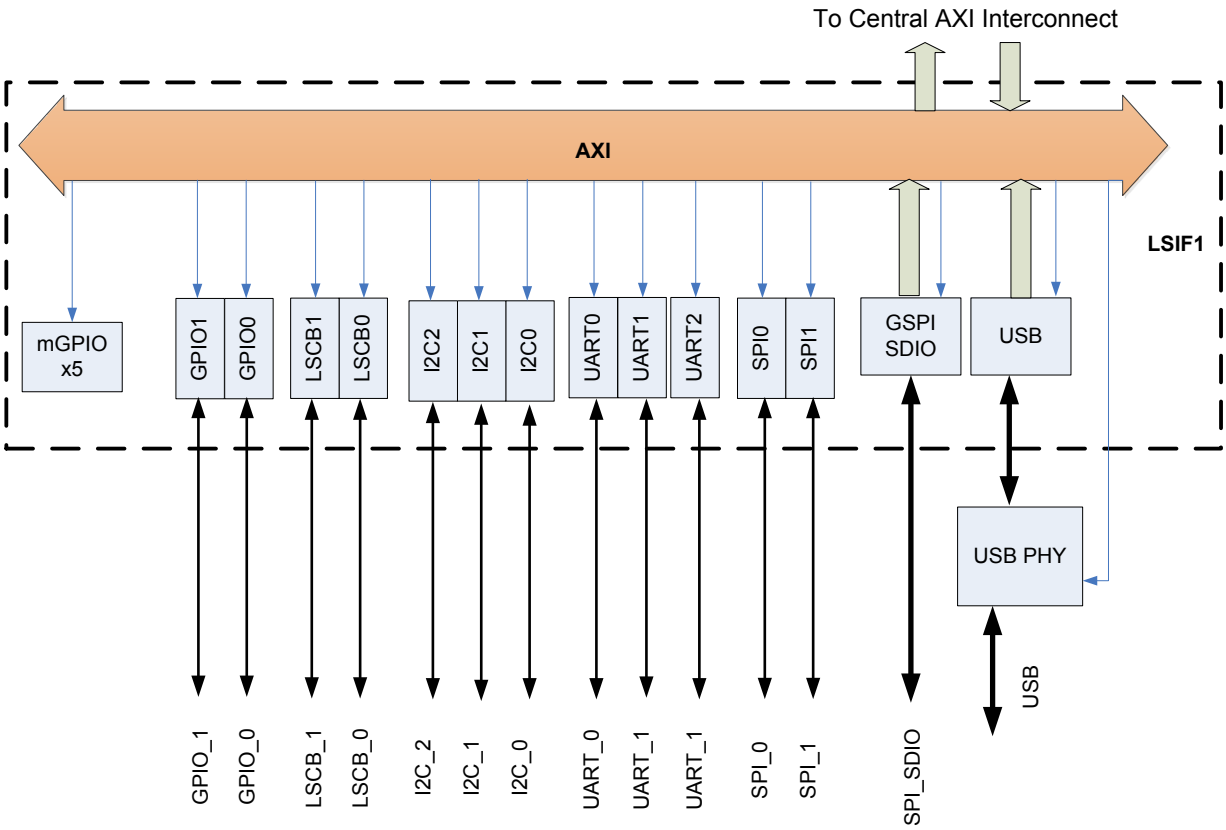


Рисунок 1.196 - Структурная схема подсистемы LSIF1

Подсистема LSIF0 состоит из следующих блоков:

- Два контроллера канала мультиплексного обмена по стандарту ГОСТ Р 52070-2003 (LSCB0, LSCB1;
- Три контроллера шины I2C (I2C0, I2C1, I2C2;
- Три интерфейсных контроллера UART (UART0, UART1, UART2;
- Два контроллера SPI (SPI0, SPI1;
- Контроллер GSPI/SDIO;
- Контроллер USB;
- Два контроллера портов программируемого ввода-вывода GPIO (GPIO0, GPIO1;
- Пять контроллеров мультиплексируемых портов программируемого ввода-вывода mGPIO;
- Блок управляющих регистров LSIF1_CTRL.

1.3.12.1.2 Адресация блоков подсистемы LSIF1

Таблица 1.593 - Карта адресов подсистемы LSIF1

Блок	Адрес на шине APB
LSCB0	0x3C050000 – 0x3C054FFF
LSCB1	0x3C055000 –0x3C059FFF
I2C0	0x3C05A000 – 0x3C05AFFF
I2C1	0x3C05B000 – 0x3C05BFFF
I2C2	0x3C05C000 – 0x3C05CFFF
UART0	0x3C05D000 – 0x3C05DFFF
UART1	0x3C05E000 – 0x3C05EFFF
UART2	0x3C05F000 – 0x3C05FFFF
LSIF1 CONFIG	0x3C060000 – 0x3C060FFF
SPI0	0x3C061000 – 0x3C061FFF
SPI1	0x3C062000 – 0x3C062FFF
GSPI	0x3C063000 – 0x3C063FFF
SDIO	0x3C064000 – 0x3C064FFF
GPIO0	0x3C065000 – 0x3C065FFF
GPIO1	0x3C066000 – 0x3C066FFF
mGPIO x 5	0x3C067000 – 0x3C06BFFF (APBx5)
USB PHY	0x3C06C000 – 0x3C06CFFF
USB (AHB)	0x3C06D000 – 0x3C06DFFF

1.3.12.1.3 Блок управляющих регистров LSIF1_CTRL

Блок управляющих регистров LSIF1_CTRL подсистемы LSIF1 микросхемы предназначен для настройки работы подсистемы. Блок позволяет управлять следующими функциями:

- Расширение адресного пространства для контроллеров USB и GSPI_SDIO;
- Управление значениями входных для реализации сценариев управления питанием портов блока USBPHY.

Таблица 1.594 - Карта регистров блока LSIF1_ CTRL

Название	**Адрес**	**Доступ**	**Значение после сброса**	**Описание**
ID	0x0	RO	0x3149534c	Регистр идентификации
USB_ADDR_EXTEND	0x04	RW	0x0	Расширение адреса для 36-разрядной адресации контроллера USB: Биты [3:0] - соответствуют 4 старшим разряда адреса
SDIO_RADDR_EXTEND	0x08	RW	0x0	Расширение адреса для 36-разрядной адресации канала чтения из GSPI_SDIO: Биты [3:0] - соответствуют 4 старшим разряда адреса
SDIO_WADDR_EXTEND	0x0C	RW	0x0	Расширение адреса для 36-разрядной адресации канала записи GSPI_SDIO: Биты [3:0] - соответствуют 4 старшим разряда адреса
USB_RST	0x10	RW	0x3	Задает значение входных портов POR_reset, utmi_reset , NRSTX и SUSPENDM для реализации сценариев управления питанием блока USBPHY

1.3.12.1.3.1 Поля регистра USB_RST

Задает значение входных портов POR_reset, utmi_reset , NRSTX и SUSPENDM для реализации сценариев управления питанием блока USBPHY (см.раздел 1.3.12.2.8).

Таблица 1.595 - Поля регистра USB_RST

Биты	Поле	Знач. по умолч.	Описание
0	POR_reset	0x1	Задает значение входного порта POR_reset блока USBPHY
1	utmi_reset_phy	0x1	Задает значение входного порта utmi_reset блока USBPHY
2	utmi_reset_musb	0x0	Задает значение входного порта NRSTX блока USB20OTG
3	utmi_suspendm_en	0x0	Сигнал разрешения передачи состояния выходного порта SUSPENDM блока USB20OTG на входной порт utmi_suspendm блока USBPHY: 0 - на входной порт utmi_suspendm блока USBPHY потоянно подается 0; 1 - на входной порт utmi_suspendm блока USBPHY передается состояние выходного порта SUSPENDM блока USB20OTG
31:4	-	-	reserve

1.3.12.1.4 Схема мультиплексирования портов интерфейсных контроллеров подсистемы LSIF1 и контроллеров портов программируемого ввода-вывода (mGPIO)

Контроллеры программируемого ввода-вывода (GPIO) имеют встроенный функционал, позволяющий, через мультиплексирование, подключать к внешним выводам микросхемы либо программно доступные регистры портов ввода-вывода (для выполнения программируемого ввода-вывода информации), либо порты контроллеров внешних интерфейсов микросхемы.

Информация о мультиплексировании программно доступных портов ввода-вывода и портов контроллеров внешних интерфейсов, реализованном в подсистеме LSIF1, приведена в таблице 1.596.

Таблица 1.596 - Схема мультиплексирования LSIF1

Наименование вывода микросхемы	Направление работы вывода микросхемы	Подключение в функциональном режиме	Подключение в GPIO режиме
LSCB0_CLK_IN_0	in	LSCB0/CLK_IN	mGPIO2/GPIN [0]
	out	-	mGPIO2/GPOUT[0]
LSCB0_TXA0	in	-	mGPIO0/GPIN [0]
	out	LSCB0/TXA	mGPIO0/GPOUT[0]
LSCB0_nTXA0	in	-	mGPIO0/GPIN [1]
	out	LSCB0/nTXA	mGPIO0/GPOUT[1]
LSCB0_RXA0	in	LSCB0/RXA	mGPIO0/GPIN [2]
	out	-	mGPIO0/GPOUT[2]
LSCB0_nRXA0	in	LSCB0/nRXA	mGPIO0/GPIN [3]
	out	-	mGPIO0/GPOUT[3]
LSCB0_TX_INH_A0	in	-	mGPIO2/GPIN [1]
	out	LSCB0/TX_INH_A	mGPIO2/GPOUT[1]

Продолжение таблицы 1.596

Наименование вывода микросхемы	Направление работы вывода микросхемы	Подключение в функциональном режиме	Подключение в GPIO режиме
LSCB0_TXB0	in	-	mGPIO0/GPIN [4]
	out	LSCB0/TXB	mGPIO0/GPOUT[4]
LSCB0_nTXB0	in	-	mGPIO0/GPIN [5]
	out	LSCB0/nTXB	mGPIO0/GPOUT[5]
LSCB0_RXB0	in	LSCB0/RXB	mGPIO0/GPIN [6]
	out	-	mGPIO0/GPOUT[6]
LSCB0_nRXB0	in	-	mGPIO0/GPIN [7]
	out	LSCB0/nRXB	mGPIO0/GPOUT[7]
LSCB0_TX_INH_B0	in	-	mGPIO2/GPIN [1]
	out	LSCB0/TX_INH_B	mGPIO2/GPOUT[1]
LSCB1_CLK_IN_1	in	LSCB1/CLK_IN	mGPIO2/GPIN [3]
	out	-	mGPIO2/GPOUT[3]
LSCB1_TXA1	in	-	mGPIO1/GPIN [0]
	out	LSCB1/TXA	mGPIO1/GPOUT[0]
LSCB1_nTXA1	in	-	mGPIO1/GPIN [1]
	out	LSCB1/nTXA	mGPIO1/GPOUT[1]
LSCB1_RXA1	in	LSCB1/RXA	mGPIO1/GPIN [2]
	out	-	mGPIO1/GPOUT[2]
LSCB1_nRXA1	in	LSCB1/nRXA	mGPIO1/GPIN [3]
	out	-	mGPIO1/GPOUT[3]
LSCB1_TX_INH_A1	in	-	mGPIO2/GPIN [4]
	out	LSCB1/TX_INH_A	mGPIO2/GPOUT[4]
LSCB1_TXB1	in	-	mGPIO1/GPIN [4]
	out	LSCB1/TXB	mGPIO1/GPOUT[4]
LSCB1_nTXB1	in	-	mGPIO1/GPIN [5]
	out	LSCB1/nTXB	mGPIO1/GPOUT[5]
LSCB1_RXB1	in	LSCB1/RXB	mGPIO1/GPIN [6]
	out	-	mGPIO1/GPOUT[6]
LSCB1_nRXB1	in	-	mGPIO1/GPIN [7]
	out	LSCB1/nRXB	mGPIO1/GPOUT[7]
LSCB1_TX_INH_B1	in	-	mGPIO2/GPIN [5]
	out	LSCB1/TX_INH_B	mGPIO2/GPOUT[5]
UART0_RXD	in	UART0/RXD	mGPIO2/GPIN [6]
	out	-	mGPIO2/GPOUT[6]
UART0_TXD	in	-	mGPIO2/GPIN [7]
	out	UART0/TXD	mGPIO2/GPOUT[7]

Продолжение таблицы 1.596

Наименование вывода микросхемы	Направление работы вывода микросхемы	Подключение в функциональном режиме	Подключение в GPIO режиме
UART1_RXD	in	UART1/RXD	mGPIO3/GPIN [0]
	out	-	mGPIO3/GPOUT[0]
UART1_TXD	in	-	mGPIO3/GPIN [1]
	out	UART1/TXD	mGPIO3/GPOUT[1]
UART2_RXD	in	UART2/RXD	mGPIO3/GPIN [2]
	out	-	mGPIO3/GPOUT[2]
UART2_TXD	in	-	mGPIO3/GPIN [3]
	out	UART2/TXD	mGPIO3/GPOUT[3]
SPI0_SCLK	in	-	mGPIO3/GPIN [4]
	out	SPI0/SCLK	mGPIO3/GPOUT[4]
SPI0_MOSI	in	-	mGPIO3/GPIN [5]
	out	SPI0/MOSI	mGPIO3/GPOUT[5]
SPI0_MISO	in	SPI0/MISO	mGPIO3/GPIN [6]
	out	-	mGPIO3/GPOUT[6]
SPI1_SCLK	in	-	mGPIO4/GPIN [0]
	out	SPI1/SCLK	mGPIO4/GPOUT[0]
SPI1_MOSI	in	-	mGPIO4/GPIN [1]
	out	SPI1/MOSI	mGPIO4/GPOUT[1]
SPI1_MISO	in	SPI1/MISO	mGPIO4/GPIN [2]
	out	-	mGPIO4/GPOUT[2]
SDIO_CMD	in	gspi_sdio_cmd_i	mGPIO4/GPIN [3]
	out	gspi_sdio_cmd_o	mGPIO4/GPOUT [3]
SDIO_D0	in	gspi_sdio_dat0_i	mGPIO4/GPIN [4]
	out	gspi_sdio_dat0_o	mGPIO4/GPOUT [4]
SDIO_D1	in	gspi_sdio_dat1_i	mGPIO4/GPIN [5]
	out	gspi_sdio_dat1_o	mGPIO4/GPOUT [5]
SDIO_D2	in	gspi_sdio_dat2_i	mGPIO4/GPIN [6]
	out	gspi_sdio_dat2_o	mGPIO4/GPOUT [6]
SDIO_D3	in	gspi_sdio_dat3_i	mGPIO4/GPIN [7]
	out	gspi_sdio_dat3_o	mGPIO4/GPOUT [7]

1.3.12.2 Описание блоков подсистемы LSIF1

1.3.12.2.1 Контроллер портов программируемого ввода-вывода GPIO и mGPIO

Модуль ввода/вывода общего назначения PrimeCell GPIO (GPIO – General Purpose Input/Output) представляет собой периферийное устройство типа «система на кристалле», совместимое с шиной AMBA (Advanced Microcontroller Bus Architecture), разработанное, испытанное и лицензированное компанией ARM.

Модуль PrimeCell GPIO работает в качестве ведомого устройства, подключенного к шине APB (Advanced Peripheral Bus). Модуль PrimeCell GPIO предоставляет восемь программируемых вводов или выводов, управляемых в двух режимах:

программное управление через интерфейс шины APB

аппаратное управление через интерфейс аппаратного управления

Предоставляется возможность создавать порты различной ширины (например, 16, 24, 32 и 40 битов) посредством задания множества значений. Предоставляется интерфейс для работы с прерываниями, обеспечивающий возможность конфигурирования любого количества контактов в качестве источников прерываний. Пользователь может генерировать прерывания, как по уровню, так и по переходу. При сбросе системы линии PrimeCell GPIO по умолчанию переходят в режим вводов. Модуль PrimeCell GPIO связывается с площадками ввода и вывода посредством ввода данных, вывода данных и разрешение вывода для каждой площадки.

На рисунке 1.197 показаны интерфейсы модуля PrimeCell GPIO.

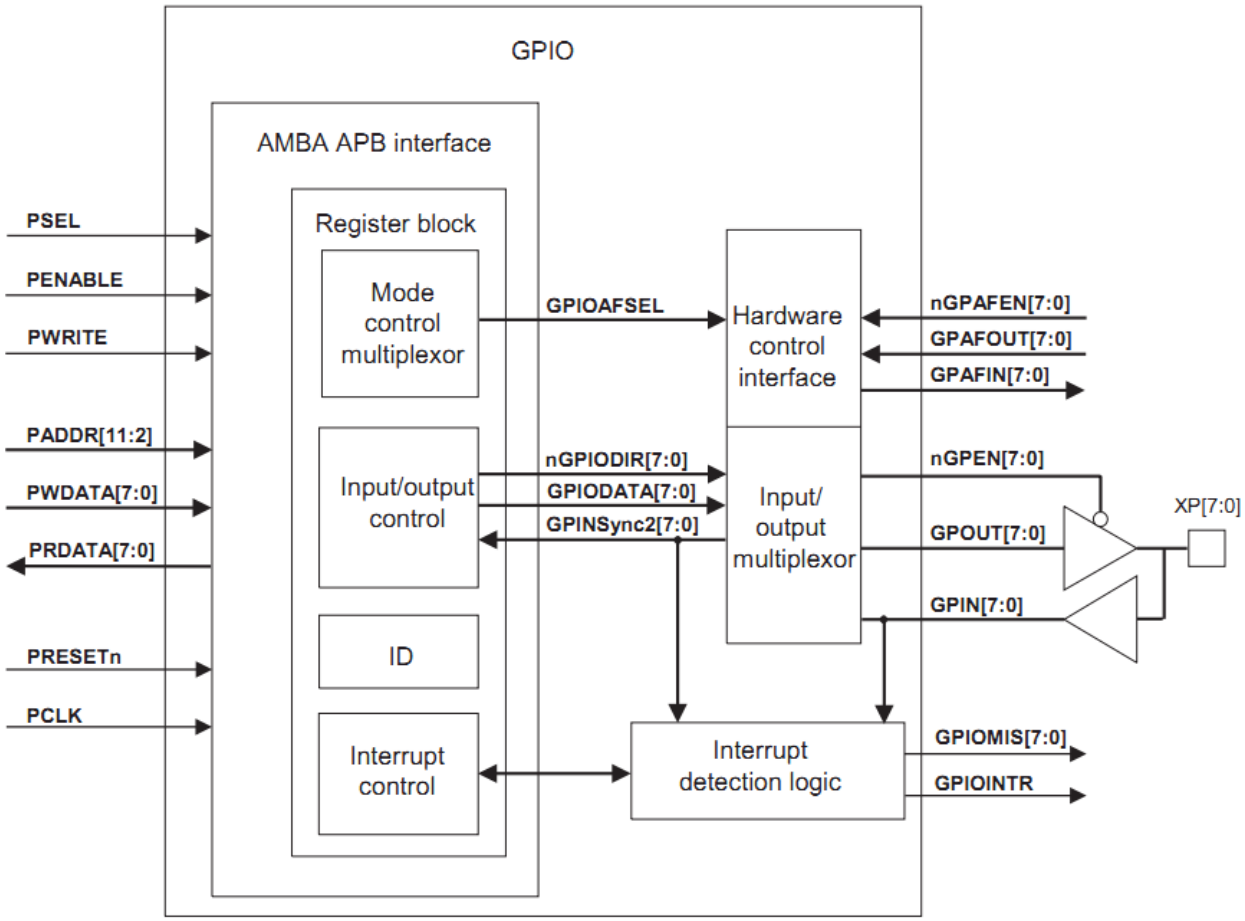


Рисунок 1.197 - Блок-схема модуля PrimeCell GPIO

1.3.12.2.1.1 Основные характеристики модуля PrimeCell GPIO

Модуль PrimeCell GPIO:

- Удовлетворяет требованиям спецификации AMBA (Rev 2.0), что обеспечивает простую интеграцию модуля в систему на кристалле.
- Восемь отдельно программируемых входных/выходных контактов, при сбросе переходящих в режим ввода по умолчанию.

- Расширяемость посредством задания множества значений на 16, 24, 32, 40 или более битов.
- Возможность программируемой генерации прерываний, как по уровню, так и по переходу, на любом количестве контактов.
- Возможность аппаратного управления линиями модуля PrimeCell GPIO для различных конфигураций системы.
- Побитовое маскирование для операций чтения и записи через адресные линии.
- Наличие регистров идентификации, однозначно идентифицирующих модуль PrimeCell GPIO.
-

1.3.12.2.1.2 Обзор функциональных характеристик

1.3.12.2.1.2.1 Общий обзор модуля PrimeCell GPIO

Модуль ввода/вывода общего назначения PrimeCell GPIO представляет собой ведомое устройство с расширенной шинной архитектурой (AMBA - Advanced Microcontroller Bus Architecture), подключаемое к расширенной периферийной шине (APB – Advanced Peripheral Bus). Модуль PrimeCell GPIO предоставляет восемь программируемых вводов или выводов, управляемых в двух режимах:

программное управление через интерфейс шины APB

аппаратное управление через интерфейс аппаратного управления

Центральный процессор считывает и записывает данные, а также управляющую информацию и информацию о состоянии модуля PrimeCell GPIO через интерфейс шины APB.

Периферийное устройство PrimeCell GPIO содержит следующие регистры:

- Регистр направления передачи данных
- Регистр данных
- Регистры управления прерываниями
- Регистр выбора режима
- Регистры идентификации
-

1.3.12.2.1.2.1.1 Регистр направления передачи данных

Регистр направления передачи данных – это восьмибитный регистр, конфигурирующий каждый контакт как входной или выходной.

1.3.12.2.1.2.1.2 Регистр данных

Регистр данных – это восьмибитный регистр, выполняющий следующие функции:

считывание входных данных на линиях PrimeCell GPIO, сконфигурированных как входные

программирование значений на линиях PrimeCell GPIO, сконфигурированных как выходные

Регистр данных имеет 256 адресов в карте распределения памяти. Это позволяет использовать адресную шину [9:2] в качестве дополнительной функции побитового маскирования.

1.3.12.2.1.2.1.3 Регистры управления прерываниями

Модуль PrimeCell GPIO оснащен функцией генерации прерываний. Пользователю предоставляется возможность сконфигурировать любое количество независимых внешних линий PrimeCell GPIO для запуска прерывания через соответствующий бит в семи регистрах прерываний.

1.3.12.2.1.2.1.4 Регистр выбора режима

Управление линиями модуля PrimeCell GPIO может осуществляться программно через шину APB или аппаратно через интерфейс аппаратного управления. Режим линии модуля PrimeCell GPIO выбирается регистром выбора режима.

При разрешении аппаратного управления на конкретной линии будет осуществляться управление и передача данных по этой линии из дополнительного источника.

1.3.12.2.1.2.1.5 Регистры идентификации

Регистры идентификации модуля PrimeCell GPIO содержат информацию о периферийных устройствах и базовой системе ввода/вывода (BIOS), позволяющую однозначно идентифицировать периферийное устройство. Автоматически выбирающая конфигурацию система BIOS может осуществлять сканирование памяти для поиска идентификационного номера BIOS. После нахождения идентификационного номера BIOS может считывать идентификаторы периферийных устройств и автоматически создавать собственную конфигурацию.

1.3.12.2.1.3 Функциональное описание модуля PrimeCell GPIO

1.3.12.2.1.3.1 Логика детектирования прерываний

Модуль PrimeCell GPIO способен генерировать маскированные прерывания по уровню или переходу любой из линий PrimeCell GPIO.

Линия прерывания по вводу/выводу общего назначения (GPIOINTR – General Purpose Input Output Interrupt) указывает контроллеру прерываний на наличие прерывания на одной или нескольких линиях PrimeCell GPIO.

Пользователь может сконфигурировать прерывания таким образом, чтобы их генерация осуществлялась или по изменению уровня линии, или по переднему или заднему фронту сигнала модуля PrimeCell GPIO. Фронт сигнала и уровень, по которым будет сгенерировано прерывание, программируются.

Семь регистров интерфейса AMBA APB, контролирующие различные функции и состояния в цепи запуска прерываний, обеспечивают следующие возможности:

- генерация прерываний по изменению уровня, по одному или двум фронтам сигнала модуля PrimeCell GPIO
- считывание состояния прерываний без учета и с учетом маскирования
- чтение с и запись в регистр разрешения прерывания
- сброс прерывания (только запись)

Каждая линия ввода/вывода имеет соответствующую выходную линию маскированных прерываний. Установка соответствующего бита маски на высокий уровень разрешает прерывание. Линия GPIOINTR – это комбинированная выходная линия прерываний линий состояния маскированных прерываний модуля PrimeCell GPIO. Она указывает контроллеру прерываний, что данный блок запрашивает обслуживание у одного или более источников прерывания.

Доступность как индивидуальных, так и общей линии запроса на прерывание позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйверов устройства, построенных по модульному принципу.

1.3.12.2.1.3.2 Управление режимами

Управление линиями модуля PrimeCell GPIO может осуществляться программно через шину APB или аппаратно через интерфейс аппаратного управления. Выберите режим для каждой линии модуля PrimeCell GPIO с помощью регистра выбора режима (GPIOAFSEL).

Когда разрешен режим программного управления (по умолчанию), направление передачи данных контролируется регистром направления передачей данных. Затем через интерфейс шины APB осуществляется запись и считывание данных.

Когда разрешен режим аппаратного управления, направление передачи данных контролируется через контакты управления направлением передачи данных вспомогательного порта. Аналогично, через данный порт осуществляется запись и чтение данных, но состояния контактов также может быть считано через интерфейс шины APB.

1.3.12.2.1.4 Работа модуля PrimeCell GPIO

1.3.12.2.1.4.1 Сброс настроек интерфейса

При сбросе по включению питания производится сброс всех регистров модуля (низкий уровень). Данное событие запрещает драйверы выходной цепи для линий модуля Primecell GPIO, поэтому контакты переходят в режим входных контактов.

1.3.12.2.1.4.2 Конфигурация интерфейса

При использовании общего сигнала сброса PRESETn, установленного в низкий уровень:

- прерывания в требуемой линии запрещаются сбросом настройки соответствующего бита регистра GPIOIE
- значения всех регистров обнуляются
- входные и выходные контакты переходят в режим входных контактов
- прерывания по внешнему событию маскируются как запрещенные
- прерывания без учета маскирования сбрасываются
- прерывания по фронту сигнала выбираются в качестве источника

1.3.12.2.1.4.3 Рекомендации

Если требуется сгенерировать прерывания по фронту сигнала, необходимо выполнить следующую последовательность инициализации во избежание интерпретирования системой ложных прерываний:

- запрограммируйте надлежащим образом регистр GPIOIBE на детектирование одного или двух фронтов сигнала
- запрограммируйте регистр GPIOIEV, если ранее было задано прерывание по одному фронту сигнала
- запрограммируйте регистр GPIOIS для выбора срабатывания прерывания по фронту сигнала
- задайте три тактовых импульса для очистки канала прерываний
- убедитесь в ходе выполнения данной операции, что шина GPIN[7:0] остается стабильной
- произведите сброс всех прерываний, записав «0xFF» в регистр GPIOIC
- запрограммируйте регистр GPIOIE на разрешение прерываний

1.3.12.2.1.4.4 Работа линий ввода/вывода

Модуль Primecell GPIO включает восемь программируемых линий ввода/вывода. Когда разрешен режим программного управления, регистр данных и регистр направления передачи данных обеспечивают управление данными линиями и обмен данными. При чтении регистр данных содержит текущее состояние контактов модуля Primecell GPIO на предмет их конфигурации – входные или выходные.

При записи регистр данных затрагивает только контакты, сконфигурированные как выходные.

Регистр данных

Для того чтобы независимые программные драйверы могли настроить биты регистра GPIO без затрагивания остальных контактов при единичной операции записи, при операциях чтения/записи используется адресная шина в качестве маски. Регистр данных эффективно охватывает 256 адресов в адресном пространстве. Восемь используемых адресных линий – это адресная шина PADDR [9:2].

Во время записи, если бит адреса, связанный с битом данных, имеет высокий уровень, значение регистра GPIODATA изменяется. Если уровень данного бита низкий, значение остается неизменным. Например:

Запись по адресу GPIODATA + 0x098 = 0b000010011000

PADDR[9:2] = 0b0000100110. Если значение «0xFB» записано по адресу «0x098»:

битам 5 и 1 контактов модуля PrimeCell GPIO присваивается значение «1», а биту 2 – значение «0»

значения остальных битов не изменяются

На рисунке ниже продемонстрировано вышеописанное влияние значения адреса «0x098» на значение данных «0xFB».

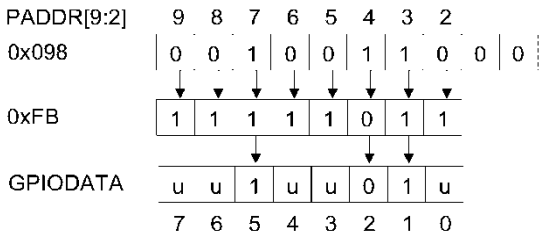


Рисунок 1.198– Запись по адресу «0x098»

Во время чтения, если бит адреса, связанный с данными, имеет высокий уровень, значение считывается, если низкий – значение бита обнуляется. Например:

Чтение с адреса GPIODATA + 0x0C4 = 0b000011000100

PADDR[9:2] = 0b0000110001. При чтении с «0x0C4»:

значения битов 5, 4 и 0 контактов модуля PrimeCell GPIO возвращаются

значения битов 7, 6, 3, 2 и 1 устанавливаются в нуль независимо от их состояния

На рисунке ниже приведен пример считывания данных с адреса «0x0C4» и их передача по линиям PRDATA[7:0].

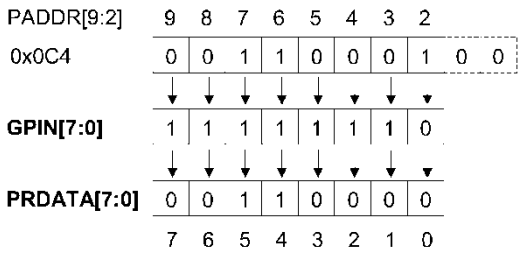


Рисунок 1.199 - Чтение с адреса «0x0C4»

Регистры направления передачи данных

Регистры направления передачи данных работают следующим образом:

значение «0» означает, что соответствующий выходной контакт определен как входной

значение «1» означает, что соответствующий выходной контакт определен как входной

На рисунке ниже представлена стандартная операция записи в регистр направления передачи данных



Рисунок 1.200 – Запись в регистр направления передачи данных

1.3.12.2.1.4.5 Прерывания

Управление прерываниями в модуле PrimeCell GPIO осуществляется группой из семи регистров. Пользователь может выбрать источник прерывания, его полярность и характеристики фронта сигнала. Когда одна или несколько линий модуля PrimeCell GPIO вызывают прерывание, на контроллер прерываний может быть отправлен один выходной сигнал прерывания GPIOINTR и/или независимые сигналы прерывания. При прерываниях по фронту сигнала программное обеспечение должно сбросить прерывание для разрешения любых последующих прерываний. При прерываниях по изменению уровня предполагается, что внешний источник будет поддерживать уровень постоянным для прерывания, подлежащего распознаванию процессором.

Для определения фронта сигнала или события, вызвавших прерывание, необходимы три регистра:

- GPIOIS
- GPIOIBE
- GPIOIEV

Рисунок 1.200 иллюстрирует схему комбинирования битов трех регистров для выбора события источника прерывания.

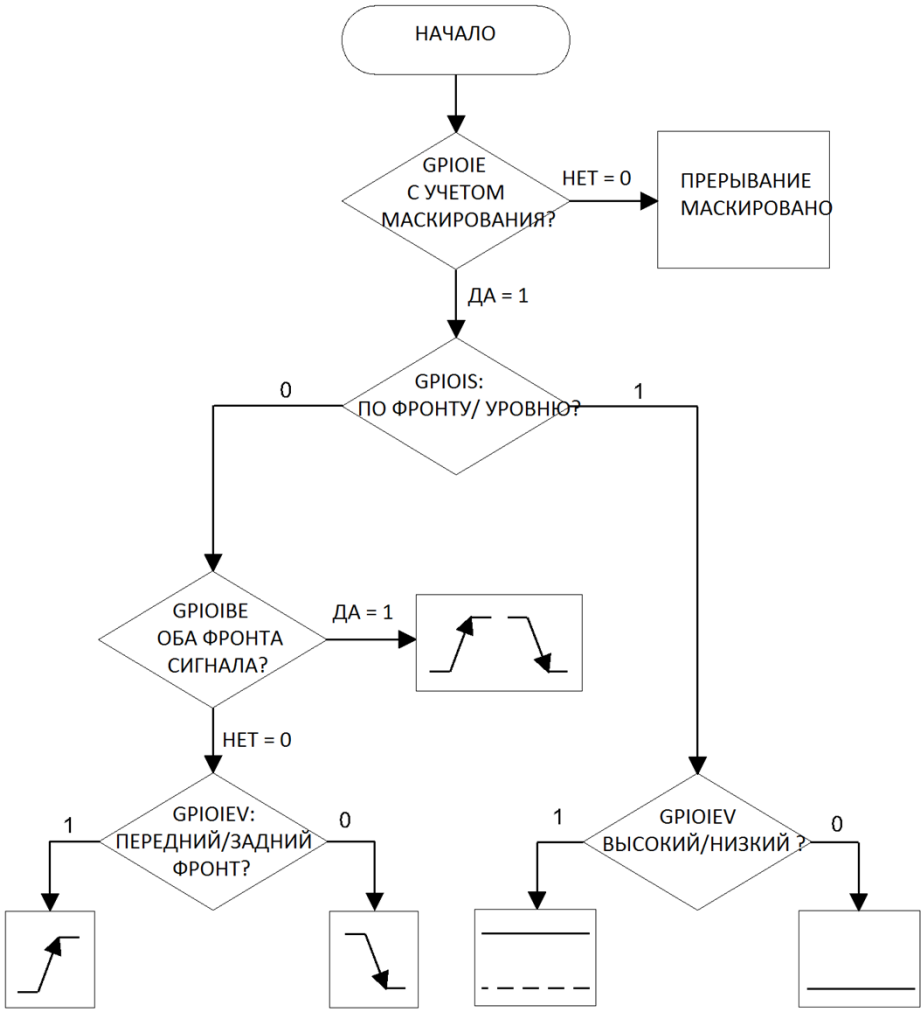


Рисунок 1.200 - Регистры прерываний модуля PrimeCell GPIO

Таблица 1.597 – Срабатывание прерывания по сигналу от контакта 2

Регистр	Пусковой сигнал	7	6	5	4	3	2	1	0
GPIOIS	0 = фронт 1 = уровень	x	x	x	x	x	0	x	x
GPIOIBE	0 = один фронт 1 = оба фронта	x	x	x	x	x	0	x	x
GPIOIEV	Низкий уровень, или отрицательный фронт Высокий уровень, или положительный фронт	x	x	x	x	x	1	x	x

Продолжение таблицы 1.597

Регистр	Пусковой сигнал	7	6	5	4	3	2	1	0
GPIOIE	0 = с учетом маскирования 1 = без учета маскирования	0	0	0	0	0	1	0	0
Регистр	Пусковой сигнал	7	6	5	4	3	2	1	0
GPIOIS	0 = фронт 1 = уровень	x	x	x	x	x	0	x	x
GPIOIBE	0 = один фронт 1 = оба фронта	x	x	x	x	x	0	x	x
GPIOIEV	Низкий уровень, или отрицательный фронт Высокий уровень, или положительный фронт	x	x	x	x	x	1	x	x
GPIOIE	0 = с учетом маскирования 1 = без учета маскирования	0	0	0	0	0	1	0	0

Примечание

Если бит регистра GPIOIE установлен в «0», срабатывание прерывания на соответствующей линии запрещено. В таблице Таблица 1. «x» означает, что значение соответствующего бита неприменимо, что является следствием маскирования бита настройками регистра GPIOIE.

Необходимо запрограммировать регистры управления прерываниями в случае запрета соответствующих прерываний. Запись в регистры управления прерываниями может сгенерировать сложные прерывания, если разрешены соответствующие биты.

1.3.12.2.1.4.6 Управление режимами

Управление линиями PrimeCell GPIO моет осуществляться через интерфейс аппаратного управления. Режим каждой линии модуля определяется регистром выбора режима.

Когда для конкретной линии разрешено аппаратное управление, сигналы от соответствующих контактов данных и направления передачи данных модуля PrimeCell GPIO игнорируются. Сигналы данных и направления передачи данных проходят через данные контакты модуля PrimeCell GPIO, после чего передаются или считываются внешним вспомогательным блоком управления.

На рисунке 1.201 приведена блок-схема мультиплексора выбора режима.

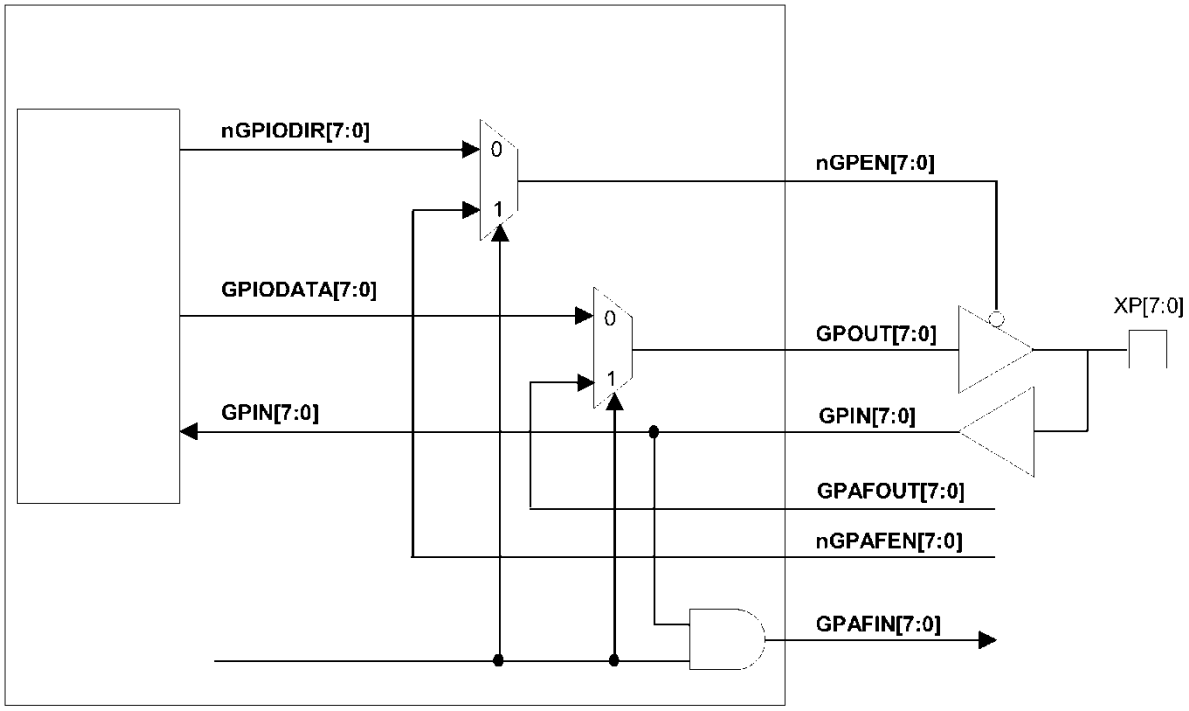


Рисунок 1.201 – Мультиплексор выбора режима

В таблице 1.598 приведены конфигурации площадок для аппаратного и программного обеспечения.

Таблица 1.598– Конфигурации площадок

	Аппаратное управление				Программное управление			
	Вывод		Ввод		Вывод		Ввод	
	7	6	5	4	3	2	1	0
Регистр выбора режима, GPIOAFSEL	1	1	1	1	0	0	0	0
Входной сигнал разрешения аппаратного управления, nGPAFEN	0	0	1	1	x	x	x	x
Входной сигнал данных от аппаратного обеспечения, GPAFOUT	A	B	x	x	x	x	x	x
Выходной сигнал данных от аппаратного обеспечения, GPAFIN	A	B	c	d	0	0	0	0
Выходной сигнал разрешения программного управления, GPIODIR	x	x	x	x	1	1	0	0
Выходной сигнал данных от программного обеспечения, GPIODATA	x	x	x	x	E	F	g	h

Продолжение таблицы 1.598

	Аппаратное управление				Программное управление			
	Вывод		Ввод		Вывод		Ввод	
	7	6	5	4	3	2	1	0
Входной сигнал данных от программного обеспечения, GPIN	A	B	c	d	E	F	g	h
Площадка разрешена, nGPEN	0	0	1	1	0	0	1	1
Вывод площадки, GPOUT	A	B	x	x	E	F	x	x
Ввод площадки, GPIN	A	B	c	d	E	F	g	h
Двунаправленная площадка, XP	A	B	c	d	E	F	g	h

	Аппаратное управление				Программное управление			
	Вывод		Ввод		Вывод		Ввод	
	7	6	5	4	3	2	1	0
Регистр выбора режима, GPIOAFSEL	1	1	1	1	0	0	0	0
Входной сигнал разрешения аппаратного управления, nGPAFEN	0	0	1	1	x	x	x	x
Входной сигнал данных от аппаратного обеспечения, GPAFOUT	A	B	x	x	x	x	x	x
Выходной сигнал данных от аппаратного обеспечения, GPAFIN	A	B	c	d	0	0	0	0
Выходной сигнал разрешения программного управления, GPIODIR	x	x	x	x	1	1	0	0
Выходной сигнал данных от программного обеспечения, GPIODATA	x	x	x	x	E	F	g	h
Входной сигнал данных от программного обеспечения, GPIN	A	B	c	d	E	F	g	h
Площадка разрешена, nGPEN	0	0	1	1	0	0	1	1
Вывод площадки, GPOUT	A	B	x	x	E	F	x	x
Ввод площадки, GPIN	A	B	c	d	E	F	g	h
Двунаправленная площадка, XP	A	B	c	d	E	F	g	h

Примечание

A, B, E, F – источники выходных данных
c, d, g, h – источники входных данных
В режиме аппаратного управления регистр GPIODATA также может считывать значения линий, сконфигурированных как входные.

Таблица 1.598 представляет собой лаконичную таблицу истинных значений для ситуации, когда контакты модуля GPIO PL061 сконфигурированы для аппаратного и программного управления. Значения двунаправленных линий XP[7:0] отображаются в последней строке таблицы, но они имеют различные источники.

Таблица 1.598 приводит более подробное описание, рассматривая каждый режим по отдельности.

1.3.12.2.1.5 Режим аппаратного управления

Четыре столбца слева таблицы 1.598 относятся к режиму аппаратного управления.

Контакты XP[7:4] сконфигурированы для аппаратного управления заданием соответствующих битов в «1» в регистре GPIOAFSEL.

Контакты сконфигурированы как выходные:

- Контакты XP[7:6] сконфигурированы как выходные заданием значения «0» для соответствующих сигналов порта nGPAFEN.
- Значения данных XP[7:6] получены от входных сигналов порта GPAFOUT[7:6] (отображены как (A, B)). Данные значения накапливаются и передаются на контакты XP[7:6]. Данные значения также передаются обратно на выводы порта GPAFIN[7:6] через контакты GPIN[7:6]. Данная функция также позволяет считывать значения XP[7:0] из регистра GPIODATA через интерфейс APB.

Контакты сконфигурированы как входные:

- Контакты XP[5:4] сконфигурированы как входные заданием значения «1» для соответствующих сигналов порта nGPAFEN.
- Значения XP[5:4] извлекаются из внешнего источника и, как упомянуто выше, данные значения (c, d) передаются на GPAFIN[5:4]. Более того, значения XP[7:0] могут быть считаны из регистра GPIODATA через интерфейс APB.

1.3.12.2.1.6 Программное управление

Четыре столбца справа таблицы Таблица 1. относятся к режиму программного управления. В режиме программного управления сигналы порта GPAFIN принудительно устанавливаются в низкий уровень для перехода в энергосберегающий режим.

Контакты XP[3:0] сконфигурированы для программного управления заданием соответствующих битов в «0» в регистре GPIOAFSEL.

Контакты сконфигурированы как выходные:

- Контакты XP[3:2] сконфигурированы как выходные посредством задания значений соответствующих битов в «1» в регистре направления передачи данных GPIODIR.
- Значения данных XP[3:2] получены от битов регистра GPIODATA[3:2] (отображены как (E,F)). Данные значения накапливаются и передаются на контакты XP[3:2]. Данные значения передаются обратно на контакты GPIN[3:2], но не на контакты GPAFIN, так как данный маршрут запрещен в режиме программного управления.
- Более того, значения контакта XP[3:2] могут быть считаны из регистра GPIODATA через интерфейс APB.

Контакты сконфигурированы как входные:

- Контакты XP[1:0] сконфигурированы как входные посредством задания значений соответствующих битов в «0» в регистре направления передачи данных GPIODIR.

- Значения XP[1:0] передаются из внешнего источника (отображены как (g, h)). Более того, значения контакта XP[1:0] могут быть считаны из регистра GPIODATA через интерфейс APB.

1.3.12.2.1.6.1 Программное управление модулем

1.3.12.2.1.6.1.1 О программном управлении модулем

Базовый адрес модуля PrimeCell GPIO не фиксирован и может быть различным в разных системах. Тем не менее, смещение каждого регистра относительно базового адреса фиксировано.

Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:

- адреса при смещениях от «0x424» до «0xFCC» зарезервированы для перспективных расширений возможностей модуля и тестирования
- адреса при смещениях от «+0xFDO» до «+0xFDC» для перспективных расширений возможностей идентификации.

1.3.12.2.1.6.1.2 Обобщенные данные о регистрах модуля PrimeCell GPIO

Регистры модуля PrimeCell GPIO представлены в таблице 1.599.

Таблица 1.599 – Обобщенные данные о регистрах модуля PrimeCell GPIO

Адрес	Тип	Размер, бит	Значение параметра сброса	Наименование	Описание
GPIO base + 0x000 to GPIO base + 0x3FC	Чтение/запись	8	0x00	GPIODATA	Регистр данных модуля PrimeCell GPIO
GPIO base + 0x400	Чтение/запись	8	0x00	GPIODIR	Регистр направления передачи данных модуля PrimeCell GPIO
GPIO base + 0x404	Чтение/запись	8	0x00	GPIOIS	Регистр опознавания прерываний модуля PrimeCell GPIO
GPIO base + 0x408	Чтение/запись	8	0x00	GPIOIBE	Регистр прерываний по двум фронтам сигнала модуля PrimeCell GPIO
GPIO base + 0x40C	Чтение/запись	8	0x00	GPIOIEV	Регистр событий прерываний модуля PrimeCell GPIO
GPIO base + 0x410	Чтение/запись	8	0x00	GPIOIE	Регистр маски прерываний модуля PrimeCell GPIO
GPIO base + 0x414	Чтение	8	0x00	GPIORIS	Регистр состояния прерываний без учета маскирования модуля PrimeCell GPIO
GPIO base + 0x418	Чтение	8	0x00	GPIOMIS	Регистр состояния прерываний с учетом маскирования модуля PrimeCell GPIO
GPIO base + 0x41C	Запись	8	0x00	GPIOIC	Регистр сброса прерываний модуля PrimeCell GPIO
GPIO base + 0x420	Чтение/запись	8	0x00	GPIOAFSEL	Регистр выбора режима модуля PrimeCell GPIO
GPIO base + 0x424-0xFCC	-	-	-	-	Зарезервирован для перспективных расширений возможностей модуля и тестирования

Продолжение таблицы 1.599

Адрес	Тип	Размер, бит	Значение параметра сброса	Наименование	Описание
GPIO base + 0xFD0-0xFDC	-	-	-	-	Зарезервирован для перспективных расширений возможностей идентификации
GPIO base + 0xFE0	Чтение	8	0x61	GPIOPeriphID0	Регистр идентификации периферийных устройств, биты 7:0
GPIO base + 0xFE4	Чтение	8	0x10	GPIOPeriphID1	Регистр идентификации периферийных устройств, биты 15:8
GPIO base + 0xFE8	Чтение	8	0x04	GPIOPeriphID2	Регистр идентификации периферийных устройств, биты 23:16
GPIO base + 0xFEC	Чтение	8	0x00	GPIOPeriphID3	Регистр идентификации периферийных устройств, биты 31:24
GPIO base + 0xFF0	Чтение	8	0x0D	GPIOPCellID0	Регистр идентификации периферийных устройств, биты 7:0
GPIO base + 0xFF4	Чтение	8	0xF0	GPIOPCellID1	Регистр идентификации периферийных устройств, биты 15:8
GPIO base + 0xFF8	Чтение	8	0x05	GPIOPCellID2	Регистр идентификации периферийных устройств, биты 23:16
GPIO base + 0xFFC	Чтение	8	0xB1	GPIOPCellID3	Регистр идентификации периферийных устройств, биты 31:24

1.3.12.2.1.6.1.3 Описание регистров

1.3.12.2.1.6.1.3.1 Регистр данных GPIODATA

Регистр GPIODATA является регистром данных. В режиме программного управления значения, записанные в регистр GPIODATA, передаются на контакты GPOUT, если соответствующие контакты были сконфигурированы в качестве выходных через регистр GPIODIR.

Для записи в GPIODATA соответствующие маскированные биты, переданные через адресную шину PADDR[9:2], должны быть установлены в высокое положение. В противном случае значения битов при записи остаются неизменными.

Аналогично, значения, считанные с данного регистра, определены для каждого бита битом маски, полученным из адреса, используемого для доступа к регистру данных PADDR[9:2]. Биты со значением «1» в маске адреса обуславливают считывание соответствующих данных в регистре GPIODATA, и биты со значением «0» в маске адреса обуславливают считывание соответствующих данных в регистре GPIODATA как «0», независимо от их значений.

При считывании из регистра GPIODATA последнему биту присваивается записанное значение, если соответствующие контакты сконфигурированы как выходные, или возвращается значение соответствующего бита входных контактов GPIN, если они сконфигурированы как входные. При перезагрузке производится сброс значений всех битов.

В таблице Таблица 1. приведено назначение битов регистра GPIODATA. Примеры маскирования адреса при считывании и записи приведены в разделе «Регистр данных».

Таблица 1.600– Регистр GPIODATA

Биты	Наименование	Тип	Назначение
7:0	Регистр данных	Чтение /запись	Ввод данных Вывод данных

1.3.12.2.1.6.1.3.2 Регистр направления передачи данных GPIODIR

Регистр GPIODIR является регистром направления передачи данных. Биты, установленные в высокий уровень в регистре GPIODIR, конфигурируют соответствующий контакт как выходной. При сбросе значения бита контакт переходит в режим входного контакта. При перезагрузке производится сброс значений всех битов. Поэтому контакты GPIO являются входными по умолчанию.

Таблица 1.601 - Регистр GPIODIR

Биты	Наименование	Тип	Назначение
7:0	Регистр направления передачи данных	Чтение/запись	Значения битов заданы, выходные контакты Значения битов сброшены, выходные контакты

1.3.12.2.1.6.1.3.3 Регистр прерываний по двум фронтам сигнала GPIOIBE

Регистр GPIOIBE является регистром прерываний по двум фронтам. Когда соответствующий бит в регистре GPIOIS назначен на детектирование фронта сигнала, биты регистра GPIOIBE, установленные в высокий уровень, конфигурируют соответствующий контакт на детектирование переднего и заднего фронтов сигнала, независимо от значения соответствующего бита в регистре GPIOIEV (регистр событий прерываний). При сбросе значения бита контакт переходит под управление регистра GPIOIEV. При перезагрузке производится сброс значений всех битов.

Таблица 1.602 – Регистр GPIOIBE

Биты	Наименование	Тип	Назначение
7:0	Регистр прерываний по двум фронтам сигнала	Чтение/ запись	Значения битов заданы на срабатывание соответствующего контакта на прерывание при детектировании обоих фронтов сигнала
			Значения битов сброшены, событие генерации прерывания находится под управлением регистра GPIOIEV
			Один фронт сигнала, определяемый соответствующим битом регистра GPIOIEV

1.3.12.2.1.6.1.3.4 Регистр событий прерываний GPIOIEV

Регистр GPIOIEV является регистром событий прерываний. Биты, установленные в высокий уровень в регистре GPIOIEV, конфигурируют соответствующий контакт на детектирование передних фронтов сигналов или высоких уровней, в зависимости от значения соответствующего бита регистра GPIOIS. Сброс значения бита конфигурирует контакт на детектирование задних фронтов сигналов или низких уровней, в зависимости от значения

соответствующего бита в регистре GPIOIS. При перезагрузке производится сброс значений всех битов.

Таблица 1.603 – Регистр GPIOIEV

Биты	Наименование	Тип	Назначение
7:0	Регистр событий прерываний	Чтение/ запись	Значения битов заданы на срабатывание соответствующих контактов на прерывание при детектировании передних фронтов сигналов или высоких уровней
			Значения битов сброшены, срабатывание соответствующих контактов на прерывание при детектировании задних фронтов сигналов или низких уровней

1.3.12.2.1.6.1.3.5 Регистр маски прерываний GPIOIE

Регистр GPIOIE является регистром маски прерываний. Биты регистра GPIOIE, установленные в высокий уровень, позволяют соответствующим контактам запускать отдельные прерывания и объединенную линию GPIOINTR. Сброс значения битов запрещает запуск прерывания для данного контакта. При перезагрузке производится сброс значений всех битов.

Таблица 1.604– Регистр GPIOIE

Биты	Наименование	Тип	Назначение
7:0	Регистр маски прерываний	Чтение/запись	Значения битов заданы, соответствующий контакт не маскирован
			Значения битов сброшены, соответствующий контакт маскирован

1.3.12.2.1.6.1.3.6 Регистр состояния прерываний без учета маскирования GPIORIS

Регистр GPIORIS – это регистр состояния прерываний без учета маскирования. Биты регистра GPIORIS, установленные в высокий уровень, отражают детектированное состояние срабатывания прерываний (без учета маскирования, до маскирования), указывая, что все требования удовлетворены, после чего они переводятся под управление регистра GPIOIE. Нулевые биты указывают на то, что соответствующие входные контакты не инициировали прерывание. Данный регистр предназначен только для чтения, и все биты сбрасываются при перезагрузке системы.

Таблица 1.605 – Регистр GPIORIS

Биты	Наименование	Тип	Назначение
7:0	Регистр состояния прерываний без учета маскирования	Чтение	Отражает детектированное состояние срабатывания прерываний на контактах (без учета маскирования, до маскирования) Значения битов заданы, соответствующие контакты отреагировали надлежащим образом, и требования удовлетворены
			Значения битов сброшены, требования не удовлетворены

1.3.12.2.1.6.1.3.7 Регистр состояния маскированных прерываний GPIOMIS

Регистр GPIOMIS – это регистр состояния маскированных прерываний. Биты регистра GPIOMIS, установленные в высокий уровень, отражают состояние входных линий, инициирующих прерывание. Биты, установленные в низкое положение, указывают, что прерывание не было сгенерировано, или прерывание было маскировано.

Регистр GPIOMIS индицирует состояние прерываний после маскирования. Данный регистр предназначен только для чтения, и все биты сбрасываются при перезагрузке системы.

Содержимое данного регистра становится доступным для внешних устройств через сигналы GPIOMIS[7:0] системы на кристалле.

Таблица 1.606 – Регистр GPIOMIS

Биты	Наименование	Тип	Назначение
7:0	Регистр состояния маскированных прерываний	Чтение	Маскированное значение прерывания, связанное с соответствующим контактом. Значения битов сброшены, прерывание в линии модуля PrimeCell GPIO в неактивном состоянии. Значения битов заданы, в линии модуля PrimeCell GPIO формируется прерывание.

1.3.12.2.1.6.1.3.8 Регистр сброса прерываний GPIOIC

Регистр GPIOIC является регистром сброса прерываний. Присвоение значения «1» биту данного регистра производит сброс соответствующего регистра логики детектирования прерываний по фронту сигнала. Запись «0» не дает результата. Данный регистр предназначен только для записи, и все биты сбрасываются при перезагрузке системы.

Таблица 1.607 – Регистр GPIOIC

Биты	Наименование	Тип	Назначение
7:0	Регистр сброса прерываний	Чтение	Присвоение биту значения «1» производит сброс логики детектирования фронта сигнала. Присвоение биту значения «0» не дает результата.

1.3.12.2.1.6.1.3.9 Регистр выбора режима GPIOAFSEL

Регистр GPIOAFSEL – это регистр выбора режима. Присвоение значения «1» любому биту данного регистра производит выбор режима аппаратного управления для соответствующей линии модуля PrimeCell GPIO. При перезагрузке системы производится сброс всех битов, поэтому линии присвоен режим аппаратного управления по умолчанию.

Таблица 1.608 – Регистр GPIOAFSEL

Биты	Наименование	Тип	Назначение
7:0	Регистр выбора режима	Чтение /запись	Задание значения биту разрешает режим аппаратного управления. Сброс значения бита разрешает режим программного управления.

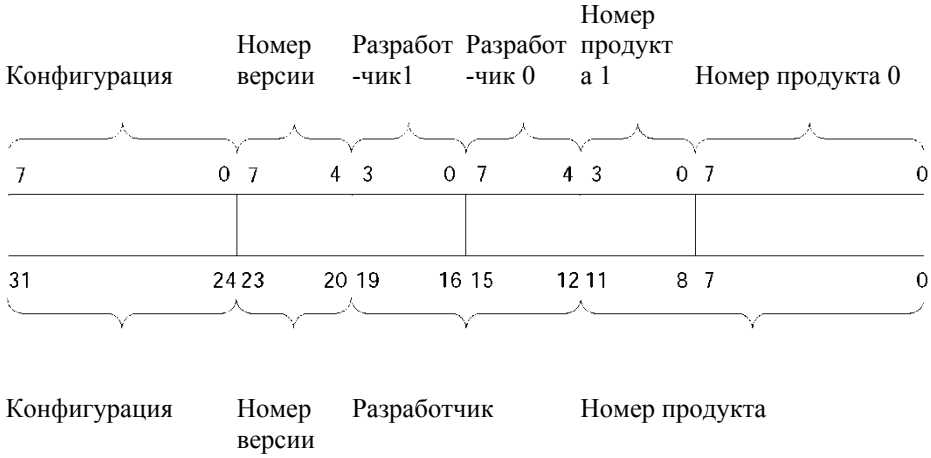
Регистры идентификации периферийных устройств GPIOPeriphID0-3

Регистры GPIOPeriphID0-3 – это четыре восьмибитных регистра, охватывающие адреса от 0xFE0 до 0xFEC. Регистры, в принципе, могут рассматриваться как один 32-битный регистр. Регистры, доступные только для чтения, предоставляют следующие опции для периферийного устройства:

- PartNumber[11:0] Используется для идентификации периферийного устройства. Используется трехразрядный код устройства 0x061.
- Designer ID[19:12] Идентификатор разработчика. Идентификатор компании ARM Ltd – 0x41 (ASCII A).
- Revision[23:20] Номер версии периферийного устройства. Номер версии начинается с 0.
- Configuration[31:24] Конфигурация периферийного устройства. Значением конфигурации является «0».

Назначение битов регистров GPIOPeriphID0-3 приведено на рисунке ниже.

Фактическое назначение разрядов регистра



Номинальное назначение разрядов регистра

Рисунок 1.202 – Назначение разрядов регистров идентификации периферийных устройств

Примечание

При разработке карты распределения памяти необходимо помнить, что объем памяти регистра составляет 4 КБ. Доступ к памяти всех регистров идентификации периферийных устройств должен быть 32-битным; доступ осуществляется согласно инструкциям LDR и STR.

Регистр GPIOPeriphID0

В регистр SSPPCellID0 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.609 – Регистр GPIOPeriphID0

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	PartNumber0	Результатом повторного считывания данных битов будет «0x61»

Регистр GPIOPeriphID1

В регистр SSPPCellID1 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.610– Регистр GPIOPeriphID1

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:4	Designer0	Результатом повторного считывания данных битов будет «0x1»
3:0	PartNumber1	Результатом повторного считывания данных битов будет «0x0»

Регистр GPIOPeriphID2

В регистр SSPPCellID2 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.892– Регистр GPIOPeriphID2

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:4	Revision	Результатом повторного считывания данных битов будет «0x0»
3:0	Designer1	Результатом повторного считывания данных битов будет «0x4»

Регистр GPIOPeriphID3

В регистр SSPPCellID3 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.611– Регистр GPIOPeriphID3

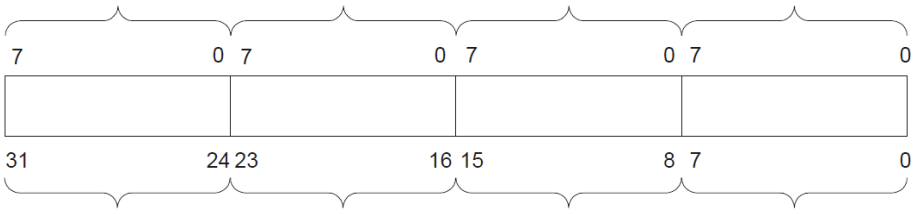
Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	Configuration	Результатом повторного считывания данных битов будет «0x00»

1.3.12.2.1.6.1.3.10 Регистры идентификации модуля PrimeCell GPIOPCellID0-3

Регистры GPIOPCellID0-3 – это четыре восьмибитных регистра, охватывающие адреса от 0xFF0 до 0xFFC. Регистры, в принципе, могут рассматриваться как один 32-битный регистр. Данный регистр используется в качестве стандартной системы идентификации периферийных устройств. Установленным значением регистра GPIOPCellID является «0xB105F00D». Назначение битов регистров GPIOPCellID0-3 приведено на рисунке ниже.

Фактическое назначение разрядов регистра

GPIOPCellID3 GPIOPCellID2 GPIOPCellID1 GPIOPCellID0



Номинальное назначение разрядов регистра

Рисунок1.175 – Назначение разрядов регистра идентификации модуля PrimeCell

Регистр GPIOPCellID0

В регистр GPIOPCellID0 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.612– Регистр GPIOPCellID0

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	GPIOPCellID0	Результатом повторного считывания данных битов будет «0x0D»

Регистр GPIOPCellID1

В регистр GPIOPCellID1 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.613– Регистр GPIOPCellID1

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	GPIOPCellID1	Результатом повторного считывания данных битов будет «0xF0»

Регистр GPIOPCellID2

В регистр GPIOPCellID2 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.614– Регистр GPIOPCellID2

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	GPIOPCellID2	Результатом повторного считывания данных битов будет «0x05»

Регистр GPIOPCellID3

В регистр GPIOPCellID3 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.615– Регистр GPIOPCellID3

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	GPIOPCellID3	Результатом повторного считывания данных битов будет «0xB1»

1.3.12.2.2 Контроллер мультиплексного канала ГОСТ Р 52070

1.3.12.2.2.1 Назначение

Сложный функциональный блок мультиплексного канала обмена (далее по тексту СФ-блок МКО) предназначен для обмена информацией и контроля передачи информации по резервированному (дублированному) мультиплексному каналу передачи данных (МКПД, ЛПИ) в режимах контроллер шины (КШ), оконечное устройство (ОУ), монитор шины (МТ, МШ).

СФ-блок МКО обеспечивает выполнение всех требований к организации обмена информацией и контроля передачи информации, изложенных в ГОСТ Р 52070-2003.

СФ-блок МКО предназначен для выполнения следующих функций:

- предоставление развитых средств поддержки протокола обмена по резервированному (дублированному) мультиплексному каналу передачи данных (МКПД, ЛПИ), согласно ГОСТ Р 52070-2003 (MIL-STD-1553B);
- обеспечение для управляющего вычислителя сопряжения (согласования) с МСП и внутренней памяти.

Для удобства восприятия последующего текста, введем некоторые сокращённые обозначения и определения:

- управляющий вычислитель (УВ), центральный процессор (ЦП) – управляющий вычислитель/центральный процессор предназначен для взаимодействия с СФ-блоком МКО по конфигурационному интерфейсу и интерфейсу доступа к внутренней памяти;
- линия передачи информации (ЛПИ), мультиплексный канал передачи данных (МКПД) – мультиплексный канал передачи данных (ГОСТ Р 52070-2003). МСП не имеет собственных, встроенных, приёмопередатчиков, поэтому для сопряжения с ЛПИ необходимо использовать внешние приёмопередающие устройства (приёмопередатчики, ППУ).
- машина связная последовательной шины данных (МСП) – часть функциональной логики СФ-блока МКО, отвечающая за обмен и контроль информации передаваемой и принимаемой по МКПД, также за работу в режимах КШ, ОУ, МШ, МТ;
-

1.3.12.2.2.2 Характерные особенности

1.3.12.2.2.2.1 Особенности структуры

СФ-блок МКО содержит:

- МСП:
  - шифратор информации ЛПИ - кодер;
  - сдвоенный дешифратор информации ЛПИ - декодер;
  - полную многопротокольную логику, обеспечивающую режимы работы контроллер шины (КШ), оконечное устройство (ОУ), монитор шины (МТ, МШ);
  - встроенное, разделяемое с процессором УВ, ОЗУ (РОЗУ) объёмом 4К 16-ти разрядных слов;

- логику прерывания управляющего вычислителя;
- встроенный счетчик временной метки с программируемым (выбираемым) разрешением между 2, 4, 8, 16, 32, 64 мкс на младший значащий разряд.

- 3-х портовый узел сопряжения;
- Встроенное ОЗУ (РОЗУ) объемом 4К 16-ти разрядных слов.
-

1.3.12.2.2.2.2 Возможности МСП.

МСП предоставляет шины адреса и данных, набор входных/выходных управляющих сигналов для организации взаимодействия с УВ конфигурационному интерфейсу, обеспечивает для УВ следующие возможности:

работу со встроенной памятью (ОЗУ) объёмом до 4К 16-ти разрядных слов;

программное конфигурирование (настройка) МСП;

выбор (при конфигурировании) способа буферизации СД сообщения для режима ОУ:

а) одиночная буферизация;

б) двойная буферизация сообщений «приема»;

в) кольцевая буферизация;

выбор разновидности режима МТ:

а) «монитор слов»;

б) «монитор сообщений»;

в) комбинированный режим «ОУ/монитор сообщений».

работу КШ в режиме «возобновление передачи сообщения» (ВПС), - автоматически повторяется пересылка сообщения в случае неудачи первой попытки (истечение времени ожидания ОС, обнаружение ошибки формата сообщения и т.п.);

формирование для КШ в ОЗУ последовательности сообщений, образующей т.н. «кадр сообщений», с однократным/автоматически повторяемым исполнением кадра сообщений (последовательной отработкой сообщений, составляющих кадр);

программируемый для каждого сообщения КШ интервал времени от начала пересылки данного сообщения до начала пересылки следующего сообщения (промежуток времени для сообщения);

автоматическое определение недопустимости командного слова для режима ОУ;

три программно задаваемых режима выдачи сигнала запроса прерывания УВ (сигнал msp_int):

а) импульсный;

б) выходной уровень, запрос прерывания ‘сбрасывается’ под управлением программы УВ, после записи лог. «1» в соответствующий разряд регистра запуска/сброса;

в) выходной уровень, запрос прерывания ‘сбрасывается’ после чтения программой УВ регистра состояния прерываний.

1.3.12.2.2.2.3 Источники тактовой частоты

Для работы СФ-блока МКО необходимы два источника тактовой частоты: 16 МГц и 100 МГц. Допускается использовать асинхронные источники тактовой частоты.

1.3.12.2.2.2.4 Приёмопередатчики

Кодер и декодеры (кодек) МСП взаимодействуют с внешними приёмопередатчиками согласно требованиям интерфейсов типа ‘Smiths’ или ‘Harris’.

-

1.3.12.2.2.3 Описание структурной схемы

Обобщенная структурная схема СФ-блока МКО приведена на рисунке 1.204.

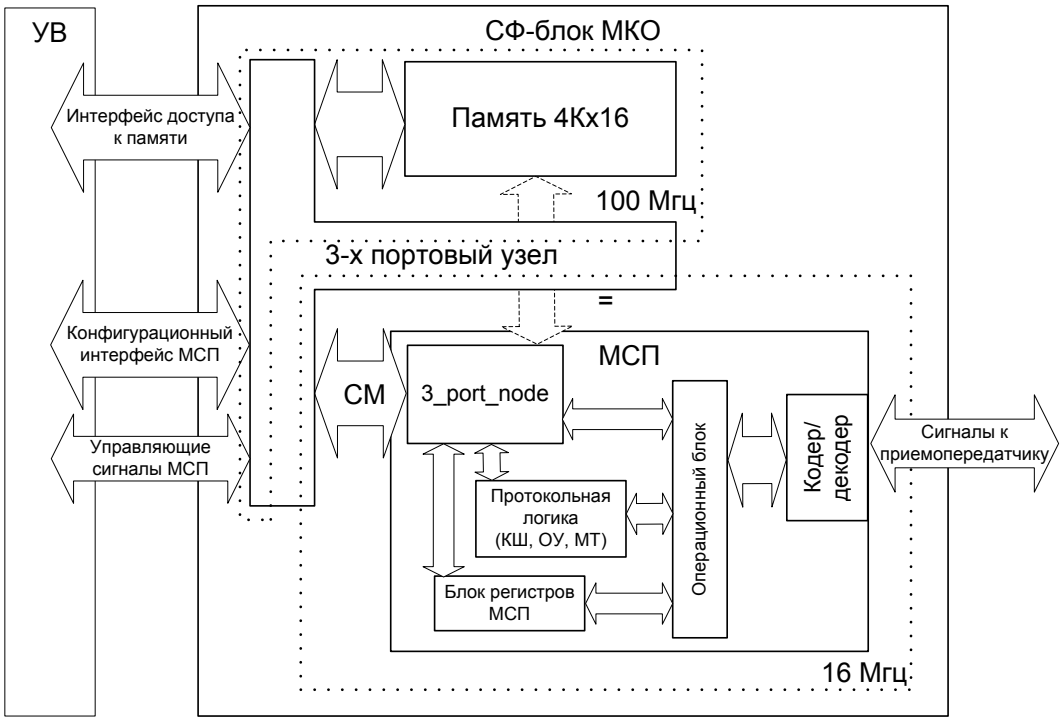


Рисунок 1.204 - Обобщенная структурная схема СФ-блока МКО

В состав СФ-блока МКО входят:
- МСП;
- 3-х портовый узел;
- внутреннее ОЗУ объемом 4К 16 разрядных слов.

В состав МСП входят следующие функциональные узлы:

- трёхпортовый узел связи (3port_node);
- протокольная логика (многопротокольная логика) - управляющие автоматы логики протокола обмена по ЛПИ для режимов работы КШ, ОУ, «монитор слов», «монитор сообщений»;
- операционный блок;
- блок конфигурационных/запускающих регистров и регистров состояния.

Трёхпортовый узел связи (в составе МСП) обеспечивает сопряжение и двунаправленную пересылку данных между следующими сущностями:

- внешней системной магистралью СМ;
- интерфейсной магистралью внутреннего ОЗУ;
- внутренними магистралями МСП.

Протокольная логика содержит управляющие автоматы «логики протокола обмена по ЛПИ» режимов работы МСП (КШ/ ОУ/«монитор сообщений»/«монитор слов»).

Операционный блок содержит:

- инструментальные средства, необходимые для производства анализа и подсчёта различных событий ЛПИ, адресных вычислений и логических действий;
- средства для отслеживания и формирования различных интервалов времени;
- устройства формирования «векторов состояния», в том числе слово состояния блока сообщении КШ/ОУ/«монитор сообщений» и ОС;
- резервированные кодер и декодер ЛПИ.

3-х портовый узел (в составе СФ-блока МКО) обеспечивает сопряжение, синхронизацию между разными частотными доменами и пересылку данных между следующими сущностями:

- конфигурационный интрефейс МСП;
- интерфейс доступа к памяти;

- системная магистраль (СМ) МСП;
- двунаправленный поток данных ОЗУ.

1.3.12.2.2.4 Функциональное описание МСП

1.3.12.2.2.4.1 Внутренние регистры МСП

Кратко опишем назначение наиболее часто используемых регистров.

Регистр маскирования прерывания используется для разрешения/запрещения выдачи сигнала запроса прерывания УВ (сигнал msp_int) для различных событий, соответствующим разрядам регистра.

Конфигурационные регистры №1 и №2 используются для выбора режимов работы, а также для программного задания значений разрядов (признаков) ответного слова ОУ, назначения «активной области ОЗУ» (А или Б), задания останова работы КШ в случае обнаружения ошибки, выбора режима управления памятью ОУ, управление «счетчиком метки времени».

Регистр запуска/сброса используется для исполнения ‘програмных команд’, таких как программный сброс, старт КШ/МТ, сброс подсистемы запроса прерывания УВ, сброс «счетчика метки времени», останова режима автоповтора кадра КШ, либо останов работы КШ после отработки текущего сообщения/кадра.

Регистр указателя на стек команд позволяет УВ определить для КШ, ОУ или «монитора сообщений» адрес описателя блока сообщения в стеке команд для текущего или последнего отработанного сообщения.

Регистр счетчика метки времени – определяет значение 16-ти разрядного счетчика. Текущее значение регистра счетчика метки времени записывается в стек команд (описатель блока сообщения) во время последовательностей операций (действий) «начало отработки сообщения» (Start-of-Message - SOM) и «завершение отработки сообщения» (End-of-Message - EOM) в режимах КШ, ОУ и «монитор сообщений» .

Регистр состояния прерывания содержит разряды, соответствующие разрядам регистра маскирования прерываний; чтение значения регистра позволяет УВ определить событие – источник запроса прерывания УВ.

Конфигурационные регистры №3, №4 и №5 используются в «расширенных возможностях конфигурирования» МСП.

К «расширенным возможностям конфигурирования» относятся:

- для режима КШ это полные слова «слово управления сообщением КШ» и «слово состояния блока сообщения», дополнительные функции «останов по ошибке» и «останов по особому состоянию», автоповтор отработки кадра сообщений, задания промежутка времени для сообщения, автоматическое «возобновление передачи сообщения» (ВПС), полное маскирование разрядов ОС, возможность генерации запроса прерывания УВ по завершении отработки любого избранного сообщения и др.;
- для режима ОУ это полное «слово состояния блока сообщения», совмещённый режим «ОУ/монитор сообщений», альтернативное (посредством программного обеспечения УВ) задание значения признаков ОС и др.;
- для режима МТ – возможность использовать совмещённый режим «ОУ/монитор сообщений» и др.
- Регистр указателя буфера данных ОУ/МТ позволяет программно оценивать состояние «буфера данных».
-

Примечание 1. Запись значений поля из последовательно расположенных разрядов (массива разрядов) будем производить в следующем формате записи значения поля разрядов –

{<длина>’}<система счисления><значение>; b – знак двоичной системы счисления, x – произвольное значение.

Например, 2’b01 – поле из двух разрядов, значение старшего разряда - лог. «0», младшего разряда - лог. «1».

Формат записи адресов ОЗУ и значений слов ОЗУ и регистров – как правило, четыре цифры в шестнадцатеричной системе счисления, без указания системы счисления.

Примечание 2. Далее в тексте документа при обозначении разрядов регистров символом ‘-’ обозначаются зарезервированные разряды, значения которых при записи должны задаваться значением лог. «0».

Таблица 1.616 - Внутренние регистры МСП

Адрес (двоичн.)	Адрес (шестн.)	Наименование	Сокращенное обозначение	Вид доступа
00000	00	Регистр маскирования прерывания	imr [15..0]	чт/зп
00001	01	Первый конфигурационный регистр	cfg1_[15..0]	чт/зп
00010	02	Второй конфигурационный регистр	cfg2_[15..0]	чт/зп
00011	03	Регистр запуска/сброса	srr [15..0]	запись
00011	03	Регистр указателя на стек команд	csp [15..0]	чтение
00100	04	Регистр слова управления сообщением КШ	bc_cw [15..0]	чт/зп
00100	04	Регистр слова управления подадресом ОУ	rt_cw [15..0]	чт/зп
00101	05	Регистр счетчика метки времени	tt_reg [15..0]	чт/зп
00110	06	Регистр состояния прерывания	isr [15..0]	чт/зп
00111	07	Третий конфигурационный регистр	cfg3_[15..0]	чт/зп
01000	08	Четвертый конфигурационный регистр	cfg4_[15..0]	чт/зп
01001	09	Пятый конфигурационный регистр	cfg5_[15..0]	чт/зп
01010	0A	Регистр указателя буфера данных ОУ/МТ	rm_dsa [15..0]	чт/зп
01011	0B	Регистр остатка времени кадра КШ	bc_ftr [15..0]	чтение
01100	0C	Регистр остатка времени текущего сообщения КШ	bc_trnm [15..0]	чтение
01101	0D	Регистр длительности кадра КШ/ Регистр командного слова сообщения ОУ/ Регистр слова срабатывания («монитор слов»)	bcft_rtlc_mttw [15..0]	чт/зп
01110	0E	Регистр ответного слова ОУ	rt_sw [15..0]	чтение
01111	0F	Регистр слова ВСК	rt_bit_w [15..0]	чтение
11000	18	Шестой конфигурационный регистр	cfg6_[15..0]	чт/зп

1.3.12.2.2.4.1.1 Регистр маскирования прерывания - imr

Адрес регистра 00000; чт/зп.

Регистр маскирования прерывания (imr) предназначен для разрешения прерывания (разрешение формирования выходного сигнала msp_int) при происшествии 14-ти различных событий/условий, соответствующих установленным в лог. «1» разрядам данного регистра.

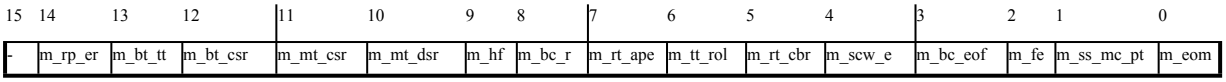
В состоянии «нерасширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») используются только разряды с 7 по 0 регистра imr. Разряды с 14 по 8 регистра не задействованы.

Если МСП в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») и разряд «разрешение расширенных возможностей прерывания» (ei, 15-ый разряд регистра cfg2_) установлен в лог. «1», то все 14 событий/условий возможны (разряды с 13 по 0).

Внешний сигнал прерывания (msp_int) будет формироваться в виде импульса или в виде уровня, как это определено разрядом l_p_ir, 3-им разрядом регистра cfg2_, всякий раз при возникновении обстоятельств, порождающих прерывание, однако, это будет происходить только при наличии лог. «1» в соответствующем разряде регистра маскирования прерывания (imr).

Соответствующие разряды регистра состояния прерывания (isr) будут устанавливаться в значение лог. «1» в случае возникновения соответствующих событий при одном следующих условий:

- МСП находится в состоянии «нерасширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «0») и соответствующий разряд регистра маскирования прерывания в лог. «1» (разряды с 7 по 0);
- МСП находится в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1») и разряд «разрешение расширенных возможностей прерывания» в лог. «0» (ei, 15-ый разряд регистра cfg2), и соответствующий разряд регистра маскирования прерывания в лог. «1» (разряды с 7 по 0);
- МСП находится в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»), и разряд «разрешение расширенных возможностей прерывания» в лог. «1» (ei, 15-ый разряд регистра cfg2), вне зависимости от значения разрядов регистра маскирования прерывания (разряды с 13 по 0).
-
- Структура регистра imr[15..0]:



Разряды 15, 14 – зарезервированы, при записи должны задаваться значением лог. «0».

Наименования разрядов регистра imr:

m_rp_er (Ram Parity Error): Ошибка четности при доступе к ОЗУ;
m_bt_tt (BC/RT transmiter Timeout): затягивание работы передатчика;
m_bt_csr (BC/RT Command Stack Rollover): ‘прокручивание’ стека команд КШ/ОУ;
m_mt_csr (MT Command Stack Rollover): ‘прокручивание’ стека команд монитора сообщений;
m_mt_dsr (MT Data Stack Rollover): ‘прокручивание’ буфера данных МТ;
m_hf (Handshake Failure): истечение времени ожидания подтверждения (квитирования);
m_bc_r (BC Retry): ВПС в режиме КШ;
m_rt_ape (RT Address parity Error): ошибка контроля по четности адреса ОУ;
m_tt_rol (Time Tag Rollover): переполнение счетчика метки времени;
m_rt_cbr (RT Circular Buffer Rollover): ‘прокручивание’ кольцевого буфера данных ОУ;
m_scw_e (RT/BC Selective Control Word EOM): завершение обработки избранного сообщения для режимов КШ/ОУ;
m_bc_eof (BC End Of Frame): завершение обработки кадра сообщений;
m_fe (Format Error): ошибка формата сообщения;
m_ss_mc_pt (BC Status Set/ RT Mode Code/ MT Pattern Trigger):
- для режима КШ: особое состояние ОУ;
- для режима ОУ: принята избранная команда управления;
- для режима «монитор слов»: принятое КС совпало со словом срабатывания;
m_eom (End Of Message): завершение отработки сообщения.

Функциональное описание разрядов регистра imr:

m_rp_er (Ram Parity Error) - разряд ошибка четности при доступе к ОЗУ.

Этот разряд применим только в «режиме расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») и если разряд «разрешения проверки на четность» (rp_en) регистра cfg2_ установлен в лог. «1»; m_rp_er разрешает выдавать

(активизировать) запрос прерывания в результате ошибки «четности», происходящей при доступе к ОЗУ в режиме чтения.

Данный разряд не задействован в МСП.

m_bt_tt (BC/RT transmiter Timeout) - разряд затягивание работы передатчика.

Если разряд установлен в лог. «1», то разрешается прерывание после срабатывания сторожевого устройства, отслеживающего истечение заданного времени непрерывной работы передатчика МСП. Это происходит, если попытка кодирования данных для передачи затягивается на время, больше чем 668 мкс. Прерывание вырабатывается только в режимах КШ или ОУ.

m_bt_csr (BC/RT Command Stack Rollover) - разряд ‘прокручивание’ стека команд КШ/ОУ.

Если разряд установлен в лог. «1», разрешает прерывание после ‘прокручивания’ командного стека КШ или ОУ.

m_mt_csr (MT Command Stack Rollover) - разряд ‘прокручивание’ стека команд монитора сообщений.

Если разряд установлен в лог. «1», то разрешается прерывание после ‘прокручивания’ стека команд монитора сообщений. Применяется как для режима «монитора сообщений», так и для комбинированного режима работы «ОУ/ монитор сообщений». Размер командного стека можно выбирать среди значений 256 (64 сообщения), 1024, 4096 и 16384 слов (4096 сообщений) посредством разрядов 12 и 11 регистра **cfg3_**. Это прерывание способно вырабатываться только в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»).

m_mt_dsr (MT Data Stack Rollover) - разряд ‘прокручивание’ буфера данных МТ.

Если разряд установлен в лог. «1», то разрешается прерывание после ‘прокручивания’ буфера данных в режиме «монитора слов» или «монитор сообщений». Размер буфера данных выбирается среди 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536 слов посредством разрядов 10, 9, и 8 регистра **cfg3_**.

m_hf (Handshake Failure) - разряд истечение времени ожидания подтверждения (квитирования).

Данный разряд не задействован в МСП.

m_bc_r (BC Retry) - разряд ВПС в режиме КШ.

Если разряд установлен в лог. «1», то разрешается прерывание после происшествия ВПС в режиме КШ. Если разряд установлен в лог. «1», то прерывание будет происходить вне зависимости от того, успешно ли завершилась попытка ВПС или нет.

m_rt_ape (RT Address parity Error) - разряд ошибка контроля по четности адреса ОУ.

Если разряд установлен в лог. «1», то разрешается прерывание в случае обнаружения ошибки контроля по четности в адресе ОУ; т.е., когда сумма лог. значений входных сигналов msp_rtad [4:0] и msp_rtap имеет четное значение.

m_tt_rol (Time Tag Rollover) - разряд переполнение счетчика метки времени.

Если разряд установлен в лог. «1», то разрешается прерывание в случае перехода значением 16-ти разрядного регистра счетчика метки времени из FFFF в 0000.

m_rt_cbr (RT Circular Buffer Rollover) - разряд ‘прокручивание’ кольцевого буфера данных ОУ.

Если этот разряд установлен в лог. «1», и МСП находится в режиме «расширенные возможности управления памятью ОУ» (разряд 1 регистра **cfg2_** установлен в лог. «1»), и разряд/разряды «разрешения прерывания по переполнению кольцевого буфера сообщений «передачи»/«приёма»/«группового приема»» в слове управления подадресом ОУ для соответствующего подадреса установлен в лог. «1», то запрос прерывания будет порождаться по событию ‘прокручивание’ (переполнение).

‘Прокручивание’ - ситуации, когда соответствующий указатель адреса, взятый из поисковой таблицы, после попытки пересечь максимальную (и кратную заданному размеру буфера) по значения адреса границу кольцевого буфера возвращается к значению, в котором

переходят из лог. «1» в лог. «0» младшие разряды, соответствующие заданному размеру данного буфера.

В общем случае, ‘прокручивание’ не обязательно приводит к начальному значению указателя, который может содержать значение любого адреса в границах выделяемой под буфер области памяти. Например, начальное значение указателя равное 9a77h задаёт выделение под буфер, скажем в 512 слов, области ОЗУ с адресами от 9a00h до 9affh.

Если разряд «переписывать недостоверные данные» (разряд 11 регистра cfg2_) находится в лог. «0», то запрос на прерывание будет порождаться непосредственно сразу после того, как последняя, прилегающая к максимальной (и кратной заданному размеру буфера) по значения адреса границе кольцевого буфера ячейка памяти была подвержена чтению или записи.

Если разряд «переписывать недостоверные данные» (разряд 11 регистра cfg2_) установлен в лог. «1»,то запросы прерывания будут происходить в конце отработки сообщений «передачи» или достоверных сообщений «приема», в которых слово, прилегающее к максимальной по значения адреса границе кольцевого буфера в адресном пространстве ОЗУ, было прочитано/записано. Для сообщений «приема» запросы прерывания не будут порождаться, если разряд 11 регистра **cfg2_** установлен и произошла ошибка сообщения, даже в случае доступа к последнему (с граничным местоположением) слову буфера.

m_scw_e (BC MSG/RT Subaddress Control Word EOM) - разряд завершение обработки избранного сообщения КШ/ОУ.

Для режима КШ, если этот разряд установлен в лог. «1», МСП находится в состоянии «расширенных возможностей конфигурирования», и разряд «разрешение полного слова управления сообщением КШ» (разряд 12 регистра **cfg4_**) установлен в лог. «1», и «разрешение прерывания по EOM» (4-ый разряд соответствующего слова управления сообщением КШ) установлен в лог. «1», то запрос прерывания будет порожден в конце текущего сообщения.

Для режима ОУ, если этот разряд установлен, и МСП находится в режиме «расширенное управление памятью ОУ», и разряд «разрешение прерывания по EOM» установлен в лог. «1» в слове «управления подадресом ОУ» для соответствующего (передачи/приёма/групповой) подадреса, то прерывание будет выработано в конце текущего сообщения.

m_bc_eof (BC End Of Frame) - разряд завершение обработки кадра сообщений:

Если разряд установлен в лог. «1», то прерывание будет порождено после окончания обработки последнего сообщения кадра, когда очередное увеличение на единицу счетчика сообщений области А/Б привело его к состоянию максимально возможного значения - FFFF(шестн.). Если отработка кадра КШ прекращена прежде достижения счетчиком сообщений значения FFFF, то прерывания **bc_eof** не произойдет. Заметим, что если задан останов работы после окончания текущего сообщения (в состоянии «расширенных возможностей конфигурирования» установлен в лог. «1» один из разрядов 10, 12 регистра **cfg1_**) и событие произошло, то кадр завершается прежде достижения счетчиком сообщений значения FFFF(шестн.).

m_fe (Format Error) - разряд ошибка формата сообщения.

Если этот разряд установлен в лог. «1», то это приведет к запросу прерывания при любой из следующих ситуаций:

- «ошибка проверочного возврата»: проверка ‘возвратившегося’ слова производится для каждого слова, переданном МСП (режимы ОУ и КШ), для каждого сообщения. ‘Возврат’ передаваемого в ЛПИ слова осуществляется: а) в режиме внутреннего самотестирования (разряд olst слова управления сообщением КШ в лог. «1») - непосредственной пересылкой передаваемого слова от выхода кодера на вход декодера; б) вне режима внутреннего самотестирования (разряд olst слова управления сообщением КШ в лог. «0») – путем приёма ППУ передаваемого слова и последующим его декодированием. Если принятая версия одного или более слов расшифрована декодером как недостоверная и/или принятая версия последнего слова, переданного в сообщении, не равна исходной (посланной) версии, то в этом случае фиксируется «ошибка проверочного возврата».

- «ошибка в сообщении»: принятое сообщение содержит нарушения критериев достоверности принятой информации по ГОСТ Р 52070-2003 – недостоверное слово (тип синхросигнала, количество разрядов, контроль по четности и т.п), неверное количество слов, некорректный адрес ОУ в ОС. Для режима ОУ ситуация «ошибка формата сообщения» не учитывает наличие ошибки в принятом КС, т.к. если обнаруживается недостоверное командное слово, то сообщение полностью игнорируется.
- «Истечение времени ожидания ОС»: В режимах КШ или «монитора сообщений» это происходит, когда ОУ либо не откликнулось на сообщение, либо откликнулось после истечения программно устанавливаемой длительности «времени ожидания ОС». Это время для КШ и ОУ (при передаче «ОУ-ОУ») может быть выбрано посредством разрядов 10 и 9 регистра **cfg5_**. В режиме принимающего ОУ истечение времени ожидания ОС происходит, если в передаче «ОУ-ОУ» передающее ОУ не отозвалось ОС в установленный для этого срок.
- Пауза перед выдачей ОС меньше 4 мкс. и установлен режим «расширенных возможностей конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «1») и «разрешение проверки минимальной паузы перед выдачей ОС» (gce, 8-ой разряд регистра **cfg5_** в лог. «1»). Согласно ГОСТ Р 52070-2003, 4 мкс. - это промежуток времени между серединами зон разряда контроля по четности последнего КС/СД и синхроимпульса ОС.

m_ss_mc_pt (BC Status Set/ RT Mode Code/ MT Pattern Trigger), смысл разряда зависит от режима работы:

- для режима КШ – ‘особое состояние ОУ’ или ОС содержит неправильный адрес ОУ в адресном поле;

- для режима ОУ – принята избранная команда управления;

- для режима «монитор слов» – принятое КС совпало со словом срабатывания (заданным в регистре слова срабатывания).

В режиме КШ - ОС, полученное от отвечающего абонента, либо содержит неправильный адрес ОУ в адресном поле, либо ‘особое состояние ОУ’ т.е. один из одиннадцати признаков/разрядов ОС содержит (имеет) неожидаемое значение. Ожидаемыми значениями для этих одиннадцати признаков являются лог. «0» за следующими исключениями:

1. Признаки ОС установленные в лог. «1» замаскированы соответствующими разрядами слова управления сообщением КШ.

2. В некоторых состояниях конфигурации МСП ожидаемым значением для признака ОС «Принята групповая команда» становится лог. «1», а не лог. «0». Это последнее исключение верно для каждого из следующих состояний конфигурации МСП:

1) МСП находится в состоянии «нерасширенные возможности конфигурирования» (15-ый разряд (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «0»);

2) МСП находится в состоянии «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»),

и

разряд bm_ex регистра **cfg4_** установлен в лог. «0»,

и

разряд «маскирование групповых сообщений» в слове управления сообщением КШ установлен в лог. «1».

В режиме ОУ этот запрос прерывания УВ будет формироваться при завершении приёма избранной (заданной) КУ. Оно может происходить только в состоянии «расширенных возможностей конфигурирования» МСП (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1») и, если разрешена расширенная обработка КУ (emch, 0-ой разряд регистра **cfg3_**, установлен в лог. «1»). Если эти два разряда установлены, то прерывание для любой из КУ может быть разрешено программированием в лог. «1» предназначенного для этой команды разряда в соответствующем слове из области памяти с адресами 0108-010F (таблица прерываний для

избранных КУ). Получение избранной (заданной) КУ будет вызывать со стороны МСП порождение запроса прерывания УВ в конце обработки КУ.

В режиме «монитор слов» этот запрос прерывания УВ будет формироваться только в состоянии «расширенных возможностей конфигурирования» МСП (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»), когда принятое КС совпало со словом срабатывания (заданным в регистре слова срабатывания). Принятое достоверное командное слово должно поразрядно совпасть со значением, заданным программно в регистре слова срабатывания.

m_eom (End Of Message) - разряд завершение отработки сообщения.

Если разряд установлен в лог. «1», то будет вызываться запрос прерывания в режимах КШ, ОУ и «монитор сообщений» при завершении обработки каждого сообщения (вне зависимости от достоверности).

1.3.12.2.2.4.1.2 Первый конфигурационный регистр - cfg1_

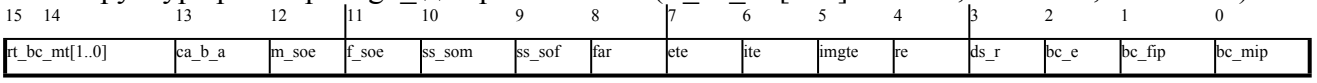
Адрес регистра 00001; чт/зп.

Смысл разрядов регистра зависит от числового значения его собственных двух старших разрядов rt_bc_mt[1..0] и от состояний 15-го и 5-го разрядов (eme, aswe) регистра cfg3_. (см. таблицу 1.617).

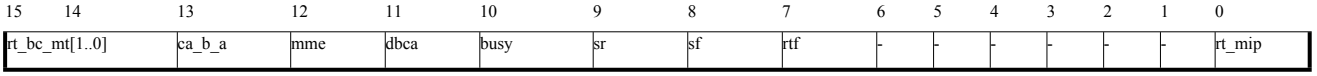
Таблица 1.617

rt_bc_mt[1]	rt_bc_mt[0]	eme	aswe	Режимы работы МСП
0	0	x	x	КШ
1	0	x	0	ОУ, стандартное формирование ОС
1	0	1	1	ОУ, альтернативное формирование ОС
0	1	x	x	МТ

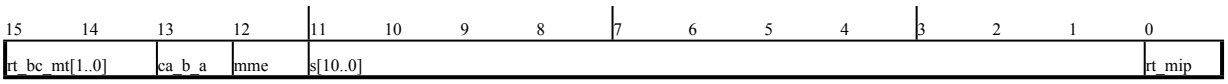
Структура регистра cfg1_ для режима КШ (rt_bc_mt[1..0] = 2'b00, eme = bx, aswe = bx):



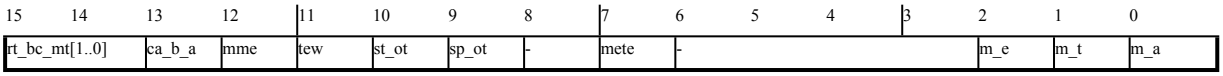
Структура регистра cfg1_для режима ОУ, стандартное формирование ОС (rt_bc_mt[1..0] = 2'b10, eme = bx, aswe = b0):



Структура регистра cfg1_для режима ОУ, альтернативное формирование ОС (rt_bc_mt[1..0] = 2'b10, eme = b1, aswe = b1):



Структура регистра cfg1_для режима МТ (rt_bc_mt[1..0] = 2'b01, eme = bx, aswe = bx):



Выбор режимов МСП в зависимости от содержимого разрядов rt_bc_mt[1..0], mme регистра cfg1_ и разряда eme регистра cfg3_ иллюстрируется нижеследующей таблицей 1.618:

Таблица 1.618

cfg1_			cfg3_	Режим
Разряд 15 rt_bc_mt[1]	Разряд 14 rt_bc_mt[0]	Разряд 12 mme	Разряд 15 eme	
0	0	X	0	КШ, «нерасширенные возможности конфигурирования»
0	0	X	1	КШ, «расширенные возможности конфигурирования»
0	1	0	X	Монитор слов
0	1	1	0	Монитор слов, «нерасширенные возможности конфигурирования»
0	1	1	1	Монитор сообщений, «расширенные возможности конфигурирования»
1	0	X	0	ОУ, «нерасширенные возможности конфигурирования»
1	0	0	1	ОУ, «расширенные возможности конфигурирования»
1	0	1	1	ОУ/монитор сообщений, «расширенные возможности конфигурирования»
1	1	X	X	Режим холостого хода

Замечания:

После аппаратного или программного сброса (посредством регистра запуска/сброса) МСП будет попадать в пассивный режим - режим холостого хода.

Для СФ-блока МКО верны следующие утверждения:

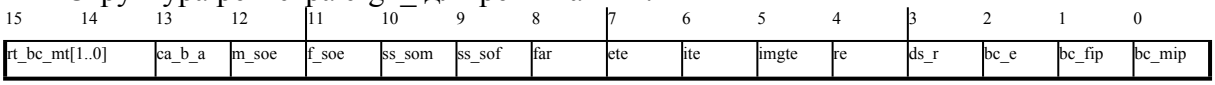
- если переключить МСП из режима ОУ в комбинированный режим ОУ/«монитор сообщений» в середине обработки сообщения, обработка текущего сообщения не прервётся, выполнившись до конца;

- если переключение МСП из комбинированного режима ОУ/«монитор сообщений» в режим ОУ произвести в середине сообщения, обработка текущего сообщения не прервётся, выполнившись до конца;

- переключение режимов работы МСП во время обработки сообщения рекомендуется начинать с программного или аппаратного сброса МСП.

Чтобы записать значения разрядов регистра cfg3_ с 14-го по 0-й необходимо при первоначальном конфигурировании (после ‘сброса’) предварительно записать в регистр cfg3_ значение 16’b1xxx_xxxx_xxxx_xxxx - перевести МСП в состояние «расширенных возможностей конфигурирования». При последующих операциях записи в регистр cfg3_ значение старшего разряда должно оставаться лог. «1». Ненулевые значения в разряды регистров cfg4_, cfg5_, cfg6_ (кроме разрядов cfg5_, задающих собственный адрес устройства) будут записаны только после предварительной (однократной, при первоначальном конфигурировании) записи в регистр cfg3_ значения 16’b1xxx_xxxx_xxxx_xxxx.

Структура регистра cfg1_ для режима КШ:



Наименования разрядов регистра cfg1_ для режима КШ:

rt_bc_mt[1..0] выбор режима работы, имеет значение 2’b00;

ca_b_a(Current Area Б/А*): задание активной области ОЗУ;

m_soe (Message Stop-On-Error): останов работы после окончания текущего сообщения в случае обнаружения ошибки;

f_soe (Frame Stop-On-Error): останов работы после окончания текущего кадра КШ в случае обнаружения ошибки;

ss_som (Status Set Stop-On-Message): останов работы после окончания текущего сообщения в случае обнаружения ‘особого состояния ОУ’;

ss_sof (Status Set Stop-On-Frame): останов работы после окончания текущего кадра КШ, в случае обнаружения ‘особого состояния ОУ’;

far (Frame Auto-Repeat): автоповтор отработки кадра;

ete (External Trigger Enable): разрешение старта кадра КШ от внешнего сигнала;

ite (Internal Trigger Enable): разрешение старта кадра КШ от внутреннего счетчика времени кадра КШ;

imgte (Inter-Message Gap Timer Enabled): разрешение задания промежутка времени для сообщения

re (Retry Enabled): разрешение ВПС;

ds_r (Double/Single Retry): два/одно* ВПС;

bc_e (BC Enabled): (только чтение) флаг работы в режиме КШ;

bc_fip (BC Frame In Progress): (только чтение) флаг процесса отработки кадра;

bc_mip (BC Message In Progress): (только чтение) флаг процесса отработки сообщения.

Функциональное описание разрядов регистра cfg1_ для режима КШ:

ca_b_a (Current Area B/A) – разряд задание активной области ОЗУ.

Значение разряда выбирает (задаёт) активную группу области ОЗУ. Для режима КШ в каждую из этих групп областей входят: стек команд, указатель на стек команд, счетчик сообщений, начальное значение указателя на стек команд (для работы в режиме «автоповтор отработки кадра»), начальное значение счетчика сообщений (для работы в режиме «автоповтор отработки кадра»).

Активная область ОЗУ используется в текущей работе протокольной логики МСП. Позволяет реализовать для УВ ‘глобальную’ двойную буферизацию: после переключения МСП на работу с активной областью ОЗУ другая область находится в распоряжении УВ для считывания результатов отработки кадра и формирования следующего кадра.

Отметим, что, если значение данного разряда изменяется процессором УВ во время отработки кадра, то величина, которая возвращается процессору операцией чтения, будет оставаться старой до тех пор, пока кадр КШ не завершится. После того, как текущий кадр КШ завершится, значение разряда **ca_b_a** (13-ый разряд регистра **cfg1_**), после прочтения значения регистра cfg1_ процессором УВ, окажется переключенным и возвратит свое новое значение.

Следует также отметить, что выбор активной области никак не соотносится с выбором или работой каналов для резервированной магистральной ЛПИ (канал А или канал В).

Если этот разряд установлен в лог. «0», то активная область ОЗУ А. В область ОЗУ А входят: стек команд области А, указатель на стек команд области А, счетчик сообщений области А, начальное значение указателя на стек команд области А (для работы в режиме «автоповтор отработки кадра»), начальное значение счетчика сообщений области А (для работы в режиме «автоповтор отработки кадра»).

Если этот разряд установлен в лог. «1», то активная область ОЗУ Б. В область ОЗУ Б входят: стек команд области Б, указатель на стек команд области Б, счетчик сообщений области Б, начальное значение указателя на стек команд области Б (для работы в режиме «автоповтор отработки кадра»), начальное значение счетчика сообщений области Б (для работы в режиме «автоповтор отработки кадра»).

m_soe (Message Stop-On-Error) – разряд останов работы после окончания текущего сообщения в случае обнаружения ошибки.

Если этот разряд установлен в лог. «1», то МСП будет преждевременно прекращать (останавливать) дальнейшую обработку сообщений после окончания обработки текущего сообщения в случае обнаружения ошибки: ошибка в типе синхроимпульса, кодирования, контроля по чётности, некорректность адреса ОУ в ОС, большее или меньшее количество слов, превышено время ожидания ОС, «ошибка проверочного возврата».

Если ВПС разрешено для данного сообщения и при его отработке обнаружилась ошибка, то ВПС будет осуществляться даже в случае логической «1» в m-soe. Следует отметить, что обработка последующих сообщений будет продолжена, если отработка ошибочного сообщения была успешно повторена.

Для режима КШ разряды с 11 по 0 регистра **cfg1_** определены лишь в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_**, установлен в лог. «1») и имеют следующий смысл:

f_soe (Frame Stop-On-Error) – разряд останов работы после окончания текущего кадра КШ в случае обнаружения ошибки.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15 -ый разряд регистра **cfg3_**, установлен в лог. «1»). Данный разряд проявляет себя, если **far**, 8-ой разряд регистра **cfg1_**, запрограммирован в лог. «1». Если **f_soe** установлен в лог. «1», то МСП будет завершать обработку сообщений после окончания текущего кадра КШ (прекращается автоповтор отработки кадра) в случае обнаружения ошибки: межсловного промежутка времени, типа синхроимпульса, кодирования, контроля по чётности, некорректность адреса ОУ в ОС, большее или меньшее количество слов, превышено время ожидания ОС, «ошибка проверочного возврата».

Если попытки ВПС разрешены для данного сообщения и при его отработке обнаружилась ошибка, то повторы передачи сообщения будут осуществляться даже в случае содержания разрядом **f_soe** лог. «1». Следует отметить, что обработка последующих кадров КШ будет продолжена, если отработка ошибочного сообщения была успешно повторена.

ss_som (Status Set Stop-On-Message): – разряд останов работы после окончания текущего сообщения в случае обнаружения ‘особого состояния ОУ’.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15 -ый разряд регистра **cfg3_**, установлен в лог. «1»).

Если **ss_som** запрограммирован в лог. «1», МСП будет преждевременно завершать обработку последующих сообщений (прекратит работу) после окончания обработки текущего сообщения при событии ‘особое состояние ОУ’.

Событие ‘особое состояние ОУ’ определяется как ‘неожидаемое’ значение для хотя бы одного из одинадцати признаков полученного ОС.

Если разряд «разрешение полного слова управления сообщением» КШ (**e_bc_cwe**, 12-ый разряд регистра **cfg4_**) запрограммирован в лог. «0», то ‘особое состояние ОУ’ охватывает все 11 признаков в ОС. ‘Ожидаемым’ значением для любого из 11-ти признаков ОС является лог. «0», за следующими исключениями:

Если bm_ex, 11-ый разряд регистра **cfg4_**, установлен в лог. «0» и слово управления сообщением КШ содержит разряд «маскирование групповых сообщений» установленным в лог. «1», то ‘ожидаемой‘ величиной признака «Принята групповая команда» становится лог. «1» а не лог. «0». Если разряд bm_ex запрограммирован в лог. «1», то разряд «маскирование групповых сообщений» в слове управления сообщением используется по прямому своему назначению - для маскирования признака «Принята групповая команда» ОС, а не для исполнения операции «XOR» с ним. В этом случае событие ‘особое состояния ОУ’ возникает при значении лог. «1» признака «Принята групповая команда» в полученном ОС, когда разряд «маскирование групповых сообщений» в слове управления сообщением находится в лог. «0».

Если разрешение полного слова управления сообщением КШ (**e_bc_cwe**, 12-ый разряд регистра **cfg4_**) запрограммирован в лог. «1», то признаки ОС маскируются разрядами с 14 по 9 слова управления сообщением КШ. В этом случае, соответствующий разряд слова управления сообщением КШ должен быть установлен в лог. «0», чтобы событие ‘особое состояние ОУ’ происходило, когда значение соответствующего признака, в полученном ОС, равно лог. «1». В этом случае событие ‘особое состояние ОУ’ не происходит, если соответствующий разряд слова управления сообщением КШ установлен в лог. «1».

Если для какого-либо отдельного сообщения ВПС разрешены для события ‘особое состояние ОУ’, они будут производиться даже в случае равенства содержимого разряда ss_som

логической единице. Следует отметить, что обработка последующих сообщений КШ будет продолжаться, если для сообщения, вызвавшего событие ‘особое состояние ОУ’, было ВПС и при повторной его пересылке событие ‘особое состояние ОУ’ не произошло.

ss_sof (Status Set Stop-On-Frame) – разряд останов работы после окончания текущего кадра КШ, в случае обнаружения ‘особого состояния ОУ’.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Если установлен в состояние лог. «1», то МСП будет преждевременно завершать обработку последующих кадров КШ после завершения текущего кадра, если произошло событие ‘особое состояние ОУ’ в одном или более сообщении кадра. Событие ‘особое состояние ОУ’ было определено при описании предыдущего разряда.

Если для какого-либо отдельного сообщения ВПС разрешены для события ‘особое состояние ОУ’, они будут производиться даже в случае равенства содержимого разряда **ss_som** логической единице. Следует отметить, что обработка последующих кадров КШ будет продолжена, если сообщение, содержавшее событие ‘особое состояние ОУ’, было повторено, и при повторной его пересылке событие ‘особое состояние ОУ’ не произошло.

far (Frame Auto-Repeat) – разряд автоповтор отработки кадра. Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»). Если данный разряд установлен в лог. «0», МСП произведет (обработает) один кадр, после чего работа будет остановлена. Для этого случая предусмотрено сохранять значения указателей на стеки команд и счетчиков сообщений соответственно в местоположениях ОЗУ с адресами 0100 и 0101 для области А, или с адресами 0104 и 0105 для области Б.

Если разряд far установлен в лог. «1», кадры будут повторяться бесконечно, до тех пор, пока одна из команд «программный сброс МСП», «остановить работу после завершения текущего сообщения КШ» или «остановить работу после завершения текущего кадра сообщений КШ» не будет исполнена с помощью регистра запуска/сброса; или до происшествия ошибки или события ‘особое состояние ОУ’, когда разрешена остановка работы КШ на кадре или сообщении для случая появления ошибки или события ‘особое состояние ОУ’. В режиме автоповтора кадров так же, как и в режиме «единственного кадра», текущие значения «указателей на стеки команд» КШ и «счетчиков сообщений» КШ сохраняются соответственно в местоположениях ОЗУ с адресами 0100 и 0101 для области А, и/или с адресами 0104 и 0105 для области Б. Вдобавок для обеспечения автоматического повторения кадров в режиме автоповтора кадров начальные величины, которые помещаются в указатели на стеки команд и в счетчики сообщений в начале каждого кадра, должны быть записаны соответственно в местоположения ОЗУ с адресами 102 и 103 для области А, и/или с адресами 106 и 107 для области Б. Заметим, что необходимо при конфигурировании записать ненулевое значение в регистр длительности кадра КШ -**bcft_rtlc_mttw**.

Если разряд «разрешение старта кадра КШ от внутреннего счетчика времени кадра КШ» (**ite**, 6-ой разряд регистра **cfg1_**) установлен в лог. «1», то кадр КШ будет повторяться с фиксированным промежутком времени, задаваемым регистром **bcft_rtlc_mttw** (изменяется в пределах от 100 мкс до 6,55 сек), либо с промежутком времени полной отработки кадра, если последний превышает значение, заданное в регистре **bcft_rtlc_mttw**.

Альтернативно, если разряд «разрешение старта работы от внешнего сигнала» (**ete**, 7-ой разряд регистра **cfg1_**) установлен в состояние логической «1», каждый повторный кадр может запускаться посредством внешнего импульса, подаваемого на предусмотренный для этих целей вход **msp_ssflag_ext_trig**.

Если в лог. «1» установлены сразу оба разряда данного регистра: **ite** и **ete**, то общее начало работы будет производиться от внешнего запускающего источника, в то время как все последующие повторы кадров будут инициализироваться внутренним счетчиком времени кадра, отслеживающим истечение заданной длительности кадра.

ete (External Trigger Enable) – разряд разрешение старта кадра КШ от внешнего сигнала.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Если разряд в лог. «0», то КШ МСП может запускаться в работу только посредством разряда старт регистра запуска/сброса. Если разряд в лог. «1», то пуск обработки кадра КШ может осуществляться внешним сигналом, подаваемым на входной сигнал **msp_ssflag_ext_trig** или посредством разряда **bc_mt_st** «запуск работы КШ/МТ» регистра запуска/сброса. Если и far, и ite, и ete установлены в логические единицы, то первый кадр запустится сигналом на входе msp_ssflag_ext_trig, а все последующие кадры будут запускаться от внутреннего счетчика времени кадра.

ite (Internal Trigger Enable) – разряд разрешение старта кадра КШ от внутреннего счетчика времени кадра КШ.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (**em**e, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Этот разряд используется совместно с разрядом far для того чтобы разрешать или запрещать автоматическое повторение кадров. Если разряд **far** установлен в лог. «1» и разряд **ite** установлен в лог. «0», то КШ МСП будет прекращать обработку сообщений после завершения единственного кадра; в этом сценарии последующие кадры все-таки могут быть запущены посредством внешнего сигнала на **msp_ssflag_ext_trig** в случае, если разряд ete перед этим своевременно был установлен в лог. «1». Если разряд ite установлен в лог. «1», кадры КШ будут повторяться с фиксированным периодом, как предписывается регистром **bcft_rtlc_mttw** (от 100 мкс до 6,55 сек), либо с промежутком времени полной отработки кадра, если последний превышает значение, заданное в регистре **bcft_rtlc_mttw**. Если оба разряда, **ite** и **ete**, установлены в логические единицы, первый кадр будет запускаться по входу **msp_ssflag_ext_trig**, а все последующие будут стартовать от «внутреннего счетчика времени кадра».

imgte (Inter-Message Gap Timer Enabled) – разряд разрешение задания промежутка времени для сообщения.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Если этот разряд установлен в лог. «0», то временной разрыв между соседними сообщениями имеет фиксированную минимальную величину, равную примерно 8-11 мкс. Если разряд установлен в лог. «1», то промежуток времени для каждого сообщения определяется третьим словом в соответствующем описателе блока сообщения. Это величина времени, которая задаёт промежуток от старта текущего сообщения до момента старта последующего, может задаваться начиная от минимального времени (приблизительно 8-11 мкс.) до 65535 мкс с шагом 1 мкс. Если запрограммированное значение оказывается меньшим, чем время, требуемое для полноценной обработки текущего сообщения, текущее сообщение будет обрабатываться вплоть до полного завершения, отсрочивая тем самым запуск последующего. В вышеописанном случае временной разрыв между соседними сообщениями будет также иметь минимальную величину, равную примерно 8-11 мкс.

Промежуток времени для сообщения включает в себя возможные ВПС. Т.е. в случае выполнения КШ попыток ВПС интервал (промежуток) времени на отработку сообщения не отсчитывается заново, с повторного сообщения.

re (Retry Enabled) – разряд разрешение ВПС.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Если разряд в лог. «0», то ВПС запрещены для всех сообщений. Если разряд в лог. «1», то ВПС могут разрешаться на базе индивидуального сообщения, когда слово управления сообщением КШ имеет значение 8-го разряда равным лог. «1».

ds_r (Double/Single Retry) – разряд два/одно* ВПС.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Если ВПС разрешены (re, 4-ый разряд регистра **cfg1_** установлен в лог. «1»), этот разряд используется для задания количества ВПС, которое будет производиться в случае разрешения ВПС и наступления соответствующего события (ошибка, ‘особое состояние ОУ’). Значение разряда лог. «0» будет вызывать выполнение одного ВПС, в то время как лог. «1» вызовет выполнение до двух ВПС.

bc_e (BC Enabled) – разряд (только чтение) флаг работы в режиме КШ.

Этот разряд активен только в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Этот разряд будет возвращать значение лог. «1» после того, как КШ МСП был запущен в работу либо программными средствами - с использованием регистра запуска/сброса; либо аппаратно - подачей сигнала на вход **msp_ssflag_ext_trig**. Этот разряд будет продолжать возвращение единицы при чтении регистра до тех пор, пока не произойдёт какое-нибудь из следующих событий: завершение единственного кадра сообщения, если только МСП не находится в режиме автоматического повтора кадров; посредством регистра запуска/сброса издаётся программная команда «программный сброс МСП», или «остановить работу после завершения текущего сообщения КШ», или «остановить работу после завершения текущего кадра сообщений КШ»; или, если остановка на сообщении или кадре при обнаружении ошибки или события ‘особое состояние ОУ’ разрешена, и какое-нибудь из перечисленных событий произошло.

bc_fip (BC Frame In Progress) – разряд (только чтение) флаг процесса отработки кадра.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_,** установлен в лог. «1»).

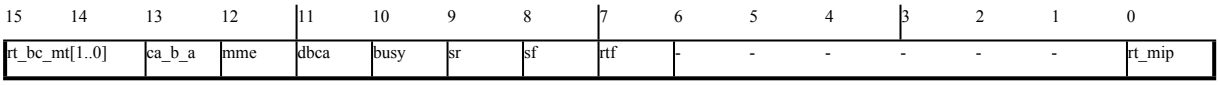
Этот разряд возвращает при чтении величину лог. «1» в промежутке времени между выполнением МСП последовательности действий SOM для первого сообщения кадра КШ и последовательности действий EOM для последнего запрограммированного сообщения кадра КШ. В режиме автоматического повторения кадров **bc_fip** будет автоматически возвращаться к лог. «1» непосредственно перед стартом первой последовательности действий SOM для нового кадра.

bc_mip (BC Message In Progress) – разряд (только чтение) флаг процесса отработки сообщения.

Этот разряд может быть употреблен только в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Этот разряд будет возвращать значение лог. «1» во время обработки всех сообщений КШ. Этот разряд будет перебрасываться из лог. «0» в лог. «1» непосредственно перед последовательностью SOM и возвращаться к лог. «0» сразу после завершения КШ последовательности EOM.

Структура регистра **cfg1**_для режима ОУ, стандартное формирование ОС («без альтернативное ОС») (rt_bc_mt[1..0] = 2’b10, eme = bx, aswe = b0):



Наименования разрядов регистра cfg1_ для режима ОУ, стандартное («без альтернативное») формирование ОС:

rt_bc_mt[1..0] выбор режима работы , имеет значение 2’b10;

ca_b_a(Current Area Б/А*): задание активной области ОЗУ;

mme (Message Monitor Enabled): разрешение режима «монитор сообщений»;

dbca (Dynamic Bus Control Acceptance*): инверсное значение признака ОС «Принято управление интерфейсом»;

busy (Busy*): инверсное значение признака ОС «Абонент занят»;

sr (Service Request*): инверсное значение признака ОС «Запрос на обслуживание»;

sf (Subsystem Flag*): инверсное значение признака ОС «Неисправность абонента»;

rtf (RT Flag *): инверсное значение признака ОС «Неисправность ОУ» (только в состоянии «расширенных возможностей конфигурирования»);

rt_mip (RT Message In Progress) (только чтение): флаг процесса отработки сообщения ОУ;

При стандартном формировании ОС («без альтернативного ОС») значения его признаков (разрядов) задаются согласно ГОСТ Р 52070-2003, при этом программно задаются значения следующих признаков: «Принято управление интерфейсом», «Абонент занят», «Запрос на обслуживание», «Неисправность абонента, и, только в состоянии «расширенных возможностей конфигурирования», «Неисправность ОУ».

При альтернативном формировании ОС значения всех одиннадцати признаков ОС (с 9 по 19) задаются программно.

Функциональное описание разрядов регистра cfg1_ для режима ОУ, стандартное формирование ОС:

ca_b_a (Current Area B/A) – разряд задание активной области ОЗУ. Значение разряда выбирает (задаёт) активную группу области ОЗУ.

Активная область ОЗУ используется в текущей работе протокольной логики МСП.

Отметим, что, если значение данного разряда изменяется процессором УВ во время отработки ОУ сообщения величина, которая возвращается процессору операцией чтения, будет оставаться старой до тех пор, пока отработка ОУ сообщения не завершится. После того, как текущая отработка ОУ сообщения завершится, значение разряд ca_b_a (13-ый разряд регистра cfg1_), после прочтения значения регистра cfg1_ процессором УВ, окажется переключенным и возвратит свое новое значение.

Следует также отметить, что выбор активной области никак не соотносится с выбором или работой каналов для резервированной магистральной ЛПИ (канал А или канал В).

Если этот разряд установлен в лог. «0», то активная область ОЗУ А. Если этот разряд установлен в лог. «1», то активная область ОЗУ Б.

Для режима ОУ «Без альтернативного ОС» (5-ый разряд регистра cfg3_), установленным в лог. «0», ОС МСП будет вырабатываться с полным соответствием требованиям ГОСТ Р 52070-2003. В этом режиме разряды с 11-го по 7-й и разряд 0 определены нижеследующим описанием:

dbca (Dynamic Bus Control Acceptance*) – разряд инверсное значение признака ОС «Принято управление интерфейсом»:

Если в лог. «0», это разрешает ОУ МСП отвечать на команду управления ‘Принять управление интерфейсом’ (но только не в отклике на другие команды) ОС с установленным в лог. «1» признаком «Принято управление интерфейсом». Если этот разряд в лог. «1», признак «Принято управление интерфейсом» в ОС будет в лог. «0».

busy (Busy*): инверсное значение признака ОС «Абонент занят».

Если УВ устанавливает данный разряд в лог. «0», то МСП будет отвечать лог. «1» в соответствующем одноименном признаке ОС (признак «Абонент занят»), и в ответ на сообщение, требующее передачи СД, данные передаваться не будут.

Если данный разряд «инверсное значение признака ОС «Абонент занят»» (busy) установлен в лог. «0», и разряд «запрет сохранения принимаемых СД в состоянии «Абонент занят»» (b_rx_td, 3-ий разряд регистра cfg3_) установлен в лог. «0», то МСП будет сохранять поступающие данные в ОЗУ. Тем не менее, если разряд busy установлен в лог. «0» и разряд b_rx_td установлен в лог. «1», то поступающие на вход МСП данные не будут сохраняться в ОЗУ.

Если разряд «инверсное значение признака ОС «Абонент занят»» (busy) регистра cfg1_ установлен в «1», то признак «Абонент занят» в передаваемом ОС будет оказываться установленным в лог. «0» при выполнении следующих условий:

разряд «Разрешение использования таблицы занятости подадресов» (blute, 13-ый разряд регистра cfg2_) установлен в лог. «0»;

или

разряд «Разрешение использования таблицы занятости подадресов» (blute, 13-ый разряд регистра cfg2_) установлен в лог. «1» и соответствующий текущему командному слову разряд в таблице занятости подадресов установлен в лог. «0».

sr (Service Request*) – разряд инверсное значение признака ОС «Запрос на обслуживание».

Если разряд в лог. «0», то признак «Запрос на обслуживание» в ОС установится в лог. «1». Если разряд «автосброс разряда «запрос на обслуживание»», 2-ой разряд регистра cfg2_, установлен в лог. «1», то разряд «инверсное значение признака ОС «Запрос на обслуживание»» будет автоматически устанавливаться после того, как ОУ получит и ответит на команду управления ‘Передать векторное слово’.

sf (Subsystem Flag*) - инверсное значение признака ОС «Неисправность абонента».

Если разряд в лог. «0», то признак «Неисправность абонента» в ОС установится в лог. «1». Кроме того, данный признак ОС будет устанавливаться под воздействием входного сигнала msp_ssflag_ext_trig (при низком уровне сигнала). Следует отметить, что значение входного сигнала msp_ssflag_ext_trig не влияет на состояние разряда sf регистра cfg1_.

rtf (RT Flag*): – разряд инверсное значение признака ОС «Неисправность ОУ».

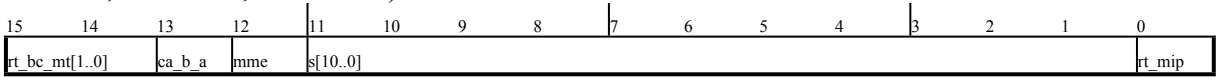
Данный разряд может использоваться лишь в состоянии «расширенных возможностей конфигурирования» (eme, 15-тый разряд регистра cfg3_, установлен в лог. «1»).

Если разряд rtf установлен в лог. «0», то это будет вызывать установление в лог. «1» одноименного признака ОС «Неисправность ОУ». Признак «Неисправность ОУ», передаваемый в ОС, будет также устанавливаться в лог. «1», если rt_f_we (2-ой разряд регистра cfg3_ - «управление формированием признака «Неисправность ОУ»» в ОС) запрограммирован в лог. «1», и произошло либо истечение разрешенного времени непрерывной работы передатчика (668 мкс), либо «ошибка проверочного возврата» для предыдущего негруппового сообщения. Проверочный возврат слова (эхо-контроль передаваемого слова) оценивает достоверность (синхронизация, кодирование, количество разрядов, четность) для принятой версии каждого переданного слова и равенство значения принятой версии последним переданного слова в сообщении исходной версии этого слова.

rt_mip (RT Message In Progress) (только чтение) - разряд флаг процесса отработки сообщения ОУ;

Данный разряд может использоваться лишь в состоянии «расширенных возможностей конфигурирования» МСП (eme, 15-тый разряд регистра cfg3_, установлен в лог. «1»). Если разряд rt_mip при чтении возвращает лог. «1», то значит ОУ МСП в текущий момент времени занимается обработкой сообщения. Разряд rt_mip устанавливается в лог. «1» непосредственно перед выполнением SOM последовательности действий и возвращается к лог. «0» сразу после выполнения EOM последовательности действий. В комбинированном режиме «ОУ/монитор сообщений» лог. «1» разряда rt_mip указывает на обработку сообщения ОУ или монитором сообщений.

Структура регистра cfg1_для режима ОУ, альтернативное формирование ОС (rt_bc_mt[1..0] = 2’b10, eme = b1, aswe = b1):



Наименования разрядов регистра cfg1_ для режима ОУ, альтернативное формирование ОС:

rt_bc_mt[1..0] выбор режима работы, имеет значение 2’b10;

ca_b_a(Current Area Б/А*): задание активной области ОЗУ;

mme (Message Monitor Enabled): разрешение режима «монитор сообщений»;

s[10..0] (S10 through S0): прямое программное управление одиннадцатью младшими разрядами ОС;

rt_mip (RT Message In Progress) (только чтение): флаг процесса отработки сообщения ОУ;

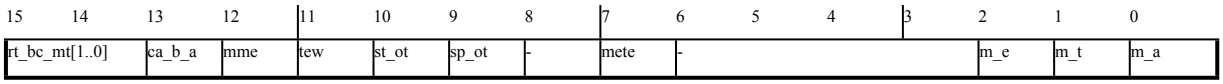
Функциональное описание разрядов регистра cfg1_ для режима ОУ, альтернативное формирование ОС:

Совпадающие разряды регистра cfg1_ для режимов ОУ - альтернативное формирование ОС и стандартное формирование ОС имеют одинаковый смысл.

Для режима ОУ с «альтернативным ОС» (5-ый разряд регистра cfg3_ установлен в лог. «1») все 11 признаков ОС программируются (задаются) УВ. Альтернативное ОС может быть использованным лишь в режиме «расширенных возможностей конфигурирования» ОУ (eme, 15-тый разряд регистра cfg3_, установлен в лог. «1»). В этом режиме разряды с 11 по 0 определены следующим образом:

s[10..0]: В МСП предусмотрена возможность прямого программного управления одиннадцатью младшими признаками ОС. S10 управляет признаком ОС «Ошибка в сообщении». S0 управляет признаком ОС «Неисправность ОУ». Отметим, что логическое значение разрядов такое же (не инверсное) как и для соответствующих признаков в ОС. Следует также отметить, что признак «Ошибка в сообщении» в ОС, управляемый разрядом S10, установится, кроме того, если отдельное командное слово (с конкретными значениями «(не)групповая», «Прием-передача», «Подадрес», «Число СД/Код команды») было определено недопустимым.

Структура регистра cfg1_для режима МТ («монитор слов», «монитор сообщений»):



Наименования разрядов регистра cfg1_ для режима МТ:

rt_bc_mt[1..0] - выбор режима работы, имеет значение 2'b01;

ca_b_a (Current Area Б/А*): задание активной области ОЗУ;

mme (Message Monitor Enabled): разрешение режима «монитор сообщений»;

tew (Trigger Enabled Word): разрешение запуска/останова по событию;

st_ot (Start-On-Trigger): старт записи слов по слову срабатывания;

sp_ot (Stop-On-Trigger): останов записи слов по слову срабатывания;

mete (Monitor External Trigger Enable): разрешение запуска по внешнему сигналу;

m_e (Monitor Enabled): (только чтение) флаг режима МТ;

m_t (Monitor Triggered): (только чтение) флаг события;

m_a (Monitor Active): (только чтение) флаг активности МТ.

Функциональное описание разрядов регистра cfg1_ для режима МТ:

ca_b_a (Current Area Б/А*): – разряд задание активной области ОЗУ.

Задаёт активную (текущую) область ОЗУ Б/А* ; в частности, задаёт фиксированные местоположения для значений указателя стека команд и указателя стека данных для «монитора сообщений», и указатель буфера данных для «монитора слов». При значении лог. «0» активной выбирается область ОЗУ А, лог. «1» - область Б.

Режим «монитор слов». Выбор активной области влияет на адрес расположения указателя буфера данных. Если активная область А, то указатель буфера данных расположен в ячейке 0100. Если активная область Б, то указатель буфера данных расположен в ячейке 0104. Если МСП находится уже вовлечённой в работу в режиме «монитор слов», переустановка этого разряда безразлична (не имеет влияния на работу МСП). Следует также отметить, что выбор активной

области никак не соотносится с выбором или работой каналов для резервированной магистральной ЛПИ (канал А или канал В).

Для режима «монитора слов» в состоянии «расширенных возможностей конфигурирования» (eme, 15 -ый разряд регистра **cfg3_**, установлен в лог. «1») разряды с 11 по 0 данного регистра определены как указано ниже. Отметим, что они не имеют смысловой нагрузки для режима «монитора слов» в состоянии «нерасширенные возможности конфигурирования» (15-ый разряд регистра **cfg3_** установлен в лог. «0») МСП.

Для режима «монитора сообщений» определены разряды с 15 по 12 и 0 разряд в регистре **cfg1_**.

tew (Trigger Enabled Word) - разряд разрешение запуска/останова по событию:

Используется только в режиме «монитор слов».

Для активизации логики событий в режиме «монитор слов» данный разряд следует запрограммировать в лог. «1». Возможности для запуска по «событию» имеются лишь в режиме «монитор слов». Они отсутствуют в режиме «монитор сообщений». Событием является или т.н. «срабатывание» (равенство достоверного принятого командного слова содержимому регистра слова срабатывания) или внешний импульс, поступивший на вход **msp_ssflag_ext_trig**. Разряд «разрешение запуска/останова по событию» должно быть запрограммировано в лог. «1», чтобы позволить использование для запуска внешний импульс, подаваемый на **msp_ssflag_ext_trig**, и использовать разряды **st_ot, sp_ot**. Логика событий в режиме «монитор слов» может обеспечивать старт наблюдения ЛПИ (сохранение распознаваемых слов в ОЗУ), остановку наблюдения ЛПИ (прекращение сохранения распознаваемых слов в ОЗУ), установку в лог. «1» разряда флаг события (**m_t**, 1-ый разряд регистра **cfg1_**) и/или порождение запрос на обслуживание прерывания.

st_ot (Start-On-Trigger) - разряд старт записи слов по слову срабатывания.

Используется только в режиме «монитор слов».

Если этот разряд установлен в лог. «1», МСП будет начинать сохранение слов в ОЗУ после принятия достоверного командного слова, которое равно слову, предварительно записанному в регистр слова срабатывания. В результате слово, вызвавшее запуск работы монитора, будет первым сохраненным в ОЗУ словом.

sp_ot (Stop-On-Trigger) - разряд останов записи слов по слову срабатывания:

Используется только в режиме «монитор слов».

Если этот разряд установлен в лог. «1», МСП будет останавливать сохранение слов после принятия достоверного командного слова, которое равно по значению содержимому регистра слова срабатывания. Командное слово, приведшее к данному событию, будет последним словом, сохраненным в ОЗУ для данного сеанса работы МТ.

mete (Monitor External Trigger Enable) - разряд разрешение запуска по внешнему сигналу.

Используется только в режиме «монитор слов».

Если в единице, разрешает внешнему импульсу, поступающему на вход **msp_ssflag_ext_trig**, обеспечивать запуск в режиме «монитора слов».

m_e (Monitor Enabled (Read Only)) - разряд флаг режима МТ.

Используется только в режиме «монитор слов».

Разряд m_e возвращает (при чтении) значение лог. «1», если МТ был запущен в работу и находится в режиме «наблюдения» ЛПИ. МТ МСП считается запущенным в работу после записи программным путем лог. «1» в первый разряд регистра запуска/сброса или, для режима «монитор слов», после обнаружения соответствующего внешнего сигнала на входе **msp_ssflag_ext_trig**. При этом запись слов в ОЗУ может не осуществляться.

m_t (Monitor Triggered (Read Only)) - разряд флаг события.

Данный разряд применяется только в режиме «монитор слов». Он неприменим для режима «монитор сообщений» и будет возвращать (при чтении) лог. «0».

Этот разряд будет возвращаться к лог. «0» после записи лог. «1» в первый разряд регистра запуска/сброса (команда старт). Этот разряд устанавливается в состояние лог. «1» после того, как предварительно разрешенные события были обнаружены. Два события, которые разрешаются

посредством разрядов 11 и 7 регистра **cfg1_** это - принятие достоверного командного слова, равного по значению содержимому «регистра слова срабатывания» или обнаружения перехода из ‘неактивного’ в ‘активный’ уровня внешнего сигнала на **msp_ssflag_ext_trig**. Однажды установленный в лог. «1», разряд **m_t** будет находиться в этом состоянии до тех пор, пока МСП не сбросят или не остановят. Если «монитор слов» оказывается остановленным по слову срабатывания (разрешается **sp_ot**, 9-ым разрядом регистра **cfg1_**), то **m_t** будет сохранять лог. «1» до тех пор, пока не будет повторно запущен.

m_a (Monitor Active (read Only)) - разряд флаг активности МТ.

В режиме «монитор слов»» разряд m_a будет возвращать (при чтении) лог. «1» после того, как «монитор слов» был запущен в работу. То есть после того, как «монитор слов» был запущен в работу командой старт (запись лог. «1» в 1-ый разряд регистра запуска/сброса) или после обнаружения перехода из неактивного в активный уровня внешнего сигнала на **msp_ssflag_ext_trig**. При этом запись слов в ОЗУ может не осуществляться.

В режиме «монитор сообщений» **m_a** будет возвращать лог. «1» (при чтении), только когда «монитор сообщения» находится на стадии отработки сообщения (от SOM до EOM последовательности).

Замечания:

В комбинированном режиме работы «ОУ/монитор сообщений» используются определения разрядов, данные для ОУ. Определения разрядов для работы в качестве монитора не используются в данном режиме.

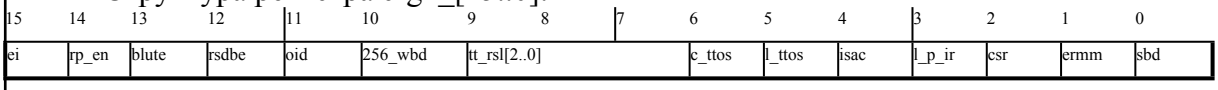
В режиме ОУ (даже если выбрано альтернативное формирование ОС), в случае использования логики задания множества недопустимых сообщений, 10-ый признак ОС «Ошибка в сообщении» автоматически устанавливается для недопустимого сообщения.

Применение разрядов **s[10..0]** описана в разделе "работа оконечного устройства”.

1.3.12.2.2.4.1.3 Второй конфигурационный регистр - cfg2_

Адрес регистра 00010; чт/зп.

Структура регистра cfg2_[15..0]:



Наименования разрядов регистра cfg2_:

ei (Enhanced Interrupts), разрешение расширенных возможностей прерывания;

rp_en (Ram Parity Enable), разрешение проверки четности для ОЗУ;

blute (Busy Look Up Table Enable), разрешение использования таблицы занятости подадресов;

rsdbe (RX SA Double Buffer Enable), разрешение глобальной двойной буферизации для сообщений «приема»;

oid (Overwrite Invalid Data), переписывать недостоверные данные;

256_wbd (256-Word boundary Disable), запрещение 256-словных ограничений;

tt_rsl[2..0] (Time Tag Resolution), разрешение (цена МЗР) для счетчика метки времени;

c_ttos (Clear Time tag On Sinchronize), сброс счетчика метки времени при принятии КУ ‘Синхронизация’;

l_ttos (Load Time tag On Sinchronize), загрузка СД в «регистр счетчика метки времени» при принятии КУ ‘Синхронизация (с СД)’;

isac (Interrupt Status Auto Clear), автоматический сброс запроса прерывания и значения регистра isr;

l_p_ir (Level/Pulse Interrupt Request), вид сигнала msp_int - импульс/уровень;

csr (Clear Service Request), ‘автосброс’ разряда ‘запрос на обслуживание’;

ermm (Enhanced RT Memory Management), расширенное управление памятью ОУ;

sbd (Separate Brodcast Data), отделение данных групповых сообщений.

Функциональное описание разрядов регистра **cfg2_:**

ei (Enhanced Interrupts) – разряд разрешение расширенных возможностей прерывания;

Данный разряд может быть использован лишь в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Разряд **ei** управляет работой и взаимодействием разрядов регистров маскирования прерывания (**imr**) и состояния прерывания (**isr**). Если запрограммирован в лог. «0», в формировании запроса прерывания участвуют лишь те события, которые могут быть замаскированы разрядами с 0 по 7 регистра imr. Кроме того, когда разряд ei установлен в лог. «0», отдельный разряд в регистре isr может изменять своё значение, реагируя на событие, только если соответствующий ему разряд регистра imr запрограммирован в лог. «1».

Если ei установлен в лог. «1», то все из 14 возможных событий участвуют в формировании запроса прерывания посредством разрядов с 13 по 0 регистра imr. Кроме того, когда **ei** в лог. «1», разряды регистра **isr** устанавливаются в логические единицы, реагируя на прерывающие события, вне зависимости от значений соответствующих им разрядов регистра imr.

rp_en (Ram Parity Enable), – разряд разрешение проверки четности для ОЗУ;

Данный разряд не задействован в МСП.

Данный разряд может быть использован лишь в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»). Разряд **rp_en**, когда установлен в лог. «1», разрешает проверку четности для 17-ти разрядного буферного ОЗУ. Это влечет за собой вырабатывание значения 17-го разряда (разряд (не)четности) для обоих видов доступа к ОЗУ в режиме записи (запись со стороны процессора и протокольной логики МСП) и проверку четности для всех видов чтения ОЗУ. Если проверка четности приносит неудовлетворительный результат для какого-либо доступа чтения, то МСП способен отреагировать на это формированием запроса обработки прерывания и установлением разряда **rp_er** в «регистре состояния прерываний» в лог. «1».

blute (Busy Look Up Table Enable), – разряд разрешение использования таблицы занятости подадресов;

Данный разряд может быть использован лишь в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Этот разряд разрешает процессору УВ для МСП в режиме ОУ программировать задание лог. «1» значения признака «Абонент занят» в ОС в зависимости от значения командного слова: (не)групповая команда, разряд «Прием-передача», поле «Подадрес».

Таблица занятости подадресов размещается по фиксированным адресам ОЗУ (от 0240 шестн. до 0247 шестн.).

Программирование разряда **blute** в лог. «0» запрещает формирование в ОЗУ таблицы занятости подадресов и высвобождает фиксированную для неё область.

rsdbe (RX SA Double Buffer Enable), – разряд разрешение глобальной двойной буферизации для сообщений «приема»;

Данный разряд может быть использован лишь в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»).

Если разряд **rsdbe** запрограммирован в состояние лог. «0», то двойная буферизация для сообщений «приема» запрещается.

Если разряд rsdbe запрограммирован в лог. «1» и разряд «расширенное управление памятью ОУ» (ermm, 1-ый разряд регистра **cfg2_**) установлен в лог. «0», то двойная буферизация для сообщений «приема» используется для всех подадресов команд «приема» (команды «приема», «группового приема»).

Если оба разряда - **rsdbe** и **ermm** - запрограммированы в лог. «1», то двойная буферизация для сообщений «приема» может быть задействована для индивидуального подадреса команд «приема», «группового приема» установкой 15-го разряда в соответствующем «слове управления

подадресом ОУ» в состояние лог. 1 и программированием принадлежащих этому слову «разрядов управления памятью» (MM2, MM1, MM0) в логические нули.

oid (Overwrite Invalid Data), – разряд переписывать недостоверные данные;

Данный разряд не задействован в МСП. Эл. связь, «Переписывать недостоверные данные», подсоединена не к выходу данного триггера, а к уровню лог. «0». При чтении разряд oid будет возвращать записанное в него ранее, наиболее свежее значение.

256_wbd (256-Word boundary Disable), – разряд запрещение 256-словных ограничений;

Этот разряд определяет схему управления памятью МСП как для блоков сообщений КШ, так и в двух следующих случаях для блоков данных ОУ:

(1) для режима, установленного по умолчанию, - разряд «расширенное управление памятью ОУ» (ermm, 1-ый разряд регистра **cfg2_**) установлен в лог. «0») или

(2) в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»), в случае режима «расширенное управление памятью ОУ» (ermm, 1-ый разряд регистра **cfg2_**, установлен в лог. «1»), когда сделан выбор режима «одиночная буферизация» или «двойная буферизация сообщенрий «приема»» для каких-либо заданных подадресов.

Если разряд **256_wbd** установлен в лог. «0», то ‘прокручивание' в рамках 256-словных границ разрешено. Термин ‘прокручивание’ означает следующее: когда логика управления памятью обрабатывает сообщение, последовательно формируемые значения адреса памяти будут переходить от значений XXFF (шестн.) к значениям XX00 (шестн.), а не увеличиваться до (XX + 1)00 (шестн.).

Если разряд **256_wbd** установлен в лог. «1», то логика управления памятью будет способна пересекать 256-словные границы. То есть, когда 256-словная граница встречается в пространстве разделяемого ОЗУ, значения адреса будут увеличиваться от значений XXFF (шестн.) до значений (XX + 1)00 (шестн.).

Следует отметить, что разряд 256_wbd («запрещение 256-словных ограничений») не влияет на стеки команд в КШ и ОУ. То есть для области ОЗУ стек команд в обоих режимах работы (КШ, ОУ) прокручивание в рамках границ 256-(или 512-, 1024-, 2048-) слов является всегда заданным принудительно.

Также следует обратить внимание на то что, если «запрещение 256-словных ограничений» запрограммировано в состояние лог. «1», то в режиме ОУ размеры всех кольцевых буферов данных становятся равными 65536 слов, вне зависимости от значений соответствующих разрядов в слове управления подадресом ОУ.

Настоятельно рекомендуется устанавливать разряд **256_wbd** («запрещение 256-словных ограничений») в лог. «1» для режима КШ и в лог. «0» для режима ОУ.

tt_rsl[2..0] (Time Tag Resolution), – разряд разрешение (цена МЗР) для счетчика метки времени;

Эти разряды определяют цену младшего разряда для регистра счетчика метки времени, которая может устанавливаться программным образом. Возможен выбор среди следующего перечня величин: 64 мкс (по умолчанию), 32 мкс, 16 мкс, 8 мкс, 4 мкс, 2 мкс, и внешний тактовый синхросигнал. Также возможен тестовый режим работы для регистра счетчика метки времени.

Замечание: в тестовом режиме «счетчик метки времени» увеличивает свое значение на единицу, когда процессор УВ записывает лог. «1» в разряд «программно управляемый тактовый синхроимпульс регистра счетчика метки времени» (tttc, 4-ый разряд регистра **srr**).

Таблица 1.619 - – Разрешение счетчика метки времени

Разряд 9 tt_rsl 2	Разряд 8 tt_rsl 1	Разряд 7 tt_rsl 0	Разрешение счетчика метки времени
0	0	0	64 мкс
0	0	1	32 мкс
0	1	0	16 мкс
0	1	1	8 мкс
1	0	0	4 мкс
1	0	1	2 мкс
1	1	0	Режим тест
1	1	1	Внешняя синхронизация

c_ttos (Clear Time tag On Sinchronize), – разряд сброс счетчика метки времени при принятии КУ ‘синхронизация’;

Если разряд установлен в лог. «1», то принятие КУ ‘Синхронизация’ в режиме ОУ будет вызывать сброс значения регистра счетчика метки времени МСП в 0000 (шестн.).

l_ttos (Load Time tag On Sinchronize), – разряд загрузка СД в «регистр счетчика метки времени» при принятии КУ ‘Синхронизация (с СД)’;

Если установлен в лог. «1», то принятие КУ ‘Синхронизация с СД’ будет вызывать загрузку слова данных сообщения в регистр счетчика метки времени.

isac (Interrupt Status Auto Clear), автоматический сброс запроса прерывания и значения регистра isr;

Если разряд isac установлен в лог. «1», то значение регистра isr будет ‘очищаться’ в 0000 (шестн.) после прочтения его процессором УВ. Также при этом, если разряды isac и l_p_ir (соответственно 4-ый и 3-ий разряды регистра cfg2_) одновременно установлены в лог. «1», то выходной сигнал «запрос обработки прерывания» (msp_int) будет сброшен в неактивное высокоуровневое состояние после прочтения регистра isr.

l_p_ir (Level/Pulse Interrupt Request*), разряд вид сигнала msp_int - импульс/уровень;

Если разряд l_p_ir установлен в лог. «0», то запрос на обработку прерывания на выходном сигнале msp_int будет формироваться в виде «отрицательного импульса» (с нулевым уровнем напряжения) шириной приблизительно 500 нс. Если разряд l_p_ir установлен в лог. «1», прерывание будет запрашиваться установкой на выходе msp_int напряжения низкого уровня. В случае выбора уровнего вида запроса msp_int будет «сбрасываться» в неактивное состояние после прочтения регистра isr (подразумевается, что разряд isac регистра cfg2_ установлен в лог. «1»), или когда записывается единица в разряд «Сброс прерывания» регистра srr. При включении питания l_p_ir оказывается в состоянии лог. «0», задавая этим импульсный режим работы выхода msp_int.

csr (Clear Service Request), – разряд ‘автосброс’ разряда ‘запрос на обслуживание’;

Если разряд csr установлен в лог. «0», то признак «Запрос на обслуживание» в ОС может управляться УВ лишь программным образом. Если csr в лог. «1», то «Запрос на обслуживание» может по-прежнему устанавливаться и сбрасываться под управлением программы и, вдобавок к тому, разряд «инверсное значение признака ОС «Запрос на обслуживание»» регистра cfg1_ будет автоматически очищаться (переходить в лог. «1») после отклика ОУ МСП на КУ ‘Передать векторное слово’. То есть, если разряд csr является установленным в лог. «1» в то время, когда разряд «инверсное значение признака ОС «Запрос на обслуживание»» в регистре cfg1_ запрограммирован в лог. «0», то ОУ МСП будет откликаться с установленным в лог. «1» признаком «Запрос на обслуживание» в ОС для всех КС, пока ОУ не получит и отзовется на КУ ‘Передать векторное слово’. В случае отклика на КУ ‘Передать векторное слово’ лог. «1» в признаке «Запрос на обслуживание» ОС по-прежнему будет сохранено, однако для последующих сообщений значение разряда «инверсное значение признака ОС «Запрос на

обслуживание»» в регистре cfg1_ будет уже в лог. «1» и будет удерживаться в этом ‘очищенном’ значении для последующих сообщений, впредь до его установки УВ в лог. «0».

ermm (Enhanced RT Memory Management), – разряд расширенное управление памятью ОУ;

Если разряд ermm установлен в лог. «0», то логика управления памятью ОУ МСП подразумевает использование единого режима буферизации СД для всех значений подадреса, при этом:

1) Если разряд rsdbe (12-ый разряд регистра cfg2_) запрограммирован в лог. «0», то задаётся режим «одиночной буферизации» (для всех передающих подадресов и для всех принимающих подадресов, в том числе и групповых сообщений «приёма»).

2) Если разряд eme (15-ый разряд регистра cfg3_) установлен в лог «1» и разряд rsdbe (12-ый разряд регистра cfg2_) в состоянии лог. «1», то задается режим «двойной буферизации сообщений «приема»», который обеспечивается для всех принимающих подадресов, в том числе и групповых сообщений «приёма».

Если разряд ermm установлен в лог. «1», то возможности по управлению памятью ОУ МСП расширяются, обеспечивая работу в режиме «одиночной буферизации», или «двойной буферизации сообщений «приема»», «кольцевой буферизации» (с задаваемым размером буфера и начальным значением указателя). Режим «буферизации» задается для каждого подадреса в соответствующем «слове управления подадресом ОУ» отдельно для команд «приема», «группового приема» и команд «передачи».

sbd (Separate Brodcast Data), – разряд отделение данных групповых сообщений.

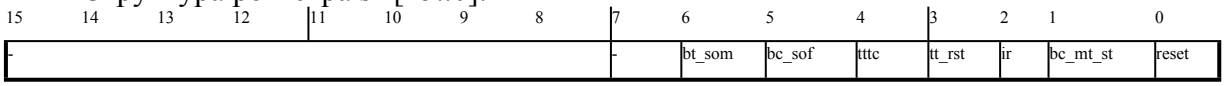
Если разряд sbd установлен в лог. «0», то указатели на блок данных, как для групповых, так и для негрупповых сообщений «приема» содержаться в общей таблице - таблице указателей для подадресов на «прием»/«групповой прием».

Если разряд sbd установлен в лог. «1», то «групповые данные» отделяются от данных негруппового приёма и указатели на блок данных для групповых сообщений «приема» находятся в таблице указателей для подадресов на «групповой прием».

1.3.12.2.2.4.1.4 Регистр запуска/сброса - srr

Адрес регистра 00011; зп.

Структура регистра srr[15..0]:



Регистр запуска/сброса является регистром, доступным лишь по записи. Он обеспечивает командные функции для сброса МСП, для запуска работы КШ или МТ, для сброса «регистра состояния прерываний» и выходного напряжения на выходном сигнале «запрос прерывания УВ», для сброса регистра счетчика метки времени. Он также содержит разряд, обеспечивающий тестирование регистра счетчика метки времени, разряды, останавливающие работу КШ по завершении отработки текущего сообщения или текущего кадра.

Наименования разрядов регистра srr:

bt_som (BC/MT Stop-On-Message), остановить работу после завершения текущего сообщения КШ и МТ;

bc_sof (BC Stop-On-Frame), остановить работу после завершения текущего кадра сообщений КШ;

tttc (Time Tag Test Clock), программно управляемый тактовый синхроимпульс регистра счетчика метки времени;

tt_rst (Time Tag Reset), сброс значения регистра счетчика метки времени;

ir (Interrupt Reset), сброс запроса прерывания;

bc_mt_st (BC/MT Start), запуск работы КШ/МТ;

reset (Reset), программный сброс МСП.

Функциональное описание разрядов регистра:

bt_som (BC/MT Stop-On-Message), – разряд остановить работу после завершения текущего сообщения КШ и МТ;

Запись единичного значения в этот разряд вызовет завершение дальнейшей обработки сообщений для КШ и «монитора сообщений» после завершения текущего сообщения. Если в текущий момент обработка (пересылка) сообщения не производится, прекращение работы (переход в нерабочее состояние) происходит немедленно. В режиме «ОУ/монитор сообщений» разряд **bt_som** не задействован.

bc_sof (BC Stop-On-Frame), – разряд остановить работу после завершения текущего кадра сообщений КШ;

Запись единичного значения в этот разряд в режиме КШ вызовет переход в состояние «бездействующего КШ» после завершения текущего кадра сообщений КШ. Если во время записи единицы в разряд **bc_sof** КШ МСП находится в ожидании старта следующего кадра, то выполнение работы контроллером шины прекратится немедленно.

tttc (Time Tag Test Clock), – разряд программно управляемый тактовый синхроимпульс регистра счетчика метки времени;

Когда разряды «разрешения (цена МЗР) для счетчика метки времени» запрограммированы на проверочный режим работы (разряды 9, 8 и 7 установлены соответственно в 1,1,0), запись единичного значения в этот разряд будет вызывать увеличение значения «счетчика метки времени» на единицу. В «проверочном режиме работы счетчика метки времени» регистр счетчика метки времени реагирует приращением своего значения только на запись лог. «1» в разряд tttc.

tt_rst (Time Tag Reset), – разряд сброс значения регистра счетчика метки времени;

Запись единичного значения в этот разряд вызывает сбрасывание в нуль (0000) значения регистра счетчика метки времени.

ir (Interrupt Reset), – разряд сброс запроса прерывания;

Запись единичного значения в этот разряд сбрасывает состояние регистра isr в 0000 (За исключением разряда «Ошибки контроля по четности адреса ОУ», если она присутствует). К тому же, если разряд **l_p_ir** регистра **cfg2_** запрограммирован в лог. «1», то запись лог. «1» в этот разряд очищает активность на выходе msp_int, устанавливая его в лог. «1».

bc_mt_st (BC/MT Start), – разряд старт КШ/МТ;

Когда лог. «1» записывается в этот разряд в режиме КШ, то МСП стартует обработку запрограммированного кадра сообщений; когда лог. «1» записывается в этот разряд в режиме МТ, то МСП включается в ‘прослушивание’ ЛПИ и, возможно, запускает сохранение принимаемых слов в разделяемое ОЗУ. Следует отметить, что команду «старт МТ» необходимо издавать для старта либо в режиме «монитор слов», либо в режиме «монитор сообщений», находящегося при исполнении исключительно функций монитора. В комбинированном режиме ОУ/«монитор сообщений» издавать команду «старт МТ» не требуется, поскольку, в предположении, что МСП находится в состоянии «расширенных возможностей конфигурирования», ОУ и МТ будут включаться в работу немедленно после того, как регистр cfg1_ был запрограммирован в режим ОУ/«монитор сообщений».

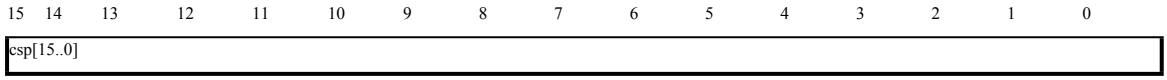
reset (Reset), – разряд программный сброс МСП;

Этот разряд обеспечивает возможность программного сброса для МСП. Когда лог. «1» записывается в этот разряд, обработка какого-либо сообщения в режимах КШ или ОУ немедленно прекращается, не завершившись по правилам. В режиме монитора МСП выключается немедленно и останавливает сохранение принимаемых данных. Все регистры сбрасываются в 0000 (шестн.); все внутренние состояния принимают значения, определённые по умолчанию (как при включении питания).

1.3.12.2.2.4.1.5 Регистр указателя на стек команд - csp

Адрес регистра 00011; чт.

Структура регистра csp[15..0]:



csp[15..0] (Command Stack Pointer): Этот регистр обеспечивает процессор УВ доступом по чтению к текущему значению «указателя на стек команд» в режимах КШ, ОУ и «монитор сообщений».

В режиме КШ значение регистра указателя на стек команд будет равняться значению слова по фиксированному адресу ОЗУ - указателя на стек команд активной области А/Б.

Для ОУ, «монитора избранных сообщений», комбинированного режима ОУ/«монитор сообщений»» величина, прочитанная из регистра указателя на стек команд будет всегда меньше на четыре (по модулю размера стека команд), чем значению слова по фиксированному адресу ОЗУ - указателя на стек команд области А/Б. В режимах ОУ, «монитор сообщений» и в комбинированном режиме ОУ/«монитор сообщений» значение регистра указателя на стек команд увеличивается на четыре во время последовательности SOM, переходя от значения указателя для предыдущего сообщения к значению указателя для текущего сообщения.

В течение SOM-последовательности в комбинированном режиме работы ОУ/«монитор сообщений», регистр указателя на стек команд будет обновляться текущим значением указателя на стек команд ОУ, для сообщения ОУ и текущим значением указателя на стек команд «монитора сообщений» для сообщения наблюдаемого монитором сообщений.

В режиме КШ величина регистра указателя на стек команд приращивается на четыре во время последовательности EOM, изменяясь от величины указателя для текущего сообщения к величине указателя для следующего сообщения.

Регистр указателя на стек команд не используется в режиме «монитор слов». Следует отметить, что размер командного стека КШ/ОУ по умолчанию принимает значение 256 слов в состоянии «нерасширенные возможности конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «0») МСП. В состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») размер может быть запрограммирован посредством разрядов 14 и 13 регистра cfg3_ в одну из следующих величин: 256, 512, 1024, 2048 слов.

1.3.12.2.2.4.1.6 Регистр слова управления сообщением КШ, слова управления подадресом ОУ

Адрес регистра 00100; чт/зп.

Как средство, облегчающее самопроверку, этот регистр обеспечивает доступ по чтению/записи к текущему «слову управления сообщением КШ» или «слову управления подадресом ОУ». Регистр содержит разряды, которые указывают на: активный канал шины и вид сообщения, разрешение проведение встроенной самопроверки, маскирование разрядов ОС, разрешение на проведение ВПС и прерывания.

Хотя этот регистр может быть прочитан в любое время процессором УВ, в него невозможно произвести запись, когда:

МСП запрограммирована в режим ОУ или МТ;

или

идет обработка кадра в режиме КШ.

В регистр слова управления сообщением КШ возможно записать информацию, когда МСП находится: в режиме холостого хода или в режиме КШ (если в текущий момент времени обработка кадра сообщений не производится).

Структура регистра слова управления сообщением КШ bc_cw[15..0], (адрес 00100; чт/зп):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	mem	srbm	sbbm	sfbm	tfbm	rbm	cw_re	bc_a_b	olst	mbb	eom_ie	1553_a_b_s	mcf	bf	rt_rt_f

Наименования разрядов регистра:

mem (Message Error Mask), маскирование признака «Ошибка в сообщении»;
srbm (Service Request Bit Mask), маскирование признака «Запрос на обслуживание»;
sbbm (Subsys Busy Bit Mask), маскирование признака «Абонент занят»;
sfbm (Subsys Flag Bit Mask), маскирование признака «Неисправность абонента»;
tfbm (Terminal Flag Bit Mask), маскирование признака «Неисправность ОУ»;
rbm (Reserved Bit Mask), маскирование «Резервных признаков»;
cw_re (Retry Enabled), разрешение ВПС;
bc_a_b (Bus Chanal A/B), выбор канала;
olst (Off Line Self Test), режим внутреннего самотестирования;
mbb (Mask Broadcast Bit), маскирование признака «Принята групповая команда»;
eom_ie (EOM Interrupt Enable), разрешение прерывания после окончания сообщения;
1533_a_b_s (1553A/B Select), выбор протокола обмена;
mcf (Mode Code Format), формат КУ ;
bf (Broadcast Format), формат группового сообщения;
rt_rt_f (RT-To-RT Format), сообщение формата «ОУ-ОУ».

Структура регистра слова управления подадресом ОУ **rt_cw**[15..0], (адрес 00100; чт/зп):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rx_dbe	tx_eom_i	tx_cbi	tx_mm[2..0]			rx_eon_i	rx_cbi	rx_mm[2..0]				bcst_eom_i	bcst_cbi_i	bcst_mm[2..0]	

Наименования разрядов регистра:

rx_dbe (RX: Double Buffer Enable), разрешение двойной буферизации для сообщений «приёма»;

tx_eom_i (TX: EOM Int), разрешение прерывания по завершению сообщений «передачи»;

tx_cbi (TX: Circ Buf Int), разрешение прерывания по переполнению кольцевого буфера для сообщений «передачи»;

tx_mm[2..0] (TX: Memory Management), способ управления памятью для сообщений «передачи»;

rx_eon_i (RX: EOM Int), разрешение прерывания по завершению сообщений «приёма»;

rx_cbi (RX: Circ Buf Int), разрешение прерывания по переполнению кольцевого буфера для сообщений «приёма»;

rx_mm[2..0] (RX: Memory Management), способ управления памятью для сообщений «приёма»;

bcst_eom_i (BCST: EOM Int), разрешение прерывания по завершению сообщений «группового приёма»;

bcst_cbi_i (BCST: Circ Buf Int), разрешение прерывания по переполнению кольцевого буфера для сообщений «группового приёма»;

bcst_mm[2..0] (BCST: Memory Management), способ управления памятью для сообщений «группового приёма».

1.3.12.2.2.4.1.7 Регистр счетчика метки времени - tt_reg

Адрес регистра 00101; чт/зп.

Структура регистра tt_reg[15..0]:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tt_reg[15..0]															

Таблица 1.620 – Разрешение регистра счетчика метки времени

Второй конфигурационный регистр			Разрешение регистра счетчика метки времени	Время прокручивания
Разряд 9 TTR2	Разряд 8 TTR1	Разряд 7 TTR0		
0	0	0	64 мкс	4,194 с
0	0	1	32 мкс	2,097 с
0	1	0	16 мкс	1,048 с
0	1	1	8 мкс	524 мс
1	0	0	4 мкс	262 мс
1	0	1	2 мкс	131 мс
1	1	0	Тестовый режим	
1	1	1	Внешняя синхронизация	

tt_reg[15..0] (Time Tag Register): Регистр счетчика метки времени обеспечивает доступ в режиме чтения к независимому в своей работе счетчику метки времени, а также доступ в режиме записи для задания начального значения счётчика. Разрешение счетчика программируется посредством разрядов 9, 8 и 7 регистра cfg2_ и может принимать одну из следующих величин: 64, 32, 16, 8, 4 и 2 мкс на младший значащий разряд. Существует также проверочный режим работы, в котором регистр счетчика метки времени может приращивать на единицу свое значение под управлением программного обеспечения в целях самотестирования. Вдобавок регистр счетчика метки времени может стробироваться от внешнего источника синхросигнала, подаваемого на вход TAG_CLK.

Регистр tt_reg сбрасывает свое значение после программного (ttr, 3-ий разряд регистра srr) или аппаратного «сброса» или после получения КУ ‘Синхронизация’ при установленном в лог. «1» разряде 6 регистра cfg2_. Регистр tt_reg загружается полученными данными после принятия КУ ‘Синхронизация’, что становится возможным при установленном в лог. «1» 5-ом разряде регистра cfg2_.

1.3.12.2.2.4.1.8 Регистр состояния прерываний - isr

Адрес регистра 00110; чт.

Регистр состояния прерывания содержит указатели источников запроса, позволяет определить причину формирования запроса на обработку прерывания.

Регистр состояния прерывания позволяет УВ определить причину формирования запроса на обработку прерывания, выполнив лишь одну операцию чтения. Всего в регистре имеется шестнадцать разрядов. Содержимое регистра isr будет каждый раз очищаться после его прочтения, если автоматический сброс запроса прерывания (isac, 4-ый разряд регистра cfg2_) и значения регистра isr в логической «1». Регистр также будет очищаться, если УВ записывает лог. «1» в разряд «сброс запроса прерывания» (ir, 2-ой разряд регистра запуска/сброса).

Следует отметить, что для состояния «нерасширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») в регистре isr[15..0] активны только лишь разряды с 7 по 0. В состоянии «расширенных возможностей конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») и если «расширенные возможности

прерывания» разрешены (ei, 15-ый разряд регистра cfg2_, в лог. «1»), все 15 разрядов (14-0) могут оказаться установленными в лог. «1».

В состоянии «нерасширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») при происшествии различных событий/условий разряды с 7 по 0 регистра imr используются и для разрешения прерываний (формирование сигнала на выходе msp_int), и для разрешения установки в лог. «1» соответствующих разрядов в регистре состояния прерывания (isr). То есть, разряды регистра isr не могут устанавливаться в единичные состояния, если соответствующие им разряды регистра imr «сброшены» в нулевые уровни. Разряды с 14 по 8 не употребляются в состоянии «нерасширенные возможности конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «0»).

В состоянии «расширенных возможностей конфигурирования» и «расширенных возможностей прерывания» (ei, 15-ый разряд регистра cfg2_, уст. в лог. «1»), поддерживается применение, основанное на опросе (не на прерываниях), и значения разрядов регистра imr не влияют на работу регистра isr. Если «расширенные возможности прерывания» разрешены, разряды регистра isr обретают независимость, и для возможности установки в лог. «1» разряда регистра isr отпадает необходимость в получении разрешения от соответствующего ему разряда в регистре масок. То есть в «расширенном режиме прерываний» эти разряды будут переходить в единичные состояния после происшествия соответствующих событий/условий вне зависимости от состояний соответствующих разрядов регистра imr. В обоих режимах работы МСП прерывание (формирование сигнала на выходе msp_int) для какого-либо из 15 событий/условий (8 в состоянии «нерасширенные возможности конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «0»)) разрешено, только если соответствующий разряд регистра imr для события/условия был предварительно запрограммирован в лог. «1».

Формирование сигнала прерывания на выходе msp_int происходит в конце SOM/EOM последовательности, кроме следующих событий/условий прерывания: bt_tt, hf, rt_ape, tt_roll, rt_cbr, ss_mc_pt - режим «монитора слов», mt_dsr – режим «монитор сообщений».

Структура регистра isr[15..0] (адрес 00110; чтение):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mi	rp_er	bt_tt	bt_csr	mt_csr	mt_dsr	hf	bc_r	rt_ape	tt_rol	rt_cbr	scw_e	bc_eof	fe	ss_mc_pt	eom

Наименования разрядов регистра:

mi (Master Interrupt), общий флаг запроса прерывания УВ;

rp_er (Ram Parity Error), ошибка чётности при доступе к ОЗУ;

bt_tt (BC/RT transmiter Timeout), затягивание работы передатчика;

bt_csr (BC/RT Command Stack Rollover), ‘прокручивание’ стека команд КШ/ОУ;

mt_csr (MT Command Stack Rollover), ‘прокручивание стека команд монитора сообщений;

mt_dsr (MT Data Stack Rollover), ‘прокручивание’ буфера данных МТ;

hf (Handshake Failure), истечение времени ожидания подтверждения (квитирования);

bc_r (BC Retry), ВПС в режиме КШ;

rt_ape (RT Address parity Error), ошибка контроля по четности адреса ОУ;

tt_rol (Time Tag Rollover), переполнение счетчика метки времени;

rt_cbr (RT Cirular Buffer Rollover), ‘прокручивание’ кольцевого буфера данных ОУ;

scw_e (RT Subaddress Control Word EOM/BC Selective EOM), завершение обработки избранного сообщения для режимов КШ/ОУ;

bc_eof (BC End Of Frame), завершение обработки кадра сообщений;

fe (Format Error), ошибка формата сообщения;

ss_mc_pt (BC Status Set/ RT Mode Code/ MT Pattern Trigger),

- для режима КШ: особое состояние ОУ;
- для режима ОУ: принята избранная команда управления;
- для режима «монитор слов»: принятое КС совпало со словом срабатывания;

eom (End Of Message), завершение отработки сообщения.

Функциональное описание разрядов регистра:

mi (Master Interrupt), – разряд общий флаг запроса прерывания УВ:

Лог. «1» указывает на то, что запрос прерывания сигналом **msp_int** для УВ был заявлен. Если «расширенные возможности прерывания» разрешены (15-ый разряд регистра **cfg2_** в лог. «1»), то **mi** будет возникать только тогда, когда какой-либо из младших остальных 15-ти разрядов регистра isr[15..0] оказывается установленным в единичное состояние.

rp_er (Ram Parity Error), – разряд ошибка чётности при доступе к ОЗУ:

Разряд **rp_er** не задействован в МСП. Этот разряд может быть использован только в «режиме расширенных возможностей» (**eme**, 15-ый разряд регистра **cfg3_** установлен в лог. «1»), и если «разрешены расширенные возможности прерывания» (**ei**, 15-ый регистра **cfg2_** установлен в лог. «1»). Если разряд **rp_er** установлен в лог. «1», то это указывает на ошибку чётности.

bt_tt (BC/RT transmiter Timeout), – разряд затягивание работы передатчика:

Разряд **bt_tt** может быть использован только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»), и если «расширенные возможности прерывания» разрешены (**ei**, разряд 15 регистра **cfg2_**, установлен в лог. «1»).

Если разряд **bt_tt** установлен в лог. «1», то произошло срабатывание сторожевого устройства, отслеживающего истечение максимально допустимого времени непрерывной работы передатчика МСП. Это происходит, если попытка кодирования и выдачи данных в ЛПИ затягивается на время, большее чем 668 мкс. Прерывание вырабатывается только в режимах КШ или ОУ.

bt_csr (BC/RT Command Stack Rollover), – разряд ‘прокручивание’ стека команд КШ/ОУ:

Разряд **bt_csr** может быть использован только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»), и если «расширенные возможности прерывания» разрешены (**ei**, разряд 15 регистра **cfg2_**, установлен в лог. «1»).

Если разряд **bt_csr** установлен в лог. «1», значит произошло ‘прокручивание’ командного стека КШ или ОУ. Размеры стека программно выбираются из ряда следующих величин 256 слов (64 сообщений), 512, 1024 или 2048 (512 сообщений) посредством разряда 14 и 13 регистра **cfg3_**.

mt_csr (MT Command Stack Rollover), – разряд ‘прокручивание стека команд монитора сообщений:

Разряд **mt_csr** может быть использован только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_,** установлен в лог. «1»), и если «расширенные возможности прерывания» разрешены (**ei**, 15-ый разряд регистра **cfg2_**, установлен в лог. «1»).

Если разряд **mt_csr** установлен в лог. «1», значит произошло ‘прокручивание’ стека команд монитора сообщений. Применяется как для режима монитора сообщений, так и для комбинированного режима работы «ОУ/ монитор сообщений». Размер командного стека выбирается среди 256 (64 сообщения), 1024, 4096 и 16384 слов (4096 сообщений) посредством разрядов 12 и 11 регистра **cfg3_.**

mt_dsr (MT Data Stack Rollover), – разряд ‘прокручивание’ буфера данных МТ:

Разряд может быть использован только в состоянии «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_,** установлен в лог. «1»), и если «расширенные возможности прерывания» разрешены (**ei**, 15-ый разряд регистра **cfg2_,** установлен в лог. «1»).

Если разряд mt_dsr установлен в лог. «1», значит произошло ‘прокручивание’ буфера данных в режиме «монитора слов» или «монитора сообщений». Размер буфера данных выбирается среди 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536 слов посредством разрядов 10, 9, и 8 регистра cfg3_.

hf (Handshake Failure), – разряд истечение времени ожидания подтверждения (квитирования):

Данный разряд не задействован в МСП.

bc_r (BC Retry), – разряд ВПС в режиме КШ:

Разряд может быть использован только в состоянии «расширенных возможностей конфигурирования»(eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»), и если «расширенные возможности прерывания» разрешены (**ei**, 15-ый разряд регистра **cfg2_**, установлен в лог. «1»).

Если разряд **bc_r** установлен в лог. «1», то это указывает на происшествия ВПС в режиме КШ.

Прерывание будет происходить, если оно разрешено, вне зависимости от того, успешно ли завершилось ВПС или нет. Прерывание будет порождаться после повторной попытки переслать неудачей завершенную прежнюю посылку, вне зависимости от номера ВПС (первая или вторая) и от её успешности.

rt_ape(RT Address parity Error), – разряд ошибка контроля по четности адреса ОУ:

Значение разряда rt_ape в лог. «1», указывает на чётное значение суммы разрядов **msp_rtad**[4:0] и msp_rtadp, а не нечётное, как это требуется для нормальной реакции ОУ на сообщения, направляемые по его собственному адресу. То есть сумма лог. значений уровней сигналов имеющихся на входных сигналах **msp_rtad**[4:0] и msp_rtadp имеет четное значение.

Следует отметить, что, если ошибка контроля по чётности адреса ОУ происходит, соответствующий разряд в регистре состояния прерываний будет очищен в лог. «0» после того, как в разряд «сброса прерывания» регистра запуска/сброса была прописана лог. «1».

Разряд **rt_ape** будет удерживаться в состоянии лог. «0» и не вернётся к лог. «1» до той поры, пока не произойдёт одна из следующих последовательностей событий:

Условие, вызывающее возникновение ошибки чётности исчезает, а за тем появляется вновь, или

разряд **m_rt_ape** в регистре маскирования прерывания очищается (либо прописью в него нуля, либо программным, либо аппаратным сбросом МСП) и затем снова возвращается в лог. «1», а условия, вызывающее возникновение ошибки чётности, продолжают при этом сохраняться.

tt_rol (Time Tag Rollover), – разряд переполнение счетчика метки времени:

Если установлен в лог. «1», указывает на происшествие «прокручивания» счётчика метки времени - переход значения 16-ти разрядного регистра счетчика метки времени из FFFF в 0000.

rt_cbr (RT Circular Buffer Rollover), – разряд ‘прокручивание’ кольцевого буфера данных ОУ:

Этот разряд будет выставляться в лог. «1» в режиме ОУ в конце сообщения при следующем условии:

- МСП находится в режиме «расширенные возможности управления памятью ОУ» (разряд 1 регистра **cfg2_** установлен в лог. «1»);

- и разряд «‘прокручивание’ - запрос прерывания» в соответствующем слове управления подадресом ОУ для соответствующего КС (передачи/приёма/группового) только что выполненного сообщения установлен в лог. «1»;

- и текущее сообщение вызывает событие ‘прокручивание’.

‘Прокручивание’- ситуация, когда соответствующий указатель адреса, взятый из поисковой таблицы, после попытки пересечь максимальную (и кратную заданному размеру буфера) по значения адреса границу кольцевого буфера возвращается к значению, в котором переходят из лог. «1» в лог. «0» младшие разряды, соответствующие заданному размеру данного буфера.

В общем случае, ‘прокручивание’ не обязательно приводит к начальному значению указателя, который может содержать значение любого адреса в границах выделяемой под буфер области памяти. Например, начальное значение указателя равное 9a77h задаёт выделение под буфер, скажем в 512 слов, области ОЗУ с адресами от 9a00h до 9affh.

Если разряд 11 регистра **cfg2_** («переписывать недостоверные данные») находится в лог. «0», то запрос прерывания УВ будет порождаться непосредственно сразу после того, как последнее, непосредственно прилегающее к максимальной (и кратной заданному размеру буфера) по значения адреса границе кольцевого буфера ячейка памяти, слово было прочитано или записано.

Если разряд «переписывать недостоверные данные» (разряд 11 регистра **cfg2_**) установлен в лог. «1»,то запросы прерывания будут происходить в конце отработки сообщений «передачи» или достоверных сообщений «приема», в которых слово, прилегающее к максимальной по значения адреса границе кольцевого буфера в адресном пространстве ОЗУ, было прочитано/записано. Для сообщений «приема» запросы прерывания не будут порождаться, если разряд 11 регистра **cfg2_** («переписывать недостоверные данные») установлен в лог. «1» и произошла ошибка сообщения, даже в случае доступа к последнему (с граничным местоположением) слову буфера.

scw_e (RT Subaddress Control Word EOM/BC Selective EOM), – разряд завершение обработки избранного сообщения КШ/ОУ

Для режима КШ этот разряд оказывается установленным в лог. «1» по завершении обработки сообщения при условии, что МСП находится в состоянии «расширенных возможностей конфигурирования», и разряд «разрешение полного слова управления сообщением КШ» (12-ый разряд регистра cfg4_) установлен в лог. «1», и «разрешение прерывания по ЕОМ» (4-ый разряд соответствующего слова управления сообщением КШ) установлен в лог. «1».

Для режима ОУ этот разряд оказывается установленным после завершения сообщения при условии, что МСП находится в состоянии «расширенное управление памятью ОУ» и разряд «прерывание по ЕОМ» в слове управления подадресом ОУ» для соответствующего КС (передачи/приёма/группового) установлен в лог. «1».

bc_eof(BC End Of Frame), – разряд завершение обработки кадра сообщений:

Указывает на завершение обработки последнего сообщения кадра, когда очередное увеличение на единицу счетчика сообщений области А/Б привело его к состоянию максимально возможного значения - FFFF(шестн.). Если отработка кадра КШ прекращена прежде достижения счетчиком сообщений значения FFFF, то прерывания **bc_eof** не произойдет. Заметим, что если задан останов работы после окончания текущего сообщения (в состоянии «расширенных возможностей конфигурирования» установлен в лог. «1» один из разрядов 10, 12 регистра **cfg1_**) и событие произошло, то кадр завершается прежде достижения счетчиком сообщений значения FFFF(шестн.).

fe (Format Error), – разряд ошибка формата сообщения:

Указывает, что завершённое сообщение (в режимах КШ, ОУ и «Монитор сообщений») содержало одну из следующих ошибок:

«ошибка проверочного возврата»: проверка ‘возвратившегося’ слова производится для каждого слова, переданном МСП (режимы ОУ и КШ), для каждого сообщения. ‘Возврат’ передаваемого в ЛПИ слова осуществляется: а) в режиме внутреннего самотестирования (разряд olst слова управления сообщением КШ в лог. «1») - непосредственной пересылкой передаваемого слова от выхода кодера на вход декодера; б) вне режима внутреннего самотестирования (разряд olst слова управления сообщением КШ в лог. «0») – путем приёма ППУ передаваемого слова и последующим его декодированием. Если принятая версия одного или более слов расшифрована декодером как недостоверная и/или принятая версия последнего слова, переданного в сообщении, не равна исходной (посланной) версии, то в этом случае фиксируется «ошибка проверочного возврата».

«ошибка в сообщении»: принятое сообщение содержит нарушения критериев достоверности принятой информации по ГОСТ Р 52070-2003 – недостоверное слово (тип синхросигнала, количество разрядов, четность и т.п), неверное количество слов, ошибка межсловного промежутка времени, некорректный адрес ОУ в ОС. Для режима ОУ ситуация «ошибка формата сообщения» не учитывает наличие ошибки в принятом КС, т.к. если обнаруживается недостоверное командное слово, то сообщение полностью игнорируется.

«Истечение времени ожидания ОС»: В режимах КШ или «Монитора сообщений» это происходит, когда ОУ либо не откликнулось на сообщение, либо откликнулось после истечения программно устанавливаемой длительности «выбора интервала ожидания ОС». Это время для КШ и ОУ (при передаче «ОУ-ОУ») может быть выбрано посредством разрядов 10 и 9 регистра **cfg5_**. В режиме принимающего ОУ истечение времени ожидания ОС происходит, если в передаче «ОУ-ОУ» передающее ОУ не отозвалось ОС в установленный для этого срок.

Пауза перед выдачей ОС меньше 4 мкс. и установлен режим «разрешение расширенных возможностей конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «1») и «разрешение проверки минимальной паузы перед выдачей ОС» (**gce**, 8-ой разряд регистра **cfg5_** в логической «1»). Согласно ГОСТ Р 52070-2003, 4 мкс. - это промежуток времени между серединами зон разряда контроля по четности последнего КС/СД и синхроимпульса ОС.

ss_mc_pt (BC Status Set/ RT Mode Code/ MT Pattern Trigger), смысл разряда зависит от режима работы:

- для режима КШ: ‘особое состояние ОУ’ или ОС содержит неправильный «Адрес ОУ» в адресном поле;
- для режима ОУ: принята избранная команда управления;
- для режима «монитор слов»: принятое КС совпало со словом срабатывания (заданным в регистре слова срабатывания).

В режиме КШ - ОС, полученное от отвечающего абонента, либо содержит неправильный адрес ОУ в адресном поле, либо наступило событие ‘особое состояние ОУ’ т.е. один из одиннадцати признаков/разрядов ОС содержит (имеет) неожидаемое значение. Ожидаемыми значениями для этих одиннадцати признаков являются лог. «0» за следующими исключениями:

1. Признаки ОС установленные в лог. «1» замаскированы соответствующими разрядами слова управления сообщением КШ.
2. В некоторых состояниях конфигурации МСП ожидаемым значением для признака ОС «Принята групповая команда» становится лог. «1», а не лог. «0». Это последнее исключение верно для каждого из следующих состояний конфигурации МСП:

1) МСП находится в состоянии «нерасширенные возможности конфигурирования» (15-ый разряд (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «0»);

2) МСП находится в состоянии «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»),

и

разряд bm_ex регистра cfg4_ установлен в лог. «0»,

и

разряд «маскирование групповых сообщений» в слове управления сообщением КШ установлен в лог. «1».

В режиме ОУ этот запрос прерывания УВ будет формироваться при завершении приёма избранной КУ. Оно может происходить только в состоянии «расширенных возможностей конфигурирования» МСП (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1») и, если разрешена «расширенная обработка КУ» (emch, 0-ой разряд регистра **cfg3_**, установлен в лог. «1»). Если эти два разряда установлены, то прерывание для любой из КУ может быть разрешено программированием в лог. «1» предназначенного для этой команды разряда в соответствующем слове из области разделяемой памяти с адресами 0108-010F (таблица прерываний для избранных КУ). Получение заданной КУ будет вызывать со стороны МСП порождение запроса прерывания УВ в конце обработки КУ.

В режиме «монитор слов» этот запрос прерывания УВ будет формироваться только в состоянии «расширенных возможностей конфигурирования» МСП (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «1»), когда принятое КС совпало со словом срабатывания (заданным в регистре слова срабатывания). Принятое достоверное командное слово должно поразрядно совпасть со значением, заданным программно в регистре слова срабатывания.

eom (End Of Message), – разряд «завершение отработки сообщения»:

Если установлен в лог. «1», указывает (в режимах КШ, ОУ и «Монитор сообщений») на завершение отработки сообщения вне зависимости от наличия «ошибки формата сообщения».

1.3.12.2.2.4.1.9 Третий конфигурационный регистр - cfg3_

Адрес регистра 00111; чт/зп.

Структура регистра cfg3_[15..0]:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
eme	bt_csz[1..0]		mt_csz[1..0]		mt_dsz[2..0]			id	ome	aswe	i_rx_td	b_rx_td	rt_f_we	1553a_mce	emch

Наименования разрядов регистра:

eme (Enhanced Mode Enable), разрешение расширенных возможностей конфигурирования;

bt_csz[1..0] (BC/RT Command Stack size), размер стека команд КШ/ОУ;

mt_csz[1..0] (MT Command Stack size), размер стека команд монитора сообщений;

mt_dsz[2..0] (MT data Stack size), размер буфера данных монитора;

id (Illegalization Disabled), запрещение в ОУ проверки КС на недопустимость;

ome (Override Mode T/R Error), обрабатывать КУ на «прием» без СД как резервные КУ;

aswe (Alternate Status Word Enable), разрешение альтернативного формирования ОС;

i_rx_td (Illegal RX transfer Disable), запрещение сохранения принимаемых СД данных недопустимого сообщения;

b_rx_td (Busy RX transfer Disable), запрещение сохранения принимаемых СД в состоянии ‘абонент занят’;

rt_f_we (RT Fail/Flag Wrap Enable), управление формированием признака «Неисправность ОУ» в ОС;

1553a_mce (1553A Mode Code Enable), включение протокольных требований MIL_STD_1553A;

emch (Enhanced Mode Code Handling), расширенная обработка КУ.

Функциональное описание разрядов регистра:

eme (Enhanced Mode Enable), – разряд разрешение расширенных возможностей конфигурирования:

Если этот разряд запрограммирован в состояние лог. «0», функциональные возможности МСП ограничиваются, МСП находится в состоянии «нерасширенные возможности конфигурирования». Занесение в этот разряд лог. «1» позволяет использование многих дополнительных (расширенных) свойств МСП, которые разрешаются посредством регистра маскирования прерываний, различных конфигурационных регистров, а также слова управления сообщением КШ и слова управления подадресом ОУ. Эти свойства включают всевозможные функции, задаваемые различными разрядами регистров **cfg3_**, **cfg4_**, **cfg5_**, **cfg6_**.

Следует отметить, что для того, чтобы разрешить какое-либо «расширенное свойство», разряд eme должен быть запрограммирован в лог. «1» перед установкой каких-либо других разрядов в конфигурационных регистрах, разрешающих эти «расширенные свойства». Это относится к свойствам, которые задаются 15-тью младшими разрядами (14-0) самого регистра **cfg3_**, а также разрядами других регистров. Особый случай - работа с регистром **cfg3_**: 15-ый разряд сначала должен быть запрограммирован в лог. «1» записью 8000 (шестн.), а затем уже разрешается приступить к установке какого-либо из младших разрядов (14-0). Установка этих разрядов должна сопровождаться прописью в eme лог. «1».

В состоянии «нерасширенные возможности конфигурирования» (15-ый разряд регистра **cfg3_** установлен в лог. «0») разрешается использовать функции, задаваемые следующими разрядами регистров:

- Регистр маскирования прерываний imr и регистр состояния прерываний isr, разряды 7-0;
- Регистр cfg1_: для режима КШ - разряды 15-12; для режима ОУ (без альтернативного «ответного ОС») - разряды 15-13 и 11-8; для «монитор слов» - разр. 15-13;
- Регистр cfg2_ - разряды 11-0;
- Регистр srr - разряды 4-0;
- Указатель на стек команд КШ/ОУ;
- Регистр слова управления подадресом ОУ;
- Регистр счетчика метки времени;
- Регистр состояния прерывания - разряды 15 и 7-0.
- Регистр cfg5_ - разряды 5-0 (только чтение);
- Регистр указателя буфера данных ОУ/МТ;
- Регистр командного слова сообщения ОУ
- Регистр ответного слова ОУ;
- Регистр слова ВСК;

В состоянии «расширенных возможностей конфигурирования» дополнительно ко всем указанным выше функциям могут быть задействованы и другие функции, которые разрешаются следующими регистрами или регистровыми разрядами:

- Регистры isr и imr - разряды 14-8;
- Регистр cfg1_: режим КШ - разряды 11-0;
- режим ОУ - разряды 12 и 7-0, и ОУ с альтернативным ОС;
- монитор, разряды 12-9,7 2-0;
- «монитор сообщений» и «ОУ/монитор сообщений»;
- Регистр cfg2_ - разряды 15-12; регистр srr - разряды 6-5; cfg3_, cfg4_, cfg5_, cfg6_;
- Регистр указателя буфера данных ОУ/МТ;
- Регистр остатка времени кадра КШ;
- Регистр остатка времени текущего сообщения КШ;
- Регистр длительности кадра КШ/ Регистр командного слова сообщения ОУ/ Регистр слова срабатывания («монитор слов»);
- Регистр ответного слова ОУ;
- Регистр слова ВСК.

Для всех трех режимов (КШ/ ОУ/«монитор сообщений») состояние «расширенных возможностей конфигурирования» включает в работу (задействует) различные доступные лишь по чтению разряды (разряды 2-0) в регистре cfg1_.

МСП должен находиться в состоянии «расширенных возможностей конфигурирования» для того, чтобы разрешить использование «расширенных возможностей прерываний», устанавливаемых 15-ым разрядом регистра cfg2_.

В режиме КШ перечень свойств, которые требуют установления разряда eme в лог. «1», включает в себя: полное слово управление сообщением КШ и слово состояния блока сообщения КШ, дополнительные функции - останов при обнаружении ошибки, останов при возникновении события ‘особое состояние ОУ’, автоматическое повторение кадра, программируемый промежуток времени между началом отработки данного и началом отработки следующего сообщения, ВПС, полное маскирование признаков ОС, способность вырабатывать запрос на обработку прерывания после отработки какого-либо избранного сообщения.

Для режима ОУ перечень свойств, которые требуют установления разряда eme в лог. «1», включает в себя: полное слово состояния блока сообщения ОУ, комбинированный режим работы «ОУ/монитор сообщений», зависимость признака «Неисправность ОУ» ОС от длительности непрерывной работы передатчика или результата «проверочного возврата», схема «двойной буферизации сообщений «приема»» для индивидуальных принимающих (групповых)

подадресов, альтернативное управление (полная программная доступность) признаков ОС, программирование признака «Абонент занят» ОС с помощью таблицы занятости подадресов.

Для режима МТ состояние «расширенных возможностей конфигурирования» разрешает использование как монитора сообщений, так и комбинированного режима работы «ОУ/монитор сообщений», и, кроме того, использование слова срабатывания в режиме монитора слов.

bt_csz[1..0] (BC/RT Command Stack size), – разряд размер стека команд КШ/ОУ:

Эти два разряда определяют размер стека команд КШ/ОУ. По умолчанию оба разряда установлены в логические нули, что соответствует размеру стека в 256 слов (64 сообщения).

Таблица 1.621 показывает соответствия между различными значениями разрядов bt_csz[1..0] и задаваемыми ими размерами стека команд КШ/ОУ:

Таблица 1.621 – Размер стека команд КШ/ОУ

Разряд 14	Разряд 13	Размер стека КШ/ОУ (слов)
0	0	256 (64 блоков описателей сообщения)
0	1	512 (128 блоков описателей сообщений)
1	0	1024 (256 блоков описателей сообщений)
1	1	2048 (512 блоков описателей сообщений)

mt_csz[1..0] (MT Command Stack size), – разряд размер стека команд монитора сообщений:

Эти два разряда определяют размер стека команд монитора сообщений. По умолчанию значением этих двух разрядов является лог. «0», что соответствует размеру в 256 слов. Таблица 1.622 показывает соответствия между различными значениями разрядов mt_csz[1..0] и задаваемыми ими размерами стека команд монитора сообщений:

Таблица 1.622- Размер стека команд монитора сообщений

Разряд 12	Разряд 11	Размер стека (слов)
0	0	256 (64 сообщения)
0	1	1024 (256 сообщений)
1	0	4096 (1024 сообщений)
1	1	16384 (4096 сообщений)

mt_dsz[2..0] (MT data Stack size), – разряд размер буфера данных монитора:

Эти три разряда выбирают размер буфера данных мониторного. Этот стек используется в обоих режимах монитора – «слов» и «сообщений». По умолчанию значение этих трех разрядов оказывается установленным в лог. «0», что задает размер стека 65536 слов. Таблица 1.623 показывает соответствия между различными значениями разрядов mt_dsz[2..0] и задаваемыми ими размерами буфера данных мониторного:

Таблица 1.623– Размер буфера данных монитора

Разряд 10	Разряд 9	Разряд 8	размер буфера данных монитора (слов)
0	0	0	65_536 (по умолчанию)
0	0	1	32_768
0	1	0	16_384
0	1	1	8_192
1	0	0	4_096
1	0	1	2_048
1	1	0	1_024
1	1	1	512

id (Illegalization Disabled), – разряд запрещение в ОУ проверки КС на недопустимость:

Предназначен для режима ОУ.

Если запрограммирован в лог. «0», свойство «определения допустимости КС» МСП разрешено. То есть, область памяти с адресами от 0300 до 03FF (шестн.) выделяется таблице допустимости КС. Если этот разряд установлен в лог. «1», функция допустимости КС запрещена. В этой конфигурации адреса с 0300 по 03FF могут быть употреблены для размещения стека команд или буфера данных сообщений. Этот разряд не имеет функционального смысла в режимах КШ или МТ.

ome (Override Mode T/R Error), – разряд обрабатывать КУ на «прием» без СД как резервные КУ:

Разряд применим только для режима ОУ и «монитор сообщений».

КУ приёма с 5’b00000 по 01111 не определены в ГОСТ Р 52070-2003.

Если ОУ МСП не в режиме «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») или, если разряд «обрабатывать КУ на «прием» без СД как резервные КУ» для КУ с кодом от 5’b00000 до 5’b01111 и разрядом «Приём-передача» равным лог. «0»» (ome, 6-ой разряд регистра cfg3_) установлен в лог «0», то ОУ МСП не будет откликаться на сообщение и признак (разряд) «Ошибка в сообщении» ОС будет устанавливаться в лог. «1».

Если МСП находится в режиме «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») и разряд «обрабатывать КУ на «прием» без СД как резервные КУ» для КУ с кодом от 5’b00000 до 5’b01111 и разрядом «Приём-передача» равным лог. «0»» установлен в лог «1», то ОУ МСП будет обращаться с такими сообщениями как с «резервными КУ». То есть, откликаться на них ОС с установленным в лог. «0» признаком «Ошибка в сообщении» (если команда не была определена недопустимой).

Режим «монитор сообщений». Если принятое КС с разрядом «Приём-передача» в лог. «0», со значением поля «Подадрес/Режим управления» равным 5’b00000 или 5’b11111 (т.е. КУ) и значение поля «Число СД/Код команды» между 5’b00000 или 01111 и разряд ome («обрабатывать КУ на «прием» без СД как резервные КУ») регистра cfg3_ установлен в лог. «0», то «монитор сообщений» установит в лог. «1» разряд cwce («Принятое КС не согласуется с требованиями ГОСТ Р 52070-2003») в слове состояния блока сообщения на данное сообщение.

aswe (Alternate Status Word Enable, – разряд разрешение альтернативного формирования ОС:

Для режима ОУ; если запрограммирован в лог. «0», то только признаки, «Абонент занят», «Запрос на обслуживание», «Неисправность абонента», «Неисправность ОУ» в ОС находятся под управлением УВ, что обеспечивается разрядами с 11 по 7 регистра cfg1_. Когда этот разряд запрограммирован в лог. «1», процессор УВ получает возможность программировать значение любого из 11 признаков ОС посредством разрядов с 11 по 1 регистра cfg1_.

i_rx_td (Illegal RX transfer Disable), – разряд запрещение сохранения принимаемых СД данных недопустимого сообщения:

Если установлен в лог. «0» (по умолчанию), и МСП получает команду «приём», которая была оценена как «недопустимая», МСП будет сохранять поступающие данные в разделяемом ОЗУ. Если в единице, и МСП получает команду «приём», которая, как выясняется затем в ходе анализа, является «недопустимой», то МСП не будет сохранять поступающие данные в разделяемое ОЗУ.

b_rx_td (Busy RX transfer Disable), – разряд запрещение сохранения принимаемых СД в состоянии ‘абонент занят’:

Если данный разряд установлен в лог. «0» (что происходит по умолчанию), и УВ записал лог. «0» в разряд busy регистра cfg1_, или для частного, принятого по ЛПИ, командного слова «Принять данные» была выявлена лог. «1» в соответствующем разряде таблицы занятости подадресов, то МСП будет отвечать установлением лог. «1» в признаке «Абонент занят» ОС и будет сохранять принимаемые данные в памяти.

Если данный разряд установлен в лог. «1», и УВ записал лог. «0» в разряд busy регистра cfg1_, или для частного, принятого по ЛПИ, командного слова «Принять данные», была выявлена лог. «1» в соответствующем разряде таблицы занятости подадресов, то МСП будет

отвечать установлением лог. «1» в признаке «Абонент занят» ОС и не будет сохранять принимаемые данные в памяти.

rt_f_we (RT Fail/Flag Wrap Enable), – разряд управление формированием признака «Неисправность ОУ» в ОС.

Имеет смысл только в режиме ОУ.

Если данный разряд запрограммирован в лог. «0» (по умолчанию так и случается), признак «Неисправность ОУ» в ОС находится в исключительном, полном распоряжении УВ посредством регистра cfg1_. Если этот разряд установлен в лог. «1», то признак «Неисправность ОУ» в ОС подпадает также под зависимость условий истечения дозволенного промежутка времени непрерывной работы передатчика (668 мкс) и/или обнаружена «ошибка проверочного возврата» для предыдущего негруппового сообщения.

1553a_mce (1553A Mode Code Enable), – разряд включение протокольных требований MIL_STD_1553A:

Данный разряд не задействован в МСП. То есть эл. цепь, задающая вид протокола (1553 A или 1553 B), внутри МСП жёстко, аппаратно, подсоединена не к выходу данного триггера, а к уровню лог. «0», поэтому ОУ МСП всегда распознает и откликается на все режимные коды в соответствии с требованиями стандарта MIL_STD_1553B (ГОСТ Р 52070-2003). При чтении разряд 1553a_mce будет возвращать записанное в него ранее, наиболее свежее значение.

emch (Enhanced Mode Code Handling), – разряд расширенная обработка КУ:

Имеет смысл только для режима ОУ.

Если установлен в лог. «0», способность МСП вырабатывать «запрос прерывания» при получении особенной, специфицированной КУ отсутствует. Если разряд в лог. «0», прерывания для КУ могут разрешаться глобально - для любых «передающих» и/или «принимающих» КУ с подадресами 0 и/или 31. Кроме того, если emch в лог. «0», то слова данных для любых КУ распределяются на одни те же местоположения в ОЗУ для «принимающих» и «передающих» (и групповых, отделяемых по желанию) подадресов 0 и 31. Два исключения составляют КУ ‘Передать последнюю команду’ и ‘Передать слово ВСК ОУ’, в которых слова данных извлекаются из внутренних регистров МСП.

Если этот разряд установлен в лог. «1», открывается способность запрашивать прерывания после принятия сообщений с отдельными, специфицированными (избранным) КУ. Разрешение прерывания для избранных КУ является функцией от признака «групповая/негрупповая», разряда «Передать/Принять» и поля «код команды» в принятой КУ. Задание (процессором УВ) отдельных, специфицированных КУ при конфигурировании, а в последствии и автоматическое, самостоятельное определение их МСП, осуществляется с помощью специальной таблицы прерываний для избранных КУ с адресами расположения в ОЗУ с 0108 по 010F.

Ко всему прочему, если разряд emch установлен в лог. «1», слова данных для КУ распределяются по области ОЗУ в таблице слов данных КУ в зависимости от признаков «прием/передача», «групповая/негрупповая» и младших четырех разрядов в поле «число СД/код команды» (предполагается, что 5-ый разряд поля «число СД/код команды» должен находиться в лог. «1» для КУ с данными). Если emch в лог. «1», то адресные местоположения в ОЗУ с 0110 по 013F употребляются исключительно для сохранения слов данных, пересылаемых в сообщениях совместно с КУ.

Если emch в лог. «0», то в случае сообщения КУ указатель на однословный блок данных из таблицы указателей для подадресов будет сохраняться в третьей позиции описателя блока сообщения ОУ в стеке команд ОУ. Если разряд emch установлен в лог. «1», то принимаемое или передаваемое слово данных, непосредственно входящее в сообщение КУ (а не указатель на него) будет сохраняться в третьей позиции описателя блока сообщения ОУ в стеке команд. Если emch в лог. «1», то никакая запись не будет производиться в третью позицию описателя блока сообщения ОУ для КУ без слов данных.

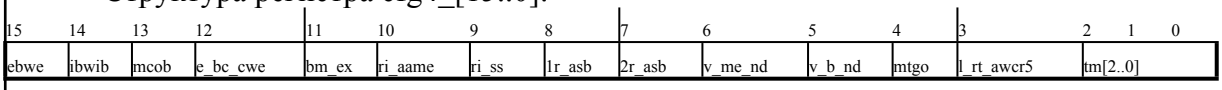
Следует отметить, что «расширенная обработка КУ» не влияет на слова данных, пересылаемые в откликах на получение КУ ‘Передать последнюю команду’. Слово данных, при

отзыве МСП на эту команду, представляет собою предыдущее командное слово, полученное ОУ, и всегда извлекается из внутреннего регистра МСП, а не из местоположения в ОЗУ.

1.3.12.2.2.4.1.10 Четвертый конфигурационный регистр - cfg4_

Адрес регистра 01000; чт/зп.

Структура регистра cfg4_[15..0]:



Наименования разрядов регистра:

ebwe (External Bit Word Enable), разрешение внешнего слова ВСК;

ibwib (Inhibit Bit Word IF busy), препятствовать передаче слова ВСК при ‘абонент занят’;

mcob (Mode Cobe Override Busy), игнорирование ‘абонент занят’ для КУ;

e_bc_cwe (Expanded BC Control Word Enable), разрешение полного слова управления сообщением КШ;

bm_ex (Broadcast Mask ENA/XOR), разрешение/XOR* маскирования признака «Принята групповая команда»;

ri_aame (Retry If 1553A And M.E.), ВПС, если 1553A и ситуация ‘ошибка в сообщении’;

ri_ss (Retry If Status Set), ВПС при событии ‘особое состояние ОУ’;

1r_asb (1st Retry Alt/Same Bus), выбор канала для 1-го ВПС;

2r_asb (2nd Retry Alt/Same Bus), выбор канала для 2-го ВПС;

v_me_nd (Valid M.E./No Data), разрешение ОУ не передавать СД в ситуации ‘ошибка в сообщении’;

v_b_nd (Valid Busy/No Data), разрешение ОУ не передавать СД в ситуации ‘абонент занят’;

mtgo (MT Tag Gap Option), модификации формирования ID - слова в режиме монитор слов;

l_rt_awcr5 (Latch RT Addr With Config. Reg. #5), защёлкивание адреса ОУ процедурой записи в регистр cfg5_;

tm[2..0] (Test Mode), задание проверочных режимов работы МСП.

Примечание - МСП должен быть установленным в состояние «расширенных возможностей конфигурирования» (**eme**, 15-ый разряд регистра **cfg3_**, должен быть установлен в лог. «1») пред попытками активизировать какую-либо функцию, разрешаемую регистром **cfg4_**.

Функциональное описание разрядов регистра:

ebwe (External Bit Word Enable), – разряд разрешение внешнего слова ВСК:

Имеет смысл только для режима ОУ. Если запрограммирован в лог. «0», ОУ МСП будет отзываться, отвечая на КУ ‘Передать слово ВСК ОУ’, информацией, извлекаемой из внутреннего регистра слова ВСК. Если запрограммирован в лог. «1», МСП будет брать «слово ВСК» из местоположения в ОЗУ. В последнем случае «слово ВСК» должно записываться в память управляющим вычислителем. Адрес расположения «слова ВСК» в ОЗУ следующий:

Если emch (0-ой разряд регистра **cfg3_** - «расширенная обработка КУ») запрограммирован в лог. «0», слово ВСК будет читаться из места, на которое указывает содержимое ячейки в таблице указателей для подадресов на «передачу» для подадресов 0 и 31. Для этой конфигурации МСП следует отметить, что слово данных в отзыве ОУ на КУ ‘Передать векторное слово’ будет читаться из той же самой пары адресных местоположений, в которой сохраняется внешнее слово ВСК.

Если оба разряда **ebwe** и **emch** установлены в логические единицы, слово ВСК будет читаться из фиксированного местоположения в ОЗУ по адресу 0123.

ibwib (Inhibit Bit Word IF busy), – разряд препятствовать передаче слова ВСК при ‘абонент занят’:

Имеет смысл только для режима ОУ.

Если ibwib запрограммирован в лог. «0», и либо бит busy - «инверсное значение признака ОС «Абонент занят»» (10-ый разряд регистра **cfg1_**) находится в положении лог. «0», либо бит «разрешение использования таблицы занятости подадресов» (**blute**, 13-ый разряд регистра **cfg2_**) в лог. «1», а соответственные разряды в таблице занятости подадресов (разряд 0 местоположения 0242 и/или разряд 15 местоположения 0243) запрограммированы в лог. «1», то МСП будет отзываться на КУ ‘Передать слово ВСК ОУ’ ОС с установленным в лог. «1» признаком «Абонент занят», после чего следует передача внутреннего или внешнего слова ВСК.

Если ibwib запрограммирован в лог. «1», и либо бит **busy** - «инверсное значение признака ОС «Абонент занят»» (10-ый разряд регистра **cfg1_**) находится в положении лог. «1», либо соответственный (принадлежащий сообщению) разряд таблицы занятости подадресов в лог. «1», то МСП будет откликаться ОС, в котором признак «Абонент занят» будет установлен в лог. «1», но слово данных при этом посылаться в ЛПИ не будет.

mcob (Mode Cobe Override Busy), – разряд игнорирование ‘абонент занят’ для КУ:

Имеет смысл только для режима ОУ.

Если mcob запрограммирован в лог. «0», и либо бит busy - «инверсное значение признака ОС «Абонент занят»» (10-ый разряд регистра **cfg1_**) находится в положении лог. «0», либо бит «разрешение использования таблицы занятости подадресов» (**blute**, 13-ый разряд регистра **cfg2_**) в лог. «1», а соответственные разряды в таблице занятости подадресов (0-ой разряд в местоположении 0242 и/или 15-ый в местоположении 0243) запрограммированы в лог. «1», то МСП будет передавать только ОС с признаком «Абонент занят», установленным в лог. «1». МСП не станет отправлять данные в своих откликах на получение команды ‘Предать векторное слово’ или ‘Резервный код команды передачи’ (передающие коды команд 5’b10110-5’b11111).

Если mcob запрограммирован в лог. «1», и либо бит busy - «инверсное значение признака ОС «Абонент занят»» (10-ый разряд регистра **cfg1_**) находится в положении лог. «1», либо бит blute (13-ый разряд регистра **cfg2_**) в лог. «1», а соответственный разряд в таблице занятости подадресов (0-й разряд в местоположении с адресом 0242 и/или 15-ый разряд в местоположении 0243) запрограммирован в лог. «1», то МСП в своих откликах на КУ ‘Передать векторное слово’ или ‘Резервный код команды передачи’ (передающие коды команд 5’b10110-5’b11111) будет передавать ОС с установленным в лог. «1» признаком «Абонент занят», за которым, соприкасаясь, будет следовать единственное слово данных.

e_bc_cwe (Expanded BC Control Word Enable), – разряд разрешение полного слова управления сообщением КШ:

Имеет смысл только для режима КШ. Если либо МСП в состоянии «нерасширенные возможности конфигурирования» (15-ый разряд (eme, 15-ый разряд регистра **cfg3_**, установлен в лог. «0»), либо разряд «разрешение полного слова управления сообщением КШ» запрограммирован в лог. «0», то слово управления сообщением КШ урезается. То есть, действующими разрядами являются лишь 7, 6, 5, 2, 1 и 0; в этой конфигурации функции других разрядов слова не могут быть задействованы. Если оба разряда, eme и **e_bc_cwe**, запрограммированы в лог. «1», то все 15 разрядов (с 14 по 0) слова управления сообщением КШ могут быть использованы.

bm_ex (Broadcast Mask ENA/XOR), – разряд разрешение/XOR* маскирования признака «Принята групповая команда»:

Этот разряд воздействует на работу 5-го разряда слова управления сообщением КШ - «маскирование признака «Принята групповая команда»». Если **bm_ex** в лог. «0», и бит «маскирование признака «Принята групповая команда»» слова управления сообщением КШ находится в состоянии лог. «1», то ожидаемой величиной признака «Принята групповая команда» в ОС становится лог. «1», а не лог. «0». Если **bm_ex** запрограммирован в лог. «1», бит

«маскирование признака «Принята групповая команда»» слова управления сообщением КШ используется для маскирования признака «Принята групповая команда» ОС, а не для выполнения логической операции XOR с последним. В этом случае событие ‘особое состояние ОУ’ возникает при получении ОС с установленным в лог. «1» признаком «Принята групповая команда» в нем, когда бит «маскирование признака «Принята групповая команда»» находится в лог. «0» (см. таблицы 6.10, 6.11).

ri_aame (Retry If 1553A And M.E.), – разряд ВПС, если 1553А и ситуация ‘ошибка в сообщении’:

Данный разряд не задействован в МСП. Эл. цепь, задающая правила выполнения ВПС, внутри МСП жёстко подсоединена не к выходу данного триггера, а к уровню лог. «0»; поэтому КШ МСП будет всегда выполнять ВПС по правилам, описанным ниже по тексту, для случая, когда разряд ri_aame был запрограммирован в лог. «0». При чтении разряд ri_aame будет возвращать записанное в него ранее, наиболее свежее значение.

ri_ss (Retry If Status Set), – разряд ВПС при событии ‘особое состояние ОУ’:

Этот разряд воздействует на выполнение ВПС в режиме КШ.

Если **ri_ss** запрограммирован в лог. «0», КШ не будет пытаться вновь выполнить сообщение, при отработке которого возникло событие ‘особое состояние ОУ’.

Если **ri_ss** запрограммирован в лог. «1», КШ будет пытаться вновь выполнить сообщение, при отработке которого возникло событие ‘особое состояние ОУ’.

Если «разрешение полного слова управления сообщением КШ» (**e_bc_cwe**, 12-ый разряд регистра **cfg4_**) установлен в лог. «0», то событие ‘особое состояние ОУ’ охватывает все 11 признаков ОС. Ожидаемыми значениями для 11-ти признаков являются лог. «0», за следующими исключениями:

Если **bm_ex** (11-й разряд регистра **cfg4_**) запрограммирован в лог. «0», «маскирование признака «Принята групповая команда»» слова управления сообщением КШ в лог. «1», то ожидаемые значением признака ОС «Принята групповая команда» становится лог. «1».

Если **bm_ex** (11-й разряд регистра **cfg4_**) запрограммирован в лог. «1», то «маскирование признака «Принята групповая команда»» слова управления сообщением КШ используется для маскирования признака «Принята групповая команда» ОС, а не для выполнения логической операции XOR с последним. В этом случае событие ‘особое состояние ОУ’ возникает при получении ОС с установленным в лог. «1» признаком «Принята групповая команда», когда бит «маскирование признака «Принята групповая команда»» находится в лог. «0».

Если «разрешение полного слова управления сообщением КШ» (**e_bc_cwe**, 12-ый разряд регистра **cfg4_**) установлен в лог. «1», событие ‘особое состояние ОУ’ может возникнуть только для тех признаков ОС, находящихся в лог. «1», для которых соответствующие разряды слова управления сообщением КШ были предварительно запрограммированы в лог. «0».

1r_asb (1st Retry Alt/Same Bus), – разряд выбор канала для 1-го ВПС; **2r_asb** (2nd Retry Alt/Same Bus), – разряд выбор канала для 2-го ВПС:

Эти два разряда влияют на выполнение ВПС в режиме КШ. **1r_asb** (8-ой разряд) задает канал шины, по которому будет выполняться первое «возобновление передачи сообщения» (ВПС) (после неудачи изначальной попытки переслать сообщение). Если **1r_asb** в лог. «0», ВПС будет производиться на той же самой шине, по которой передавалось оригинальное сообщение. Если **1r_asb** в лог. «1», первое ВПС будет выполняться по другому каналу шины по отношению к исходному каналу, употребленному для передачи подлинного сообщения во время первоначальной пересылки.

Аналогично, **2r_asb** (7-ой разряд) выбирает шину для второго ВПС. Второе ВПС будет производиться только, если разряд «Два/одно ВПС»(3-ий разряд регистра **cfg1_**) был своевременно запрограммирован в лог. «1». Второе ВПС имеет место только, если первое ВПС перед этим завершилась неудачей. Если **2r_asb** в лог. «0», второе ВПС делается на той же самой шине, где пересылалось подлинное сообщение. Если **2r_asb** в лог. «1», второе ВПС будет производиться по другому каналу по отношению к исходному, употребленному для передачи подлинного сообщения во время первоначальной пересылки.

v_me_nd (Valid M.E./No Data), – разряд разрешение ОУ не передавать СД в ситуации ‘ошибка в сообщении’:

Используется в режиме КШ.

Когда этот разряд запрограммирован в лог. «0», то если ОУ отвечает на сообщение передачи СД с установленным в лог. «1» признаком «Ошибка в сообщении» в ОС, ответ считается достоверным, когда после ОС следует затребованное количество слов данных. Кроме того, ответ только ОС, за которым отсутствуют информационные слова, расценивается как ошибка формата сообщения.

Если запрограммирован в лог. «1», ответ ОУ на сообщение передачи СД ОС с установившимся в нём признаком «Ошибка в сообщении», вслед за которым неразрывно последовало поступление затребованного количества слов данных, считается достоверным откликом. Вдобавок ответ ОС с установленным в нем признаком «Ошибка в сообщении» и отсутствием в дальнейшем информационных слов в этом режиме также считается достоверным откликом, а не ошибкой формата сообщения.

v_b_nd (Valid Busy/No Data), – разряд разрешение ОУ не передавать СД в ситуации ‘абонент занят’:

Используется в режиме КШ.

Когда данный разряд запрограммирован в лог. «0», и ОУ откликается на сообщение передачи СД с признаком «Абонент занят» в лог. «1» в своем ОС, то ответ ОУ считается достоверным только, если за ОС следует затребованное количество информационных слов. В этом сценарии отклик одним лишь ОС, без следования за ним СД, расценивается как ошибка формата сообщения.

Если разряд **v_b_nd** запрограммирован в лог. «1», то реакция ОУ на сообщение передачи СД, в которой ОС с установившимся в нем признаком «Абонент занят» предшествует затребованному количеству информационных слов, расценивается как достоверная. К тому же в этой конфигурации ответ ОС с установленным в нем признаком «Абонент занят», после которого отсутствуют СД, также считается достоверной реакцией, а не ошибкой формата сообщения.

mtgo (MT Tag Gap Option), – разряд модификации формирования ID - слова в режиме монитор слов:

Этот разряд влияет на способ формирования слова ID при работе МСП в режиме «монитор слов».

Данный разряд не воздействует на работу разряда **cd_g** и поля **gt**[7...0] ID слова в МСП. Разряд mtgo влияет на работу разряда mc ID слова «монитора слов».

Если **mtgo** запрограммирован в лог. «0», и слово было принято по другому каналу относительно предыдущей команды, то **cd_g** сбросится в лог. «0» (даже если время кадра слова текущего слова частично совпадает с кадром предыдущего слова), а поле «длительность межсловного промежутка» будет принимать значение на 20 мкс большее, чем действительный промежуток (если имеется) между концом предыдущего слова на другом канале и началом текущего слова на текущей шине. Если mtgo запрограммирован в лог. «0», тогда разряд mc слова ID (индикатор КУ) уровнем лог. «1» указывает, что слово не содержит КУ, в то время как лог. «0» указывает, что слово является КУ.

Если **mtgo** запрограммирован в лог. «1» и слово было принято по другому каналу относительно предыдущей команды внутри диапазона времени, частично совпадающего с кадром принятия предыдущего слова, то **cd_g** будет установлен в лог. «1» и полю «длительность межсловного промежутка» будет присвоено значение нуля. Если разрыв во времени между окончанием предыдущего слова по другому каналу и началом текущего слова на текущей шине существует, то разряду **cd_g** будет присваиваться значение лог. «0», и 8-ми разрядное поле «длительность межсловного промежутка» будет содержать правильное значение времени между предыдущим и текущим словами - значение без смещения в большую сторону на 20 мкс.

Если **mtgo** запрограммирован в лог. «1», то разряд mc наблюдаемого целевого слова расширит свое функциональное назначение и mc будет устанавливаться в лог. «1» не только при

приеме слова отличного от достоверной КУ, но также будет указывать на происшествие ошибки «квитирования» (затягивания процедуры получения МСП разрешения от УВ на обладание системной магистралью). Неправильное выполнение процедуры «квитирования» может случаться в «прозрачном» режиме сопряжения УВ с МСП. Когда происходит истечение максимально допустимого срока ожидания МСП доступа к разделяемому с УВ ОЗУ, наблюдаемое (полученное) слово вместе со своим ID теряются, и МСП установит разряд mc в ID для следующего принятого слова, с целью указать, что произошло истечение срока ожидания, и слово было потеряно.

l_rt_awcr5 (Latch RT Addr With Config. Reg. #5), – разряд защёлкивание адреса ОУ процедурой записи в регистр cfg5_:

Этот разряд не задействован в МСП. При чтении **l_rt_awcr5** будет возвращать записанное в него прежде, наиболее свежее значение.

tm[2..0] (Test Mode), – разряд задание проверочных режимов работы:

При использовании эти разряды должны устанавливаться в нулевые значения. Значения **tm**[2..0] зарезервированы для проверочных целей (см. таблицу 1.624):

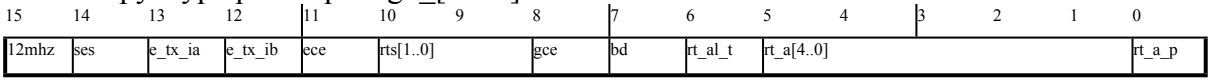
Таблица 1.624

tm[2]	tm[1]	tm[0]	Режим
0	0	0	Нормальные операции
0	0	1	Резерв
0	1	0	Тест кодера при производственном контроле
0	1	1	Резерв
1	0	0	Резерв
1	0	1	Резерв
1	1	0	Резерв
1	1	1	Резерв

1.3.12.2.2.4.1.11 Пятый конфигурационный регистр - cfg5_

Адрес регистра 01001; чт/зп.

Структура регистра cfg5_[15..0]:



Наименования разрядов регистра:

12mhz (12 MHz Clock Select), задание частоты входного тактового синхросигнала

ses (Single-Ended Select), выбор однопроводного сопряжения с приёмопередатчиками;

e_tx_ia (External TX Inhibit A), указатель состояния внешнего блокирования канала А;

e_tx_ib (External TX Inhibit B), указатель состояния внешнего блокирования канала В;

ece (Expanded Crossing Enabled), разрешение удвоения частоты выборки декодером входного сигнала;

rts[1..0] (Response Timeout Select), выбор интервала ожидания ОС;

gce (Gap Check Enable), разрешение проверки минимальной паузы перед выдачей ОС;

bd (Broadcast Disable), запрет групповых сообщений;

rt_al_t (RT Addr Latch/transparent), формирования адреса ОУ – защелкивание / сквозная пересылка;

rt_a[4..0] (RT Address), адрес ОУ;

rt_a_p (RT Address Parity), разряд контроля по чётности адреса ОУ.

Примечание - МСП должен быть установленным в состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_ должен быть установлен в лог. «1») пред попытками активизировать какую-либо функцию, разрешаемую регистром cfg5_. Разряды rt_a[4..0] и rt_a_p всегда доступны по чтению и не зависят от разряда eme регистра cfg3_.

Функциональное описание разрядов регистра:

12mhz (12 MHz Clock Select), – разряд задание частоты входного тактового синхросигнала:

По умолчанию разряд 12mhz устанавливается в лог. «0», разрешая работу от 16 МГц. Если 12mhz находится в лог. «1», частота для входного тактового сигнала должна равняться 12 МГц.

ses (Single-Ended Select), – разряд выбор однопроводного сопряжения с приёмопередатчиками:

Этот разряд не задействован в МСП. Внутренняя цепь «выбора однопроводного сопряжения с приёмопередатчиками» жёстко, аппаратно, подсоединена к лог. «0» внутри устройства, что делает однопроводное сопряжение с приёмниками (оптоволоконных каналов) невозможным, зато разрешает внутреннему декодеру МСП принимать сигнал по двум электрическим цепям от приемника, соответствующего Г О С Т Р 52070-2003 (манчестерское бифазовое манипулирование). При чтении ses будет возвращать записанное в него прежде, наиболее свежее значение.

e_tx_ia (External TX Inhibit A), – разряд указатель состояния внешнего блокирования канала A:

Этот разряд не задействован в МСП. В МСП отсутствует входной сигнал для подачи извне сигнала блокирования канала A; при чтении e_tx_ia будет возвращать записанное в него прежде, наиболее свежее значение.

e_tx_ib (External TX Inhibit B), – разряд указатель состояния внешнего блокирования канала B:

Этот разряд не задействован в МСП. В МСП отсутствует входной сигнал для подачи извне сигнала блокирования канала B; при чтении e_tx_ib будет возвращать записанное в него прежде, наиболее свежее значение.

ece (Expanded Crossing Enabled), – разряд разрешение удвоения частоты выборки декодером входного сигнала:

Этот разряд не задействован в МСП.. При чтении ece будет возвращать записанное в него прежде, наиболее свежее значение.

rts[1..0] (Response Timeout Select), – разряд выбор интервала ожидания ОС:

Эти два разряда могут быть задействованы для выбора величины временного промежутка, разрешающего ожидание ОС для МСП. Этот параметр (переменная) используется в режиме КШ, в режиме ОУ (в сообщениях типа «ОУ-ОУ», в которых МСП является принимающим ОУ) и в режиме «монитор сообщений». Существующие четыре возможных задаваемых программно варианта выбора (величина срока ожидания) представлены в таблице 1.625.

Примечание - по умолчанию для состояния «нерасширенные возможности конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «0») МСП выбирается значение 18,5 мкс.

Таблица 1.625 – Выбор времени ожидания ОС

cfg5_[10] (rts[1])	cfg5_[9] (rts[0])	Величина времени ожидания ОС (мкс)
0	0	18,5*
0	1	22,5
1	0	50,5
1	1	130

gce (Gap Check Enable), – разряд разрешение проверки минимальной паузы перед выдачей ОС:

Если запрограммирован в лог. «0», МСП не делает проверку выдержки ‘мертвого времени’ ЛПИ перед началом передачи ОС.

Если запрограммирован в лог. «1», МСП делает проверку обязательной минимальной выдержки в 2 мкс (‘мертвого времени’) перед стартом передачи ОС по ЛПИ. Если минимальный промежуток времени не выдержан, МСП считает, что командное слово КШ или ОС являются недостоверными, то есть выполненными с ошибкой формата сообщения.

Если запрограммирован в лог. «1», то проверяется, что ОУ не отвечает ОС через время меньшее, чем 4 мкс (согласно ГОСТ Р 52070-2003 - это промежуток между серединами зон разряда контроля по четности последнего КС/СД и синхроимпульса ОС), т.е. реально проверяется, что период отсутствия декодируемых сигналов в ЛПИ (‘мёртвое время’) меньше 2 мкс

bd (Broadcast Disable), – разряд запрет групповых сообщений:

Применяется только в режимах ОУ и МТ. Если bd запрограммирован в лог. «0», МСП будет распознавать адрес 31 как групповой адрес. В этой конфигурации «адрес ОУ» 31 не может быть использован как адрес отдельного оконечного устройства. Если bd запрограммирован в лог. «1», МСП не будет распознавать адрес ОУ 31 как групповой адрес. В этом случае, «адрес ОУ» 31 может быть использован как адрес отдельного ОУ.

Если bd был запрограммирован в лог. «1» в режиме монитор слов, то 5-ый разряд ID (brcst) будет всегда возвращать лог. «1» (даже для командного слова, направленного по адресу ОУ 31).

rt_al_t (RT Addr Latch/transparent), – разряд формирования адреса ОУ – защелкивание / сквозная пересылка:

Этот разряд не задействован в МСП. Эл. цепь, определяющая способ формирования адреса ОУ – защелкивание / сквозная пересылка жестко (аппаратно) подключена к лог. «0» внутри МСП, задавая тем самым режим постоянного отслеживания разрядами rt_a[4..0] и rt_a_p состояния на входных сигналах msp_rtad[4:0], msp_rtadp. При чтении rt_al_t будет возвращать записанное в него прежде, наиболее свежее значение.

rt_a[4..0] (RT Address), – разряд адрес ОУ;

rt_a_p (RT Address Parity), – разряд контроля по чётности адреса ОУ:

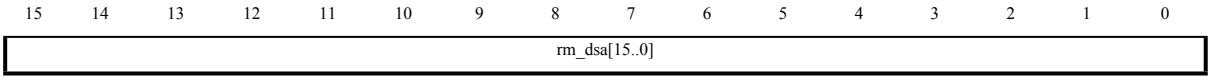
Эти шесть разрядов (rt_a[4..0] и rt_a_p) возвращают при чтении значения, установленные на входных сигналах «адрес ОУ» и «контроля по четности адреса ОУ», которые защелкивается внутри МСП. Если сумма значений разрядов «адрес ОУ» и разряда контроля по чётности кратна двум (чётна), то МСП не будет распознавать и отвечать на командные слова, направляемые по собственному адресу ОУ МСП. Он будет, однако, принимать сообщения, направляемые по групповому адресу 31, но только если групповые сообщения не были запрещены.

Примечание. МСП работает в режиме постоянного отслеживания разрядами rt_a[4..0] и rt_a_p состояния на входных сигналах msp_rtad[4:0], msp_rtadp.

1.3.12.2.2.4.1.12 Регистр указателя буфера данных ОУ/МТ - rm_dsa

Адрес регистра 01010; чт/зп.

Структура регистра rm_dsa[15..0]:



rm_dsa[15..0] (RT/Monitor Data Stack Address), регистр указателя буфера данных ОУ/МТ:

В режиме ОУ слово (указатель), прочитанное из таблицы указателей для подадресов во время стартовой последовательности действий (SOM), вначале загружается в регистр указателя буфера данных ОУ. Величина регистра затем увеличивается на единицу (по модулю размера буфера данных) после каждой успешной обработки (чтение/запись) СД сообщения. Исключения: Если КС относится к КУ «передачи» и передаваемое слово данных берется из внутреннего регистра МСП (напр. ‘Передать слово ВСК ОУ’, ‘Передать векторное слово’, ‘Передать последнюю команду’), то регистр rm_dsa[15:0] и указатель на блок данных из состава описателя блока сообщения ОУ содержат передаваемое слово данных.

Регистр rm_dsa[15:0] и указатель на блок данных из состава описателя блока сообщения ОУ содержат слово ВСК, если КС относится к неопределенной КУ (разряд «приём/передача» установлен в лог. «0» и поле «код команды» имеет значение 5’b10011) и 15 разряд ebwe регистра cfg4_ «разрешение внешнего слова ВСК» равен лог. «0».

Если 0-й разряд emch регистра cfg3_ «расширенная обработка КУ» равен лог. «1», то после отработки сообщения, где КС относится к неопределенной КУ (разряд «приём/передача» установлен в лог. «0» и поле «код команды» имеет значение 5’b10010), то регистр rm_dsa[15:0] и указатель на блок данных содержат предыдущее достоверное командное слово.

В режиме монитор слов rm_dsa[15:0] содержит текущую величину указателя на буфер данных.

В режиме монитора сообщений или в комбинированном режиме «ОУ/монитор сообщений», этот регистр содержит текущее значение указателя на буфер данных монитора сообщений.

1.3.12.2.2.4.1.13 Регистр остатка времени кадра КШ - bc_ftr

Адрес регистра 01011; чт.

Структура регистра bc_ftr[15..0]:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bc_ftr[15..0]															

bc_ftr[15..0] (BC Frame Time Remaining): Счетчик, который загружается во время старта пересылки кадра сообщений, значением регистра длительности кадра КШ bcft_rtlc_mttw[15..0] и затем декрементируется до нулевого значения; по достижению счетчиком нулевого значения и завершении кадра (в режиме автоповтора) МСП повторяет старт кадра; регистр остатка времени кадра КШ доступен УВ по чтению, показывает время, оставшееся до завершения временного интервала, задаваемого регистром bcft_rtlc_mttw[15..0] (BC Frame Time). Время кадра КШ программируется с разрешением в 100 мкс/МЗР, до 6,55 мс.

1.3.12.2.2.4.1.14 Регистр остатка времени текущего сообщения КШ - bc_trnm

Адрес регистра 01100; чт.

Структура регистра bc_trnm[15..0]:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bc_trnm[15..0]															

bc_trnm[15..0] (BC Message Time Remaining), регистр остатка времени текущего сообщения КШ; имеет разрешение 1 мкс на младший разряд (МЗР). Максимальный временной срок работы (заполнения) этого счетчика равен 65535 мкс.

1.3.12.2.2.4.1.15 Регистр длительности кадра КШ/ Регистр командного слова сообщения ОУ/ Регистр слова срабатывания («монитор слов») - bcft_rtlc_mttw

Адрес регистра 01101; чт/зп.

Структура регистра bcft_rtlc_mttw[15..0]:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bcft_rtlc_mttw[15..0]															

В режиме КШ время кадра КШ, используемое в режиме автоматического повторения кадров, задается посредством этого регистра. Длительности кадра сообщений программируется с возможным наименьшим шагом в 100 мкс/МЗР до максимального значения 6,55с.

В режиме ОУ этот регистр сохраняет командное слово текущего или последнего в своей обработке сообщения ОУ МСП.

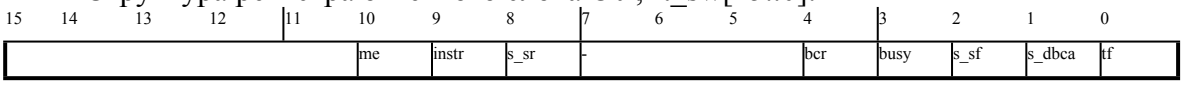
В режиме «монитор слов» этот регистр сохраняет содержимое слова срабатывания. Это слово должно поставляться процессором УВ. Когда «монитор слов» МСП находится в состоянии работы, он сравнивает содержимое этого регистра со всеми поступающими из ЛПИ достоверными словами. «Запускающий механизм» монитора может быть использован для того, чтобы стартовать или, наоборот, остановить сохранение, запись наблюдаемых слов в буфер данных монитора или породить запрос прерывания УВ. «Запускающие» свойства монитора не могут быть задействованы в режимах «монитор сообщений» и «ОУ/монитор сообщений».

1.3.12.2.2.4.1.16 Регистр ответного слова ОУ - rt_sw

Адрес регистра 01110; чт.

Регистр ответного слова ОУ rt_sw[15..0] обеспечивает доступ в режиме чтения к внутреннему для ОУ значению ОС МСП.

Структура регистра ответного слова ОУ, rt_sw[15..0]:



Наименования разрядов регистра (признаков ОС):

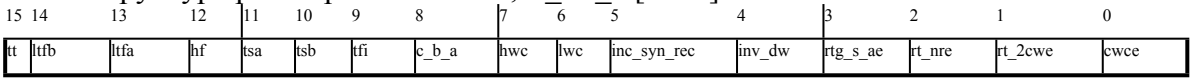
me (Message Error): «Ошибка в сообщении»;
instr (Instrumentation): «Передача ОС»;
s_sr (Service Request): «Запрос на обслуживание»;
bcr (Broadcast Command Received): «Принята групповая команда»;
busy (Busy): «Абонент занят»;
s_sf (Subsystem Flag): «Неисправность абонента»;
s_dbca (Dynamic Bus Control Accept): «Принято управление интерфейсом»;
tf (Terminal Flag): «Неисправность ОУ».

1.3.12.2.2.4.1.17 Регистр слова ВСК - rt_bit_w

Адрес регистра 01111; чт.

Слово Встроенного Теста ОУ rt_bit_w[15..0] обеспечивает доступ в режиме чтения к внутреннему регистру слова ВСК ОУ.

Структура регистра слова ВСК, rt_bit_w[15..0]:



Наименования разрядов регистра:

tt (Transmitter Timeout)»; Указатель затягивания передачи на время, большее, чем 668 мкс;

ltfb, ltfa (Loop Test failure B, Loop Test failure A): ошибка проверочного возврата;

hf (Handshake Failure): истечение времени ожидания подтверждения (квитирования);

tsa, tsb(Transmitter Shutdown A, Transmitter Shutdown B): блокировка передатчика канала А/В КУ ‘Блокировать передатчик’;

tfi (Terminal Flag Inhibited): индикатор КУ ‘Блокировать признак неисправность ОУ’;

c_b_a (Channel B/A): индикатор канала сообщения А/В;

hwc (Hight Word Count): превышение количества переданных или принятых СД;

lwc (Low Word Count): заниженное количество переданных или принятых СД;

inc_syn_rec (Incorrect Sync Type Received): синхроимпульс КС в СД;

inv_dw (invalid data word): обнаружено недостоверное слово в сообщении;

rtg_s_ae (RT-RT Gap/Sync/Address Error): ошибка в передаче ОУ-ОУ: несоблюдение минимальной паузы перед выдачей ОС/ синхросигнала или формата / адрес ОУ;

rt_nre (RT-RT No Response Error): ошибка в передаче ОУ-ОУ»: интервала ожидания ОС;

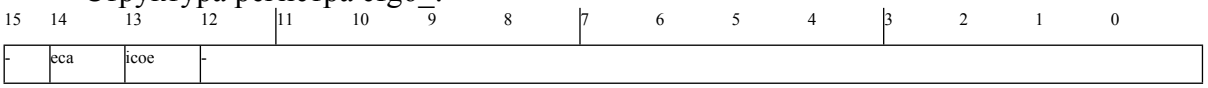
rt_2cwe (RT-RT 2nd Command Word Error): индикатор ошибки 2-ого командного слова;

cwce (Command Word Contents Error): принятое КС не согласуется с требованиями ГОСТ Р 52070-2003.

1.3.12.2.2.4.1.18 Шестой конфигурационный регистр - cfg6_

Адрес регистра 11000; чт/зп.

Структура регистра cfg6_:



Наименования разрядов регистра:

eca (Enhanced Cpu Access): включение режима ускоренного доступа УВ;

icoe (Increment csp On Eom): увеличение указателя на стек команд при EOM последовательности.

Примечание - МСП должен быть установленным в состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_ должен быть установлен в лог. «1») пред попытками активизировать какую-либо функцию, разрешаемую регистром cfg6_.

Функциональное описание разрядов регистра:

eca (Enhanced Cpu Access), – разряд включение режима ускоренного доступа УВ:

Если данный разряд установлен в лог. «0» (по умолчанию), то циклы обращения УВ к регистрам или памяти МСП не могут завершиться во время SOM/EOM последовательности, т.е. УВ ждет окончания SOM или EOM последовательности, прежде чем завершить свой цикл доступа к регистрам/памяти МСП. Если разряд eca установлен в лог. «1», то МСП разрешает УВ доступ к регистрам или памяти во время SOM/EOM последовательности.

В таблице 1.626 приведены наихудшие времена задержек циклов доступа УВ к МСП при различных значениях разряда eca.

Таблица 1.626

Частота	eca[14] = 1’ b0	eca[14] = 1’b1 Цикл чтения	eca[14] = 1’b1 Цикл записи
16 МГц	3880 мкс	1880 нс	1880 нс

icoe (Increment csp On Eom), – разряд увеличение указателя на стек команд при EOM последовательности.

Применяется в режимах ОУ, «монитор сообщений» и комбинированный режим «ОУ/монитор сообщений». В режимах КШ и «монитор слов» разряд icoe не используется.

Если данный разряд установлен в лог. «0» (по умолчанию), то значение указателя на стек команд области А/Б, хранящееся в ОЗУ, будет увеличиваться на четыре во время SOM-последовательности.

Если разряд icoe установлен в лог. «1», то значение указателя на стек команд области А/Б, хранящееся в ОЗУ, будет увеличиваться на четыре во время EOM-последовательности.

Установка разряда icoe в лог. «1», помогает УВ определить момент завершения отработки текущего сообщения по опросу указателя на стек команд области А/Б. Если же разряд icoe установлен в лог. «0», то УВ необходимо ждать завершения SOM-последовательности второго (следующего) сообщения, для определения момента окончания отработки первого сообщения.

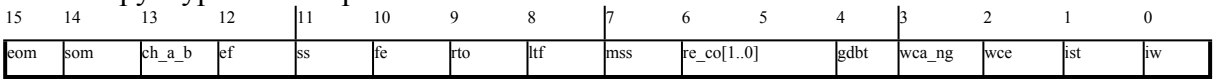
Эта возможность полезна при работе УВ с МСП без обработки прерываний, - на основе опроса значения указателя на стек команд области А/Б.

1.3.12.2.2.4.1.19 Слово состояния блока сообщения

Слово состояния блока сообщения сохраняется в первом местоположении описателя блока сообщения в командном стеке для режимов КШ, ОУ, «МТ сообщений», и «ОУ/МТ сообщений». Для режима «ОУ/МТ сообщений» существует два командных стека, поддерживаемых: один для ОУ, другой для МТ. В режиме «монитора слов» командного стека не существует. Для всех режимов, исключая «монитор слов», слово состояния блока сообщения обновляется логикой управления памяти МСП как в начале, так и при завершении соответствующего сообщения. Оно содержит информацию касательно завершенности обработки (передачи/получения) сообщения, используемого для пересылки канала, наличие каких-либо ошибок сообщения.

Примечание - если в режиме КШ производится ВПС, разряды слова состояния блока сообщения отражают результаты последней попытки ВПС.

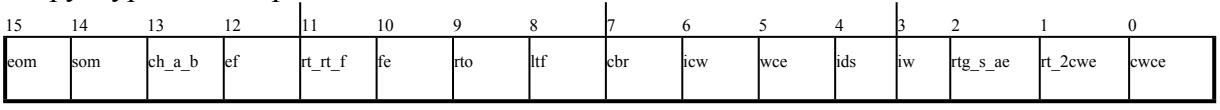
Структура слова в режиме КШ:



Наименования разрядов регистра:

eom (END-OF-MESSAGE), завершение отработки сообщения;
som (START-OF-MESSAGE), старт отработки сообщения;
channel_b_a (channel b/a), канал обмена B/A;
ef (Error flag), признак ошибки;
ss (Status Set), особое состояние ОУ;
fe (Format Error), ошибочный формат сообщения;
rto (Response Time out), истечение времени ожидания ответного слова ОУ;
ltf (Loop Test Fail), ошибка проверочного возврата;
mss (Masked Status Set), маскируемый признак ‘особое состояние ОУ’;
re_co[1..0] (RETRY COUNT [1..0]), количество ВПС;
gdbt (Good Data Block Transfer), успешная пересылка блока данных;
wsa_ng (Wrong Status Address/No Gap), неверный адрес в ответном слове ОУ или нарушение минимальной паузы перед выдачей ОС;
wce (Word Count Error), ошибка количества слов в сообщении;
ist (Incorrect Sync Type), некорректный тип синхросигнала;
iw (Invalid Word), признак недостоверности слова;

Структура слова в режиме ОУ:

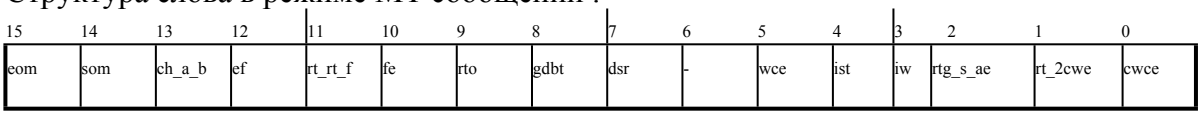


Наименования разрядов регистра:

eom (END-OF-MESSAGE), завершение отработки сообщения;
som (START-OF-MESSAGE), старт отработки сообщения;
channel_b_a (channel b/a), канал обмена B/A;
ef (Error flag), признак ошибки;
rt_rt_f (RT-to-RT FORMAT), принимающий ОУ в пересылке типа ОУ-ОУ;
fe (Format Error), ошибочный формат сообщения;
rto (Response Time out), истечение времени ожидания ответного слова ОУ;
ltf (Loop Test Fail), ошибка проверочного возврата;
cbr (Circular Buffer Rollover), ‘прокручивание’ кольцевого буфера;
icw (Illegal Command Word), признак недопустимости командного слова;

wce (Word Count Error), ошибка количества слов в сообщении;
ids (Incorrect Data Sync), некорректный тип синхросигнала в слове данных;
iw (Invalid Word), признак недостоверности слова;
rtg_s_ae (RT-RT Gap/Sync/Address Error), ошибка в передаче ОУ-ОУ: интервала ожидания ОС / синхросигнала или формата / адрес ОУ;
rt_2cwe (RT-RT 2nd Command Word Error), индикатор ошибки 2-ого командного слова;
cwce (Command Word Contents Error), принятое КС не согласуется с требованиями ГОСТ Р 52070-2003.

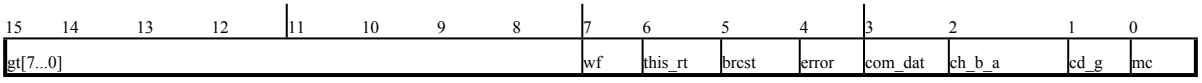
Структура слова в режиме МТ сообщений :



Наименования разрядов регистра:
eom (END-OF-MESSAGE), завершение отработки сообщения;
som (START-OF-MESSAGE), старт отработки сообщения;
channel_b_a (channel b/a), канал обмена B/A;
ef (Error flag), признак ошибки;
rt_rt_f (RT-to-RT FORMAT): признак пересылки типа ‘ОУ-ОУ’;
fe (Format Error): ошибка формата сообщения;
rto (Response Time out): истечение времени ожидания ответного слова ОУ;
gdbt (Good Data Block Transfer): успешная пересылка блока данных;
dsr (data stack Rollover): ’прокручивание’ буфера данных;
wce (Word Count Error): ошибка количества слов в сообщении;
ist (Incorrect Sync Type): некорректный тип синхросигнала;
iw (Invalid Word): недостоверное слово;
rtg_s_ae (RT-RT Gap/Sync/Address Error): ошибка в передаче ОУ-ОУ: интервала ожидания ОС / синхросигнала или формата / адрес ОУ;
rt_2cwe (RT-RT 2nd Command Word Error): индикатор ошибки 2-ого командного слова;
cwce (Command Word Contents Error): принятое КС не согласуется с требованиями ГОСТ Р 52070-2003;

1.3.12.2.2.4.1.20 Мониторное слово распознавание (ID – слово)

Структура слова:



Наименования разрядов:
gt[7...0] (gap time): длительность межсловного промежутка;
wf (word flag): признак слова распознавания;
this_rt:признак равенства значения в поле КС ‘адрес ОУ’ собственному адресу ОУ;
brcst (BROADCAST): признак группового сообщения;
error: признак ошибки;
com_dat (COMMAND/DATA* SYNC): тип синхросигнала слова - КС/ОС либо СД;
ch_b_a (CHANNEL B/A*): использовавшийся для передачи слова канал обмена A/B;

cd_g (CONTIGUOUS DATA/GAP): признак соприкосновения слов;
mc (MODE CODE*): признак КУ/ошибка подтверждения (квитирования).

1.3.12.2.2.5 Работа в режиме контроллера шины (КШ)

Протокольная логика МСП в режиме КШ позволяет выполнять обмены сообщениями во всех форматах, предусмотренных ГОСТ Р 52070-2003. Форматы сообщений программируются (выбираются) на основе индивидуального сообщения посредством разрядов слова управления сообщением КШ и разряда «Приём-передача» командного слова соответствующего сообщения.

Дополнительно к заданию формата сообщения слово управления сообщением КШ позволяет задать для конкретного сообщения канал обмена (A/B), режим самотестирования, возобновление передачи сообщения (ВПС) в случае ошибок при предыдущих попытках, разрешение прерывания после завершения работы с данным (избранным) сообщением и маскирование признаков ответного слова ОУ.

КШ производит все, требуемые ГОСТ Р 52070-2003, проверки ошибок, включая оценку максимального времени ожидания ОС, контроль типа синхросигнала, кодирования синхросигнала, манчестерского кодирования (биполярный фазоманипулированный код), контроля по чётности, количества разрядов в слове, количества слов в сообщении, поля «Адрес ОУ» ответного слова ОУ. Длительность максимального разрешенного времени ожидания ОС может выбираться из набора величин: 18, 22, 50 и 130 мкс. Наиболее долгое время позволяет работать на длинных магистралях и/или использовать ретрансляторы.

КШ может быть запрограммирован на обработку кадра сообщений (набора последовательно пересылаемых сообщений), содержащего до 512 сообщений без вмешательства УВ. Возможно задание режима «одиночного кадра» или режима «автоповтор отработки кадра». В режиме автоматического повторения кадра темп повторения может программироваться как от внутреннего счетчика кадра, так и от внешнего сигнала. Внутреннее время повторения кадра может быть выбрано программно из диапазона в 6,55 сек, с шагом 100 мкс. Дополнительно, есть возможность задания паузы между сообщениями, которая определяется как время от старта текущего сообщения до старта последующего и программируется (задается) на основе индивидуального сообщения. Время между двумя последовательными сообщениями выбирается программно из диапазона в 65,5 мс с шагом 1 мкс.

1.3.12.2.2.5.1 Организация памяти КШ

Таблица 1. показывает типичное распределение (карту) памяти для режима КШ. Необходимо отметить, что для двух областей ОЗУ (А и Б) существуют неперемещаемые местоположения для двух указателей на стека команд (адреса расположения 0100 и 0104) и для двух счетчиков сообщений (0101 и 0105). Активная область ОЗУ (А или Б) задается в регистре cfg1_. До старта в режиме «обработка одного кадра» УВ должен инициализировать значения указателя на стек команд и счетчика сообщений для активной глобальной области. Разрешение режима «автоповтор отработки кадра» влечет за собой резервирование еще четырех мест в пространстве памяти; это места для начальных значений указателей на стек команд (адреса расположения 102 и 106) и для начальных значений счетчиков сообщений (103 и 107).

Для размещения областей стека команд и блоков сообщений разрешается использование других зон, отличных от указанных в таблице 1.627.

Таблица 1.627 - Типичное распределение (карта) памяти для режима КШ (для размера ОЗУ 4К слов)

Адрес (шестн.)	Описание
0000-00FF	Стек команд области А
0100	Указатель на стек команд области А (неперемещаемое местоположение)
0101	Счетчик сообщений области А (неперемещаемое местоположение)
0102	Начальное значение указателя на стек команд области А (см. Примечание)
0103	Начальное значение счётчика сообщений области А (см. Примечание)
0104	Указатель на стек команд области Б (неперемещаемое местоположение)
0105	Счетчик сообщений области Б (неперемещаемое местоположение)
0106	Начальное значение указателя на стек команд области Б (см. Примечание)
0107	Начальное значение счётчика сообщений области Б (см. Примечание)
0108-012D	Блок сообщения 0
012E-0153	Блок сообщения 1
0154-0179	Блок сообщения 2
.	.
.	.
.	.
0ED6-0EFB	Блок сообщения 93
0EFC-0EFF	Не используется
0F00-0FFF	Стек команд области Б

Примечание - Начальные значения указателей на стек команд и начальные значения счётчиков сообщений используются КШ только в состоянии «расширенных возможностей конфигурирования» при разрешении режима «автоповтор отработки кадра» (режим автоповтор отработки кадра требует задания ненулевого значения регистра длительности кадра КШ).

1.3.12.2.2.5.1.1 256-словные границы

Следует отметить, что для КШ (также как и для ОУ) указатель на стек команд будет ‘циклически переполняться’ (‘прокручиваться’) в границах, задаваемых 14-ым и 13-ым разрядами регистра cfg3_ (по умолчанию, в границах 256-словной области). Размеры стека команд программно выбираемы из ряда следующих величин: 256 слов (64 блоков описателей сообщений), 512, 1024 или 2048 слов (512 блоков описателей сообщений). По умолчанию установленным оказывается 256-словное ограничение области стека команд. Это означает, что указатель на стек команд будет переходить из XXFF в XX00, а не в (XX + 1)00.

Если 10-ый разряд регистра cfg2_ «запрещение 256-словных ограничений» установлен в лог. «0», то 256-словные ограничения также оказываются действующими для блоков сообщений КШ. Однако, если разряд «запрещение 256-словных ограничений» установлен в лог. «1», то указатель на блок сообщения КШ не будет придерживаться 256-словных ограничений. Это позволяет осуществлять более эффективное распределение адресного пространства разделяемого ОЗУ для хранения блоков сообщений КШ.

Настоятельно рекомендуется при работе МСП в режиме КШ разряд «запрещение 256-словных ограничений» устанавливать в лог. «1».

Для упрощения в таблице 6.1 предполагается, что сообщения имеют максимальную длину для каждого блока сообщения. Максимальный размер блока сообщения имеет 38 слов для «передача» ОУ-ОУ из 32 слов (слово управления сообщением КШ + 2 КС + слово проверочного возврата + 2 ответных слова + 32 слова данных). Заметим, что этот пример предполагает, что разряд «запрещение 256-словных ограничений» установлен в лог. «1».

1.3.12.2.2.5.1.2 Механизм глобальной двойной буферизации

Одним из свойств управления памятью является механизм глобальной двойной буферизации с помощью разбиения ОЗУ КШ на «активную» и «неактивную» области (А и Б). Это обеспечено наличием двух структур данных МСП в режиме КШ, в каждую из которых входят слова - указатель на стек команд, счетчик сообщений, начальное значение указателя на стек команд, начальное значение счетчика сообщений. В любой момент времени внутренней логике управления памятью МСП доступны лишь те структуры, которые принадлежат активному подмножеству. В любой момент времени для УВ доступна как активная область, так и неактивная область ОЗУ. В большинстве применений УВ производит операции чтения/записи с неактивной областью в то время, когда МСП обрабатывает (принимает, передает) сообщения в активной области ОЗУ.

Разряд 13 регистра cfg1_ задает активную область ОЗУ (А или Б). Важно отметить, что УВ может изменить назначение активной области и в тот момент, когда МСП проводит работу с кадром сообщений (изменяя значение 13 разряда конфигурационного регистра cfg1_), однако реально активная область не будет переключена до завершения обработки текущего кадра.

Использование глобальной двойной буферизации исключает возможность некорректности данных, например, когда МСП передает или УВ читает смесь из старых и новых данных.

Другой подход используется в механизме «множественной буферизации». В этом случае программируется сценарий для работы с несколькими кадрами сообщений. Это достигается размещением нескольких областей стека команд в адресном пространстве ОЗУ. Когда обработка очередного кадра закончена, УВ просто записывает новые значения указателя на стек команд и счетчика сообщений перед запуском работы со следующим кадром.

При работе в режиме «обработка одного кадра» УВ до запуска обработки каждого кадра должен записать значения счетчика сообщений и указателя на стек команд для активной области.

1.3.12.2.2.5.2 Программирование кадров сообщений

1.3.12.2.2.5.2.1 Управление памятью в режиме КШ

Схема управления памятью в режиме КШ приведена на рисунке 1.205. КШ может быть запрограммирован (настроен) на передачу кадра, состоящего из множества сообщений. Количество сообщений, подлежащее обработке, задается в слове счетчик сообщений для активной области ОЗУ. К тому же, УВ должен заполнить информацией перед запуском МСП в самостоятельную работу указатель на стек команд для активной области ОЗУ.

Указатель на стек команд ссылается на (адресует) описатель блока сообщения, состоящий из четырех слов и находящийся в области стека команд ОЗУ.

Описатель блока сообщения формируется для каждого сообщения, которое должно быть обработано. Последовательность обработки сообщений соответствует последовательности размещения описателей блоков сообщений в стеке команд (от младшего адреса к старшему). Каждое обрабатываемое сообщение должно быть предварительно записано УВ в блок сообщения (в ОЗУ). В четвертое слово соответствующего описателя блока сообщения УВ должно быть записано значение адреса (указатель) блока сообщения (до начала обработки сообщения).

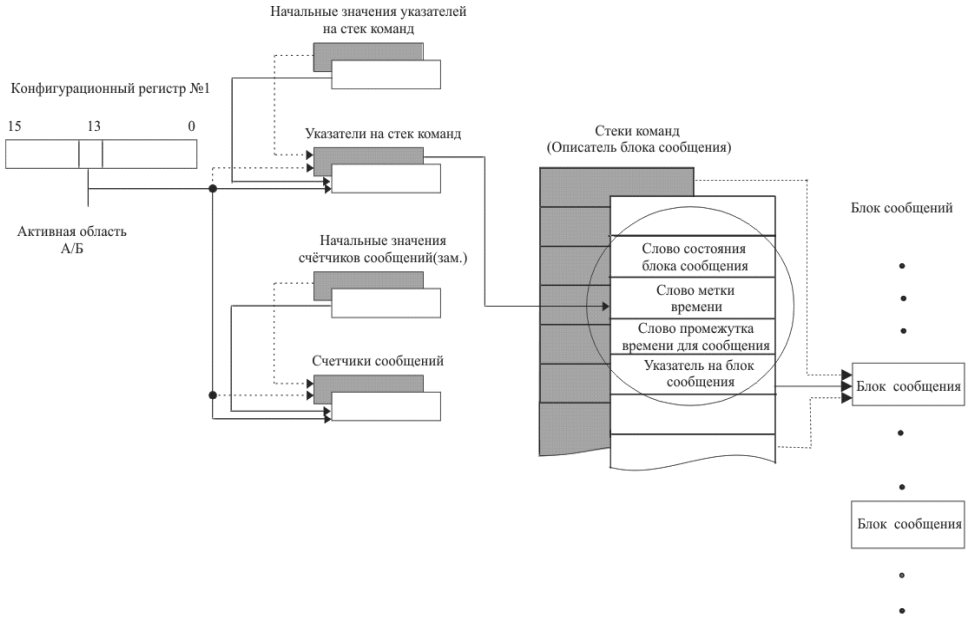


Рисунок 1.205 - Управление памятью в режиме КШ

Примечание. Начальное значения указателей на стек команд и Начальное значения счетчиков сообщений используются только в режиме «автоповтор отработки кадра».

Как показано на рисунке 1.205, УВ может делать выбор между глобальной областью ОЗУ А и глобальной областью ОЗУ Б, задавая значение 13 разряда регистра cfg1_. На рисунке 1.205 предполагается, что область А (незатененная) – активна, а область Б (затененная) – неактивна.

В любой момент времени указатель на стек команд области А/Б адресует первое слово описателя блока сообщения для обрабатываемого сообщения. Как показано на рисунке 1.205, каждый описатель блока сообщения состоит из четырех слов. Первые два слова резервируются под слово состояния блока сообщения и слово метки времени. Данные в эти слова записываются протокольной логикой КШ в начале и в конце обработки каждого сообщения.

Слово состояния блока сообщения КШ содержит разряды, относящиеся к текущему состоянию обработки сообщения, завершению обработки сообщения, контролю корректности обработки сообщения и каналу обмена (А/В), по которому ведется пересылка сообщения.

Слово метки времени фиксирует значение регистра счетчика метки времени (tt_reg) МСП в начале и в конце обработки сообщения. Регистр счетчика метки времени имеет программируемую точность счета в диапазоне от 2 до 64 мкс на младший разряд, с дискретом степеней числа два (2, 4, 8,…,64). Регистр счетчика метки времени может тактироваться от внешнего источника.

Следующие два слова описателя блока сообщения должны записываться УВ.

Слово промежутка времени для сообщения используется в состоянии «расширенных возможностей конфигурирования» КШ (разряд 15 регистра cfg3_ запрограммирован в состояние логической «1») и к тому же задано «разрешение задания промежутка времени для сообщения» (imgte, разряд 05 регистра cfg1_ запрограммирован в состояние лог. «1»). Слово промежутка времени для сообщения задает интервал времени от старта текущего сообщения до старта следующего, другими словами – задаёт промежуток времени на отработку данного сообщения (включая возможные ВПС). Дискрет значений слова промежутка времени для сообщения составляет 1 мкс на младший разряд.

Четвертое слово – указатель на блок сообщения – должно содержать адрес первого слова соответствующего блока сообщения. Это значение должно быть записано УВ до начала обработки данного сообщения. Используется логикой управления памятью МСП для доступа в начало соответствующего блока сообщения.

Значение указателя на стек команд для активной области А/Б увеличивается на четыре после завершения обработки каждого сообщения КШ.

Два других объекта, размещенных в фиксированных позициях адресного пространства разделяемого ОЗУ, которые должны быть проинициализированы УВ, если задан режим КШ, состояние «нерасширенные возможности конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «0»), это слова указатель на стек команд и счетчик сообщений. Указатели на стек команд размещаются в местоположениях с адресами 0100 (для области А) и 0104 (для области Б). Указатель на стек команд должен быть проинициализирован значением адреса первого слова описателя блока сообщения для того сообщения, которое будет обрабатываться первым. Счётчики сообщений расположены по адресам 0101 (для области А) и 0105 (для области Б). Счётчик сообщений активной области А/Б должен быть своевременно загружен процессором УВ числом сообщений, которое необходимо обработать, в дополнительном коде (например, число fffe, будет представлять единичное количество сообщений). Значение счётчика сообщений увеличивается на единицу после обработки каждого сообщения.

Заметим, что если КШ запрограммирован в режим «автоповтора кадров», то УВ необходимо инициализировать местоположения начального значения указателя на стек команд и начального значения счетчика сообщений, а не указатель на стек команд» и счетчик сообщений.

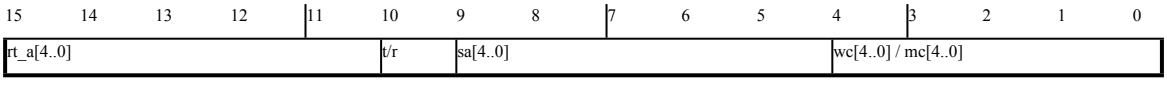
Счетчик сообщений (начальное значение счетчика сообщений) должен быть проинициализирован УВ значением, равным дополнительному коду числа сообщений в данном кадре. Например, если кадр содержит одно сообщение, то значение счетчика сообщений должно быть FFFE. Величина счетчика сообщений увеличивается на 1 в конце обработки каждого сообщения.

1.3.12.2.2.5.2.2 Форматы блоков сообщений

В режиме КШ МСП поддерживает все форматы сообщений по ГОСТ Р 52070-2003. Для каждого формата МСП требует в пределах блока сообщения запись слов в определенной последовательности. Задана последовательность размещения слова управления сообщением КШ, командного слова и (передаваемых) слов данных, которые предварительно записываются УВ и считываются средствами протокольной логики МСП при обработке сообщения. Дополнительно должны быть зарезервированы слова для сохранения получаемых значений «слова проверочного возврата», ответного слова ОУ, слов данных.

Таблица 1.628 иллюстрирует размещение разрядов в командном слове и таблица 1.629 – в ответном слове ОУ (более подробная информация о командном слове и ответном слове ОУ дана в описании ГОСТ Р 52070-2003). Рисунок 1.206 поясняет структуру блоков сообщений для различных форматов сообщений по ГОСТ Р 52070-2003. Заметим, что для всех форматов сообщений слово управления сообщением КШ размещается в пределах блока сообщения первым.

Таблица 1.628 - Структура командного слова



Наименования разрядов:

rt_a[4..0]	(RT Address), «Адрес ОУ»;
t/r	(transmit/receive), «Приём-передача»;
sa[4..0]	(subaddress), «Подадрес/Режим управления»;
wc[4..0] / mc[4..0]	(word count/ mode code), « Число СД / Код команды».

Таблица 1.629 - Структура ответного слова

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rt_a[4..0]					me	instr	s_sr	reserv	reserv	reserv	bcr	busy	s_sf	s_dbca	tf

Наименования разрядов:

rt_a[4..0] (RT Address), «Адрес ОУ»;
me (Message Error), «Ошибка в сообщении»;
instr (Instrumentation), «Передача ОС»;
s_sr (Service Request), «Запрос на обслуживание»;
reserv (reserved);
reserv (reserved);
reserv (reserved);
bcr (Broadcast Command Received), «Принята групповая команда»;
busy (Busy): «Абонент занят»;
s_sf (Subsystem Flag), «Неисправность абонент»;
s_dbca (Dynamic Bus Control Accept), «Принято управление интерфейсом»;
tf (Terminal Flag), «Неисправность ОУ».

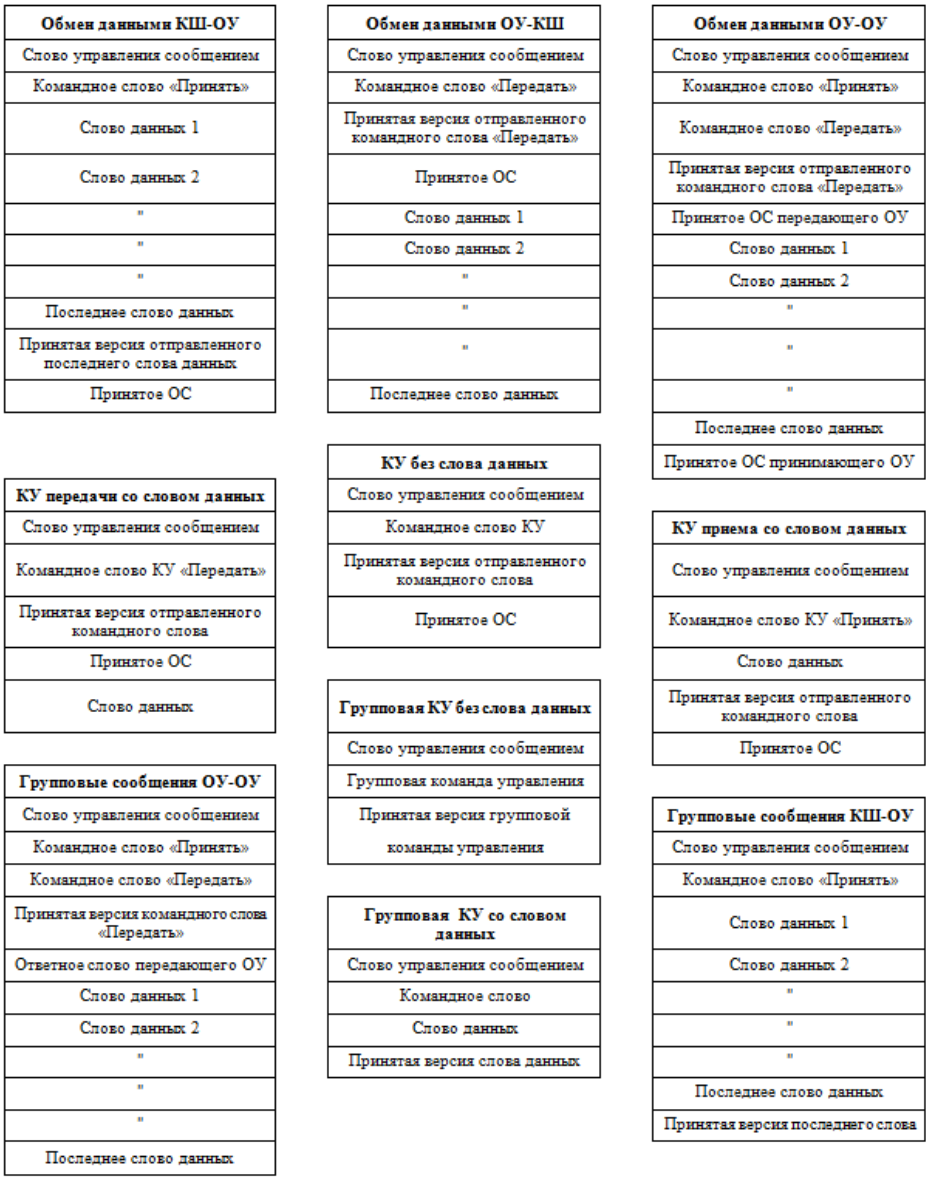


Рисунок 1.206 - Форматы блоков сообщений КШ

Первым словом каждого блока сообщения (с номером «0») является слово управления сообщением КШ. Слово управления сообщением КШ не передается по ЛПИ. Слово управления сообщением КШ является принципиальной управляющей сущностью для каждого сообщения КШ.

После слова управления сообщением КШ записывается командное слово (команда). Для форматов «ОУ-ОУ» и «Группового ОУ-ОУ» это слово является первым из передаваемых по ЛПИ двух командных слов. Это слово читается протокольной логикой и выдаётся в ЛПИ. За первым командным словом может следовать второе командное слово или слова данных, которые должны считываться из ОЗУ и передаваться по ЛПИ. Местоположение в ОЗУ после последнего слова данных, подлежащего передаче, зарезервировано для «слова проверочного возврата» (принятая версия).

Следующие последовательно расположенные ячейки памяти в разделяемом ОЗУ зарезервированы для размещения слов данных, ответных слов, ожидаемых от ОУ. Эти слова сохраняются в указанных местоположениях блока сообщения КШ, если прежде не произошло события «истечения интервала ожидания ОС».

Если тест проверочного возврата не обнаружил ошибки, и ОУ прислало ‘корректное’ ОС («ожидаемые значения» в поле «адрес ОУ» и младших 11 разрядов (признаков)) до истечения заданного интервала времени ожидания ответного слова, и ОУ прислало корректное число достоверных слов данных, то, выполняя «последовательность действий завершения сообщения» (EOM), МСП информирует записью в слово состояния блока сообщения КШ (в описателе блока сообщения), что ‘обработка сообщения завершена, нет ошибок’. Заметим, что для пересылки ОУ-ОУ МСП проверяет ответные слова и от передающего, и от принимающего ОУ.

Слова передаются и принимаются в том порядке, в котором они размещены в блоке сообщения КШ.

1.3.12.2.2.5.2.3 Слово управления сообщением КШ

Первым словом каждого блока сообщения (с номером «0») является слово управления сообщением КШ. Слово управления сообщением КШ не передается по ЛПИ. На рисунке 1.206 представлена структура блоков сообщений для всех возможных форматов сообщений КШ. Слово управления сообщением КШ является принципиальной управляющей сущностью для каждого сообщения КШ. Дополнительно к заданию формата сообщения, слово управления сообщением КШ позволяет задать для конкретного сообщения канал обмена (A/B), режим самотестирования, разрешение ВПС в случае ошибок при предыдущих попытках, разрешение прерывания после завершения работы с данным (избранным) сообщением, маскирование признаков ответного слова ОУ.

Размещение и назначение разрядов слова управления сообщением КШ приведены в таблице 1.630.

Таблица 1.630 - Слово управления сообщением КШ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	mem	srbm	sbbm	sfbm	tfbm	rbm	cw_re	bc_a_b	olst	mbb	eom_ie	1533a_b_s	mcf	bf	rt_rt_f

mem (Message Error Mask), маскирование признака «Ошибка в сообщении»;
srbm (Service Request Bit Mask), маскирование признака «Запрос на обслуживание»;
sbbm (Subsys Busy Bit Mask), маскирование признака «Абонент занят»;
sfbm (Subsys Flag Bit Mask), маскирование признака «Неисправность абонента»;
tfbm (Terminal Flag Bit Mask), маскирование признака «Неисправность ОУ»;
rbm (Reserved Bit Mask), маскирование «Резервных признаков»;
cw_re (Retry Enabled), разрешение ВПС;
bc_a_b (Bus Chanal A/B), выбор канала;
olst (Off Line Self Test), режим внутреннего самотестирования;

mbb (Mask Broadcast Bit), маскирование признака «Принята групповая команда»;
eom_ie (EOM Interrupt Enable), разрешение прерывания после окончания сообщения;
mcf (Mode Code Format), формат КУ;
bf (Broadcast Format), формат группового сообщения;
rt_rt_f (RT-to-RT Format), сообщение формат «ОУ-ОУ».

Когда задан режим КШ, и состояние «расширенных возможностей конфигурирования» (разряд 15 регистра cfg3_ запрограммирован в состояние логической «1») и, кроме того, задано «Разрешение полного слова управления сообщением КШ» (разряд 12 регистра cfg4_ запрограммирован в состояние логической «1»), то значимы все (кроме 15-го и 3-го) разряды слова управления сообщением КШ; в противном случае используются только разряды 7, 6, 5, 2, 1, 0.

При использовании ‘неполного’ слова управления сообщением КШ событие ‘особое состояние ОУ’ указывает на то, что хотя бы один из признаков полученного ОС установлен в лог. «1». Исключение составляет признак ОС «Принята групповая команда» - событие ‘особое состояние ОУ’ возникает если его значение отличается от значения разряда mbb (Mask Broadcast Bit), «маскирование признака ОС «Принята групповая команда»», слова управления сообщением КШ.

При использовании ‘полного’ слова управления сообщением КШ событие ‘особое состояние ОУ’ указывает на то, что в лог. «1» находится один или несколько признаков полученного ОС, для которых соответствующие разряды маски слова управления сообщением КШ в лог. «0».

При использовании ‘полного’ слова управления сообщением КШ значение признаков полученного ОС, для которых соответствующие разряды маски слова управления сообщением КШ в лог. «1», не влияют на формирование события ‘особое состояние ОУ’.

Однако, и при работе с использованием ‘полного’ слова управления сообщением КШ, для признака ответного слова ОУ «Принята групповая команда» разряд «маскирование признака ОС «Принята групповая команда»» слова управления сообщением КШ работает как маска только в случае установки значения лог. «1» разряда «разрешение/XOR* маскирования признака ОС «Принята групповая команда»» (bm_ex, 11-ый разряд регистра cfg4_); при значении лог. «0» разряда bm_ex событие ‘особое состояние ОУ’ возникает при несовпадении значений признака ответного слова ОУ «Принята групповая команда» и разряда «маскирование признака ОС «Принята групповая команда»» (mbb, 5-ый разряда слова управления сообщением КШ).

Разряды маскирования признаков принятого ОС слова управления сообщением КШ позволяет рассматривать различные признаки принятого ОС как ‘существенные’ (при значении лог. «0»), либо как ‘несущественные’ (при значении лог. «1»). Если один или более признаков в принятом ОС, рассматриваемые как ‘существенные’, установлены в лог. «1», то выполнится, из ниже перечисленных, пункт 1, и, возможно, пункты 2, 3, 4:

- Установится в лог. «1» разряд ss (11-й разряд, слова состояния блока сообщения КШ).
- Если разряд m_ss_mc_pt регистра imr в лог. «1», то возникнет запрос прерывания УВ.
- Если разряд m_ss_mc_pt регистра imr в лог. «1», или ei (15-й разряд регистра cfg2_) в лог. «1», то разряд ss_mc_pt регистра isr установится в лог. «1».
- Если re (4-й разряд регистра cfg1_) и ri_ss (9-й разряд регстра cfg4_) и cw_re (8-й разряд слова управления сообщением КШ) все установлены в лог. «1», то будет выполнена последовательность действий ВПС.

mem (Message Error Mask), маскирование признака «Ошибка в сообщении»:

Применяется, только если задан режим КШ, состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1»), и задано «Разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, в лог. «1»).

Если значение разряда mem - лог. «0», то событие ‘особое состояние ОУ’ возникает в случае значения лог. «1» признака «Ошибка в сообщении» в полученном ответном слове; если значение разряда mem - лог. «1», то значение признака «Ошибка в сообщении» в полученном ответном слове не влияет на формирование события ‘особое состояние ОУ’.

srbm (Service Request Bit Mask), маскирование признака «Запрос на обслуживание»:

Применяется, только если задан режим КШ, состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1»), и задано «Разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, в лог. «1»).

Если значение разряда srbm - лог. «0», то событие ‘особое состояние ОУ’ возникает в случае значения лог. «1» признака «Запрос на обслуживание» в полученном ответном слове; если значение разряда srbm - лог. «1», то значение признака «Запрос на обслуживание» в полученном ответном слове не влияет на формирование события ‘особое состояние ОУ’.

sbbm (Subsys Busy Bit Mask), маскирование признака «Абонент занят»:

Применяется, только если задан режим КШ, состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1»), и задано «Разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, в лог. «1»).

Если значение разряда sbbm - лог. «0», то событие ‘особое состояние ОУ’ возникает в случае значения лог. «1» признака «Абонент занят» в полученном ответном слове; если значение разряда sbbm - лог. «1», то значение признака «Абонент занят» в полученном ответном слове не влияет на формирование события ‘особое состояние ОУ’.

sfbm (Subsys Flag Bit Mask), маскирование признака ОС «Неисправность абонента»:

Применяется, только если задан режим КШ, состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1»), и задано «Разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, в лог. «1»).

Если значение разряда sfbm - лог. «0», то событие ‘особое состояние ОУ’ возникает в случае значения лог. «1» признака «Неисправность абонента» в полученном ответном слове; если значение разряда sfbm - лог. «1», то значение признака «Неисправность абонента» в полученном ответном слове не влияет на формирование события ‘особое состояние ОУ’.

tfbm (Terminal Flag Bit Mask), маскирование признака «Неисправность ОУ»:

Применяется, только если задан режим КШ, состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1»), и задано «Разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, в лог. «1»).

Если значение разряда tfbm - лог. «0», то событие ‘особое состояние ОУ’ возникает в случае значения лог. «1» признака «Неисправность ОУ» в полученном ответном слове; если значение разряда tfbm - лог. «1», то значение признака «Неисправность ОУ» в полученном ответном слове не влияет на формирование события ‘особое состояние ОУ’.

rbm (Reserved Bit Mask), маскирование «Резервных признаков»:

Применяется, только если задан КШ, состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1»), и задано «Разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, в лог. «1»).

Если значение разряда rbm - лог. «0», то событие ‘особое состояние ОУ’ возникает в случае значения лог. «1» в одном и более признаков полученного ответного слова: instr («Передача ОС»), reserv[7:5] («Резервные признаки»), s_dbca[1] («Принято управление интерфейсом»).

Если значение разряда rbm - лог. «1», то значение разрядов instr[9], reserv[7:5], s_dbca[1] в полученном ответном слове не влияет на формирование события ‘особое состояние ОУ’.

cw_re (Retry Enabled), разрешение ВПС:

Применяется, только если задано состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1») и задано «Разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, в лог. «1»).

Если значение разряда cw_re лог. «0», то ВПС проводиться не будет.

Если значение разряда cw_re лог. «1» и «разрешение ВПС» (re, 4-ый разряд регистра cfg1_) в лог. «1», то будет предпринято ВПС данного сообщения в ситуации обнаружения ошибки: в

слове состояния блока сообщения КШ в лог. «1» установлен хотя бы один из разрядов: fe (Format Error), «ошибочный формат сообщения», или rto (Response Time out), «истечение времени ожидания ответного слова ОУ».

Если значение разряда cw_re - лог. «1», и «разрешение ВПС» (re, 4-ый разряд регистра cfg1_) - лог. «1», и «ВПС при событии ‘особое состояние ОУ’» (ri_ss, 9-ый разряд регистра cfg4_) - лог. «1», то будет предпринято ВПС данного сообщения в случае «особого состояния ОУ» (ss, 11-ый разряд в слове состояния блока сообщения КШ, в лог. «1»).

Количество ВПС данного сообщения задается значением «два/одно* ВПС» (ds_r, разряд 03 регистра cfg1_) - лог. «0» для однократной попытки повторения, лог. «1» – для двойной попытки повторения. Выбор канала обмена (тот же либо альтернативный) для первой и второй попытки повторения задается значениями разрядов 08 и 07 регистра cfg4_.

bc_a_b (Bus Chanal A/B), выбор канала:

Если значение разряда bc_a_b - лог. «0», то обмен будет осуществляться по каналу обмена B; если значение разряда bc_a_b - лог. «1», то обмен будет осуществляться по каналу обмена A.

olst (Off Line Self Test), режим внутреннего самотестирования:

Значение лог. «1» задает для соответствующего сообщения обмен в режиме автономного (внутреннего) самотестирования. В этом режиме не используется передатчик, нет выдачи сигналов в ЛПИ. Осуществляется непосредственная пересылка информации от выхода кодера на вход декодера для заданного канала обмена. После завершения обработки сообщения результат самотестирования (норма или ошибка) можно определить, прочтя «слово проверочного возврата» блока сообщения и значение «Ошибки проверочного возврата» (ltf, 8-ой разряд слово состояния блока сообщения КШ).

mbb (Mask Broadcast Bit), маскирование признака «Принята групповая команда»:

Если значение разряда mbb - лог. «1» и «разрешение маскирования/XOR* признака «Принята групповая команда»» (bm_ex, 11-ый разряд регистра cfg4_) - лог. «0», то «ожидаемое значение» признака ответного слова «Принята групповая команда» - лог. «1». Это означает, что значение лог. «0» признака «Принята групповая команда» в полученном ответном слове приведет к возникновению события ‘особое состояние ОУ’.

Если задано состояние «расширенных возможностей конфигурирования» (разряд 15 регистра cfg3_ в состоянии логической «1»), и задано «разрешение полного слова управления сообщением КШ» (e_bc_cwe, разряд 12 регистра cfg4_ в лог. «1»), и значение «разрешение маскирования/XOR* признака «Принята групповая команда»» (bm_ex, 11-ый разряд регистра cfg4_) - лог. «1», то разряд mbb взаимодействует с признаком «Принята групповая команда» (из полученного ответного слова) не как операнд логической функции «исключающее или», а в качестве маски признака «Принята групповая команда». В этом случае, при использовании разряда mbb в качестве маски, значение лог. «1» признака «Принята групповая команда» в полученном ответном слове приведет к возникновению события ‘особое состояние ОУ’, если значение разряда mbb – лог. «0»; если же значение разряда mbb – лог. «1», то значение признака «Принята групповая команда» в полученном ответном слове не влияет на формирование события ‘особое состояние ОУ’.

eom_ie (EOM Interrupt Enable), разрешение прерывания после окончания сообщения:

Применяется, только если задано состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1»), и задано «разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, в лог. «1»).

Если значение разряда eom_ie в лог. «1» и значение разряда 04 регистра imr - лог. «1, то после завершения пересылки данного сообщения генерируется запрос прерывания.

mcf (Mode Code Format), формат КУ;

bf (Broadcast Format), формат группового сообщения;

rt_rt_f (RT-To-RT Format), сообщение формат «ОУ-ОУ».

Разряды mcf, bf, rt_rt_f предназначены для задания формата сообщения, как это показано в таблице 1.631. Пример: Если bf[1] = 1’b1, mcf[2] = 1’b0, то это групповая команда, где t/r[10] разряд командного слова определяет наличие СД (t/r =1’b0) в формате сообщения. Если bf[1] =

1’b1, mcf[2] = 1’b1, то это групповая команда управления, где wc/mc[4] разряд командного слова определяет передачу одного слова данных (wc/mc[4] = 1’b1).

Таблица 1.631

mcf [2] формат КУ	bf [1] формат «групповое сообщение»	rt_rt_f [0] формат «ОУ-ОУ»	Формат сообщения
0	0	0	КШ-ОУ, при разряде командного слова «Прием-передача» = 0; ОУ-КШ, при разряде командного слова «Прием-передача»= 1.
0	0	1	ОУ-ОУ
0	1	0	Групповое сообщение
0	1	1	ОУ-ОУ (групповое сообщение)
1	0	0	Команда Управления
1	0	1	Не используется
1	1	0	Команда Управления (групповое сообщение)
1	1	1	Не используется

Слово состояния блока сообщения КШ

УВ может определить состояние обработки каждого сообщения путем чтения четырех слов соответствующего описателя блока сообщения (размещенного в области стека команд). Первое слово содержит слово состояния блока сообщения КШ. В режиме КШ слово состояния блока сообщения содержит информацию о том, находится ли сообщение в процессе обработки (пересылки) или обработка завершена, по какому каналу ведется обмен, обнаружены ли ошибки при обработке.

Разрядные распределение и назначения в слове состояния блока сообщения КШ сведены в таблицу 1.632.

Таблица 1.632 - Структура слова состояния блока сообщения КШ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
eom	som	channel_b_a	ef	ss	fe	rto	ltf	mss	re_co[1..0]		gdbt	wsa_ng	wce	ist	iw

Наименование разрядов:

eom (END-OF-MESSAGE), завершение отработки сообщения;
som (START-OF-MESSAGE), старт отработки сообщения;
channel_b_a (channel b/a), канал обмена B/A;
ef (Error flag), признак ошибки;
ss (Status Set), особое состояние ОУ;
fe (Format Error), ошибочный формат сообщения;
rto (Response Time out), истечение времени ожидания ответного слова ОУ;
ltf (Loop Test Fail), ошибка проверочного возврата;
mss (Masked Status Set), маскируемый признак ‘особое состояние ОУ’;
re_co[1..0] 1) (RETRY COUNT [1..0]), количество ВПС;
gdbt (Good Data Block Transfer), успешная пересылка блока данных;
wsa_ng (Wrong Status Address/No Gap), неверный адрес в ответном слове ОУ или нарушение минимальной паузы перед выдачей ОС;
wce (Word Count Error), ошибка количества слов в сообщении;
ist (Incorrect Sync Type), некорректный тип синхросигнала;
iw (Invalid Word), признак недостоверности слова;

1) В режиме КШ, если проводились ВПС, то слово состояния блока сообщения КШ отображает результат последнего из них.

eom (END-OF-MESSAGE), завершение отработки сообщения:

Устанавливается по завершению сообщения, вне зависимости от наличия ошибки в сообщении.

som (START-OF-MESSAGE), старт отработки сообщения:

Устанавливается в начале и очищается в конце отработки сообщения КШ.

channel_b_a (channel b/a), канал обмена B/A:

Этот разряд будет установлен в лог. «0», если для сообщения применялся канал обмена A, или лог. «1», если канал обмена B.

ef (Error flag), признак ошибки:

Если этот разряд установлен в лог. «1» и один или более из следующих разрядов слова состояния блока сообщения КШ 10, 9 и/или 8 установлен в лог. «1», это указывает на одну или более из следующих ошибок, произошедших при пересылке сообщения, ошибку формата, интервала ожидания ОС и/или теста «проверочного возврата».

ss (Status Set), особое состояние ОУ:

Разряд «особое состояние ОУ», если установлен в лог. «1», указывает, что один из младших 11 разрядов принятого ОС имеет «неожидаемую» величину. «Ожидаемые» значения для этих 11 разрядов – нули.

Имеется одно исключение: если eme, 15-ый разряд регистра cfg3_ («разрешение расширенных возможностей конфигурирования») установлен в лог. «0», или «разрешение маскирования/XOR* признака «Принята групповая команда»» (bm_ex, 11-ый разряд регистра cfg4_) установлен в лог. «0», и разряд «маскирование признака «Принята групповая команда»» в слове управления сообщением КШ в лог. «1», тогда ожидаемым значением признака ответного слова «Принята групповая команда» становится лог. «1», а не лог. «0».

Значение разряда ss не подвержено влиянию со стороны других разрядов масок (разряды с 14 по 9 слова управления сообщением КШ).

fe (Format Error), ошибочный формат сообщения:

Если установлен в лог. «1», то указывает, что принятая порция сообщения содержит одно или более нарушений критериев достоверности сообщения (тип синхронизации, кодирование, контроль по четности, количество разрядов, количество слов и т.д.), или полученное ОС содержит некорректное поле «адрес ОУ».

rto (Response Time out), истечение интервала/времени ожидания ОС:

Если установлен в лог. «1», указывает, что передающее ОУ либо не ответило, либо ответило позднее установленного срока, определяемого значением «интервала ожидания ОС». «Время ожидания ответного слова» определено стандартом как временной промежуток, измеряемый от момента пересечения нулевого уровня сигналом в разряде контроля по четности (середина разряда) командного слова, до момента пересечения нулевого уровня синхросигналом (середина синхросигнала) ответного слова от ОУ.

Если задано состояние «нерасширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0»), то величина времени ожидания ответного слова принимает значение от 17,5 до 19,5 мкс. Если задано состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»), то величина времени ожидания ответного слова может выбираться из следующего перечня – 18,5; 22,5; 50,5 и 130 мкс. (±1 мкс.) посредством задания значений разрядов 10 и 9 регистра cfg5_.

ltf (Loop Test Fail), ошибка проверочного возврата:

Тест «проверочного возврата» производится на предаваемой части каждого сообщения режима КШ. МСП делает проверку корректности принятой версии каждого переданного в ЛПИ слова. К тому же для каждого сообщения в КШ проводится поразрядное сравнение слова, передававшегося последним. Если принятая версия какого-либо слова определена как ошибочная (ошибка синхронизации, кодирования, количества разрядов и/или контроля по чётности) и/или принятая версия последним переданного слова не равна исходной, разряд ltf будет установлен в лог. «1».

Следует отметить, что если «разрешение расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «0»), разряды с 7 по 0 будут всегда возвращать (при чтении) состояния лог. «1».

mss (Masked Status Set), маскируемый признак ‘особое состояние ОУ’:

Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») и «Разрешения полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, запрограммирован в состояние логической «1»).

Будет устанавливаться в лог. «1» в случае наличия хотя бы одного из следующих обстоятельств:

(1) Если один (или более) из «разрядов маскирования» (с 14 по 9) в слове управления сообщением КШ находится в положении лог. «0» и хотя бы один из соответствующих признаков в принятом ответном слове ОУ имеет значение лог. «1». Особую оговорку следует сделать относительно «Резервированных признаков» ОС ОУ: Если «маскирование «Резервированных признаков»» (9-ый разряд слова управления сообщением КШ) в лог. «0» и оказался установленным в лог. «1» хотя бы один какой-нибудь из трёх «Резервированных признаков» в принятом «ответном слове ОУ», то это означает происшествие обнаружения «Маскируемого признака ‘особое состояние ОУ’», чему соответствует установление данного разряда (mss) в лог. «1»;

(2) Если «разрешение маскирования/XOR* признака «Принята групповая команда»» (bm_ex, 11-ый разряд регистра cfg3_) в лог. «1», и «маскирование признака «Принята групповая команда»» в слове управления сообщением КШ для сообщения в лог. «0», и признак «Принята групповая команда» в полученном ОС ОУ находится в лог. «1».

re_co[1..0] (RETRY COUNT 1 and RETRY COUNT 0), количество ВПС:

Применяются только когда режим «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1»), и «разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_) установлен в лог. «1», и «разрешение ВПС» (re, 4-ый разряд регистра cfg1_) установлен в лог. «1», и «разрешение ВПС» (cw_re, 8-ой разряд слова управления сообщением КШ) установлен в лог. «1».

Следует отметить, что «количество ВПС» (re_co[1..0]) также подвержено влиянию разряда «ВПС при наличии ‘особого состояния ОУ’» (9-ый разряд регистра cfg4_). Возможности задавать различные количества ВПС разрядами re_co[1..0] представлены в таблице 1.633.

Таблица 1.633 - Задание количества ВПС

re_co1 (6-ой разряд)	re_co0 (5-ый разряд)	Количество ВПС
0	0	0
0	1	1
1	0	2
1	1	Не используется

gdbt (Good Data Block Transfer), успешная пересылка блока данных:

Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1»).

Разряд gdbt устанавливается в значение лог. «1» после корректного (безошибочного) завершения передачи сообщения в форматах «ОУ-КШ», «ОУ-ОУ» или после корректного выполнения «КУ передачи со словом данных».

Разряд gdbt всегда устанавливается в значение лог. «0» в случае завершения передачи сообщения в форматах «КШ-ОУ», «КУ приема со словом данных», «КУ без данных», а также после недостоверного сообщения. Результаты «проверочного возврата» не влияют на значение разряда gdbt. Разряд gdbt может использоваться для определения безошибочности передаваемой части сообщения при обмене ОУ-ОУ. Завершение обмена ОУ-ОУ с установленными в лог. «1»

разрядами «Успешная пересылка данных» и «признак ошибки» в слове состояния блока сообщения КШ означает, что передающее ОУ ответило корректно, однако была обнаружена ошибка в принимаемой части сообщения.

wsa_ng (Wrong Status Address/No Gap), неверный адрес в ответном слове ОУ или нарушение минимальной паузы перед выдачей ОС:

Применяется только в режиме «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1»).

Этот разряд устанавливается в лог. «1», если происходит что-либо из следующего перечня:

разряд «разрешение проверки минимальной паузы перед выдачей ОС» (gce, 8-ой разряд регистра cfg5_) установлен в лог «1» и передающий ОУ отвечает ОС через время меньшее, чем 4 мкс (согласно ГОСТ Р 52070-2003 - это промежуток между серединами зон разряда контроля по четности последнего КС/СД и синхроимпульса ОС), т.е. реально проверяется, что период отсутствия декодируемых сигналов в ЛПИ (‘мёртвое время’) меньше 2 мкс.;

и/или значение поля «Адрес ОУ» в ответном слове ОУ не равно значению поля «Адрес ОУ» в командном слове.

wce (Word Count Error), ошибка количества слов в сообщении:

Применяется только в режиме «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1»).

Используется для пересылок типа «ОУ-к-КШ», «ОУ-ОУ» или «КУ передачи со словом данных».

В случае лог. «1» указывает на то, что отвечающий ОУ не передал заявленного количества слов данных.

Будет всегда устанавливаться в значение лог. «0» после обработки пересылки типа «КШ-к-ОУ», сообщения с «КУ приёма» или сообщения с «КУ передачи без слова данных».

ist (Incorrect Sync Type), некорректный тип синхросигнала:

Применяется только в режиме «расширенные возможности конфигурирования» МСП» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1»).

В случае лог. «1» указывает, что ОУ передал синхроимпульс КС/ОС в слове данных или, наоборот, синхроимпульс, предназначенный для слова данных, был передан ОУ в ОС.

iw (Invalid Word), признак недостоверности слова:

Применяется только в режиме «расширенные возможности конфигурирования» МСП» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1»).

В случае лог. «1» указывает, что ОУ отозвался сообщением с одним или более словом, содержащем одну или более ошибку следующих типов: ошибка синхроимпульса, ошибка кодирования, ошибка контроля по четности и/или ошибка количества разрядов.

1.3.12.2.2.5.2.4 Задание промежутка времени для сообщения

Слово промежутка времени для сообщения (длительность сообщения) в описателе блока сообщения используется в состоянии «расширенных возможностей конфигурирования» КШ (разряд 15 регистра cfg3_ запрограммирован в состояние логической «1») и задано «разрешение задания промежутка времени для сообщения» (imgte, разряд 05 регистра cfg1_ запрограммирован в состояние лог. «1»). Слово промежутка времени для сообщения задает интервал времени от старта текущего сообщения до старта следующего, другими словами – задаёт промежуток времени на отработку данного сообщения (включая возможные ВПС). Дискрет значений слова промежутка времени для сообщения составляет 1 мкс на младший разряд. Максимальное значение слова задаёт 65535 мкс.

Если не задано «разрешение задания промежутка времени для сообщения» (imgte, разряд 05 регистра cfg1_ запрограммирован в состояние лог. «0»), то значение этого слова игнорируется.

Если значение слова промежутка времени для сообщения задаёт продолжительность меньшую, чем требуется для завершения текущего сообщения, то обработка следующего сообщения будет начинаться непосредственно после завершения обработки текущего

сообщения. В этом случае промежуток времени сообщения будет реально равен времени обработки данного сообщения плюс время минимальной паузы между сообщениями, приблизительно равной от 8 до 11 мкс. (от завершения предыдущего сообщения до начала следующего).

Механизм задания промежутка времени для сообщения позволяет реализовать пересылку кадра с программируемой временной диаграммой без вмешательства УВ.

1.3.12.2.2.5.2.5 Автоповтор отработки кадра

При работе в режиме «автоповтор отработки кадра» УВ должен инициализировать слово начальное значение указателя на стек команд и слово начальное значение счетчика сообщений для активной области ОЗУ. Слово начальное значение указателя на стек команд и слово начальное значение счетчика сообщений необходимо записывать лишь один раз до запуска обработки кадра в режиме «автоповтор отработки кадра». До старта в режиме «автоповтор отработки кадра» УВ должен записать ненулевое значение в регистр длительности кадра КШ.

В режиме автоматического повторения кадра темп повторения может программироваться как значением регистра длительности кадра КШ, так и от внешнего сигнала. Дополнительно, есть возможность задания промежутка времени для сообщения, которая определяется как время от старта текущего сообщения до старта последующего и программируется (задается) на основе индивидуального сообщения.

Возможность автоповтора кадра используется, только если задан режим КШ, состояние «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1»). Оставшееся время старта повтора кадра сообщений КШ находиться в регистре остатка времени кадра КШ (bcft [15..0], адрес регистра – 5’b01011).

Настройка КШ на работу в режиме автоповтора кадра требует, чтобы по фиксированным адресам ОЗУ были записаны начальное значение указателя на стек команд и начальное значение счетчика сообщений (по адресам 0102 и 0103 для области А, либо по адресам 0106 и 0107 для области Б), а не значения указателя на стек команд и счетчика сообщений (по адресам 0100 и 0101 для области А, либо по адресам 0104 и 0105 для области Б).

При наступлении события повтора кадра (задаваемого программой УВ, счетчиком времени кадра КШ, или внешним сигналом) МСП копирует значения слов начальное значение указателя на стек команд и начальное значение счетчика сообщений в слова указателя на стек команд и счетчика сообщений (для текущей активной области А, либо Б), после чего начинает обработку кадра. Текущая активная область задается значением ca_b_a, 13-го разряда регистра cfg1_. Два раздельных местоположения для этих значений необходимы, поскольку значения указателя на стек команд и счетчика сообщений модифицируются МСП в процессе обработки кадра.

В режиме автоповтора кадра период повторения кадра программируется заданием значения регистра длительности кадра КШ (bcft_rtlc_mttw [15..0], адрес регистра – 5’b01101). Дискретность программирования составляет 100 мкс/МЗР, что обеспечивает диапазон изменения вплоть до максимальной величины 6,5535 с.

Для автоповтора кадра необходимо, чтобы регистр bcft_rtlc_mttw [15..0] был проинициализирован не нулевым значением. Если регистр bcft_rtlc_mttw [15..0] равен нулю, то автоповтор отработки кадра не начнется.

1.3.12.2.2.5.2.6 Минорный и мажорный кадры

Во многих системах от КШ требуется обработка сообщений для различных ОУ и для различных подадресов с варьируемой периодичностью. Например, некоторые сообщения требуется передавать с частотой 5 Гц, другие – 10 Гц, 20 Гц, …, 100 Гц. Общим механизмом для обеспечения варьируемого темпа обмена является использование минорных и мажорных фреймов.

На рисунке 1.207 показан сценарий построения пересылки сообщений, организованный в минорных кадрах. Мажорный кадр заключает в себе множество минорных кадров. Если

предполагается, что длительность минорного кадра составляет 10 мс, может быть поддержан темп обмена до 100 Гц. То есть сообщения с темпом 100 Гц необходимо помещать в каждом минорном кадре, с частотой 50 Гц - через кадр, в то время как с темпом 1 Гц - в каждый сотый минорный кадр.

Использованная архитектура в КШ МСП дает возможность набирать минорные кадры посредством слова промежутка времени для сообщения. Слово промежутка времени для сообщения точно описывает (специфицирует) временной промежуток от начала текущего сообщения до начала последующего. Временем минорного кадра является сумма времен отработки всех сообщений внутри минорного кадра. Времена всех минорных кадров задаются одинаковыми (например, 10 мс).

Если время, требуемое для полноценной обработки сообщения, оказалось большим, чем в слове промежутка времени для сообщения, то обработка сообщения, включая возможные ВПС, будет выполнена (это приведёт к превышению времени отработки кадра). Результирующий промежуток времени перед следующим сообщением будет равняться приблизительно от 8 до 11 мс. Если использование ВПС задано, рекомендуется программировать слово промежутка времени для сообщения с запасом, таким образом, чтобы позволительно было делать одно или два ВПС. Это обеспечит возможность автоматически производить ВПС без нарушения временных рамок кадра.

Мажорный кадр заключает в себе группу минорных кадров. Для выполнения мажорного кадра можно задать «режим автоматического повтора кадров». В этой ситуации время кадра МСП становится временем мажорного кадра системы. На практике может оказаться желательным синхронизировать мажорные кадры от внешнего по отношению к МСП источника интервалов времени. Тогда ite, 6-ой разряд регистра cfg1_, необходимо запрограммировать в лог. «0», а ete, 7-ой разряд регистра cfg1_, - в лог. «1». В такой конфигурации запуск мажорного кадра будет происходить по нарастающему фронту сигнала на входном сигнале msp_ssflag_ext_trig.

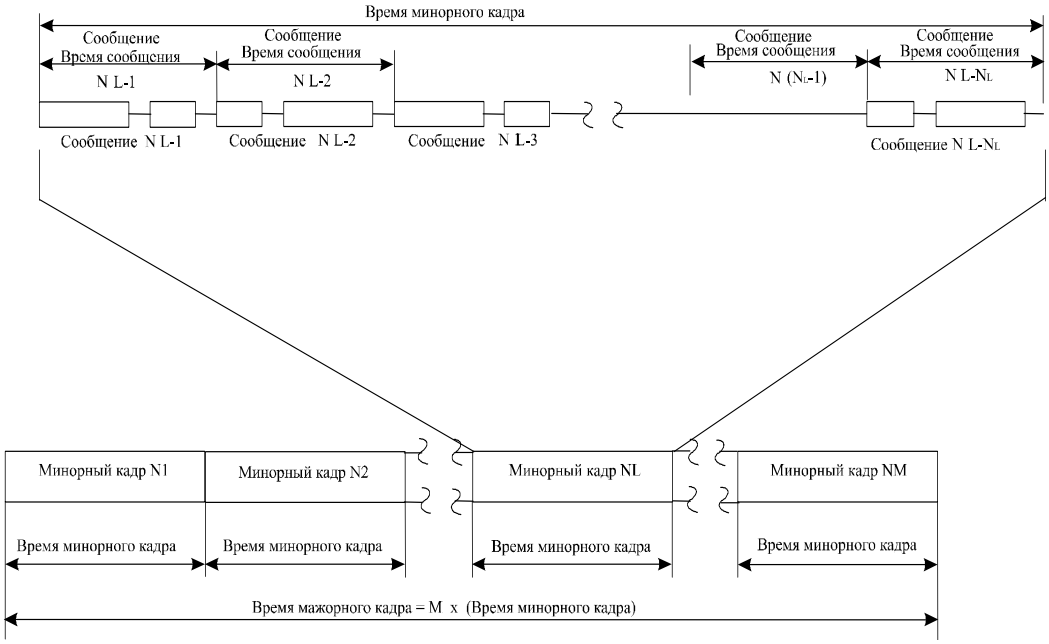


Рисунок 1.207 - Минорные и мажорные кадры.

1.3.12.2.2.5.2.7 Различные возможности старта работы КШ

МСП предлагает ряд разновидностей по управлению кадрового обмена КШ. МСП может быть запрограммирован на пересылку как единственного кадра, так и множества повторяющихся кадров. Вдобавок запуск первого кадра может инициироваться и командой программного запуска, и нарастающим краем импульса на входном сигнале msp_ssflag_ext_trig.

Конфигурирование особенного порядка запуска КШ в работу осуществляется посредством разрядов 8, 7 и 6 регистра cfg1_.

Если разряд far (Frame Auto-Repeat), «автоповтор отработки кадра», (8-ой разряд регистра cfg1_) в лог. «0» или, если ite (Internal Trigger Enable), «разрешение старта кадра КШ от внутреннего счетчика времени кадра КШ», (6-ой разряд регистра cfg1_) в лог. «0», то МСП будет отрабатывать обмен только для единственного кадра. Кадр может содержать от одного до 512 сообщений. Для дополнительной информации обращайтесь к таблице 1.634.

Если разряд far (Frame Auto-Repeat), «автоповтор отработки кадра», (8-ой разряд регистра cfg1_) в лог. «1» и ite (Internal Trigger Enable), «разрешение старта кадра КШ от внутреннего счетчика времени кадра КШ», (6-ой разряд регистра cfg1_) в лог. «1», то КШ запрограммирован на работу с автоматическим повторением кадров. При этом, слово «начальное значение указателя на стек команд» и слово «начальное значение счетчика сообщений» необходимо записывать лишь один раз до запуска обработки кадра в режиме «автоповтор отработки кадра». До старта в режиме «автоповтор отработки кадра» УВ должен записать ненулевое значение в регистр длительности кадра КШ.

Временная диаграмма на рисунке 1.207 иллюстрирует определение времени кадра МСП для режима автоповтора кадров. Отметим, что время кадра КШ определёно как время от начала первого сообщения (сообщение №1) для одного кадра до начала первого сообщения в последующем кадре.

В режиме «автоповтор отработки кадра» протокольная логика КШ МСП обращается для считывания к местоположениям слов начального указателя на стек команд и начального счётчика сообщений при каждом старте отработке кадра и переписывает их значения в соответствующие ячейки памяти указатель на стек команд и счётчик сообщений.

Для старта в режиме одиночного кадра или для запуска первого кадра в режиме автоповтора, в любом случае, первое сообщение может запускаться программой; то есть прописыванием 0002(шестн.) в регистр запуска/сброса scr. Вдобавок, если ete (External Trigger Enable), «разрешение старта кадра КШ от внешнего сигнала», (7-ой разряд регистра cfg1_) в лог. «1», то обработку первого кадра можно стартовать от внешнего сигнала, то есть подачей нарастающего перепада напряжений входной сигнал msp_ssflag_ext_trig.

Таблица 1.634 - Последовательности действий при запуске в режиме КШ

far[8] (регистр cfg1_) «Автоматич. повторение кадров»	ete[7] (регистр cfg1_) «Разрешение старта КШ от внешнего сигнала»	ite[6] (регистр cfg1_) «Разрешение старта КШ от внутреннего счетчика времени кадра КШ»	Последовательность, порядок запускающих действий
0	0	X	Одиночный кадр; запускается программно; указатель на стек команд и счётчик сообщений должны загружаться по адресам 100,101 (или 104,105) перед началом обработки каждого кадра.
0	1	X	Одиночный кадр; запускается либо программно, либо внешним импульсом, поступающим на входной сигнал msp_ssflag_ext_trig; указатель на стек команд и счётчик сообщений должны загружаться по адресам 100,101 (или 104,105) перед началом обработки каждого кадра.
1	0	0	Одиночный кадр; запускается программно; указатель на стек команд и счётчик сообщений должны быть загружены по адресам 102, 103 (или 106, 107) один раз.

Продолжение таблицы 1.634

far[8] (регистр cfg1_) «Автоматич. повторение кадров»	ete[7] (регистр cfg1_) «Разрешение старта КШ от внешнего сигнала»	ite[6] (регистр cfg1_) «Разрешение старта КШ от внутреннего счетчика времени кадра КШ»	Последовательность, порядок запускающих действий
1	0	1	Повторяющийся кадр; первый кадр запускается программно; кадры повторяются под управлением внутреннего устройства, с использованием регистра длительности кадра КШ. Указатель на стек команд и счётчик сообщений должны быть загружены по адресам 102, 103 (или 106, 107) один раз.
1	1	0	Одиночный кадр; запускается либо программно, либо внешним импульсом, поступающим на входной сигнал msp_ssflag_ext_trig; указатель на стек команд и счётчик сообщений должны быть загружены по адресам 102, 103 (или 106, 107) один раз.
1	1	1	Повторяющийся кадр; первый кадр запускается либо, либо внешним импульсом, поступающим на входной сигнал msp_ssflag_ext_trig; кадры повторяются под управлением внутреннего устройства, с использованием регистра длительности кадра КШ; указатель на стек команд и счётчик сообщений должны загружаться по адресам 102, 103 (или 106, 107) один раз.

1.3.12.2.2.5.2.8 Прерывания в режиме КШ

КШ МСП предлагает набор условных обстоятельств, которые могут приводить к формированию запроса на обработку прерывания процессору УВ. Они могут быть выбраны средствами регистра маскирования прерываний imr. Следует отметить, что, если разряд ei (Enhanced Interrupts), «разрешение расширенных возможности прерывания», регистра cfg2_ установлен в лог. «1», то УВ может определить, какое из прерывающих событий произошло, опрашивая регистр состояния прерывания (isr). В этом режиме работы МСП различные разряды в регистре isr могут устанавливаться в единичные значения, указывая на происшествие определённых событий, вне зависимости от программирования регистра маскирования прерывания (imr). Сведения о назначении восьми возможных прерывающих обстоятельств для КШ сведены в нижеследующем описании:

Завершение обработки кадра сообщений: Для большинства применений КШ после завершения обработки кадра, состоящего из множества сообщений, бывает желательно порождать запрос на прерывание, что разрешается установкой «завершение обработки кадра сообщений» (m_bc_eof, 3-ий разряд регистра imr). В обоих режимах, и в «режиме единственного кадра», и в «режиме автоповтора кадров», это прерывание будет порождаться после выполнения обработки каждого кадра сообщений.

Завершение отработки сообщения (EOM): Это прерывание вырабатывается по завершению каждого сообщения. Для сообщений с ВПС прерывание происходит после последнего ВПС.

Завершение обработки избранного сообщения КШ: Сигнал запроса прерывания УВ может быть выработан по завершении отработки избранного сообщения.

Это разрешается, если одновременно выполняются все следующие условия:

- состояние «расширенных возможностей конфигурирования» разряд eme (разряд 15 регистра cfg3_) установлен в лог. «1»;

- разряд e_bc_cwe «разрешение полного слова управления сообщением КШ» (разряд 12 регистра cfg4_) установлен в лог. «1»;

- «завершение обработки избранного сообщения» (m_scw_e, 4-ый разряд регистра imr) в лог. «1»;

- разряд eom_ie (EOM Interrupt Enable), «разрешение прерывания после окончания сообщения» (разр.4) соответствующего слова управления сообщением КШ установлен в лог. «1».

Запрос прерывания порождается в конце обработки текущего сообщения.

Ошибка формата сообщения: Это прерывание случается после обработки сообщения, содержащего недостоверное слово или ошибочное количество слов (подробнее см. описание соответствующего разряда регистра isr).

ВПС в режиме КШ: Это событие/условие происходит после отработки последнего ВПС (неважно, успешно завершённого или нет) для какого-нибудь сообщения (подробнее см. описание соответствующего разряда регистра isr).

Особое состояние ОУ: Это прерывание будет порождаться МСП, если ОС, полученное от отвечающего абонента, содержит один или более признаков ОС установлен в лог «1» и не замаскированных разрядами слова управления сообщением КШ. Прерывание ss_mc_pt также будет выработано, если ОС содержит неверный «Адрес ОУ». В режиме КШ, состояние «расширенных возможностей конфигурирования» признаки ОС ОУ, которые были замаскированы соответствующими им разрядами при программировании слова управления сообщением КШ, не будут вызывать возникновения события ‘особое состояние ОУ’ (подробнее см. описание соответствующего разряда регистра isr и описание разрядов слова управления сообщением КШ и Таблицу 6.10).

‘Прокручивание’ стека команд КШ/ОУ: Это прерывание случается, когда содержимое указателя на стек команд КШ прокручивается, достигнув предопределённого граничного значения в адресном пространстве ОЗУ МСП. Размер стека команд задаётся программно и может принимать значения: 256, 512, 1024 или 2048 слов (подробнее см. описание соответствующего разряда регистра isr).

Переполнение счетчика метки времени: Это прерывание случается, когда содержимое счётчика метки времени ‘прокручивается’ от своего максимального значения ffff к нулевому.

Кроме того, в режиме КШ могут использоваться запросы прерывания по следующим событиям: «затягивание работы передатчика», «истечение времени ожидания подтверждения (квитирования)»

1.3.12.2.2.5.3 Другие функции КШ

1.3.12.2.2.5.3.1 Автоматическое возобновление передачи сообщения (ВПС)

ВПС осуществимо КШ в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»);

ВПС задаются для индивидуального сообщения программированием в лог. «1» разряда cw_re, Retry Enabled, «разрешение ВПС» в слове управления сообщением КШ.

Если значение разряда cw_re лог. «1» и «разрешение ВПС» (re, 4-ый разряд регистра cfg1_) в лог. «1», то будет предпринято ВПС данного сообщения в ситуации обнаружения ошибки: в слове состояния блока сообщения КШ в лог. «1» установлен хотя бы один из разрядов fe (Format Error), «ошибочный формат сообщения», или rto (Response Time out), «истечение времени ожидания ответного слова ОУ».

Если значение разряда cw_re - лог. «1», и «разрешение ВПС» (re, 4-ый разряд регистра cfg1_) - лог. «1», и «ВПС при событии ‘особое состояние ОУ’» (ri_ss, 9-ый разряд регистра cfg4_) - лог. «1», то будет предпринято ВПС данного сообщения в случае ‘особого состояния ОУ’.

Количество ВПС данного сообщения задается значением «два/одно* ВПС» (ds_r, разряд 03 регистра cfg1_) - лог. «0» для однократной попытки повторения, лог. «1» – для двойной попытки повторения.

Выбор канала обмена (тот же либо альтернативный) для первой и второй попытки повторения задается значениями разрядов 08 и 07 регистра cfg4_.

Зависимость разрядов слова состояния блока сообщения от ВПС. EOM для сообщений с ВПС происходит после завершения последнего из возможных ВПС, а не после исходного

неудавшегося сообщения или после неуспешного первого из двух ВПС. МСП записывает слово состояния блока сообщения в ОЗУ во время ЕОМ последовательности действий. «Единственное ВПС» (5-ый разряд слова состояния блока сообщения КШ), находясь в единичном состоянии, указывает на, то, что состоялось, по крайней мере, одно ВПС. Лог. «1» в «двойное ВПС» (6-ой разряд слова состояния блока сообщения КШ) указывает на два ВПС. Логические нули в «Единственном ВПС» и в «двойном ВПС» говорят о том, что ни одного ВПС для сообщения не выполнялось. Если последнее ВПС терпит неудачу, обработка сообщения прекращается, а в слово состояния блока сообщения заносится значение, указывающее причину неудачного завершения обработки (истечение интервала ожидания ОС, ошибка формата сообщения, ‘особое состояние ОУ’ и т.п.).

Прерывания по ВПС. Если запросы на обслуживание для ЕОМ и/или ВПС разрешены регистром маскирования прерываний, они будут вырабатываться после выполнения ВПС для сообщения, вне зависимости от успешности их выполнения.

Если ВПС разрешены для отдельного сообщения, они будут отрабатываться, даже если один или более из следующих разрядов регистра cfg1_ (разряды с 12 по 9) запрограммирован в лог. «1»:

m_soe - (Message Stop-On-Error): «останов работы после окончания текущего сообщения в случае обнаружения ошибки»;

f_soe- (Frame Stop-On-Error): «останов работы после окончания текущего кадра КШ в случае обнаружения ошибки»;

ss_som - (Status Set Stop-On-Message): «останов работы после окончания текущего сообщения в случае обнаружения ‘особого состояния ОУ’»;

ss_sof - (Status Set Stop-On-Frame): «останов работы после окончания текущего кадра КШ в случае обнаружения ‘особого состояния ОУ’».

Полагая, что m_soe или f_soe установлены в единицы и, что проводимые ВПС безуспешны, обработка сообщений будет остановлена по завершению текущего сообщения или текущего кадра сообщений. Тем не менее, обработка последующих сообщений КШ внутри кадра и/или последующих кадров КШ будет продолжена, если неудавшееся сообщение было успешно обработано при выполнении ВПС.

1.3.12.2.2.5.3.2 Время ожидания ОС в режиме КШ

По определению ГОСТ Р 52070-2003 пауза перед выдачей ответного слова (определяет «интервал ожидания ОС» в режиме КШ) соответствует промежутку времени от момента пересечения нулевого уровня серединой сигнала последнего перед паузой разряда слова до момента пересечения нулевого уровня серединой синхросигнала следующего за паузой слова и должна быть от 4 до 12 мкс.

Для режима КШ величина «интервала ожидания ОС» программно задаваема. В состоянии «нерасширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «0») величина этого промежутка равна 18,5 мкс. Для возможности приспособиться, при необходимости, к условиям передачи по «длинным» линиям различной протяжённости, в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1») величина интервала ожидания ОС переменна и может устанавливаться УВ; её можно задать посредством регистра cfg5_, как это показано в таблице 1.635.

Таблица 1.635 - Выбор интервала ожидания ОС (регистр cfg5_)

Выбор интервала ожидания ОС		Величина интервала ожидания ОС
Разряд 10	Разряд 9	(мкс)
0	0	18,5
0	1	22,5
1	0	50,5
1	1	130

Замечание - для состояния «нерасширенных возможностей конфигурирования» определена величина 18,5 мкс.

1.3.12.2.2.5.3.3 Маскирование признаков ОС в режиме КШ

Находясь в режиме КШ, МСП способен маскировать избранные признаки (разряды) ОС для отдельного сообщения в кадре КШ; это осуществимо в режиме «расширенных возможностей конфигурирования (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») и «разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_, установлен в лог. «1»). Для состояния «нерасширенных возможностей конфигурирования» или, если «разрешение полного слова управления сообщением КШ» в лог. «0», действующими, в слове управления сообщением КШ, являются лишь 7, 6, 5, 2, 1 и 0 разряды. В этом случае принимающим участие в «маскировании состояния» является лишь 5-ый разряд - «маскирование признака «Принята групповая команда»».

Если «разрешение полного слова управления сообщением КШ» (e_bc_cwe, 12-ый разряд регистра cfg4_) в лог. «1», то 14 разрядов слова управления сообщением КШ (разряды с 14 по 0, кроме 3-го), включая все «маскирующие» (разряды с 14 по 9, и 6) активны, то есть воздействуют на поведение КШ. Разряды в слове управления сообщением КШ, маскирующие признаки ОС, определяют, какие из признаков соответственного ОС будут вызывать возникновение события ‘особое состояние ОУ’. Происшествие ‘особого состояния ОУ’, в свою очередь, вызовет установление разряда в слове состояния блока сообщения КШ, может вызвать прерывание, может остановить работу КШ после завершения текущего сообщения или по завершению текущего кадра, и/или инициировать выполнение ВПС.

Разряд ss («особое состояние ОУ») слова состояния блока сообщения КШ, если установлен в лог. «1», указывает, что один из младших 11 признаков принятого ОС имеет «неожидаемую» величину. «Ожидаемые» значения для этих 11 признаков – нули, если эти признаки не замаскированы в слове управления сообщением КШ.

Значение лог. «1» разрядов маскирования в слове управления сообщением КШ делает ‘несущественными’ соответствующие признаки полученного ОС. Любое назначенное подмножество из следующих признаков может учитываться при формировании события ‘особое состояние ОУ’: «Ошибка в сообщении», «Запрос на обслуживание», «Абонент занят», «Неисправность абонента», «Неисправность ОУ», «Резервные признаки» (к резервным отнесены и разряды «Передача ОС», «Принято управление интерфейсом»). Признак «Принята групповая команда» может либо маскироваться (проигнорирован), либо проверяться на равенство лог. «0» или лог. «1». Если соответствующий разряд маски запрограммирован в лог. «0», то данный разряд включён в рассмотрение при формировании события ‘особое состояние ОУ’.

Исключая признак «Принята групповая команда» ОС, сведения о влиянии состояния «расширенных возможностей конфигурирования, «разрешения полного слова управления сообщением КШ», различных значений принятого ОС и соответственных разрядов маскирования в слове управления сообщением КШ на происшествие события ‘особое состояния ОУ’, при соответствующем признаке ОС в лог. «1», сведены в таблицу 1.636.

Таблица 1.636 - Маскирование признаков ОС (за исключением признака «Принята групповая команда»)

Разряд eme «разрешение расширенных возможностей конфигури-рования» (регистр cfg3_)	Разряд e_bc_cwe «разрешение полного слова управления сообщении-ем КШ» (регистр cfg4_)	Соответ-ствующий разряд маскирова-ния отдельного признака ОС (слово управления сообщением КШ)	Разряд ss «особое состоя-ние ОУ» (слово состоя-ния блока сообще-ния КШ)	Разряд mss «маскиру-емый признак ‘особое состояние ОУ’» (слово состояния блока сообщения КШ)	Разряд ss_mc_pt Запрос прерывания ‘особое состояние ОУ’, (регистр isr), при m_ss_mc_pt в лог. «1»	Описание
0	x	x	1	1	1	В этом режиме, КШ ожидает ОС только с нулевыми признаками. Разряды с 7 по 0 слова состояния блока сообщения КШ при eme[15]= 0 всегда равны лог. «1». Маскирование признаков ОС не происходит.
1	0	x	1	0	1	В этом режиме, КШ ожидает ОС только с нулевыми признаками ОС.
1	1	0	1	1	1	В этом режиме, КШ ожидает ОС только с нулевыми признаками.
1	1	1	1	0	0	В этом режиме КШ, признаки ОС маскируются соответствующими разрядами слова управления сообщением КШ. Признаки ОС instr[9] и s_dbca[1] маскируются разрядом «Маскирование «Резервных признаков»».

Функция разряда «маскирование признака «Принята групповая команда»» определяется значением разряда «разрешение маскирования/XOR* признака «Принята групповая команда»» (bm_ex, 11-ый разряд регистра cfg4_). Если bm_ex запрограммирован в лог. «1», то величина лог. «1» в разряде «маскирование признака «Принята групповая команда»» (5-разряд слова управления сообщением КШ) будет вызывать безразличие у КШ к признаку «Принята групповая команда» в ОС ОУ при формировании события ‘особое состояние ОУ’, в то время как лог. «0» в разряде «маскирование признака «Принята групповая команда»» будет порождать событие ‘особое состояние ОУ’, если признак «Принята групповая команда» ОС в лог. «1». Программирование bm_ex в лог. «0» заставит МСП сравнивать между собой значения признаков «Принята групповая команда» в ответном слове ОУ и «маскирование признака «Принята групповая команда»» (5-ый разряд слова управления сообщением КШ). В случае их неравенства, будет вырабатываться событие ‘особое состояние ОУ’.

Работа разрядов «разрешение маскирования/XOR* признака «Принята групповая команда»» и «маскирование признака «Принята групповая команда»» проиллюстрирована в таблице 1.637.

Таблица 1.637 - Маскирование признаков ОС (включая признак «Принята групповая команда»)

Разряд eme «разрешение расширенных возможностей конфигурирования» (регистр cfg3_)	Разряд bm_ex «разрешение маскирования/XOR* признака «Принята групповая команда»» (регистр cfg4_)	Разряд e_bc_cwe «разрешение полного слова управления сообщением КШ» (регистр cfg4_)	Разряд mbb «маскирование признака «принята групповая команда»» (слово управления сообщением КШ)	Признак «Принята групповая команда» (ответное слово ОУ)	Разряд ss «особое состояние ОУ» (слово состояния блока сообщения КШ)	Разряд mss «маскируемый признак ‘особое состояние ОУ’» (слово состояния блока сообщения КШ)	Разряд ss_mc_pt Запрос прерывания «особое состояние ОУ», (регистр isr), при m_ss_mc_pt в лог. «1»
0	X	X	0	0	-	-	-
				1	1	-	1
			1	0	1	-	1
				1	-	-	-
1	0	0	0	0	-	-	-
				1	1	-	1
			1	0	1	-	1
				1	-	-	-
1	0	1	0	0	-	-	-
				1	1	1	1
			1	0	1	1	1
				1	-	-	-
1	1	0	0	0	-	-	-
				1	1	0	1
			1	0	-	-	-
				1	0	-	-
1	1	1	0	0	-	-	-
				1	1	1	1
			1	0	-	-	-
				1	0	-	-

Примечание: Выбранная конфигурация не устанавливает разряды в лог. «1», логические значения которых не указаны.

Слово состояния блока сообщения КШ содержит два разряда, указывающие на ‘особое состояние ОУ’. Первый - «особое состояние ОУ» (11-ый разряд); данный разряд будет установленным в лог. «1», если один из разрядов ОС находится в лог. «1» вне зависимости от маскирующих разрядов слова управления сообщением КШ. Второй - «маскируемый признак ‘особое состояние ОУ’» (7-ой разряд) - будет устанавливаться в лог. «1» только, если разряд в ОС ОУ установлен в лог. «1» и не замаскирован средствами слова управления сообщением КШ (за исключением возможности установки ‘ожидаемого’ значения для признака «Принята групповая команда», как было описано выше).

Прерывание будет порождаться для события ‘особого состояния ОУ’ (с учётом разрядов маскирования в слове управления сообщением КШ) в состоянии «расширенных возможностей конфигурирования» МСП, если разряд m_ss_mc_pt «особое состояние ОУ» (1-ый разряд регистра маскирования прерывания) находится в лог. «1». Прерывание будет происходить по окончании текущего сообщения (EOM), а не сразу после получения «ответного слова» (случай с командой «передача»).

Событие ‘особое состояние ОУ’, для состояния «расширенных возможностей конфигурирования (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»), может вызывать остановку работы КШ в двух случаях. Случай первый: если разряд «останов работы после окончания текущего сообщения в случае обнаружения ‘особого состояния ОУ’» (ss_som, 10-ый разряд регистра cfg1_) установлен в лог. «1», КШ будет останавливать обработку текущего кадра по завершении текущего сообщения. Случай второй: находясь в режиме «автоповтора кадра», КШ будет останавливать работу по завершении обработки последнего сообщения кадра, если разряд «останов работы после окончания текущего кадра КШ в случае обнаружения ‘особого состояния ОУ’» (ss_sof, 9-ый разряд регистра cfg1_) установлен в лог. «1».

1.3.12.2.2.5.4 Программная процедура инициализации КШ

Нижеследующая программная последовательность - типичные шаги, которые следует выполнять процессору УВ сразу после включения питания для того, чтобы настроить (сконфигурировать) МСП на работу в режиме КШ; для большинства применений многие из перечисленных ниже шагов могут быть пропущены:

(1) Произвести программный сброс записью величины 0001 в регистр запуска/сброса;

(2) Если должна быть использована хотя бы одна из функций режима «расширенные возможности конфигурирования» (например использование программируемых ВПС, промежутка времени для сообщения, полного слова управления сообщением КШ и пр.), то необходимо задать «разрешение расширенных возможностей конфигурирования» МСП, записав 8000(шестн.) в регистр cfg3_ .

(3) Присвоить начальное значение регистру маскирования прерывания (imr). Как правило, для большинства применений КШ прерывание по событию «завершение отработки сообщения» разрешают. Разряд «завершение отработки сообщения» (EOM) может порождать генерацию запроса прерывания после завершения каждого выполненного сообщения. Прерывание «завершение обработки избранного сообщения для режимов КШ/ОУ» позволяет порождать запрос прерывания после выполнения избранного (посредством слова управления сообщением КШ) сообщения. «ВПС в режиме КШ» вызовет генерацию запроса прерывания после отработки ВПС для сообщения, вне зависимости от успешности его выполнения. «Ошибка формата сообщения» будет приводить к запросу прерывания для какого-либо ОС, содержащего хотя бы одну из следующих ошибок: ошибку синхронизации, кодирования, количества разрядов, контроля по чётности или неверное количество слов. Прерывание «особое состояние ОУ» обуславливается одним из следующих обстоятельств:

- Если разряд «разрешение полного слова управления сообщением КШ» (разряд 12 регистра cfg4_) запрограммирован в лог. «0») не разрешено и один или более из 11-ти признаков ОС в лог. «1»; или

- Если разряд «разрешение полного слова управления сообщением КШ» (разряд 12 регистра cfg4_) запрограммирован в лог. «1»), и для одного или более из признаков ОС соответственный маскирующий разряд в слове управления сообщением КШ запрограммирован в лог. «0», и величина полученного в ОС ОУ самого разряда равна лог. «1».

(4) Присвоить начальное значение первому регистру конфигурирования cfg1_. Разряды 15 и 14, оба, должны быть запрограммированы в лог. «0», чтобы сконфигурировать МСП в режим КШ. Должна быть назначена текущая (активная) область ОЗУ – разряд (ca_b_a). Разряды «останов работы после окончания текущего сообщения в случае обнаружения ошибки» (m_soe), «останов работы после окончания текущего кадра КШ в случае обнаружения ошибки» (f_soe), «останов работы после окончания текущего сообщения в случае обнаружения ‘особого состояния ОУ’» (ss_som), «останов работы после окончания текущего кадра КШ в случае обнаружения ‘особого состояния ОУ’» (ss_sof) следует программировать соответственно текущим требованиям. Разряд m_soe разрешает остановку (по концу текущего сообщения) в случае следующих ошибок: ошибка в типе синхроимпульса, кодирования, контроля по чётности, некорректность адреса ОУ в ОС, большее или меньшее количество слов, превышено время ожидания ОС, «ошибка проверочного возврата».

Если желательно стартовать кадр КШ, используя входной сигнал msp_ssflag_ext_trig, то разряд ete «разрешение старта кадра КШ от внешнего сигнала» (7-й, регистра cfg1_) необходимо установить в лог. «1». Вне зависимости от состояния ete и разряда «разрешение старта кадра КШ от внутреннего счетчика времени кадра КШ» (ite, регистр cfg1_) кадр КШ первоначально может быть запущен командой программного старта КШ - записью в регистр запуска/сброса (srr) значения с разрядом bc_mt_st в лог. «1». Если это желательно, чтобы запрограммированный кадр сообщений выполнялся, автоматически повторяясь (без вмешательства процессора УВ), необходимо разряды «автоповтор отработки кадра» (far) и разряд ite запрограммировать в лог. «1». Если желательно стартовать первый кадр сообщений посредством входного сигнала msp_ssflag_ext_trig и получать автоматически повторяемые кадры в последующем, разряды far, ite и ete следует запрограммировать в лог. «1». Для того, чтобы регулировать время разрыва между сообщениями, разряд «разрешение задания промежутка времени для сообщения» (imgte) необходимо установить в лог. «1». Если imgte в лог. «0», промежуток времени, начиная от старта текущего сообщения и, кончая стартом последующего, будет таким, что разрыв между сообщениями КШ составит разрешенную требованиями ГОСТ Р 52070-2003 величину, задаваемую по умолчанию, и равную приблизительно 9 мкс. Если выполнение ВПС для конкретного сообщения проводить необходимо, то ВПС следует разрешить посредством 8-го разряда в слове управления сообщением КШ (re, регистр cfg1_); в этот разряд необходимо занести лог. «1». Если ВПС разрешены, максимальное количество ВПС для каждого сообщения (одно или два) задаются посредством «количество ВПС - два/одно» (ds_r, регистр cfg1_).

(5) Присвоить начальное значение второму регистру конфигурирования - cfg2_. В режиме КШ настоятельно рекомендуется, чтобы «запрещение 256-словных ограничений» (256_wbd, 10-ый разряд) был запрограммирован в лог. «1»; это позволяет наилучшим образом использовать адресное пространство МСП для размещения блоков сообщений. При работе УВ с прерываниями, рекомендуется устанавливать в лог. «1» разряд ei (Enhanced Interrupts) «разрешение расширенные возможностей прерывания»; разряд «вид сигнала msp_int - импульс/уровень» (l_p_ir) следует установить в лог. «1», если процессор УВ воспринимает уровневый, а не импульсный тип сигнала прерывания. Разрядами «Разрешение (цена МЗР) для счетчика метки времени» (tt_rsl[2..0]) необходимо установить требуемое разрешение для работы регистра счётчика метки времени. По умолчанию разряды tt_rsl[2..0] принимают значение «000», задавая тем самым разрешение 64 мкс/МЗР.

(6) Присвоить начальное значение третьему регистру конфигурирования - cfg3_. Если какие-либо из младших 15-ти разрядов этого регистра или какие-либо другие свойства расширенного режима конфигурирования МСП должны быть использованы, «разрешение расширенных возможностей конфигурирования» (eme, 15-ый разряд) всё время необходимо удерживать в логической «1» при любом обращении по записи в этот регистр. Размер стека команд программируется посредством разрядов 14 и 13 и может принимать одну из следующих величин - 256 (по умолчанию, для 64 сообщений), 512, 1024 и 2048 слов (для 512 сообщений).

(7) Присвоить начальное значение четвёртому регистру конфигурирования - cfg4_. Для того, чтобы можно было полностью использовать разряды слова управления сообщением КШ, разряд «разрешение полного слова управления сообщением КШ» (e_bc_cwe) должен быть установлен в лог. «1». Это позволит выполнить задание функции маскирования, ВПС и прерываний на основе индивидуального сообщения. «Разрешение маскирования/XOR* признака «Принята групповая команда»» (bm_ex) следует устанавливать в лог. «1», если хочется использовать разряд «маскирование признака «Принята групповая команда»» (5-ый разряд) слова управления сообщением КШ именно в целях маскирования, а не для спецификации в принятом ОС ОУ ожидаемого значения признака «Принята групповая команда».

(8) Присвоить начальное значение пятому регистру конфигурирования - cfg5_. По умолчанию МСП считает, что на его тактовый вход (вход msp_clk16) подаётся синхросигнал частотой 16 МГц.

Разряды «выбор интервала ожидания ОС» (rts[1..0]) выбирают величину временного промежутка, разрешающего ожидание ОС для КШ МСП, среди следующих величин: 18,5 мкс (по

умолчанию); 22,5; 50,5 или 130 мкс. Если разряд «разрешение проверки минимальной паузы перед выдачей ОС» (gce) установить в лог. «1», то КШ станет проверять выдержку ОУ минимальной паузы перед выдачей ОС; нарушение требований проверки вызовет установление разряда «ошибки формата сообщения» в слове состояния блока сообщения, в результате чего, может вырабатываться прерывание и/или ВПС.

(9) Присвоить начальное значение пятому регистру конфигурирования – cfg6_. Установка в лог. «1» разряда eca (включение режима ускоренного доступа УВ) позволит УВ производить циклы обращения к регистрам или памяти МСП во время SOM/EOM последовательности.

(10) Присвоить начальное значение регистру счетчика метки времени, если требуется. Следует отметить - значение регистра счетчика метки времени может быть сброшено в нулевое значение записью лог. «1» в разряд tt_rst, «сброс значения регистра счетчика метки времени», одновременно с формированием команды программного старта КШ, как составляющая часть шага 15.

(11) Если используется режим автоповтора кадра (ite, 6-разряд регистра cfg1_, и far, 8-разряд регистра cfg1_, запрограммированы в лог. «1»), то необходимо проинициализировать ненулевым значением регистр длительности кадра КШ (bcft_rtlc_mttw[15..0]).

Этот регистр-счётчик имеет разрешение 100 мкс/МЗР.

(12) Загрузить стартовое местоположение стека команд в местоположение для указателя на стек команд в активной области ОЗУ (А или Б).

Если автоповтор кадров задействован, также следует загрузить начальное значение указателя на стек команд в активной области ОЗУ.

(13) Проинициализировать стек команд активной области. Для описателя блока сообщения для каждого сообщения, которое должно быть отработано, адрес блока сообщения, указывающий на первое слово соответствующего блока сообщения КШ (слово управления сообщением КШ) должно быть записано процессором УВ.

Если используется свойство ‘задания промежутка времени для сообщения, то слово промежутка времени для сообщения должно быть записано в описателе блока сообщения для каждого сообщения. Цена разряда слова промежутка времени для сообщения - 1 мкс/МЗР. Загруженное в это слово значение задаёт время, протекающее от начала текущего сообщения до старта последующего сообщения (т. е. время отработки данного сообщения). Если это время оказалось меньшим, чем требуемое для отработки сообщения, то пауза перед началом отработки следующего сообщения по умолчанию устанавливается равной 9 мкс. Если желательно проводить опрос стека команд во время обработки кадра сообщений, то процессору УВ значения слова промежутка времени для сообщения следует прописать нулями.

(14) Загрузить управляющие, командные, информационное слова, которые должны быть переданы, в соответствующие блоки сообщений КШ. По желанию, местоположения для слов, ожидаемых при проведении проверки возврата, записывается УВ содержимым, отличающимся от значения последнего слова сообщения, которое подлежит отправлению в ЛПИ. Процессор УВ должен убедиться, что в блоке сообщения имеется достаточное количество ячеек после последнего слова, подлежащего передаче, и зарезервированных для возможности размещения в них ожидаемых слова проверочного возврата, ОС ОУ и информационных слов.

(15) В местоположение счётчик сообщений, в активной области ОЗУ, следует загрузить (в дополнительном коде) значение числа сообщений, которое необходимо обработать в кадре.

Если используется режим автоповтора кадров, то в слово начальное значение счётчика сообщений также следует загрузить требуемое значение.

(16) Если «разрешение старта кадра КШ от внешнего сигнала» (ete, 7-ой разряд регистра cfg1_) запрограммирован в лог. «0», то кадр сообщений должен запускаться программной командой старта КШ. Это делается путём записи величины 0002 (шестн.) в регистр запуска / сброса. Если «разрешение старта кадра КШ от внешнего сигнала» (ete, 7-ой разряд регистра cfg1_) запрограммирован в лог. «1», то кадр сообщений может запускаться и от программной команды старта КШ, и от нарастающего края импульса на входном сигнале msp_ssflag_ext_trig.

1.3.12.2.2.5.5 Исключительные обстоятельства КШ

В своих откликах на различные ошибки сообщений и прочие исключительные события КШ МСП производит определённые действия и оповещает о случившемся многочисленными признаками, или флажками. Таблица 1.638 показывает перечень всевозможных для КШ исключительных ситуаций (событий) и соответственные отклики на них, их воздействия на разряды слова состояния блока сообщения КШ, отработку ВПС, а также их влияние на разряды регистра состояния прерывания и на порождение самих запросов прерывания.

Таблица 1.638 - События при работе МСП в режиме КШ

	Разряды слова состояния блока сообщения КШ	ВПС (если разрешено)	Прерывания, разряды регистра состояния прерывания	Другие действия
Нет ОС	ef, rto, re_co[1..0] (если ВПС разрешено)	Да	fe, bc_r (если разрешено)	
Принято неправильное количество слов	ef, wce, fe, re_co[1..0] (если ВПС разрешено)	Да	fe, bc_r (если разрешено)	
Ошибка типа синхросигнала в принятом слове	ef, ist, fe, re_co[1..0] (если ВПС разрешено)	Да	fe, bc_r (если разрешено)	
Принято недостоверное слово (ошибка кодирования, количества битов, контроля по чётности)	ef, fe, iw, re_co[1..0] (если ВПС разрешено)	Да	fe , bc_r (если разрешено)	См. замеч. 1
Неверный адрес ОУ в принятом ОС	ef, fe, wsa_ng, re_co[1..0] (если ВПС разрешено)	Да	ss_mc_pt, bc_r (если разрешено)	
Ошибка квитирования. Замечание: в режиме КШ ошибка квитирования не будет обнаруживаться во время отработки последовательности SOM, EOM или ВПС. Ошибка квитирования может произойти только во время передачи второго Командного слова (для передачи ОУ-ОУ), Информационного слова или ответного слова ОУ. Ошибка квитирования может случиться в «Прозрачном» режиме, а не в «буферизованном» режиме.	слово состояния блока сообщения КШ, полученное при выполнении SOM последовательности	Нет	hf	Кадр КШ окончен
Пауза перед выдачей ОС меньше 4 мкс. и разряд «разрешение расширенных возможностей конфигурирования» (15-ый разряд регистра cfg3_) в лог. «1» и разряд «разрешение проверки минимальной паузы перед выдачей ОС» (разряд 8, регистра cfg5_) в лог. «1»	ef, fe, wsa_ng, re_co[1..0] (если ВПС разрешено)	Да	fe, bc_r (если разрешено)	
Ошибка проверочного возврата	ef, ltf	Нет	fe	
Задержка передатчика: передатчик пытается передавать дольше, чем 668 мкс	Слово состояния блока сообщения сохраняется во время отработки последовательности SOM.	Нет	bt_tt	Кадр КШ окончен

Продолжение таблицы 1.638

	Разряды слова состояния блока сообщения КШ	ВПС (если разрешено)	Прерывания, разряды регистра состояния прерывания	Другие действия
Разряд «разрешение расширенных возможностей конфигурирования» (eme) в лог. «0» или разряд «разрешение полного слова управления сообщением КШ» (e_bc_cwe) в лог. «0» и один или более признаков в ОС ОУ установлен.	ss, re_co[1..0] (если ВПС разрешено, если ri_ss, разряд 9 регистра cfg4_, лог. «1»)	Да, если ri_ss, разряд 9 регистра cfg4_, лог. «1»	ss_mc_pt, bc_r (если ВПС разрешены для ss)	
Разряд «разрешение расширенных возможностей конфигурирования» (eme) в лог. «1», разряд «разрешение полного слова управления сообщением КШ» (e_bc_cwe) в лог. «1», и один или более признаков в ОС ОУ установлено в лог. «1», при соответствующих разрядах маскирования в слове управления сообщением КШ в лог. «0».	ss, mss, re_co[1..0] (если ВПС разрешены для ss)	Да, если ri_ss, разряд 9 регистра cfg4_, лог. «1»	ss_mc_pt, bc_r (если ВПС разрешены для ss)	
	Разряды слова состояния блока сообщения	ВПС (если разрешено)	Прерывания, Разряды «регистра состояния прерывания»	Другие действия
Разряд «разрешение расширенных возможностей конфигурирования» (eme) в лог. «1», разряд «разрешение полного слова управления сообщением КШ» (e_bc_cwe) в лог. «1», ив ОС ОУ в лог. «1» установлены те, один или более, признаки, для которых соответственные разряды маскирования флагов ошибки принятого ответного слова ОУ в слове управления сообщением КШ также установлены в лог. «1», и не установлено в лог. «1» ни одного признака в ОС ОУ, для которых соответственные разряды маскирования в слове управления сообщением КШ в лог. «0».	ss	Нет	Нет изменений	
В отклике на КС передачи СД, при разряде v_me_nd («разрешение ОУ не передавать СД в ситуации ‘ошибка в сообщении’», регистр cfg4_) в лог. «0» , получено ОС с установленным в лог. «1» разрядом «Ошибка в сообщении»; после ОС СД отсутствуют.	ef, ss, mss (см.замеч. 2), fe, wce, re_co[1..0] (если ВПС разрешено)	Да	fe, ss_mc_pt (см.замеч. 2), bc_r (если разрешено)	

Продолжение таблицы 1.638

	Разряды слова состояния блока сообщения КШ	ВПС (если разрешено)	Прерывания, разряды регистра состояния прерывания	Другие действия
В отклике на КС передачи СД, при разряде v_me_nd («разрешение ОУ не передавать СД в ситуации ‘ошибка в сообщении’», регистр cfg4_) в лог. «0» , получено ОС с установленным в лог. «1» разрядом «ошибка в сообщении»; после ОС поступило правильное количество СД.	ss, mss (см.замеч. 2), re_co[1..0] (если ВПС разрешено)	Да (если ВПС разрешено, разряд ri_ss регистра cfg4_в лог. «1» и верно замеч. 2)	ss_mc_pt (см.замеч. 2) , bc_r (если ВПС разрешено, разряд ri_ss регистра cfg4_в лог. «1» и верно замеч. 2)	

Замечания:

1) если КШ МСП принимает недостоверное СД, будет продолжаться прием всех последующих слов сообщения без сохранения в ОЗУ. КШ будет ждать до освобождения ЛПИ перед выполнением последовательности EOM, или выполнением ВПС, или передачей следующего сообщения.

2) если разряд «разрешение полного слова управления сообщением КШ» (e_bc_cwe) в лог. «0» или разряд, маскирующий ошибку сообщения в слове управления сообщением КШ, в лог. «0».

1.3.12.2.2.5.6 Обслуживание МСП в процессе обработки кадров КШ

Завершение обработки сообщений или кадров КШ может определяться УВ с помощью опроса или прерывания. Существует несколько приемов ведения опроса, в которых задействуются и регистр cfg1_, и регистр состояния прерывания isr. Вдобавок могут опрашиваться указатель на стек команд и счётчик сообщений из активной области ОЗУ. Указатель на стек команд увеличивается на 4 при завершении отработки (EOM) каждого сообщения КШ. По окончании кадра КШ он будет указывать на 4 ячейки дальше (по модуль размера стека команд) от первого слова описателя блока сообщения (положение слова состояния блока сообщения) для последнего сообщения в кадре. Счетчик сообщений увеличивается на единицу по окончании каждого сообщения. После завершения обработки кадра слово счётчик сообщений должен содержать величину FFFF (шестн.).

В состоянии «разрешение расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») КШ младшие три разряда регистра cfg1_ могут опрашиваться для определения состояния КШ. Разряд «флаг работы в режиме КШ» (bc_e, разряд 2 регистра cfg1_) будет возвращать при чтении лог. «1» после того, как обработка кадра КШ была запущена на выполнение (программной командой старта КШ или от внешнего запуска) и вплоть до момента завершения единственного кадра или остановки обработки повторяющихся кадров. После чего разряд «флаг работы в режиме КШ» перейдет в лог. «0».

Разряд «флаг процесса отработки кадра» аналогичен разряду «флаг работы в режиме КШ» лишь с одним исключением: в режиме автоповтора кадров «флаг процесса отработки кадра» (bc_fip, разряд 1 регистра cfg1_) будет возвращать лог. «0» в промежутке после выполнения последовательности действий завершения отработки сообщения (EOM) для последнего сообщения кадра и вплоть до начала последовательности действий «старт отработки сообщения» (SOM) для первого сообщения последующего кадра.

Разряд «флаг процесса отработки сообщения» (bc_mip, разряд 0 регистра cfg1_) будет возвращать лог. «1» между SOM последовательностью и EOM последовательностью действий для каждого обрабатываемого сообщения.

Если расширенные возможности прерывания разрешены (ei, 15-разряд регистра cfg2_ в лог. «1») и регистр маскирования прерывания (imr) запрограммирован в лог. «0» (запрещая запросы прерывания от МСП), тем не менее регистр состояния прерывания isr может использоваться для опроса УВ. В число этих событий входят: «’прокручивание’ стека команд КШ/ОУ» (разряд 12), «ВПС в режиме КШ» (разряд 8), «переполнение счетчика метки времени» (разряд 6), «завершение обработки избранного сообщения для режимов КШ/ОУ» (разряд 4), «завершение обработки кадра сообщений» (разряд 3), «ошибка формата сообщения» (разряд 2), «особое состояние ОУ» (разряд 1), «завершение отработки сообщения» (разряд 0).

Отметим, что любое из событий, перечисленных выше, может быть использовано для генерации сигнала запроса прерывания процессора УВ.

Прерывающие обстоятельства «ошибка формата сообщения» и «ВПС в режиме КШ» могут использоваться для информирования УВ о каких-либо ошибочных происшествиях на ЛПИ. «Особое состояние ОУ» может быть использовано для указания УВ, что один из незамаскированных признаков в принятом ОС находился в лог. «1» или неверный «Адрес ОУ» в ОС. Отдельному разряду ОС может разрешаться вызывать «особое состояние ОУ» программированием соответствующего разряда маскирования признаков ОС в слове управления сообщением КШ в лог. «0».

Разряды маскирования признаков ОС могут быть использованы совместно с разрядом «останов работы после окончания текущего сообщения в случае обнаружения ‘особого состояния ОУ’» (10-й разряд регистра cfg1_).

При этом (или в случае «останов работы после окончания текущего сообщения в случае обнаружения ошибки) УВ способен вставлять асинхронные сообщения. Примером такого асинхронного сообщения, после останова работы, является обработка в другой активной области ОЗУ КШ кадра, содержащего единственное сообщение - команду управления – ‘Передать векторное слово’. После того, как УВ обработал это сообщение и получил векторное слово ОУ, он может вернуться к обработке прерванного кадра.

1.3.12.2.2.5.7 Прекращение обработки кадра КШ

В большинстве случаев для запрограммированного кадра КШ будут обрабатываться все запрограммированные сообщения кадра. То есть МСП будет продолжать обработку сообщений вплоть до момента, когда счетчиком сообщений достигается (путем принудительной инкрементации) значение FFFF (шестн.). В режиме автоповтора кадров с разрядом «разрешение старта кадра КШ от внутреннего счетчика времени кадра КШ» в лог. «1», заданием начальных значений счётчика сообщений и указателя на стек команд, при ненулевом значении регистра длительности кадра КШ последующий кадр КШ будет автоматически перезапускаться, после завершения текущего кадра.

МСП поддерживает несколько технических приемов для прекращения обработки кадра. Разряды с 12 по 9 регистра cfg1_ обеспечивают механизм автоматической обработки остановки КШ МСП. Если установить в лог. «1» разряды «останов работы после окончания текущего сообщения в случае обнаружения ошибки» и «останов работы после окончания текущего сообщения в случае обнаружения ‘особого состояния ОУ’», то это вызовет прекращение обработки сообщений по завершению текущего сообщения, если при его отработке были обнаружены ошибка или ‘особое состояние ОУ’. Если МСП был запрограммирован в режим автоповтора кадров, «останов работы после окончания текущего кадра КШ в случае обнаружения ошибк» и «останов работы после окончания текущего кадра КШ, в случае обнаружения ‘особого состояния ОУ’» принудят КШ МСП прекратить дальнейшую обработку сообщений по окончании текущего кадра, если обнаруживается ошибка или ‘особое состояние ОУ’.

Некоторые разряды в регистре запуска/сброса обеспечивают средство и возможность для процессора УВ самостоятельно прекратить обработку сообщений КШ УВ. Установка в лог. «1»

разряда «программный сброс МСП» будет вызывать немедленное завершение отработки сообщения, то есть МСП будет прекращать обработку сообщения, не завершив его предписанным способом. Установка в лог. «1» разряда «остановить работу после завершения текущего сообщения КШ» или «остановить работу после завершения текущего кадра сообщений КШ» принуждают МСП завершить обработку сообщения по окончании текущего сообщения или кадра.

1.3.12.2.2.5.8 SOM, EOM и ВПС последовательности действий КШ

По началу и завершению каждого сообщения и перед выполнением ВПС МСП производит последовательность действий - пересылок в/из разделяемого с процессором УВ ОЗУ.

Последовательность «старт отработки сообщения» (SOM) КШ

1: Значение «указатель на стек команд» читается из местоположения для указателя на стек команд области А/Б. Этот адрес употребляется для доступа к описателю блока сообщения в стеке команд активной области.

2: Если разряд eme регистра cfg3_ и разряд imgte регистра cfg1_ установлены в лог. «1», то слово промежутка времени для сообщения выбирается из третьей ячейки описателя блока сообщения.

3: Слово указателя блока сообщения выбирается из четвертой ячейки описателя блока сообщения. Этот указатель содержит адрес начала блока сообщения КШ (адрес слова управления сообщением КШ).

4: Слово управления сообщением КШ выбирается из первой ячейки блока сообщения КШ.

5: (Первое) командное слово выбирается из второй ячейки блока сообщения КШ.

6: Слово метки времени записывается во вторую ячейку описателя блока сообщения.

7: Слово состояния блока сообщения КШ записывается в первую ячейку описателя блока сообщения.

Последовательность завершения сообщения (EOM) КШ

1: Слово счётчик сообщений области А/Б считывается из местоположения счетчика активной области ОЗУ. Если величина слова счётчик сообщений меньше, чем FFFF (шестн.), то МСП самостоятельно увеличивает ее на единицу. Если величина этого слова равна FFFF (шестн.), то текущее сообщение является последним сообщением кадра и текущий кадр завершится после окончания текущей EOM последовательности.

2: Значение регистра счетчика метки времени записывается во вторую позицию описателя блока сообщения.

3: Слово состояния блока сообщения КШ записывается в первую позицию описателя блока сообщения.

4: Значение счётчика сообщений записывается в местоположение слова счетчик сообщений области А/Б.

5: Величина «указателя на стек команд» увеличивается на четыре. Обновленная величина записывается в ячейку указателя на стек команд области А/Б.

Последовательность ВПС сообщения КШ

Последовательность ВПС сообщения КШ совпадает с SOM-последовательностью КШ. Если разряд eme регистра cfg3_ и разряд imgte регистра cfg1_ установлены в лог. «1», то слово промежутка времени для сообщения будет считываться из ОЗУ, но использоваться не будет.

1.3.12.2.2.5.9 Самотестирование КШ с выдачей и без выдачи в ЛПИ

МСП имеет ряд свойств для обеспечения самотестирования.

МСП наделён тестовым средством проверочного возврата, которое обеспечивает возможность проведения самотестирования как в процессе выполнения реальной работы, так и вне рабочего режима.

Самотестирование работы КШ без выдачи в ЛПИ (т.е. без передачи команд и данных в ЛПИ) обеспечивает средство для использования и проверки параллельных эл. цепей адреса и

данных, кодера, цепей «транспортировки» последовательных данных, декодера, части логики подсчитывания слов и управления форматом сообщения, управляющего автомата протокольной логики и логики управления памятью. Во время проведения самотестирования без выдачи в ЛПИ МСП не производит выдачу во внешнюю ЛПИ. Вместо этого выход кодера, работающего по правилам бифазового манипулирования (манчестер 2), подсоединяется напрямую ко входу декодера. Внутреннее самотестирование (без выдачи в ЛПИ) может проводиться после включения питания как часть системного самотестирования или, когда МСП работает в режиме ОУ, - как часть программно управляемого отклика на КУ ‘Установить ОУ в исходное состояние’. Общим элементом самотестирования с выдачей в ЛПИ и без выдачи в ЛПИ является метод проверки возврата (возвратившегося слова). В каждом случае проверяется достоверность принимаемых версий для всех переданных слов (синхронизация, кодирование, контроль по четности, количество разрядов) и принятая версия последним переданного КШ слова сравнивается с его исходной (переданной) версией. Индицируется «ошибка проверочного возврата» в слове состояния блока сообщения КШ, если не подтверждается достоверность или сравнение слов выявило различия в принятой и переданной версиях.

Для того, чтобы выполнить самотестирование на отдельном сообщении, 6-ой разряд слова управления сообщением КШ («режим внутреннего самотестирования») должен быть запрограммирован в лог. «1». Исходом обработки негруппового сообщения в процессе самотестирования явится установление в лог. «1» разряда «истечение интервала ожидания ОС» в слове состояния блока сообщения для этого сообщения. После выполнения самотестирующего группового сообщения «истечение интервала ожидания ОС» не будет установлен в лог. «1». Если самотестирование проходит с удачным исходом, разряд «ошибка проверочного возврата» в слове состояния блока сообщения не будет установлена лог. «1». Другая часть, определяющая успешность проведения самотестирования, - подтверждение равенства содержимого слова проверочного возврата и переданной версии для последнего информационного слова сообщения.

Для того чтобы обеспечить «режим внутреннего самотестирования» с исчерпывающей, по возможности, полнотой, необходимо обработать в этом режиме большое количество сообщений. Чтобы задействовать всевозможные подмножества протокольной логики и логики управления памятью МСП, начальные адреса блоков сообщений, форматы сообщений, количества слов данных и содержимое последних слов, подлежащих передачи, для отдельных сообщений должны отличаться, разниться, между собой.

Во время проведения самотестирования без выдачи в ЛПИ может быть также задействован механизм прерывания по событию «ошибка формата сообщения», «завершение отработки сообщения», «завершение обработки избранного сообщения для режимов КШ/ОУ», «завершение обработки кадра сообщений КШ». Это может быть сделано программированием соответствующих разрядов регистра маскирования прерывания. Если МСП сконфигурирован в режим передачи кадра из множества сообщений, состоящего из самотестирующихся без выдачи в ЛПИ сообщений, то прерывания будут происходить после каждого сообщения, если разрешено прерывание по событию «завершение отработки сообщения».

Аналогично, запрос обработки прерывания по «ошибке формата сообщения» будет порождаться после обработки каждого негруппового сообщения из-за возникновения события «истечение интервала ожидания ОС». Если прерывание по событию «завершение отработки сообщения» запретить и разрешить прерывание по событию «завершение обработки кадра сообщений», то запрос прерывания УВ будет порождаться только после обработки последнего сообщения в кадре.

Свойство «останов работы после окончания текущего сообщения в случае наличия ошибки» может быть проверено установлением в лог. «1» соответствующего разряда в регистре cfg1_, заданием для МСП обработку кадра из множества сообщений, обеспечивающих выполнение самотестирования без выдачи в ЛПИ и проверкой исполнения остановки обработки сообщений после завершения первого негруппового сообщения кадра.

Работоспособность свойств «разрешение задания промежутка времени для сообщения» и слов метки времени могут быть проверены собственными силами (самостоятельно) путем

задания значений «Слово промежутка времени для сообщения» для сообщений в кадре и проверки, после отработки кадра, затраченного времени на обработку кадра посредством чтения значений слов метки времени из соответствующих описателей блока сообщения.

Автоматическое выполнение ВПС может быть проверено, так как ВПС должны вырабатываться МСП, если разрешены, в ответ на событие «истечение интервала ожидания ОС», которые происходят при обработке негрупповых самотестирующих без выдачи в ЛПИ сообщений.

Для самотестирования с выдачей в ЛПИ последнее слово, переданное ППУ, обрабатывается приемником ППУ КШ и поставляется обратно на вход декодера для формирования «слова проверочного возврата». По аналогии с тестом без выдачи в ЛПИ разряд «ошибка проверочного возврата» в слове состояния блока сообщения становится либо установленным в лог. «1», либо сброшенным в лог. «0» в результате проведения проверки ‘возврата’, а само ‘слово проверочного возврата’ заносится в блок сообщения в разделяемое ОЗУ. Вдобавок к цепям, задействованным при самотестировании без выдачи в ЛПИ, самотестирование с выдачей в ЛПИ проверяет работу передатчиков, приемников и саму линию передачи информации на большом протяжении: внешние трансформаторы, отводы от магистральной ЛПИ, кабель магистральной ЛПИ, ‘развязывающие коробки’ и согласующие сопротивления на концах. Несогласованность электрических цепей или прочие неисправности внешних компонентов будут вызывать отражение сигналов, их искажения; эти искажения могут быть причиной неудачного исхода в проведении ‘проверки возвратившегося слова’.

1.3.12.2.2.6 Работа в режиме оконечного устройства (ОУ)

1.3.12.2.2.6.1 Краткое описание свойств ОУ МСП

ОУ МСП обладает следующими возможностями:

выполнение обработки всех форматов сообщений и всех команд режима управления по ГОСТ Р 52070-2003;

поддержка функций взаимодействия по резервированному (дублированному) мультиплексному каналу передачи данных;

задание допустимости КС;

выполнение функции «занятости» для конкретного подадреса;

формирование слова ВСК;

исчерпывающая проверка наличия ошибок, в том числе ошибок передачи «ОУ-ОУ».

Важным качеством ОУ МСП является наличие следующих возможностей:

1) самостоятельное формирование ОС и слова ВСК в соответствии с правилами ГОСТ Р 52070-2003;

2) программное управление ОС и словом ВСК;

3) отработка УВ особых ситуаций по прерыванию и/или по опросу, с использованием регистра состояния прерывания.

Собственный адрес ОУ задаётся с помощью входных сигналов. Чтобы сделать возможной работу ОУ, на шесть входных сигналов: msp_rtad[4:0] и msp_rtadp, необходимо корректно подать логические значения адреса ОУ и разряда контроля чётности.

Запуск в работу МСП в режиме ОУ осуществляется записью соответствующего значения в регистр cfg1_.

Для каждого КС (сообщения) адресованного ОУ, в ОУ МСП предусмотрен механизм самостоятельного определения допустимости командных слов (сообщений). Схема определения допустимости позволяет задавать кокое-либо допустимое подмножество из 4096 возможных комбинаций командного слова сообщения по значениям: адресов групповых/не групповых, разряда «прием/передача», разрядов «число СД/код команды».

Каждому подадресу (командного слова) на «прием»/ «групповой прием»/ «передачу» слов данных сопоставлен свой собственный буфер данных в ОУ МСП. МСП обеспечивает программно настраиваемый выбор вида управления памятью для слов данных (буферов данных) ОУ. ОУ МСП в соответствие с требованиями ГОСТ Р 52070-2003 может, но необязательно, сохранять данные принятые в групповом сообщении отдельно от данных полученных в обычных (негрупповых) сообщениях приёма. То есть, буфера данных подадресов на «прием» и «групповой прием» слов данных могут совпадать, а могут и нет. Перечень видов управления памятью для буфера/буферов данных ОУ включает в себя:

-режим «одиночной буферизации»;

-режим «двойной буферизации сообщений «приема»» для отдельных, задаваемых при конфигурировании, подадресов; Используя «двойную буферизацию сообщений «приема»», процессор УВ способен без труда получать доступ к наиболее свежим, полностью достоверным словам данных. «Двойная буферизация сообщений «приема»» используется только для подадресов на «прием» и «групповой прием» слов данных.

-режим «кольцевой буферизации», поддерживает пересылку/прием большого количества данных размером от 128 до 8192 слов;

Режим «одиночная буферизация». Буфер данных соответственного подадреса на «прием»/ «групповой прием»/ «передачу» состоит из 32-ух слов (1 блок данных) и, следовательно, соответствующий блок данных будет неоднократно перезаписываться или перечитываться при отработки ОУ МСП сообщения для этого подадреса на «прием»/ «групповой прием»/ «передачу». Режим «одиночной буферизации» разрешен при:

- разряд ermm («расширенное управление памятью ОУ», 1-ый разряд регистра cfg2) равен лог. «0», то «одиночная буферизация» применена для всех подадресов на «прием», «групповой прием», «передачу» слов данных.

- разряд ermm («расширенное управление памятью ОУ», 1-ый разряд регистра cfg2) равен лог. «1», то «одиночная буферизация» разрешается для каждого подадреса отдельно в соответствующем слове управления подадресом ОУ.

Режим «кольцевая буферизация». Буфер данных соответственного подадреса на «прием»/ «групповой прием»/ «передачу» принимает размер от 128 слов до 8192 слов. Указатель кольцевого буфера МСП может быть настроен на работу без обновления значения после принятия недостоверного блока СД сообщения. Это свойство разрешается при значении разряда oid лог. «1» («переписывать недостоверные данные» - 11-ый разряд регистра cfg2_). В этом случае, повторно пересланные КШ во время ВПС последовательности СД будут автоматически перезаписаны поверх недостоверных данных принятых во время предыдущих попыток передачи сообщения. Это способствует упрощению пересылок данных большого объёма пакетами, равными, по количеству передаваемых СД, заданному размеру области кольцевового буфера. Программе УВ достаточно, например, установить исходное положение указателя кольцевого буфера данных (в поисковой таблице) в начало области кольцевого буфера, дождаться прерывания по переполнению (‘прокручиванию’) кольцевого буфера и обслужить прерывание, извлекая из кольцевого буфера достоверные слова принятого пакета данных. В таком режиме процессор УВ, работающий с ОУ, не должен заботиться об обработке отдельных сообщений, ВПС, или ошибок. Режим «кольцевой буферизации» разрешен при:

- разряд ermm («расширенное управление памятью ОУ», 1-ый разряд регистра cfg2) равен лог. «1», то «кольцевая буферизация» разрешается для каждого подадреса отдельно в соответствующем Слове управления подадресом ОУ.

Режим «двойная буферизация сообщений приема». Режим «двойной буферизации сообщений приема» применяется только для подадресов на «прием» и «групповой прием» слов данных. Буфер данных соответственного подадреса на «прием» или «групповой прием» состоит из 64-х слов (два блока данных). Режим «двойной буферизации сообщений приема» разрешен при:

- разряд eme («разрешение расширенных возможностей конфигурирования», 15-ый разряд регистра cfg3_) установлен в лог. «1» и разряд ermm («расширенное управление памятью ОУ»,

1-ый разряд регистра cfg2) равен лог. «0» и разряда rsdbe («Разрешение глобальной двойной буферизации сообщений «приема»», 12-ый разряд регистра cfg2_) равен лог. «1», то «двойная буферизация» применена для всех подадресов на «прием», «групповой прием».

- разряд eme («разрешение расширенных возможностей конфигурирования», 15-ый разряд регистра cfg3_) установлен в лог. «1» и разряд ermm («расширенное управление памятью ОУ», 1-ый разряд регистра cfg2) равен лог. «1» и разряда rsdbe («Разрешение глобальной двойной буферизации сообщений «приема»», 12-ый разряд регистра cfg2_) равен лог. «1», то «двойная буферизация» разрешается для каждого подадреса на «прием»/ «групповой прием» отдельно в соответствующем слове управления подадресом ОУ.

Другие возможности, управляемые конфигурационным регистром cfg2_, включают автоматическое очищение (сброс) признака «Запрос на обслуживание» в регистр ответного слова ОУ после получения КУ ‘Передать векторное слово’, способность очистить/загрузить регистр счетчика метки времени после получения КУ ‘Синхронизация’/‘Синхронизация (с СД)’.

Многие достоинства ОУ МСП проявляются только в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1»).

В состоянии «расширенных возможностей конфигурирования» могут вырабатываться прерывания для отдельных, задаваемых при конфигурировании КУ; раздельные, фиксированные местоположения в памяти могут использоваться для размещения слов данных КУ «приёма» и «передачи» (задаваемых при конфигурировании).

МСП также обеспечивает комбинированный режим «ОУ/Монитор сообщений». В этом режиме работы доступны все свойства ОУ и свойства полноценного «монитора сообщений». Режим «монитор сообщений» позволяет пользователю избирательно отслеживать сообщения, реализуя свой выбор на основе поля КС «Адрес ОУ», разряда (разряд «Приём-передача»), и поля КС «Подадрес/Режим управления». «Монитор сообщений» имеет отдельный от ОУ стек команд и кольцевой буфер данных.

1.3.12.2.2.6.2 Организация памяти ОУ

Типичное распределение памяти ОУ МСП показано в таблице

Таблица 1.. Два указателя на стек команд постоянно присутствуют в ОЗУ по адресам: 0100 (шестн.) - для указателя на стек команд области А; 0104 (шестн.) - для указателя на стек команд области Б. В дополнение к указателям на стек команд, для данного распределения памяти ОУ, заданы и другие области памяти, имеющие неперемещаемые (фиксированные) адреса – поисковые таблицы для подадресов ОУ, для областей А и Б.

Поисковые таблицы для подадресов ОУ расположены в зоне адресов с 0140 по 01BF для области А и в зоне адресов с 01C0 по 023F для области Б. Поисковые таблицы для подадресов ОУ обеспечивают распределение буферов данных для конкретных (всех) подадресов на «прием», «групповой прием», «передачу» слов данных. Поисковые таблицы для подадресов ОУ содержат: слово управления подадресом ОУ для каждого подадреса, указатели на буфера данных для каждого подадреса «приема», «группового приема», «передачи» слов данных.

Области стеков команд и буфера данных (блоки данных) сообщений могут быть размещены где угодно в пространстве разделяемой памяти, вне (без пересечения адресов) фиксированных зон.

В таблице 1.639 показано типичное распределение памяти для состояния «нерасширенные возможности конфигурирования» ОУ МСП. В этом случае местоположения с адресами 0280-3FF (шестн.), могут быть употреблены для расположения в них стека команд или блоков для размещения СД сообщений (буферов данных).

Таблица 1.639 - Типичное распределение памяти ОУ (режим «нерасширенные возможности конфигурирования»

Адрес	Описание
0000-00FF	Стек команд области А
0100	Указатель на стек команд области А (фиксированная позиция)
0104	Указатель на стек команд области Б (фиксированная позиция)
0140-01BF	Поисковая таблица для подадресов ОУ области А (фиксированная область)
01C0-023F	Поисковая таблица для подадресов ОУ области Б (фиксированная область)
0240-025F	Блок данных 0
0260-027F	Блок данных 1
0280-029F	Блок данных 2
02A0-02BF	Блок данных 3
02C0-02DF	Блок данных 4
02E0-02FF	Блок данных 5
0300-03FF	Таблица допустимости КС (фиксированная область)
0400-041F	Блок данных 6
	.....
0EE0-0EFF	Блок данных 93
0F00-0FFF	Стек команд области Б

В таблице 1.640 показано распределение памяти для случая, когда используется режим «расширенные возможности конфигурирования».

Таблица 1.640 - Типичное распределение памяти ОУ (режим «расширенные возможности конфигурирования

Адрес	Описание
0000-00FF	Стек команд области А
0100	Указатель на стек команд области А (фиксированное положение)
0101-0103	Резерв
0104	Указатель на стек команд области Б (фиксированное положение)
0105-0107	Резерв
0108-010F	Таблица прерываний для избранных КУ (фиксированная область)
0110-013F	Таблица слов данных КУ (фиксированная область)
0140-01BF	Поисковая таблица для подадресов ОУ области А (фиксированная область)
01C0-023F	Поисковая таблица для подадресов ОУ области Б (фиксированная область)
0240-0247	Таблица занятости подадресов (фиксированная область)
0248-025F	Не используется
0260-027F	Блок данных 0
0280-029F	Блок данных 1
	.....
02E0-02FF	Блок данных 4
0300-03FF	Таблица допустимости КС (фиксированная область)
0400-041F	Блок данных 5
0420-043F	Блок данных 6
	.....
0FE0-0FFF	Блок данных 100

Обратите внимание, что распределение памяти, показанное для режима «расширенные возможности конфигурирования» ОУ в таблице 1.640, не содержит стека команд для области Б. Таблица 1.640 преследует чисто пояснительные цели.

В случае использования определения допустимости поступающих командных слов, адреса с 0300 по 03FF определяют зону (область) для размещения таблицы допустимости КС.

В режиме «расширенные возможности конфигурирования» ОУ строго рекомендуется использовать двойную буферизацию для сообщений «приема» для подадреса, а не глобальную двойную буферизацию с использованием областей А/Б, как средство, более надежно, эффективно обеспечивающее целостность данных в обрабатываемых сообщениях. Обратите внимание, что в таблице 1.641 область, предназначенная для стека команд, занимает 256 ячеек. Тем не менее, в режиме «расширенные возможности конфигурирования» разрешено устанавливать большие размеры для области стека. Размеры стека программно выбираемы из ряда следующих величин 256 слов (64 сообщений), 512, 1024 или 2048 (512 сообщений) посредством разряда 14 и 13 регистра cfg3_(см. таблицу 1.641).

Таблица 1.641 - Размер стека команд ОУ

14 разряд BC/RT CMD Размер стека 1	13 разряд BC/RT CMD Размер стека 2	Размер стека КШ/ОУ (в словах)
0	0	256 (64)
0	1	512 (128)
1	0	1024 (256)
1	1	2048 (512)

1.3.12.2.2.6.3 Управление памятью в режиме ОУ

Для выполнения требований ГОСТ Р 52070-2003 и лучшего обеспечения пересылки информационных массивов большого размера в МСП введены широкие возможности управления памятью ОУ. Эти свойства позволяют разделять данные групповых сообщений от данных негрупповых сообщений (разряд sbd регистра cfg2_), помогают обеспечивать целостность принимаемых блоков данных сообщений (например, использованием двойной буферизации для сообщений «приема»), а в целом позволяют разгрузить, высвободить процессор УВ для возможности более успешного решения им задач реального времени.

Задание значений разрядов регистра cfg2_ позволяют использовать следующие возможности управления памятью ОУ:

- использование таблицы занятости подадресов (разряд blute[13]);
- разрешение режима «глобальной двойной буферизация для сообщений «приема»» (разряд rsdbe[12]);
- переписывать недостоверные данные (разряд oid[11]);
- «расширенное управление памятью ОУ» (разряд ermm[1]);
- отделение данных групповых сообщений «приема» (разряд sbd[0]).

МСП обеспечивает механизм всеобщей «глобальной двойной буферизации» (не путать с «глобальной двойной буферизация для сообщений «приема») посредством 13-ого разряда регистра cfg1_ (ca_b_a). В любой момент времени один указатель на стек команд, область стека команд, поисковая таблица для подадресов ОУ назначены (определены) как активные (область А/ область Б), т.е. используемые МСП для обработки сообщений, и другое подмножество соответствующих структур данных как неактивное(область Б/ область А). Обе области памяти всегда доступны управляющему процессору.

Таблица 1.показывает структуру поисковой таблицы для подадресов ОУ. Для каждого подадреса ОУ назначен буфер данных на «прием»/«групповой приём» СД, буфер данных на «передачу» СД, буфер данных на «групповой приём» СД (необязательно) и слово управления подадресом ОУ (необязательно). Поисковая таблица для подадресов ОУ содержит указатели на эти буфера данных и слова управления подадресами ОУ. Поисковая таблица для подадресов ОУ состоит из 4 частей:

1) Таблица указателей для подадресов на «прием»/«групповой приём»;
2) Таблица указателей для подадресов на «передачу»;

3) Таблица указателей для подадресов на «групповой приём». Таблица необязательная (cfg2_, разряд sbd[0]);

4) Таблица слов управления подадресом ОУ. Таблица необязательная (cfg2_, разряд ermm[1]).

Таблица 1.642 - Поисковая таблица для подадресов ОУ

Область А	Область Б	Описание	Комментарий
0140 ... 015F	01C0 ... 01DF	Rx(/Bcst) SA0 ... Rx(/Bcst) SA31	Таблица указателей для подадресов на «прием»/«групповой приём»
0160 ... 017F	01E0 ... 01FF	Tx SA0 ... Tx SA31	Таблица указателей для поадресов на «передачу»
0180 ... 019F	0200 ... 021F	Bcst SA0 ... Bcst SA31	Таблица указателей для подадресов на «групповой приём» (необязательная)
01A0 ... 01BF	0220 ... 023F	SACW SA0 ... SACW SA31	Таблица слов управления подадресом ОУ (необязательная)

1.3.12.2.2.6.3.1 Слово управления подадресом ОУ

В режиме «расширенное управление памятью ОУ» (разряд ermm регистра cfg2__) каждое из 32 слов управления подадресом ОУ точно и однозначно описывает схему управления памятью для соответствующего подадреса. В каждом слове управления подадресом ОУ для каждого типа сообщений – «приёма», «группового приёма», и «передачи» - имеются отдельные пять разрядов, которые отвечают за выбор схемы управления памятью для слов данных и разрешение прерываний. В дополнение к этому, разряд 15 (СЗР) может быть использован для разрешения режима «двойной буферизации сообщений «приема»» для подадресов КС «приём» (и/или групповых КС «приём»). (см. таблицы Таблица 1. и Таблица 1.).

В каждом слове управления подадресом ОУ для каждого «передающего», «принимающего» или «группового» буфера данных подадреса имеются три разряда, с помощью которых можно задать желаемую схему управления памятью буфера данных. Для каждого подадреса для сообщений «приёма»/«группового приема» схема управления памятью может быть выбрана среди режима «одиночная буферизация», режима «двойной буферизации сообщений «приема»» и режима «кольцевой буферизации». Для каждого подадреса сообщений «передачи» схема управления памятью может быть выбрана среди режима «одиночная буферизация» и режима «кольцевая буферизация». Заметим, что режим «двойной буферизации сообщений «приема»» не применим для подадресов сообщений «передача».

Следует отметить, что для сообщений, в которых используются КУ, слово управления подадресом ОУ применяется, только если «расширенная обработка КУ» запрещена. «Расширенная обработка КУ» становится запрещенной в режиме нерасширенных возможностей МСП (разряд eme регистра cfg3_ установлен в лог. «0») или, если emch, 0-ой разряд регистра cfg3_ - «расширенная обработка КУ», установлен в лог. «0».

«Расширенная обработка КУ» разрешается в режиме расширенных возможностей конфигурирования МСП, когда 0-ой разряд регистра cfg3_ установлен в лог. «1».

В режиме «одиночной буферизации» один буфер/блок данных постоянно, снова и снова прочитывается (для передаваемых данных) или снова и снова записывается (для принимаемых или групповых данных). Совершенно по-другому обстоит дело в режиме «кольцевая буферизация». Слова данных для последовательных сообщений в/из какого-либо отдельного

подадреса читаются или записываются в прилегающее (смежное) местоположение блока соответственного кольцевого буфера.

Размер кольцевого буфера для каждого подадреса в КС «приём», «передача» или «групповое» может быть настроен на одну из следующих величин: 128, 256, 512, 1024, 2048, 4096 или 8192 слов. Для каждого подадреса (для сообщений «приёма», «передачи», «групповых») два разряда слова управления подадресом ОУ (задействованы) для разрешения прерываний. Один из этих разрядов будет приводить к выработке прерывания после каждой пересылки сообщения, направленного к подадресу. Другой из этих двух разрядов будет приводить к выработке запроса на обработку прерывания в конце каждого обработанного сообщения в случае, если указатель кольцевого буфера для соответствующего подадреса ‘прокручивается’ через всю область пространства буфера (достигает максимальную по значению адреса границу и перескакивает в начало) кольцевого буфера.

В режиме «двойной буферизации сообщений «приема»» для подадреса пара 32-словных блоков данных задействуется для отдельного подадреса команды «приём» или «принять групповое сообщение». В этом режиме слова данных следующих одно за другим сообщений, приходящие по одному и тому же подадресу, сохраняются попеременно то в одном, то в другом блоке.

Таблица 1.643 - Слово управления подадресом ОУ rt_cw[15..0] (адрес 00100; чт/зп)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rx_dbe	tx_eom_i	tx_cbi	tx_mm[2..0]			rx_eom_i	rx_cbi	rx_mm[2..0]			bcst_eom_i	bcst_cbi_i	bcst_mm[2..0]		

rx_dbe(RX: Double Buffer Enable): разрешение двойной буферизации для сообщений «приёма»;

tx_eom_i (TX: EOM Int): разрешение прерывания по завершению сообщений «передачи»;

tx_cbi (TX: Circ Buf Int): разрешение прерывания по переполнению кольцевого буфера для сообщений «передачи;

tx_mm[2..0] (TX: Memory Management): способ управление памятью для сообщений «передачи»;

rx_eom_i (RX: EOM Int): разрешение прерывания по завершению сообщений «приёма»;

rx_cbi (RX: Circ Buf Int): разрешение прерывания по переполнению кольцевого буфера для сообщений «приёма»;

rx_mm[2..0] (RX: Memory Management): способ управление памятью для сообщений «приёма»;

bcst_eom_i (BCST: EOM Int): разрешение прерывания по завершению сообщений «группового приёма»;

bcst_cbi_i (BCST: Circ Buf Int): разрешение прерывания по переполнению кольцевого буфера для сообщений «группового приёма»;

bcst_mm[2..0] (BCST: Memory Management): способ управления памятью для сообщений «группового приёма».

Таблица 1.644 - Схема организации памяти для буфера СД подадреса

rx_dbe (см. замечание)	MM 2	MM 1	MM 0	Описание	Комментарии
0	0	0	0	Единственное сообщение	
1	0	0	0	Для приема или группового приёма: двойная буферизация. Для передачи: одиночная буферизация.	
X	0	0	1	128 слов	Кольцевой буфер определённого размера
X	0	1	0	256 слов	
X	0	1	1	512 слов	
X	1	0	0	1024 слов	
X	1	0	1	2048 слов	
X	1	1	0	4096 слов	
X	1	1	1	8192 слов	

Замечание - Для того, чтобы использовать режим «двойной буферизации сообщений «приема»» для подадреса, МСП должен быть запрограммирован (переведен) в режим «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»).

Для того, чтобы выбрать режим «двойной буферизации сообщений «приема»» для подадреса средствами слова управления подадресом ОУ, разряд «разрешение глобальной двойной буферизации для сообщений «приема»» (rsdbe, 12-ый разряд регистра cfg2_) и разряд «расширенное управление памятью ОУ» (ermm, 1-ый разряд регистра cfg2_) должны быть установлены в лог. «1» .

Если МСП находится в состоянии «расширенные возможностей конфигурирования» и разряд «разрешение глобальной двойной буферизации для сообщений «приема»» (rsdbe, 12-ый разряд регистра cfg2_) установлен в лог. «1» и «расширенное управление памятью ОУ» установлен в лог. «0», то режим «двойной буферизации сообщений «приема»» будет употреблен для всех сообщений/подадресов на «приём»/«приёма группового».

1.3.12.2.2.6.3.2 Режим одиночной буферизация

Если ermm, 1-ый разряд регистра cfg2_, находится в состоянии логического «0» , то предполагается, что схема управления памятью МСП находится в том состоянии, в какое она попадает после включения питания «по умолчанию», то есть «нерасширенные возможности конфигурирования», нерасширенное управление памятью. В обычном режиме операций ОУ протокол управления памятью «одиночная буферизация» применяется для всех подадресов команд «приём», «передача» или команд «группового приёма». Отметим, что в случае «расширенного управления памятью ОУ» (ermm, 1-ый разряд регистра cfg2_), режим «одиночной буферизации» может также быть выбранным для индивидуального подадреса команд «приём», «передача» или команд «группового приёма». Это происходит в том случае, если разряд «разрешение двойной буферизации для сообщений «приёма»» (разряд 15) и три используемых в текущий момент разряда «Управления памятью» - MM[2..0] - соответствующего слова управления подадресом ОУ, все сразу, находятся в лог. «0» . Работа ОУ в режиме «одиночной буферизации» показана на рисунке 1.208.

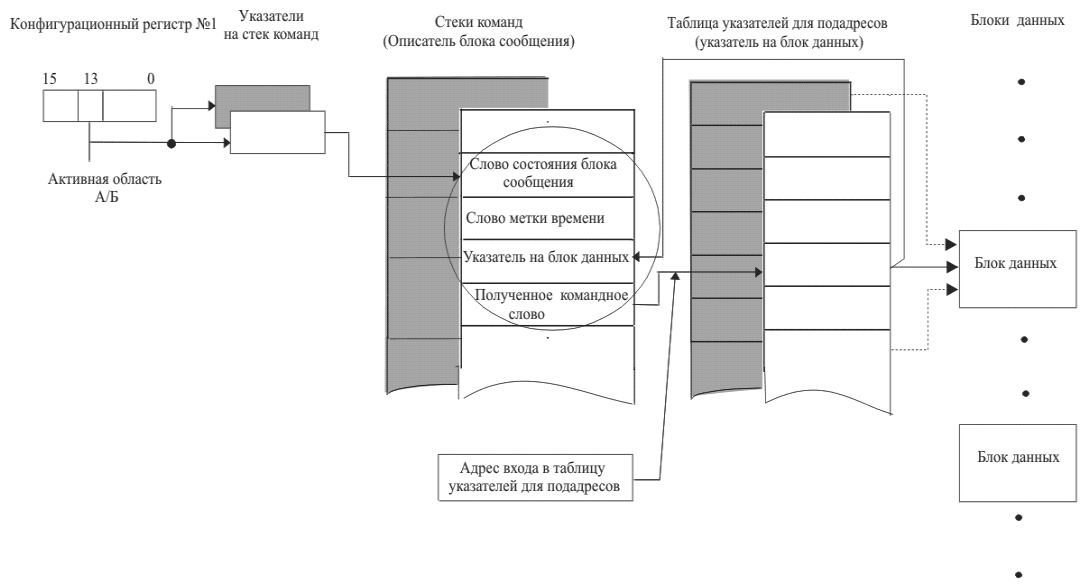


Рисунок 1.208 - Управление памятью ОУ: Режим «одиночной буферизации»

Замечание. Таблица указателей для подадресов не используется для КУ, когда разрешена расширенная обработка КУ

В режиме «одиночной буферизации» определяемая подадресом командного слова ячейка поисковой таблицы для подадресов ОУ должна быть заблаговременно сформирована со стороны УВ записью значения указателя на буфер данных (на начало блока данных) (в процессе конфигурирования). В момент старта обработки каждого сообщения содержимое данной ячейки загружается, автоматически, в третью позицию соответствующего описателя блока сообщений в области командного стека ОЗУ. Принимаемые слова данных записываются в или передаваемые слова данных читаются из блока слов данных, на который ссылается этот указатель. В режиме «одиночной буферизации» текущий указатель поисковой таблицы для подадресов ОУ не становится объектом для записи со стороны логики управления памятью МСП и, следовательно, не изменяет своего значения в конце обработки сообщения. Поэтому, если последующее сообщение является обращенным к тому же подадресу, то тот же самый буфер/блок словных данных будет перезаписываться или же перечитываться.

1.3.12.2.2.6.3.3 256-словные границы

В конфигурации «одиночной буферизации» разряд «запрещение 256-словных ограничений» (256_wbd, 10-ый разряд регистра cfg2_) должен приниматься во внимание для обоих режимов управления памятью: устанавливаемого по умолчанию, когда ermm, 1-ый разряд регистра cfg2_ находится в лог. «0»; и в случае режима «расширенное управление памятью ОУ» (ermm, 1-ый разряд регистра cfg2_, установлен в лог. «1»), когда режим «одиночной буферизации» назначается для каких-либо (передачи/приёма/групповой) подадресов.

Настоятельно рекомендуется устанавливать разряд 256_wbd («запрещение 256-словных ограничений») в лог. «0» для режима ОУ.

Если разряд «запрещение 256-словных ограничений» установлен в лог. «0» и МСП настроен на работу в режиме «одиночной буферизации», то в адресном пространстве разделяемой (между МСП и УВ) памяти будут соблюдаться 256-словные ограничения. То есть, если разряд «запрещение 256-словных ограничений» установлен в лог. «0», то указатель на буфер данных будет ‘прокручиваться’, переходя из XXFF в XX00, а не увеличиваться до (XX + 1)00.

Как бы то ни было, если «запрещение 256-словных ограничений» установлен в лог. «1» (позволяет, в виде исключения, для конфигурации «одиночной буферизации» экономно распределять ОЗУ), счетчик адресов МСП не будет придерживаться 256-словных ограничений. В этом случае адреса слов данных задаются не 8-ми разрядным, а 16-ти разрядным счетчиком. То есть счётчик адреса будет увеличиваться, переходя от XXFF к (XX + 1)00, а не откатываться назад к XX00.

1.3.12.2.2.6.3.4 Режим «двойной буферизации сообщений «приема»»

ОУ МСП обеспечивает механизм «двойной буферизации сообщений «приема»» для принимаемых данных, который выборочно устанавливается на подадресной основе для любых подадресов команд «приём» и/или команд группового «приёма» Следует отметить, что режим «двойной буферизации сообщений «приема»» применим лишь для принимаемых данных, не для передаваемых.

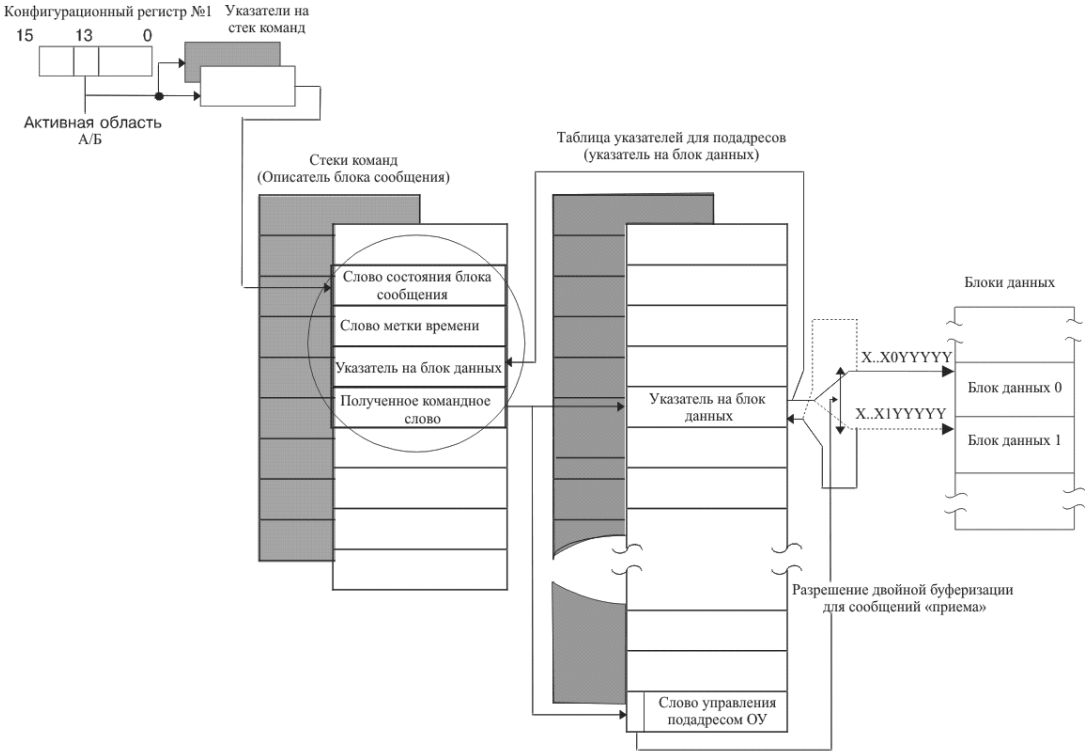


Рисунок 1.209 - Управление памятью ОУ: режим «двойной буферизации сообщений «приема»

Назначение режима «двойной буферизации сообщений «приема»» - обеспечение высокой степени целостности данных. Это достигается введением использования двух 32-словных блоков данных для каждого отдельного (индивидуального) подадреса ОУ на «прием» и/или «групповой приём» слов данных. Один из двух блоков назначается активным – соответствует текущему значению указателя на буфер данных (блок данных) в таблице указателей для подадресов на «прием»/«групповой приём», в то время как другой – пассивным. Данные, которые поступят со следующей командой «приём», обращенной к тому же самому подадресу, будут сохраняться в активном блоке. При завершении обработки сообщения, при условии, что сообщение оказалось достоверным и двойная буферизация для «приема» разрешена, МСП автоматически переключит, поменяв местами, активный и пассивный блоки для этого подадреса. Это означает, что самый последний, достоверный, завершенный в обработке блок всегда без задержки поступает в распоряжение УВ.

«Двойная буферизация сообщений «приема»» возможна только в режиме «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1»). Если разряд «расширенное управление памятью ОУ» (ermm, 1-ый разряд регистра cfg2_) установлен в лог. «0», то избрание режима «Двойная буферизация сообщений «приема»» для всех подадресов «приёма» становится возможным простым установлением в лог. «1» разряда rsdbe (12-ый разряд регистра cfg2_). В этом случае «двойная буферизация сообщений «приема»» разрешена для всех подадресов на «приём»/«групповой прием» слов данных (слова данных «группового приема» могут находиться отдельно от слов данных «приёма», если разряд sbd[0] равен лог. «1»).

Если разряд «расширенное управление памятью ОУ» установлен в лог. «1», то режим «двойная буферизация сообщений «приема»» разрешается для «выборочных» подадресов переводом в состояние лог. «1» разряда rsdbe (12-ый разряд регистра cfg2_) и программированием желаемого слова управления подадресом ОУ. Слово управления подадресом ОУ должно быть запрограммировано следующим образом: в поле rx_mm[2..0], разряды 7-5, установлены в лог. «0» и в разряд rx_dbe (разряд 15) должна быть записана лог. «1». Если «отделение данных групповых сообщений» разрешено (sbd, 0-ой разряд регистра cfg2_ установлен в 1), то в соответствующем слове управления подадресом ОУ для групповых команд в поле bcst_mm[2..0] разряды 2÷0 программируются в лог. «0» (rx_dbe в лог. «1»), если «двойная буферизация сообщений «приема»» желательна.

Программный алгоритм чтения самого последнего (свежего) блока данных таков. Сначала, программно со стороны УВ, запрещается «двойная буферизация сообщений «приема»» для желаемого подадреса. Если режим «расширенное управление памятью ОУ» не разрешён (ermm, 1-ый разряд регистра cfg2_ установлен в лог. «0»), «двойная буферизация сообщений «приема»» должна быть запрещена для всех подадресов путем установления разряда «разрешение глобальной двойной буферизации для сообщений «приема»» (rsdbe, 12-ый разряд регистра cfg2_) в лог. «0». Если «расширенное управление памятью ОУ» разрешено (настоятельно рекомендуется использовать!), «двойная буферизация сообщений «приема»» может запрещаться на подадресной основе, записью лог. «0» в разряд «разрешение двойной буферизации для сообщений «приема»» (разряд 15) принадлежащего этому подадресу, соответственного слова управления подадресом ОУ. Процессору УВ следует затем прочитать текущее значение указателя на блок данных, расположенного в таблице указателей для подадресов на «прием»/«групповой приём». Этот указатель ссылается на активный блок словных данных. Последний блок последовательных данных может быть прочитан из пассивного блока данных сообщения, т.е. по адресу на который ссылается текущий указатель на блок данных с инвертированным значением 5-го разряда в нём. Текущее содержимое самой таблицы указателей для подадресов на «прием»/«групповой приём» при этом не следует подвергать изменениям.

После прочтения блока данных процессору УВ необходимо вновь разрешить «двойную буферизацию сообщений «приема»» установлением в лог. «1» либо разряда rsdbe, 12-ый разряд регистра cfg2_, (если «расширенное управление памятью ОУ» не разрешён), либо разряда rx_dbe, разряд 15, в принадлежащем подадресу, соответственном слове управления подадресом ОУ (для случая, если «расширенное управление памятью ОУ» разрешено).

1.3.12.2.2.6.3.5 Режим «кольцевой буферизации»

В режиме «расширенное управление памятью ОУ» отдельный какой-либо подадрес ОУ на «приём», «групповой приём» и «передачу» слов данных может быть запрограммирован в режим «одиночной буферизации», и в режим «кольцевой буферизации».

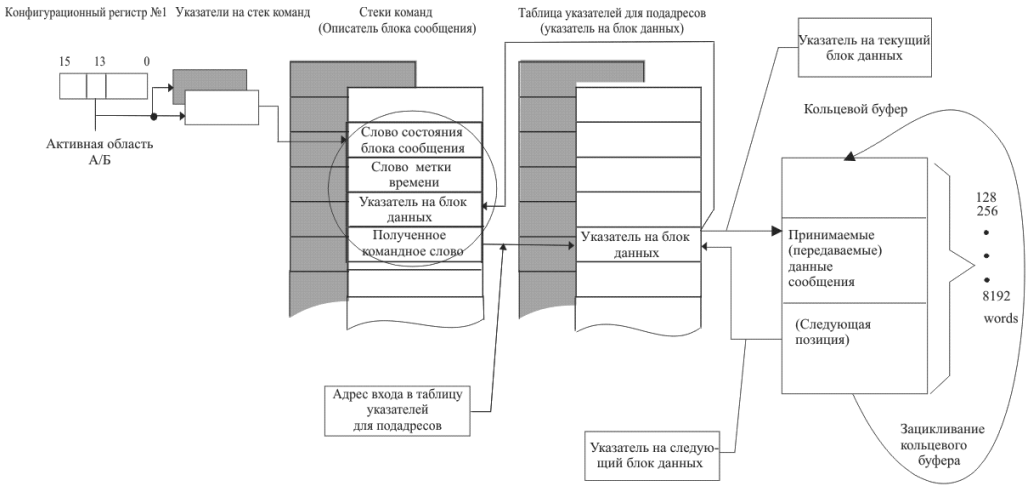


Рисунок 1.176 - Управление памятью ОУ: Режим «кольцевая буферизация»

Работа в режиме «кольцевой буферизации» управления памятью подадреса ОУ объясняется при помощи рисунка 1.210. Как и в режиме «одиночной буферизации», перед началом работы режима «кольцевой буферизации» УВ записывает (инициализирует) в индивидуальную ячейку таблицы указателей для подадресов на «прием»/«групповой приём»/«передачу» значение указателя на начало кольцевого буфера. При запуске обработки каждого сообщения указанный элемент таблицы указателей для подадресов на «прием»/«групповой приём»/«передачу» активной области ОЗУ автоматически загружается в третью позицию описателя блока сообщения в области стека команд в ОЗУ. Принимаемые или передаваемые слова данных перемещаются в/из кольцевого буфера, начиная с местоположения, на которое ссылается слово (указатель), взятое из таблицы указателей для подадресов на «прием»/«групповой приём»/«передачу». Если разряд «переписывать недостоверные данные» (oid, 11-ый разряд регистра cfg2_) установлен в лог. «0», адрес местоположения, находящегося после последней, подвергнутой чтению, либо записи позиции в блоке данных сообщения, становится загруженным в соответствующую ячейку таблицы указателей для подадресов на «прием»/«групповой приём»/«передачу» вне зависимости от того, произошла ли какая-либо ошибка в только что выполненном (обработанном) сообщении или нет. Таким образом, данные для следующего сообщения, направляемого к тому же самому подадресу (передачи/приёма/групповой), будут передаваться в/из следующий, прилегающий по местоположению к текущему, соседний блок данных кольцевого буфера.

Если разряд «переписывать недостоверные данные» (oid, 11-ый разряд регистра cfg2_) установлен в лог. «1», адрес местоположения, находящегося после последней, подвергнутой чтению, либо записи позиции в блоке данных сообщения, становится загруженным в соответствующую ячейку таблицы указателей для подадресов на «прием»/«групповой приём»/«передачу» только после принятия (или передачи) достоверного сообщения. Предполагая, что разряд «переписывать недостоверные данные» установлен в лог. «1», слово(указатель) из таблицы указателей для подадресов на «прием»/«групповой приём»/«передачу» не будет подвергаться обновлению в конце сообщения, если последнее обработано с ошибкой. Это позволяет при пересылке данных большого объема автоматически исправлять неудавшиеся сообщения (оказавшиеся недостоверными при пересылке) посредством ВПС, не нарушая структуры данных кольцевого буфера, без вмешательства процессора УВ.

Настоятельно рекомендуется, чтобы при использовании режима «кольцевой буферизации» разряд «переписывать недостоверные данные» был установлен в лог. «1».

Расположение выбранного кольцевого буфера в ОЗУ зависит от заданного размера буфера и значения загружаемого УВ указателя на начальную позицию блока сообщения - словных данных, и определяется по следующим правилам: адреса минимальных по значению адреса границ кольцевых буферов данных кратны заданному размеру буфера (128, 256, …, 8192 слов. Примеры определений значений адресов границ кольцевых буферов данных (двоичных) в зависимости от размера буфера («x» – значения соответствующих старших разрядов указателя на начальную позицию блока сообщения - словных данных):

- для буфера размером 128 слов адрес начала определяется как (16'bxxxx_xxxx_x000_0000), адрес завершения – (16'bxxxx_xxxx_x111_1111),

- для буфера размером 256 слов адрес начала определяется как (16'bxxxx_xxxx_0000_0000), адрес завершения – (16'bxxxx_xxxx_1111_1111),

…

- для буфера размером 8192 слов адрес начала определяется как (16'bxxx0_0000_0000_0000), адрес завершения – (16'bxxx1_1111_1111_1111).

Значение загружаемого УВ указателя на начальную позицию блока сообщения (словных данных) может совпадать с адресом начала выбранного буфера, либо содержать любой адрес в пределах границ данного буфера. В процессе обработки сообщений ОУ (по приёму или передаче каждого очередного слова данных) инкрементируются только младшие разряды данного указателя, количество которых определяется размером кольцевого буфера (7 младших разрядов для буфера в 128 слов, 8 младших разрядов для буфера в 256 слов, и т.д.). Когда указателем блока сообщения достигается завершающий адрес кольцевого буфера, его значение «прокручивается», возвращаясь к началу кольцевого буфера, как это показано на рисунке 1.210.

Для лучшего, более полного использования свойств, преимуществ режима «кольцевой буферизации», разряд «переписывать недостоверные данные» (oid, 11-ый разряд регистра cfg2_) и разряд «разрешение прерывания по переполнению кольцевого буфера для сообщений «приёма»/» (rx_cbi, разряд 8 в слове управления подадресом ОУ) должны быть запрограммированы в лог. «1». УВ необходимо инициализировать начальное значения указателя кольцевого буфера, задать его размер и затем ждать прерывания для события переполнения ('прокручивания') кольцевого буфера. Затем, после того, как это прерывание происходит, УВ может читать из ОЗУ очередной, состоящий из СД множества сообщений, блок достоверных, принятых слов данных. Использование свойства «кольцевой буферизации» подадреса ОУ исключает необходимость для процессора обслуживать индивидуальные сообщения или иметь дело с разбором ошибок сообщений или с возобновлением передачи сообщений (ВПС).

Таблица 1.645 - Разряды, влияющие на видоизменение свойств, режимов управления памятью подадресов ОУ

Свойства	«Разрешение глобальной двойной буферизации для сообщений «приёма»» (rsdbe, 12-ый разряд регистра cfg2_)	«переписы-вать недостовер-ные данные» (oid, 11-ый разряд регистра cfg2_)	«Расширен-ное управле-ние памятью ОУ» (ermm, 1-ый разряд регистра cfg2_)	«Отделение данных групповых сообщений» (sbd, 0-ой разряд регистра cfg2_)	«Разрешение расширенных возможно-стей конфигури-рования» (eme, 15-ый разряд регистра cfg3_)
Режим по умолчанию (по включению питания) - режим «одиночной буферизации» для всех подадресов (передачи/приёма/группово-го приема)	0	0	0	0	0

Продолжение таблицы 1.645

Свойства	«Разрешение глобальной двойной буферизации для сообщений «приёма»» (rsdbe, 12-ый разряд регистра cfg2_)	«переписывать недостоверные данные» (oid, 11-ый разряд регистра cfg2_)	«Расширенное управление памятью ОУ» (ermm, 1-ый разряд регистра cfg2_)	«Отделение данных групповых сообщений» (sbd, 0-ой разряд регистра cfg2_)	«Разрешение расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_)
Режим «одиночной буферизации» для всех передаваемых подадресов; «двойная буферизация для сообщений «приема»» для всех принимающих (и групповых) подадресов	1 (см. замечание)	0	0	X	1 (см. замечание)
Отделение данных групповых сообщений	X	X	X	1	X
Возможна кольцевая буферизация для отдельных подадресов; недостоверная информация НЕ перезаписывается	X	0	1	X	X
Возможна кольцевая буферизация для отдельных подадресов; с перезаписыванием недостоверной информации	X	1	1	X	X
Возможна двойная буферизация для отдельных принимающих (и групповых) подадресов	1 (см. замечание)	X	1	X	1 (см. замечание)

Замечание - Для возможности режима «двойной буферизации сообщений «приема»» подадреса для принимаемых СД разряд «разрешение расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_) должен быть запрограммирован в лог. «1» перед записью лог. «1» в разряд «разрешение глобальной двойной буферизации для сообщений «приёма»»» (rsdbe, 12-ый разряд регистра cfg2_).

1.3.12.2.2.6.4 Описатель блока сообщения ОУ

В режиме ОУ область стека команд ОЗУ содержит хронологию реального времени всех сообщений, обрабатываемых МСП. Для каждого обрабатываемого сообщения в стеке имеется (содержится) 4-х словный описатель блока сообщения. Исключение: если групповые сообщения разрешены, и ОУ МСП получает сообщение типа ОУ-ОУ, формат 8, где второе командное слово на передачу данных адресовано собственному адресу ОУ МСП, то в стеке сообщений содержится два 4-х словных описателя блока сообщения. Первый описатель блока сообщений относиться к первому вытесненному групповому командному слову. Второй описатель блока сообщений относиться ко второму командному слову на передачу данных.

Четыре элемента каждого описателя блока сообщения – слово состояния блока сообщения, слово метки времени, указатель блока данных (указатель начальной позиции расположения слов данных сообщения/ указатель на буфер данных подадреса) и 16-ти разрядное полученное (принятое по ЛПИ) «командное слово сообщения». Перед началом обработки сообщений УВ надлежит должным образом заполнить (инициализировать) указатель на стек команд. В

некоторых применениях может оказаться полезным «обнуление» всей области стека перед получением сообщений.

В режиме ОУ УВ может использовать один из способов определения того, что обработка сообщения закончилась, а именно либо процессор опрашивает регистр isr (регистр состояния прерывания), либо местоположение памяти указателя на стек команд активной области (А/Б). Указатель на стек команд ОУ увеличивается на четыре (по модулю размера стекового) при старте обработки каждого сообщения. После того, как сообщение отработано, УВ следует прочитать слово состояния блока сообщения, слово метки времени, указатель блока данных и полученное «командное слово сообщения» из описателя блока сообщения. При предположении достоверности полученного сообщения, принятые слова данных считываются из соответствующего блока сообщения (буфера данных подадреса).

1.3.12.2.2.6.4.1 Слово состояния блока сообщения ОУ

Ниже приводятся разрядные распределение и описания для слова состояния блока сообщения ОУ.

Таблица 1.646 - Структура слова состояния блока сообщения ОУ:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
eom	som	ch_a_b	ef	rt_rt_f	fe	rto	ltf	cbr	icw	wce	ids	iw	rtg_s_ae	rt_2cwe	cwce

eom (END-OF-MESSAGE), завершение отработки сообщения;
som (START-OF-MESSAGE), старт отработки сообщения;
ch_a_b (channel b/a*), канал обмена B/A;
ef (Error flag), признак ошибки;
rt_rt_f (RT-to-RT FORMAT), принимающий ОУ в пересылке типа ОУ-ОУ;
fe (Format Error), ошибочный формат сообщения;
rto (Response Time out), истечение времени ожидания ответного слова ОУ;
ltf (Loop Test Fail), ошибка проверочного возврата;
cbr (Circular Buffer Rollover), ‘прокручивание’ кольцевого буфера;
icw (Illegal Command Word), признак недопустимости командного слова;
wce (Word Count Error), ошибка количества слов в сообщении;
ids (Incorrect Data Sync), некорректный тип синхросигнала в слове данных;
iw (Invalid Word), признак недостоверности слова;
rtg_s_ae (RT-RT Gap/Sync/Address Error), ошибка в передаче ОУ-ОУ»: интервала ожидания ОС / синхросигнала или формата / адрес ОУ;
rt_2cwe (RT-RT 2nd Command Word Error), индикатор ошибки 2-ого командного слова;
cwce (Command Word Contents Error), принятое КС не согласуется с требованиями ГОСТ Р 52070-2003.

eom (END-OF-MESSAGE), – разряд завершение отработки сообщения: Устанавливается по завершении сообщения вне зависимости от наличия ошибки.

som (START-OF-MESSAGE), – разряд старт отработки сообщения: Устанавливается приблизительно через 3-4 мкс после прохождения середины разряда контроля по четности командного слова и очищается в конце сообщения.

ch_a_b (channel b/a*), – разряд канал обмена B/A: Этот разряд будет установлен в лог. «0» , если сообщение передавалось (было передано) по каналу А, или лог. «1» - если была определена пересылка по каналу В.

ef (Error flag), – разряд признак ошибки:

Если этот разряд установлен в лог. «1» и один или более из разрядов 10, 9 и/или 8 установлены в лог. «1», это указывает на наличие одной или более из следующих ошибок,

произошедших при пересылке сообщения: ошибочный формат сообщения, истечение интервала ожидания ОС и/или ошибка проверочного возврата.

rt_rt_f (RT-to-RT FORMAT), – разряд принимающий ОУ в пересылке типа ОУ-ОУ: Применим только для режима «расширенных возможностей конфигурирования», то есть когда eme, 15-ый разряд регистра cfg3_, установлен в лог. «1».

Этот разряд будет устанавливаться в лог. «1», если МСП является принимающим ОУ при пересылке типа «ОУ-ОУ».

Если групповые сообщения разрешены, и ОУ МСП получает сообщение типа ОУ-ОУ, формат 8, где второе командное слово на передачу данных адресовано собственному адресу ОУ МСП, то первое групповое командное слово будет вытеснено вторым командным словом на передачу данных и разряд rt_rt_f не установиться в лог. «1».

fe (Format Error), – разряд ошибочный формат сообщения: Если установлен в лог. «1», то указывает, что принятая порция сообщения содержит одно или более нарушение критериев достоверности сообщения (типа синхроимпульса, кодирование, четность, количество разрядов, количество слов и т.д.). Значение этого разряда определяется логической суммой (дизъюнкцией) разрядов 5, 4, 3, 2, 1 и 0.

rto (Response Time out), – разряд истечение времени ожидания ответного слова ОУ: Указывает на то, что МСП выступал в качестве принимающего ОУ в передаче типа «ОУ-ОУ» и что передающий ОУ или не ответил вовсе, или ответил позднее установленного для ожидания ОС срока. «Интервал ожидания ОС» определён как временной промежуток от момента пересечения нулевого уровня разрядом контроля по четности (в середине разряда) командного слова «передача» до момента пересечения нулевого уровня синхросигналом (середина синхросигнала) ОС. В нерасширенном режиме работы (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») «интервал ожидания ОС» равен 18,5 мкс (±1 мкс). Если eme, 15-ый разряд регистра cfg3_, установлен в лог. «1», значение интервала ожидания ОС может выбираться посредством разрядов 10 и 9 регистра cfg3_ из следующего перечня: 18,5; 22,5; 50,5 и 130 мкс (±1 мкс).

ltf (Loop Test Fail), – разряд ошибка проверочного возврата: В режиме ОУ «проверка возврата» производится на предаваемой части каждого негруппового сообщения. МСП делает оценку достоверности принятой версии каждого переданного в ЛПИ слова. К тому же проводится поразрядное сравнение последнего слова, передаваемого ОУ, в каждом сообщении. Если принятая версия какого-либо слова определена как ошибочная (ошибка синхронизации, кодирования, количества разрядов и/или контроля по чётности) и/или принятая версия последним переданного слова не равна исходной, то разряд ltf будет установлен в лог. «1».

Так же разряд ltf установится, если во время выдачи слов данных оконечным устройством поступит недостоверное слово (слова) по альтернативному каналу.

cbr (Circular Buffer Rollover), – разряд ‘прокручивание’ кольцевого буфера: Применяется только, если включен режим «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»), разрешен разряд «расширенное управление памятью ОУ» (ermm, 1-ый разряд регистра cfg2_ установлен в лог. «1»), и сделан выбор «кольцевая буферизация» для подадреса текущего сообщения средствами слова управления подадресом ОУ. Этот разряд будет установлен в лог. «1», если значение указателя кольцевого буфера, в поисковой таблице, перейдёт к максимально по значению адреса верхней границе кольцевого буфера и после операции чтения/записи слова по указателю произойдёт событие – ‘прокручивание’.

Если «переписывать недостоверные данные» разрешено (oid, 11-ый разряд регистра cfg2_ установлен в лог. «1»), этот разряд будет устанавливаться только после завершения сообщения «передача», либо достоверного сообщения «приём», которые привели к прокручиванию кольцевого буфера. Если «переписывать недостоверные данные» запрещено (oid, 11-ый разряд регистра cfg2_ установлен в лог. «0»), то этот разряд будет устанавливаться сразу после ‘прокручивания’ вне зависимости от достоверности пересылаемого при этом сообщения.

Следует отметить, что в режиме нерасширенных возможностей конфигурирования («разрешение расширенных возможностей конфигурирования», разряд 15 регистра cfg3_, установлен в лог. «0»), разряды с 7 по 0 всегда будут находиться в состоянии лог. «0».

icw (Illegal Command Word), – разряд признак недопустимости командного слова: Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»).

Если этот разряд установлен в лог. «1», это указывает, что сообщение было признано недопустимым. Недопустимым сообщение признаётся, если (разряд eme регистра cfg3_ установлен в лог. «0» или разряд id, «запрещение в ОУ проверки КС на недопустимость», регистра ctg3_ установлен в лог. «0») и разряд, отвечающий за допустимость, для данной комбинации разрядов принятого командного слова установлен в таблице допустимости КС в состояние лог. «1». Адреса расположения таблицы допустимости КС в разделяемом ОЗУ - 0300÷03FF (шестн.).

При работе ОУ недопустимость сообщения определяется поиском по таблице допустимости КС для значения принятого достоверного КС с использованием следующих разрядов КС: флаг «групповое/негрупповое» КС (BROADCAST*/OWN_ADDRESS), разряд «Приём-передача» (T/R), поле разрядов «Подадрес/Режим управления» (SA[4..0]), поле разрядов «Число СД/Код команды». Флаг «групповое/негрупповое» КС (BROADCAST*/OWN_ADDRESS) имеет значение 1’b0 для групповых сообщений и 1’b1 – для собственного адреса ОУ.

wce (Word Count Error), – разряд ошибка количества слов в сообщении: Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»). В случае лог. «1» указывает на то, что КШ не передал заявленного количества слов данных.

ids (Incorrect Data Sync), – разряд некорректный тип синхросигнала в слове данных: Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»). В случае лог. «1» указывает, что КШ передал командный синхроимпульс в слове данных.

iw (Invalid Word), – разряд признак недостоверности слова: Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»). Указывает, что КШ (или передающий ОУ в сообщении типа «ОУ-ОУ») передал сообщение с одним или более словом, содержащим одну или более ошибку из следующего списка: ошибка поля синхроимпульса, ошибка кодирования, ошибка контроля по четности и/или ошибка количества разрядов.

rtg_s_ae (RT-RT Gap/Sync/Address Error), – разряд ошибка в передаче ОУ-ОУ» (интервала ожидания ОС / синхросигнала или формата / адрес ОУ): Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»). Этот разряд устанавливается в лог. «1», если ОУ МСП является принимающим ОУ для передачи «ОУ-ОУ» и происходит хотя бы одно событие из следующего списка:

gce, 8-ой разряд регистра cfg5_ - разрешение проверки минимальной паузы перед выдачей ОС - установлен в лог «1», и предающий ОУ отвечает ОС через время меньшее, чем 4 мкс (согласно ГОСТ Р 52070-2003 - это промежуток между серединами зон разряда контроля по четности последнего КС/СД и синхроимпульса ОС), т.е. реально проверяется, что период отсутствия декодируемых сигналов в ЛПИ (‘мёртвое время’) меньше 2 мкс;

определена ошибка типа синхросигнала или формата (ошибка кодирования, количества разрядов и/или контроля по чётности) в предаваемом ОУ слове;

значение адреса в передаваемом ОС не равно значению адреса в командном слове «передача».

rt_2cwe (RT-RT 2nd Command Word Error), – разряд индикатор ошибки 2-ого командного слова: Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»). Если МСП является принимающим

ОУ для передачи типа «ОУ-ОУ», то появление лог. «1» в rt_2cwe указывает на наличие одной или более ошибки из следующего списка ошибок в командном слове «передача»:

1) разряд «Приём-передача» установлен в лог. «0»;

2) подадрес равен 5’b00000 или 5’b11111

3) адрес ОУ в адресном поле такой же как и в командном слове «приём».

cwce (Command Word Contents Error), – разряд принятое КС не согласуется с требованиями ГОСТ Р 52070-2003: Применяется только в режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»). Указывает, что «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003». Это касается следующих разрядных комбинаций в командных словах:

1) разряд bd (7-ой разряд регистра cfg5_) установлен в лог. «0» и командное слово сообщения не является КУ и поле КС «Адрес ОУ» имеет значение 5’b11111 (групповая) и разряд КС «Приём-передача» установлен в лог. «1» (передача);

2) разряд ome («Обрабатывать КУ на «прием» без СД как резервные КУ» регистра cfg3_ установлен в лог. «0» и сообщение с разрядом «Приём-передача» установленным в лог. «0» и со значением поля «Подадрес/Режим управления» равным 5b’00000 или 5’b11111 и со значением поля кода для КУ между 5’b00000 и 01111;

3) разряд bd («запрет групповых сообщений») регистра cfg5_ установлен в лог. «0» и КУ, для которой не разрешено применение в групповом сообщении (например, разряд КС «Приём-передача» установлен в лог. «1»), содержит в поле КС «Адрес ОУ» значение 5’b11111.

1.3.12.2.2.6.4.2 Слово метки времени

Слово метки времени занимает третье местоположение в описателе блока сообщения в стеке команд ОУ, обновляется (подвергается записи) и во время старта отработки сообщения (SOM), и во время завершения отработки сообщения (EOM). Последовательность SOM автоматных действий МСП случается сразу, непосредственно после получения достоверного командного слова по адресу ОУ или группового адреса. Последовательность EOM происходит приблизительно после 3-4 мкс после последнего принятого или переданного слова сообщения.

Во время, когда производятся указанные последовательности, текущее значение внутреннего регистра счетчика метки времени tt_reg переписывается в соответствующее местоположение ОЗУ. Время разрешения регистра tt_reg задается программно и может быть выбрано из следующего перечня значений 2, 4, 8, 16, 32, 64 мкс/МЗР, или определяется внешним источником тактовой частоты, посредством разрядов 9, 8 и 7 регистра cfg2_.

Переполняясь, значение регистра tt_reg будет возвращаться от FFFF (шестн.) к 0000(шестн.) после отсчета каждых 65536 тактов (количество изменений МЗР). Когда происходит переполнение, запрос прерывания по переполнению счётчика метки времени будет вырабатываться, если прерывание от данного источника запроса разрешено в регистре imr. Вдобавок к тому, разряд «переполнение счетчика метки времени» (tt_rol) в регистре состояния прерываний будет оказываться установленным в лог. «1» после переполнения, если:

1) прерывание разрешено (не замаскировано) или

2) «расширенные возможности прерывания» разрешены («расширенные возможности конфигурирования» и разряд 15 регистра cfg2_ запрограммированы в лог. «1»).

Если разряд «сброс счетчика метки времени при принятии КУ ‘Синхронизация’» (c_ttos, 6-ой разряд регистра cfg2_) запрограммирован в лог. «1», то значение регистра tt_reg ОУ МСП будет очищено (приведено в состояние 0000(шестн.)) после получения КУ ‘Синхронизация’.

Если разряд «загрузка СД в «регистр счетчика метки времени» при принятии КУ ‘Синхронизация (с СД)’» (l_ttos, 5-ой разряд регистра cfg2_) установлен в лог. «1», то значение слова данных, пришедшего совместно с КУ ‘Синхронизация (с СД)’, будет загружаться в регистр tt_reg ОУ МСП.

1.3.12.2.2.6.4.3 Указатель блока данных сообщения или команда управления

Третье местоположение внутри описателя блока сообщения ОУ может быть употреблено для различных целей. Для «сообщений пересылки данных» или для команд управления при запрещенной «расширенной обработки КУ» (разряды 15 или 0 регистра cfg3_ установлены в лог. «0»), значение, прочитанное из таблицы указателей для подадресов на «прием»/«групповой прием»/«передачу», сохраняется (записывается) в этом местоположении. Это значение указывает на начальную (первую) ячейку блока (буфера) данных соответствующего подадреса отработанного сообщения.

Для команд управления, сопровождаемых данными, или в случае, когда разрешена «расширенная обработка КУ» (разряд 15 и 0 регистра cfg3_ установлены в лог. «1»), слова данных, которые были переданы или получены вместе с командой управления, записываются в это местоположение во время EOM последовательности.

Для команд управления, не сопровождаемых данными, в третье местоположение описателя блока сообщения ничего не записывается.

Имеются особые ситуации для команд управления.

Если КС является командой управления «передача» и передаваемое слово данных берется из регистра МСП (напр. ‘Передать слово ВСК ОУ’, ‘Передать векторное слово’, ‘Передать последнюю команду’), то указатель на блок данных ОУ и регистр rm_dsa[15:0] содержат передаваемое слово данных.

Указатель на блок данных ОУ и регистр rm_dsa[15:0] содержат слово ВСК, если КС было неопределенной командой управления, разряд «Приём-передача» равен лог. «0», поле «Число СД/код команды» равно 5’b10011 и 15 разряд ebwe регистра cfg4_ («разрешение внешнего слова ВСК») равен лог. «0».

Если 0 разряд emch регистра cfg3_ («расширенная обработка КУ») равен лог. «1», то после отработки сообщения, где КС было неопределенной командой управления, разряд «Приём/передача» равен лог.0, поле «Число СД/Код команды» равно 5’b10010, регистр rm_dsa[15:0] и указатель на блок данных ОУ содержат предыдущее достоверное командное слово.

1.3.12.2.2.6.4.4 Принятое командное слово

Для сообщений принятое командное слово записывается в четвертую позицию описателя блока сообщения, что происходит во время выполнения SOM последовательности действий.

1.3.12.2.2.6.4.5 Вытеснение (замещение) команд

Последовательность вытеснения сообщения происходит, когда отработка достоверного сообщения, пересылка которого осуществлялась по одному каналу ЛПИ, не закончилась, а появилось новое КС по другому, резервному каналу ЛПИ. Если старое и новое КС, направлены по одному и тому же «Адресу ОУ», то данное ОУ обязано остановить обработку текущего сообщения и приступить к обработке нового КС. Если новое КС направлено по другому адресу, то ОУ, отвечающий на первое КС, должен довести обработку текущего сообщения до конца, игнорируя активность на другом канале.

МСП в режиме ОУ будет прекращать обработку первого сообщения и приступать к обработке нового сообщения. Единственным исключением этого будет случай, когда МСП работает в комбинированном режиме ОУ/ «монитор сообщений», и первая команда направлялась по собственному «Адресу ОУ», а вторая - по адресу другого ОУ (не по групповому адресу). Как ОУ, МСП должен продолжить обработку первой команды (в соответствии с ГОСТ Р 52070-2003) и игнорировать новую команду, направленную другому ОУ, по другому каналу ЛПИ.

Монитор сообщений МСП будет всегда замещать сообщения по тому же самому каналу ЛПИ за исключением случая, когда вытесняющая команда может быть интерпретирована как ОС. Следовательно, ответный отклик ОУ с неправильным адресом ОУ в ОС не будет

определяться МСП, в режиме монитора сообщений, в качестве новой команды, но будет распознаваться как неправильное ОС .

При вытеснении сообщения обычная SOM последовательность производится для нового вытесняющего сообщения. Отметим, что для старого, незаконченного, вытесняемого сообщения последовательность EOM не выполняется. Это будет приводить к тому, что разряды SOM и EOM слова состояния блока сообщения в стеке команд будут установлены в лог. «1» и лог. «0» соответственно. УВ может различить между собой «сообщение в обработке» и «вытесненное сообщение» по значению указателя на стек команд монитора. Если указатель на стек команд монитора был увеличен, и слово состояния блока сообщения указывает на SOM последовательность, тогда сообщение, по-видимому, было замещено.

1.3.12.2.2.6.5 Прерывание в режиме ОУ

МСП предлагает большую степень гибкости с точки зрения возможностей УВ по обработке прерываний от ОУ. В некоторых системах передача и прием сообщений с особенными подадресами означает завершение пересылки множества последовательных данных. ОУ МСП позволяет вырабатывать запрос прерывания после завершения сообщения на «прием»/»передачу» данных адресованному к конкретному/определенному подадресу ОУ. В этом случае пользователю следует применять разряд(ы) «разрешение прерывания по завершению сообщений «передачи»/«приема»/«группового приема»» в слове управления подадресом ОУ для того, чтобы породить запрос на обработку прерывания только для особенного, определенного «принимающего/передающего» подадреса команды, а не после обработки каждого сообщения. Можно для УВ воспользоваться затем возможностью переключения активной области ОЗУ (А/Б), изменяя значение 13-го разряда регистра cfg1_.

Запрос прерывания при ‘прокручивании’ кольцевого буфера будет возникать только после окончания «сообщения передачи данных», во время которого последнее местоположение кольцевого буфера было прочитано, или после окончания достоверного «сообщения приема данных», для которого последнее неиспользованное местоположение в буфере было подвержено записи.

Если «расширенная обработка КУ» включена («разрешение расширенных возможностей конфигурирования» ,eme в лог. «1», и emch, 0-ой разряд регистра cfg3_, установлен в лог. «1»), то прерывания могут быть разрешены для отдельных, задаваемых при конфигурировании КУ. Эти прерывания могут быть разрешены посредством таблицы прерываний для избранных КУ с адресами расположения в разделяемой памяти 0108÷010F.

1.3.12.2.2.6.6 Выполнение передачи данных большого объема

В системах, вовлеченных в пересылку данных большого объема по ЛПИ, удобнее всего использовать кольцевую буферизацию данных. УВ следует установить разряд «переписывать недостоверные данные» (oid, 11-ый разряд регистра cfg2_) и разрешить запрос прерывания по событию «‘прокручивание’ кольцевого буфера данных ОУ» (разряд m_rt_cbr регистра imr). При этом утомительная задача пересылки множества сообщений по выбранному подадресу, включая реакции на ошибки и ВПС, становится легко выполнимой для УВ. МСП будет порождать запрос прерывания только после того, как ожидаемое количество достоверных слов данных становится принятым по определенному подадресу. Ожидаемое количество слов, которое может быть передано/получено в/из ЛПИ, устанавливается программно и может достигать 8192 слов.

1.3.12.2.2.6.7 Задание и соблюдение допустимости КС

МСП обеспечивает встроенный механизм для определения допустимости для поступающих командных слов.

Множество недопустимых сообщений для данного ОУ задаётся при конфигурировании значениями слов в таблице, размещаемой по фиксированным адресам ОЗУ (от 300 шестн. до 3FF

шестн.). При работе ОУ недопустимость сообщения определяется поиском по таблице допустимости КС для значения принятого достоверного КС с использованием следующего шаблона (значений разрядов КС): флаг «групповое/негрупповое» КС (BROADCAST*/OWN_ADDRESS), разряд «Приём-передача» (T/R), поле разрядов «Подадрес/Режим управления» (SA[4..0]), поле разрядов «Количество СД/Код команды» (WC[4:0]/MC[4:0]). Флаг «групповое/негрупповое» КС (BROADCAST*/OWN_ADDRESS) имеет значение 1'b0 для групповых сообщений и 1'b1 – для собственного адреса ОУ.

Преимущество от обладания МСП этим свойством состоит, прежде всего, в том, что понижаются требования к объему памяти УВ - исключается необходимость производить оценку допустимости КС с использованием внешних (дополнительных) устройств ПЗУ, ПЛИС или ОЗУ. Схема определения допустимости КС обеспечивает наибольшую гибкость в задании какого-либо допустимого для конкретного электронного модуля подмножества из 4096 возможных разрядных комбинаций командного слова сообщения.

При включении питания, по умолчанию, процедура проверки допустимости КС оказывается разрешенной (включенной). Проверку допустимости КС можно отключить сначала установлением режима «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, уст. в лог. «1») и, затем, установлением id, 7-ого разряда регистра cfg3_ («запрещение в ОУ проверки КС на недопустимость») в лог. «1». Установление id надо делать, непременно, в то время, когда eme, 15-ый разряд регистра cfg3_, заранее уже был установлен в лог. «1», одновременно с этим вновь прописывая в eme высокий лог. уровень.

Если процедура проверки допустимости КС используется, адреса ОЗУ с 0300 по 03FF считаются зарезервированными для функции установления допустимости КС и не должны заниматься стеком или содержать блоки данных. Если процедура проверки допустимости КС запрещена, МСП предполагает, что все достоверные и определенные в рамках ГОСТ Р 52070-2003, полученные командные слова допустимы. Если проверка допустимости КС запрещена, местоположения с адресами с 0300 по 03FF могут быть заняты под стек команд или блоки данных.

Следует отметить, что состояние id, 7-ого разряда регистра cfg3_ - «запрещение в ОУ проверки КС на недопустимость» - не имеет значения для режимов КШ или МТ. Для проверки допустимости командных слов в режиме ОУ в разделяемом ОЗУ МСП выделяется 256-словная область с адресами расположения с 0300 по 03FF. Групповые КС проверяются раздельно от негрупповых.

КС могут проверяться на допустимость вплоть до уровня, определяемого количеством слов в команде. Например, КС «приём» с одним СД с подадресом 1 может оказаться допустимой (законной), в то время как аналогичное КС, но для двух СД, с тем же подадресом, может оказаться недопустимой.

1.3.12.2.2.6.7.1 Результаты проверки допустимости КС

Если протокольная логика ОУ МСП выявляет, что негрупповое сообщение недопустимо, МСП откликнется на это установлением признака «Ошибка в сообщении» в ОС ОУ. Если команда «передача» была определена как недопустимая, то МСП откликнется на неё выдачей в ЛПИ ОС с установленным признаком «Ошибка в сообщении», но слова данных в ответ на команду в ЛПИ передавать не будет.

Если сообщение «приёма» СД было определено недопустимым и МСП находится в режиме «нерасширенные возможности конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «0») или i_rx_td, 4-ый разряд регистра cfg3_ - «запрещение сохранения принимаемых СД данных недопустимого сообщения» - установлен в лог. «0», то слова данных, получаемые в сообщении типа «ОУ-ОУ» (или «КШ-к-ОУ»), будут сохраняться в ОЗУ.

Однако, если МСП находится в режиме «расширенные возможности конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «1») и i_rx_td, 4-ый разряд регистра cfg3_ - «запрещение сохранения принимаемых СД данных недопустимого сообщения» - установлен в

лог. «1», то слова данных, получаемые в сообщении типа «ОУ-ОУ» (или «КШ-к-ОУ»), не будут сохраняться в ОЗУ.

В случае недопустимости КС группового сообщения, МСП не будет выдавать ответное слово, но будет устанавливать в лог. «1» признаки «Ошибка в сообщении» и «Принята групповая команда» в регистре ответного слова ОУ. Если следующим сообщением будет ‘Передать ОС’ или ‘Передать последнюю команду’, МСП откликнется на это ОС, с установленным в лог. «1» признаком «Ошибка в сообщении». Станут ли сохраняться данные в памяти МСП или нет, при получении команды «группового приема», также определяется разрядом «запрещение сохранения принимаемых СД данных недопустимого сообщения приема», как было описано выше.

Если ОУ МСП установлен в режим «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»), и КС было определено как недопустимое, то признак недопустимости КС icw, 6-ой разряд слова состояния блока сообщения, будет устанавливаться в лог. «1» во время записи слова состояния блока сообщения в описателе блока сообщения, располагающийся в стеке команд, и во время SOM последовательности действий, и во время EOM последовательности действий.

1.3.12.2.2.6.7.2 Таблица допустимости

Таблица 1.647 – Правило формирования указателя на «слово определения допустимости КС» в таблице допустимости КС

Разряд	Описание
15	лог. «0»
14	лог. «0»
13	лог. «0»
12	лог. «0»
11	лог. «0»
10	лог. «0»
9	лог. «1»
8	лог. «1»
7	BROADCAST*/OWN_ADDRESS
6	«Приём-передача»
5	SA4
4	SA3
3	SA2
2	SA1
1	SA0
0	WC4/MC4

Замечания к таблице 1.647:

1) Разряд 7, BROADCAST*/OWN_ADDRESS - лог. «0» для группового КС, лог. «1» - негруппового КС.

2) Разряд 6, «Приём-передача» - высокий уровень для КС «передачи», низкий - для КС «приема».

3) Разряды с 5 по 1 (SA4 – SA0) определяются полем «Подадрес/Режим управления» КС. Подадреса, равные значению 5’b00000 или 5’b11111, предполагают принадлежность сообщения к КУ. Подадреса с 5’b00001 по 5’b11110 - к обычным КС информационного обмена.

4) Адресный разряд 0, WC4/MC4, определяется старшим разрядом поля «Число СД/код команды». Если WC4/MC4 равен лог. «0», то соответствующее слово в таблице допустимости КС используется для задания допустимости КС со значениями поля «Число СД/Код команды» от 15 до 0. Если WC4/MC4 равен лог. «1», соответствующее слово в таблице допустимости КС используется для задания допустимости командам со значениями в поле «Число СД/Код команды», изменяющимися в пределах от 31 до 16.

5) Значение 5'b00000 (WC), как и для поля КС «Число СД/Код команды», используется для 32-словного сообщения.

6) Обозначения: WCn, MCn, SAn, - n-й разряд значения «количество слов», «код команды» управления, «подадрес/режим управления», соответственно.

Организация таблицы допустимости КС в памяти показана в таблице 7.10. Базовый адрес таблицы - 0300 (шестн.). Согласно таблице 7.9, указатель на «слово определения допустимости КС» в таблице допустимости КС формируется посредством следующих разрядов КС: поле «Адрес ОУ» - «групповое/негрупповое» (BROADCAST*/OWN_ADDRESS), разряд «Приём-передача» (T/R), поле «Подадрес/Режим управления» (SA[4..0]) и старшего разряда поля «Число СД/Код команды» (WC4/MC4).

Первые 64 слова таблицы допустимости КС используются для задания и соблюдения допустимости «групповых КС приема» (2 слова определения допустимости на подадрес). Следующие 64 слова приписаны к «групповым КС передачи». Так как «групповые КС передачи» не определены для ГОСТ Р 52070-2003, то программировать (инициализировать) зону с табличными адресами от 0372 по 037D (шестн.) нет необходимости. Следующие 64 слова соответствуют «не групповым КС» (см. таблицу 1.648).

Таблица 1.648 - Структура таблицы допустимости КС

Здесь Rx – «приём», Tx – «передача».

Адреса	Описание
0300	Brdcst/Rx ,SA 0, MC 15-0
0301	Brdcst/Rx, SA 0, MC 31-16
0302	Brdcst/Rx, SA 1, WC 15-0
0303	Brdcst/Rx, SA 1, WC 31-16
...	...
033F	Brdcst/Rx, SA 31, MC 31-16
0340	Brdcst/Tx, SA 0, MC 15-0
0341	Brdcst/Tx, SA 0, MC 31-16
0342	Brdcst/Tx, SA 1, WC 15-0
...	...
Замечание:	Групповые информационные команды «передача» (SA1-SA30) не определены в ГОСТ Р 52070-2003. Следовательно , адреса 0342-037D не нужно программировать.
037D	Brdcst/Tx, SA 30, WC 31-16
037E	Brdcst/Tx, SA 31, MC 15-0
037F	Brdcst/Tx, SA 31, MC 31-16
0380	Own Addr/Rx, SA 0, MC 15-0
0381	Own Addr/Rx, SA 0, MC 31-16
0382	Own Addr/Rx, SA 1, WC 15-0
0383	Own Addr/Rx, SA 1, WC 31-15
...	...
03BE	Own Addr/Rx, SA 31, MC 15-0
03BF	Own Addr/Rx, SA 31, MC 31-16
03C0	Own Addr/Tx, SA 0, MC 15-0
03C1	Own Addr/Tx, SA 0, MC 31-16
03C2	Own Addr/Tx, SA 1, WC 15-0

Продолжение таблицы 1.648

Здесь Rx – «приём», Tx – «передача».

Адреса	Описание
03C3	Own Addr/Tx, SA 1, WC 31-16
...	...
03FC	Own Addr/Tx, SA 30, WC 15-0
03FD	Own Addr/Tx, SA 30, WC 31-16
03FE	Own Addr/Tx, SA 31, MC 15-0
03FF	Own Addr/Tx, SA 31, MC 31-16

Для каждой комбинации разрядов КС: поле «Адрес ОУ» - «групповое/негрупповое» КС (BROADCAST*/OWN_ADDRESS), разряд «Приём-передача» (T/R), поле «Подадрес/Режим управления» (SA[4..0]), в таблице допустимости КС имеется по два слова (суммарно - 32 разряда), что делает возможным задать допустимость для любого из 32 возможных значений поля КС «Число СД/Код команды».

Назначение разрядов двух «слов определения допустимости КС» для одного подадреса поясняется в таблицах 1.649 и 1.650.

Таблица 1.649 - Слово определения допустимости КС для WC4/MC4 = 0

Номер разряда	Значения разрядов «слова определения допустимости КС» при WC4/MC4 = 0 (четное слово) (значение разряда 1'b0 – признак допустимости, 1'b1 - недопустимости)
15 (СЗР)	допустимость КС с полем «Число СД/Код команды» равным 15
14	допустимость КС с полем «Число СД/Код команды» равным 14
13	допустимость КС с полем «Число СД/Код команды» равным 13
12	допустимость КС с полем «Число СД/Код команды» равным 12
11	допустимость КС с полем «Число СД/Код команды» равным 11
10	допустимость КС с полем «Число СД/Код команды» равным 10
9	допустимость КС с полем «Число СД/Код команды» равным 9
8	допустимость КС с полем «Число СД/Код команды» равным 8
7	допустимость КС с полем «Число СД/Код команды» равным 7
6	допустимость КС с полем «Число СД/Код команды» равным 6
5	допустимость КС с полем «Число СД/Код команды» равным 5
4	допустимость КС с полем «Число СД/Код команды» равным 4
3	допустимость КС с полем «Число СД/Код команды» равным 3
2	допустимость КС с полем «Число СД/Код команды» равным 2
1	допустимость КС с полем «Число СД/Код команды» равным 1
0(МЗР)	допустимость КС с полем «Число СД/Код команды» равным 0

Таблица 1.650 - Слово определения допустимости КС для WC4/MC4=1

Номер разряда	Значения разрядов «слова определения допустимости КС» при WC4/MC4 = 1 (нечетное слово) (значение разряда 1'b0 – признак допустимости, 1'b1 - недопустимости)
15 (СЗР)	допустимость КС с полем «Число СД/Код команды» равным 31
14	допустимость КС с полем «Число СД/Код команды» равным 30
13	допустимость КС с полем «Число СД/Код команды» равным 29
12	допустимость КС с полем «Число СД/Код команды» равным 28
11	допустимость КС с полем «Число СД/Код команды» равным 27
10	допустимость КС с полем «Число СД/Код команды» равным 26
9	допустимость КС с полем «Число СД/Код команды» равным 25
8	допустимость КС с полем «Число СД/Код команды» равным 24
7	допустимость КС с полем «Число СД/Код команды» равным 23

Продолжение таблицы 1.650

Номер разряда	Значения разрядов «слова определения допустимости КС» при WC4/MC4 = 1 (нечетное слово) (значение разряда 1'b0 – признак допустимости, 1'b1 - недопустимости)
6	допустимость КС с полем «Число СД/Код команды» равным 22
5	допустимость КС с полем «Число СД/Код команды» равным 21
4	допустимость КС с полем «Число СД/Код команды» равным 20
3	допустимость КС с полем «Число СД/Код команды» равным 19
2	допустимость КС с полем «Число СД/Код команды» равным 18
1	допустимость КС с полем «Число СД/Код команды» равным 17
0(МЗР)	допустимость КС с полем «Число СД/Код команды» равным 16

В отношении таблиц 1.649 и 1.650 следует сделать следующие замечания:

1) Чтобы установить недопустимость какого-либо КС, с данным сочетанием значений поле «Адрес ОУ» - «групповое/негрупповое» (BROADCAST*/OWN_ADDRESS), разряда «Приём-передача» (T/R), поля «Подадрес/Режим управления» (SA[4..0]) и данным значением поля «Число СД/Код команды», соответствующий разряд слова таблицы допустимости КС («слово определения допустимости КС»), позиция которого (в одном из двух слов таблицы допустимости КС) определяется значением поля «Число СД/Код команды», должен быть запрограммирован в лог. «1»; значение лог. «0» определяет соответствующее командное слово как допустимое.

2) для величин подадресов с 5'b00001 по 5'b11110 поле WC/MC употребляется в командном слове для задания «количества слов сообщения»; для подадресов 5'b00000 и 5'b11111 поле WC/MC употребляется в командном слове для задания КУ.

3) так как сообщения «групповой передачи данных» не определены для ГОСТ Р 52070-2003, 60 слов таблицы допустимости КС с адресами с 0342 по 037D соответствующие этим командам не нуждаются в инициализации; МСП никак не будет реагировать на такие команды «групповой передачи», за исключением разве что установки признака «Ошибка в сообщении» в регистре ответного слова ОУ вне зависимости от того, был ли установлен в лог. «1» соответственный разряд «слова определения допустимости КС» таблицы или нет. Если следующим поступит сообщение с КУ 'Передать ОС' или 'Передать последнюю команду', то МСП ответит ОС с установленным в лог. «1» признаком «Ошибка в сообщении».

1.3.12.2.2.6.7.3 Работа с таблицей допустимости КС

Для того, чтобы использовать свойство определения допустимости КС, УВ следует записывать информацию в 256 словных ячеек таблицы допустимости КС ОЗУ, как часть общего потока данных процедуры конфигурирования (инициализации) МСП, сразу после включения питания. После того, как таблица ОЗУ оказывается загруженной, процессор, по желанию, может прочитать записанные в неё данные для проверки (сравнения получившегося и желаемого результатов инициализации). Когда инициализация таблицы допустимости КС завершается, УВ следует установить МСП в режим ОУ, записав соответствующий код в регистр cfg1_.

Если это желательно, УВ может регулярно перечитывать содержимое памяти в зоне таблицы допустимости КС для постоянного контроля целостности данных, загруженных в неё при инициализации. Самостоятельно же «логика протокола обмена по ГОСТ Р 52070-2003» МСП будет производить только единственный доступ по чтению во время SOM последовательности действий ОУ. Результатом этого доступа по чтению к таблице и дополнительных действий МСП по анализу прочитанного слова явится определение: было или не было разрешено управляющим вычислителем для использования конкретное, принятое командное слово обрабатываемого сообщения. Так как область памяти таблицы допустимости КС не должна использоваться стеком команд либо буферами данных сообщений (предполагается, что проверка допустимости разрешена), логика протокола обмена по ГОСТ Р 52070-2003 МСП никогда самостоятельно не будет что-либо записывать в эту зону памяти.

1.3.12.2.2.6.8 Обработка КУ, таблица слов данных для КУ

МСП имеет механизм для сохранения слов данных, принадлежащих КУ, в индивидуальные (особые) местоположения разделяемого с УВ ОЗУ. Дополнительно, МСП имеет возможность выборочно порождать запросы прерывания для определенных КУ, основанную на таблице прерываний для избранных КУ. Эти свойства включаются в режиме «расширенных возможностей конфигурирования» МСП посредством установки в лог. «1» eme, 15-го разряда регистра cfg3_, и последующей за этим установкой в лог. «1» emch, 0-ого разряда регистра cfg3_,- «расширенная обработка КУ».

«Расширенная обработка КУ» разрешает три различных вида обработки КУ:

1) Для всех обычных (информационных) сообщений и для сообщений, содержащих командные слова КУ, при запрещенной «расширенной обработке КУ», третье слово описателя блока сообщения в стеке зарезервировано для указателя на блок данных сообщения (см. таблицу Таблица 1.). Указатель на блок данных сообщения ссылается на первую позицию блока данных для принимаемых, либо передаваемых данных. В случае КУ блок данных состоит из единственного слова данных.

Тем не менее, если «расширенная обработка КУ» разрешена, для КУ третье слово в описателе блока сообщения в стеке команд ОУ будет содержать подлинное слово данных (переданное или полученное), а не указатель на однословный блок данных. Для КУ без данных третье слово описателя не подвергается записи.

2) «Расширенная обработка КУ» влияет (воздействует) на адресное распределение слов данных для КУ «передать» СД или КУ «принять» СД. С разрешенной «расширенной обработкой КУ» для «КУ со словами данных» (команды «передача» и «приём» с кодами с 10 по 1F(шестн.)) МСП будет читать данные из или сохранять данные в индивидуальные (для каждой КУ свое) фиксированные местоположения в ОЗУ. Адреса местоположений, связанные с «КУ приема», - 110-11F(шестн.). Выбор адресов конкретных местоположений слов данных основывается на значениях разрядов поля «Код команды» КС КУ (разряды 3÷0). Таблица слов данных КУ «передача» будут читаться из местоположений с адресами 120÷12F (выбор конкретного местоположения также основан на разрядах 3÷0 поля «Код команды» КС). Если выбор «отделение данных групповых сообщений» сделан (sbd, 0-ой разряд регистра cfg2_, установлен в лог. «1»), слова данных «групповой КУ приема» будут сохраняться в местоположениях с адресами с 130 по 13F(шестн.). Если выбор «отделение данных групповых сообщений» не сделан (sbd, 0-ой разряд регистра cfg2_, установлен в лог. «0»), то принимаемые данные «групповой КУ» будут сохраняться в зоне с адресами 110÷11F(шестн.). Таблица 1. показывает местоположения ОЗУ для каждого из КС «КУ с данными».

Распределение в ОЗУ таблицы слов данных КУ и структура слова приведено в таблице Таблица 1..

3) Прерывания по заданным КУ разрешаются в МСП для режима «расширенные возможности конфигурирования» путем установления разряда «расширенная обработка КУ» (emch) регистра cfg3_ в лог. «1» и установлением разряда «принята избранная команда управления» (m_ss_mc_pt) регистра imr в лог. «1» и инициализацией таблицы прерываний для избранных КУ (область ОЗУ с адресами 108÷10F(шестн.)). Ячейки в таблице прерываний для избранных КУ адресуются указателем, который получается в результате сложения базового адреса 108(шестн.) с 3-х разрядным значением - {(BROADCAST/OWN_ADDRESS*), (T/R), WC4/MC4}, состоящим из флага «групповое/негрупповое» КС (BROADCAST/OWN_ADDRESS*), разряда «Приём-передача» (T/R), и разряда 4 (старшего) поля КС «Код команды». Разрядное местоположение (внутри 16-ти разрядного слова), которое определяет, следует ли вырабатывать прерывание, однозначно задаётся 4-мя младшими разрядами (3-0) поля КС «Число СД/Код команды». Если соответствующий разряд был своевременно запрограммирован в лог. «1», МСП сформирует прерывание в конце текущего сообщения КУ. Отметим, что некоторые возможные комбинации значений разрядов в

командных словах не определены стандартом. Поскольку при приёме каждое командное слово проверяется ОУ на удовлетворение критериям достоверности по ГОСТ Р 52070-2003, прерывание будет вырабатываться в конце сообщения, даже если отдельная команда не определены стандартом или недопустима. Распределение в ОЗУ таблицы прерываний для избранных КУ и структура слова таблицы прерываний для избранных КУ приведены в таблицах 1.651 и 1.652.

Таблица 1.651 -Таблица слов данных КУ

Местоположение в ОЗУ	Код команды (поле КС)
0110	Не определено
0111	Синхронизация (с СД)
0112	Не определено
0113	Не определено
0114	Блокировать i-й передатчик
0115	Разблокировать i-й передатчик
0116-011F	Резерв (КУ приема)
0120	Передать векторное слово
0121	Не определено
0122	Не определено
0123	Передать слово ВСК ОУ
0124	Не определено
0125	Не определено
0126-012F	Резерв (КУ передачи)
0130	Неопределенная групповая
0131	Групповая синхронизация (с СД)
0132	Неопределенная групповая
0133	Неопределенная групповая
0134	Групповая блокировать i-й передатчик
0135	Групповая разблокирование i-го передатчика
0136-013F	Резерв (групповая КУ приема)

Таблица 1.652 - Таблица прерываний для избранных КУ

Адреса	Описание
0108	КУ приема 0-15 (не определено)
0109	КУ приема 16-31 (с данными)
010A	КУ передачи 0-15 (без данных)
010B	КУ передачи 16-31 (с данными)
010C	КУ группового приема 0-15 (не определено)
010D	КУ группового приема 16-31 (с данными)
010E	КУ групповой передачи 0-15 (без данных)
010F	КУ групповой передачи 16-31 (не определено/резерв)

Таблица 1.653 - Структура слова таблицы прерываний для избранных КУ

Разряд	«Код команды» [4:0]	Структура слова таблицы прерываний для избранных КУ (показано для местоположения с адресом 010A)
15	4’b1111	Резервный код
14	4’b1110	Резервный код
13	4’b1101	Резервный код
12	4’b1100	Резервный код
11	4’b1011	Резервный код
10	4’b1010	Резервный код
9	4’b1001	Резервный код

Проолжение таблицы 1.653

Разряд	«Код команды» [4:0]	Структура слова таблицы прерываний для избранных КУ (показано для местоположения с адресом 010A)
8	4’b1000	Установить ОУ в исходное состояние
7	4’b0111	Разблокировать признак неисправности ОУ
6	4’b0110	Блокировать признак неисправности ОУ
5	4’b0101	Разблокировать передатчик
4	4’b0100	Блокировать передатчик
3	4’b0011	Начать самоконтроль ОУ
2	4’b0010	Передать ОС
1	4’b0001	Синхронизация
0	4’b0000	Принять управление интерфейсом

Примечание - 0 - младший разряд, 15 - старший разряд
Замечание - лог. «1» в разряде разрешает формирование запроса прерывания.

1.3.12.2.2.6.9 Запрет групповых сообщений

В режиме ОУ МСП разрешает использование групповых сообщений по выбору, устанавливаемому программным обеспечением. В режиме «нерасширенные возможности конфигурирования» ОУ, если разряд «запрет групповых сообщений» (bd) регистра cfg5_ установлен в «0», МСП будет откликаться на адрес с номером 31 как на групповой адрес. Если разряд bd запрограммировать в лог. «1», то адрес 31 не будет распознаваться как групповой и может быть употреблен как отдельный (дополнительный) адрес оконечного устройства. Следует заметить, что согласно требованиям ГОСТ Р 52070-2003 адрес 31 не может использоваться в качестве отдельного (собственного) адреса ОУ.

Если нет запрета групповых сообщений и МСП получает групповое КС (адрес ОУ 31), затем без паузы, слитное с ним КС «передача», направленное к собственному адресу ОУ, что говорит о групповом сообщении «передача» типа «ОУ-ОУ», то первое групповое командное слово будет вытеснено, и ОУ МСП будет отвечать на КС «передача». Для более детального описания смотри раздел «вытеснение команд».

1.3.12.2.2.6.10 Состояние «занято» ОУ (таблица занятости подадресов)

ОУ МСП позволяет программно задавать состояние «занято» (состояние ‘абонент занят’) для всех сообщений адресованных ОУ МСП или для определенной комбинации значений разрядов КС: флаг «групповое/негрупповое» КС (BROADCAST/OWN_ADDRESS*), разряд «Приём-передача» (T/R), поле «Подадрес/Режим управления» (SA[4..0]). Состояние «занято» означает, что ОУ МСП отвечает на сообщения, адресованные ему, ОС с признаком «Абонент занят» и не передает СД (за исключением КУ и разряд mcob, 13 разряд регистра cfg4_, равен лог. «1»).

В режиме «нерасширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «0») или, если «разрешение альтернативного формирования ОС» не установлено (aswe, 5-ый разряд регистра cfg3_, установлен в лог. «0»), состояние «занято» ОУ (признак «Абонент занят» в ОС) управляется посредством busy, 10-ого разряда регистра cfg1_. В режиме «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, в лог. «1») и, если «разрешение альтернативного формирования ОС» установлено (aswe, 5-ый разряд регистра cfg3_, в лог. «1»), тогда состояние «занято» ОУ (признак «Абонент занят» в ОС) управляется посредством s[3], 4-ого разряда регистра cfg1_.

При состоянии «занято» (признак «Абонент занят» в ОС установлен лог. «1»):

1) Если МСП не находится в режиме «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в «0») или, если «запрещение сохранения принимаемых СД в состоянии ‘абонент занят’» не установлено (b_rx_td, 3-ий разряд регистра

cfg3_, установлен в «0»), то для КС «приём» данные будут записываться в блок данных в ОЗУ по адресу, указанному в соответствующем слове таблицы указателей для подадресов на «прием/групповой прием».

2) Если МСП находится в режиме «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в «1») и, если установлено «запрещение сохранения принимаемых СД в состоянии ‘абонент занят’» (b_rx_td, 3-ий разряд регистра cfg3_, установлен в «1»), для команд «приём» (кроме КУ) данные не будут записываться в блок данных в ОЗУ по адресу, указанному в таблице указателей для подадресов на «прием/групповой прием» .

3) Для сообщения «передача» и не КУ МСП будет отзываться ОС с установленным признаком «Абонент занят», однако после ОС слова данных передаваться не будут.

Если выбор «разрешение альтернативного формирования ОС» не сделан, МСП может по необходимости программировать установление признака «Абонент занят» в ОС для определенной комбинации разрядов КС: флаг «групповое/негрупповое» КС (BROADCAST/OWN_ADDRESS*), разряд «Приём-передача» (T/R), поле «Подадрес/Режим управления» (SA[4..0]). «Альтернативное формирования ОС» не может устанавливаться вне режима «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») или, если разряд aswe, 5-ый разряд регистра cfg3_ («разрешение альтернативного формирования ОС»), установлен в лог. «0».

Если режим «альтернативное формирования ОС» не задействован, установление признака «Абонент занят» в ОС программируется для определенной комбинации разрядов КС: флага «групповое/негрупповое» КС (BROADCAST/OWN_ADDRESS*), разряда «Приём-передача» (T/R), поля «Подадрес/Режим управления» (SA[4..0]); и разрешается только в режиме «расширенных возможностей конфигурирования» ОУ (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»).

Использование таблицы занятости подадресов разрешается установлением разряда «разрешение использования таблицы занятости подадресов» (blute, 13-ый разряд регистра cfg2_) в лог. «1».

Таблица формируется/заполнятеся при конфигурировании ОУ. Таблица размещается по фиксированным адресам ОЗУ (от 0240 шестн. до 0247 шестн.).

При работе ОУ формирования значения лог. «1» признака «Абонент занят» в ОС определяется поиском по таблице занятости подадресов для значения принятого достоверного КС данного сообщения с использованием следующего шаблона (значений разрядов КС): флаг «групповое/негрупповое» КС (BROADCAST/OWN_ADDRESS*), разряд «Приём-передача» (T/R), поле разрядов «Подадрес/Режим управления» (SA[4..0]). Флаг «групповое/негрупповое» КС (BROADCAST/OWN_ADDRESS*) имеет значение 1’b1 для групповых сообщений и 1’b0 – для собственного адреса ОУ.

Как показано в таблице 1.654, смещение относительно базового адреса таблицы занятости подадресов (0240 шестн.) формируется для каждой декодированного командного слова на основе его разрядов: флаг «групповое/негрупповое» КС (BROADCAST/OWN_ADDRESS*), разряд «Приём-передача» (T/R), старшего разряда (4-го разряда) поля «Подадрес/Режим управления» (SA[4]).

Для любой возможной комбинации флага «групповое/негрупповое» КС (BROADCAST/OWN_ADDRESS*), разряда «Приём-передача» (T/R) в таблице занятости подадресов имеется пара слов, т.е. 32 разряда, с целью возможности установления признака «Абонент занят» в ответ на команду, обращенную к какому-либо из 32 возможных подадресов. Старший разряд группы подадреса (SA4) используется для выбора конкретного, одного из двух, слова, а значение, задаваемое разрядами подадреса с 3-го по 0-ой (SA3-SA0), предназначено для определения, какой из 16 разрядов в выбранном слове должен быть использован для установления особенного подадреса как «занятого». Лог. «0» в соответствующем «разрядном» местоположении указывает, что признак «Абонент занят» в ОС не будет устанавливаться для этой команды, в то время как лог. «1» указывает на обратное, а именно - признак «Абонент

занят» будет оказываться установленным в ОС ОУ, которое следует за получением КС с данным значением поля разрядов «Подадрес/Режим управления».

Если МСП в состоянии «занято» для КУ ‘Передать векторное слово’, Слово данных (векторное слово) не станет передаваться в «нерасширенном режиме конфигурирования» или, если «игнорирование ‘абонент занят’ для КУ» не установлено (mcob, 13-ый разряд регистра cfg4_, запрограммирован в лог. «0»). Аналогично, если МСП в состоянии «занято» для КУ ‘Синхронизация (с СД)’, слово данных будет сохранено в разделяемой памяти в «нерасширенном режиме конфигурирования» МСП или, если не установлено «запрещение сохранения принимаемых СД в состоянии ‘абонент занят’» (b_rx_td, 3-ий разряд регистра cfg3_ , установлен в лог. «0»).

Если МСП запрограммирована в состояние «расширенных возможностей конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «1») и задано «игнорирование ‘абонент занят’ для КУ» (mcob, 13-ый разряд регистра cfg4_, запрограммирован в лог. «1»), то для КУ ‘Передать слово ВСК ОУ’ слово данных будет передаваться в ЛПИ. Слово данных для КУ ‘Синхронизация (с СД)’ будет сохраняться, даже если МСП в состоянии «занято» и разряд «запрещение сохранения принимаемых СД в состоянии ‘абонент занят’» (b_rx_td, 3-ий разряд регистра cfg3_) установлен в лог. «1».

Если МСП в состоянии «занято» для КУ ‘Передать слово ВСК ОУ’, слово данных (слово ВСК) будет передаваться, если МСП запрограммирована в состояние «нерасширенных возможностей конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «0») или, если разряд «препятствовать передаче слова ВСК при ‘абонент занят’» (ibwib, 14-ый разряд регистра cfg4_) установлен в лог. «0». Если МСП в состоянии «занято» для КУ ‘Передать слово ВСК ОУ’, то данные (слово ВСК) не станут передаваться, если МСП запрограммирована в состояние «расширенных возможностей конфигурирования» (15-ый разряд регистра cfg3_ установлен в лог. «1») работы и, если разряд «препятствовать передаче слова ВСК при ‘абонент занят’» (ibwib, 14-ый разряд регистра cfg4_) установлен в лог. «1».

Таблица 1.654 - Указатель слова в таблице занятости подадресов

Разряд	Состояние
15(СЗР)	0
14	0
13	0
12	0
11	0
10	0
9	1
8	0
7	0
6	1
5	0
4	0
3	0
2	BROADCAST/OWN ADDR*
1	«Приём-передача»
0(МЗР)	SA4

Замечания:

Второй разряд адреса: BROADCAST/OWN ADDR*, содержит лог. «1» для групповых команд, лог. «0» - для негрупповых команд.

Первый разряд адреса: «Приём-передача», имеет высокий уровень для команд передачи, низкий уровень - для команд приема.

Нулевой разряд адреса: SA4, определяется старшим разрядом поля «Подадрес/Режим управления» в принятом командном слове.

Таблица 1.655 - Слово в таблице занятости подадресов для SA4 = 0

Поисковая таблица для определения (задания) значения признака «Абонент занят» (для SA4 = 0)	
Разряд	Описание
15(СЗР)	«Абонент занят» для подадреса 15
14	«Абонент занят» для подадреса 14
13	«Абонент занят» для подадреса 13
12	«Абонент занят» для подадреса 12
11	«Абонент занят» для подадреса 11
10	«Абонент занят» для подадреса 10
9	«Абонент занят» для подадреса 9
8	«Абонент занят» для подадреса 8
7	«Абонент занят» для подадреса 7
6	«Абонент занят» для подадреса 6
5	«Абонент занят» для подадреса 5
4	«Абонент занят» для подадреса 4
3	«Абонент занят» для подадреса 3
2	«Абонент занят» для подадреса 2
1	«Абонент занят» для подадреса 1
0(МЗР)	«Абонент занят» для подадреса 0

Таблица 1.656 - Слово в таблице занятости подадресов для SA4 = 1

Поисковая таблица для определения (задания) значения признака «Абонент занят» (для SA4 = 1)	
Разряд	Описание
15(СЗР)	«Абонент занят» для подадреса 31
14	«Абонент занят» для подадреса 30
13	«Абонент занят» для подадреса 29
12	«Абонент занят» для подадреса 28
11	«Абонент занят» для подадреса 27
10	«Абонент занят» для подадреса 26
9	«Абонент занят» для подадреса 25
8	«Абонент занят» для подадреса 24
7	«Абонент занят» для подадреса 23
6	«Абонент занят» для подадреса 22
5	«Абонент занят» для подадреса 21
4	«Абонент занят» для подадреса 20
3	«Абонент занят» для подадреса 19
2	«Абонент занят» для подадреса 18
1	«Абонент занят» для подадреса 17
0(МЗР)	«Абонент занят» для подадреса 16

1.3.12.2.2.6.11 Входы адреса ОУ

Для того, чтобы МСП откликался на отдельный (негрупповой) адрес ОУ, адрес в принятом командном слове должен равняться значению, подаваемому на входные сигналы msp_rtad[4:0], а лог. величина на входном сигнале msp_rtap должна быть таковой, чтобы в результате суммирования её с числом лог. единиц на msp_rtad[4:0] получалось число с нечётным количеством двоичных единичек. Если групповые сообщения не запрещены (bd, 7-ой разряд регистра cfg5_, запрограммирован в лог. «1»), то способность МСП принимать сообщения по групповому адресу, адрес ОУ 31 (5'b11111), не зависит от значения адреса, установленного на msp_rtad[4:0]. Установленное на соответствующих входных сигналах значение адреса считывается ОУ МСП примерно через 2 мкс после пересечения входным сигналом напряжения

нулевого уровня в середине области разряда «контроля по чётности» поступающего командного слова.

Нечётность контрольной суммы требует присутствия нечётного общего количества логических единиц на входных сигналах msp_rtad[4:0] и msp_rtap. Следовательно, msp_rtap должен быть установленным в лог. «0» для адресов ОУ 1, 2, 4, 7, 8, 11, 13, 14, 16, 19, 21, 22, 25, 26, 28 и 31 (если групповые сообщения запрещены). RTADP должен быть установленным в лог. «1» для адресов ОУ 0, 3, 5, 6, 9, 10, 12, 15, 17, 18, 20, 23, 24, 27, 29 и 30.

В МСП обеспечена жесткая (аппаратная) привязка к адресу ОУ. В полном режиме работы значение «Адрес ОУ» и разряда «контроль по чётности адреса ОУ» могут читаться из регистра cfg5_ (разряды с 5 по 0). «Жёсткая, аппаратная, привязка к адресу ОУ» подразумевает, что состояния на входных сигналах msp_rtad[4:0] и msp_rtap» непрерывно отслеживаются «внутренним регистром адреса ОУ» - разряды rt_a[4..0], rt_a_p регистра cfg5_.

Для МСП выбор «защёлкиваемого адреса ОУ» отсутствует.

1.3.12.2.2.6.12 Ответное слово ОУ

МСП поддерживает два способа формирования значений разрядов/признаков в ОС ОУ:

1) Стандартное ОС. Этот способ используется для обычных применений в соответствии с требованиями ГОСТ Р 52070-2003. В этой конфигурации МСП автоматически обновляет значения признаков «Ошибка в сообщении», «Принята групповая команда», «Принято управление интерфейсом» в соответствие с правилами, определенными ГОСТ Р 52070-2003. Значения признаков «Абонент занят», «Запрос на обслуживание» и «Неисправность абонента» устанавливаются УВ. К тому же, признак «Передача ответного слова» и резервные разряды автоматически устанавливаются в нулевые состояния в ответном ОС ОУ. Существует также возможность выбора - автоматическая установка разряда «Неисправность ОУ» для отклика во время последующего негруппового сообщения, после происшествия самотестирования ОУ с обнаружением ошибки.

2) Режим с альтернативным ОС. В этом режиме все 11 разрядов ОС устанавливаются программно процессором УВ посредством разрядов с 11 по 1 регистра cfg1_. Этот режим может быть употреблен для поддержки применений отличных от ГОСТ Р 52070-2003.

По умолчанию, сразу после установления номинального напряжения питания МСП оказывается в режиме не альтернативного, стандартного, способа формирования ОС. Для МСП этот режим становится выбранным в случае «нерасширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») или, когда aswe, 5-ый разряд регистра cfg3_ («разрешение альтернативного формирования ОС»), установлен в лог. «0».

Для того чтобы ОУ МСП оказался в режиме работы с «альтернативным ОС», должны быть установлены в единичные состояния разряды eme и aswe регистра cfg3_.

Для обоих способов формирования ОС поддержано определение допустимости командных слов.

Для обоих режимов величины, содержащиеся в регистре cfg1_, «выбираются» (считываются) схемой управления протоколом обмена по ЛПИ приблизительно через 4-5 мкс после момента пересечения входным сигналом нулевого уровня в зоне разряда контроля по четности в принимаемом командном слове.

И в том и в другом режиме значение ответного слова ОУ, которое было передано в ответ на последнее обработанное негрупповое сообщение, поступившее по собственному адресу ОУ МСП, может быть прочитано из доступного лишь по чтению регистра ответного слова ОУ (адрес регистра 01110).

Программирование признаков ответного слова ОУ МСП приведено в таблице 1.657.

Таблица 1.657 - Возможности программирования разрядов ОС ОУ

№ разряда	Функция признака	Действие в режиме «стандартного ответного слова»	Действие в режиме «альтернативного ответного слова» (см. замечания 8 и 9)
1-3	Поле синхронизации	Поле синхронизации	Поле синхронизации
4-8	«Адрес ОУ»	«Адрес ОУ»	«Адрес ОУ»
9	«Ошибка в сообщении»	Устанавливается в «1» вследствие обнаружения ошибки формата информационной порции сообщения «приёма» или при определении недопустимости КС в соответствии с ГОСТ Р 52070-2003. См. замечание 1.	Управляется s[10], 11-ым разрядом регистра cfg1_. Кроме того, устанавливается в «1» при определении недопустимости КС в соответствии с ГОСТ Р 52070-2003. См. замечание 10.
10	«Передача ОС»	Лог. «0».	Управляется s[9], 10-ым разрядом регистра cfg1_.
11	«Запрос на обслуживание»	Определяется значением разряда «инверсное значение признака ОС «Запрос на обслуживание»» (sr, 9-ый разряд регистра cfg1_). Может быть автоматически обнулен КУ ‘Передать векторное слово’. См. Замечание 2.	Управляется s[8], 9-ым разрядом регистра cfg1_.
12	Резерв	Лог. «0».	Управляется s[7], 8-ым разрядом регистра cfg1_.
13	Резерв	Лог. «0».	Управляется s[6], 7-ым разрядом регистра cfg1_.
14	Резерв	Лог. «0».	Управляется s[5], 6-ым разрядом регистра cfg1_.
15	«Принята групповая команда»	В соответствии с ГОСТ Р 52070-2003. См. Замечание 3.	Управляется s[4], 5-ым разрядом регистра cfg1_.
16	«Абонент занят»	Определяется значением разряда busy (10-разряд регистра cfg1_) и таблицы занятости подадресов. См. Замечание 4.	Управляется s[3], 4-ым разрядом регистра cfg1_. Может влиять на принятое СД. См. замечание 11.
17	«Неисправность абонента»	Определяется значениями разряда «инверсное значение признака ОС «Неисправность абонента»» (sf, 8-ой разряд регистра cfg1_) или входным сигналом «msp_ssflag_ext_trig». См. замечание 5.	Управляется s[2], разряд 3-им разрядом регистра cfg1_.
18	«Принято управление интерфейсом»	Устанавливается в соответствии с ГОСТ Р 52070-2003, определяется значением разряда dbca, 11-го разряда регистра cfg1_. См. замечание 6.	Управляется s[1], 2-ым разрядом регистра cfg1_.
19	«Неисправность ОУ»	В расширенном режиме конфигурирования (15-ый разряд регистра cfg3_ установлен в лог. «1») определяется разрядом «инверсное значение признака ОС «Неисправность ОУ»» (rtf, 7-ой разряд регистра cfg1_). По желанию, может быть запрограммирован в режим автоматической установки в лог. «1», как результат неудовлетворительной самопроверки ОУ. См. замечание 7.	Управляется s[0], 1-ым разрядом регистра cfg1_. См. Замечание 12.
20	контроль по чётности	Дополнение до нечетного значения слова в соответствии с ГОСТ Р 52070-2003.	Дополнение до нечетного значения слова в соответствии с ГОСТ Р 52070-2003.

Примечание: номера и функции признаков ОС определяются согласно ГОСТ Р 52070-2003

Замечания к таблице 1.657:

1) При ошибке в порции СД сообщения «приём» (ошибки синхронизации, кодирования, количества разрядов, количества слов или контроля по чётности) ОУ МСП не станет отзываться на текущее сообщение. Тем не менее, внутренний триггер МСП «Ошибка в сообщении» будет устанавливаться при этом в лог. «1». Состояние указанного триггера отражается одноименным разрядом в доступном лишь по чтению регистре ответного слова ОУ.

Если МСП принимает КС, которое было определено как недопустимое посредством таблицы допустимости КС, МСП будет отвечать установлением признака «Ошибка в сообщении» в ОС. Присуждение КС статуса недопустимости будет вызывать установку в лог. «1» 6-го разряда, «признак недопустимости командного слова», в слове состояния блока сообщения ОУ. Для недопустимой команды «передача» МСП будет лишь предавать ОС с установленным в «1» признака «Ошибка в сообщении»; передаваться какие-либо данные по ЛПИ при этом не будет. Для достоверного, недопустимого КС «приём» МСП будет сохранять поступающие СД, если МСП не находится в состоянии «расширенных возможностей конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») и не установлено «запрещение сохранения принимаемых СД данных недопустимого сообщения» (i_rx_td, 4-ый разряд регистра cfg3_, установлен в лог. «0»). В противном случае принимаемые СД не станут сохраняться в ОЗУ.

В обоих случаях, если следующим сообщением явится КУ ‘Передать ОС’ или ‘Передать последнюю команду’, то ОУ МСП ответит на следующую команду ОС, с установленным в лог. «1» признаком «Ошибка в сообщении». Для всех других видов КС ОУ МСП будет отвечать нулевым значением признаком «Ошибка в сообщении».

2) Отметим, что установка разряда «инверсное значение признака ОС «Запрос на обслуживание»» (sr) регистра cfg1_ в лог. «0» будет вызывать установление лог. «1» признака «Запрос на обслуживание», в передаваемом ОС. Также отметим, что этот разряд подвержен влиянию со стороны 2-го разряда регистра cfg2_ - «’автосброс’ разряда ‘запрос на обслуживание’» (csr). Если разряд csr находится в состоянии лог. «1», значение признака «Запрос на обслуживание» в ОС будет автоматически ‘сброшено’ после того, как ОУ МСП ответил на КУ ‘Передать векторное слово’. То есть, если разряд csr регистра cfg2_ находится в лог. «1» и разряд sr регистра cfg1_ в лог. «0», то ОУ МСП будет откликаться лог. «1» в признаке «Запрос на обслуживание» ОС до тех пор, пока ему не придется ответить на команду ‘Передать векторное слово’. Для этого сообщения МСП ответит по-прежнему единичным значением признака «Запрос на обслуживание» в ОС. После этого сообщения разряд sr в регистре cfg1_ ‘сбрасывается’ (устанавливается в лог. «1») и остается в таком состоянии до программной переустановки его процессором УВ.

3) В соответствии с ГОСТ Р 52070-2003 значение признака «Принята групповая команда» ОС будет устанавливаться в лог. «1» после принятия достоверного КС группового сообщения. Эта величина будет отражена одноименным разрядом доступного лишь по чтению регистра ответного слова ОУ - rt_sw [15..0]. Если следующим сообщением после этого будет КУ ‘Передать последнюю команду’, ОУ МСП откликнется на нее единичным значением признака «Принята групповая команда» в ОС. Для всех других команд МСП будет откликаться нулевым значением этого признака.

4) Если разряд busy регистра cfg1_ запрограммирован в лог. «0» или МСП принимает КС, по значению множества разрядов КС которого было определено ‘абонент занят’ посредством соответствующей таблицы занятости подадресов, то МСП будет откликаться по ЛПИ ОС, с установленным в нём в лог. «1» признаком «Абонент занят».

Отметим, что программирование разряда busy регистра cfg1_ в лог. «0» будет вызывать установление в единичное значение признака «Абонент занят» в передаваемом ОС.

Во время отклика на КС «передача», для которого было определено состояние ‘абонент занят’ ОУ, МСП будет ограничиваться лишь передачей ОС с установленным в нем признака «Абонент занят»; СД при этом передаваться в ЛПИ не будут.

В обоих случаях, если в следующем сообщении приходит КУ ‘Передать ОС’ или ‘Передать последнюю команду’, ОУ МСП будет отвечать ОС с установленным в лог. «1» признаком «Абонент занят». Для всех других КС МСП будет откликаться значением признака «Абонент занят», определяемым разрядом busy регистра cfg1_ и программированием соответствующего «слова таблицы занятости подадресов» для этого КС (а не для предыдущего).

5) Отметим, что программирование разряда «инверсное значение признака ОС «Неисправность абонента»» (sf) регистра cfg1_ в лог. «0» будет вызывать установление в лог. «1» признака «Неисправность абонента» в передаваемом по ЛПИ ОС.

Данный признак ОС будет оказываться в единичном значении также, если оконечным устройством во время выполнения SOM последовательности, которая производится приблизительно 4 мкс спустя после середины разряда контроля по четности в принимаемом командном слове, была произведена выборка лог. «0» со входа msp_ssflag_ext_trig. Возможным использованием входа msp_ssflag_ext_trig является указание о неисправности процессора УВ, при подсоединении данного входного сигнала к выходу устройства контроля по истечению тайм-аута (watchdog timer) процессора УВ.

Отметим, что значение сигнала на входе msp_ssflag_ext_trig не влияет на состояние разряда «инверсное значение признака ОС «Неисправность абонента»», записываемого в или читаемого из регистра cfg1_. Также следует отметить, что значение сигнала на входе msp_ssflag_ext_trig никак не воздействует на ОУ в режиме «альтернативное ОС».

6) В соответствии с ГОСТ Р 52070-2003 признак «Принято управления интерфейсом» (dbca) в ОС будет устанавливаться в лог. «1» только в случае, если разряд «инверсное значение признака ОС «Принято управление интерфейсом»» регистра cfg1_ (разряд 11) был своевременно запрограммирован в значение лог. «0» и ОУ МСП отвечает на КУ ‘Принять управления интерфейсом’. Для всех других КС признак dbca ОС будет находиться в значении лог. «0». Следует отметить, что МСП не переключает автоматически режим ОУ на режим КШ после получения КУ ‘Принять управления интерфейсом’, признавая за процессором УВ исключительного права на это ответственное действие.

7) В режиме «расширенные возможности конфигурирования» МСП (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») признак «Неисправность ОУ» в передаваемом ОС может быть установленным в лог. «1» посредством программирования разряда «инверсное значение признака ОС «Неисправность ОУ»» (rtf, 7-ой разряд регистра cfg1_) в лог. «0».

Взаимоисключающе, в режиме «расширенные возможности конфигурирования» ОУ МСП может быть запрограммирован на автоматическую передачу лог. «1» в признаке «Неисправность ОУ» ОС после «неудачи» самотестирования, проводимого ОУ МСП ‘в рабочем порядке’ (при обработке сообщения). В этом случае, если результаты самотестирования ‘в рабочем порядке’ не удовлетворительны для отдельного сообщения, то в отклике на последующее негрупповое сообщение признак «Неисправность ОУ» ОС будет установленным в лог. «1». Это становится осуществимым после занесения в rt_f_we, 2-ой разряд регистра cfg3_ («управление формированием признака «Неисправность ОУ» в ОС»), лог. «1».

В режиме ОУ, самотестирование ‘в рабочем порядке’ производится для каждого негруппового сообщения, обрабатываемого МСП. Считается, что самотестирование ‘в рабочем порядке’ проведено с неудовлетворительным (неудачным) исходом в одном из двух (или одновременном наличии) обстоятельств:

- «Ошибка проверочного возврата». Проверка проводится для передаваемой части каждого негруппового сообщения. Оценочная проверка производится после приёма из ЛПИ и декодирования версии каждого передаваемого ОУ слова. В дополнение к тому, для последнего передаваемого слова производится поразрядное сравнение его с принятой версией. Если либо принятая версия какого-либо переданного слова была определена как недостоверная (ошибка

синхронизации, кодирование, количество разрядов или контроля по чётности), либо принятая версия последнего слова не равна исходной, переданной версии, индицируется «ошибка проверочного возврата».

- Истечение разрешенного времени непрерывной работы передатчика. Состояние «истечения времени непрерывной работы передатчика» случается, когда срабатывает сторожевое устройство передатчика МСП, которое жестко, аппаратно, настроено на промежуток в 668 мкс и при своем срабатывании вызывает прекращение передачи по ЛПИ.

8) В режиме «альтернативного ответного слова ОУ» часть «разрядных» местоположений (с 9 по 19) ОС ОУ» (см. ГОСТ Р 52070-2003) управляются напрямую разрядами с 11 по 1 регистра cfg1_, то есть соответствующий разряд ответного слова ОУ устанавливается в лог. «1» программированием соответствующего разряда регистра cfg1_ в лог. «1» (не в лог. «0»).

9) Следующие функции не применяются в режиме «альтернативного ответного слова ОУ»:

а) ‘автосброс’ разряда ‘запрос на обслуживание’ (см. замечание 2);

б) действие признака «Принята групповая команда» (см. замечание 3);

в) входной сигнал msp_ssflag_ext_trig (см. замечание 5);

г) действие признака «Принято управления интерфейсом» (см. замечание 6);

д) воздействие самотестирования ОУ ‘в рабочем порядке‘ на признак 19 («Неисправность ОУ») (см. замечание 7).

10) При ошибке в отработке СД сообщения «прием», которая происходит вслед за получением достоверного командного слова (ошибка синхронизации, манчестерского кодирования, количества слов, количества разрядов, контроля по чётности), ОУ МСП не будет отвечать (отзываться) на текущее сообщение. Тем не менее, внутренний признак «Ошибка в сообщении» установится в лог. «1». Это значение будет отражено в доступном лишь по чтению регистре ответного слова ОУ. Следует отметить, что значение этого признака не будет воздействовать на признак «Ошибка в сообщении» ОС в следующем сообщении.

Если МСП принимает командное слово, которое было определено как недопустимое посредством таблицы допустимости КС, то МСП будет отвечать ОС с установленным в лог. «1» признаком «Ошибка в сообщении». Недопустимая команда будет вызывать установку в лог. «1» разряда «признак недопустимости командного слова» (6-ой разряд слова состояния блока сообщения ОУ).

Для недопустимого КС «передача» МСП будет передавать лишь ОС с установленным в нем признаком «Ошибка в сообщении». Передача данных производиться не будет.

Для достоверного недопустимого КС «приём» МСП будет сохранять в ОЗУ полученные СД, кроме случая, когда МСП находится в режиме «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1») и разряд «запрещение сохранения принимаемых СД данных недопустимого сообщения» (4-ый разряд регистра cfg3_) установлен в лог. «1»..

11) Если s[3], 4-ый разряд регистра cfg1_, установлен в лог. «1», МСП будет отвечать на команду «передача» ОС с лог. «1» признака «Абонент занят» (согласно разрядной сетке ГОСТ Р 52070-2003), предваряющем пересылку требуемого количества слов данных.

Если s[3] находится в лог. «1» и «Запрещение сохранения принимаемых СД в состоянии ‘абонент занят’» (b_rx_td, 3-ий разряд регистра cfg3_) установлен в лог. «0», поступающие слова данных будут сохраняться в ОЗУ, разделяемом с МСП. Если оба разряда (s[3] и «Запрещение сохранения принимаемых СД в состоянии ‘абонент занят’ на команды приема») запрограммированы в лог. «1», поступающие на вход ОУ слова данных не сохраняются в ОЗУ.

В режиме «альтернативное ОС» таблица занятости подадресов не задействована.

12) В режиме «альтернативное ОС», обнаружение «ошибки проверочного возврата», а также истечения времени непрерывной работы передатчика (668 мкс) не воздействуют на значение признака «Неисправность ОУ» ОС, какой бы не была логическая величина разряда rt_f_we (2-ой разряд регистра cfg3_ - «управление формированием признака «Неисправность ОУ» в ОС»).

1.3.12.2.2.6.12.1 Интервал ожидания ОС в передаче типа «ОУ-ОУ»

Интервал ожидания ОС МСП применяется для режима ОУ, когда ОУ МСП является принимающим ОУ из устройств, участвующих в передаче типа «ОУ-ОУ»; для определения возможного случая отсутствия отклика или чрезмерной задержки отклика со стороны передающего ОУ используется сторожевое устройство, отслеживающее истечение определенного промежутка времени. Величина интервала ожидания ОС соответствует времени, которое измеряется от середины разряда контроля по четности (момент пересечения входным сигналом нулевого уровня в зоне разряда контроля по чётности) командного слова «передача» до середины синхроимпульса ОС, посылаемого передающим ОУ. В «нерасширенном режиме конфигурирования» наибольшая разрешённая величина этого промежутка по умолчанию принимается равной 18 мкс. Для возможности приспособиться при необходимости к условиям передачи по ‘длинным’ линиям различной протяжённости, в МСП введена дополнительная переменная, ограничивающая интервал ожидания ОС. Величина этой переменной может устанавливаться процессором УВ. В режиме «расширенные возможности конфигурирования» работы (eme, 15-ый разряд регистра cfg3_, уст. в лог. «1») значение интервала ожидания ОС может быть задано посредством регистра cfg5_, как это показано в таблице 1.658.

Таблица 1.658

Выбор интервала ожидания ОС		Величина времени
Разряд 10	Разряд 9	ожидания отклика ОУ (мкс)
0	0	18,5
0	1	22,5
1	0	50,5
1	1	130

1.3.12.2.2.6.13 Слово ВСК (rt_bit_w)

ОУ МСП предлагает на выбор к употреблению два различных, по способу формирования, слова встроенной системы контроля (ВСК): внутреннее ВСК и внешнее ВСК.

При использовании внутреннего слова ВСК МСП отвечает на КУ ‘Передать слово ВСК ОУ’ передачей содержимого регистра слова ВСК, предварив её посылкой ОС. В случае использования внешнего слова ВСК МСП отвечает «словом ВСК», которое хранится в ОЗУ МСП. Значение этого слова полностью определяется программным установлением со стороны УВ.

Следует отметить, что в режиме ОУ «нерасширенные возможности конфигурирования» УВ может читать значение внутреннего «слова ВСК» посредством доступного лишь по чтению регистра слова ВСК, с адресом 0F(шестн.). Этот регистр может быть прочитан вне зависимости от текущей настройки МСП - на работу с внутренним ВСК, или на работу с внешним ВСК.

В режиме «нерасширенные возможности конфигурирования» или, если ebwe, 15-ый разряд регистра cfg4_ - «разрешение внешнего слова ВСК», запрограммирован в лог. «0», передаваемое «слово ВСК» берется из регистра слова ВСК.

Если МСП в режиме «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, уст. в лог. «1») и, если разряд ebwe регистра cfg4_ уст. в лог. «1» , то «слово ВСК» берется из местоположения в ОЗУ:

1) Если emch, 0-ой разряд регистра cfg3_ - «расширенная обработка КУ», установлен в лог. «0», то «слово ВСК» будет читаться из места, на которое указывает содержимое ячейки таблицы указателей для подадресов на «передачу» активной области ОЗУ при КС «передача» для подадресов 0 и 31. Для этой конфигурации МСП следует отметить, что слово данных в отзыве ОУ на КУ ‘Передать векторное слово’ будет читаться из той же самой пары адресных местоположений, в которой сохраняется внешнее слово ВСК.

2) Если оба разряда ebwe и emch установлены в логические единицы, слово ВСК будет читаться из фиксированного местоположения в разделяемом ОЗУ по адресу 0123.

Разрядное распределение и описание для слова ВСК (rt_bit_w[15..0]) приведены в таблице 1.659.

Таблица 1.659 - Разрядное распределение и описание для слова ВСК

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tt	ltfb	ltfa	hf	tsa	tsb	tfi	c_b_a	hwc	lwc	inc_syn_rec	inv_dw	rtg_s_ae	rt_nre	rt_2cwe	cwce

Замечание - Разряды с 15 по 9 очищаются только активным (низким) уровнем входного сигнала prst_n*, программным сбросом с помощью регистра запуска/сброса или КУ ‘Установить ОУ в исходное состояние’. Разряды с 8 по 0 обновляются в результате обработки каждого сообщения.

ltfb, ltfa (Loop Test failure B, Loop Test failure A): ошибка проверочного возврата;

hf (Handshake Failure): истечение времени ожидания подтверждения (квитирования);

tsa, tsb(Transmitter Shutdown A, Transmitter Shutdown B): блокировка передатчика канала А/В КУ ‘Блокировать передатчик’;

tfi (Terminal Flag Inhibited): индикатор КУ ‘Блокировать признак неисправность ОУ’;

c_b_a (Channel B/A): индикатор канала сообщения А/В;

hwc (Hight Word Count): превышение количества переданных или принятых СД;

lwc (Low Word Count): заниженное количество переданных или принятых СД;

inc_syn_rec (Incorrect Sync Type Received): синхроимпульс КС в СД;

inv_dw (invalid data word): обнаружено недостоверное слово в сообщении;

rtg_s_ae (RT-RT Gap/Sync/Address Error): ошибка в передаче ОУ-ОУ: несоблюдение минимальной паузы перед выдачей ОС/ синхросигнала или формата / адрес ОУ;

rt_nre (RT-RT No Response Error): ошибка в передаче ОУ-ОУ»: интервала ожидания ОС;

rt_2cwe (RT-RT 2nd Command Word Error): индикатор ошибки 2-ого командного слова;

cwce (Command Word Contents Error): принятое КС не согласуется с требованиями ГОСТ Р 52070-2003.

tt (Transmitter Timeout), – разряд указатель затягивания передачи на время, большее, чем 668 мкс: Установлен, если сторожевым устройством МСП были обнаружены ошибочные обстоятельства передачи, а именно затягивание передачи. Схема отслеживания истечения промежутка времени, выделенного для работы передатчика, будет автоматически отключать работу передатчика канала А или канала В ЛПИ в случае, если непрерывная передача затягивается на время, большее, чем 668 мкс. В режиме ОУ МСП будет завершать обработку текущего сообщения в результате срабатывания сторожевого устройства, тем не менее она будет отвечать (реагировать) на следующее принятое сообщение.

ltfb, ltfa, (Loop Test failure B, Loop Test failure A), – разряд ошибка проверочного возврата: проводится на передаваемой части каждого негруппового сообщения. Принятая версия каждого переданного МСП слова проверяется на достоверность. Вдобавок последнее слово, передаваемое ОУ в каждом сообщении, проходит через поразрядное сравнение. Если принятая версия какого-либо переданного слова определена как недостоверная (обнаружена ошибка синхронизации, кодирования, количества разрядов, контроля по чётности) и/или принятая версия переданного последним слова не равна исходной, то есть переданной версии, или обнаружено затягивание времени передачи слова для соответствующего канала, разряд ltfa (ltfb) для соответствующего канала будет установлен.

Так же разряд ltfa (ltfb) установится, если во время выдачи слов данных оконечным устройством поступит недостоверное слово (слова) по альтернативному каналу В (А).

hf (Handshake Failure), – разряд истечение времени ожидания подтверждения (квитирования):

Данный разряд не задействован в МСП.

tsa, tsb (Transmitter Shutdown A, Transmitter Shutdown B), – разряд блокировка передатчика канала А/В КУ ‘Блокировать передатчик’: Указывает, что передатчик на соответствующем канале обмена был выключен КУ ‘Блокировать передатчик’, принятой по другому каналу. При получении по другому каналу ЛПИ КУ ‘Разблокировать передатчик’, логическая величина разряда возвращается к прежнему состоянию.

tfi (Terminal Flag Inhibited), – разряд индикатор КУ ‘Блокировать признак неисправность ОУ’: Установлен в лог. «1», если признак «Неисправность ОУ» был запрещен с помощью КУ ‘Блокировать признак неисправность ОУ’. Будет возвращен в прежнее логическое состояние (лог. «0») после принятия КУ ‘Разблокировать признак неисправность ОУ’ или ‘Установить ОУ в исходное состояние’.

c_b_a (Channel B/A), – разряд индикатор канала сообщения А/В: Лог. «0», если предыдущее сообщение было принято по каналу А, лог. «1», если - по каналу В.

hwc (Hight Word Count), – разряд превышение количества переданных или принятых СД: Установлен в лог. «1», если в предыдущем сообщении было обнаружено «превышение количества переданных или принятых СД».

lwc (Low Word Count), – разряд заниженное количество переданных или принятых СД: Установлен в лог. «1», если в предыдущем сообщении было обнаружено «заниженное количество переданных или принятых СД».

inc_syn_rec (Incorrect Sync Type Received), – разряд синхроимпульс КС в СД: Если установлен, указывает, что МСП обнаружил синхроимпульс командного слова во время приёма слова данных.

inv_dw (invalid data word), – разряд обнаружено недостоверное слово в сообщении: Указывает, что ОУ МСП принято слово данных, которое содержит одну или более ошибку из следующих видов: ошибка типа синхросигнала, ошибка кодирования, контроля по чётности, и/или количества разрядов.

rtg_s_ae (RT-RT Gap/Sync/Address Error), – разряд ошибка в передаче ОУ-ОУ: несоблюдение минимальной паузы перед выдачей ОС/ синхросигнала или формата/ адрес ОУ: Этот разряд устанавливается в лог. «1», если ОУ МСП является принимающим ОУ в пересылке типа «ОУ-ОУ» и происходит хотя бы одно событие из следующего списка:

(1) Если имеется «разрешение проверки минимальной паузы перед выдачей ОС» (gce, 8-ой разряд регистра cfg5_, установлен в лог. «1») и передающий ОУ отвечает ОС через время меньшее, чем 4 мкс (согласно ГОСТ Р 52070-2003 - это промежуток между серединами зон разряда контроля по четности последнего КС/СД и синхроимпульса ОС), т.е. реально проверяется, что период отсутствия декодируемых сигналов в ЛПИ (‘мёртвое время’) меньше 2 мкс;

(2) обнаружен неправильный тип синхросигнала или ошибка формата (кодирование, количество разрядов, и/или ошибка контроля по чётности) в ОС передающего ОУ;

(3) значение, заключенное в поле адреса в ОС передающего ОУ, не равно значению поля адреса ОУ в КС «передача».

rt_nre (RT-RT No Response Error), – разряд ошибка в передаче ОУ-ОУ»: интервала ожидания ОС: Если лог. «1», то указывает, что в предыдущем сообщении МСП выступал в роле принимающего ОУ, и что передающее ОУ либо не ответило вообще, либо отозвалось позднее допустимого для пересылки типа «ОУ-ОУ» интервала ожидания ОС со стороны передающего ОУ.

«Интервал ожидания ОС» определено как временной промежуток от момента перехода через нуль сигналом в середине области разряда контроля по четности командного слова «передача», до момента перехода через нуль сигналом, который поступает на вход приёмника в

середине области синхросигнала ОС ОУ. В режиме «нерасширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») величина «интервала ожидания ОС» равняется 18,5 мкс (контролируется с точностью ±1 мкс). Если разряд «разрешение расширенных возможностей конфигурирования» установлен в лог. «1», значение интервала ожидания ОС может выбираться из следующего перечня величин: 18,5; 22,5; 50,5 и 130 мкс (±1 мкс); выбор определяется посредством разрядов 10 и 9 регистра cfg5_.

rt_2cwe (RT-RT 2nd Command Word Error), – разряд индикатор ошибки 2-ого командного слова: Ошибка второго командного слова в сообщении типа «ОУ-ОУ»: Если МСП является принимающим ОУ в сообщении типа «ОУ-ОУ», то установленный в лог. «1», данный разряд указывает на одно или более ошибочное состояние командного слова «передача»:

разряд «Прием-передача» имеет состояние лог. «0»;

значение подадресного поля равно нулю, либо 1F (шестн.);

значение «Адреса ОУ» то же самое, что и в командном слове «приём».

cwce (Command Word Contents Error), – разряд принятое КС не согласуется с требованиями ГОСТ Р 52070-2003: Указывает, что принятое командное слово определено как несоответствующее ГОСТ Р 52070-2003. Это происходит в следующих обстоятельствах и для следующих запрещенных стандартом командных слов:

«запрет групповых сообщений» (bd, 7-ой разряд регистра cfg 5_ установлен в лог. «0») и командное слово не является КС КУ, групповым и определяющим передачу данных по ЛПИ;

разряд «обрабатывать КУ на «прием» без СД как резервные КУ» (ome, 6-ой разряд регистра cfg3_) установлен в лог. «0» и значение разряда «Прием-передача» в командном слове равно нулю, значение поля «Подадрес/Режим управления» равно «5’b00000», либо «5’b11111» и значения поля разрядов КС «Код команды» принадлежат области значений от «0» до «F»;

«запрет групповых сообщений» (bd), 7-ой разряд регистра cfg5_, установлен в лог. «0» и по групповому адресу (1F) принято КС КУ, для которого запрещено использовать групповые обмены.

1.3.12.2.2.6.14 SOM и EOM последовательности в режиме ОУ

Приблизительно через 1,25 мкс после середины области разряда контроля по четности в принимаемом командном слове (момент пресечения входным сигналом нулевого уровня), МСП начинает вырабатывать последовательность действий «старт отработки сообщения ОУ» - SOM (Start-of-Message). Примерно через 6 мкс после окончания последнего, передаваемого или принимаемого ОУ слова (или принимаемого, для группового сообщения), МСП начинает выполнять последовательность действий «завершение отработки сообщения» EOM (End-of-Message). SOM и EOM последовательности ОУ содержат несколько процедур чтения и записи из / в разделяемое с УВ ОЗУ. Содержание и порядок выполнения действий SOM и EOM последовательностей представлены ниже.

1.3.12.2.2.6.14.1 SOM (Start-of-Message) последовательность действий для оконечного устройства

SOM (Start-of-Message) последовательность действий для оконечного устройств состоит из следующих шагов:

1) Если проверка допустимости КС включена (id, 7-ый разряд регистра cfg3_ установлен в лог. «0»), соответствующее слово читается из таблицы допустимости КС.

2) Если разряд icoe (13-ый разряд регистра cfg6_) равен лог. «1» и произошло вытеснение команды, то значение указателя на стек команд, увеличивается на четыре и записывается в местоположение указателя на стек команд активной области А/Б.

3) Указатель на стек команд активной области А/Б читается для использовании при доступе к описателю блока сообщения в стеке команд.

4) Если используется таблица занятости подадресов (blute, 13-ый разряд регистра cfg2_ установлен в лог. «1»), то соответствующее слово читается из таблицы занятости подадресов.

5) Если «расширенное управление памятью ОУ» включено (ermm, 1-ый разряд регистра cfg2_, запрограммирован в лог. «1»), то слово управления подадресом ОУ, определяющее тип буферизации СД, считывается из области поисковой таблицы для подадресов ОУ.

6) Значение адреса буфера данных (указатель блок данных) считывается из соответствующего местоположения в таблице указателей для подадресов на «прием»/«групповой приём»/«передачу», за исключением КУ без слов данных или, когда разрешена «расширенная обработка КУ».

7) Командное слово записывается в четвертое местоположение описателя блока сообщения;

8) Значение адреса (указатель блока данных) записывается в третье местоположение в описателе блока сообщения, за исключением КУ без слов данных или, когда разрешена «расширенная обработка КУ».

9) Значение регистра счетчика метки времени записывается во вторую позицию описателя блока сообщения;

10) Слово состояния блока сообщения записывается в первое местоположение описателя блока сообщения;

11) Если разряд icoe (13-ый разряд регистра cfg6_) равен лог. «0» , то значение указателя на стек команд, считанное на третьем шаге, увеличивается на четыре и записывается в местоположение указателя на стек команд активной области А/Б.

Если разряд icoe (13-ый разряд регистра cfg6_) равен лог. «1» , то данный шаг не выполняется, и увеличение значения указателя на стек команд на четыре происходит не при SOM, а при EOM последовательности (шаг 5 EOM-последовательности).

1.3.12.2.2.6.14.2 EOM (End-of-Message) последовательность действий для оконечного устройства

EOM (End-of-Message) последовательность действий для оконечного устройства состоит из следующих шагов:

1) Если для текущего сообщения используется «двойная буферизация сообщений «приема»», указатель блок данных считывается из таблицы указателей для подадресов; или, если сообщение - КУ и «расширенная обработка КУ» разрешена, то переданное или принятое слово данных для КУ будет записываться в третье местоположение описателя блока сообщения. Если «расширенная обработка КУ» разрешена, то для «КУ без данных» слово данных не будет сохраняться в описателе блока сообщения ОУ.

2) Если для текущего сообщения используется «кольцевая буферизация» или «двойная буферизация сообщений «приема»» и указатель на буфер данных нуждается в обновлении, новое значение указателя записывается в таблицу указателей для подадресов.

3) Регистр счетчика метки времени сохраняется во втором местоположении в описателе блока сообщения;

4) Слово состояния блока сообщения записывается в первое местоположение в описателе блока сообщения.

5) Если разряд icoe (13-ый разряд регистра cfg6_) равен лог. «1», то значение указателя на стек команд, считанное на третьем шаге SOM-последовательности, увеличивается на четыре и записывается в местоположение указателя на стек команд активной области А/Б.

1.3.12.2.2.6.15 Перечень исключительных (особых) ситуаций для режима ОУ

В своих откликах на различные ошибки и прочие исключительные события в режиме ОУ МСП производит определённые действия и оповещает о случившемся многочисленными индикаторами (флажками). Таблица 1. показывает перечень всевозможных для ОУ МСП исключительных ситуаций и соответственные отклики на них, их воздействия на признаки (разряды) ОС, слова состояния блока сообщения МСП, а также их влияние на разряды регистра состояния прерывания и на формирование запросов прерывания.

Следует отметить, что отклики на ошибочные обстоятельства для «сообщений режима управления» описаны отдельно в таблице 1.660.

Таблица 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполага-ется «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разряды регистра состояния прерывания (если разрешено)
Недостоверное командное слово	Нет ОС (информация игнорируется)	Нет изменений; последователь-ности SOM и EOM не выполняются	Нет изменений	Нет изменений
Ошибка контроля по четности собственного адреса ОУ	ОУ не будет принимать слова или откликаться на сообщение, отправленное по его собственному адресу. ОУ будет принимать слова, посылаемые по групповому адресу (31) , если только 7 разряд регистра cfg5_ не установлен в лог. «1». Для группового сообщения признак «Принята групповая команда» устанавливается в лог. «1» в регистре ответного слова ОУ.	(см. примечание 1)	Нет изменений	rt_ape
Тип синхросигнала КС/ОС в слове данных, следующем за принятым достоверным КС.	Нет ОС. Признак «Ошибка в сообщении» регистра ответного слова ОУ устанавливается в лог. «1». Для группового сообщения признак «Принята групповая команда» устанавливается в лог. «1» в регистре ответного слова ОУ.	eom, ef, fe, ids, iw	inc_syn_rec	eom; fe; scw_e или ss_mc_pt (если выбрано)
За принятием достоверного командного слова следует принятие недостоверного (ных) СД (кодирование, количество разрядов, контроль по чётности)	Нет ОС. Разряд «Ошибка в сообщении» регистра ответного слова ОУ устанавливается в лог. «1». Для группового сообщения признак «Принята групповая команда» устанавливается в лог. «1» в регистре ответного слова ОУ.	eom, ef, fe, iw	inv_dw	eom; fe; scw_e или ss_mc_pt (если выбрано)
За достоверным командным словом следует ложное число СД	Нет ОС. Разряд «Ошибка в сообщении» регистра ответного слова ОУ устанавливается в лог. «1». Для группового сообщения признак «Принята групповая команда» устанавливается в лог. «1» в регистре ответного слова ОУ.	eom, ef, fe, wce	hwc или lwc	eom; fe; scw_e или ss_mc_pt (если выбрано)
МСП является принимающим ОУ в безошибочной пересылке типа «ОУ-ОУ»	Ответ «нормальным» ОС. Для группового сообщения признак «Принята групповая команда» устанавливается в лог. «1» в регистре ответного слова ОУ.	eom, rt_rt_f	Нет изменений	eom; scw_e или ss_mc_pt (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполагается «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разряды регистра состояния прерывания (если разрешено)
Истечение интервала ожидания ОС при передаче типа ОУ-ОУ: МСП является принимающим ОУ в сообщении типа ОУ-ОУ и передающий ОУ не ответил в положенный срок	Нет ОС. Разряд «Ошибка в сообщении» регистра ответного слова ОУ устанавливается в лог. «1». Для группового сообщения признак «Принята групповая команда» устанавливается в лог. «1» в регистре ответного слова ОУ.	eom, ef, rt_rt_f, rto	rt_nre	eom; fe; scw_e или ss_mc_pt (если выбрано)
Режим «нерасширенные возможности конфигурирования» МСП или «Расширенная обработка КУ» запрещена, при приеме достоверной и допустимой КУ (включая резервные коды)	В случае команды ‘Передать ОС’ или ‘Передать последнюю команду’ - ОС, проистекающее от предыдущего сообщения, становится переданным в ЛПИ, и регистр ответного слова ОУ не обновляется. Иначе, в зависимости от обуславливающих обстоятельств могут быть задействованы следующие признаки ОС ОУ: «Ошибка в сообщении» (если недопустимое сообщение), «Абонент занят» (под воздействием регистра cfg1_ и таблицы занятости подадресов), «Принято управление интерфейсом», «Неисправность ОУ». Для группового сообщения признак «Принята групповая команда» устанавливается в лог. «1» в регистре ответного слова ОУ. Указатель на блок данных для КС с полем «Подадрес/Режим управления» 0 или 31, прочитанный из таблицы указателей для подадресов сохраняется в третьей ячейке описателя блока сообщения. Для КУ ‘Передать последнюю команду’ или ‘Передать слово ВСК ОУ’ (если не включено использование «внешнего слова ВСК»), передаваемое слово данных читается из регистра слова ВСК. Для других КУ со словами данных, СД записываются/считываются в/из местоположение, определяемое указателями таблицы указателей для подадресов на «прием» для соответствующего КС «приём» с полем «Подадрес/Режим управления» от 0 до 31.	eom; если недопустимая команда: icw	Какое-либо из следующих установок или сбросов, в зависимости от конкретных обстоятельств: tfi, tsa, tsb	eom scw_e (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполага-ется «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разряды регистра состояния прерывания (если разрешено)
Режим «расширенные возможности конфигурирования» МСП и «Расширенная обработка КУ» разрешена; принимается достоверное сообщение, определенное как КУ (включая резервные коды).	В случае команды ‘Передать ОС’ или ‘Передать последнюю команду’, - ОС, проистекающее от предыдущего сообщения, становится переданным в ЛПИ и регистр ответного слова ОУ не обновляется. Иначе, может повлиять на следующие признаки ОС в зависимости от обуславливающих обстоятельств: «Ошибка в сообщении» (если недопустимое сообщение), «Абонент занят» (под воздействием регистра cfg1_ и таблицы занятости подадресов), «Принято управление интерфейсом», «Неисправность ОУ». Для КУ ‘Передать последнюю команду’ или ‘Передать слово ВСК ОУ’ (если не включено использование «внешнего слова ВСК»), передаваемое слово данных читается из регистра слова ВСК. Для других КУ с данными, слова извлекаются из/ либо записываются в соответствующее слово таблицы слов данных КУ. Вдобавок, СД, принимаемое или передаваемое, будет сохраняться в третью позицию описателя блока сообщения.	eom; если недопустимая команда: icw	В зависимости от ситуации может произойти что-либо из следующего списка установок и сбросов: tfi, tsa, tsb	eom; ss_mc_pt (если выбрано для определенного КУ посред-ством таблицы прерываний для избранных КУ - 0108-010F)
«Ошибка проверочного возврата»: принятая версия переданного слова была определена как недостоверная (кодирование, количество разрядов, контроль по чётности) и/или принятая версия последним переданного слова оказалась отличной по содержанию от переданной версии.	Если режим «расширенных возможностей конфигурирования» МСП включен и 2 разряд регистра cfg3_, rt_f_we, установлен в лог. «1», то признак «Неисправность ОУ» ОС ОУ будет установленным в лог. «1» при отклике на следующее негрупповое сообщение.	eom, ef, ltf	ltfa или ltfb	eom; fe; scw_e или ss_mc_pt (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполагается «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разряды регистра состояния прерывания (если разрешено)
Ошибка «истечение времени непрерывной работы передатчика (668 мкс)».	Нет ОС. Если режим «расширенные возможности конфигурирования» МСП включен и 2 разряд регистра cfg3_, rt_f_we, установлен в лог. «1», то признак «Неисправность ОУ» ОУ будет установленным в лог. «1» в ОС на следующее негрупповое сообщение. Истечение времени непрерывной работы передатчика не влияет на отклик ОУ в последующем сообщении.	eom	tt	eom; scw_e или ss_mc_pt (если выбрано)
Ошибка подтверждения/ квитирования (своевременного получения отклика на запрос обладания системной магистралью УВ): Для последовательности «старт отработки сообщения» SOM или передачи данных (чтение/ запись). Заметьте, что ошибка квитирования не может произойти для EOM.	Если ошибка подтверждения (квитирования) случилась во время SOM последовательности действий, МСП прекращает обработку сообщения: сообщение будет проигнорировано, и, следовательно, не будет итоговых результатов в описателе блока сообщения. Сохранение полученного КС и/или СД будет прекращено в результате ошибки квитирования.	Если ошибка квитирования случилась во время SOM, то не будет установленных разрядов, т.е. сообщение не будет обработано (проигнориро-вано). Если ошибка квити-рования проис-ходит во время последовательности пересыл-ки СД, то: eom, ef.	hf	hf
Режим «расширенные возможности конфигурирования» установлен и получено достоверное командное слово «приём», которое было определено перед этим как недопустимое посредством таблицы допустимости КС.	ОУ откликается ОС с установленным признаком «Ошибка в сообщении». Признак «Ошибка в сообщении» в регистре ответного слова ОУ устанавливается в лог. «1». Если сообщение оказалось при этом и групповым, то в регистре ответного слова ОУ устанавливается в лог. «1» признак «Принята групповая команда». Если «запрещение сохранения принимаемых СД данных недопустимого сообщения» не установлено (4 разряд регистра cfg3_ установлен в лог. «0»), то принимаемые СД сохраняются в ОЗУ.Если «запрещение сохранения принимаемых СД данных недопустимого сообщения» установлено (4 разряд регистра cfg3_ установлен в лог. «1»), то принимаемые слова не сохраняются в ОЗУ.	eom, icw, fe	Нет изменений	eom, fe; scw_e или ss_mc_pt (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполага-ется «расширенные возможности конфигурирова ния»)	Разряды слова ВСК	События запроса прерывания УВ, разря-ды регистра состояния прерывания (если разрешено)
Режим «расширенные возможности конфигурирования» установлен и достоверное командное слово «передача», которое было определено перед этим как недопустимое посредством таблицы допустимости КС.	ОУ откликается ОС с установленным в лог «1» признаком «Ошибка в сообщении». Слова данных не передаются. Признак «Ошибка в сообщении» в регистре ответного слова ОУ устанавливается в лог. «1». Если сообщение оказалось при этом и групповым, то в регистре ответного слова ОУ устанавливается в лог. «1» признак «Принята групповая команда».	eom, icw	Нет изменений	eom; scw_e или ss_mc_pt (если выбрано)
Режим «расширенные возможности конфигурирования» установлен и ОУ в состоянии «занято» для достоверной команды «приём» (или глобально, или для конкретного достоверного сообщения «приёма»).	ОУ откликается ОС с установлением в лог «1» признаком «Абонент занят». Признак «Абонент занят» в регистре ответного слова ОУ устанавливается в лог. «1». Признак «Ошибка в сообщении» в регистре ответного слова ОУ устанавливается в лог. «1». Если сообщение оказалось при этом и групповым, то в регистре ответного слова ОУ устанавливается в лог. «1» признак «Принята групповая команда». Если разряд «запрещение сохранения принимаемых СД в состоянии ‘абонент занят’» (3 разряд регистра cfg3_) установлен в лог. «0», то принимаемые слова сохраняются в ОЗУ. Если же этот разряд установлен в лог. «1», то принимаемые данные не сохраняются в разделяемом ОЗУ.	eom, ef, fe	Нет изменений	fe; eom; scw_e или ss_mc_pt (если выбрано)
Режим «расширенные возможности конфигурирования» установлен и ОУ в состоянии «занято» для достоверной команды «передача» (или глобально, или для конкретного достоверного сообщения «передача»).	ОУ откликается ОС с установлением в лог «1» признаком «Абонент занят». Признак «Абонент занят» в регистре ответного слова ОУ устанавливается в лог. «1». Данные не передаются. Если сообщение оказалось при этом и групповым, то в регистре ответного слова ОУ устанавливается в лог. «1» признак «Принята групповая команда».	eom, icw	Нет изменений	eom ; scw_e или ss_mc_pt (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполага-ется «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разря-ды регистра состояния прерывания (если разрешено)
Режим «расширенные возможности конфигурирования» установлен, МСП является принимающим ОУ в пересылке типа «ОУ-ОУ» при этом: 1) разряд gce - «разрешение проверки минимальной паузы перед выдачей ОС», gce - 8-ой разряд регистра cfg5_ запрограммирован в лог. «1» и время минимальной паузы перед выдачей ОС передающим ОУ меньше, чем 2 мкс,) или 2) Некорректный тип синхроимпульса или ошибка формата (кодирование, количество слов и/или контроль по чётности) в ОС передающего ОУ, или 3) Адрес ОУ в переданном ОС отличен от адреса ОУ командного слова «передача».	Нет ОС. Признак «Ошибка в сообщении» регистра ответного слова ОУ устанавливается в лог. «1». Если сообщение оказалось при этом и групповым, то в регистре ответного слова ОУ устанавливается в лог. «1» признак «Принята групповая команда».	eom, ef, rt_rt_f, fe, rtg_s_ae	rtg_s_ae	fe; eom; scw_e или ss_mc_pt (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполагается «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разряды регистра состояния прерывания (если разрешено)
Режим «расширенные возможности конфигурирования» установлен, и МСП является принимающим ОУ в пересылке типа «ОУ-ОУ». Обнаружена ошибка в командном слове для передающего ОУ: разряд КС «Приём-передача» равен лог. «0», или поле «Подадрес/Режим управления» равно 5'b00000, или 5'b11111, или поле «Адрес ОУ» имеет такое же как в командном слове для принимающего ОУ	Нет ОС. Признак «Ошибка в сообщении» регистра ответного слова ОУ устанавливается в лог. «1». Если сообщение оказалось при этом и групповым, то в регистре ответного слова ОУ устанавливается в лог. «1» признак «Принята групповая команда».	eom, ef, rt_rt_f, fe, rt_2cwe	rt_2cwe	fe; eom; scw_e или ss_mc_pt (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполага-ется «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разря-ды регистра состояния прерывания (если разрешено)
Режим «расширенные возможности конфигурирования» установлен, и МСП является принимающим ОУ в пересылке типа «ОУ-ОУ» при этом; 1) Обнаружена ошибка в ОС передающего ОУ: (синхронизации, кодирования, количества разрядов, контроля по чётности, количества слов). или 2) ОС от передающего ОУ содержит значение лог. «1» хотя бы в одном из признаков «Ошибка в сообщении», «Абонент занят» и, вслед за данным ОС, отсутствуют СД.	Нет ОС. Признак «Ошибка в сообщении» регистра ответного слова ОУ устанавливается в лог. «1». Если сообщение оказалось при этом и групповым, то в регистре ответного слова ОУ устанавливается в лог. «1» признак «Принята групповая команда».	eom, ef, rt_rt_f, fe, и одно из следующего: wce или ids или iw.	Одно из следующего (в зависимости от ситуации): hwc или lwc или inc_syn_rec или inv_dw.	fe; eom; scw_e или ss_mc_pt (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполагается «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разряды регистра состояния прерывания (если разрешено)
Режим «расширенные возможности конфигурирования» установлен, обнаружена ошибка в КС: 1) разряд bd, «запрет групповых сообщений» (регистр cfg5_) установлен в лог. «0» и получено КС некорректно задающее сообщение -‘групповое, передача, не КУ’ или 2) разряд bd, «запрет групповых сообщений» (регистр cfg5_) установлен в лог. «0» и получено КС некорректно задающее сообщение -‘групповое’ для КУ, которой не разрешено быть групповой или 3) разряд ome, «Обрабатывать КУ на «прием» без СД как резервные КУ» (регистр cfg3_) запрограммирован в лог. «0» и командное слово имеет нулевое значение разряда «Приём-передача», поле разрядов «Подадрес/ Режим управления» равным 5’b00000 или 5’b11111, и поле разрядов КС «Число СД/Код команды» имеет значение между 5’b00000 и 5’b 01111.	Нет ОС. Признак «Ошибка в сообщении» регистра ответного слова ОУ устанавливается в лог. «1». Если сообщение оказалось при этом и групповым, то в регистре ответного слова ОУ устанавливается в лог. «1» признак «Принята групповая команда».	eom, ef, rt_rt_f, fe, cwce	cwce	fe; eom; scw_e или ss_mc_pt (если выбрано)

Продолжение таблицы 1.660

Исключительные обстоятельства	Ответное слово ОУ, разряды регистра ответного слова ОУ	Разряды слова состояния блока сообщения ОУ (предполагается «расширенные возможности конфигурирования»)	Разряды слова ВСК	События запроса прерывания УВ, разряды регистра состояния прерывания (если разрешено)
Вытеснение сообщения: 1) ОУ принимает незавершённое сообщение по одному каналу, после чего следует разрыв в сообщении с максимальным промежутком времени от 6 до 10 мкс, после чего следует достоверное сообщение по тому же каналу ЛПИ или по другому (резервному) каналу ЛПИ, или 2) ОУ отрабатывает сообщение «передача» по одному каналу ЛПИ и начинает приём достоверного сообщения по альтернативному каналу ЛПИ.	ОУ МСП прекратит обработку первого (вытесняемого) сообщения и полностью переключится на обработку второго (вытесняющего) сообщения. Если оба сообщения были групповыми, то признак «Принята групповая команда» будет установлен в лог. «1» в регистре ответного слова ОУ. Если второе сообщение не было групповым, или была КУ ‘Передать ОС’ или ‘Передать последнюю команду’, то признак «Принята групповая команда» в регистре ответного слова ОУ будет установлен в лог. «0». Если групповые сообщения разрешены, и ОУ МСП получает групповое командное слово (адрес ОУ равен 31), за которым неразрывно следует командное слово «передача», направленное по собственному адресу ОУ, определяя групповое сообщение пересылки типа «ОУ-ОУ», то групповое, первым полученное, командное слово окажется вытесненным, и ОУ МСП откликнется на команду «передача».	(см. примечание 2)	Нет изменений	Прерывание не будет происходить после первого сообщения. Для второго сообщения: eom; scw_e или ss_mc_pt (если выбрано)

Примечания:

1) Отсутствие реакции в процессе формирования слова состояния блока сообщения для сообщения, посланного по собственному адресу ОУ. Нормальное слово состояния блока сообщения, включая разряд EOM, для группового сообщения (если разрешено).

2) Для первого, то есть вытесненного, сообщения в описателе блока сообщения сохранится информация, занесённая в него во время проведения МСП последовательности действий SOM. Это слово будет иметь лог. «0» в разряде EOM и лог. «1» в разряде SOM; слово состояния блока сообщения для второго сообщения будет сохранено в последующем в стеке команд описателе блока сообщения. Принимая, что второе сообщение в дальнейшем само не будет вытеснено, его слово состояния блока сообщения будет иметь: разряд EOM установленным в лог. «1» и разряд SOM – в лог. «0». Все прочие разряды будут характеризовать второе сообщение как достоверное.

1.3.12.2.2.6.16 Перечень откликов ОУ МСП на команды управления

МСП поддерживает все возможные коды команды управления (КУ) ГОСТ Р 52070-2003. Сводка откликов МСП на КУ с реакциями на различные ошибочные обстоятельства приведена в таблице 1.661.

Таблица 1.661 - Сводная таблица КУ

Разряд «Прием-передача»	Код команды	Функция	Формат сооб-ния	Состояние ОУ
Не групповые команды управления				
0	5'b00000 – 5'b01111	Неопределенные команды (см. примечание 1)	КС	Нет ОС; rt_sw: me[10]=1; rt_bit_w: cwce[0]=1;
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: cwce[0]=1, hwc[7]=1;
		Резервные команды (см. примечание 1)	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b10000	Неопределенная команда		Нет ОС; rt_sw: me[10]=1; rt_bit_w: cwce[0]=1;
	5'b10001	‘Синхронизация (с СД) ‘ (см. примечание 2)	КС с СД	ОС
			КС	Нет ОС; rt_sw: me[10]=1; rt_bit_w: lwc[6]=1;
			КС с 2 СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b10010 5'b10011	Неопределенные команды		Нет ОС; rt_sw: me[10]=1; rt_bit_w: cwce[0]=1;
	5'b10100	‘Блокировать i-ый передатчик’ (см. примечание 3)	КС с СД	ОС
			КС	Нет ОС; rt_sw: me[10]=1; rt_bit_w: lwc[6]=1;
			КС с 2 СД	Нет ОС; rt_sw [15:0]: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b10101	‘Разблокировать i-ый передатчик’ (см. примечание 3)	КС с СД	ОС
			КС	Нет ОС; rt_sw: me[10]=1; rt_bit_w: lwc[6]=1;
			КС с 2 СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b10110 - 5'b11111	Резервные команды	КС с СД	ОС
			КС	Нет ОС; rt_sw: me[10]=1; rt_bit_w: lwc[6]=1;
			КС с 2 СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
1	5'b00000	‘Принять управление интерфейсом’	КС	ОС (Если cfg1_: dbca[11]=1, то rt_sw: s_dbca[1] = 1, иначе rt_sw: s_dbca[1] = 0);
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b00001	‘Синхронизация’	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b00010	‘Передать ОС’	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b00011	‘Начать самоконтроль ОУ’	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b00100	‘Блокировать передатчик’	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b00101	‘Разблокировать передатчик’	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b00110	‘Блокировать признак неисправности ОУ’	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b00111	‘Разблокировать признак неисправности ОУ ‘	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b01000	‘Установить ОУ в исходное состояние’	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;

Продолжение таблицы 1.661

Разряд «Прием-передача»	Код команды	Функция	Формат сооб-ния	Состояние ОУ
	5'b01001 - 5'b01111	Резервные команды	КС	ОС
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b10000	‘Передать векторное слово’	КС	ОС с СД
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b10001	Неопределенная команда		Нет ОС; rt_sw: me[10]=1; rt_bit_w: cwce[0]=1;
	5'b10010	‘Передать последнюю команду’	КС	ОС с СД
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b10011	‘Передать слово ВСК ОУ’ (см. примечание 4)	КС	ОС с СД
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
	5'b10100 5'b10101	Неопределенные команды		Нет ОС; rt_sw: me[10]=1; rt_bit_w: cwce[0]=1;
	5'b10110 - 5'b11111	Резервные команды	КС	ОС с СД
			КС с СД	Нет ОС; rt_sw: me[10]=1; rt_bit_w: hwc[7]=1;
0	5'b00000 - 5'b01111	Неопределенные команды (eme[15] = 0 или eme[15] = 1, ome[6] = 0)	КС	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;
		Резервные команды (eme[15]= 1, ome[6] = 1)	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b10000	Неопределенная команда		rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;
	5'b10001	‘Синхронизация (с СД) ‘	КС	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: lwc[6]=1;
			КС с СД	rt_sw: bcr[4]=1;
			КС с 2 СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b10010	Неопределенная команда		rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;
	5'b10011	Неопределенная команда		rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;
	5'b10100	‘Блокировать i-ый передатчик’	КС	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: lwc[6]=1;
			КС с СД	rt_sw: bcr[4]=1;
			КС с 2 СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b10101	‘Разблокировать i-ый передатчик’	КС	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: lwc[6]=1;
			КС с СД	rt_sw: bcr[4]=1;
			КС с 2 СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b10110 - 5'b11111	Резервные команды	КС	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: lwc[6]=1;
			КС с СД	rt_sw: bcr[4]=1;
			КС с 2 СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;

Продолжение таблицы 1.661

Разряд «Прием-передача»	Код команды	Функция	Формат сооб-ния	Состояние ОУ
1	5'b00000	Неопределенная команда		rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;
	5'b00001	‘Синхронизация’	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b00010	Неопределенная команда		rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;
	5'b00011	‘Начать самоконтроль ОУ’	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b00100	‘Блокировать передатчик’	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b00101	‘Разблокировать передатчик’	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b00110	‘Блокировать признак неисправности ОУ‘	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b00111	‘Разблокировать признак неисправности ОУ’	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b01000	‘Установить ОУ в исходное состояние’	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b01001 - 5'b01111	Резервные команды	КС	rt_sw: bcr[4]=1;
			КС с СД	rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: hwc[7]=1;
	5'b10000 - 5'b10101	Неопределенные команды		rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;
	5'b10110 - 5'b11111	Неопределенные команды		rt_sw: me[10]=1, bcr[4]=1; rt_bit_w: cwce[0]=1;

Примечания:

КУ приёма с 5’b00000 по 5’b01111 не определены для ГОСТ Р 52070-2003. Если ОУ МСП не в режиме «разрешение расширенных возможностей конфигурирования» (ете, 15-ый разряд регистра cfg3_, установлен в лог. «1») или, если разряд «Обрабатывать КУ на «прием» без СД как резервные КУ» (отe, 6-ой разряд регистра cfg3_) установлен в лог «0», то ОУ МСП не будет откликаться на сообщение и признак «Ошибка в сообщении» ОС будет устанавливаться в лог. «1». Если МСП находится в режиме «разрешение расширенных возможностей конфигурирования» (ете, 15-ый разряд регистра cfg3_, установлен в лог. «1») и разряд «Обрабатывать КУ на «прием» без СД как резервные КУ» установлен в лог «1», то ОУ МСП будет обращаться с такими сообщениями как с «резервными КУ». То есть, откликаться на них ОС с установленным в лог. «0» признаком «Ошибка в сообщении» (если команда не была определена недопустимой).

Если разряд «загрузка СД в «регистр счетчика метки времени» при принятии КУ ‘Синхронизация (с СД)’» (l_ttos, 5-ый разряд регистра cfg2_) запрограммирован в лог. «1», то принятое совместно с этой командой слово данных загружается в регистр счетчика метки времени.

Кроме того, что ОУ откликается с очищенными (нулевыми) признаками в ОС, ничего не производится.

В режиме «нерасширенные возможности конфигурирования» (ете в лог. «0») или, если «разрешение внешнего слова ВСК» (ebwe, 15-ый разряд регистра cfg4_) установлен в лог. «0», в ЛПИ передаётся «слово ВСК», формируемое регистром слова ВСК, структура которого показана в таблице 7.21. В режиме «расширенные возможности конфигурирования» (ете, 15-ый разряд регистра cfg3_, запрограммирован в лог. «1») и, если «разрешение внешнего слова ВСК» установлен в лог. «1», в ЛПИ будет передаваться «слово ВСК», читаемое из ОЗУ. Для справок смотри раздел КУ ‘Передать слово ВСК ОУ’.

Кроме того, что ОУ откликается с очищенными (нулевыми) признаками в ОС, ничего не производится.

1.3.12.2.2.6.16.1 Подробное функциональное описание КУ (Реализация ГОСТ Р 52070-2003)

Резервные/Неопределенные команды (см. примечание 1) («Прием-передача» = 0; 5’b00000 - 5’b01111)

Последовательность сообщения =>

Резервная команда («прием/передача» = 0) --- ОС

Неопределенная команда («прием/передача» = 0)

КУ приема с кодами от 5’b00000 до 5’b01111 относятся к резервным командам при условии, что разряды регистра cfg3_: eme и ome равны логической единице. Если разряды регистра cfg3_: eme и ome не равны логической единице, то КУ приема с кодами от 5’b00000 до 5’b01111 являются неопределенными командами.

По резервной команде приема МСП отвечает ОС. В случае групповой команды устанавливается признак «Принята групповая команда» в регистре ответного слова ОУ и запрещается передача ОС.

Неопределенная команда. Нет ОС. Признаки «Ошибка в сообщении» в регистре ответного слова ОУ и cwce - «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1».

Если командное слово со словом данных, то: нет ОС, устанавливаются признаки: «Ошибка в сообщении» в регистре ответного слова ОУ; разряды: «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» и «превышение количества переданных или принятых СД» в регистре слова ВСК.

Ошибочные условия (Резервная команда):

Недостоверная команда. Нет отклика, команда игнорируется.

Команда сопровождается словом данных. Нет ОС на команду. Устанавливается признак «Ошибка в сообщении» (в регистре ответного слова ОУ) и «превышение количества переданных или принятых СД» (в регистре слова ВСК).

Принять управление интерфейсом («Прием-передача» = 1; 5’b00000)

Последовательность сообщения = ‘Принять управление интерфейсом’ ----- ОС

МСП отвечает ОС. Если процессор УВ записал лог. «0» в разряд dbca (11 разряд) регистра cfg1_, признак «Принято управление интерфейсом» в ОС ОУ будет устанавливаться в лог. «1» Когда МСП отвечает установкой в лог. «1» признака «Принято управление интерфейсом», МСП остается в режиме ОУ до тех пор, пока процессор УВ не установит режим КШ.

Ошибочные обстоятельства.

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Групповой адрес. Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1».

Синхронизация («прием/передача» = 1; 5’b00001)

Последовательность сообщения = ‘Синхронизация’ ----- ОС

МСП отвечает ОС. Если послано групповое сообщение, признак «Принято групповое сообщение» устанавливается в лог. «1» и выдача ОС подавляется. Если разрешено условие «сброса счётчика метки времени при принятии КУ ‘Синхронизация’» (c_ttos, 6-ой разряд регистра cfg2_, установлен в лог. «1»), МСП обнуляет внутренний регистр счетчика метки времени при получении данной КУ. Т.е. регистр счетчика метки времени сбрасывается в ноль сразу после приема КС, а не вовремя ЕОМ последовательности.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Передать ОС («Прием-передача» = 1; 5’b00010)

Последовательность сообщения = ‘Передать ОС’ ----- ОС

Регистр ответного слова ОУ не обновляется перед передачей и содержит состояние, полученное в результате обработки предыдущего сообщения (предполагается, что ими не были сообщения режима управления ‘Передать ОС’ или ‘Передать последнюю команду’).

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Групповой адрес. Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1».

Начать самоконтроль ОУ («Прием-передача» = 1; 5'b00011)

Последовательность сообщения => 'Начать самоконтроль ОУ' ----- ОС

МСП отвечает ОС. Если команда была групповая, признак «Принято групповое сообщение» устанавливается в лог. «1» и передача ОС подавляется. Если сообщение было негрупповым, выполняется обычный тест «проверочного возврата» по переданному ОС. Если результат проверки неудовлетворителен, признак «Неисправность ОУ» в регистре ответного слова ОУ автоматически устанавливается в лог. «1». Этот результат будет отражен в ОС в отклике на следующее негрупповое сообщение (предполагается, что разряд rt_f_we, 2-ой разряд регистра cfg3_, установлен в лог. «1»). Тест «проверочного возврата» заканчивается приблизительно 5 мкс спустя после пересечения нулевого уровня входным сигналом, которое происходит в середине области разряда контроля по чётности.

Ошибочные условия:

Недостоверная команда. Нет ОС, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Результат «проверки слова возврата» неудовлетворителен. Признак «Неисправность ОУ» в регистре ответного слова ОУ (ОС для следующей негрупповой команды) устанавливается в лог. «1». В регистре ответного слова ОУ устанавливается разряд «ошибка проверки возврата» для канала A/B.

Блокировать передатчик («Прием-передача» = 1; 5'b00100)

Последовательность сообщения => 'Блокировать передатчик' ----- ОС

Эта команда применяется только в системах с резервированной магистральной ЛПИ. МСП откликается ОС. После передачи ОС МСП блокирует какую-либо дальнейшую передачу по другому резервному каналу ЛПИ. Однажды заблокированный, передатчик может быть задействован вновь лишь с помощью КУ 'Разблокировать передатчик' или «программным сбросом» (запись лог. «1» в 0-ой разряд регистра запуска/сброса) или «аппаратным сбросом» (входной сигнал prst_n*). Отметим, что приёмники на обоих каналах ЛПИ всегда сохраняют свою активность, даже когда передатчики запрещены.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с

требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Разблокировать передатчик («Прием-передача» = 1; 5'b00101)

Последовательность сообщения => 'Разблокировать передатчик' ----- ОС

Эта команда применяется только в системах с резервированной магистральной ЛПИ. МСП откликается ОС. После передачи ОС МСП отменяет блокировку передатчика другого, резервного канала ЛПИ. Если команда была групповой, «Принято групповое сообщение» устанавливается в регистре ответного слова ОУ, а передача ОС подавляется.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Блокировать признак неисправности ОУ («Прием-передача» = 1; 5'b00110)

Последовательность сообщения => 'Блокировать признак неисправности ОУ' ----- ОС

МСП откликается ОС и препятствует в дальнейшем установлению признака «Неисправность ОУ» в регистре ответного слова ОУ. Однажды, если признак «Неисправность ОУ» был заблокирован, он снова может быть введён в употребление КУ 'Разблокировать признак неисправности ОУ' или 'Установить ОУ в исходное состояние', (сброс) или программным, или аппаратным сбросом МСП. Если команда была групповой, «Принято групповое сообщение» устанавливается, а состояние признака «Неисправность ОУ» остаётся неизменным в регистре ответного слова ОУ и передача ОС подавляется.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Разблокировать признак неисправности ОУ («Прием-передача» = 1; 5'b00111)

Последовательность сообщения => ‘Разблокировать признак неисправности ОУ’ ----- ОС

МСП откликается ОС и вновь даёт возможность устанавливаться признаку «Неисправность ОУ» в регистре ответного слова ОУ. Если команда была групповой, то «Принято групповое сообщение» устанавливается, а передача ОС подавляется.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Установить ОУ в исходное состояние (сброс) («Прием-передача» = 1; 5’b01000)

Последовательность сообщения => ‘Установить ОУ в исходное состояние’ ----- ОС

МСП откликается ОС и устанавливает в исходные состояния свои внутренние ресурсы. «Ошибка в сообщении» и «Принята групповая команда» сбрасываются в лог. «0» в регистре ответного слова ОУ. Регистр слова ВСК сбрасывается в нуль. Если передатчики были заблокированными, блокировка передатчиков отвергается. Если признак «Неисправность ОУ» был заблокированным, блокировка признака «Неисправность ОУ» отвергается. Получение данной команды не сбрасывает какие-либо программируемые процессором УВ регистры МСП. Если команда была групповой, «Принято групповое сообщение» устанавливается, а передача ОС подавляется. Также, если команда была получена как групповая и признак «Неисправность ОУ» был установлен в результате «проверки возврата» для предыдущего сообщения, признак «Неисправность ОУ» не сбросится в лог. «0». Выполнение операции завершается примерно через 8 мкс после пересечения нулевого уровня входным сигналом, которое происходит в середине области разряда контроля по четности КУ ‘Установить ОУ в исходное состояние’.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Резервные КУ («Прием-передача» = 1; 01001 - 01111)

Последовательность сообщения => Резервная КУ ----- ОС

МСП откликается ОС. Если команда была определена как недопустимая посредством таблицы допустимости КС, то признак «Ошибка в сообщении» ОС установится в лог. «1».

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» находится в лог. «0». Если это неопределенная команда, то: Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: МСП откликается ОС.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Если это неопределенная команда, то: Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1». Если это резервная команда, то: Нет ОС. Разряд «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в лог. «1».

Передать векторное слово («Прием-передача» = 1; 5’b10000)

Последовательность сообщения => ‘Передать векторное слово’ ----- ОС, «векторное слово»

МСП откликается ОС, за которым передается «векторное слово». В режиме «расширенные возможности конфигурирования» и «расширенной обработки КУ» (eme[15], emch[0] разряды регистра cfg3_ установлены в лог. «1») содержимое «векторного слова» берётся из местоположения ОЗУ с адресом 120 (шестн.). Если «расширенные возможности МСП» или «расширенная обработка КУ» не установлены то, содержимым «векторного слова» становится однословный блок данных, адресуемый таблицей указателей для подадресов на «передачу», а именно её ячейкой для передающего подадреса 5’b00000 или 5’b11111.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом следует слово данных. Нет ОС на команду. Признак «Ошибка в сообщении» в регистре ответного слова ОУ и «превышение количества переданных или принятых СД» в регистре слова ВСК устанавливаются в единичные значения.

Разряд «Прием-передача» установлен в лог. «0». Нет ОС, признак «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1».

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Нет ОС. Признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1».

Групповой адрес. Нет ОС. Признак «Ошибка в сообщении» и «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК устанавливаются в лог. «1» .

Синхронизация (с СД) («Прием-передача» = 0; 5’b10001)

Последовательность сообщения => ‘Синхронизация (с СД)’, Слово данных ----- ОС

В режиме «расширенные возможности конфигурирования» и «расширенной обработки КУ» (eme[15], emch[0] разряды регистра cfg3_ установлены в лог. «1») поступающее «слово данных» сохраняется в местоположение ОЗУ с адресом 0111 (шестн.). Если «расширенные возможности конфигурирования» или «расширенная обработка КУ» не задействованы (разряды eme или emch регистра cfg3_ установлены в лог. «0»), то поступающее «слово данных» сохраняется в однословном блоке данных, который адресуется таблицей указателей для подадресов на «прием»/«групповой прием» с полем КС «Подадрес/Режим управления» 5’b00000 или 5’b11111.

Если «загрузка СД в «регистр счетчика метки времени» при принятии КУ ‘Синхронизация (с СД)’» разрешена (l_ttos, 5-ый разряд регистра cfg2_, запрограммирован в лог. «1»), МСП загрузит принятое 16-ти разрядное слово в регистр счетчика метки времени при получении этой КУ. Загрузка СД в регистр счетчика метки времени при принятии КУ ‘Синхронизация (с СД)’ произойдет сразу после приема «слова данных» следующее за КС, а не вовремя ЕОМ последовательности. Если команда была групповая, «Принято групповое сообщение» устанавливается в «1» , а передача ОС подавляется.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За командным словом не следует слово данных. Нет ОС на команду. Устанавливается: признак «Ошибка в сообщении» в регистре ответного слова ОУ и разряд «заниженное количество переданных или принятых СД» в регистре слова ВСК.

Вслед за командным словом поступает слишком много слов данных. Нет ОС на команду. Устанавливается: «Ошибка в сообщении» в регистре ответного слова ОУ, «превышение количества переданных или принятых СД» в регистре слова ВСК.

Разряд «Приём-передача» установлен в лог. «1». Нет ОС на команду. Устанавливается: «Ошибка в сообщении» в регистре ответного слова ОУ, «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Разряд «Приём-передача» установлен в лог. «1» и групповой адрес. Устанавливаются в единичные значения: «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ, «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Передать последнюю команду («Прием-передача» = 1; 5’b10010)

Последовательность сообщения => ‘Передать последнюю команду’ ----- ОС, «последняя команда»

Регистр ответного слова ОУ не обновляется перед передачей его содержимого в ЛПИ. Он сохраняет состояние для предыдущей команды. Передаваемое в ЛПИ слово данных является предыдущей достоверной командой при условии, что ею не была КУ ‘Передать последнюю команду’.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За правильным командным словом следует слово данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова О), разряд «превышение количества переданных или принятых СД» в регистре слова ВСК.

Разряд «Прием-передача» установлен в лог. «0». Нет ОС. Устанавливаются: «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Нет ОС. Устанавливаются в единичные значения: признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Групповой адрес. Нет ОС. Устанавливаются: «Ошибка в сообщении» и «Принято групповое сообщение» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Передать слово ВСК ОУ («Прием-передача» = 1; 5’b10011)

Последовательность сообщения => ‘Передать слово ВСК ОУ’ ----- ОС, «слово ВСК»

МСП откликается ОС, после чего следует передача «слова ВСК». В режиме «нерасширенных возможностей обработки КУ» или в случае, когда «разрешение внешнего слова ВСК» (ebwe, 15-ый разряд регистра cfg4_) установлен в лог. «0» , содержимое «слова ВСК» определяется внутренним регистром слова ВСК.

В режиме «расширенные возможности конфигурирования» (eme, 15-ый разряд регистра cfg3_ установлен в лог. «1») и в случае, когда «разрешение внешнего слова ВСК» (ebwe, 15-ый разряд регистра cfg4_) установлен в лог. «1», передаваемое «слово ВСК» будет читаться из памяти. В этом случае, если «расширенная обработка КУ» запрещена (режим «расширенные возможности конфигурирования» не разрешён или emch, 0-ой разряд регистра cfg3_ установлен в лог. «0»), содержимым слова ВСК становится однословный блок данных, адресуемый таблицей указателей для подадресов на «передачу», а именно её ячейкой для передающего подадреса 5’b00000 или 5’b11111. В этом случае, следует отметить, что для данного подадреса (5’b00000 или 5’b11111) то же самое слово данных будет передаваться и при отклике на КУ ‘Передать слово ВСК ОУ’, и при отклике на КУ ‘Передать векторное слово’.

Если «расширенная обработка КУ» разрешена (разряды 15 или 0 регистра cfg3_ запрограммированы в лог. «1»), внешнее «слово ВСК» будет забираться из местоположения ОЗУ с адресом 0123.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За правильным командным словом следует слово данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова ОУ, разряд «превышение количества переданных или принятых СД» в регистре слова ВСК.

Разряд «Прием-передача» установлен в лог. «0». Нет ОС. Устанавливаются: «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Разряд «Прием-передача» установлен в лог. «0» при групповом сообщении. Нет ОС. Устанавливаются в единичные значения: признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Групповой адрес. Нет ОС. Устанавливаются: «Ошибка в сообщении» и «Принято групповое сообщение» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Блокировать i-ый передатчик (см. примечание 4) («Прием-передача» = 0; 5’b10100)

Последовательность сообщения => ‘Блокировать i-ый передатчик‘, Слово данных ----- ОС

Принятое слово данных переносится в память и ОС передается. Если «расширенная обработка КУ» не разрешена, слово данных записывается в однословный блок данных, который адресуется указателями из таблицы указателей для подадресов на «прием»/«групповой прием» с полем «Подадрес/Режим управления» 5’b00000 или 5’b11111. Если «расширенная обработка КУ» разрешена, слово данных запоминается в ячейке с шестнадцатеричным адресом 0114. Никаких других действий МСП не производит. В результате выполнения этой КУ передатчики не блокируются. Эта команда предназначена для использования в ОУ с двукратно резервированным каналом (информационной магистралью). Если команда была групповая, признак «Принято групповое сообщение» устанавливается в «1», а передача ОС подавляется.

Ошибочные условия:

Недостоверная команда. Нет отклика, команда игнорируется.

За правильным командным словом не следует слово данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова ОУ, разряд «заниженное количество переданных или принятых СД» в регистре слова ВСК.

За командным словом следует слишком много слов данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова ОУ, разряд «превышение количества переданных или принятых СД» в регистре слова ВСК.

Разряд «Приём-передача» установлен в лог. «1». Нет ОС. Устанавливаются: «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Разряд «Приём-передача» установлен в лог. «1» и групповой адрес. Нет ОС. Устанавливаются в единичные значения: признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Разблокировать i-ый передатчик (см. примечание 4) («Прием-передача» = 0; 5b'10101)

Последовательность сообщения => 'Разблокировать i-ый передатчик', Слово данных ----- ОС.

Принятое слово данных переносится в память. Если «расширенная обработка КУ» запрещена, однословный блок данных адресуется указателями из таблицы указателей для подадресов на «прием»»/«групповой прием» с полем «Подадрес/Режим управления» 5'b00000 или 5'b11111. Если «расширенная обработка КУ» разрешена, слово данных сохраняется в ячейке памяти с адресом 0115 (шестн.). Нет прежде заблокированных передатчиков, которые активизировались бы этой командой. Никакое другое действие МСП не совершает.

Эта команда предназначена для использования в ОУ с двукратно резервированным каналом (информационной магистралью). Если команда была групповая, признак «Принято групповое сообщение» в регистре ответного слова ОУ устанавливается в «1» и передача ОС подавляется.

Ошибочные условия:

Недостоверная команда. Нет ОС на команду, команда игнорируется.

За правильным командным словом не следует слово данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова ОУ, разряд «заниженное количество переданных или принятых СД» в регистре слова ВСК.

За командным словом следует слишком много слов данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова ОУ, разряд «превышение количества переданных или принятых СД» в регистре слова ВСК.

Разряд «Приём-передача» установлен в лог. «1». Нет ОС. Устанавливаются: «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Разряд «Приём-передача» установлен в лог. «1» и групповой адрес. Нет ОС. Устанавливаются в единичные значения: признаки «Ошибка в сообщении», «Принято групповое сообщение» в регистре ответного слова ОУ и разряд «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Резервные команды («Прием-передача» =1/0; 5'b10110 - 5'b11111)

Последовательность сообщения =>

Резервная команда («Прием-передача» = 1) --- ОС, Слово данных

Резервная команда («Прием-передача» = 0), Слово данных --- ОС

По резервной команде «приема» МСП запоминает слово данных в разделяемом ОЗУ. В случае групповой команды устанавливается признак «Принято групповое сообщение» и подавляется передача ОС. По резервной команде «передачи» МСП отвечает ОС с одним словом данных. Слово данных считывается из ОЗУ. Ячейки ОЗУ адресуются указателями из таблицы указателей для подадресов с полем «Подадрес/Режим управления» 0 - 31. Если «расширенная обработка КУ» разрешена (eme[15], emch[0] разряды регистра cfg3_ установлен в лог. «1»), то для команды «приема» доступны ячейки ОЗУ с адресами 0116 - 011F, а для команды «передачи» доступны ячейки ОЗУ с адресами 0126 - 012F; и если разряд sbd[0] «отделение данных групповых сообщений» регистра cfg2_ равен лог. «1», то для групповых команд «приема» доступны ячейки ОЗУ с адресами 0136 - 013F.

Ошибочные условия («Прием-передача» = 1):

Недостоверная команда. Нет ОС на команду, команда игнорируется.

Команда сопровождается словом данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова ОУ, разряд «превышение количества переданных или принятых СД» в регистре слова ВСК.

Групповая команда. Нет ОС. Устанавливаются: «Ошибка в сообщении» в регистре ответного слова ОУ и «принятое КС не согласуется с требованиями ГОСТ Р 52070-2003» в регистре слова ВСК.

Ошибочные условия («Прием-передача» = 0):

Недостоверная команда. Нет ОС на команду, команда игнорируется.

Команда не сопровождается соответствующим словом данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова ОУ, разряд «заниженное количество переданных или принятых СД» в регистре слова ВСК.

Команда сопровождается слишком большим количеством слов данных. Нет ОС на команду. Устанавливаются: признак «Ошибка в сообщении» в регистре ответного слова ОУ, разряд «превышение количества переданных или принятых СД» в регистре слова ВСК.

1.3.12.2.2.6.17 Программная процедура инициализации оконечного устройства

Нижеследующая программная последовательность - типичные шаги, которые следует выполнять процессору УВ сразу после включения питания для того, чтобы настроить (сконфигурировать) МСП на работу в режиме ОУ:

(1) Произвести программный сброс записью величины 0001 в регистр запуска/сброса;

(2) Если должна быть использована хотя бы одна из функций режима «расширенные возможности конфигурирования», установить «расширенные возможности конфигурирования», записав 8000(шестн.) в регистр cfg3_ .

(3) Присвоить начальное значение регистру маскирования прерывания (imr).

(4) Загрузить исходное значение указателя на стек команд активной области А/Б.

(5) По желанию, инициализировать стек команд активной области. Если есть желание опрашивать стековую область ОЗУ во время, когда сообщение еще находиться в обработке, в местоположения слов состояния блока сообщения для соответствующих описателей блока сообщения следует записать значения 0000. Нулевое значение слова состояния блока сообщения (SOM = EOM = 0) указывает, что обработка сообщение ещё не начиналась.

(6) Загрузить начальные значения в поисковую таблицу для подадресов ОУ активной области. Для каждого передающего, принимающего и, по желанию, группового подадреса в соответствующие разделы поисковой таблицы для подадресов необходимо занести значения указателей на блоки данны и, возможно, слов управления подадресом ОУ (если ОУ намереваются использовать в режиме «расширенное управление памятью ОУ», то слово управления подадресом ОУ задаёт выбор вида управления памятью и прерывания для каждого подадреса, в активной области). Если для ОУ имеется несколько неиспользуемых подадресов, то рекомендуется, чтобы указатели из таблицы указателей для подадресов на «прием»/«групповой приём»/«передача» были прописаны одинаковыми значениями в целях сохранения адресного пространства.

(7) Если «расширенная обработка КУ» (emch, 0-ой разряд регистра cfg3_) не используется, указатели для принимающих подадресов 0 и 31 (для сообщения ‘Синхронизация (с СД)’) обычно загружаются одним и тем же значением. Аналогично тому, указатели в таблице указателей для подадресов для подадресов 0 и 31 (для сообщения ‘Передать векторное слово’) обычно загружаются одним и тем же значением. Если «расширенная обработка КУ» используется, то слова данных для этих КУ сохраняются в местоположении с адресом 0111 (для ‘Синхронизация (с СД)’) и 0130 (для ‘Передать векторное слово’).

(8) Присвоить начальное значение регистру конфигурирования cfg2_. Это включает в себя выбор использования следующих функций.

«Расширенное управление памятью ОУ» (ermm) следует выбрать, если есть необходимость применения буферизации СД сообщения, основанного на подадресе.

«Отделение данных групповых сообщений» (sbd) следует устанавливать, если необходимо обеспечить согласие с требованием ГОСТ Р 52070-2003 для групповых сообщений. Это позволит

отделить друг от друга таблицы указателей для подадресов на «прием»/«групповой прием» для негрупповых и групповых сообщений «приём».

Для режима ОУ «запрещение 256-словных ограничений» (256_wbd) следует программировать в лог. «0». Если этот разряд устанавливается в лог. «1», то размеры всех кольцевых буферов данных становятся равными 64К слова.

Трёхразрядное поле «разрешения счётчика метки времени» (время на мл. зн. разряд) следует настраивать на желаемое разрешение для «счетчика метки времени». Существует выбор - 2, 4, 8, 16, 32, 64 мкс/МЗР или внешний синхросигнал.

Разряд «сброс счетчика метки времени при принятии КУ ‘Синхронизация’» (c_ttos) следует запрограммировать в лог. «1», если необходимо устанавливать регистр счетчика метки времени в нулевое значение при получении команды с КУ ‘Синхронизация’. Аналогично тому, «загрузка СД в «регистр счетчика метки времени» при принятии КУ ‘Синхронизация (с СД)’» (l_ttos) следует устанавливать в лог. «1», если желательно загружать регистр счетчика метки времени при получении команды с КУ ‘Синхронизация (с СД)‘, значением, пришедшим с этой командой.

«Разрешение расширенных возможностей прерывания» (ei) следует разрешать, если процессору УВ необходимо использовать опрос регистра состояния прерывания и/или, чтобы одно или более из следующих обстоятельств вызывало порождение запроса прерывания: «затягивание работы передатчика» (bt_tt), «‘прокручивание’ стека команд КШ/ОУ» (bt_csr) или для режима ОУ «принята избранная команда управления» (ss_mc_pt).

«Автоматический сброс запроса прерывания и значения регистра isr» (isac) следует установить в лог. «1», если требуется автоматически очищать регистр состояния прерывания и сбрасывать сигнал запроса прерывания на выходном сигнале msp_int (в случае уровневой конфигурации сигнала) после прочтения регистра состояния прерывания. «Вид выходного сигнала msp_int - импульс/уровень» (l_p_ir) следует установить в лог. «0» для импульсного типа прерывания (500 нс) или в лог. «1» для уровневого типа прерывания.

«‘Автосброс’ разряда ‘запрос на обслуживание’» (csr) следует установить в лог. «1», если желательно автоматически очищать признак «Запрос на обслуживание» в ОС ОУ после принятия КУ ‘Передать векторное слово’.

«Переписывать недостоверные данные» (oid) следует установить в лог. «1», если режим «кольцевая буферизация» употребляется для одного или более подадреса. Разряд «разрешение глобальной двойной буферизации для сообщений «приема»» (rsdbe) следует установить в лог. «1», для использования этого режима буферизации.

«Разрешение использования таблицы занятости подадресов» (blute) следует установить в лог. «1», если существует необходимость программирования признака «Абонент занят» в ОС ОУ для особенных принимающих, передающих, групповых подадресов.

(9) Присвоить начальное значение регистру конфигурирования cfg3_.

Если одно или более свойство расширенного режима работы МСП должно быть использовано, 15-ый разряд (eme) необходимо удерживать в лог. «1». Размер стека команд ОУ программируется 14-ым и 13-ым разрядами (bt_csz[1..0]) с возможностью выбора значений в 256 слов (устанавливается по умолчанию, 64 сообщения), 512, 1024 или 2048 слов.

Другие свойства, которые могут оказаться выбранными с помощью этого регистра, являются «запрещение в ОУ проверки КС на недопустимость» (id) (по умолчанию разрешено), «разрешение альтернативного формирования ОС» (разрешает программно устанавливать все одиннадцать разрядов в ОС ОУ), «запрещение сохранения принимаемых СД данных недопустимого сообщения / в состоянии ‘абонент занят’» (разряды i_rx_td и b_rx_td соответственно) и «расширенная обработка КУ» (emch).

Если выбрана «расширенная обработка КУ», то слова данных для КУ сохраняются в местоположениях ОЗУ с адресами 0110-013F, а запросы прерывания УВ для отдельных КУ могут разрешаться посредством таблицы, располагающейся в ОЗУ по адресам 0108-010F.

Ещё одним свойством, которое можно задействовать с помощью регистра cfg3_, является «управление формированием признака «Неисправность ОУ» в ОС» (rt_f_we). Программирование

разряда rt_f_we в лог. «1» вызывает автоматическое установление признака «Неисправность ОУ» в ОС ОУ всякий раз, когда превышается допустимая длительность непрерывной работы передатчика ОУ или после неудачного результата «проверки слова возврата».

Разряд «обрабатывать КУ на «прием» без СД как резервные КУ» (ome) следует запрограммировать в лог. «1», если с КУ «приём» (значения в поле разрядов КС «Число СД/Код команды» в диапазоне от 5'b00000 по 01111) хочется обходиться как с определёнными (разрешёнными) стандартом КУ.

(10) Присвоить начальное значение регистру конфигурирования cfg4_.

Если разряд «разрешение внешнего слова ВСК» (ebwe) установлен в лог. «1», то МСП будет извлекать слово ВСК из местоположения в ОЗУ, а не из регистра слова ВСК. Если «расширенная обработка КУ» (emch, 0-ой разряд регистра cfg3_) не используется, то слово ВСК будет читаться из места, на которое указывает содержимое ячейки таблицы указателей для подадресов на «передачу активной области ОЗУ для подадресов 0 и 31. Если оба разряда ebwe и emch установлены в логические единицы, слово ВСК будет читаться из фиксированного местоположения в разделяемом ОЗУ с адресом 0123.

Разряд «препятствовать передаче слова ВСК при 'абонент занят'» (ibwib) предотвращает передачу слова ВСК во время, когда ОУ в состоянии 'абонент занят'. Если «игнорирование 'абонент занят' для КУ» (mcob) запрограммирован в лог. «1», то ОУ МСП разрешается передавать слово данных в отклике на принятие КУ 'Передать векторное слово' или резервной КУ в состоянии 'абонент занят'.

(11) Присвоить начальное значение регистру конфигурирования cfg5_.

По умолчанию МСП считает, что на его тактовый входной сигнал подаётся синхросигнал частотой 16 МГц. «Выбор частоты входного тактового синхросигнала» (12mhz, 15-ый разряд регистра cfg5_) должен устанавливаться в лог. «1» при работе МСП от частоты в 12 МГц. Для ознакомления с дополнительными частностями по установлению рабочей частоты см. программирование конфигурационного регистра cfg5_ в разделе «Внутренние регистры МСП» настоящего руководства.

Разрядами «выбор интервала ожидания ОС» (rts[1..0]) выбирают величину временного промежутка, разрешающего ожидание отклика для МСП, среди 18,5 мкс (по умолчанию); 22,5; 50,5 или 130 мкс.

Если «разрешение проверки минимальной паузы перед выдачей ОС» (gce) установить в лог. «1», то ОУ станет проверять наличие минимально допустимого 'мертвого времени' перед стартом ОС в передаче типа «ОУ-ОУ», в случае, когда ОУ МСП является принимающим.

(12) Если определение допустимости КС задействовано, процессору УВ следует прописать таблицу допустимости КС (адреса расположения 0300-03FF) необходимыми значениями.

(13) Если таблица занятости подадресов разрешена, выберите желаемые подадреса, которые должны быть занятыми, программируя значения этой таблицы в области адресов с 0240 по 0247.

(14) Если «расширенные возможности прерывания» и «расширенная обработка КУ» разрешены, прерывания для отдельных сообщений режима управления могут позволяться программированием таблицы прерываний для избранных КУ - с 0108 по 010F.

(15) Данные, подлежащие передачи по ЛПИ (в откликах на команды «передача»), следует занести в соответствующие «блоки данных». По желанию, ячейкам ожидаемых слов данных можно присвоить нулевые начальные значения.

(16) Чтобы сконфигурировать МСП в рабочее состояние ОУ, т.е. включенного, запущенного в работу устройства, необходимо записать в регистр cfg1_ соответствующее значение, в котором 15-ый разряд (СЗР) в лог. «1» и 14-разряд в лог. «0». Текущая активная область памяти выбирается 13-ым разрядом (0 для А, 1 для Б). Если «разрешение альтернативного формирования ОС» не разрешено, в разряды с 11 по 8 следует занести начальные значения для разрядов (инвертированные), которые будут передаваться в ОС ОУ: «Принято управление интерфейсом», «Абонент занят», «Запрос обслуживания», «Неисправность абонента». Кроме того, в режиме «расширенные возможности конфигурирования» посредством

7-го разряда регистра cfg1_ в ответном слове ОУ становится программируемым ещё и признак «Неисправность ОУ». В эти разряды должны прописываться обратные значения от желаемых значений соответствующих им признаков в ОС ОУ. Если «разрешение альтернативного формирования ОС» разрешено, признаки ОС с 10 по 0 будут определяться разрядами с 11 по 1 регистра cfg1_ соответственно. В этом случае логические величины (не инверсные) предполагаемого ОС задаются программным образом.

1.3.12.2.2.6.18 Обслуживание завершенных сообщений

ОУ МСП обеспечивает (поддерживает) ряд технических способов для определения момента завершения обработки сообщения. Эти методы рассчитаны на программное обеспечение либо выполняющее циклический опрос, либо работающее по прерыванию. Существует несколько методов опроса, которые могут быть употреблены:

(1) В режиме «расширенные возможности конфигурирования» УВ может постоянно опрашивать разряд rt_mip (RT Message In Progress) (только чтение): флаг процесса отработки сообщения ОУ, 0-ой разряд регистра cfg1_.

Этот разряд будет возвращать в себе величину лог. «0», всякий раз, когда ОУ МСП находится вне состояния обработки сообщения. В то время, когда ОУ МСП обслуживает сообщение (после получения командного слова), этот разряд будет отвечать лог. «1». Когда сообщение завершается, значение разряда будет снова возвращаться к лог. «0».

(2) УВ может опрашивать содержимое ячейки указателя на стек команд в памяти. Указатель на стек команд активной области увеличивается на четыре в начале обработки каждого сообщения (после принятия командного слова), если иное не задано в регистре cfg6_.

(3) Если УВ нуждается в определении поступления особенного командного слова, это можно организовать, опрашивая чтением регистр командного слова сообщения ОУ.

Следует отметить, что содержимое данного регистра обновляется в начале обработки сообщения. УВ может затем приступить к опросу разряда eom (завершение отработки сообщения) регистра состояния прерывания, чтобы определить момент, когда сообщение окажется выполненным. Смотри разъяснения ниже по тексту.

(4) Если МСП в режиме «расширенные возможности конфигурирования» и «расширенные возможности прерывания» (15-ый разряд регистра cfg2_) разрешены, УВ может опрашивать регистр состояния прерывания.

В этом режиме различные разряды регистра состояния прерывания устанавливаются в единичные положения без оглядки на запрограммированные величины соответствующих им разрядов регистра маскирования прерывания, то есть независимо. В этом режиме УВ может определить завершение сообщения, опрашивая чтением регистр состояния прерывания пока разряд eom («завершение сообщения») не возвратит лог. «1».

В этом режиме процессор УВ может также определять опросом завершение отработки сообщения для особенного передающего, принимающего или группового подадреса. Для разрешения этого должен быть задействован режим «расширенное управление памятью ОУ» (1-ый разряд регистра cfg2_ в лог. «1»). Чтобы получить установление в лог. «1» по отдельному, избранному подадресу на «прием»/«групповой прием»/«передачу», необходимо записать лог. «1» в соответствующий разряд в слове управления подадресом ОУ. Разряды для других слов из таблицы управления подадресом ОУ следует установить в лог. «0». Это вызовет установление в лог. «1» разряда scw_e «завершение обработки избранного сообщения для режимов КШ/ОУ» после завершения желаемого сообщения.

Аналогично, УВ может по опросу ожидать отработку избранной КУ. Это свойство разрешается «расширенной обработкой КУ» (0-ой разряд регистра cfg3_). Желаемый код команды может быть выбран программированием соответственного разряда в таблице прерываний для избранных КУ (область адресов 0108-010F). Когда отдельное, точно описанное сообщение оказалось выполненным, разряд ss_mc_pt – «принята избранная команда управления» регистра состояния прерывания станет возвращать в операции чтения лог. «1».

Аналогично, регистр состояния прерывания может опрашиваться для определения события «ошибка формата сообщения», «‘прокручивание’ стека команд КШ/ОУ», «‘прокручивание’ кольцевого буфера данных ОУ».

1.3.12.2.2.6.19 Обработка ошибочных сообщений ОУ

Предпочтительным способом обработки ошибочных сообщений ОУ является использование «кольцевой буферизации» данных и/или «двойной буферизации сообщений «приема»». В случае режима «кольцевая буферизация» разряд «переписывать недостоверные данные» следует устанавливать в лог. «1». При этом способе обработки сообщения слова данных, прежде принятые и сохранённые для недостоверного сообщения, будут автоматически переписаны ОУ МСП (после ВПС КШ). В большинстве систем контроллер ЛПИ будет воспроизводить неудавшиеся сообщения. При использовании таких приёмов возникновение ошибок и последующие попытки возобновления сообщений делаются невидимыми для процессора УВ.

Если необходимо, процессор УВ ОУ может установить (обнаружить) неудавшееся сообщение несколькими приёмами:

Определение (опросом или по прерыванию) происшествия события «ошибка формата сообщения».

Чтение слова состояния блока сообщения для всех обработанных сообщений. Разряды 12-9 и 6-0 укажут на происшествие какой-либо ошибки в принятых сообщениях.

1.3.12.2.2.7 Работа в режиме монитора шины (МТ)

МСП может функционировать в трёх вариантах режима прослушивания (мониторинга) магистрали:

наблюдение за словами («монитор слов»);

наблюдение за избранными сообщениями («монитор сообщений»);

комбинированный режим «ОУ/монитор сообщений».

Когда МСП находится в режиме «нерасширенные возможности конфигурирования» (разряд eme регистра cfg3_ в лог. «0») работы, то «монитор слов» способен производить лишь наблюдение за словами, передаваемыми по ЛПИ, последовательно занося их в ОЗУ и сопровождая вспомогательными данными, с целью возможности последующего восстановления (реконструкции) сообщений в постобработке. В этом режиме, при прослушивании магистрали, для каждого принятого слова (КС, ОС, СД) в кольцевом буфере ОЗУ МСП сохраняется 16 разрядов данных принятого слова и, вслед за ним, соответствующее идентификационное слово (ID – слово).

ID-слово содержит следующую информацию: по какому (из двух - А или В) каналу передавалось данное слово, корректно ли оно и на какой интервал времени отстоит начало приёма данного слова от завершения приёма предыдущего слова (с точностью 0.5 мкс., от 0 до 127.5 мкс.; для большего интервала берётся значение в 128 мкс.); В этом режиме работы предусмотрена также возможность избирательного запуска для монитора слов.

В режиме работы МСП «расширенные возможности конфигурирования» «монитор слов» снабжен программируемым «словом срабатывания». «Слово срабатывания» определяет особенное, точно описанное командное слово, которое может быть использовано для того, чтобы запустить процесс сохранения слов, принимаемых из ЛПИ в память или для того, чтобы завершить процесс сохранения слов или запросить прерывание. Кроме того, «монитор слов» также возможно запустить в работу от внешнего источника запуска.

В режиме работы МСП «расширенные возможности конфигурирования» (разряд eme регистра cfg3_ в лог. «1») могут быть задействованы оба вида работы МТ, связанные с наблюдением сообщений, а именно режим «монитора сообщений» и комбинированный режим «ОУ/ монитор сообщений».

«Монитор сообщений» поддерживает фильтрацию сообщений, основанную на разрядах КС «Адрес ОУ», «Прием-передача», «Подадрес/Режим управления», и снабжен раздельными стеками команд и буфером данных. «Монитор сообщений» также записывает в память содержимое 16-ти разрядного регистра счетчика метки времени.

Режим «ОУ/монитор сообщений» обеспечен полным набором свойств режима ОУ для сообщений, направляемых по собственному адресу ОУ МСП, плюс свойства наблюдения заданных, задаваемых при конфигурировании, сообщений для всех других адресов. По аналогии с режимом ОУ в стеке команд «монитора сообщений» сохраняются все командные слова, слова меток времени, слова состояния блоков сообщений и указатели блоков словных данных для каждого контролируемого (наблюдаемого) сообщения. Размеры стеков команд и данных задаются посредством разрядов с 12 по 8 регистра cfg3_.

1.3.12.2.2.7.1 Режим «монитор слов»

В режиме «монитор слов» МСП ведёт наблюдение по обоим канала ЛПИ. После программной инициализации и последовательности запуска работы монитора (возможно по событию) МСП начинает записывать в ОЗУ все КС, ОС и СД, принимаемые по обоим каналам ЛПИ. Для каждого слова, принятого по любому из каналов пара слов записывается в ОЗУ. Первое из слов - 16-ти разрядные данные полученного слова. Второе слово – ID слово. ID слово содержит информацию относительно канала (A или B), достоверности слова и межсловного промежутка времени. Слова данных (КС, СД, ОС) и ID -слова сохраняются в кольцевом буфере данных в ОЗУ (стек данных). Размер буфера данных может выбираться среди 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536 слов посредством разрядов 10, 9, и 8 регистра cfg3_.

1.3.12.2.2.7.2 Распределение памяти монитора слов

Типичное распределение памяти для «монитора слов» приведено в таблице 1.662.

Таблица 1.662

Адрес (шестн.)	Функция
0000	Первое принятое слово
0001	Слово ID для первого слова
0002	Второе принятое слово
0003	Слово ID для второго слова
…	…
…	…
0100	Начальный указатель буфера данных области А (Фиксированное положение)
…	…
0104	Начальный указатель буфера данных области Б (Фиксированное положение)
…	…
…	…
FFFF	…

Предполагается, что для монитора слов МСП доступно полное, в 64К слов, адресное пространство разделяемого ОЗУ. Начальный указатель буфера данных активной области предназначен для содержания адреса, в котором сохраняется первое наблюдаемое слово после запуска монитора в работу. Например, предположим, что активный начальный указатель буфера данных области А (местоположение 0100) проинициализирован значением «0000». Первое принятое слово данных будет записано в местоположение «0000», а слово ID для первого слова будет занесено в местоположение «0001» и т.д. Начальный указатель буфера данных области Б расположен по адресу 0104.

Текущий адрес в буфере данных (указатель буфера данных) обеспечивается с помощью внутреннего регистра. Значение адреса (указателя буфера данных) может быть прочитано процессором УВ из регистр указателя буфера данных ОУ/МТ (rm_dsa[15..0]). Важно отметить, что по достижению указателем буфера данных адреса расположения начального указателя буфера данных «0100» (для активной области А) или «0104» (для активной области Б), поверх первоначального значения, содержащегося в начальном указателе буфера данных, записывается наблюдаемое слово данных или ID слово. Когда величина указателя буфера данных достигает адреса FFFF (или 0FFF, если используется только внутренняя 4К словное ОЗУ МСП), указателя буфера данных прокручивается на нулевое значение.

Чтобы настроить МСП на выполнение наблюдения в режиме «монитора слов», УВ следует запрограммировать разряды 15, 14 и 12 регистра cfg1_ в лог. «0», лог. «1» и лог. «0» соответственно. Далее, в начальный указатель буфера данных для активной области в адресном пространстве разделяемого ОЗУ необходимо загрузить стартовое местоположение буфера данных. В завершение, для старта наблюдения (мониторинга) следует записать в регистр запуска/сброса слово с установленным в лог. «1» разрядом «запуск работы КШ/МТ».

Замечание - в расширенном режиме работы «монитор слов» может быть запущен в работу внешним источником запуска (вход msp_ssflag_ext_trig).

Слово распознавания (ID - слово) содержит «признак слова распознавания» (wf, всегда в лог. «1») плюс информацию касательно использовавшегося во время пересылки канала ЛПИ, словной достоверности, типа синхроимпульса (КС/ОС либо СД), а также информацию о времени межсловного промежутка. Последнее состоит из разряда «признак соприкосновения слов» и 8-ми разрядного поля временного отстояния (разрыва), указывающего время в диапазоне от 0 до 127 мкс с разрешением 0,5 мкс/МЗР.

Для того чтобы выключить наблюдение (мониторинг), УВ должен записать в регистр запуска/сброса слово с установленным в лог. «1» разрядом «программный сброс МСП».

1.3.12.2.2.7.2.1 Монитор слов - «слово распознавания» (ID - слово)

Структура слова распознавания (ID - слова) приведена в таблице 1.663.

Таблица 1.663



gt[7...0] (gap time): длительность межсловного промежутка;
wf (word flag): признак слова распознавания;
this_rt: признак равенства значения в поле КС ‘адрес ОУ’ собственному адресу ОУ;
brcst (BROADCAST): признак группового сообщения;
error: признак ошибки;
com_dat (COMMAND/DATA* SYNC): тип синхросигнала слова - КС/ОС либо СД;
ch_b_a (CHANNEL B/A*): использовавшийся для передачи слова канал обмена A/B;
cd_g (CONTIGUOUS DATA/GAP): признак соприкосновения слов;
mc (MODE CODE*): признак КУ/ошибка подтверждения (квитирования).

gt[7...0] длительность межсловного промежутка: Если «соприкасающиеся данные», разряд cd_g, установлен в лог. «1», то эти восемь разрядов не используются. Если разряд cd_g установлен в лог. «0», то поле «длительность межсловного промежутка» содержит значение промежутка времени между концом предыдущего слова и началом данного слова; значение может изменяться с разрешением в 0,5 мкс на младший значащий разряд вплоть до максимального значения в 127,5 мкс. Для межсловного промежутка времени, большего, чем 127,5 мкс, поле промежутка времени gt[7...0] будет иметь значение FF(шестн.). Следует отметить что, если текущее слово было принято по другому (относительно предыдущего) каналу ЛПИ, то поле «длительность межсловного промежутка» будет показывать величину приблизительно на 20 мкс большую, чем действительный промежуток времени отстояния от слова, полученного по первому каналу. То есть, если текущее слово получено по другому каналу относительно предыдущего слова и текущий межсловный промежуток меньше 20 мкс, это указывает на то, что пересылка текущего слова частично перекрывало во времени предыдущее слово (полученное по другому каналу).

wf признак слова распознавания: Всегда установлен в лог. «1» для слова распознавания. В каждом местоположении в ОЗУ, где ожидается сохранить слово распознавания, УВ следует предварительно загрузить лог. «0».

this_rt признак равенства значения в поле КС ‘адрес ОУ’ собственному адресу ОУ: Если разряд установлен в лог. «0», то это показывает, что принятое слово было достоверным, содержащим тип синхроимпульса КС/ОС, имело в поле адреса ОУ значение, равное адресу, установленному на входах msp_rtad[4:0] МСП и корректное значение сигнала «контроль по чётности» на входе msp_rtadp. В противном случае этот разряд будет иметь единичное значение.

brcst признак группового сообщения: если данный разряд установлен в лог. «0», то он показывает, что принятое слово было достоверным, содержащим тип синхроимпульса КС/ОС и значение в адресном поле равное 31. Иначе, этот разряд равен лог. «1».

Замечание - Если МСП запрограммирована в «расширенный режим конфигурирования» и установлен «запрет групповых сообщений» (bd, 7-ой разряд регистра cfg5_ в лог. «1»), то разряд brcst всегда будет возвращать (при чтении данного слова) лог. «1».

error признак ошибки: если данный разряд в лог. «0», то показывает, что принятое слово удовлетворяет всем условиям, предъявляемым к достоверности ГОСТ Р 52070-2003. Если в лог. «1» - указывает на ошибку в слове: ошибка кодирования (бифазовая манипуляция), ошибка синхроимпульса, количества разрядов или контроля по чётности.

com_dat тип синхросигнала слова - КС/ОС либо СД: если данный разряд в лог. «1», то это показывает, что слово содержит тип синхроимпульса КС/ОС. Если этот разряд в лог. «0», то это указывает, что слово содержит синхросигнал СД.

ch_b_a «использовавшийся для передачи слова канал обмена А/В»: если данный разряд в лог. «0», то это указывает, что слово было принято по каналу А ЛПИ. Если в лог. «1» - по каналу В.

cd_g признак соприкосновения слов: если данный разряд в лог. «1», то это указывает, что либо между словами не было промежутка, либо ‘мертвое время’, равное 2 мкс от конца предыдущего слова до начала текущего (настоящего), не было выдержано. В этом случае поле «длительность межсловного промежутка» не имеет смысла. Если этот разряд в лог. «0», то это указывает на наличие промежутка, большего, чем 2 мкс (от конца предыдущего слова до начала текущего слова). В этом случае поле «длительность межсловного промежутка» содержит достоверную информацию.

mc признак КУ/ошибка подтверждения (квитирования): Если данный разряд в лог. «0», то это указывает на то, что принятые данные достоверны, имели тип синхроимпульса КС/ОС и содержали в поле «Подадрес/Режим управления» величину 5’b00000, либо 5’b11111. Иначе, данный разряд будет в лог. «1».

Если mtgo запрограммирован в лог. «1», то разряд mc наблюдаемого целевого слова расширит свое функциональное назначение и mc будет устанавливаться в лог. «1» не только при приеме слова отличного от достоверной КУ, но также будет указывать на происшествие ошибки «квитирования» (затягивания процедуры получения МСП разрешения от УВ на обладание системной магистралью). Неправильное выполнение процедуры «квитирования» может случаться в «прозрачном» режиме сопряжения УВ с МСП. Когда происходит истечение максимально допустимого срока ожидания МСП доступа к разделяемому с УВ ОЗУ, наблюдаемое (полученное) слово вместе со своим ID теряются, и МСП установит разряд mc в ID для следующего принятого слова, с целью указать, что произошло истечение срока ожидания, и слово было потеряно.

1.3.12.2.2.7.3 Инициализация монитора слов

В режиме «монитор слов» начальный указатель буфера данных активной области А/Б обеспечивает МСП адресом, где сохраняется первое наблюдаемое после старта работы слово. Этот указатель должен быть прописан нужным (желаемым) адресом процессором УВ перед запуском отслеживания событий ЛПИ. Информация первого принятого слова будет сохранена в местоположении, на которое ссылается начальный указатель буфера данных активной области А/Б. Важно отметить, что оба начальных указателя буфера данных (области А и Б) будут испорчены впоследствии, записью, полученным словом данных, либо ID.

1.3.12.2.2.7.3.1 Запуск и остановка работы

Для запуска МТ МСП необходимо записать величину 0002 в регистр запуска/сброса. Прекращение дальнейшего сохранения слов из ЛПИ осуществляется записью значения 0001 в регистр запуска/сброса.

1.3.12.2.2.7.3.2 Последовательность программной инициализации для режима «монитор слов»

Программная процедура инициализации МСП в режиме «монитор слов» содержит следующие пункты:

- Произвести программный сброс записью величины 0001 в регистр запуска/сброса;
- Сконфигурировать МСП в режим МТ с активной областью А(Б), записав значение 5000(7000) в регистр cfg1_. Выбор активной области влияет на адрес расположения начального

указателя буфера данных. Если активная область А, то начальный указатель буфера данных расположен в ячейке 0100. Если активная область Б, то начальный указатель буфера данных расположен в ячейке 0104.

- Записать желаемое местоположение для сохранения первого слова в начальный указатель буфера данных области А(Б) с местоположением 0100 (0104).
- По желанию, очистить область разделяемого ОЗУ, где ожидается расположить принимаемые данные и «ID слова».
- Запустить монитор в работу по сохранению принимаемых слов, записав значение 0002 в регистр запуска/сброса.
- Для последующей остановки (прекращения сохранения принимаемых слов в ОЗУ) запишите значение 0001 в регистр запуска/сброса.

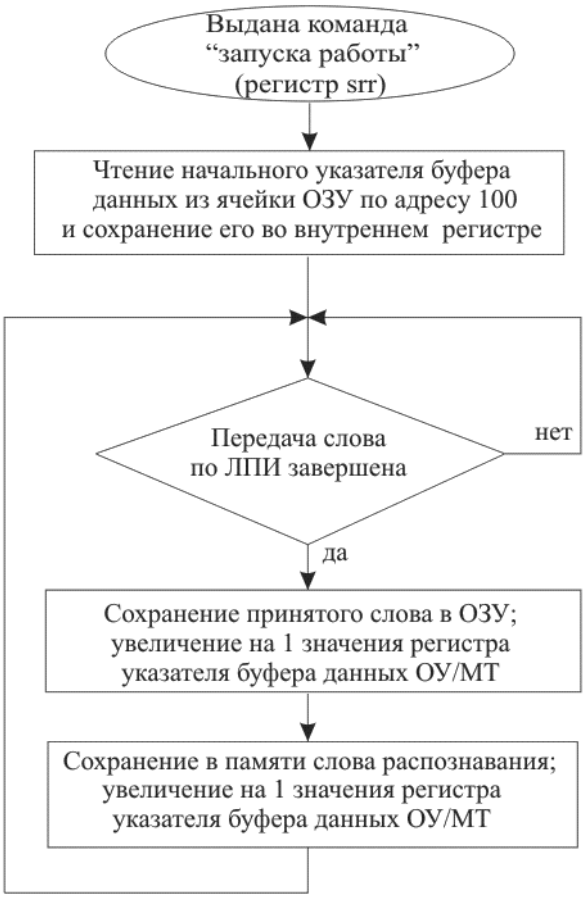


Рисунок 1.211 - Сохранение данных в режиме монитор слов

1.3.12.2.2.7.4 Различение командных слов и ответных слов: восстановление сообщений

В режиме «монитор слов» восстановление сообщения связано с определением, с какого именно события (слова) отдельное, принятое сообщение начинается и когда заканчивается. Существенной частью этого процесса, который должен выполняться программно (в реальном ли времени или нет), является различение между командными словами и ответными словами. Вероятно, что универсального алгоритма для абсолютно точного (100%) различения между командными и ответными словами не существует. В алгоритме нужно учитывать и замещение (вытеснение) команд, и переключение каналов ЛПИ, и, вдобавок, наличие множества ошибок в сообщениях. Тем не менее, правила, приведенные ниже, подтвердят успешность их

использования в определении КС и ОС для огромного большинства (больше, чем 99,99%) из всех реальных ГОСТ Р 52070-2003 применений:

- Слово, содержащее тип синхросигнала КС/ОС и какое-либо значение в поле подадреса, отличное от 0 и 8, является командным словом.
- Слово, содержащее тип синхросигнала КС/ОС, которому предшествует промежутком ‘мёртвого времени’, больший заданного для МСП «интервала ожидания ОС», является командным словом. Для большинства применений длительность в 14 мкс является вполне приемлемой начальной величиной для «интервала ожидания ОС». Для применений с очень большой длиной кабеля ЛПИ (до ста метров и более) может потребоваться больший период «интервала ожидания ОС». Слово, содержащее тип синхросигнала КС/ОС, полученное по альтернативному каналу шины относительно последнего (предшествующего) полученного слова, можно и должно считать командным словом.
- Слово, содержащее тип синхросигнала КС/ОС, полученное по альтернативному каналу (по отношению к последнему полученному слову) может быть рассмотрено как КС.
- Два соприкасающихся соседних слова (без промежутка времени между ними), содержащее тип синхросигнала КС/ОС, с промежутком, предшествующим первому из двух слов, являются двумя командными словами для передачи типа «ОУ-ОУ» или групповой передачи «ОУ-ОУ».
- Слово, содержащее тип синхросигнала КС/ОС, и величиной 31 в поле «Адрес ОУ» (разряды с 15 по 11) является командным словом, направленным по «групповому адресу».
- Слово, следующее после командного слова, направленного по групповому адресу, ‘несоприкасающееся’, содержащее тип синхросигнала КС/ОС, является командным словом.
- Принципиальной задачей для программного обеспечения центрального процессора, анализирующего буфер данных монитора слов, является восстановление (реконструкция) сообщений. Когда процедура грамматического разбора встречает командное слово, отмечая начало нового сообщения, ей следует продолжать исследовать, осматривать сообщение, проверяя его на соответствие одному из существующих форматов ГОСТ Р 52070-2003.
- Если программное обеспечение реконструкции делает грамматический разбор текущего сообщения, которое не является пересылкой типа «ОУ-ОУ», встречает слово с синхросигналом типа КС/ОС и полем «Адрес ОУ» отличным от поля «Адрес ОУ» в предыдущем слове с синхросигналом типа КС/ОС, то такое слово может быть рассмотрено как командное слово следующего сообщения.
- Если «программа реконструкции сообщения», разбирая поток СД, определяет конец завершенного, достоверного сообщения, после которого следует промежуток времени (возможно, меньший, чем величина максимального «интервала ожидания ОС»), то следующее слово с синхросигналом типа КС/ОС будет командным словом. В данном случае безразлично, содержит ли слово тот же самый «Адрес ОУ» или другой «Адрес ОУ» относительно предыдущего командного слова.

1.3.12.2.2.7.5 Запуск/останов «монитора слов»

В режиме «монитор слов» существует возможность запуска по событию ‘распознавание эталона’, возможность порождения прерывания по событию ‘распознавания эталона’. 16-ти разрядное эталонное слово, употребляемое и для запуска, и для порождения прерывания, сохраняется в регистре слова срабатывания (bcft_rtlc_mttw [15..0]). 16-ти разрядное эталонное слово должно обязательно иметь тип синхросигнала КС/ОС. Прерывание по распознаванию эталона разрешается записью единицы в m_ss_mc_pt, 2-разряд регистра маскирования

прерывания. Срабатывание по распознаванию эталона («слова срабатывания») разрешается программированием разряда «разрешение запуска/останова по событию» (tew, 11-ый разряд регистра cfg1_) в лог. «1». Кроме того, имеется выбор вида работ после срабатывания - либо «старт записи слов по слову срабатывания» (st_ot, 10-ый разряд регистра cfg1_), либо «останов записи слов по слову срабатывания» (sp_ot, 9-ый разряд регистра cfg1_).

Таблица 1.663 - Таблица истинности срабатывания «монитора слов» по распознаванию эталона (разряды регистра cfg1_)

Старт записи слов по слову срабатывания (st_ot, 10 разряд)	Останов записи слов по слову срабатыва-ния (sp_ot, 9-ый разряд)	Разрешение запуска по внешнему сигналу (mete, 7-ый разряд)	Запуск монитор слов и сохранение данных
0	0	0	Программный запуск, МТ сохраняет все данные, передающиеся по ЛПИ.
0	0	1	Программный или старт от внешнего источника запуска, МТ сохраняет все данные, передающиеся по ЛПИ.
0	1	0	Программный старт, МТ сохраняет все данные до тех пор пока не встретится слово, одинаковое по значению со словом, содержащимся в регистре слова срабатывания МТ. Это командное слово становится последним, которое сохраняется в буфере данных монитора слов.
0	1	1	Программный или старт от внешнего источника запуска, МТ сохраняет все данные до тех пор, пока не встретится слово, одинаковое по значению со словом, содержащимся в регистре слова срабатывания МТ. Это командное слово становится последним, которое сохраняется в буфере данных монитора слов.
1	0	0	Программный старт, сохраняются все данные после обнаружения достоверного командного слова, равного по своему значению слову, содержащемуся в регистре слова срабатывания МТ. Это командное слово также оказывается сохраненным в буфере данных монитора слов.
1	0	1	Программный или старт от внешнего источника запуска, сохраняются все данные после обнаружения достоверного командного слова, равного по своему значению слову, содержащемуся в регистре слова срабатывания МТ. Это командное слово также оказывается сохраненным в буфере данных монитора слов.
1	1	0	Программный старт, сохраняется только слово, равное по значению слову, находящемуся в регистре слова срабатывания МТ. После распознавания эталонного слова, монитор слов останавливает свою работу.
1	1	1	Программный старт или старт от внешнего источника запуска, сохраняется только слово, равное по значению слову, находящемуся в регистре слова срабатывания МТ. После распознавания эталонного слова, монитор слов останавливает свою работу.

Таблица истинности срабатывания «монитор слов» (таблица 1.663) предполагает следующее:

Пользователь находится в режиме «расширенных возможностей конфигурирования» МСП.

Режим «монитор слов» является выбранным с помощью регистра cfg1_ - разряд 15 равен «0», разряд 14 равен «1», разряд 12 равен «0».

Пользователь, программно со стороны УВ, записывает необходимое значение в регистр слова срабатывания монитора.

Разряд «разрешение запуска/останова по событию» (разряд 11 в регистре cfg1_) установлен в лог. «1».

1.3.12.2.2.7.6 Работа в режиме «монитор сообщений»

МСП снабжена гибким интерфейсом, основанным на использовании разрядов «Адрес ОУ», «Приём-передача», «Подадрес/Режим управления» командного слова, который способен отслеживать избранные, отдельные сообщения форматов, предусмотренных ГОСТ Р 52070-2003, с минимальным вмешательством УВ. Режим отслеживания сообщений в МСП («монитор сообщений») производит восстановление всех командных и ответных посылок на ЛПИ (по каналу А и В) и сохраняет «следы от них» в ОЗУ, опираясь на заданный пользователем фильтр – на основании значения полей «Адрес ОУ», «Приём-передача», «Подадрес/Режим управления» командного слова. Этот монитор может быть употреблен исключительно как «монитор сообщений» или же в комбинации с ОУ - «ОУ/монитор сообщений». Монитор сообщений стек команд и буфер данных, которые не зависят от стека команд ОУ. Указатели для этих стеков располагаются в фиксированных местоположениях ОЗУ.

Отслеживание сообщений разрешается в режиме «ОУ/монитор сообщений» после разрешения «расширенных возможностей конфигурирования» (разряд 15 в регистре cfg3_ установлен в лог. «1»), программирования поля rt_bc_mt[1..0] регистра cfg1_ в значение 2’b01 (разряд 15 в лог. «0», разряд 14 в лог. «1») и разряда «разрешение режима монитор сообщений» (mme, разряд 12 регистра cfg1_) в лог. «1».

Следует отметить, что издание команды «запуск работы КШ/МТ» (установка в лог. «1» первого разряда регистра запуска/сброса) необходимо для того, чтобы стартовать монитор сообщений. Тем не менее, в комбинированном режиме «ОУ/монитор сообщений» издавать команду «запуск работы КШ/МТ» не требуется.

1.3.12.2.2.7.6.1 Функции «монитора сообщений»

Во время работы в режиме «монитор сообщений», принимая достоверную команду, МСП будет обращаться к таблица определения избранных сообщений (таблица с фиксированным местоположением в ОЗУ). Таблица определения избранных сообщений предназначена для того, чтобы определить, задано ли ‘отслеживание’ сообщений с данным командным словом или нет. Местоположение слова (разрешающего/запрещающего отслеживание данного КС) в таблице определяется с помощью поля разрядов «Адрес ОУ» (msp_rtad[4:0]), разряда «Прием-передача» и четвертого разряда (старшего - SUBADRESS_4) из поля «Подадрес/Режим управления» текущего командного слова, а именно - значение этого подмножества разрядов арифметически складывается с базовым адресом 0280 (шестн.) самой таблицы (см. таблицу 8.3). Внутри найденного 16-ти разрядного слова таблицы определения избранных сообщений, положение разряда задаётся разрядами 3..0 поля «Подадрес/режим управления» текущего командного слова. Значение разряда, который выбирается 4-х разрядным полем «Подадрес/Режим управления» командного слова, определяет избранность (разрешение/запрещение) текущего командного слова (сообщения). Если определенный указанным способом разряд содержит лог. «0», то “отслеживание’ сообщения с принятым КС не разрешено и МСП прекратит обработку этого сообщения и приступит к отслеживанию следующего, нового командного слова. Следует отметить, что ОС может быть опознано, интерпретировано, МСП как новое командное слово. Если соответственный разряд таблицы определения избранных сообщений окажется установленным в лог. «1», то командное слово разрешено (входит в список избранных слов), и МСП создаст точку входа в мониторном стеке команд (основан на указателе стека команд монитора) и сохранит данные, относящиеся к этой команде в последовательно расположенные ячейки памяти в мониторном буфере данных.

Таблица 1.664 - Правило формирования указателя на слово в таблице определения избранных сообщений

Разряд адреса	источник значения
15(СЗР)	лог. «0»
14	лог. «0»
13	лог. «0»
12	лог. «0»
11	лог. «0»
10	лог. «0»
9	лог. «1»
8	лог. «0»
7	лог. «1»
6	msp_rtad[4]
5	msp_rtad[3]
4	msp_rtad2]
3	msp_rtad[1]
2	msp_rtad[0]
1	«Приём-передача»
0(МЗР)	SUBADRESS_4

Для четного значения адреса в таблице определения избранных сообщений в режиме «монитор сообщений» (SUBADRESS_4=A0=0), разряд 15 задаёт отслеживание для подадреса 15, разряд 14 задаёт отслеживание для подадреса 14, ..., разряд 0 задаёт отслеживание для подадреса 0. Для нечетного номера адреса в таблице определения избранных сообщений (SUBADRESS_4 = 1) разряд 15 разрешает «отслеживание» для подадреса 31, разряд 14 задаёт отслеживание для подадреса 30, ..., разряд 0 разрешает отслеживание для подадреса 16. Установка разряда в лог. «0» запрещает отслеживание сообщения для соответствующего значения множества разрядов {«Адрес ОУ», «Прием-передача», «Подадрес/Режим управления»} командного слова, то есть сообщение с данным КС будет проигнорировано. Установка разряда в лог. «1» разрешает отслеживание для соответствующего значения множества разрядов командного слова - {«Адрес ОУ», «Прием-передача», «Подадрес/Режим управления»} -, то есть сообщение с данным КС будет сохранено в ОЗУ.

1.3.12.2.2.7.6.2 Форматы блоков сообщений в режиме «монитор сообщений»

Порядок размещения информации в буфере данных монитора зависит от формата наблюдаемого сообщения. Сообщение формата 1 («КШ к ОУ» «приём») вызовет сперва сохранение командного слова в стеке команд монитора (в описателе блока сообщения), затем сохранение «N-слов» данных, пришедших за ним к моменту принятия ОС, и ОС в мониторном буфере данных.

Замечание - Если задано (разрешено) КС «приём», то для сообщения типа «ОУ-ОУ» разряд «признак пересылки типа ‘ОУ-ОУ’» (разряд 11) слова состояния блока сообщения будет устанавливаться в лог. «1» в независимости от того, было ли КС «передача» задано (разрешено). КС «передача» пишется в буфер данных. В случае, когда МСП выбирает только команду «передача» в сообщении типа «ОУ-ОУ», первым словом, сохраненным в мониторном буфере данных, будет второе командное слово сообщения («передача»), за ним расположатся ОС передающего ОУ, слова данных и ОС принимающего ОУ, то есть МСП будет игнорировать командное слово «приём» и ОС от принимающего ОУ.

Чтение процессором УВ принятых (отслеженных) сообщений потребует проанализировать несколько слов:

слово состояния блока сообщения для определения, принадлежит ли сообщение к типу «ОУ-ОУ»;

командное слово для определения формата («передача», «прием», команда управления, групповое сообщение и т.д.) и количества слов;

указатель на блок сообщения для определения адреса блока сообщения в буфере данных (вход в блок).

Рисунок 1.212 представляет полный перечень представления всех форматов сообщений в буфере данных «монитора сообщений».

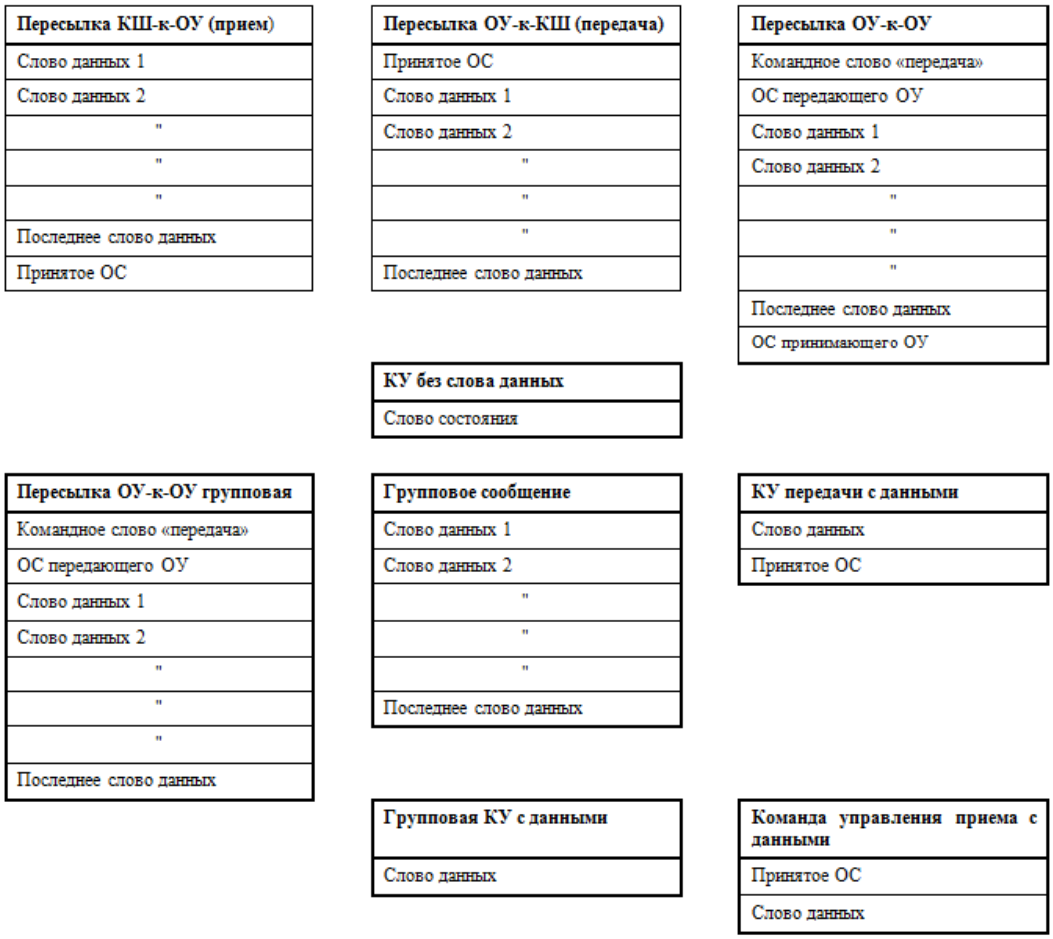


Рисунок 1.212 - Форматы сообщений в буфере данных «монитора сообщений»

Типичное распределение памяти для МСП в режиме «монитора сообщений» отображено в таблице 1.665. В этом режиме работы МСП существует несколько особенных, свойственных только ему, фиксированных местоположений ОЗУ, которые размещены таким образом, что перехлёстывания, наложения, с зарезервированными для режима ОУ ячейками не происходит.

Это позволяет осуществить комбинированный режим работы - «ОУ/монитор сообщений».

Таблица 1.665 - Типичное распределения памяти монитора сообщений (показано для 4К ОЗУ; исключительно режим монитора)

Адрес (шестн.)	Описание
0000-0101	Не используется
0102	Указатель на стек команд области А монитора сообщений (фиксированная ячейка)
0103	Указатель на буфер данных области А монитора сообщений (фиксированная ячейка)
0104-0105	Не используется
0106	Указатель на стек команд области Б монитора сообщений (фиксированная ячейка)
0107	Указатель на буфер данных области Б монитора сообщений (фиксированная ячейка)
0108-027F	Не используется
0280-02FF	Таблица определения избранных сообщений (фиксированная область)
0300-03FF	Не используется
0400-07FF	Стек команд бласти А монитора сообщений
0800-0FFF	Буфер данных области А монитора сообщений

В таблице 1.666 приведено типичное распределение памяти для комбинированного режима работы МСП «ОУ/монитор сообщений».

Фиксированное распределение памяти складывается из двух указателей на стек команд (местоположения 102 и 106 шестн.), двух указателей на буфер данных (положения 103 и 107 шестн.) и таблицы определения избранных сообщений (положения с 0280 по 02FF шестн.); задание/определение избранности сообщения основано на разрядах «Адрес ОУ», «Принять-Передать», «Подадрес/Режим управления» командного слова. Предполагается, что размер командного стека - 1 Кслов и размер буфера данных - 2 Кслов.

Таблица 1.666 - Типичное распределения памяти «ОУ/МТ сообщений» (показано для 12К ОЗУ, комбинированного режима «ОУ/монитор сообщений»)

Адрес (шестн.)	Описание
0000-00FF	Стек команд области А - ОУ
0100	Указатель на стек команд области А (фиксированная ячейка) - ОУ
0101	Резерв
0102	Указатель на стек команд области А монитора сообщений (фиксированная ячейка)
0103	Указатель на буфер данных области А монитора сообщений (фиксированная ячейка)
0104	Указатель на стек команд области Б (фиксированная ячейка) - ОУ
0105	Резерв
0106	Указатель на стек команд области Б монитора сообщений (фиксированная ячейка)
0107	Указатель на буфер данных области Б монитора сообщений (фиксированная ячейка)
0108-010F	Таблица прерываний для избранных КУ (фиксированная область) - ОУ
0110-013F	Таблица слов данных КУ (фиксированная область) - ОУ
0140-01BF	Поисковая таблица для подадресов ОУ области А (фиксированная область) - ОУ
01C0-023F	Поисковая таблица для подадресов ОУ области Б (фиксированная область) - ОУ
0240-0247	Таблица занятости подадресов (фиксированная область) - ОУ
0248-025F	Не используются
0260-027F	Блок 0 данных ОУ
0280-02FF	Таблица определения избранных сообщений (фиксированная область)
0300-03FF	Таблица допустимости КС (фиксированная область) - ОУ
0400-07FF	Стек команд области А монитора сообщений
0800-0FFF	Буфер данных области А монитора сообщений

Продолжение таблицы 1.666

Адрес (шестн.)	Описание
1000-101F	Блок 0 данных
1020-103F	Блок 1 данных
.	.
.	.
.	.
2FE0-2FFF	Блок 255 данных

Размер мониторного стека команд может выбираться из следующего перечня величин: 256, 1К, 4К и 16К. Выбор размера мониторного стека команд обеспечивается разрядами 12 и 11 регистра cfg3_. Размер мониторного буфера данных может быть выбран одним из следующего списка величин: 512, 1К, 2К, 4К, 8К, 16К, 32К или 64К слов. Выбор размера мониторного буфера данных обеспечивается разрядами с 10 по 8 регистра cfg3_.

Прерывание может вырабатываться при переполнении «мониторного буфера данных» (разряд 10 регистра состояния прерывания) или конца сообщения (разряд 0 регистра состояния прерывания).

Рисунок 1.213 демонстрирует работу в режиме «монитор сообщений». При получении достоверного командного слова МСП будет обращаться к таблице определения избранных сообщений для того, чтобы определить, разрешено ли отслеживание сообщения с данным КС. Если для текущего КС мониторинг сообщения запрещён, то МСП проигнорирует (и не сохранит в ОЗУ) текущее сообщение. Если разрешён, то МСП произведёт вход в мониторный стек команд по указателю на стек команд монитора.

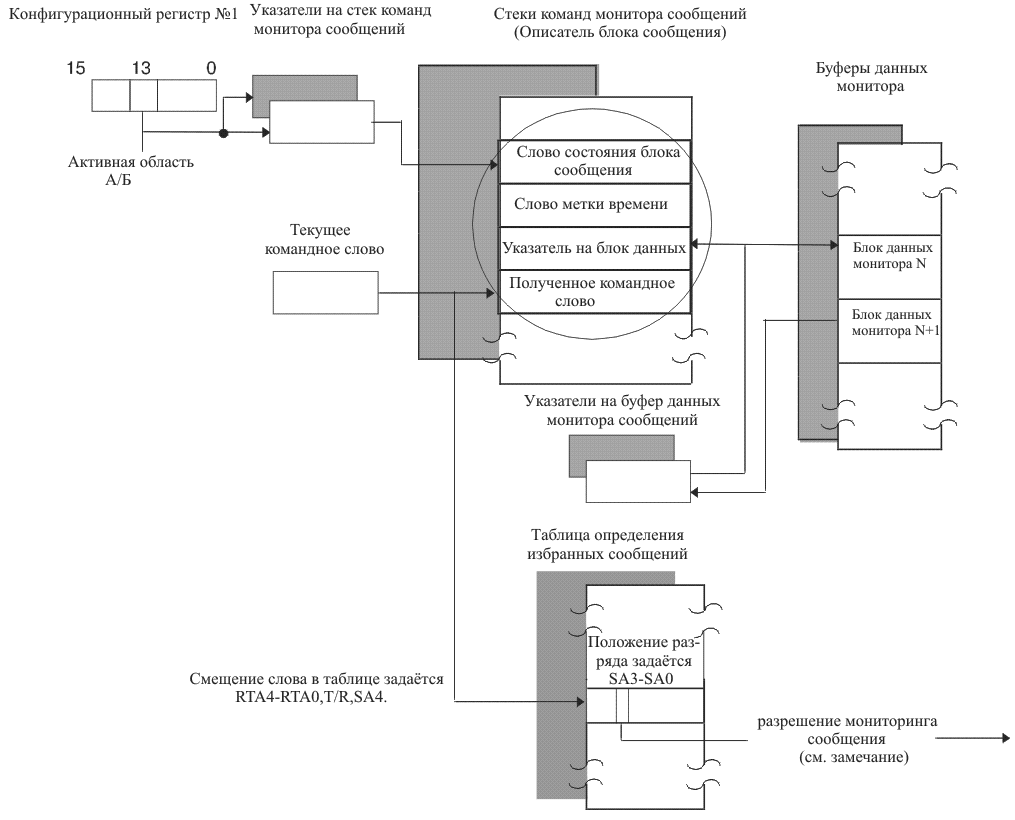


Рисунок 1.213 - Обслуживание сообщения в режиме «монитор сообщений»

Замечание. Если этот разряд установлен в лог. «0», то для соответствующего сообщения не происходит сохранение слов ни в стек команд ни в буфер данных, а также не меняются значения указателей на стек команд и буфер данных.

1.3.12.2.2.7.6.3 Слово состояния блока сообщения в режиме «монитор сообщений»

Распределение и описание разрядов слова стояния блока сообщения «монитор сообщений» показано в таблице 1.667.

Таблица 1.667 - Слово состояния блока сообщения в режиме «монитор сообщений»

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
eom	som	ch_a_b	ef	rt_rt_f	fe	rto	gdbt	dsr		wce	ist	iw	rtg_s_ae	rt_2cwe	cwce

eom (END-OF-MESSAGE): завершение отработки сообщения;
som START-OF-MESSAGE): старт отработки сообщения;
ch_a_b (channel b/a*): канал обмена B/A;
ef (Error flag): признак ошибки;
rt_rt_f (RT-to-RT FORMAT): признак пересылки типа ‘ОУ-ОУ’;
fe (Format Error): ошибка формата сообщения;
rto (Response Time out): истечение времени ожидания ответного слова ОУ;
gdbt (Good Data Block Transfer): успешная пересылка блока данных;
dsr (data stack Rollover): ’прокручивание’ буфера данных;
wce (Word Count Error): ошибка количества слов в сообщении;
ist (Incorrect Sync Type): некорректный тип синхросигнала;
iw (Invalid Word): недостоверное слово;
rtg_s_ae (RT-RT Gap/Sync/Address Error): ошибка в передаче ОУ-ОУ: интервала ожидания ОС / синхросигнала или формата / адрес ОУ;
rt_2cwe (RT-RT 2nd Command Word Error): индикатор ошибки 2-ого командного слова;
cwce (Command Word Contents Error): принятое КС не согласуется с требованиями ГОСТ Р 52070-2003;

eom (END-OF-MESSAGE) завершение отработки сообщения: Устанавливается по завершению наблюдаемого сообщения вне зависимости от наличия ошибки в сообщении.

som (START-OF-MESSAGE) старт отработки сообщения: Устанавливается приблизительно через 3-4 мкс после получения (середины разряда контроля по чётности) командного слова и очищается в конце сообщения.

channel_b_a (channel b/a*) канал обмена B/A: Этот разряд будет установлен в лог. «0», если сообщение наблюдалось по каналу A или лог. «1», если была определена пересылка по каналу B.

ef (Error flag) признак ошибки: Если этот разряд установлен в лог. «1» и разряд 10 и/или 9 установлены в лог. «1», то это указывает на ошибку формата или отсутствие отклика в сообщении. Если этот разряд установлен в лог. «1», МСП настроен на «прозрачный» режим сопряжения с УВ и разряды 10, 9, оба, установлены в лог. «0», то это указывает, что произошла ошибка подтверждения (квитирования). Ошибка подтверждения определяется в случае, когда на входе DTGRT* не устанавливается низкий уровень, либо низкий уровень устанавливается слишком поздно после момента установления в лог. «0» выходного сигнала DTREQ*. Ошибка подтверждения (квитирования) происходит также, если УВ удерживает на управляющем входе STRBD* активный уровень слишком долго после того, как МСП установил сигнал готовности READYD* в низкое логическое состояние. Если произошла ошибка подтверждения (квитирования), сообщение следует считать недостоверным. Допустимая длительность времени для ответной реакции со стороны УВ, обеспечивающей низкий уровень на входе DTGRT* или очищение STRBD* входа, составляет 9 мкс.

rt_rt_f (RT-to-RT FORMAT) признак пересылки типа ‘ОУ-ОУ’: Этот разряд будет устанавливаться, если МСП является принимающим ОУ при пересылке типа ОУ-ОУ.

fe (Format Error) ошибка формата сообщения: Если установлен, то указывает, что принятая порция сообщения содержит одно или более нарушений оценочных критериев сообщения по ГОСТ Р 52070-2003 (тип синхронизации, кодирование, контроль по чётности, количество разрядов, количество слов и т.д.).

rto (Response Time out) истечение времени ожидания ответного слова ОУ: Если установлен в лог. «1», указывает, что передающий ОУ либо не ответил, либо ответил позднее установленного срока, определяемого значением «интервала ожидания ОС». «Интервал ожидания ОС» определено стандартом как временной промежуток от момента пересечения нулевого уровня разрядом контроля по четности (середина разряда) командного слова «передача» до момента пересечения нулевого уровня разрядом синхронизации (середина разряда) ответного слова ОУ. Желаемое значение «интервала ожидания ОС» может быть выбрано посредством разрядов 10 и 9 регистра cfg3_ из следующего перечня величин: 18,5; 22,5; 50,5 и 130 мкс.

gdbt (Good Data Block Transfer) успешная пересылка блока данных: Устанавливается в лог. «1» после завершения достоверного (свободного от ошибок) сообщения; устанавливается в лог. «0» после ошибочного (недостоверного) сообщения.

dsr (Data Stack Rollover) ’прокручивание’ буфера данных: В случае лог. «1» указывает на то, что при наблюдении (отслеживании) текущего сообщения, величина указателя мониторного буфера данных была ‘прокручена’, перескочив с максимальной по значению адреса границы области буфера в ОЗУ к минимальной по значению адреса границе области буфера в ОЗУ. Размер мониторного буфера данных может устанавливаться посредством разрядов с 10 по 8 регистра cfg3_ одним из следующих значений: 512, 1K, 2K, 4k, 8k, 16k, 32k, и 64k слов.

wce (Word Count Error) ошибка количества слов в сообщении: В случае лог. «1» указывает на то, что либо КШ, либо отвечающий ОУ не передал заявленного количества слов данных.

ist (Incorrect Sync Type) некорректный тип синхросигнала: В случае лог. «1» указывает, что ОУ или КШ передал синхроимпульс типа КС/ОС в слове данных, или, наоборот, синхроимпульс, предназначенный для слова данных, был передан КШ в командном слове; либо ОУ отзывался в сообщении синхроимпульсом типа КС/ОС в слове данных, либо, наоборот, синхроимпульсом, предназначенным для слова данных, в ОС.

iw (Invalid Word) недостоверное слово: Указывает, что КШ передал сообщение (или ОУ отозвался) с одним или более словом, содержащим одну или более ошибку следующего типа: ошибка поля синхроимпульса, ошибка кодирования, ошибка контроля по четности и/или ошибка количества разрядов.

rtg_s_ae (RT-RT Gap/Sync/Address Error) ошибка в передаче ОУ-ОУ: интервала ожидания ОС / синхросигнала или формата / адрес ОУ: Этот разряд устанавливается, если происходит что-либо из следующего перечня событий:

(1) Если gce , 8-ой разряд регистра cfg5_ - «разрешение проверки минимальной паузы перед выдачей ОС» - установлен в лог «1» и передающий ОУ отвечает ОС через время меньшее, чем 4 мкс (согласно ГОСТ Р 52070-2003 - это промежуток между серединами зон разряда контроля по четности последнего КС/СД и синхроимпульса ОС), т.е. реально проверяется, что период отсутствия декодируемых сигналов в ЛПИ (‘мёртвое время’) меньше 2 мкс;

и/или

(2) определена ошибка типа синхросигнала или формата (ошибка кодирования, количества разрядов и/или контроля по чётности) в предаваемом ОУ ОС;

и/или

(3) значение в поле «Адрес ОУ» в передаваемом ОС не равно значению в поле «Адрес ОУ» в командном слове «передача».

rt_2cwe (RT-RT 2nd Command Word Error) индикатор ошибки 2-ого командного слова: для информационной пересылки данных типа «ОУ-ОУ», будучи установленным в лог. «1», указывает на наличие одной или более из следующих ошибок во втором, из двух смежных,

командном слове: 1) разряд «Приём-передача» установлен в лог. «0»; 2) подадрес равен 5’b00000 или 5’b11111 и/или значение в поле «Адрес ОУ» такое же как и значение в поле «Адрес ОУ» в командном слове «приём».

cwce (Command Word Contents Error) принятое КС не согласуется с требованиями ГОСТ Р 52070-2003: Указывает, что принятое КС не согласуется с требованиями ГОСТ Р 52070-2003. Это касается следующих неопределенных разрядных комбинаций в командных словах:

1) «запрет групповых сообщений» (bd, 7-ой разряд регистра cfg5_) установлен в лог. «0» и командным словом явилось КС, не сообщения КУ, групповой обмен «передача»;

2) разряд ome («обрабатывать КУ на «прием» без СД как резервные КУ» регистра cfg3_ установлен в лог. «0» и сообщение с разрядом «Приём-передача» в лог. «0», со значением поля «Подадрес/Режим управления» равным 5’b00000 или 5’b11111 (т.е. КУ), со значением поля «Число СД/Код команды» между 5’b00000 или 01111;

3) разряд bd («запрет групповых сообщений») регистра cfg5_ установлен в лог. «0» и КУ, для которой не разрешен групповой обмен, была послана по направлению группового адреса (5’b11111).

1.3.12.2.2.7.6.4 Вытеснение (замещение) команд

Последовательность вытеснения сообщения происходит, когда достоверное, вовлечённое в обработку сообщение, пересылка которого осуществлялась по одному каналу ЛПИ, по истечении времени, отпущенного сообщению, не закончившись, продолжается появлением нового КС по другому, резервному каналу ЛПИ. Если оба, старое и новое КС, направлены по одному и тому же «Адресу ОУ», то ОУ обязан остановить обработку первого, старого, сообщения и приступить к обработке нового. Если новое КС направлена по другому адресу ОУ, то ОУ, отвечающий на первое КС, должен довести обработку сообщения до конца, игнорируя активность на другом канале.

МСП в режиме «монитора сообщений» будет прекращать обработку первого сообщения и приступать к обработке нового сообщения. Единственным исключением этого будет случай, когда МСП работает в комбинированном режиме «ОУ/ Монитор сообщений» и первая команда направлялась по собственному «Адресу ОУ», а вторая - по адресу другого ОУ (не по групповому адресу). Как ОУ, МСП должен продолжить обработку сообщения с первым КС (в соответствии с ГОСТ Р 52070-2003) и игнорировать новое сообщение, направленное другому ОУ, по другому каналу ЛПИ.

Монитор сообщений МСП будет всегда замещать сообщения по тому же самому каналу ЛПИ за исключением случая, когда вытесняющая команда может быть интерпретирована как ОС. Следовательно, ответный отклик ОУ с неправильным адресом ОУ в ОС не будет определяться МСП, в режиме монитора сообщений, в качестве нового КС, но будет распознаваться как неправильное ОС.

При вытеснении сообщения обычная SOM последовательность действий производится для нового вытесняющего сообщения. Отметим, что для старого, незаконченного, вытесняемого сообщения EOM последовательность действий не выполняется. Это будет приводить к тому, что разряды SOM и EOM слова состояния блока сообщения в стеке команд будут оказываться установленными в лог. «1» и лог. «0» соответственно. УВ может различить между собой «сообщение в обработке» и «вытесненное сообщение» по значению указателя на стек команд монитора. Если указатель на стек команд монитора был увеличен и слово состояния блока сообщения указывает на SOM последовательность, тогда сообщение, по-видимому, было замещено.

1.3.12.2.2.7.7 SOM и EOM последовательности действий в режиме «монитор сообщений»

Приблизительно через 1,25 мкс после середины разряда контроля по четности (момент пресечения сигналом нулевого уровня) принимаемого командного слова МСП производит последовательность действий в режиме «монитор сообщений» «старт отработки сообщения» - SOM (Start-of-Message). После выполнения первых двух шагов «SOM - последовательности»

МСП становится понятно, задан мониторинг этого сообщения или нет, то есть, нужно ли его сохранять в ОЗУ. Задание мониторинга определяется по значению КС: является функцией от значений поля «Адрес ОУ», разряда «Прием-передача» и поля «Подадрес/Режим управления» принимаемого командного слова. Примерно через 6 мкс после окончания приема последнего слова для избранного сообщения МСП выполняет последовательность действий «завершение отработки сообщения» EOM (End-of-Message). Мониторные SOM и EOM последовательности операций содержат несколько процедур чтения и записи в разделяемое с УВ ОЗУ.

1.3.12.2.2.7.7.1 SOM (Start-of-Message) последовательность действий

Последовательности действий SOM и EOM подробно расписаны ниже:

0) Если разряд icoe (13-ый разряд регистра cfg6_) равен лог. «1» и произошло вытеснение(замещение) команды, то значение указателя на стек команд, увеличивается на четыре и записывается в местоположение указателя на стек команд области А/Б.

Соответствующее значению группы разрядов принятого командного слова - «Адрес ОУ», «Прием-передача» и «Подадрес/Режим управления», - заданное при конфигурировании слово читается из таблицы определения избранных сообщений. Если в результате анализа слова из таблицы определения избранных сообщений становится очевидным, что сообщение должно сохраняться в памяти, то продолжением данного шага становится шаг №2. Если сообщение не является «избранным», то SOM последовательность завершается;

Указатель на стек команд мониторный читается из местоположения, предназначенного для хранения указателя на стек команд области А/Б (по адресу 102 или 106);

Указатель на блок данных мониторного читается из местоположения указатель на буфер данных области А/Б (либо из местоположения 103, либо из 107 в зависимости от того, какая из областей является активной);

Командное слово записывается в четвертое местоположение текущего описателя блока сообщения;

Указатель на блок данных записывается в третье слово текущего описателя блока сообщения. Эта величина используется для определения первой позиции в мониторном буфере данных для расположения текущего сообщения;

Значение регистра счетчика метки времени записывается во вторую позицию описателя блока сообщения;

Слово состояния блока сообщения записывается в первое местоположение описателя блока сообщения;

Если разряд icoe (13-ый разряд регистра cfg6_) равен лог. «0» , то значение указателя на стек команд, считанное во втором пункте, увеличивается на четыре и записывается в указатель на стек команд области А/Б. Если разряд icoe (13-ый разряд регистра cfg6_) равен лог. «1», то шаг 8 не выполняется, и увеличение значения указателя на стек команд на четыре происходит не при SOM, а при EOM последовательности (шаг 4 EOM-последовательности).

1.3.12.2.2.7.7.2 EOM (End-of-Message) последовательность действий

Адрес ячейки памяти последнего слова, сохраняемого в буфере данных, увеличивается на единицу (модуль размера буфера данных) и записывается в местоположение указатель на буфер данных области А/Б;

«Метка времени» сохраняется во втором местоположении описателя блока сообщения;

Слово состояния блока сообщения записывается в первое местоположение в описателя блока сообщения;

Если разряд icoe (13-ый разряд регистра cfg6_) равен лог. «1» , то значение указателя на стек команд, считанное на втором шаге SOM-последовательности, увеличивается на четыре и записывается в местоположение указатель на стек команд области А/Б.

1.3.12.2.2.7.8 Пример работы в режиме «монитор сообщений»

Подразумевается: 4К ОЗУ, использование прерывания по событию «завершение отработки сообщения».

Все значения слов МСП приводятся в шестнадцатеричной системе счисления.

1.3.12.2.2.7.8.1 Инициализация (конфигурирование и запуск в работу)

Записать 0001 в регистр запуска/сброса для сброса устройства.

Записать 8000 в регистр cfg3_ - разрешение режима «расширенных возможностей конфигурирования» МСП.

Записать 8D00 в регистр cfg3_ - установление размера стека команд монитора в 1К слов и размера буфера данных монитора в 2К слов.

Записать 5000 в регистр cfg1_ - установление устройства в режим «монитор сообщений».

Записать 0400 в ОЗУ по адресу 0102 - инициализация указателя на стек команд мониторного; значение 0400 будет указывать (адресовать) на описатель блока сообщения, которое должно быть обслужено первым.

Записать 0800 в ОЗУ по адресу 0103 - инициализация указателя на буфер данных мониторного.

Инициализировать таблицу определения избранных сообщений, значения в области ОЗУ адресов с 0280 по 02FF (шестн.). Заметим, что запись значения FFFF по всем адресам таблицы вызовет отслеживание МСП всех команд.

Записать 0010 в регистр cfg2_ - разрешение функции «автоматический сброс запроса прерывания и значения регистра isr».

Записать 0001 в регистр imr - разрешение прерывания по событию «завершение отработки сообщения».

Записать 0002 в регистр запуска/сброса - запуск работы.

Замечание - Порядок, в котором отрабатываются шаги с 3 по 9 безразличен.

1.3.12.2.2.7.8.2 Технические приемы обработки принимаемых сообщений МКПД при работе МСП в режиме «монитор сообщений»

Предполагается, что выполнены действия п. 8.8.1. Запись mod stack size означает выполнение операции ‘по модулю’ размера стека команд (в нашем примере размер стека команд – 1К 16-ти разрядных слов).

Начальные значения переменных STKPTR и MTCMDPTR устанавливаются равными значению указателя на стек команд монитора (местоположение 102 для активной области ОЗУ А), т.е. равными 400.

Определим процедуру обработки прерывания, которая будет выполнять следующее:

а) Чтение и сохранение значения регистра состояния прерываний isr.

б) Проверку значения регистра состояния прерывания, сравнивая с величиной 0001 (происшествие события «завершение отработки сообщения»);

в) Чтение и сохранение значения ячейки памяти с адресом 0102 (указатель на стек команд монитора). Это значение будет поименовано как MTCMDPTR;

г) повторение следующих действий в цикле:

ПОКА

((STKPTR=MTCMDPTR)

или (((STKPTR + 4) mod stack size)=MTCMDPTR) и (слово состояния блока сообщения имеет установленным признак ЕОМ))

ВЫПОЛНЯТЬ ЦИКЛ

- Прочесть значение слово состояния блока сообщения по адресу ОЗУ равным (значение STKPTR).
- Прочесть значение «метка времени» по адресу ОЗУ равным (значение STKPTR) +1.
- Прочесть значение указатель на блок данных по адресу ОЗУ равным (значение STKPTR) +2.
- Прочесть значение «КС» по адресу ОЗУ равным (значение STKPTR) +3.

- ОС и данные сохраняются последовательно в ОЗУ, начиная с местоположения, адрес которого определяется величиной значение указатель на блок данных, число слов и формат сообщения определяются АНАЛИЗОМ 11-го разряда слова состояния блока сообщения и содержимым КС. Если 11-ый разряд в слове состояния блока сообщения установлен в лог. «1», то в сообщении была пересылка типа «ОУ-ОУ». Вдобавок при чтении слов необходимо проверить, не произошло ли ‘прокручивание’ буфера данных;
- Сохранить и/или использовать информацию сообщениия согласно требованиям конкретного применения.
- STKPTR = ((STKPTR + 4) mod stack size). Увеличение для следующего сообщения, которое должно быть обслужено с ‘прокручиванием’ в 1 Кслов.

ЗАВЕРШИТЬ ЦИКЛ

д) выход из процедуры обслуживания прерываний.

1.3.12.2.2.8 Взаимодействие с СФ-блоком МКО

В данном разделе представлены рекомендации по сопряжению СФ-блока МКО с приемопередатчиками, подробно описываются временные диаграммы работы взаимодействия с УВ.

1.3.12.2.2.8.1 Сопряжение СФ-блока МКО с приемопередатчиками.

Кодер и декодеры (Кодек) МСП обеспечивают взаимодействие с приёмопередатчиками согласно требованиям интерфейсов типа «Smiths» или «Harris».

Правила взаимодействия с кодером для «Smiths-совместимых» и для «Harris-совместимых» приёмопередатчиков одинаковые и представлены на рисунке 1.213. Правила взаимодействия с декодером для «Smiths-совместимых» и для «Harris-совместимых» приёмопередатчиков различны. Различие заключается в том, что «Harris-совместимые» приёмопередатчики «молчание в линии» (отсутствие сигнала) «отображают» нулевыми логическими уровнями на прямом и обратном выходах приёмника, в то время как «Smiths-совместимые» приёмопередатчики – единичными. Правила взаимодействия с декодером для «Smiths-совместимых» и для «Harris-совместимых» приёмопередатчиков показаны на рисунке 1.213.

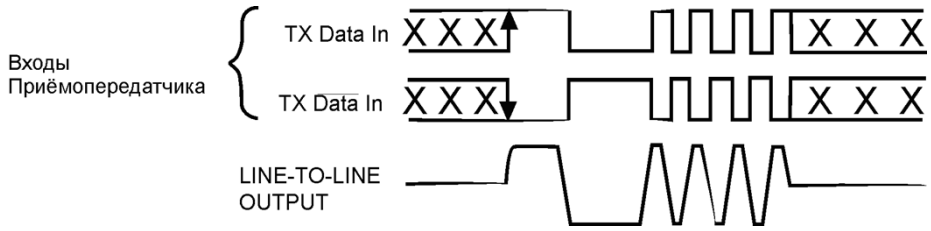


Рисунок 1.213 - Правила взаимодействия приёмопередатчиков с кодером

Замечания к рисунку 1.213:

(1) TX Data In являются логическими входными сигналами приёмопередатчика, посредством которых происходит взаимодействие приёмопередатчика с кодером.

(2) TX Data In должны иметь противоположные логические уровни во время передачи и одинаковые логические уровни, когда передача отсутствует. При одновременном переключении логических значений сигналов TXA/TXA_ (TXB/TXB_) наибольшее значение интервала между моментами переключения сигналов TXA и TXA_ (TXB и TXB_) на выходных сигналах СФ-блока МКО не превышает величины порядка 2 нс.

(3) LINE-TO-LINE output – выходной сигнал, поступающий в МКПД с аналоговых входов/выходов приёмопередатчика.

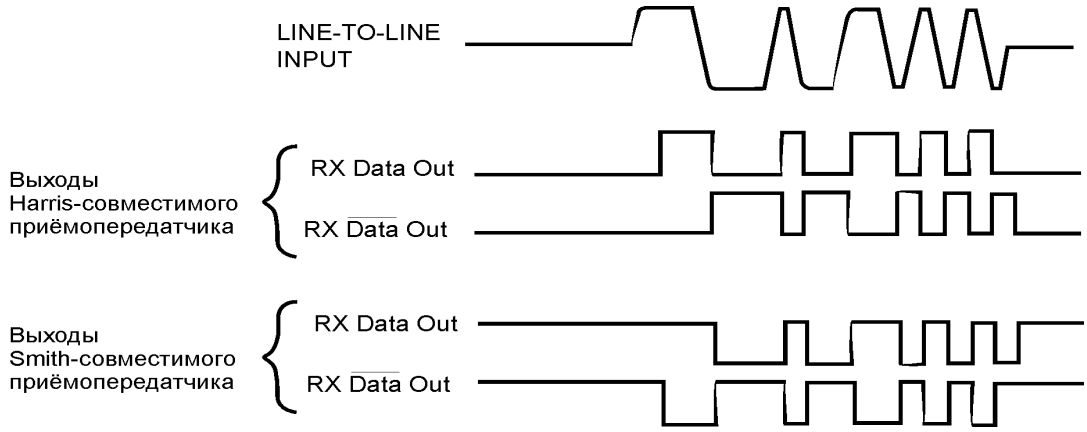


Рисунок 1.214 - Правила взаимодействия приёмопередатчиков с декодером

Замечания к рисунку 214:

(1) RX Data Out являются логическими выходными сигналами приёмопередатчика, посредством которых происходит взаимодействие приёмопередатчика с декодером. В связи с тем, что сигналы приемника являются асинхронными по отношению к тактовой частоте МСП, для достоверного распознавания декодерами входной информации выбираемый приемопередатчик должен обеспечивать длительность активного уровня сигналов, подаваемых на входы RXA, RXA_, RXB, RXB_ МСП, не меньшую длительности периода используемого значения тактовой частоты 16 МГц. Для достоверного распознавания момента завершения, принимаемого декодером сообщения, для последнего принимаемого слова перед паузой в МКПД выбираемый приемопередатчик должен обеспечивать длительность последнего интервала активного уровня соответствующего сигнала приемника, не превышающую 1300 нс.

(2) LINE-TO-LINE input – входной сигнал, поступающий из МКПД на аналоговые входы/выходы приёмопередатчика.

Приемопередатчики МКПД могут быть выполнены на микросхемах различных фирм производителей, в частности на изделиях с одним каналом приёма/передачи, например, таких как BUS-63102, BUS-8559 (фирма DDC), ВА-996А (ОАО «Авангард») и прочие. Кроме того, МСП может сопрягаться с приемопередающими устройствами, обеспечивающими взаимодействие с дублированным МКПД (с основным и резервным каналами), например, изделиями BUS-63127, BUS-631237 (фирма DDC), HI-1573 (фирма «HOLT INTEGRATED CIRCUITS»), ППУ-5 (НПО Автоматики, г. Екатеринбург). В случае одноканальных приемопередатчиков следует использовать два однотипных изделия.

Тип используемого трансформатора гальванической развязки определяется выбором конкретного приемопередатчика. Подключение к МКПД может быть выполнено как по схеме с согласующим трансформатором, так и непосредственным соединением.

Схема включения СФ-блока МКО с использованием приёмопередатчиков типа HI-1573 фирмы «HOLT INTEGRATED CIRCUITS» приведена на рис. 9.3 Цифровые «выходы передачи информации в МКПД» TXA, TXA_, TXB, TXB_ СФ-блока МКО напрямую подсоединяются к соответствующим входам приемопередатчиков микросхемы D2 (TXA, NTXA, TXB, NTXB). Цифровые «входы приема информации из МКПД» RXA, RXA_, RXB, RXB_ СФ-блока МКО напрямую подсоединяются к соответствующим выходам приемопередатчиков микросхемы (RXA, NRXA, RXB, NRXB). «Цифровые выходы запрета передачи в МКПД» TX_INH_A, TX_INH_B СФ-блока МКО подсоединяются к соответствующим входам TXINHA, TXINHB приемопередатчиков микросхемы. Лог. «1» на входах TXINHA, TXINHB запрещает передачу приёмопередатчику по соответственному каналу МКПД. Лог. «1» на входах RXENA, RXENB разрешает активность на выходах приёмника (RXA, NRXA, RXB, NRXB) для соответственного канала, тогда как лог. «0» переводит выходы приёмника для соответственного канала МКПД микросхемы в состояния лог. «0».

В качестве трансформаторов гальванической развязки T1,T2 в схеме используется трансформатор типа PM-DB2745L, который обладает следующими характеристиками:

- коэффициенты трансформации:
- 1-3:4-8 1:2,5
- 1-3:5-7 1:1,79
- импеданс, при частоте синусоидального входного сигнала от 75 кГц до 1 МГц, не менее: 4000 Ом.

Тому следует быть отмеченным особенно, что по рекомендации фирмы HOLT выводы TXINHA, TXINHB микросхемы D2 должны быть утянутыми через сопротивления с номиналом 10 кОм к напряжению питания VDD приемопередатчика, а выводы «земля» и «питание» каждого канала микросхемы приемопередатчика необходимо «шунтировать» парой фильтрующих конденсаторов: керамическим с номиналом 0,1 мкФ, и танталовым с номиналом 10 мкФ, располагая их по возможности ближе к указанным выводам.

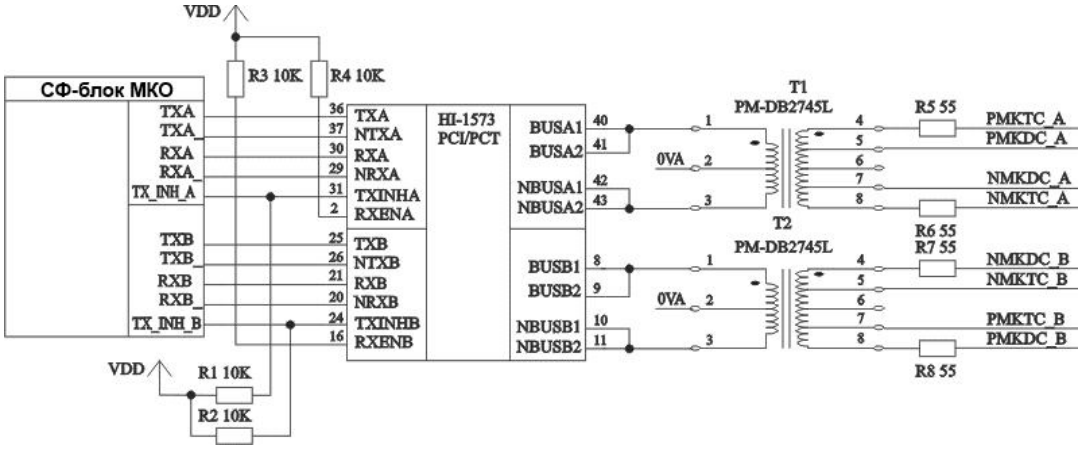


Рисунок 1.668 - Схема включения СФ-блока МКО с использованием приёмопередатчиков типа HI-1573

1.3.12.2.2.8.2 Взаимодействие СФ-блока МКО с управляющим вычислителем

СФ-блок МКО обладает значительной гибкостью для сопряжения с управляющим вычислителем. Доступ к ресурсам МСП и ОЗУ осуществляется с помощью конфигурационного интерфейса. ОЗУ также доступен по средствам интефейса доступа к памяти. Эти интерфейсы полностью независимы и обращения по одному не влияют на работу другого.

1.3.12.2.2.8.3 Входной сигнал сброса (prst_n*)

СФ-блок МКО может быть установлен в начальное рабочее состояние (или состояние по включению питания) после выполнения либо аппаратного, либо программного сбросов. Аппаратный сброс, который, как правило, всегда производят сразу после включения питания, выполняют установлением и удержанием лог. «0» на входе prst_n* в продолжение, по крайней мере, 200нс, и последующим снятием (переводом в лог. «1»).

Описание состояний внутренней логики после сброса или получения ОУ КУ ‘Установить ОУ в исходное состояние’ сведено в таблицу 1.668.

Таблица 1.668 - Состояния после сброса

	После аппаратного сброса (вход prst_n*)	После программного сброса (разряд 0 регистра запуска/сброса)	После принятия КУ ‘Установить ОУ в исходное состояние’
Обработка сообщения по ГОСТ Р 52070-2003	Обработка сообщения прекращается	Обработка сообщения прекращается	Обработка сообщения продолжается
Регистр маски прерывания	Очищается в 0000	Очищается в 0000	Не изменяется
Регистры конфигурирования	Все сбрасываются в 0000	Все сбрасываются в 0000	Не изменяются
Состояние блокирования передатчиков (от КУ), состояние блокирования признака «Неисправность ОУ», и флагов «Ошибка в сообщении» и «Принята групповая команда».	Передатчики разблокируются (если были заблокированы); признак «Неисправность ОУ» разблокируется (если был заблокирован); признаки «Ошибка в сообщении» и «Принята групповая команда». ОС сбрасываются в лог. «0».	Передатчики разблокируются (если были заблокированы); признак «Неисправность ОУ» разблокируется (если был заблокирован); признаки «Ошибка в сообщении» и «Принята групповая команда» ОС сбрасываются в лог. «0».	Передатчики разблокируются (если были заблокированы); признак «Неисправность ОУ» разблокируется (если был заблокирован); признаки «Ошибка в сообщении» сбрасывается в лог. «0», если команда ‘Установить ОУ в исходное состояние’ была установлена как законная (допустимая). Признак «Принята групповая команда» сбрасывается в лог. «0», если КУ Установить ОУ в исходное состояние’ не была принята по адресу 5’b11111. Разряды с 15 по 9 регистра слова ВСК сбрасываются в лог. «0».

1.3.12.2.2.8.4 Временные диаграммы работы конфигурационного интерфейса

Временные диаграммы доступа УВ ко внутренней памяти или регистрам МСП по средствам конфигурационного интерфейса приведены на рисунках 1.213 и 1.214. На рисунке 1.213 показаны временные диаграммы цикла чтения, на рисунке 1.214 - цикла записи.

Таблица 1.669 – УВ читает ОЗУ/Регистры

	Описание
tS	Время предустановки достоверного адреса addr[12:0] и установка сигнала valid в лог. 1
tH	Время удержания достоверного адреса addr[12:0] и сигнала valid
trd	Время удержания СФ-блоком МКО низкого уровня сигнала ready – длительность цикла чтения (замечание 4)
trdata	Время установления достоверных данных
S0	Такт установки достоверного адреса и активного значения сигнала valid (лог.1)
S1	Такт начала цикла чтения. Сигнал ready принимает активное значение, значения адреса и сигналов valid и wr_rd незначимы
S2	Такт окончания цикла чтения. Сигнал ready принимает неактивное значение, считанные данные выставляются на шину rdata[15:0]
S3	Такт считывания данных УВ с шины rdata[15:0]

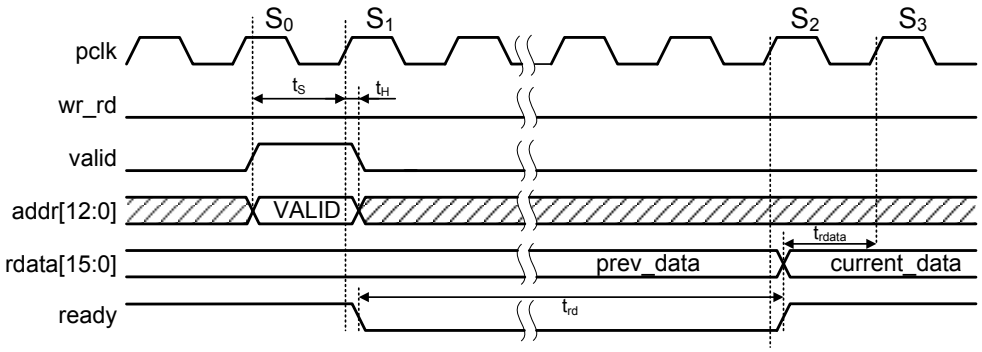


Рисунок 1.213 - УВ читает ОЗУ/Регистры

Замечания для таблицы 1.669 и рисунка 1.213:

- wr_rd должен быть высоким для осуществления записи, и низким для чтения.
- addr[12] должен быть высоким для доступа к памяти, и низким для доступа к регистрам.
- ОЗУ адресуется входами с addr[11] по addr[0]. Регистры адресуется входами с addr[4] по addr[0].

Длительность цикла чтения ОЗУ/ Регистров не превышает 500 нс.

Таблица 1.670 – УВ записывает ОЗУ/Регистры

	Описание
tS	Время предустановки достоверного адреса addr[12:0], данных wdata[15:0] и установка сигнала valid и wr_rd в лог. 1
tH	Время удержания достоверного адреса addr[12:0], данных wdata[15:0] и сигнала valid и wr_rd
twr	Время удержания СФ-блоком МКО низкого уровня сигнала ready – длительность цикла чтения (замечание 4)
S0	Такт установки достоверного адреса addr[12:0], данных wdata[15:0] и активного значения сигнала valid (лог.1) и wr_rd (лог.1)
S1	Такт начала цикла записи. Сигнал ready принимает активное значение, значения адреса, данных и сигналов valid и wr_rd незначимы
S2	Такт окончания цикла записи. Сигнал ready принимает неактивное значение

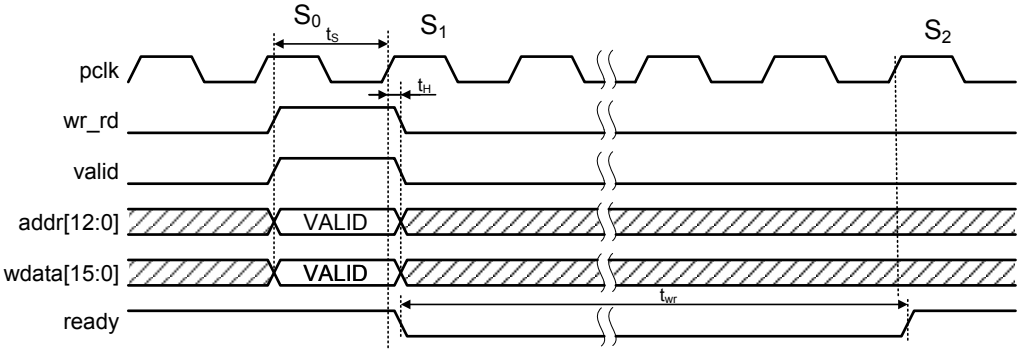


Рисунок 1.214 - УВ записывает ОЗУ/Регистры

Замечания для таблицы 1.670 и рисунка 1.214:

wr_rd должен быть высоким для осуществления записи, и низким для чтения.

addr[12] должен быть высоким для доступа к памяти, и низким для доступа к регистрам.

ОЗУ адресуется входами с addr[11] по addr[0]. Регистры адресуется входами с addr[4] по addr[0].

Длительность цикла записи ОЗУ/ Регистров не превышает 500 нс.

1.3.12.2.2.8.5 Временные диаграммы работы интерфейса доступа к памяти

Временные диаграммы доступа УВ ко внутренней памяти по средствам интерфейса доступа к памяти приведены на рисунках 1.215 и 1.216 На рисунке 1.215 показаны временные диаграммы цикла чтения, на рисунке 1.216 - цикла записи.

Таблица 1.671 – УВ читает ОЗУ

	Описание
tS	Время предустановки достоверного адреса mem_addr[11:0] и установка сигнала mem_en в лог. 1
tH	Время удержания достоверного адреса mem_addr[11:0] и сигнала mem_en
tmem_rd	Время удержания СФ-блоком МКО низкого уровня сигнала mem_ready – длительность цикла чтения памяти (замечание 2)
tmem_rdata	Время установления достоверных данных
S0	Такт установки достоверного адреса и активного значения сигнала mem_en (лог.1)
S1	Такт начала цикла чтения. Сигнал mem_ready принимает активное значение, значения адреса и сигналов mem_we и mem_en незначимы
S2	Такт окончания цикла чтения. Сигнал mem_ready принимает неактивное значение, считанные данные выставляются на шину mem_rdata[15:0]
S3	Такт считывания данных УВ с шины mem_rdata[15:0]

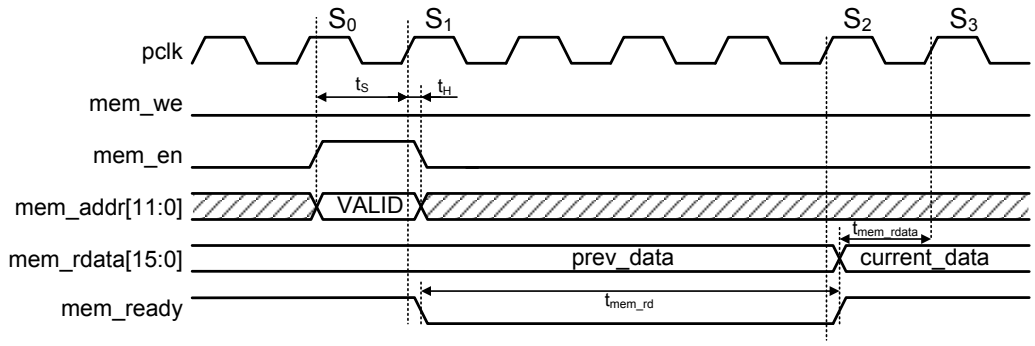


Рисунок 1.215 - УВ читает ОЗУ/Регистры

Замечания для таблицы 1.671 и рисунка Рисунок 1.:

mem_we должен быть высоким для осуществления записи, и низким для чтения.

Длительность цикла чтения ОЗУ не превышает 40 нс.

Таблица 1.672 – УВ записывает ОЗУ

	Описание
tS	Время предустановки достоверного адреса mem_addr[11:0], данных mem_wdata[15:0] и установка сигнала mem_we и mem_en в лог. 1
tH	Время удержания достоверного адреса mem_addr[11:0], данных mem_wdata[15:0] и сигнала mem_en и mem_we
tmem_wr	Время удержания СФ-блоком МКО низкого уровня сигнала mem_ready – длительность цикла чтения (замечание 2)
S0	Такт установки достоверного адреса mem_addr[11:0], данных mem_wdata[15:0] и активных значений сигналов mem_we (лог.1) и mem_en (лог.1)
S1	Такт начала цикла записи. Сигнал mem_ready принимает активное значение. Значения адреса, данных и сигналов mem_we и mem_en незначимы
S2	Такт окончания цикла записи. Сигнал mem_ready принимает неактивное значение

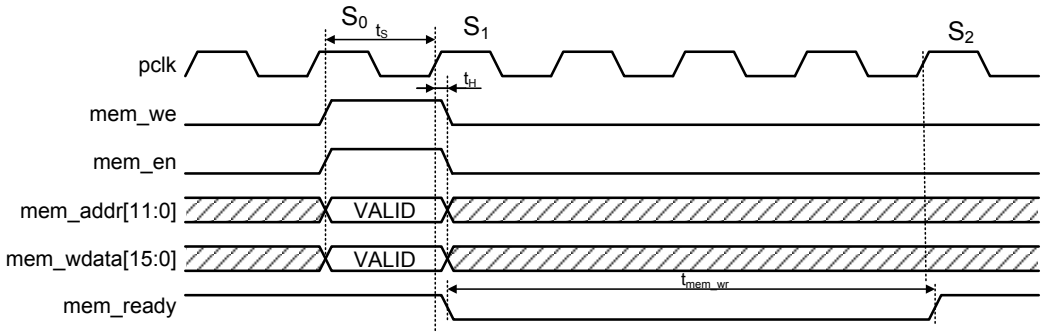


Рисунок 1.216 - УВ записывает ОЗУ/Регистры

Замечания для таблицы 1.672 и рисунка 1.216:
wr_rd должен быть высоким для осуществления записи, и низким для чтения.
Длительность цикла записи ОЗУ/ не превышает 40 нс.

1.3.12.2.2.9 Техническая терминология, сокращения

Перед изучением данного документа рекомендуется ознакомиться с ГОСТ Р 52070-2003. Зарубежный аналог ГОСТ Р 52070-2003 – стандарт MIL-STD-1553B, Notice 2.

Для основных, используемых в настоящем ТО, сокращений и терминов даны следующие определения:

- машина связная последовательной шины данных (МСП) – часть функциональной логики СФ-блока МКО, отвечающая за обмен и контроль информации передаваемой и принимаемой по МКПД.
- управляющий вычислитель (УВ), центральный процессор (ЦП) – управляющий вычислитель/центральный процессор предназначен для взаимодействия с СФ-блоком МКО по конфигурационному интерфейсу и интерфейсу доступа к внутренней памяти.
- линия передачи информации (ЛПИ), мультиплексный канал передачи данных (МКПД) – мультиплексный канал передачи данных (ГОСТ Р 52070-2003). МСП не имеет собственных, встроенных, приёмопередатчиков, поэтому для сопряжения с

ЛПИ необходимо использовать внешние приёмопередающие устройства (приёмопередатчики, ППУ).

- проверочный возврат - приём МСП передаваемой в ЛПИ информации в целях самотестирования;
- допустимость - то же, что и (не)допустимость по ГОСТ Р 52070-2003;
- особое состояние ОУ - ‘неожидаемое’ значение для хотя бы одного из одинадцати признаков полученного ОС в режиме КШ. Ожидаемыми значениями для этих одинадцати признаков являются лог. «0» за следующими исключениями: 1) Признаки ОС установленные в лог. «1» замаскированы соответствующими разрядами слова управления сообщением КШ. 2) В некоторых состояниях конфигурации МСП ожидаемым значением для признака ОС «Принята групповая команда» становится лог. «1», а не лог. «0». Это последнее исключение верно для каждого из следующих состояний конфигурации МСП: МСП находится в состоянии «нерасширенные возможности конфигурирования» (15-ый разряд (eme, 15-ый разряд регистра cfg3_, установлен в лог. «0») или МСП находится в состоянии «расширенные возможностеи конфигурирования» (eme, 15-ый разряд регистра cfg3_, установлен в лог. «1»), и разряд bm_ex регистра cfg4_ установлен в лог. «0», и разряд «маскирование групповых сообщений» в слове управления сообщением КШ установлен в лог. «1».
- начало отработки сообщения (START-OF-MESSAGE) - SOM-последовательность действий, пересылок в/из разделяемого ОЗУ, которые производит МСП в режимах КШ, ОУ и «монитор сообщений» во время старта обработки сообщения;
- завершение отработки сообщения (END-OF-MESSAGE) - EOM-последовательность действий, пересылок в/из разделяемого ОЗУ, которые производит МСП в режимах КШ, ОУ и «монитор сообщений» во время завершения обработки сообщения;
- стек команд - кольцевой буфер с инкрементом указателя (указатель на стек команд) в задаваемой области ОЗУ; предназначен для управления последовательностью отработки сообщений (в режиме КШ) или для фиксации последовательности отработки сообщений (в режиме ОУ, «монитор сообщений»); единицей записи в стек команд является четырехсловный описатель блока сообщения; значение указателя на стек команд доступно УВ чтением регистра указателя на стек команд;
- кадр сообщений (кадр) - пакет из последовательности сообщений, размещённых в ОЗУ и отрабатываемый КШ практически без вмешательства УВ;

1.3.12.2.3 Контроллер интерфейса I2C

Последовательная шина I2C имеет 2 линии (SDA – данных и SCL – тактового сигнала), контроллер I2C может производить обмен по данной шине в режиме standard mode со скоростями до 100 кбит/с, в режиме fast mode со скоростями до 400 кбит/с, а также в режиме high-speed со скоростями до 3.4 Мбит/с. Контроллер может выступать либо в роли ведущего устройства, либо в роли ведомого – тип задаётся программно. Данный контроллер имеет системный интерфейс APB, поэтому подключается в шинную систему AHB через мост.

1.3.12.2.3.1 Введение

I2C ядро совместимо стандартом Philips и поддерживает 7 и 10-битную адресацию. Непосредственно поддерживаются Стандартный режим(100Кб/сек) и Быстрый режим (400 Кб/сек). Внешними резисторами утяжки должны быть снабжены обе линии шины.

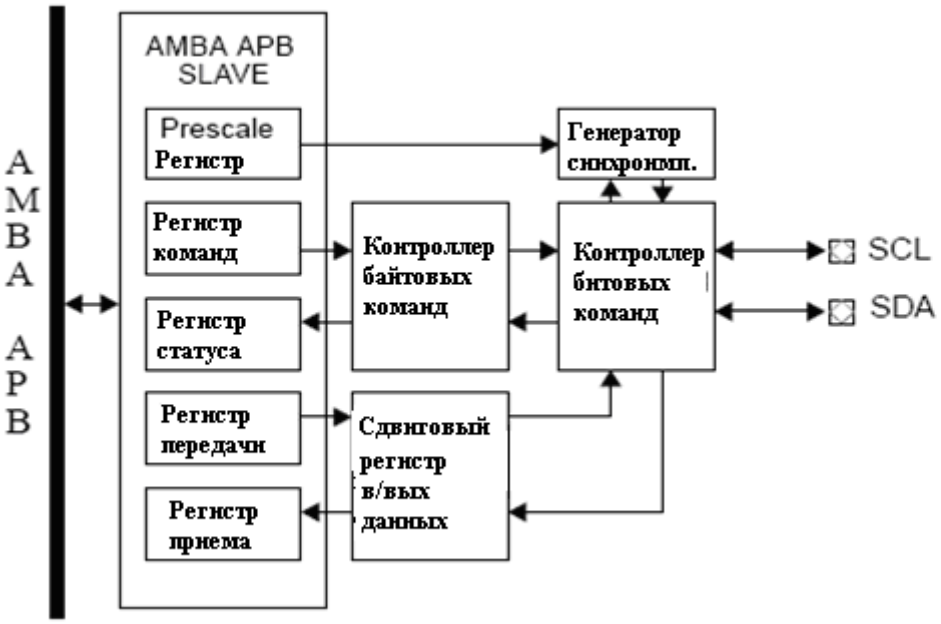


Рисунок 1.217 – Блок-схема I2C- Мастера

1.3.12.2.3.2 Описание функционирования блока

1.3.12.2.3.2.1 Протокол передачи

I2C-шина – простая 2-х-проводная последовательная шина с многими мастерами с обнаружением столкновений и арбитражем. Шина состоит из последовательной линии данных SDA и линии последовательных синхросигналов SCL..I2C-стандарт определяет 3 скорости передачи: Стандартную(100Кб/сек), Быструю (400Кб/сек) и Высокоскоростную (3,4 Mb/сек).

Передача по I2C-шине начинается с условия СТАРТ. Условие СТАРТа определяется как переход из высокого в низкий уровень SDA линии, в то время как на SCL – высокий уровень. Эти условия всегда генерируются мастером. Шина считается занятой после СТАРТ-условия и свободна после некоторого количества времени после СТОП-условия. Свободное время шины , требуемое между СТОП и СТАРТ условием определено в I2C спецификации и зависит от скорости цифрового потока (bit rate) шины.

Рисунок 1. показывает передачу данных, имеющую место на I2C шине. Мастер сначала генерирует условие СТАРТа и затем передает 7-битный SLAVE-адрес. Бит, следующий за SLAVE-адресом – $R/\overline{W}$-бит , который определяет направление передачи данных. В этом случае $R/\overline{W}$ равен 0, индицируя операцию записи. После того как мастер передал адрес и $R/\overline{W}$-бит, он

освобождает SDA линию. Приемник утягивает SDA линию в низкий уровень , чтобы подтвердить передачу. Если приемник не подтверждает передачу, мастер может генерировать

СТОП-условие, чтобы прекратить передачу или начать новую передачу сгенерировав повторное условие СТАРТа.

После того как первый байт был подтвержден, мастер передает байт данных. Если $R/\overline{W}$-бит был установлен в 1, мастер будет действовать как приемник во время фазы передачи. После того,как байт данных был передан, приемник подтверждает байт и мастер генерирует условие СТОП, чтобы завершить передачу. 3 примера передачи в перспективе от системного драйвера.

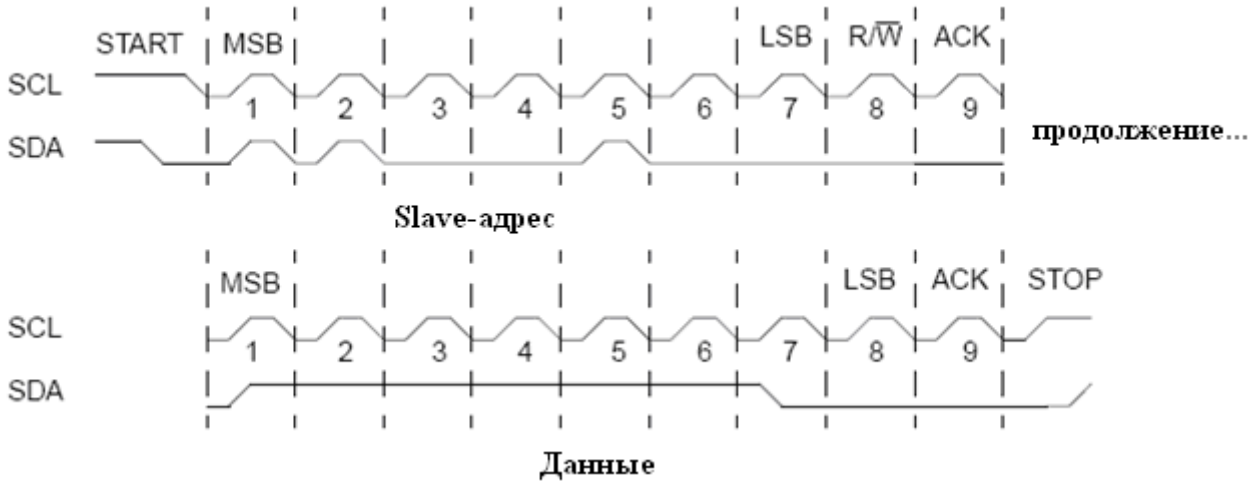


Рисунок 1.218 – Завершенная I2C передача данных

1.3.12.2.3.2.2 Генерация синхроимпульсов

Ядро использует Prescale регистр, чтобы определить частоту линии SCL синхроимпульсов и частоту 5* SCL, которую ядро использует внутри. Для того, чтобы сосчитать значение prescale используется формула:

$$Prescale = \frac{AMBAclockfrequency}{5 \cdot SCLfrequency} - 1$$

SCLfrequency равна 100кГц для операции в Стандартном режиме и 400 кГц для операции в Быстром режиме. Для того, чтобы использовать ядро в Стандартном режиме в системе с синхроимпульсами 60 МГц, выдаваемыми в шину требуемое значение prescale:

$$Prescale = \frac{60Mhz}{5 \cdot 100kHz} - 1 = 119 = 0x77$$

Заметим, что prescale регистр будет изменен только, когда ядро незначимо.Минимальное рекомендуемое значение 3 из-за синхронизационных проблем.Более низкие значения могут заставить мастера нарушить временные ограничения I2C. Это ограничивает частоту системы до 2 МГц для операции в Стандартном режиме.

1.3.12.2.3.3 Программная рабочая модель

Ядро инициализируется записью соответствующего значения в синхронизационный prescale регистр и затем установкой enable(EN) бита разрешения в управляющий регистр Прерывания разрешаются через enable(EN) бит в управляющем регистре.

Чтобы записать байт в slave I2C-мастер должен сгенерировать условие старта и послать адрес slave с битом $R/\overline{W}$, установленным в 0. После того, как slave подтвердил адрес, мастер

передает данные, ожидает подтверждения и генерирует СТОП-условие.Последовательность описанная ниже предписывает ядру выполнить запись:

- Сдвиг влево на одну позицию адреса I2C-устройства и запись результата в регистр передачи.Последний значащий бит регистра передачи $R/\overline{W}$ устанавливается в 0.
- Генерировать условие СТАРТа и послать содержимое регистра передачи установкой STA и WR битов в командном регистре.
- Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.
- Считать RXACK бит в статусном регистре. Если RXACK –“низкий”, slave подтвердил передачу, переходить к шагу 5. Если RXACK установлен, устройство не подтвердило передачу, перейти к шагу 1.
- Записать slave-данные в регистр передачи.
- Послать данные к slave и генерировать условие СТОП установкой STO и WR битов в командном регистре.
- Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.
- Проверить, что slave подтвердил данные чтением RXACK бита в регистре статуса. RXACK не будет установлен.

Чтобы считать байт из памяти , сединенной с I2C, большая часть последовательности, приведенной выше, повторяется. Данные , записываемые в этом случае – это ячейка памяти на I2C slave. После того, как адрес был записан мастер генерирует повторное условие СТАРТа и считывает данные из slave. Последовательность, которую программное обеспечение будет выполнять , чтобы считать из устройства памяти:

Сдвиг влево на одну позицию адреса I2C-устройства и запись результата в регистр передачи.Последний значащий бит регистра передачи $R/\overline{W}$ устанавливается в 0

Генерировать условие СТАРТа и послать содержимое регистра передачи установкой STA и WR битов в командном регистре...

Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.

Считать бит .RxACK в регистре статуса. Если .RxACK -“низкий”, slave подтвердил передачу, перейти к шагу 5. Если RxACK установлен, устройство не подтвердило передачу, перейти к шагу 1.

Записать ячейку памяти,которая должна быть записана из slave в регистр передачи

Установить бит WR в командном регистре.

Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.

Считать бит .RxACK в регистре статуса. RxACK будет “низким”.

Адресовать I2C снова записью его сдвинутого влево адреса в регистр передачи. Установить младший значащий бит регистра передачи (R/W) в “1” чтением из slave.

Установить STA и WR биты в командном регистре, чтобы сгенерировать повторное условие СТАРТа.

Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.

Считать бит .RxACK в регистре статуса. Slave будет подтверждать передачу.

Подготовиться получить данные из памяти, соединенной с I2C. Установить биты RD, ACK и STO в командном регистре.Установив ACK бит в 1, отвергает принятые данные и означает конец передачи.

Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.

Принятые данные теперь могут быть читаны из регистра приема.

Для того, чтобы выполнить последовательные чтения мастер может выполнять итерации 13 -15, не устанавливая ACK и STO биты в шаге 13. Чтобы закончить последовательные чтения, ACK и STO биты устанавливаются. Обращайтесь за справкой к документации I2C-slave, чтобы увидеть, что последовательные чтения поддерживаются.

Финальная последовательность иллюстрирует, как записать один байт в I2C-slave, который требует адресации. Сначала адресуется slave и ячейка памяти slave передается. После того, как slave подтвердил данные ячейки памяти, данные которые должны быть записаны передаются без генерации нового условия СТАРТа:

Сдвиг влево на одну позицию адреса I2C-устройства и запись результата в регистр передачи.Последний значащий бит регистра передачи $R/\overline{W}$ устанавливается в 0

Генерировать условие СТАРТа и послать содержимое регистра передачи установкой STA и WR битов в командном регистре

Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.

Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.

Записать ячейку памяти,которая должна быть записана из slave в регистр передачи

Установить бит в командном регистре.

Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.

Считать RxACK в статус-регистре. RxACK будет “низким.”

Записать байт данных в регистр передачи.

установить WR и STO в командном регистре, чтобы послать байт данных и сгенерировать условие СТОП.

Подождать прерывание или пока TIP бит в статусном регистре установится в состояние “низкий”.

Проверить RxACK в регистре статуса. Если запись была успешной, slave подтвердит передачу байта данных.

Представленные здесь примеры последовательностей могут быть вообще применены к I2C-slaves. Однако некоторые устройства могут отличаться от протокола выше, пожалуйста, с вопросами обращайтесь за справкой к документации I2C. Заметьте , что программный драйвер будет также отслеживать бит (AL) потери арбитража в статус-регистре.

1.3.12.2.3.4 Регистры

Ядро программируется через регистры, отображаемые в адресном пространство APB.

Таблица 1.673 – Регистры I2C

Смещение адреса APB	Регистр
0x00	Prescale регистр синхронизации
0x04	Управляющий регистр
0x08	Регистр передачи*
0x08	Регистр приема**
0x0C	Регистр команд*
0x0C	Регистр статуса**

Примечание.

*- Только запись.

**- Только чтение.

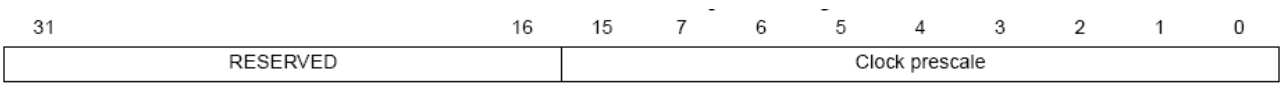


Рисунок 1.219 – Prescale регистр синхронизации I2C-мастера

31:16 Зарезервировано

15:0 Clock prescale – Значение используется, чтобы предварительно поделить частоту SCL линии. Не изменяйте значение этого регистра, пока поле EN управляющего регистра не установлено в “0”. Минимальное рекомендуемое значение этого регистра 0x0003. Меньшие значения могут вызвать у мастера нарушение временных требований I2C из-за синхронизационных ограничений.

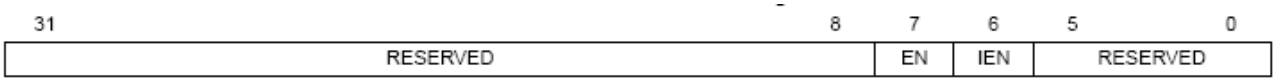


Рисунок 1.220 – Управляющий регистр I2C-мастера

31:8 Зарезервировано.

7 Разрешение (Enable(EN)) – Разрешено ядро I2C. Ядро разрешено, когда этот бит установлен в “1”.

6 Разрешение прерываний (Interrupt enable(IEN)) – Когда этот бит установлен в “1”,

ядро будет генерировать прерывания по завершению передачи.

5:0 Зарезервировано.

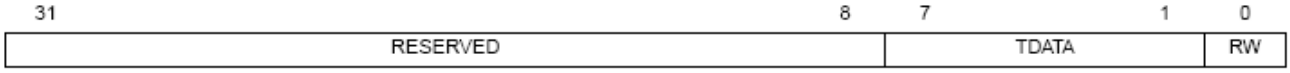


Рисунок 1.221 – Регистр передачи I2C-мастера

31:8 Зарезервировано.

7:1 Передаваемые данные (TDATA) – Старшие значащие биты следующего байта для передачи через I2C.

0 Чтение/запись(R/W) – в передаче данных это младший значащий бит. При передаче slave адреса это RW бит. “1” читается из slave и “0” записывается в slave.

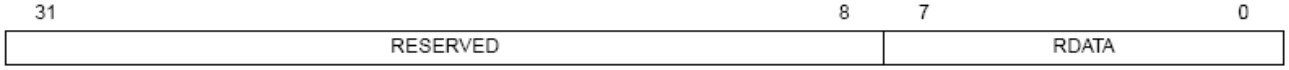


Рисунок 1.222 – Регистр приема I2C-мастера.

31:8 Зарезервировано

7:0 Принимаемые данные (RDATA) – Последний байт, принятый по I2C-шине.

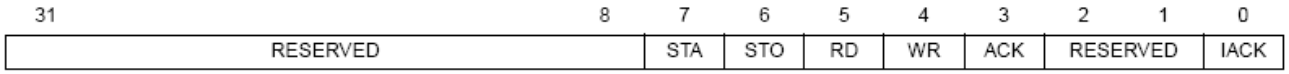


Рисунок 1.223 – Регистр команд I2C-мастера.

31:8 Зарезервировано

7 Старт (STA) – Генерировать условие СТАРТа на I2C-шине. Этот бит также используется, чтобы сгенерировать повторяющиеся условия СТАРТа.

6 СТОП (STO) - Генерировать условие СТОП.
5 Чтение (RD) - Читать из slave.
4 Запись (WR) - Писать в slave.
3 Подтверждение Acknowledge(ACK) – Используется, когда действует как приемник. “0” посылает ACK. “1” посылает NACK.
2:1 Зарезервировано.
0 Подтверждение прерывания Interrupt acknowledge (IACK) – Обнуляет флаг прерывания (IF) в статус регистре.

1.3.12.2.4 Контроллер UART

Модуль универсального асинхронного приемопередатчика (UART – Universal Synchronous Asynchronous Receiver Transmitter) представляет собой периферийное устройство типа «система на кристалле», совместимое с шиной AMBA (Advanced Microcontroller Bus Architecture), разработанное, испытанное и лицензированное компанией ARM.

Контроллер работает в качестве ведомого устройства, подключенного к шине APB (Advanced Peripheral Bus).

1.3.12.2.4.1.1.1 Характеристики

Основные характеристики модуля UART:

- Удовлетворяет требованиям спецификации AMBA (Rev 2.0), что обеспечивает простую интеграцию модуля в систему на кристалле.
- Содержит независимые буферы приема (32x12) и передачи (32x8) типа FIFO (First In First Out – первый вошел, первый вышел), что позволяет снизить интенсивность прерываний центрального процессора.
- Программное отключение FIFO позволяет ограничить размер буфера одним байтом.
- Программируемый генератор-контроллер скорости передачи. Обеспечивает возможность деления тактовой частоты опорного генератора в диапазоне (1x16 – 65535x16). Допускается использование нецелых коэффициентов деления частоты, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц.
- Поддержка стандартных элементов асинхронного протокола связи (стартового, стопового битов и бита контроля четности). Данные добавляются перед передачей и удаляются после приема.
- Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по времени ожидания приемника, по изменению линий состояния модема, а также в случае обнаружения ошибки.
- Обнаружение ложных стартовых битов.
- Формирование и обнаружение сигнала разрыва линии.
- Возможность аппаратного управления потоком данных.
- Полностью программируемый асинхронный последовательный интерфейс со следующими характеристиками:
  - данные длиной 5, 6, 7 или 8 бит
  - формирование и контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение либо не передается)
  - формирование 1 или 2 стоповых битов
  - скорость передачи данных от 0 до UARTCLK/16 Бод.

- Наличие идентификационных регистров, однозначно идентифицирующих модуль UART. Данная возможность позволяет системе производить автоматическое конфигурирование.

1.3.12.2.4.1.1.2 Программируемые параметры

Следующие ключевые параметры могут быть заданы программно:

- скорость передачи данных – целая и дробная части числа
- количество битов данных
- количество стоповых битов
- режим контроля четности
- разрешение или запрет использования буферов FIFO (глубина очереди данных – 32 элемента или один элемент, соответственно)
- выбор порога срабатывания прерывания по заполнению буферов FIFO (⅛, ¼, ½, ¾ и ⅞)
- возможность задания частоты внутреннего тактового генератора (номинальное значение – 1,8432 МГц) в диапазоне от 1,42 до 2,12 МГц для обеспечения возможности формирования битов данных с укороченной длительностью в режиме пониженного энергопотребления
- режим аппаратного управления потоком данных

Для проверки функционирования и соединений модуля предусмотрены дополнительные регистры и режимы тестирования.

1.3.12.2.4.2 Обзор функциональных характеристик

1.3.12.2.4.2.1 Общий обзор

Модуль UART выполняет следующие функции:

- преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму
- преобразование данных, передаваемых на периферийное устройство, из параллельной в последовательную форму

Центральный процессор считывает и записывает данные, а также управляющую информацию и информацию о состоянии через интерфейс шины AMBA APB. Прием и передача данных буферизуются с помощью внутренней памяти FIFO, позволяющей сохранить до 32 байтов информации независимо для режимов приема и передачи.

Модуль UART:

- содержит программируемый генератор-контроллер скорости передачи, формирующий тактовый сигнал одновременно для передачи и для приема данных на основе внутреннего тактового сигнала UARTCLK модуля UART
- обеспечивает возможности, сходные с возможностями промышленного стандарта – контроллера UART 16C650
- поддерживает максимальные скорости передачи данных до 921600 бит/с

Режим работы приемопередатчика и скорость обмена данными контролируются регистром управления линией UARTLCR_H и регистрами делителя скорости передачи данных – целой части (UARTIBRD) и дробной части (UARTFBRD).

Модуль UART может формировать следующие сигналы:

- независимые маскируемые прерывания от приемника (в том числе по времени ожидания), передатчика, а также по изменению состояния модема и в случае обнаружения ошибки
- общий сигнал прерывания, возникающий в случае возникновения одного из вышеуказанных независимых немаскированных прерываний

В случае возникновения ошибки кадровой синхронизации, контроля четности данных, а также разрыва линии соответствующий бит ошибки устанавливается и сохраняется в буфере FIFO. В случае переполнения буфера немедленно устанавливается соответствующий бит в регистре переполнения, а доступ к записи в буфер FIFO блокируется.

Существует возможность программно ограничить размер буфера FIFO одним байтом, что позволяет реализовать общепринятый интерфейс UART с двойной буферизацией.

1.3.12.2.4.2.2 Описание функционирования модуля UART

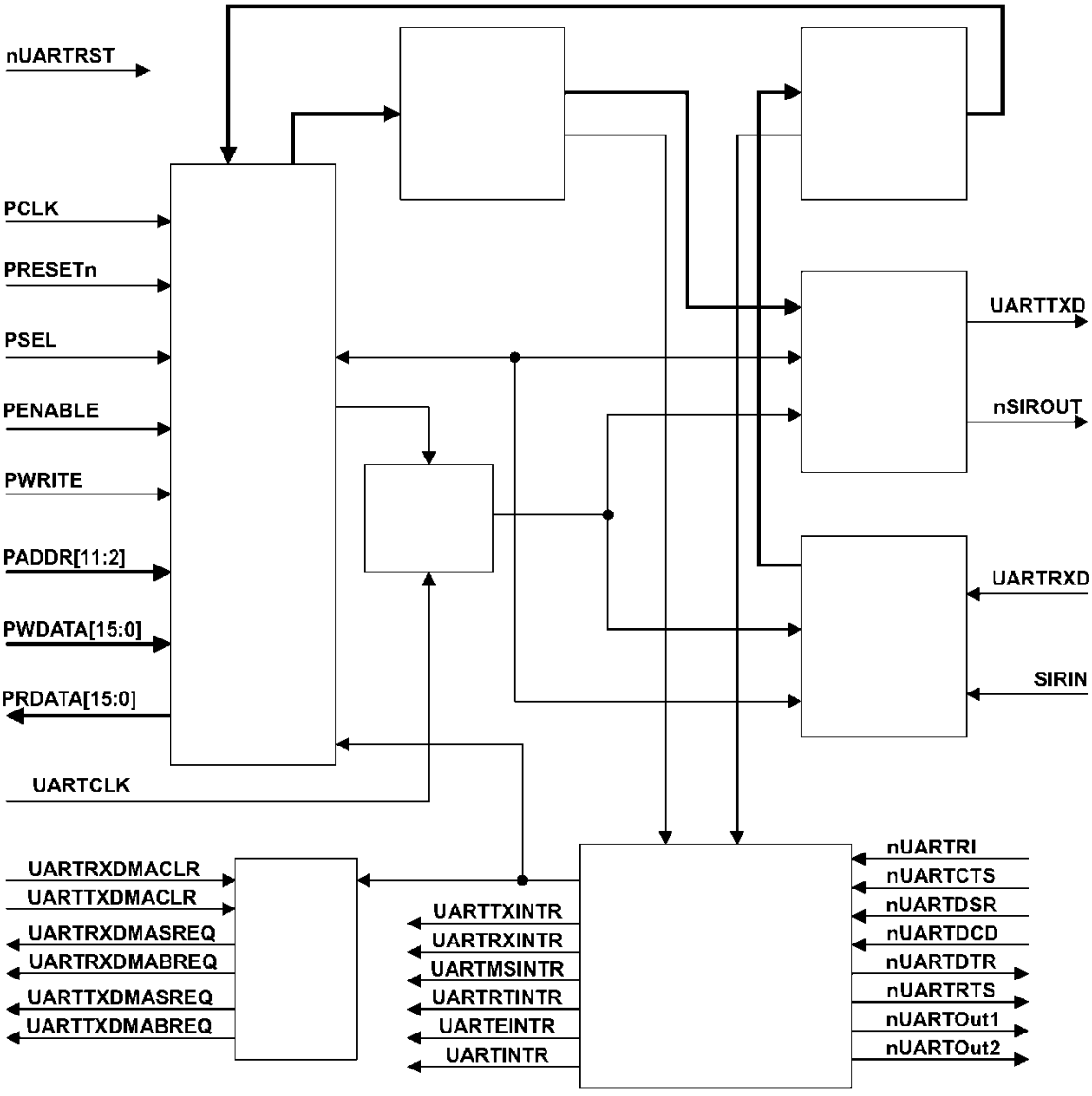


Рисунок 1.224 - Блок_схема модуля UART

1.3.12.2.4.2.2.1 Интерфейс шины AMBA APB

Интерфейс шины AMBA APB осуществляет декодирование кодов чтения и записи для доступа к регистрам управления и состояния и буферам памяти FIFO приемника и передатчика.

1.3.12.2.4.2.2.2 Блок регистров

Блок регистров осуществляет хранение записанных данных, или данных, подлежащих считыванию, через интерфейс AMBA APB.

1.3.12.2.4.2.2.3 Генератор-контроллер скорости передачи

Генератор содержит счетчик без цепи сброса, формирующий внутренний тактовый сигнал (×16) Baud16. Сигнал Baud16 используется для синхронизации схем управления приемником и передатчиком последовательного обмена данными. Сигнал Baud16 представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UARTCLK, и частотой, в 16 раз превышающей скорость передачи данных

1.3.12.2.4.2.2.4 Буфер FIFO передатчика

Буфер FIFO передатчика имеет ширину 8 бит, глубину 32 слова, схему организации доступа типа «первый вошел, первый вышел». Данные от центрального процессора, записанные через шину AMBA APB, сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных. Существует возможность деактивировать буфер FIFO передатчика, в этом случае он будет функционировать как однобайтовый буферный регистр.

1.3.12.2.4.2.2.5 Буфер FIFO приемника

Буфер FIFO приемника имеет ширину 12 бит, глубину 32 слова, схему организации доступа типа «первый вошел, первый вышел». Принятые от периферийного устройства данные и соответствующие коды ошибки сохраняются логикой приема данных в буфере FIFO приемника до тех пор, пока не будут считаны центральным процессором через шину APB. Существует возможность деактивировать буфер FIFO приемника, в этом случае он будет функционировать как однобайтовый буферный регистр.

1.3.12.2.4.2.2.6 Блок передатчика

Блок передатчика осуществляет преобразование данных, считанных из буфера FIFO передатчика, из параллельной в последовательную форму. Управляющая логика выдает последовательный поток битов в следующем порядке: стартовый бит, биты данных, начиная с младшего значащего разряда, бит проверки на четность, и, наконец, стоповые биты, в соответствии с конфигурацией, записанной в регистре управления.

1.3.12.2.4.2.2.7 Блок приемника

Блок приемника преобразует данные, полученные от периферийного устройства, из последовательной в параллельную форму после обнаружения корректного стартового импульса. Кроме того, производятся проверки переполнения буфера, детектирование ошибок контроля четности, ошибок в структуре сигнала, а также разрыва линии, и признаки обнаружения этих ошибок также сохраняются в буфере FIFO приемника.

1.3.12.2.4.2.2.8 Блок формирования прерываний

Контроллер UART генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание как функция ИЛИ независимых запросов на прерывание.

Комбинированный сигнал прерывания может быть подан на внешний контроллер прерываний системы, что обеспечивает дополнительную возможность маскирования каждого периферийного устройства. Это облегчает использование модульных драйверов устройств, всегда «знающих» местоположение разрядов регистра управления источником прерываний.

Также предоставляется возможность подачи на системный контроллер прерываний независимых запросов на прерывание, что обеспечивает маскирование выходных сигналов каждого периферийного устройства.
В данном случае сервисная программа контроллера прерываний сможет одновременно считать информацию обо всех источниках прерывания с одного регистра. Данный подход привлекателен

в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

1.3.12.2.4.2.2.9 Блок и регистры синхронизации

Модуль UART поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов PCLK и UARTCLK. Регистры синхронизации и логика квитирования реализованы и постоянно находятся в активном состоянии. Это практически не отражается на характеристиках устройства и занимаемой площади. Синхронизация сигналов управления осуществляется в обоих направлениях потока данных, то есть как из области действия PCLK в область действия UARTCLK, так и из области действия UARTCLK в область действия PCLK.

1.3.12.2.4.2.2.10 Блок и регистры тестирования

Модуль оснащен блоком и регистрами тестирования для проверки функционального блока и интегрального тестирования с использованием технологии TicTalk или кодовых векторов.

При нормальном режиме работы чтение или запись в регистры тестирования не должны осуществляться.

Интегральное тестирование позволяет проверить правильность подключения модуля UART к системе. Оно обеспечивает осуществление записи в и чтения с каждого входа и выхода.

1.3.12.2.4.2.3 Описание работы модуля

1.3.12.2.4.2.3.1 Сброс настроек интерфейса

Приемопередатчик UART может быть сброшен общим сигналом сброса PRESETn, а также специфическим для модуля сигналом nUARTRST. Внешний контроллер сброса должен использовать сигнал PRESETn для активизации сигнала nUARTRST в асинхронном режиме и его снятия синхронно с UARTCLK. Сигнал PRESETn должен быть установлен на низкий уровень в течение периода времени, достаточного для сброса самого медленного блока системы на кристалле, после чего переведен обратно на высокий уровень. В случае модуля UART необходимо, чтобы сигнал PRESETn находился на низком уровне в течение, как минимум, одного периода PCLK.

1.3.12.2.4.2.3.2 Тактовые сигналы

Частота, выбранная для тактового сигнала UARTCLK, должна обеспечивать поддержку требуемого диапазона скоростей передачи данных в бодах:

- FUARTCLK (min) □16 × baud_rate (max)
- FUARTCLK(max) □□16 × 65535 × baud_rate (min)

Например, для поддержки скорости передачи данных в диапазоне от 110 до 460800 Бод частота UARTCLK должна находиться в интервале от 7,3728 МГц до 115,34 МГц.

Частота UARTCLK, кроме того, должна выбираться с учетом возможности установки скорости передачи данных в рамках заданных требований точности.

Также существует ограничение на соотношение между тактовыми частотами PCLK и UARTCLK. Частота UARTCLK должна быть не более, чем в 5/3 раз выше частоты PCLK:

- FUARTCLK □□5/3 × FPCLK

Например, при работе в режиме UART с максимальной скоростью передачи данных 921600 бод при частоте UARTCLK 14,7456 МГц частота PCLK должна быть не менее 8,85276 МГц. Это гарантирует, что контроллер UART будет иметь достаточно времени для записи принятых данных в буфер FIFO приемника.

1.3.12.2.4.2.3.3 Работа универсального асинхронного приемопередатчика UART

Управляющая информация хранится в регистре управления линией UARTLCR. Этот регистр имеет внутреннюю ширину 30 бит, однако, внешний доступ по шине APB к нему осуществляется через следующие регистры:

UARTLCR_H определяет:

- параметры передачи данных
- длину слова
- режим буферизации
- количество передаваемых стоповых битов
- режим контроля четности
- формирование сигнала разрыва линии

UARTIBRD определяет целую часть коэффициента деления для скорости передачи данных

UARTFBRD определяет дробную часть коэффициента деления для скорости передачи данных

Дробный коэффициент деления

Коэффициент деления для формирования скорости передачи данных состоит из 22 бит, при этом 16 бит выделено для представления его целой части, а 6 бит – дробной части. Это используется генератором-контроллером скорости передачи для определения периода передачи бита. Дробный коэффициент деления позволяет осуществлять обмен данными со стандартными информационными скоростями с использованием в качестве UARTCLK тактовый сигнал с произвольной частотой более 3,6864 МГц.

Целая часть коэффициента деления записывается в 16-битный регистр, содержащий целую часть делителя скорости передачи данных UARTIBRD. Шестиразрядная дробная часть записывается в регистр, содержащий дробную часть делителя скорости передачи данных UARTFBRD. Значение коэффициента деления связано с содержимым UARTCLK следующим образом:

Коэффициент деления = UARTCLK / (16×скорость передачи данных) = BRDI + BRDF

где BRDI – целая часть, а BRDF – дробная часть коэффициента деления, отделенная десятичной запятой, как показано на рисунке ниже.



Рисунок 1.225 - коэффициент деления

Шестибитное значение, записываемое в регистр UARTFBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на 2n, где n – ширина регистра UARTFBRD) и добавления 0,5 для оценки ошибок округления:

- $m = integer(BRDF \times 2n + 0,5)$

В модуле формируется внутренний сигнал Baud16, представляющий собой последовательность импульсов с длительностью, равной периоду сигнала UARTCLK и средней частотой, в 16 раз превышающей требуемую скорость обмена данными. Данный сигнал затем делится на 16 для формирования тактового сигнала передачи. Малое значение коэффициента деления обеспечивает короткий период передачи бита, а большое значение – длинный период передачи бита.

Передача и прием данных

Принятые или передаваемые данные заносятся в 32-элементные буферы FIFO, при этом каждый элемент приемного буфера FIFO, кроме байта данных, хранит также дополнительных четыре бита информации о состоянии модема на символ.

Для передачи данные заносятся в буфер FIFO передатчика. Если работа приемо-передатчика UART разрешена, начинается передача информационного кадра с параметрами, указанными в регистре управления линией UARTLCR_H (стр. 3-12). Передача данных продолжается до опустошения буфера FIFO передатчика. После записи элемента в буфер FIFO передатчика сигнал BUSY переходит в высокое состояние (т.е. буфер FIFO не пуст), которое сохраняется в течение всего времени передачи данных. В низкое состояние сигнал BUSY переходит только после того, как буфер FIFO передатчика станет пуст, а последний бит данных (включая стоповые биты) будет передан. Сигнал BUSY может находиться в высоком состоянии даже в случае, если приемопередатчик UART будет переведен из разрешенного состояния в запрещенное.

Для каждого набора данных снимаются три показания, и сохраняется преобладающее значение. В следующих параграфах определяется средняя точка регистрации, и с каждой стороны данной точки производится выборка.

В случае если приемник находился в неактивном состоянии (на линии входного сигнала UARTRXD постоянно присутствовала единица), и произошел переход входного сигнала из высокого в низкий логический уровень (обнаружен стартовый бит), включается счетчик, тактируемый сигналом Baud16, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов (в режиме асинхронного приемопередатчика) или каждые четыре такта (в режиме ИК обмена данными) сигнала Baud16, обеспечивая более короткие импульсы для отображения логического нуля (половина периода передачи бита).

Стартовый бит считается достоверным в случае, если сигнал на линии UARTRXD сохраняет низкий логический уровень в течение восьми отсчетов сигнала Baud16 с момента включения счетчика. В противном случае детектируется и игнорируется ложный стартовый бит.

В случае если обнаружен достоверный стартовый бит, производится регистрация последовательности данных на входе приемника. Очередной бит данных фиксируются каждые 16 отсчетов тактового сигнала Baud16 (то есть спустя один период передачи бита), в соответствии с запрограммированной длиной символов данных. Затем производится регистрация бита четности (если включен режим контроля четности).

Наконец, производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UARTRXD). В случае если последнее условие не выполняется, устанавливается признак ошибки формирования кадра. После того, как слово данных принято полностью, оно заносится в буфер FIFO приемника наряду с битами признаков наличия ошибки, связанных с принятым словом.

Биты ошибки

Три бита признаков наличия ошибки, ассоциированные с принятым символом данных, заносятся на позиции [10:8] слова данных в буфере FIFO приемника. В случае переполнения буфера FIFO приемника регистрируется дополнительная ошибка переполнения, которая сохраняется в бите 11 буфера FIFO приемника.

Бит переполнения буфера

Бит переполнения непосредственно не связан с конкретным символом в буфере FIFO приемника. Признак переполнения фиксируется в случае, если буфер FIFO заполнен, в то время как очередной символ данных полностью принят (находится в регистре сдвига). При этом данные из регистра сдвига не попадают в буфер FIFO приемника, а перезаписываются в регистре сдвига. Как только в буфере приемника появляется свободное место, очередной принятый символ данных заносится в буфер FIFO вместе с текущим значением признака переполнения.

После этого состояние переполнения сбрасывается. Назначение битов буфера FIFO приемника перечислены в таблице 1.674.

Таблица 1.674 – Назначение битов слова данных в буфере FIFO приемника

Бит буфера FIFO	Назначение
11	Признак переполнения буфера
10	Ошибка – разрыв линии
9	Ошибка контроля четности
8	Ошибка кадрирования
7:0	Принятые данные

Запрет буферов FIFO

Предусмотрена возможность отключения FIFO буферов приемника и передатчика. В этом случае приемная и передающая стороны контроллера UART располагают лишь однобайтными буферными регистрами (ввод данных в буферы FIFO «снизу»). Бит переполнения буфера устанавливается при этом тогда, когда очередной символ данных уже принят, однако предыдущий еще не был считан. В настоящей реализации модуля буферы FIFO физически не отключаются, необходимая функциональность однобайтного регистра достигается за счет логических манипуляций с флагами. При этом в случае, если буфер FIFO отключен, а сдвиговый регистр передатчика пуст (не используется), запись байта данных происходит непосредственно в регистр сдвига, минуя буферный регистр.

Проверка по шлейфу

Проверка по шлейфу выполняется путем установки бита LBE (Loop Back Enable)в регистре управления контроллером UARTCR.

Данные, передаваемые на UARTTXD, принимаются на входе UARTRXD.

1.3.12.2.4.2.3.4 Кадр передачи данных UART

Рисунок 1.226 демонстрирует кадр передачи данных UART.

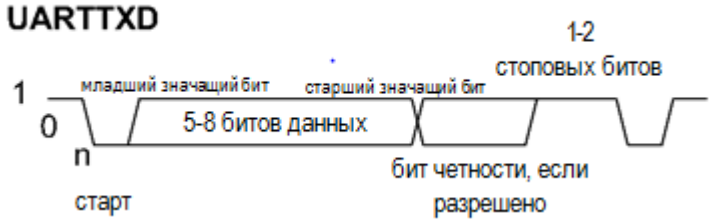


Рисунок 1.226 – кадр передачи данных UART

1.3.12.2.4.2.4 Прерывания

В модуле предусмотрено 11 маскируемых источников прерывания. Их комбинации формируют пять независимых выходных сигналов запроса на прерывание, а также один общий сигнал, представляющий собой комбинацию независимых сигналов, организованных по схеме «ИЛИ».

- UARTRXINTR
- UARTTXINTR
- UARTRTINTR
- UARTMSINTR, генерируемый в случаях:

- сигнала UARTRIINTR при изменении состояния линии nUARTRI
- сигнала UARTCTSINTR при изменении состояния линии nUARTCTS
- сигнала UARTDCDINTR при изменении состояния линии nUARTDCD
- сигнала UARTDSRINTR при изменении состояния линии nUARTDSR

  o UARTEINTR, генерируемый в случаях:

- сигнала UARTOEINTR при переполнении буфера
- сигнала UARTBEINTR при прерывании приема в случае разрыва линии
- сигнала UARTPEINTR при ошибке контроля четности в принятом символе
- сигнала UARTFEINTR при ошибке структуры кадра в принятом символе

UARTINTR – логическое ИЛИ пяти независимых маскированных сигналов

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре установки и сброса маски прерывания UARTIMSC. Установка соответствующего бита маски на высокий уровень разрешает прерывание.

Доступность как индивидуальных, так и общей линии запроса на прерывание позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйверов устройства, построенных по модульному принципу.

Прерывания от приемника и передатчика UARTRXINTR и UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных в соответствии с заданными порогами заполнения буферов FIFO приемника и передатчика.

Прерывание по обнаружению ошибки UARTEINTRформируется в случае возникновения той или иной ошибки приема данных. Предусмотрен ряд условий формирования признака наличия ошибки.

Прерывание по состоянию модема UARTMSINTR представляет собой комбинацию признаков изменения отдельных линий состояния модема.

Признаки возникновения каждого из условий прерывания можно считать либо из регистра состояния прерываний без учета маскирования UARTRIS либо из регистра состояния маскированных прерываний UARTMIS.

1.3.12.2.4.2.4.1 UARTMSINTR

Прерывание по состоянию модема возникает в случае изменения любой из линий состояний модема (nUARTCTS, nUARTDCD, nUARTDSR и nUARTRI). Сброс прерывания осуществляется путем записи «1» в соответствующий (в зависимости от линии состояния модема, вызвавшей прерывание) разряд регистра сброса прерывания UARTICR.

1.3.12.2.4.2.4.2 UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

- Буфер FIFO разрешен и его заполнение достигло заданного порогового значения.

  В этом случае линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения данных из буфера FIFO приемника до тех пор, пока его заполнение не станет меньше порога, либо после сброса прерывания.
- Буфер FIFO запрещен (имеет размер один символ), принят один символ данных. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после чтения одного байта данных из буфера FIFO приемника либо после сброса прерывания.

1.3.12.2.4.2.4.3 UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

- Буфер FIFO разрешен и его заполнение меньше или равно заданному пороговому значению. Сигнал прерывания переходит в низкое состояние после записи данных в буфер FIFO передатчика до тех пор, пока его заполнение не станет больше порога, либо после сброса прерывания.
- Буфер FIFO запрещен (имеет размер один символ), данные в буферном регистре передатчика отсутствуют. При этом линия прерывания переходит в высокое состояние. Сигнал прерывания переходит в низкое состояние после записи одного байта данных либо после сброса прерывания.
-

Для занесения данных в буфер FIFO передатчика необходимо:

- записать данные в буфер FIFO передатчика либо перед разрешением работы приемопередатчика UART и прерываний, либо после разрешения работы приемопередатчика UART и прерываний.

——————Примечание ————

Прерывание передатчика работает по фронту, а не по уровню сигнала. В случае если модуль UART и прерывания от него разрешены до осуществления записи данных в буфер FIFO передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера FIFO.

1.3.12.2.4.2.4.4 UARTRTINTR

Прерывание по времени ожидания приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание по времени ожидания снимается либо после считывания данных из буфера приемника FIFO до его опустошения посредством считывания всех данных (или считывания данных из регистра временного хранения), либо путем записи «1» в соответствующий бит регистра сброса прерываний UARTICR.

1.3.12.2.4.2.4.5 UARTEINTR

Прерывание по обнаружению ошибки возникает в случае ошибки при приеме данных модулем UART. Прерывание может быть вызвано рядом факторов:

- ошибка в структуре кадра
- ошибка контроля четности
- разрыв линии
- переполнение буфера

Причину возникновения прерывания можно определить, прочитав содержимое регистра состояния прерываний без учета маскирования UARTRIS либо регистра состояния маскированных прерываний UARTMIS. Сброс прерывания осуществляется путем записи соответствующих бит в регистр сброса прерываний UARTICR (за прерываниями по обнаружению ошибки закреплены биты с 7 по 10).

1.3.12.2.4.2.4.6 UARTINTR

Все описанные сигналы запроса на прерывание также объединяются в общую линию путем объединения по схеме «ИЛИ» отдельных сигналов с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит обеспечить дополнительное маскирование запросов на уровне периферийных устройств.

Объединенный сигнал запроса на прерывание UART формируется в случае

1.3.12.2.4.3 Программное управление модулем

1.3.12.2.4.3.1 О программном управлении модулем

Следующая информация применима ко всем регистрам контроллера UART.

- Базовый адрес модуля UART не фиксирован и может быть различным в разных системах. Смещение каждого регистра относительно базового адреса фиксировано.
- Доступ к резервным или не используемым адресам не допускается. Попытка доступа к данным адресам может привести к непредсказуемому поведению системы.
- Если не указано особо в сопровождающем тексте:
  - не допускается модификация неопределенных разрядов регистра
  - при чтении неопределенные разряды регистра необходимо игнорировать
  - все разряды регистра сбрасываются на ноль системой или при сбросе по включению питания
- Столбец «Тип» в Таблице 1.675 определяет режим доступа к регистру в соответствии с обозначениями:

RW Чтение и запись
RO Только чтение
WO Только запись

1.3.12.2.4.3.2 Обобщенные данные о регистрах модуля

Таблица 1.675 - Обобщенные данные о регистрах модуля UART

Смещение	Наименование	Тип	Значение после сброса	Размер бита	Описание
0x000	UARTDR	RW	0x---	12/8	Регистр данных UARTDR
0x004	UARTRSR/ UARTECR	RW	0x0	4/0	Регистр состояния приемника/регистр сброса ошибки UARTRSR/UARTECR
0x008-0x014	-	-	-	-	Резерв
0x018	UARTFR	RO	0b-10010---	9	Регистр флагов UARTFR
0x01C	-	-	-	-	Резерв
0x020	UARTILPR	RW	0x00	8	Регистр управления ИК обменом в режиме пониженного энергопотребления UARTILPR
0x024	UARTIBRD	RW	0x0000	16	Регистр, содержащий целую часть делителя скорости передачи данных UARTIBRD
0x028	UARTFBRD	RW	0x00	6	Регистр, содержащий дробную часть делителя скорости передачи данных UARTFBRD
0x02C	UARTLCR_H	RW	0x00	8	Регистр управления линией UARTLCR_H
0x030	UARTCR	RW	0x0300	16	Регистр управления UARTCR
0x034	UARTIFLS	RW	0x12	6	Регистр порога прерывания по заполнению буфера FIFO UARTIFLS

Продолжение таблицы 1.675 -

Смещение	Наименование	Тип	Значение после сброса	Размер бита	Описание
0x038	UARTIMSC	RW	0x000	11	Регистр установки и сброса маски прерывания UARTIMSC
0x03C	UARTRIS	RO	0x00-	11	Регистр состояния прерываний без учета маскирования UARTRIS
0x040	UARTMIS	RO	0x00-	11	Регистр состояния маскированных прерываний UARTMIS (стр. 3-20)
0x044	UARTICR	WO	-	11	Регистр сброса прерываний UARTICR
0x048	UARTDMACR	RW	0x00	3	Регистр управления ПДП UARTDMACR
0x04C-0x07C	-	-	-	-	Резерв
0x080-0x08C	-	-	-	-	Зарезервирован для тестирования
0x090-0xFCC	-	-	-	-	Резерв
0xFD0-0xFDC	-	-	-	-	Зарезервирован для перспективных расширений возможностей модуля
0xFE0	UARTPeriphID0	RO	0x11	8	Регистр UARTPeriphID0
0xFE4	UARTPeriphID1	RO	0x10	8	Регистр UARTPeriphID1
0xFE8	UARTPeriphID2	RO	0x_4a	8	Регистр UARTPeriphID2
0xFEC	UARTPeriphID3	RO	0x00	8	Регистр UARTPeriphID3
0xFF0	UARTPCellID0	RO	0x0D	8	Регистр UARTPCellID0
0xFF4	UARTPCellID1	RO	0xF0	8	Регистр UARTPCellID1
0xFF8	UARTPCellID2	RO	0x05	8	Регистр UARTPCellID2
0xFFC	UARTPCellID3	RO	0xB1	8	Регистр UARTPCellID3

1.3.12.2.4.3.3 Описание регистров

1.3.12.2.4.3.3.1 Регистр данных UARTDR

Регистр UARTDR является регистром данных. В ходе передачи данных:

- если буферы FIFO разрешены, то слово данных, записанное в рассматриваемый регистр, направляется в буфер FIFO передатчика
- если буферы FIFO не разрешены, данные сохраняются в регистре временного хранения передатчика (последний элемент буфера FIFO передатчика)

Операция записи в регистр инициирует передачу данных от модуля UART. Слово данных предваряется стартовым битом, дополняется битом контроля четности (если режим контроля четности включен) и стоповым битом. Затем осуществляется передача сформированного слова.

В ходе приема данных:

- если буферы FIFO разрешены, байт данных и четыре бита состояния (разрыв, ошибка формирования кадра, четность, переполнение) сохраняются в 12-битном буфере FIFO приемника
- если буферы FIFO не разрешены, данные сохраняются в регистре временного хранения приемника (последний элемент буфера FIFO приемника).
-

Полученные из линии связи байты данных считывается путем чтения из регистра UARTDR принятых данных совместно с соответствующими битами состояния. Информация о состоянии также может быть получена путем чтения регистра UARTRSR/UARTECR (см. Таблицу 1.676)

Таблица 1.676 – Регистр UARTDR

Биты	Наименование	Назначение
15:12	-	Резерв
11	OE	Переполнение буфера приемника. Бит устанавливается в «1» в случае, если на вход заполненного буфера FIFO приемника поступают данные. Сбрасывается в «0» после того, как в буфере появляется свободное место.
10	BE	Разрыв линии. Устанавливается в «1» при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, превышающего длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер FIFO загружается только один нулевой символ. Прием данных возобновляется только после перехода линии в логическую единицу и последующего обнаружения корректного стартового бита.
9	PE	Ошибка контроля четности. Устанавливается в «1» в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией UARTLCR_H (стр. 3-12). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
8	FE	Ошибка кадрирования. Устанавливается в «1» в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
7:0	DATA	Принимаемые данные (чтение) Передаваемые данные (запись)

Примечание

Необходимо запрещать работу приемопередатчика UART перед любым перепрограммированием его регистров управления. При переводе приемопередатчика UART в отключенное состояние во время передачи или приема символа перед тем как остановиться он завершает выполняемую операцию.

1.3.12.2.4.3.3.2 Регистр состояния приемника/регистр сброса ошибки UARTRSR/UARTECR

Регистр UARTRSR/UARTECR является регистром состояния приемника/регистром сброса ошибки. Состояние приемника также может быть считано из регистра UARTRSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра соответствует последнему символу, считанному из регистра данных UARTDR перед чтением регистра UARTRSR. Признак переполнения буфера устанавливается немедленно после возникновения этого состояния.

Запись в регистр UARTECR приводит к сбросу признаков ошибок переполнения, контроля четности, структуры кадра и разрыва линии. После сброса настроек устройства все биты устанавливаются в «0».

Таблица 1.677 – Регистр UARTRSR/UARTECR

Биты	Наименование	Назначение
7:0	-	Запись в регистр приводит к сбросу признаков ошибок переполнения, контроля четности, структуры кадра и разрыва линии. Значение данных не представляет важности.
7:4	-	Резерв, при чтении результат не определен.
3	OE	Переполнение буфера приемника. Бит устанавливается в «1» в случае, если на вход заполненного буфера FIFO приемника поступают данные. Сбрасывается в «0» после записи в регистр UARTECR. Содержимое буфера остается действительным, так как запись данных в заполненный буфер FIFO не производилась, и перезаписан был только регистр сдвига. Центральный процессор должен считать данные для освобождения буфера FIFO.
2	BE	Разрыв линии. Устанавливается в «1» при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, превышающего длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). Бит сбрасывается в «0» после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер. В случае обнаружения разрыва линии в буфер FIFO загружается только один нулевой символ. Прием данных возобновляется только после перехода линии в логическую единицу (состояние маркировки) и последующего обнаружения корректного стартового бита.
1	PE	Ошибка контроля четности. Устанавливается в «1» в случае, если четность принятого символа данных не соответствует установкам битов EPS и SPS в регистре управления линией UARTLCR_H (стр. 3-12). Сбрасывается в «0» после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
0	FE	Ошибка кадрирования. Устанавливается в «1» в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). Сбрасывается в «0» после записи в регистр UARTECR. При включенном FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.

Примечание

Перед чтением регистра состояния UARTRSR необходимо считать данные, принятые из линии, путем обращения к регистру данных UARTDR. Противоположная последовательность действий не допускается, так как регистр UARTRSR обновляет свое состояние только после чтения регистра UARTDR. Тем не менее, информация о состоянии приемника может быть получена непосредственно из регистра данных UARTDR.

1.3.12.2.4.3.3.3 Регистр флагов UARTFR

Регистр UARTFR является регистром флагов. После сброса биты регистра флагов TXFF, RXFF и BUSY устанавливаются в “0», а биты TXFE и RXFE – в «1».

Таблица 1.678– Регистр UARTFR

Биты	Наименование	Назначение
15:9	-	Резерв. Не подлежит модификации. При чтении выдаются нули
8	RI	Индикатор вызова. Данный бит дополняет входной сигнал «индикатор вызова» на линии состояния модема nUARTRI. То есть данный бит устанавливается в «1» при низком уровне линии nUARTRI.
7	TXFE	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H (стр. 2-12). Если буфер FIFO запрещен, бит устанавливается в «1», когда буферный регистр передатчика пуст. Если буфер FIFO разрешен, устанавливается бит TXFE, когда буфер FIFO передатчика пуст. Данный бит не дает никакой информации о наличии данных в регистре сдвига передатчика.
6	RXFF	Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, устанавливается данный бит, когда буферный регистр приемника заполнен. Если буфер FIFO разрешен, устанавливается бит RXFF, когда буфер FIFO приемника заполнен.
5	TXFF	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, устанавливается данный бит, когда буферный регистр передатчика заполнен. Если буфер FIFO разрешен, устанавливается бит TXFF, когда буфер FIFO передатчика заполнен.
4	RXFE	Буфер FIFO приемника пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. Если буфер FIFO запрещен, устанавливается данный бит, когда буферный регистр приемника пуст. Если буфер FIFO разрешен, устанавливается бит RXFE, когда буфер FIFO приемника пуст.
3	BUSY	Модуль UART занят. Бит равен 1, в случае если контроллер UART осуществляет передачу данных в линию. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы из регистра сдвига. Данный бит устанавливается при наличии данных в буфере FIFO передатчика, вне зависимости от состояния приемопередатчика UART (разрешен или запрещен).
2	DCD	Обнаружен информационный сигнал. Данный бит дополняет входной сигнал «обнаружен информационный сигнал» на линии состояния модема nUARTDCD. То есть данный бит устанавливается в «1» при низком уровне линии nUARTDCD.
1	DSR	Набор данных готов. Данный бит дополняет входной сигнал «набор данных готов» на линии состояния модема nUARTDCD. То есть данный бит устанавливается в «1» при низком уровне линии nUARTDSR.
0	CTS	Готов к передаче данных. Данный бит дополняет входной сигнал на линии состояния модема nUARTCTS. То есть данный бит устанавливается в «1» при низком уровне линии nUARTCTS.

1.3.12.2.4.3.3.4 Регистр, содержащий целую часть делителя скорости передачи данных UARTIBRD

Регистр UARTIBRD содержит целую часть делителя скорости передачи данных.

Таблица 1.679– Регистр UARTIBRD

Биты	Наименование	Назначение
15:0	BAUD DIVINT	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в «0».

1.3.1 Регистр, содержащий дробную часть делителя скорости передачи данных UARTFBRD

Регистр UARTFBRD содержит дробную часть делителя скорости передачи данных.

Таблица 1.680– Регистр UARTFBRD

Биты	Наименование	Назначение
5:0	BAUD DIVFRAC	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. После сброса устанавливается в «0».

Коэффициент деления вычисляется по формуле:

- Коэффициент деления BAUDDIV = (FUARTCLK /(16×скорость передачи данных)), где FUARTCLK - тактовая частота контроллера UART.

Коэффициент BAUDDIV состоит из целой и дробной частей – BAUD DIVINT и BAUD DIVFRAC, соответственно.

Примечание

Изменение содержимого регистров UARTIBRD и UARTFBRD вступают в силу только после завершения передачи и приема текущего символа данных.

Минимальный допустимый коэффициент деления – 1, максимальный – 65535 (216 - 1). Таким образом, значение UARTIBRD, равное 0, является недопустимым, при этом значение регистра UARTFBRD игнорируется.

Аналогично, при UARTIBRD = 65535 (0xFFFF) значение UARTFBRD не может быть больше нуля. Невыполнение этого условия приведет к прерыванию приема или передачи.

В таблице 1.681 представлены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 7,3728 МГц. При таких параметрах дробная часть коэффициента деления не используется, следовательно, в регистр UARTFBRD должен быть записан ноль.

Таблица 1.681 – Коэффициенты деления для типичных скоростей передачи данных при частоте UARTCLK = 7,3728 МГц

Коэффициент деления	Скорость передачи данных (бит/с)
0x1	460800
0x2	230400
0x4	115200
0x6	76800
0x8	57600
0xC	38400
0x18	19200

Продолжение таблицы 1.681

0x20	14400
0x30	9600
0xC0	2400
0x180	1200
0x105D	110

В таблице 1.682 приведены значения коэффициента деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц.

Таблица 1.682 – Коэффициенты деления для типичных скоростей передачи данных при частоте UARTCLK = 4 МГц

Целая часть	Дробная часть	Требуемая скорость (бит/с)	Реальная скорость (бит/с)	Ошибка, %
0x1	0x5	230400	231911	0,656
0x2	0xB	115200	115101	0,086
0x3	0x10	76800	76923	0,160
0x6	0x21	38400	38369	0,081
0x11	0x17	14400	14401	0,007
0x68	0xB	2400	2400	~0
0x8E0	0x2F	110	110	~0

1.3.12.2.4.3.3.6 Регистр управления линией UARTLCR_H

Регистр UARTLCR_H является регистром управления линией. Данный регистр обеспечивает доступ к разрядам с 29 по 22 регистра UARTLCR. При сбросе все биты регистра UARTLCR_H обнуляются.

Таблица 1.683 – Регистр UARTLCR_H

Биты	Наименование	Назначение
15:8	-	Резерв. Не подлежит модификации. При чтении выдаются нули
7	SPS	Передача бита четности с фиксированным значением 0 = запрещена 1 = в одном из случаев: при EPS = 0 на месте бита четности передается 1 при EPS = 1 на месте бита четности передается 0. Значение бита SPS не играет роли в случае, если формирование и проверка бита четности битом PEN запрещены. Таблица 3-11 содержит таблицу истинности для битов управления контролем четности.
6:5	WLEN	Длина слова. Количество передаваемых или принимаемых информационных битов в кадре: b11 = 8 битов b10 = 7 битов b01 = 6 битов b00 = 5 битов

Продолжение таблицы 1.683 – Регистр UARTLCR_H

Биты	Наименование	Назначение
4	FEN	Разрешение работы буферов FIFO приемника и передатчика: 0 = работа буферов FIFO приемника и передатчика запрещена (знаковый режим), то есть буферы FIFO переводятся в режим 1-байтовых регистров временного хранения 1 = работа буферов FIFO приемника и передатчика разрешена (режим FIFO).
3	STP2	Режим передачи двух стоповых битов. Если данный бит установлен в «1», в конце кадра осуществляется передача двух стоповых битов. Приемник не проверяет наличие дополнительного стопового бита в кадре.
2	EPS	Контроль четности. Контролирует тип четности, который модуль UART использует в процессе передачи и приема. 0 = нечетность. Модуль UART формирует сигнал нечетности или производит проверку на предмет наличия нечетного числа в битах данных и четности. 1 = четность. Модуль UART формирует сигнал четности или производит проверку на предмет наличия четного числа в битах данных и четности. Значение бита SPS не играет роли в случае, если формирование и проверка бита четности битом PEN запрещены. Таблица 3-11 содержит таблицу истинности для битов управления контролем четности.
1	PEN	Разрешение контроля четности: 0 = контроль четности запрещен, и кадр не содержит бита четности 1 = контроль и формирование сигнала четности разрешены Таблица истинности для битов управления контролем четности приведена в Таблице 3-11.
0	BRK	Разрыв линии. Если этот бит установлен в «1», то по завершении передачи текущего символа на выходе UARTTXD устанавливается низкий уровень сигнала. Для правильного выполнения этой операции программное обеспечение должно обеспечить передачу сигнала разрыва в течение, как минимум, времени передачи двух информационных кадров. В нормальном режиме функционирования бит должен быть установлен в «0».

Содержимое регистров UARTLCR_H, UARTIBRD и UARTFBRD совместно образует общий 30-разрядный регистр UARTLCR, который обновляется по стробу, формируемому при записи в UARTLCR_H. Таким образом, для обновления содержимого регистров UARTIBRD или UARTFBRD запись данных в регистр UARTLCR_H необходимо осуществлять в конце.

Примечание

Изменение значений трех регистров можно осуществить корректно двумя способами:
запись UARTIBRD, запись UARTFBRD, запись UARTLCR_H
запись UARTFBRD, запись UARTIBRD , запись UARTLCR_H.
Для изменения значения только одного из регистров UARTIBRD или UARTFBRD:
запись UARTIBRD, запись UARTFBRD, запись UARTLCR_H

Таблица 1.684 представляет собой таблицу истинности для бит управления контролем четности SPS, EPS, и PEN регистра управления линией UARTLCR_H .

Таблица 1.684 – Таблица истинности для битов управления контролем четности

PEN	EPS	SPS	Бит контроля четности (передаваемый или проверяемый)
0	x	x	Не передается, не проверяется
1	1	0	Проверка четности слова данных
1	0	0	Проверка нечетности слова данных
1	0	1	Бит четности постоянно равен 1
1	1	1	Бит четности постоянно равен 0

Примечание

Регистры UARTLCR_H, UARTIBRD и UARTFBRD не должны изменяться:
при разрешенной работе приемопередатчика UART
во время завершения приема или передачи данных в процессе остановки (перевода в запрещенное состояние) приемопередатчика
Целостность данных в буферах FIFO не гарантируется в следующих случаях:
после установки бита разрыва линии BRK
если программное обеспечение произвело остановку приемопередатчика UART в процессе передачи данных при наличии данных в буферах FIFO, а также после его повторного перевода в разрешенное состояние.

1.3.12.2.4.3.3.7 Регистр управления UARTCR

Регистр UARTCR является регистром управления. После сброса все биты регистра управления, за исключением битов 9 и 8, которые устанавливаются в «1», устанавливаются в нулевое состояние.

Таблица 1.685 – Регистр UARTCR

Биты	Наименование	Назначение
15	CTSEn	Разрешение управления потоком данных по CTS. Если данный бит установлен в «1», управления потоком данных по CTS разрешено. Данные передаются в линию только при активном значении сигнала nUARTCTS.
14	RTSEn	Разрешение управления потоком данных по RTS. Если данный бит установлен в «1», управления потоком данных по RTS разрешено. Запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере FIFO приемника.
13	Out2	Данный бит дополняет выходной сигнал nUARTOut2 на линии состояния модема. То есть если данный бит установлен в «1», выходной сигнал будет нулевым. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «индикатор вызова» (RI)
12	Out1	Данный бит дополняет выходной сигнал nUARTOut2 на линии состояния модема. То есть если данный бит установлен в «1», выходной сигнал будет нулевым. В режиме оконечного оборудования (DTE) эта линия может использоваться в качестве линии «обнаружен информационный сигнал» (DCD).
11	RTS	Запрос на передачу данных. Данный бит дополняет выходной сигнал запроса на передачу данных на линии состояния модема nUARTCTS. То есть если данный бит установлен в «1», линия nUARTRTS будет переведена в низкое состояние.
10	DTR	Приемник данных готов. Данный бит дополняет выходной сигнал «приемник данных готов» на линии состояния модема nUARTDTR. То есть если данный бит установлен в «1», линия nUARTDTR будет переведена в низкое состояние.

Продолжение таблицы 1.685

Биты	Наименование	Назначение
9	RXE	Прием разрешен. Установка бита в «1» разрешает работу приемника. Прием данных осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. При переводе приемопередатчика UART в отключенное состояние во время приема текущего символа перед тем как остановиться он завершает выполняемую операцию.
8	TXE	Передача разрешена. Установка бита в «1» разрешает работу передатчика. Передача данных осуществляется либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN. При переводе приемопередатчика UART в отключенное состояние во время передачи текущего символа перед тем как остановиться он завершает выполняемую операцию.
7	LBE	Шлейф разрешен. Если данный бит установлен в «1», бит SIREN установлен в «1», бит SIRTEST регистра управления тестированием UARTTCR (стр. 4-5) установлен в «1», то сигнал с выхода кодека nSIROUT инвертируется и подается на вход кодека SIRIN. Бит SIRTEST устанавливается в «1» для того, чтобы вывести устройство из полудуплексного режима, характерного для интерфейса SIR. Это необходимо для доступа к регистрам тестирования в нормальном режиме работы, и после окончания тестирования по шлейфу бит SIRTEST должен быть установлен в «0». Данная функция сокращает количество внешних сопряжений, необходимых в ходе тестирования системы. Если бит установлен в «1», а бит SIRTEST установлен в «0», то выходная линия передатчика UARTTXD коммутируется на вход приемника UARTRXD. Как в режиме SIR, так и в режиме UART, выходные линии состояния модема коммутируются на соответствующие входные линии. После сброса бит устанавливается в «0» для запрета шлейфа.
6:3	-	Резерв. Не подлежит модификации. При чтении выдаются нули
2	SIRLP	Режим ИК обмена с пониженным энергопотреблением. Данный бит производит выбор режима ИК обмена с пониженным энергопотреблением. Если бит установлен в «0», длительность импульсов данных равна 3/16 длительности передачи бита. Если бит установлен в «1», длительность импульсов данных равна трем тактам сигнала IrLPBaud16 вне зависимости от выбранной скорости передачи данных. Выбор этого режима снижает энергопотребление, однако может привести к уменьшению дальности связи.
1	SIREN	Разрешение работы кодека ИК передачи данных IrDA SIR: 0 = работа кодека ИК передачи данных IrDA SIR запрещена, сигнал nSIROUT находится в низком состоянии (световые импульсы не генерируются), данные на входе SIRIN не обрабатываются. 1 = работа кодека ИК передачи данных IrDA SIR разрешена, Данные передаются на выход nSIROUT и принимаются с входа SIRIN. UARTTXD остается на высоком уровне в состоянии маркировки. Данные на входе UARTRXD и линиях состояния модема не обрабатываются. В случае если UARTEN=0 значение бита не играет роли.
0	UARTEN	Разрешение работы приемопередатчика UART: 0 = работа UART запрещена. При переводе приемопередатчика UART в отключенное состояние во время передачи или приема символа, перед тем как остановиться он завершает выполняемую операцию. 1 = работа UART разрешена. Передача и прием данных осуществляются либо по интерфейсу асинхронного последовательного обмена, либо по интерфейсу ИК обмена SIR, в зависимости от значения бита SIREN.

Примечание

Для того чтобы разрешить передачу данных, необходимо установить в «1» биты TXE и UARTEN. Аналогично, для разрешения приема данных необходимо установить в «1» биты RXE и UARTEN.

Примечание

Рекомендуется следующая последовательность действий для программирования регистров управления:
Остановите работу модуля UART.
Дождитесь окончания приема и/или передачи текущего символа данных.
Сбросьте буфер передатчика путем установки бита FEN регистра UARTLCR_H в «0».
Измените настройки регистра UARTCR.
Возобновите работу модуля UART.

1.3.12.2.4.3.3.8 Регистр порога прерывания по заполнению буфера FIFO UARTIFLS

Регистр UARTIFLS является регистром порога прерывания по заполнению буфера FIFO. Данный регистр используется для установки порогового значения заполнения буферов FIFO передатчика и приемника, по достижении которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно.

Прерывания генерируются в момент перехода величины заполнения буфера через заданное значение. То есть прерывания генерируются, когда уровень заполнения достигает уровня срабатывания.

После сброса в регистре устанавливается порог, соответствующий заполнению половины буфера. Формат регистра и значения его битов представлены в таблице 1.686.

Таблица 1.686 – Регистр UARTIFLS

Биты	Наименование	Назначение
15:6	-	Резерв. Не подлежит модификации. При чтении выдаются нули
5:3	RXIFLSEL	Выбор порога прерывания по заполнению буфера FIFO приемника. Точки срабатывания прерывания по заполнению буфера приемника: b000 = Буфер FIFO приемника заполнен на ⅛ b001 = Буфер FIFO приемника заполнен на ¼ b010 = Буфер FIFO приемника заполнен на ½ b011 = Буфер FIFO приемника заполнен на ¾ b100 = Буфер FIFO приемника заполнен на ⅞ b101-b111 = резерв
2:0	TXIFLSEL	Выбор порога прерывания по заполнению буфера FIFO передатчика. Точки срабатывания прерывания по заполнению буфера передатчика: b000 = Буфер FIFO передатчика заполнен на ⅛ b001 = Буфер FIFO передатчика заполнен на ¼ b010 = Буфер FIFO передатчика заполнен на ½ b011 = Буфер FIFO передатчика заполнен на ¾ b100 = Буфер FIFO передатчика заполнен на ⅞ b101-b111 = резерв

1.3.12.2.4.3.3.9 Регистр установки и сброса маски прерывания UARTIMSC

Регистр UARTIMSC является регистром установки и сброса маски прерывания. Данный регистр доступен для чтения и записи.

При чтении выдается текущее значение маски. При записи «1» в конкретный бит производится установка маски на соответствующее прерывание. Запись «0» производит сброс соответствующей маски

При сбросе все биты регистра обнуляются.

Таблица 1.687 – Регистр UARTIMSC

Биты	Наименование	Назначение
15:11	-	Резерв. Не подлежит модификации. При чтении выдаются нули
10	OEIM	Маска прерывания по переполнению буфера. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTOEINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTOEINTR. Запись «0» производит сброс маски.
9	BEIM	Маска прерывания по разрыву линии. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTBEINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTBEINTR. Запись «0» производит сброс маски.
8	PEIM	Маска прерывания по ошибке контроля четности. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTPEINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTPEINTR. Запись «0» производит сброс маски.
7	FEIM	Маска прерывания по ошибке в структуре кадра. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTFEINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTFEINTR. Запись «0» производит сброс маски.
6	RTIM	Маска прерывания по времени ожидания буфера FIFO приемника. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTRTINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTRTINTR. Запись «0» производит сброс маски.
5	TXIM	Маска прерывания от передатчика. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTTXINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTTXINTR. Запись «0» производит сброс маски.
4	RXIM	Маска прерывания от приемника. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTRXINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTRXINTR. Запись «0» производит сброс маски.
3	DSRMIM	Маска прерывания по изменению состояния линии nUARTDSR. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTDSRINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTDSRINTR. Запись «0» производит сброс маски.
2	DCDMIM	Маска прерывания по изменению состояния линии nUARTDCD. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTDCDINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTDCDINTR. Запись «0» производит сброс маски.
1	CTSMIM	Маска прерывания по изменению состояния линии nUARTCTS. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTCTSINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTCTSINTR. Запись «0» производит сброс маски.
0	RIMIM	Маска прерывания по изменению состояния линии nUARTRI. При чтении текущая маска прерывания устанавливается для прерывания по сигналу UARTRIINTR. При записи «1» производится установка маски прерывания для прерывания по сигналу UARTRIINTR. Запись «0» производит сброс маски.

1.3.12.2.4.3.3.10 Регистр состояния прерываний без учета маскирования UARTRIS

Регистр UARTRIS является регистром состояния прерываний без учета маскирования. Данный регистр доступен для чтения и записи. Данный регистр содержит текущее состояние соответствующего прерывания без учета маскирования. Данные, записываемые в регистр, игнорируются.

———Внимание! ————————

После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в «0». Значение битов прерывания по состоянию модема после сброса не определено.

Таблица 1.688 – Регистр UARTRIS

Биты	Наименование	Назначение
15:11	-	Резерв. Не подлежит модификации. При чтении выдаются нули
10	OERIS	Состояние прерывания по переполнению буфера. Состояние до маскирования прерывания UARTOEINTR.
9	BERIS	Состояние прерывания по разрыву линии. Состояние до маскирования прерывания UARTBEINTR.
8	PERIS	Состояние прерывания по ошибке контроля четности. Состояние до маскирования прерывания UARTPEINTR.
7	FERIS	Состояние прерывания по ошибке в структуре кадра. Состояние до маскирования прерывания UARTFEINTR.
6	RTRIS	Состояние прерывания по времени ожидания буфера FIFO приемника. Состояние до маскирования прерывания UARTRTINTR.
5	TXRIS	Состояние прерывания от передатчика. Состояние до маскирования прерывания UARTTXINTR.
4	RXRIS	Состояние прерывания от приемника. Состояние до маскирования прерывания UARTRXINTR.
3	DSRRMIS	Состояние прерывания по изменению состояния линии nUARTDSR. Состояние до маскирования прерывания UARTDSRINTR.
2	DCDRMIS	Состояние прерывания по изменению состояния линии nUARTDCD. Состояние до маскирования прерывания UARTDCDINTR.
1	CTSRMIS	Состояние прерывания по изменению состояния линии nUARTCTS. Состояние до маскирования прерывания UARTCTSINTR.
0	RIRMIS	Состояние прерывания по изменению состояния линии nUARTRI. Состояние до маскирования прерывания UARTRIINTR.

1.3.12.2.4.3.3.11 Регистр состояния маскированных прерываний UARTMIS

Регистр UARTMIS является регистром состояния прерываний с учетом маскирования.

Данный регистр доступен для чтения и записи. Данный регистр содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением битов прерывания по состоянию модема (биты с 3 по 0), устанавливаются в «0». Значение бит прерывания по состоянию модема после сброса не определено.

Таблица 1.689 – Регистр UARTMIS

Биты	Наименование	Назначение
15:11	-	Резерв. Не подлежит модификации. При чтении выдаются нули
10	OEMIS	Состояние маскированного прерывания по переполнению буфера. Состояние маскированного прерывания UARTOEINTR.
9	BEMIS	Состояние маскированного прерывания по разрыву линии. Состояние маскированного прерывания UARTBEINTR.
8	PEMIS	Состояние маскированного прерывания по ошибке контроля четности. Состояние маскированного прерывания UARTBEINTR.
7	FEMIS	Состояние маскированного прерывания по ошибке в структуре кадра. Состояние маскированного прерывания UARTFEINTR.
6	RTMIS	Состояние маскированного прерывания по времени ожидания буфера FIFO приемника. Состояние маскированного прерывания UARTRTINTR.
5	TXMIS	Состояние маскированного прерывания от передатчика. Состояние маскированного прерывания UARTTXINTR.
4	RXMIS	Состояние маскированного прерывания от приемника. Состояние маскированного прерывания UARTRXINTR.
3	DSRMMIS	Состояние маскированного прерывания по изменению состояния линии nUARTDSR. Состояние маскированного прерывания UARTDSRINTR.
2	DCDMMIS	Состояние маскированного прерывания по изменению состояния линии nUARTDCD. Состояние маскированного прерывания UARTDCDINTR.
1	CTSMMIS	Состояние маскированного прерывания по изменению состояния линии nUARTCTS. Состояние маскированного прерывания UARTCTSINTR.
0	RIMMIS	Состояние маскированного прерывания по изменению состояния линии nUARTRI. Состояние маскированного прерывания UARTRIINTR.

1.3.12.2.4.4 Регистр сброса прерываний UARTICR

Регистр UARTICR является регистром сброса прерываний и доступен только для записи. Сброс признака прерывания по заданному событию производится путем записи «1» в соответствующий бит. Запись «0» в любой из разрядов игнорируется.

Таблица 1.690 – Регистр UARTICR

Биты	Наименование	Назначение
15:11	Reserved	Резерв. Не подлежит модификации. При чтении выдаются нули
10	OEIC	Сброс прерывания по переполнению буфера. Сброс прерывания UARTOEINTR.
9	BEIC	Сброс прерывания по разрыву линии. Сброс прерывания UARTBEINTR.
8	PEIC	Сброс прерывания по ошибке контроля четности. Сброс прерывания UARTPEINTR.
7	FEIC	Сброс прерывания по ошибке в структуре кадра. Сброс прерывания UARTFEINTR.
6	RTIC	Сброс прерывания по времени ожидания буфера FIFO приемника. Сброс прерывания UARTRTINTR.
5	TXIC	Сброс прерывания от передатчика. Сброс прерывания UARTTXINTR.
4	RXIC	Сброс прерывания от приемника. Сброс прерывания UARTRXINTR.
3	DSRMIC	Сброс прерывания по изменению состояния линии nUARTDSR. Сброс прерывания UARTDSRINTR.
2	DCDMIC	Сброс прерывания по изменению состояния линии nUARTDCD. Сброс прерывания UARTDCDINTR.
1	CTSMIC	Сброс прерывания по изменению состояния линии nUARTCTS. Сброс прерывания UARTCTSINTR.
0	RIMIC	Сброс прерывания по изменению состояния линии nUARTRI. Сброс прерывания UARTRIINTR.

1.3.12.2.4.4.1.1 Регистр управления прямым доступом к памяти UARTDMACR

Регистр UARTDMACR является регистром управления прямым доступом к памяти. Данный регистр доступен для чтения и записи. При сбросе все биты регистра обнуляются.

Таблица 1.691– Регистр UARTDMACR

Биты	Наименование	Назначение
15:3	-	Резерв. Не подлежит модификации. При чтении выдаются нули
2	DMAONERR	Ошибка прямого доступа к памяти. Если бит установлен в «1», в случае возникновения прерывания по обнаружению ошибки блокируются запросы ПДП от приемника UARTRXDMASREQ и UARTRXDMABREQ.
1	TXDMAE	Использование ПДП при передаче. Если бит установлен в «1», разрешено формирование запросов ПДП для обслуживания буфера FIFO передатчика
0	RXDMAE	Использование ПДП при приеме. Если бит установлен в «1», разрешено формирование запросов ПДП для обслуживания буфера FIFO приемника

1.3.12.2.4.4.1.2 Регистры идентификации периферийных устройств UARTPeriphID0-3

Регистры UARTPeriphID0-3 – это четыре восьмиразрядных регистра, охватывающие адреса от 0xFE0 до 0xFEC. Регистры, в принципе, могут рассматриваться как один 32-битный регистр. Регистры, доступные только для чтения, предоставляют следующие опции для периферийного устройства:

- PartNumber[11:0] идентифицирует тип периферийного устройства. Значение «0x011» соответствует асинхронному приемопередатчику (UART).
- Designer ID[19:12] идентифицирует разработчика. Значение «0x41» означает, что периферийное устройство разработано компанией ARM.
- Revision[23:20] Номер версии периферийного устройства..
- Configuration[31:24]

Вариант конфигурации периферийного устройства. Значением конфигурации является «0».



Рисунок 1.227– Назначение разрядов регистра идентификации периферийных устройств

Примечание

При разработке карты распределения памяти необходимо помнить, что объем памяти регистра составляет 4 КБ. Доступ к памяти всех регистров идентификации периферийных устройств должен быть 32-битным; доступ осуществляется согласно инструкциям LDR и STR.

- **Регистр идентификации периферийных устройств UARTPeriphID0**

В регистр UARTPeriphID0 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.692 – Регистр UARTPeriphID0

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	PartNumber0	Результатом повторного считывания данных битов будет «0x11»

Регистр идентификации периферийных устройств UARTPeriphID1

В регистр UARTPeriphID1 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.693 – Регистр UARTPeriphID1

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:4	Designer0	Результатом повторного считывания данных битов будет «0x1»
3:0	PartNumber1	Результатом повторного считывания данных битов будет «0x0»

1.3.12.2.4.4.1.3

Регистр идентификации периферийных устройств UARTPeriphID2

В регистр UARTPeriphID2 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.694 – Регистр UARTPeriphID2

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:4	Revision	Значение данного поля зависит от версии контроллера UART. r1p0 0x0 r1p1 0x1 r1p3 0x2 r1p4 0x2 r1p5 0x3
3:0	Designer1	Результатом повторного считывания данных битов будет «0x4»

Регистр идентификации периферийных устройств UARTPeriphID3

В регистр UARTPeriphID3 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.695– Регистр UARTPeriphID3

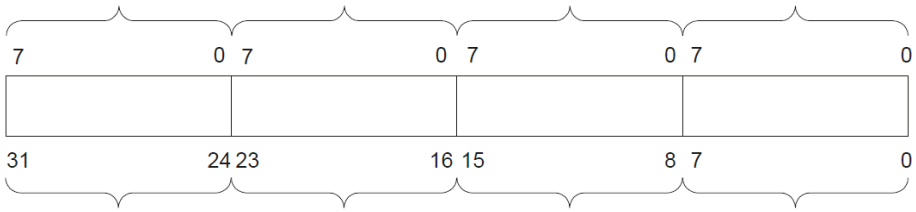
Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	Configuration	Результатом повторного считывания данных битов будет «0x00»

1.3.12.2.4.4.1.4 Регистры идентификации периферийных устройств UARTPCellID0-3

Регистры UARTPCellID0-3 – это четыре восьмибитовых регистра, охватывающие адреса от 0xFF0 до 0xFFC. Регистры, в принципе, могут рассматриваться как один 32-битный регистр. Данный регистр используется в качестве стандартной системы идентификации периферийных устройств. Значение регистра UARTPCellID – 0xB105F00D.

Фактическое назначение разрядов регистра

UARTPCellID3 UARTPCellID2 UARTPCellID1 UARTPCellID0



UARTPCellID3 UARTPCellID2 UARTPCellID1 UARTPCellID0

Номинальное назначение разрядов регистра

Рисунок 1.228 – Назначение разрядов регистра идентификации модуля PrimeCell

Регистр UARTPCellID0

В регистр UARTPCellID0 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.696 – Регистр UARTPCellID0

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	UARTPCellID0	Результатом повторного считывания данных битов будет «0x0D»

Регистр UARTPCellID1

В регистр UARTPCellID1 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.697 – Регистр UARTPCellID1

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	UARTPCellID1	Результатом повторного считывания данных битов будет «0xF0»

Регистр UARTPCellID2

В регистр UARTPCellID2 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.698 – Регистр UARTPCellID2

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	UARTPCellID2	Результатом повторного считывания данных битов будет «0x05»

Регистр UARTPCellID3

В регистр UARTPCellID3 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.699 – Регистр UARTPCellID3

Биты	Наименование	Назначение
15:8	-	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
7:0	UARTPCellID3	Результатом повторного считывания данных битов будет «0xB1»

1.3.12.2.5 Контроллер SPI

Модуль порта синхронной последовательной связи PrimeCell SSP (SSP – Synchronous Serial Port) работает в качестве ведомого устройства с расширенной шинной архитектурой для микроконтроллеров (AMBA - Advanced Microcontroller Bus Architecture), подключенного к расширенной периферийной шине (APB – Advanced Peripheral Bus). Модуль PrimeCell SSP представляет собой периферийное устройство типа «система на кристалле» (SoC – System-on-Chip), разработанное, протестированное и лицензированное компанией ARM.

Модуль PrimeCell SSP выполняет функции интерфейса последовательной синхронной связи в режиме ведущего или ведомого устройства и обеспечивает обмен данными с подключенным ведомым или ведущим периферийным устройством в соответствии с одним из протоколов:

интерфейс SPI фирмы Motorola

интерфейс SSI фирмы Texas Instruments

интерфейс Microwire фирмы National Semiconductor

Как в ведущем, так и в ведомом режиме работы модуль PrimeCell SSP обеспечивает:

преобразование данных, размещенных во внутреннем буфере FIFO передатчика (восемь 16-разрядных ячеек данных) из параллельного в последовательный формат

преобразование данных из последовательного в параллельный формат и их запись в аналогичный буфер FIFO приемника (восемь 16-разрядных ячеек данных).

Модуль формирует сигналы прерываний по следующим событиям:

необходимость обслуживания буферов FIFO приемника и передатчика

переполнение буфера FIFO приемника

наличие данных в буфере FIFO приемника по истечении времени ожидания.

1.3.12.2.5.1.1.1 Основные характеристики модуля PrimeCell SSP

Модуль PrimeCell SSP обладает следующими характеристиками:

- Удовлетворяет требованиям спецификации AMBA (Rev 2.0), что обеспечивает простую интеграцию модуля в систему на кристалле.
- Может функционировать как в ведущем, так и в ведомом режиме.
- Программное управление скоростью обмена данными и возможность предварительного деления частоты
- Содержит независимые буферы приема и передачи (8 ячеек, 16 бит) с организацией доступа типа FIFO (First In First Out – первый вошел, первый вышел)
- Программный выбор одного из интерфейсов обмена: SPI, Microwire, SSI
- Программируемая длительность информационного кадра от 4 до 16 бит
- Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, а также по возникновению события переполнения буфера приемника

- Возможность тестирования по шлейфу
- Наличие идентификационных регистров, однозначно идентифицирующих модуль PrimeCell SSP, что позволяет операционной системе производить автоматическое конфигурирование.

На рисунке 1.229 представлена структурная схема модуля PrimeCell SSP.

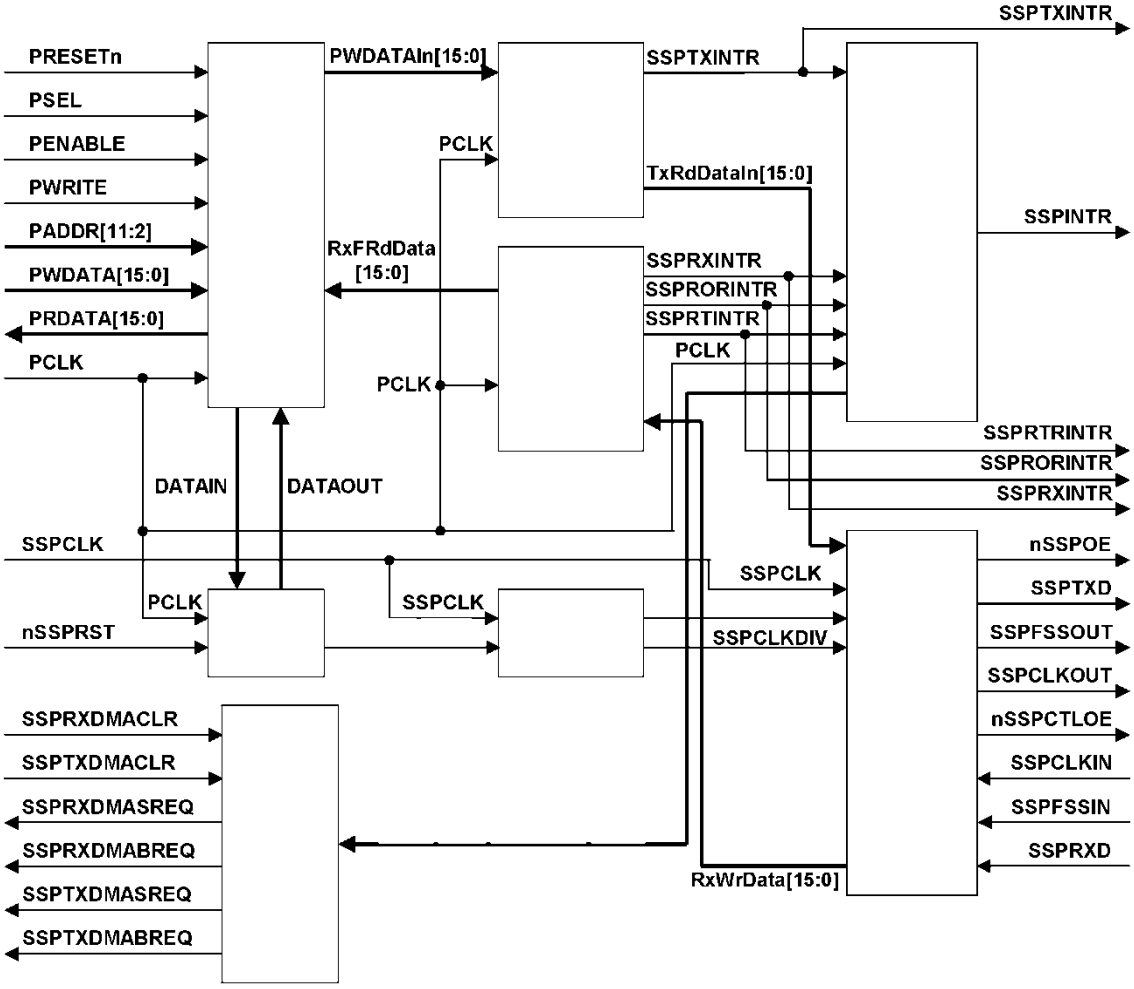


Рисунок 1.229 - Структурная схема модуля PrimeCell SSP

1.3.12.2.5.1.1.2 Программируемые параметры

Предоставляется возможность программирования следующих параметров:

- режим функционирования периферийного устройства – ведущее или ведомое
- разрешение функционирования
- формат информационного кадра
- скорость передачи данных
- фаза и полярность тактового сигнала
- размер блока данных от 4 до 16 бит
- маскирование прерываний

1.3.12.2.5.1.2 Характеристики интерфейса SPI

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии

- программное задание фазы и полярности тактового сигнала.

1.3.12.2.5.1.2.1 Характеристики интерфейса Microwire

Интерфейс Microwire фирмы National Semiconductor обеспечивает полудуплексный обмен данными с использованием восьмибитных управляющих последовательностей.

1.3.12.2.5.1.2.2 Характеристики интерфейса SSI

Синхронный последовательный интерфейс (SSI) фирмы Texas Instruments обеспечивает:

- полнодуплексный обмен данными по четырехпроводной линии
- возможность перевода линии передачи данных в третье состояние

1.3.12.2.5.2 Обзор функциональных характеристик

1.3.12.2.5.2.1 Общий обзор модуля PrimeCell SSP

Модуль PrimeCell SSP представляет собой интерфейс синхронного последовательного обмена данными с периферийными устройствами, способный функционировать в качестве ведущего или ведомого устройства и поддерживающий протоколы передачи данных SPI фирмы Motorola, Microwire фирмы National Semiconductor, а также SSI фирмы Texas Instruments.

Модуль PrimeCell SSP осуществляет преобразование данных, полученных от периферийного устройства, из последовательной в параллельную форму. Центральный процессор считывает и записывает данные, а также управляющую информацию и информацию о состоянии через интерфейс шины AMBA APB. Прием и передача данных буферизуются с помощью буферов FIFO, обеспечивающих хранение до восьми машинных слов данных шириной 16 бит независимо для режимов приема и передачи. Последовательные данные передаются по линии SSPTXD и принимаются с линии SSPRXD.

Модуль PrimeCell SSP содержит программируемые делитель и предделитель частоты, формирующие тактовый сигнал обмена данными SSPCLKOUT из сигнала, поступающего на линию SSPCLK. Скорость передачи данных может достигать более 2 МГц, в зависимости от частоты SSPCLK и максимальной скорости передачи данных подключенного периферийного устройства.

Режим обмена данными модуля PrimeCell SSP, формат информационного кадра и количество бит данных задаются программно с помощью регистров управления SSPCR0 и SSPCR1. См. «Регистр управления 0 SSPCR0» и «Регистр управления 1 SSPCR1».

Модуль формирует четыре независимо маскируемых прерывания:

- SSPTXINTR – запрос на обслуживание буфера передатчика
- SSPRXINTR – запрос на обслуживание буфера приемника
- SSPRORINTR – переполнение приемного буфера FIFO
- SSPRTINTR – истечение периода ожидания при наличии данных в приемном буфере FIFO

Кроме того, формируется общий сигнал прерывания SSPINTR, возникающий в случае активности одного из вышеуказанных независимых немаскированных прерываний.

В дополнение к вышеуказанным прерываниям, модуль также формирует сигналы запроса на прямой доступ к памяти (ПДП) для совместной работы с контроллером ПДП.

В зависимости от выбранного режима работы модуля сигнал SSPFSSOUT используется:

- или для кадровой синхронизации (интерфейс SSI, активное состояние – высокий уровень),
- или для выбора ведомого режима (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

1.3.12.2.5.2.2 Функциональное описание модуля PrimeCell SSP

На Рисунке 1.230 представлена структурная схема модуля PrimeCell SSP.

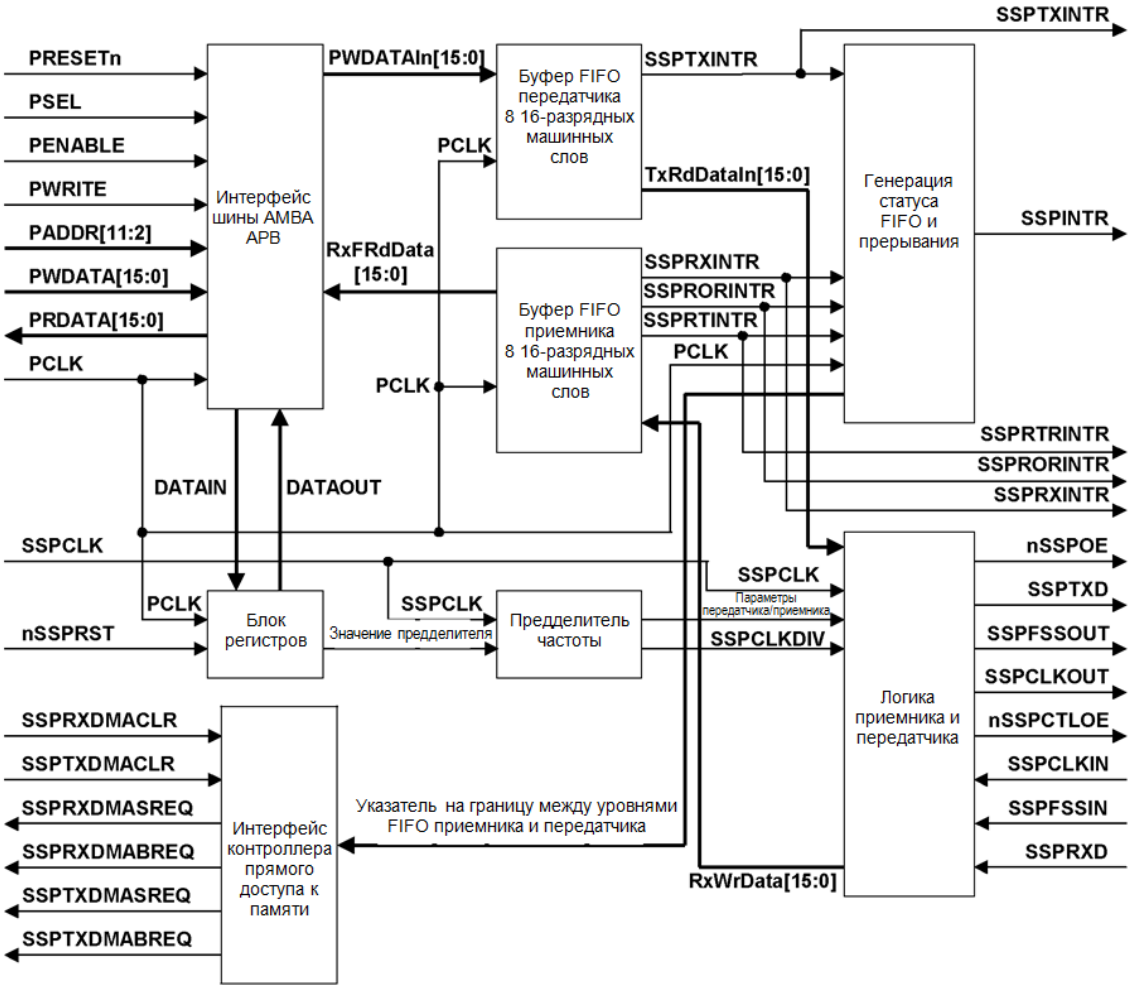


Рисунок 1.230 - Структурная схема модуля PrimeCell SSP

1.3.12.2.5.2.2.1 Интерфейс шины AMBA APB

Интерфейс шины AMBA APB осуществляет декодирование кодов чтения и записи для доступа к регистрам управления и состояния и буферам памяти FIFO приемника и передатчика.

Шина AMBA APB является локальной вторичной шиной, способной расширяться в режиме низкой мощности до широкополосной шины AMBA AHB (AHB - Advanced High-performance Bus) в пределах иерархии системы AMBA. Шина AMBA APB группирует периферийные устройства с узкой шиной для предотвращения перегрузки системной шины и обеспечивает интерфейс с помощью регистров с распределением памяти, доступ к которым осуществляется под программным управлением.

1.3.12.2.5.2.2.2 Блок регистров

Блок регистров осуществляет хранение записанных данных, или данных, подлежащих считыванию, через интерфейс AMBA APB.

1.3.12.2.5.2.2.3 Предделитель частоты

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными SSPCLKOUT с помощью внутреннего делителя частоты, состоящего из двух последовательно соединенных счетчиков без цепи сброса.

Для деления частоты сигнала SSPCLK можно задать коэффициент предварительного деления частоты в диапазоне от 2 до 254 с шагом 2 путем записи значения в регистр SSPCPSR. Так как младший значащий разряд регистра SSPCPSR не используется, исключается возможность деления частоты на нечетный коэффициент, что, в свою очередь гарантирует формирование тактового сигнала симметричной формы (с одинаковой длительностью полупериодов высокого и низкого уровня). См. «Регистр предварительного делителя частоты SSPCPSR» .

Полученное значение снова делится на коэффициент, задаваемый программно в диапазоне от 1 до 256 путем записи соответствующего значения в регистр управления SSPCR0, для получения конечного тактового сигнала обмена данными SSPCLKOUT для ведущего режима.

1.3.12.2.5.2.2.4 Буфер FIFO передатчика

Буфер FIFO передатчика имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа «первый вошел, первый вышел» (FIFO – First-In, First-Out). Данные от центрального процессора, записанные через шину AMBA APB, сохраняются в буфере до тех пор, пока не будут считаны блоком передачи данных.

В ведущем или ведомом режиме параллельно поступающие данные записываются в буфер FIFO передатчика перед преобразованием в последовательную форму и передачей подключенному ведомому или ведущему устройству, соответственно, через выходной контакт SSPTXD.

1.3.12.2.5.2.2.5 Буфер FIFO приемника

Буфер FIFO приемника имеет ширину 16 бит, глубину 8 слов, схему организации доступа типа «первый вошел, первый вышел». Данные, принятые от периферийного устройства через последовательный интерфейс, сохраняются блоком приема данных в нем до тех пор, пока не будут считаны центральным процессором через шину AMBA APB.

В ведущем или ведомом режиме последовательные данные, поступающие через выходной контакт SSPTXD, регистрируются перед параллельной загрузкой в буфер FIFO приемника подключенного ведомого или ведущего устройства, соответственно.

1.3.12.2.5.2.2.6 Блок приема и передачи данных

В режиме ведущего устройства модуль формирует тактовый сигнал обмена данными для подключенных ведомых устройств, формируемый путем деления частоты сигнала SSPCLK, как уже было описано ранее.

Блок передатчика последовательно считывает значения из буфера FIFO передатчика и производит их преобразование из параллельной в последовательную форму. Далее поток последовательных данных и сигнал управления кадром, тактированные сигналом SSPCLKOUT, передаются по линии SSPTXD к подключенным ведомым устройствам. Блок приемника выполняет преобразование данных, поступающих синхронно с линии SSPRXD, из последовательной в параллельную форму, после чего загружает их в буфер FIFO приемника, откуда они могут быть считаны через интерфейс шины APB.

В режиме ведомого устройства тактовый сигнал обмена данными формируется подключенным ведущим устройством и поступает по линии SSPCLKIN, и используется для тактирования последовательностей приема и передачи. При этом блок передатчика, тактируемый этим внешним сигналом от ведущего устройства, последовательно считывает данные из буфера FIFO передатчика, преобразует их из параллельной формы в последовательную, после чего выдает поток последовательных данных и сигнал управления кадром через выходной контакт

SSPTXD. Блок приемника выполняет преобразование данных, поступающих с линии SSPRXD, из последовательной в параллельную форму, и сохраняет их в буфере FIFO приемника для последующего считывания через интерфейс шины APB.

1.3.12.2.5.2.2.7 Блок формирования прерываний

Модуль SSP генерирует четыре независимых маскируемых прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание как функция ИЛИ независимых запросов на прерывание.

Комбинированный сигнал прерывания может быть подан на внешний контроллер прерываний системы, что обеспечивает дополнительную возможность маскирования каждого периферийного устройства. Это облегчает использование модульных драйверов устройств, всегда «знающих» местоположение разрядов регистра управления источником прерываний.

Также предоставляется возможность подачи на системный контроллер прерываний независимых запросов на прерывание, что обеспечивает маскирование выходных сигналов каждого периферийного устройства. В данном случае сервисная программа контроллера прерываний сможет одновременно считать информацию обо всех источниках прерывания с одного регистра. Данный подход привлекателен в случае, если скорость доступа к регистрам периферийных устройств значительно превышает тактовую частоту центрального процессора в системе реального времени.

Модуль SSP позволяет использовать оба описанных выше похода.

Предусмотрены независимые линии запроса на прерывание потока принимаемых и передаваемых динамических данных SSPTXINTR и SSPRXINTR, отличные от линии прерываний состояний, что позволяет производить считывание или запись данных в соответствии с уровнями срабатывания буферов FIFO приемника или передатчика.

1.3.12.2.5.2.2.8 Интерфейс прямого доступа к памяти

Модуль PrimeCell SSP предоставляет интерфейс подключения к контроллеру прямого доступа к памяти. Подробная информация приведена в разделе «Интерфейс прямого доступа к памяти модуля PrimeCell».

1.3.12.2.5.2.2.9 Блок и регистры синхронизации

Модуль PrimeCell SSP поддерживает как асинхронный, так и синхронный режимы работы тактовых генераторов PCLK и UARTCLK. Регистры синхронизации и логика квитирования реализованы и постоянно находятся в активном состоянии. Это практически не отражается на характеристиках устройства и занимаемой площади. Синхронизация сигналов управления осуществляется в обоих направлениях потока данных, то есть:

- из области действия PCLK в область действия SSPCLK
- из области действия SSPCLK в область действия PCLK

1.3.12.2.5.2.3 Работа модуля PrimeCell SSP

1.3.12.2.5.2.3.1 Сброс настроек интерфейса

Сброс настроек модуля PrimeCell SSP производится общим сигналом сброса PRESETn, а также специфическим для модуля сигналом сброса nSSPRST. Внешний контроллер сброса должен использовать сигнал PRESETn для активизации сигнала nSSPRST в асинхронном режиме и его снятия синхронно с SSPCLK. Сигнал PRESETn должен быть установлен на низкий уровень в течение периода времени, достаточного для сброса самого медленного блока системы на кристалле, после чего переведен обратно на высокий уровень. В случае модуля PrimeCell SSP необходимо, чтобы сигнал PRESETn находился на низком уровне в течение, как минимум, одного периода PCLK.

1.3.12.2.5.2.3.2 Конфигурирование модуля SSP

После сброса работа блоков модуля PrimeCell SSP блокируется до выполнения процедуры задания конфигурации.

Для этого необходимо запрограммировать регистры управления SSPCR0 и SSPCR1 для конфигурирования периферийного устройства в качестве ведущего или ведомого, работающего по одному из следующих протоколов передачи данных:

- SPI фирмы Motorola
- SSI фирмы Texas Instruments
- Microwave фирмы National Semiconductor

Для установки требуемой скорости передачи данных, задаваемой внешним сигналом SSPCLK, необходимо запрограммировать регистр предварительного делителя частоты SSPCPSR. См. «Регистр предварительного делителя частоты SSPCPSR» .

1.3.12.2.5.2.3.3 Разрешение работы модуля PrimeCell SSP

Необходимо либо инициализировать буфер FIFO передатчика путем записи в него до восьми 16-разрядных слов при деактивированном модуле PrimeCell SSP, либо разрешить прерывание центрального процессора запросом на обслуживание буфера FIFO передатчика. После разрешения работы модуля приемопередатчик начинает обмен данными по линиям передачи SSPTXD и приема SSPRXD.

1.3.12.2.5.2.3.4 Соотношения между тактовыми сигналами

В модуле имеется ограничение на соотношение между частотами тактовых сигналов PCLK и SSPCLK. Частота SSPCLK должна быть меньше или равна частоте PCLK. Выполнение этого требования гарантирует синхронизацию сигналов управления, передаваемых из зоны действия тактового сигнала SSPCLK в зону действия сигнала PCLK в течение времени, меньшего продолжительности передачи одного информационного кадра:

FSSPCLK <= FPCLK

В режиме ведомого устройства сигнал SSPCLKIN от внешнего ведущего устройства подвергается двойной синхронизации, а затем поступает на схему задержки для обнаружения фронта. Для обнаружения фронта сигнала SSPCLKIN требуются три такта сигнала SSPCLK. Сигнал SSPTXD имеет меньшее время установки по отношению к заднему фронту сигнала SSPCLKIN, по которому ведущее устройство производит отбор данных из линии.

Время установки и удержания сигнала SSPRXD по отношению к сигналу SSPCLKIN должно выбираться с запасом для обеспечения правильного считывания данных в пределах модуля SSPMS. Для обеспечения корректной работы устройства необходимо, чтобы частота SSPCLK была как минимум в 12 раз больше, чем максимальная предполагаемая частота сигнала SSPCLKIN.

Частота тактового сигнала SSPCLK должна обеспечивать поддержку требуемого диапазона скоростей обмена данными. Отношение минимальной частоты сигнала SSPCLK к максимальной частоте сигнала SSPCLKOUT в режиме ведомого устройства составляет 12, в режиме ведущего – два.

В режиме ведущего устройства для обеспечения максимальной скорости обмена 1,8432 Мбит/с частота сигнала SSPCLK должна составлять не менее 3,6864 МГц. При частоте сигнала SSPCLK, равной 3,6864 МГц, в регистр SSPCPSR должно быть записано значение 2, а поле SCR[7:0] регистра SSPCR0 должно быть установлено в 0.

В режиме ведомого устройства для обеспечения максимальной скорости обмена 1,8432 Мбит/с частота сигнала SSPCLK должна составлять не менее 22,12 МГц. Аналогично, соотношение между максимальной частотой сигнала SSPCLK и минимальной частотой

SSPCLKOUT составляет 254 x 256. См. «Регистр управления 0 SSPCR0» на стр. 3-4 и «Регистр делителя тактовой частоты SSPCPSR» на стр. 3-8.

Минимальная допустимая частота сигнала SSPCLK определяется следующими равенствами, условия которых должны выполняться одновременно:

FSSPCLK(min) => 2 x FSSPCLKOUT(max) – для режима ведущего; FSSPCLK(min) => 12 x FSSPCLKIN(max) – для режима ведомого.

Максимальная допустимая частота сигнала SSPCLK определяется следующими равенствами, условия которых должны выполняться одновременно:

FSSPCLK(max) => 254 x 256 x FSSPCLKOUT(min) – для режима ведущего; FSSPCLK(max) => 254 x 256 x FSSPCLKIN(min) – для режима ведомого.

1.3.12.2.5.2.3.5 Программирование регистра управления SSPCR0

Более подробная информация о присваивании значений битам данного регистра приведена в разделе «Регистр управления 0 SSPCR0».

Регистр SSPCR0 предназначен для:

- задания тактовой частоты, определяющей скорость информационного обмена
- выбора одного из трех протоколов обмена данными
- выбора размера слова данных (где применимо)

Значение тактовой частоты, определяющей скорость информационного обмена (Serial Clock Rate (SCR) – тактовая частота, определяющая скорость информационного обмена), наряду со значением коэффициента деления тактового сигнала CPSDVSR регистра SSPCPSR используется для определения скорости передачи и приема данных модуля PrimeCell SSP в зависимости от частоты внешнего сигнала SSPCLK.

Формат информационного кадра задается путем установки значения поля FRF, а размер слова данных – путем установки значения поля DSS.

Для протокола SPI фирмы Motorola, кроме того, задается полярность и фаза сигнала (биты SPH и SPO).

1.3.12.2.5.2.3.6 Программирование регистра управления SSPCR1

Более подробная информация о присваивании значений битам данного регистра приведена в разделе «Регистр управления 1 SSPCR1».

Регистр SSPCR1 предназначен для:

- выбора ведущего или ведомого режима функционирования приемопередатчика
- включения режима проверки канала по шлейфу
- разрешения или запрещения работы модуля PrimeCell SSP

Выбор ведущего режима модуля PrimeCell SSP осуществляется путем записи «0» в поле MS регистра SSPCR1. Данное значение устанавливается после сброса автоматически.

Запись «1» в поле MS регистра SSPCR1 переводит модуль PrimeCell SSP в режим ведомого устройства. В данном режиме разрешение или запрещение формирования сигнала SSPTXD модуля PrimeCell SSP осуществляется путем установки бита SOD (slave mode SSPTXD output disable – запрет линии SSPTXD для ведомого режима) регистра SSPCR1. Данная функция полезна при подключении к одной линии нескольких подчиненных устройств, когда ведущие устройства осуществляют параллельную трансляцию.

Для разрешения функционирования модуля PrimeCell SSP необходимо присвоить биту SSE (Synchronous Serial Port Enable – разрешение последовательного синхронного порта) значение «1».

1.3.12.2.5.2.3.7 Формирование тактового сигнала обмена данными

Тактовый сигнал обмена данными формируется путем деления частоты тактового сигнала SSPCLK. На первом этапе формирования частота данного сигнала делится на четный коэффициент CPSDVSR, находящийся в диапазоне от 2 до 254 и доступный для

программирования через регистр SSPCPSR. Сформированный сигнал далее поступает на делитель частоты с коэффициентом (1 + SCR) в диапазоне от 1 до 256, где значение SCR доступно для программирования через SSPCR0.

Частота выходного тактового сигнала обмена данными SSPCLKOUT определяется следующим равенством:

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$$

Например, в случае если частота сигнала SSPCLK составляет 3,6864 МГц, а значение CPSDVSR = 2, частота сигнала SSPCLKOUT находится в интервале от 7,2 кГц до 1,8432 МГц.

1.3.12.2.5.2.3.8 Формат информационного кадра

Каждый информационный кадр содержит, в зависимости от запрограммированного значения, от 4 до 16 бит данных. Передача данных начинается со старшего значащего разряда. Можно выбрать одну из следующих базовых структур построения кадра:

- SSI фирмы Texas Instruments
- SPI фирмы Motorola
- Microwire фирмы National Semiconductor

Во всех режимах построения кадра тактовый сигнал SSPCLKOUT находится в неактивном состоянии, когда модуль PrimeCell SSP пребывает в холостом режиме, и переход в другой режим с запрограммированной частотой производится только при активной передаче или приеме данных. Перевод сигнала SSPCLKOUT в неактивное состояние используется как признак режима ожидания приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

В режимах SPI и Microwire выходной сигнал кадровой синхронизации передатчика SSPFSSOUT имеет активный низкий уровень и поддерживается на низком уровне в течение всего периода передачи информационного кадра.

В режиме построения кадра SSI фирмы Texas Instruments перед началом каждого информационного кадра на выходе SSPFSSOUT формируется импульс длительностью, равной одному тактовому интервалу обмена данными. В данном режиме приемопередатчик PrimeCell SSP, равно как ведомое «внекристальное» устройство, передает данные на линию по переднему фронту сигнала SSPCLKOUT, а считывает данные с другого устройства из линии по заднему фронту этого сигнала.

В отличие от полнодуплексных режимов передачи данных SSI и SPI, режим Microwire фирмы National Semiconductor использует специальный способ обмена данными между ведущим и ведомым устройствами, функционирующий в режиме полудуплекса. В данном режиме на «внекристальное» ведомое устройство перед началом передачи информационного кадра посылается специальная восьмибитная управляющая последовательность. В течение всего времени передачи данной последовательности приемник не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита восьмибитной управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит. Таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

1.3.12.2.5.2.3.9 Формат последовательного синхронного обмена данными SSI фирмы Texas Instruments

На рисунке ниже продемонстрирован формат последовательного синхронного обмена данными протокола SSI фирмы Texas Instruments для одиночного обмена.

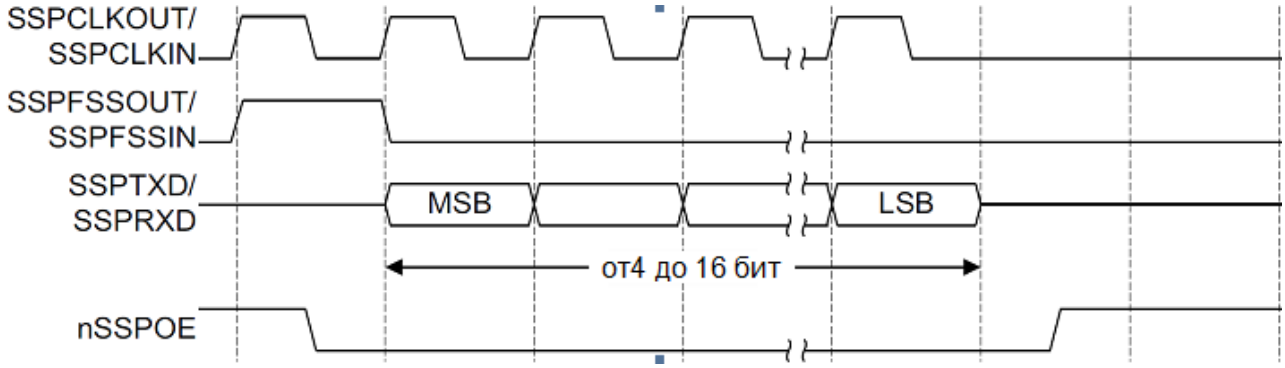


Рисунок 1.231 - Формат синхронного обмена протокола SSI фирмы Texas Instruments (одиночный обмен)

В данном режиме при неактивном модуле PrimeCell SSP сигналы SSPCLKOUT и SSPFSSOUT переводятся на низкий логический уровень, а линия передачи данных SSPTXD поддерживается в третьем состоянии. После появления хотя бы одного элемента в буфере FIFO передатчика сигнал SSPFSSOUT переводится на высокий логический уровень на время, соответствующее одному периоду сигнала SSPCLKOUT. Значение из буфера FIFO при этом переносится в сдвиговый регистр блока передатчика. По следующему переднему фронту сигнала SSPCLKOUT старший значащий разряд информационного кадра (4 – 16 бит данных) выдается на выход линии SSPTXD. Аналогично, старший значащий разряд информационного кадра сдвигается на выход линии SSPRXD «внекристальным» ведомым устройством последовательной передачи данных.

Как модуль PrimeCell SSP, так и ведомое «внекристальное» устройство последовательно загружают биты данных в сдвиговый регистр по заднему фронту сигнала SSPCLKOUT. Принятые данные переносятся из сдвигового регистра в буфер FIFO приемника после загрузки в него младшего значащего бита данных по очередному переднему фронту сигнала PCLK.

Диаграмма последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments при неравномерной передаче данных представлена на Рисунке Рисунок 1.232 – Формат последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments (непрерывный обмен).

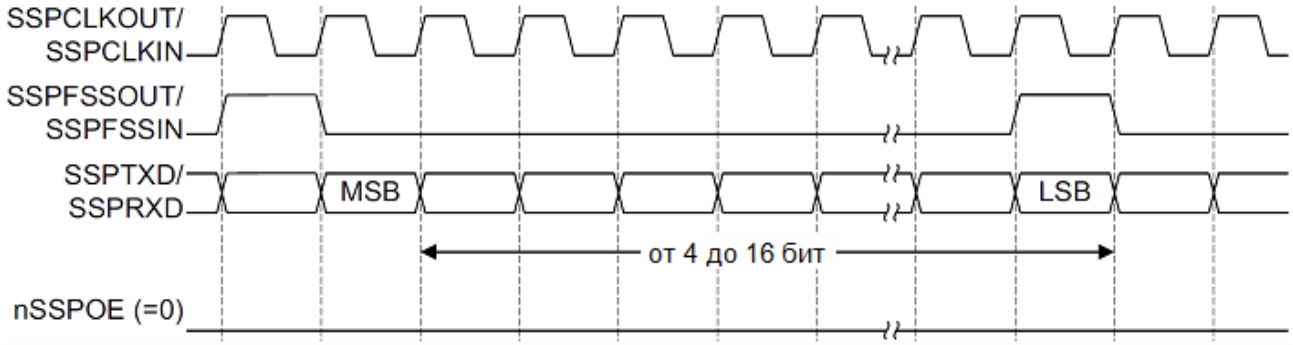


Рисунок 1.232 – Формат последовательного синхронного обмена по протоколу SSI фирмы Texas Instruments (непрерывный обмен)

1.3.12.2.5.2.3.10 Формат обмена данными SPI фирмы Motorola

Интерфейс SPI фирмы Motorola осуществляется по четырем сигнальным линиям, при этом сигнал SSPFSSOUT выполняет функцию выбора ведомого устройства. Основной функцией протокола SPI фирмы Motorola является возможность выбора состояния и фазы сигнала SSPCLKOUT в режиме ожидания (при неактивном приемопередатчике) путем задания значений битов SPO и SPH регистра управления SSPSCR0. См. «Регистр управления 0 SSPCR0».

1.3.12.2.5.2.3.11 Выбор полярности тактового сигнала – бит SPO

Если бит SPO равен 0, то в режиме ожидания линия SSPCLKOUT переводится на низкий логический уровень. В противном случае при отсутствии обмена данными линия SSPCLKOUT переводится на высокий логический уровень.

1.3.12.2.5.2.3.12 Выбор фазы тактового сигнала – бит SPH

Значение бита SPH определяет фронт тактового сигнала, по которому осуществляется выборка данных и изменение состояния на выходе линии. Оно оказывает значительное влияние на первый бит, передаваемый по разрешению или запрету тактового перехода до первого обнаружения фронта тактового сигнала.

В случае если бит SPH установлен на низкий уровень, регистрация данных приемником осуществляется после первого обнаружения фронта тактового сигнала.

Если бит SPH установлен на высокий уровень, регистрация данных приемником осуществляется после второго обнаружения фронта тактового сигнала.

1.3.12.2.5.2.3.13 Формат обмена данными SPI фирмы Motorola, SPO=0, SPH=0

На рисунках Рисунок 1. и Рисунок 1. показаны диаграммы одиночного и непрерывного обмена в режиме SPI с SPO=0, SPH=0. На рисунке Рисунок 1. продемонстрирована диаграмма одиночного обмена в режиме SPI с SPO=0, SPH=0.

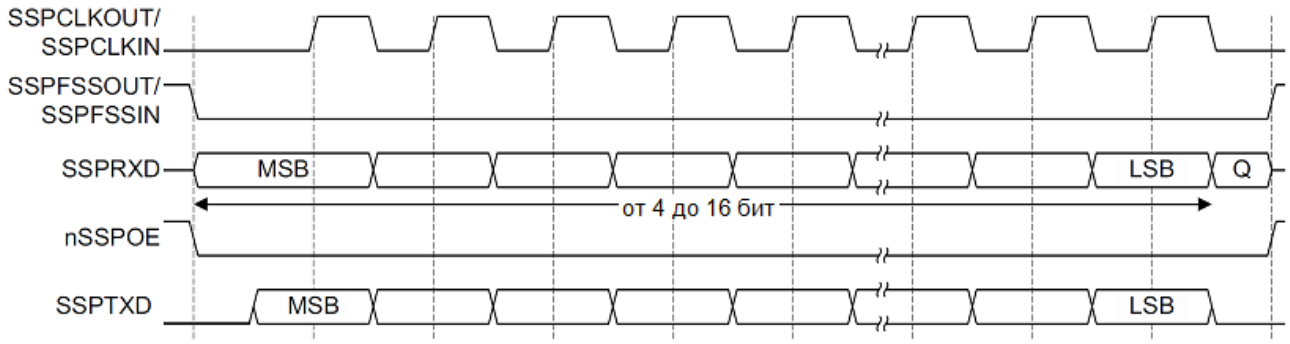


Рисунок 1.233 – Формат обмена по протоколу SPI фирмы Motorola, SPO=0, SPH=0 (одиночный обмен)

На рисунке ниже продемонстрирована диаграмма непрерывного обмена в режиме SPI с SPO=0, SPH=0.

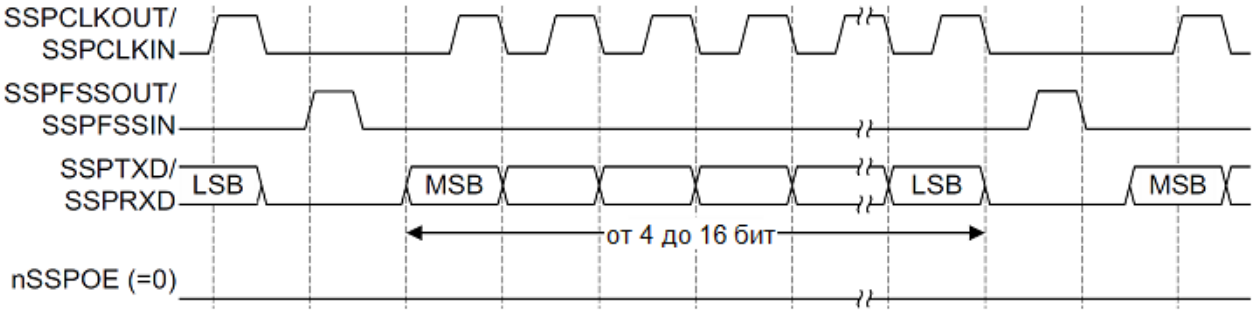


Рисунок 1.234 - Формат обмена по протоколу SPI фирмы Motorola, SPO=0, SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT переводится на низкий логический уровень
- сигнал SSPFSSOUT переводится на высокий логический уровень
- линия передачи данных SSPTXD произвольно переводится на низкий логический уровень
- сигнал nSSPOE переводится на высокий уровень, обеспечивая, таким образом, переход выходного контакта передатчика в высокоимпедансное состояние
- если модуль PrimeCell SSP сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится на низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня
- если модуль PrimeCell SSP сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится на высокий уровень, отключая выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня

Если работа модуля PrimeCell SSP разрешена, и в буфере FIFO передатчика содержатся допустимые данные, начало передачи инициализируется сигналом SSPFSSOUT от ведущего устройства, который переводится на низкий логический уровень. Данное событие запускает передачу данных от ведомого устройства на входную линию SSPTXD ведущего устройства. Сигнал nSSPOE переводится на низкий уровень, переводя выходной контакт SSPTXD ведущего устройства в активное состояние.

По истечении полутакта сигнала SSPCLKOUT на линию SSPTXD передаются допустимые данные от ведущего устройства. К этому моменту должны быть сформированы данные на линиях обмена, как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSPCLKOUT переводится на высокий логический уровень.

Далее данные регистрируются по переднему фронту и выдаются на линию по заднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSSOUT переводится на высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

Тем не менее, в режиме непрерывной неравномерной передачи данных на линии SSPFSSOUT должны формироваться импульсы высокого логического уровня между передачами каждого слова данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства на низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSPFSSIN на высокий уровень по окончании передачи каждого кадра, разрешая, таким образом, запись новых данных, последовательно передаваемых периферийным устройством. По завершении приема последнего бита блока данных линия SSPFSSOUT переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLKOUT.

1.3.12.2.5.2.3.14 Формат обмена данными SPI фирмы Motorola, SPO=0, SPH=1

На рисунке ниже продемонстрирована диаграмма одиночного и непрерывного обмена данными в режиме SPI с SPO=0, SPH=1.

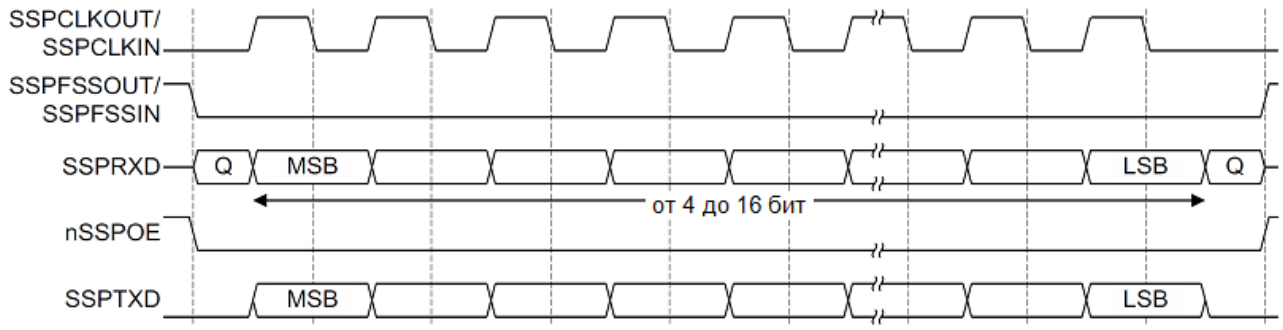


Рисунок 1.235 – Формат обмена по протоколу SPI фирмы Motorola, SPO=0,SPH=1 (одиночный и непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT переводится на низкий логический уровень
- сигнал SSPFSSOUT переводится на высокий логический уровень
- линия передачи данных SSPTXD произвольно переводится на низкий логический уровень
- сигнал nSSPOE переводится на высокий уровень, обеспечивая, таким образом, переход выходного контакта передатчика в высокоимпедансное состояние
- если модуль PrimeCell SSP сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится на низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня
- если модуль PrimeCell SSP сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится на высокий уровень, отключая выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня

Если работа модуля PrimeCell SSP разрешена, и в буфере FIFO передатчика содержатся допустимые данные, начало передачи инициализируется сигналом SSPFSSOUT от ведущего устройства, который переводится на низкий логический уровень. Сигнал nSSPOE переводится на низкий уровень, переводя выходной контакт SSPTXD ведущего устройства в активное состояние. По истечении полутакта сигнала SSPCLKOUT на соответствующих линиях обмена, как ведущего, так и ведомого устройств разрешена передача данных. В это же время включается линия SSPCLKOUT, на которой формируется передний фронт сигнала.

Далее данные регистрируются по заднему фронту и выдаются на линию по переднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSSOUT переводится на высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной неравномерной передачи данных линия SSPFSSOUT находится на низком логическом уровне между последовательными информационными словами и переводится на высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

1.3.12.2.5.2.3.15 Формат обмена данными SPI фирмы Motorola, SPO=1, SPH=0

Диаграммы одиночного и непрерывного обмена в режиме SPI с SPO=1, SPH=0 представлены на рисунках 1.236 и 1.237.

На Рисунке 1.236 продемонстрирована диаграмма одиночного обмена в режиме SPI с SPO=1, SPH=0.

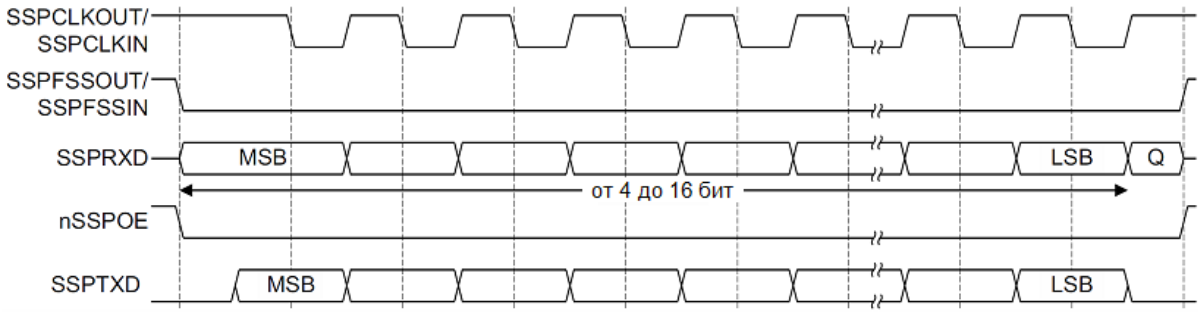


Рисунок 1.236 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)

На рисунке 1.237 представлена диаграмма непрерывного обмена в режиме SPI с SPO=1, SPH=0.

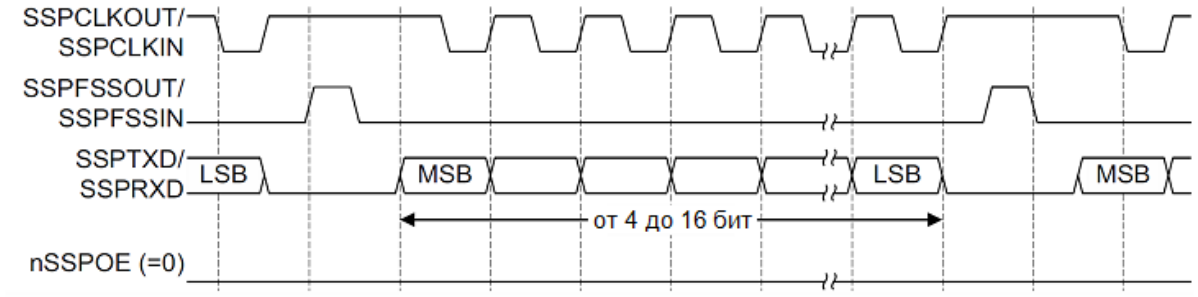


Рисунок 1.237 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1 SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT переводится на высокий логический уровень
- сигнал SSPFSSOUT переводится на высокий логический уровень
- линия передачи данных SSPTXD произвольно переводится на низкий логический уровень
- сигнал nSSPOE переводится на высокий уровень, обеспечивая, таким образом, переход выходного контакта передатчика в высокоимпедансное состояние
- если модуль PrimeCell SSP сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится на низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня
- если модуль PrimeCell SSP сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится на высокий уровень, отключая выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня

Если работа модуля PrimeCell SSP разрешена, и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится на низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на

входную линию SSPRXD ведущего. Сигнал nSSPOE переводится на низкий уровень, переводя выходной контакт SSPTXD ведущего устройства в активное состояние.

По истечении полутакта сигнала на линию SSPTXD передаются допустимые данные от ведущего устройства. К этому моменту должны быть сформированы данные на линиях обмена, как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSPCLKOUT переводится на низкий логический уровень. Это значит, что данные регистрируются по заднему фронту и выдаются на линию по переднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSSOUT переводится на высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной неравномерной передачи данных линия SSPFSSOUT находится на низком логическом уровне между последовательными информационными словами и переводится на высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

1.3.12.2.5.2.3.16 Формат обмена данными SPI фирмы Motorola, SPO=1, SPH=0

Диаграммы одиночного и непрерывного обмена в режиме SPI с SPO=1, SPH=0 представлены на рисунках 1.238 и 1.239.

На Рисунке Рисунок 1. продемонстрирована диаграмма одиночного обмена в режиме SPI с SPO=1, SPH=0.

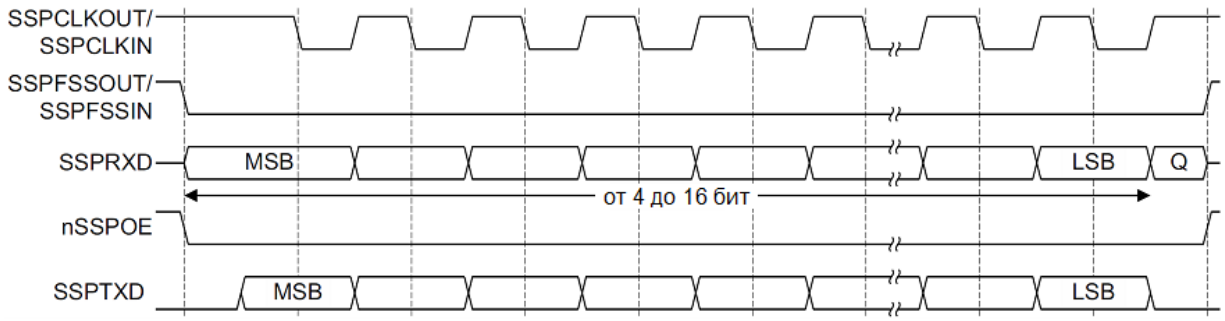


Рисунок 1.238 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1, SPH=0 (одиночный обмен)

На рисунке 1.239 представлена диаграмма непрерывного обмена в режиме SPI с SPO=1, SPH=0.

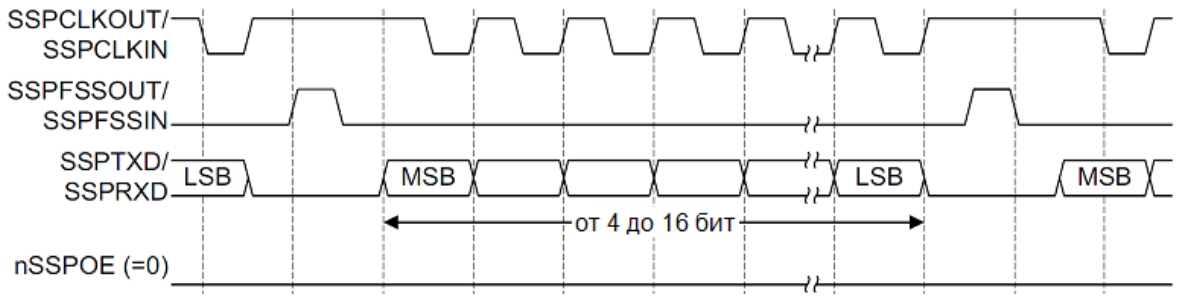


Рисунок 1.239 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1 SPH=0 (непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT переводится на высокий логический уровень
- сигнал SSPFSSOUT переводится на высокий логический уровень
- линия передачи данных SSPTXD произвольно переводится на низкий логический уровень
- сигнал nSSPOE переводится на высокий уровень, обеспечивая, таким образом, переход выходного контакта передатчика в высокоимпедансное состояние
- если модуль PrimeCell SSP сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится на низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня
- если модуль PrimeCell SSP сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится на высокий уровень, отключая выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня

Если работа модуля PrimeCell SSP разрешена, и в буфере FIFO передатчика содержатся корректные данные, сигнал SSPFSSOUT переводится на низкий логический уровень, что указывает на начало обмена данными и разрешает передачу данных от ведомого устройства на входную линию SSPRXD ведущего. Сигнал nSSPOE переводится на низкий уровень, переводя выходной контакт SSPTXD ведущего устройства в активное состояние.

По истечении полутакта сигнала на линию SSPTXD передаются допустимые данные от ведущего устройства. К этому моменту должны быть сформированы данные на линиях обмена, как ведущего, так и ведомого устройства. По истечении следующего полутакта сигнал SSPCLKOUT переводится на низкий логический уровень. Это значит, что данные регистрируются по заднему фронту и выдаются на линию по переднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSSOUT переводится на высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

Тем не менее, в режиме непрерывной неравномерной передачи данных на линии SSPFSSOUT должны формироваться импульсы высокого логического уровня между передачами каждого слова данных. Это связано с тем, что в режиме SPH=0 линия выбора ведомого устройства на низком уровне блокирует запись в сдвиговый регистр. Поэтому ведущее устройство должно переводить линию SSPFSSIN на высокий уровень по окончании передачи каждого кадра, разрешая, таким образом, запись новых данных, последовательно передаваемых периферийным устройством. По завершении приема последнего бита блока данных линия SSPFSSOUT переводится в состояние, соответствующее режиму ожидания, по истечении одного такта сигнала SSPCLKOUT.

1.3.12.2.5.2.3.17 Формат обмена данными SPI фирмы Motorola, SPO=1, SPH=1

На рисунке 1.240 продемонстрирована диаграмма одиночного и непрерывного обмена данными в режиме SPI с SPO=1, SPH=1.

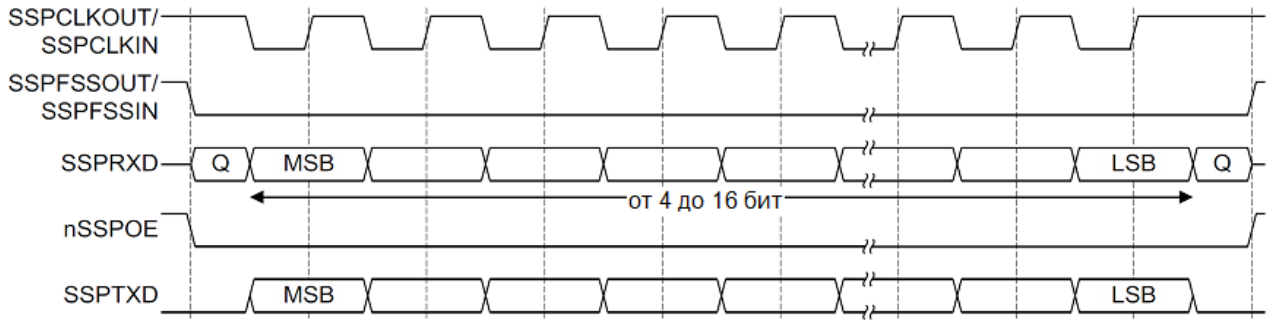


Рисунок 1.240 – Формат обмена по протоколу SPI фирмы Motorola, SPO=1,SPH=1 (одиночный и непрерывный обмен)

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT переводится на высокий логический уровень
- сигнал SSPFSSOUT переводится на высокий логический уровень
- линия передачи данных SSPTXD произвольно переводится на низкий логический уровень
- сигнал nSSPOE переводится на высокий уровень, обеспечивая, таким образом, переход выходного контакта передатчика в высокоимпедансное состояние
- если модуль PrimeCell SSP сконфигурирован как ведущее устройство, линия nSSPCTLOE переводится на низкий уровень, разрешая передачу сигнала на выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня
- если модуль PrimeCell SSP сконфигурирован как ведомое устройство, линия nSSPCTLOE переводится на высокий уровень, отключая выходной контакт SSPCLKOUT, разрешающий сигнал низкого уровня

Если работа модуля PrimeCell SSP разрешена, и в буфере FIFO передатчика содержатся допустимые данные, начало передачи инициализируется сигналом SSPFSSOUT от ведущего устройства, который переводится на низкий логический уровень. Сигнал nSSPOE переводится на низкий уровень, переводя выходной контакт SSPTXD ведущего устройства в активное состояние. По истечении полутакта сигнала SSPCLKOUT на соответствующих линиях обмена, как ведущего, так и ведомого устройств разрешена передача данных. В это же время включается линия SSPCLKOUT, на которой формируется задний фронт сигнала. Далее данные регистрируются по переднему фронту и выдаются на линию по заднему фронту сигнала SSPCLKOUT.

В случае передачи одного слова данных после приема его последнего бита линия SSPFSSOUT переводится на высокий логический уровень по истечении одного периода тактового сигнала SSPCLKOUT.

В режиме непрерывной неравномерной передачи данных линия SSPFSSOUT постоянно находится на низком логическом уровне до окончания приема последнего бита блока данных, после чего возвращается в состояние незанятости, как описано в вышеприведенном разделе.

В режиме непрерывной неравномерной передачи данных линия SSPFSSOUT находится на низком логическом уровне между последовательными информационными словами и переводится на высокий уровень по окончании приема последнего бита блока данных, как и в режиме передачи одного слова.

1.3.12.2.5.2.3.18 Формат обмена данными Microwire фирмы National Semiconductor

На рисунке 1.241 представлена диаграмма одиночного обмена в режиме Microwire, предлагаемом фирмой National Semiconductor. Рисунок 1.242 демонстрирует тот же формат при непрерывном обмене.

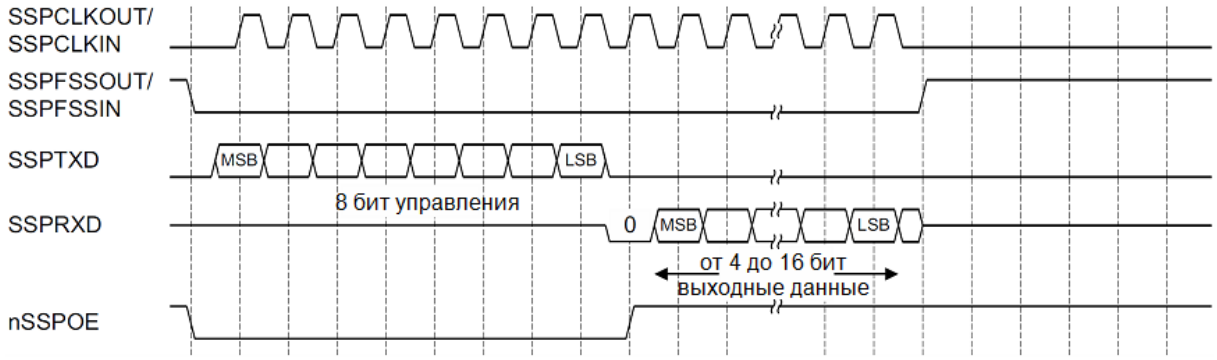


Рисунок 1.241 – Формат обмена по протоколу Microwire фирмы National Semiconductor (одиночный обмен)

Протокол передачи данных Microwire во многом схож с протоколом SPI, за исключением того, что обмен данными между ведущим и ведомыми устройствами в нем осуществляется в полудуплексном режиме вместо полнодуплексного. Каждая информационный обмен начинается с передачи модулем SSP «внекристальному» ведомому устройству специального восьмибитного управляющего слова. В течение всего времени передачи данной последовательности модуль PrimeCell SSP не обрабатывает каких-либо входных данных. После того, как сигнал передан и декодирован ведомым устройством, оно выдерживает паузу в один тактовый интервал после передачи последнего бита восьмибитной управляющей последовательности, после чего передает в адрес ведущего устройства запрошенные данные. Длительность блока данных от ведомого устройства может составлять от 4 до 16 бит. Таким образом, общая длительность информационного кадра составляет от 13 до 25 бит.

В данном режиме во время ожидания приемопередатчика:

- сигнал SSPCLKOUT переводится на низкий логический уровень
- сигнал SSPFSSOUT переводится на высокий логический уровень
- линия передачи данных SSPTXD произвольно переводится на низкий логический уровень
- сигнал nSSPOE переводится на высокий уровень, обеспечивая, таким образом, переход выходного контакта передатчика в высокоимпедансное состояние

Переход в режим информационного обмена происходит после записи управляющего байта в буфер FIFO передатчика. По заднему фронту сигнала SSPFSSOUT данные из буфера FIFO передатчика переносятся в сдвиговый регистр блока передатчика, откуда, начиная со старшего значащего разряда, восьмибитный управляющий кадр данных сдвигается без сохранения выдвигаемых разрядов в линию SSPTXD. Линия SSPFSSOUT остается на низком логическом уровне в течение всей передачи кадра. Линия SSPRXD в ходе данной передачи имеет три состояния.

Внешнее ведомое устройство осуществляет прием каждого управляющего бита данных в сдвиговый регистр по переднему фронту каждого сигнала SSPCLKOUT.

По окончании приема последнего бита ведомым устройством управляющий байт декодируется в течение одного тактового интервала, после чего ведомое устройство передает запрошенные данные в адрес модуля PrimeCell SSP. Биты данных передаются на линию SSPRXD по заднему фронту сигнала SSPCLKOUT. Модуль PrimeCell SSP, в свою очередь, регистрирует каждый бит по переднему фронту тактового сигнала SSPCLKOUT. В случае одиночного информационного обмена по окончании приема последнего бита слова данных сдвиговым регистром приемника сигнал SSPFSSOUT переводится на высокий уровень на время, соответствующее одному тактовому интервалу, что служит командой для переноса принятого слова данных из сдвигового регистра в буфер FIFO приемника.

Примечание

Внешнее ведомое устройство может перевести линию приема в режим с тремя состояниями или по заднему фронту сигнала SSPCLKOUT после получения сдвиговым регистром приемника младшего значащего бита, или после перехода линии SSPFSSOUT на высокий уровень.

В режиме непрерывной передачи данных передача начинается и завершается таким же образом, что и при одиночной передаче. Тем не менее, линия SSPFSSOUT постоянно находится на низком уровне, и передача данных осуществляется неравномерно. Управляющий байт следующего кадра следует непосредственно после младшего значащего бита полученных данных из текущего кадра. Каждое из полученных значений передается из сдвигового регистра на задний

фронт сигнала SSPCLKOUT после поступления младшего значащего бита кадра в модуль PrimeCell SSP.

На рисунке 1.242 представлена диаграмма неравномерного обмена в режиме Microwire, предлагаемом фирмой National Semiconductor.

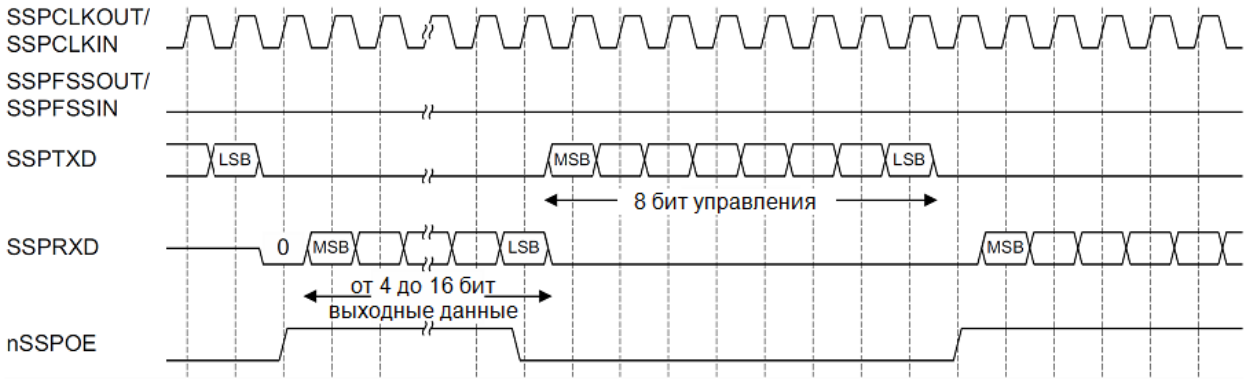


Рисунок 1.242 – Формат обмена по протоколу Microwire фирмы National Semiconductor (непрерывный обмен)

1.3.12.2.5.2.3.19 Требования к времени установки и удержания сигнала SSPFSSIN относительно тактового сигнала SSPCLKIN в режиме Microwire

Модуль PrimeCell SSP, работающий в режиме Microwire как ведомое устройство, регистрирует первый бит полученных данных по переднему фронту сигнала SSPCLKIN после перехода сигнала SSPFSSIN на низкий логический уровень. Ведущие устройства, формирующие независимый сигнал SSPCKLIN, должны обеспечивать достаточное время установки и удержания сигнала SSPFSSIN по отношению к переднему фронту сигнала SSPCLKIN.

Требования к установке и времени удержания сигнала представлены на рисунке 1.243.

По отношению к переднему фронту сигнала SSPCLKIN, по которому осуществляется регистрация первого бита полученных данных в приемнике ведомого модуля PrimeCell SSP, время установки сигнала SSPFSSIN должно быть, как минимум в два раза больше периода SSPCLK, на котором работает модуль.

По отношению к предыдущему переднему фронту сигнала SSPCLKIN время удержания сигнала SSPFSSIN должно составлять не менее одного периода тактового сигнала SSPCLK.

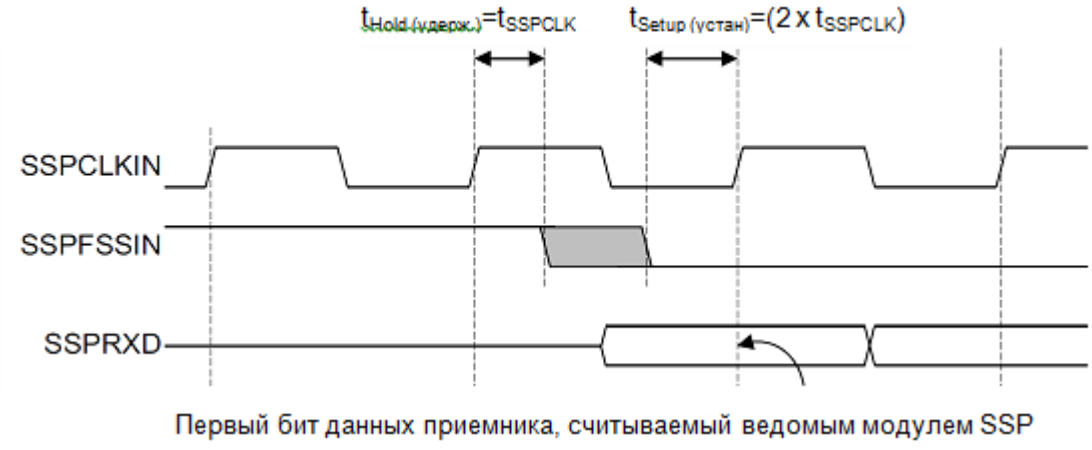


Рисунок 1.243 – Формат кадра Microwire, требования к времени установки и удержания сигнала SSPFSSIN

1.3.12.2.5.2.3.20 Примеры конфигурации модуля в ведущем и ведомом режимах

На рисунках 1.243, 1.244 и 1.245 показаны варианты подключения модуля PrimeCell SSP (PL022) к периферийным устройствам, работающим в ведущем или ведомом режиме.

Примечание

Модуль SSP (PL022) не поддерживает динамическое изменение режима ведущий-ведомый. Каждый приемопередатчик должен быть изначально сконфигурирован и подключен в одном из этих режимов.

На рисунке 1.243 показана работа модуля PrimeCell SSP (PL022) в двух режимах, ведущем и ведомом. Ведущее устройство способно передавать данные в адрес ведомого устройства по линии SSPTXD. Для ответной передачи данных ведомое устройство переводит сигнал nSSPOE на высокий уровень, разрешая, таким образом, прохождение сигнала от своей линии SSPTXD на вход SSPRXD ведущего устройства.

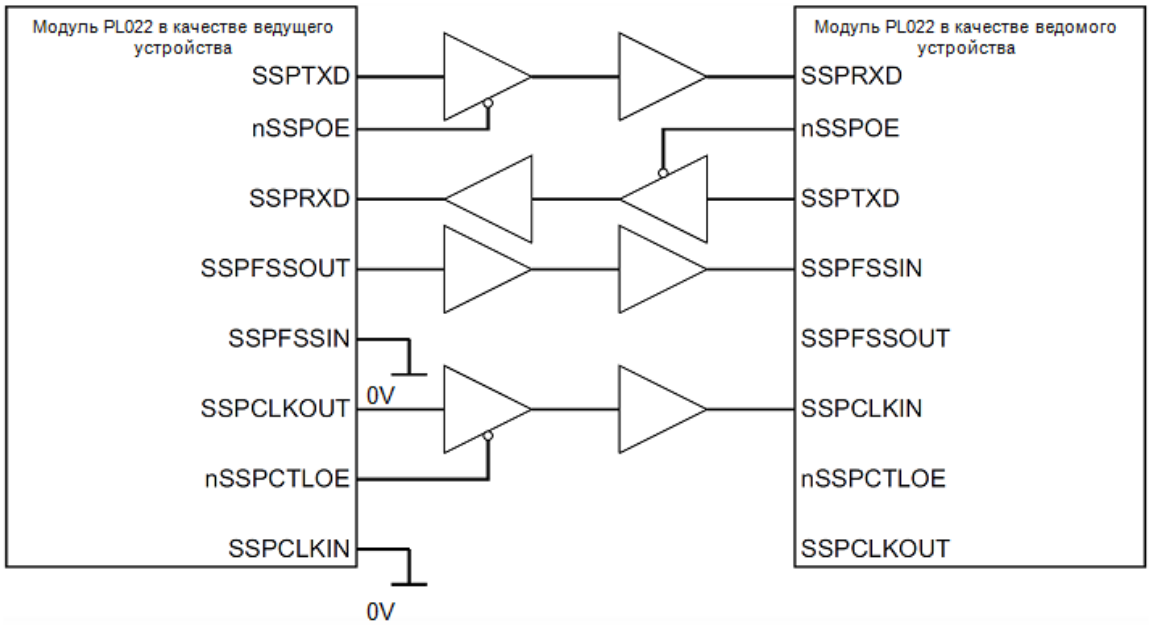


Рисунок 1.243 – Ведущее устройство PrimeCell SSP подключено к ведомому устройству PL022

На рисунке 1.244 показано подключение модуля PrimeCell SSP (PL022), сконфигурированного как ведущее устройство, к ведомому устройству, поддерживающему протокол SPI фирмы Motorola. Внешнее устройство сконфигурировано как ведомое путем установки в низкий логический уровень сигнала выбора ведомого устройства Slave Select (SS). Как и в предыдущем примере, ведущее устройство способно передавать данные в адрес ведомого устройства циркулярно через ведущий модуль PrimeCell SSP по линии SSPTXD. Ответная передача данных на входную линию SSPRXD ведущего устройства осуществляется только ведомым устройством по соответствующей линии MISO, поддерживающей протокол SPI.

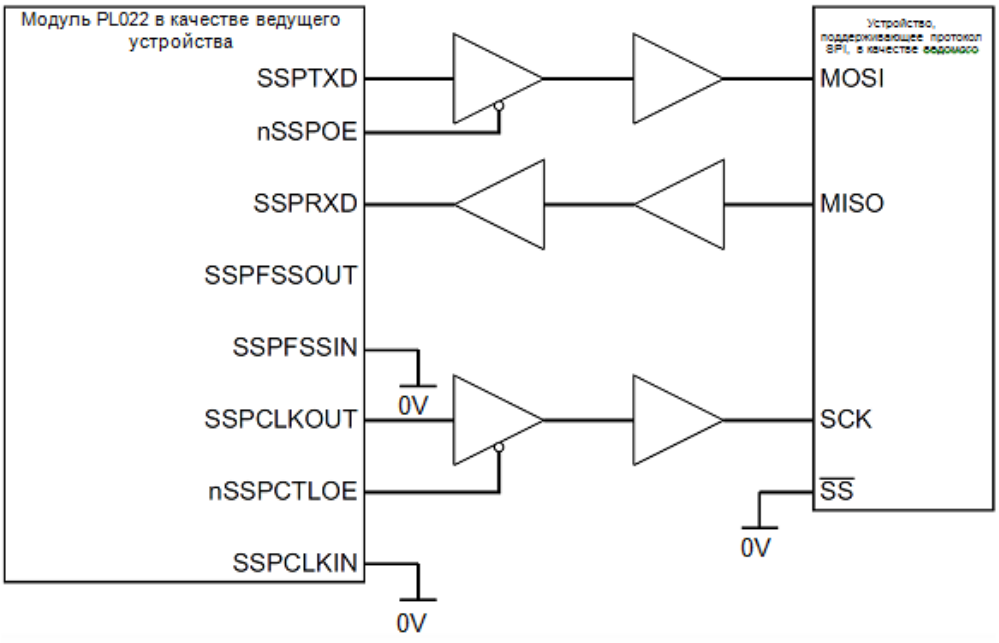


Рисунок 1.244 – Ведущее устройство PrimeCell SSP подключено к ведомому устройству, поддерживающему протокол SPI

На рисунке 1.245 показано ведущее устройство, поддерживающее протокол SPI фирмы Motorola, соединенное с модулем PrimeCell SSP (PL022), сконфигурированным для работы в ведомом режиме. В данном случае сигнал выбора ведомого устройства постоянно находится на высоком логическом уровне, что позволяет конфигурировать устройство в качестве ведущего. Ведущее устройство осуществляет передачу данных ведомому устройству по линии MOSI, поддерживающей протокол SPI, циркулярно, и ведомое устройство переводит свой сигнал nSSPOE на низкий уровень. Это обеспечивает передачу данных SSPTXD на линию MISO ведущего устройства.

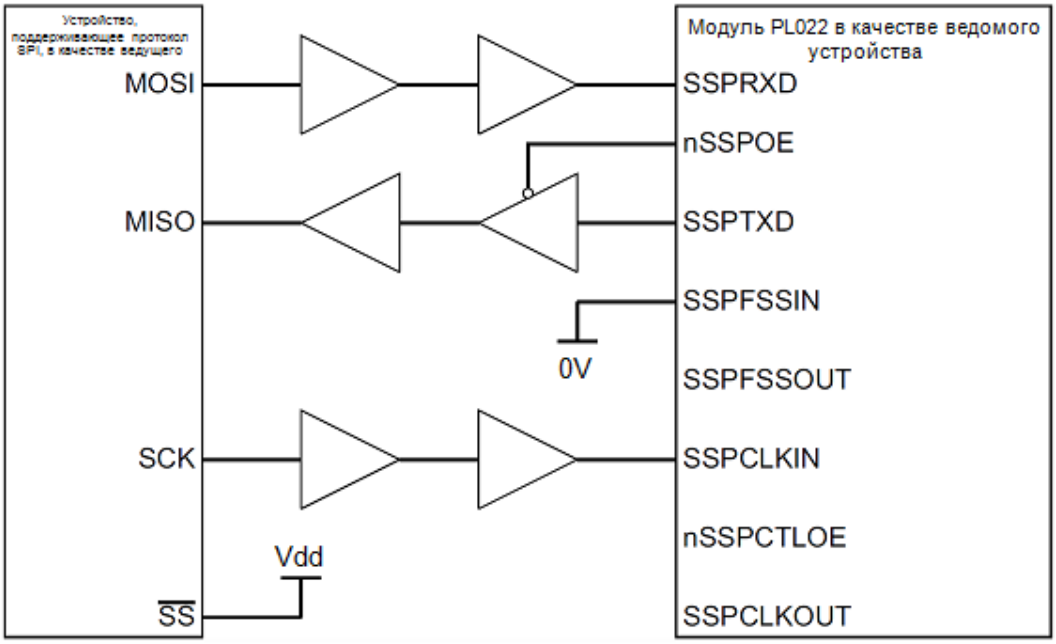


Рисунок 1.245 – Ведущее устройство, поддерживающее протокол SPI, подключено к ведомому устройству PrimeCell SSP

1.3.12.2.5.3 Программное управление модулем

1.3.12.2.5.3.1 О программном управлении модулем

Нижеприведенная информация относится к регистрам порта синхронной последовательной связи (PL022):

- Базовый адрес модуля не фиксирован и может быть различным в разных системах. Смещение каждого регистра относительно базового адреса фиксировано.
- Доступ к резервным или не используемым адресам не допускается. Попытка доступа к данным адресам может привести к непредсказуемому поведению системы.
- Если не указано особо в сопровождающем тексте:
  - не допускается модификация неопределенных разрядов регистра
  - при чтении неопределенные разряды регистра необходимо игнорировать
  - все разряды регистра сбрасываются на ноль системой или при сбросе по включению питания
- Условные обозначения типов доступа, приведенных в Таблице 1.700:

RW Чтение и запись
RO Только чтение
WO Только запись

Следующие адреса являются резервными и не должны использоваться в нормальном режиме функционирования:

- адреса со смещениями в диапазоне от +0x028 до +0x07C и от +0xFD0 до +0xFDC зарезервированы для перспективных расширений возможностей модуля
- адреса со смещениями в диапазоне от +0x080 до +0x088 зарезервированы для тестирования

1.3.12.2.5.3.2 Обобщенные данные о регистрах контроллера PrimeCell SSP

Таблица 1.700 – Обобщенные данные о регистрах модуля PrimeCell SSP

Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
+ 0x00	SSPCR0	RW	0x0000	16	Регистр управления 0 SSPCR0 (стр. 3-4)
+ 0x04	SSPCR1	RW	0x0	4	Регистр управления 1 SSPCR1 (стр. 3-5)
+ 0x08	SSPDR	RW	0x----	16	Регистр данных SSPDR (стр. 3-6)
+ 0x0C	SSPSR	RO	0x03	5	Регистр состояния SSPSR (стр. 3-7)
+ 0x10	SSPCPSR	RW	0x00	8	Регистр предварительного делителя частоты SSPCPSR (стр. 3-8)
+ 0x14	SSPIMSC	RW	0x0	4	Регистр установки и сброса маски прерывания SSPIMSC (стр. 3-9)
+ 0x18	SSPRIS	RO	0x8	4	Регистр состояния прерываний без учета маскирования SSPRIS (стр. 3-10)
+ 0x1C	SSPMIS	RO	0x0	4	Регистр состояния маскированных прерываний SSPMIS (стр. 3-11)

Продолжение таблицы 1.700

Смещение	Наименование	Тип	Значение после сброса	Размер, бит	Описание
+ 0x20	SSPICR	WO	0x0	4	Регистр сброса прерываний SSPICR (стр. 3-11)
+ 0x24	SSPDMACR	RW	0x0	2	Регистр управления прямым доступом к памяти SSPDMACR (стр. 3-12)
от + 0x28 до 0x7C	-	-	-	-	Зарезервирован
от + 0x80 до 0x8C	-	-	-	-	Зарезервирован для тестирования
от + 0x90 до 0xFCC	-	-	-	-	Зарезервирован
от + 0xFD0 до 0xFDC	-	-	-	-	Зарезервирован для перспективных расширений возможностей модуля
+ 0xFEO	SSPPeriphID0	RO	0x22	8	Регистры идентификации периферийных устройств SSPPeriphID0-3 (стр. 3-13)
+ 0xFE4	SSPPeriphID1	RO	0x10	8	
+ 0xFE8	SSPPeriphID2	RO	0x24	8	
+ 0xFEC	SSPPeriphID3	RO	0x00	8	
+ 0xFF0	SSPPCellID0	RO	0x0D	8	Регистры идентификации SSPPCellID0-3 модуля PrimeCell (стр. 3-16)
+ 0xFF4	SSPPCellID1	RO	0xF0	8	
+ 0xFF8	SSPPCellID2	RO	0x05	8	
+ 0xFFC	SSPPCellID3	RO	0xB1	8	

-

1.3.12.2.5.3.3 Описание регистров

1.3.12.2.5.3.3.1 Регистр управления 0 SSPCR0

Регистр управления SSPCR0 содержит пять битовых полей, предназначенных для управления блоками модуля PrimeCell SSP.

Таблица 1.701 – Назначение разрядов регистра SSPCR0

Биты	Наименование	Назначение
[15:8]	SCR	Скорость последовательного обмена. Значение поля SCR используется при задании скорости передачи и приема модуля PrimeCell SSP. Скорость передачи данных удовлетворяет соотношению: FSSPCLK CPSDVR x (1+SCR) где CPSDVSR – четное число в диапазоне от 2 до 254, запрограммированное через регистр SSPCPSR, и SCR – число от 0 до 255.
[7]	SPH	Фаза сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. «Формат обмена данными SPI фирмы Motorola».
[6]	SPO	Полярность сигнала SSPCLKOUT (используется только в режиме обмена SPI фирмы Motorola). См. «Формат обмена данными SPI фирмы Motorola» на стр. 2-10.

Продолжение таблицы 1.701

Биты	Наименование	Назначение	
[5:4]	FRF	Формат информационного кадра:	
		00	Формат обмена данными SPI фирмы Motorola
		01	Формат обмена данными SSI фирмы Texas Instruments
		10	Формат обмена данными Microwire фирмы National Semiconductor
		11	Резерв, неопределенный режим функционирования
[3:0]	DSS	Выбор размера слова данных:	
		0000	Резерв, неопределенный режим функционирования
		0001	Резерв, неопределенный режим функционирования
		0010	Резерв, неопределенный режим функционирования
		0011	4 бита
		0100	5 бит
		0101	6 бит
		0110	7 бит
		0111	8 бит
		1000	9 бит
		1001	10 бит
		1010	11 бит
		1011	12 бит
		1100	13 бит
		1101	14 бит
		1110	15 бит
		1111	16 бит

1.3.12.2.5.3.3.2 Регистр управления 1 SSPCR1

Регистр управления SSPCR1 содержит четыре битовых поля, предназначенных для управления блоками модуля PrimeCell SSP.

Таблица 1.702 – Назначение разрядов регистра SSPCR1

Биты	Наименование	Назначение
[15:4]	-	Резерв, при чтении результат не определен, при записи следует устанавливать в «0».
[3]	SOD	Запрет выходных линий в режиме ведомого устройства. Бит используется только в режиме ведомого устройства (MS=1). Это позволяет организовать двусторонний обмен данными в системах, содержащих одно ведущее и несколько ведомых устройств. При этом модуль PrimeCell SSP в качестве ведущего устройства будет способен передавать данные всем ведомым устройствам в системе, одновременно обеспечивая возможность передачи данных только одного ведомого устройства на соответствующую линию последовательного вывода данных. В подобных системах линии RXD, принимающие данные от множества ведомых устройств, могут быть связаны. Для работы в подобной системе необходимо установить бит SOD, если ведомый модуль PrimeCell SSP не должен в настоящее время осуществлять передачу данных в линию SSPTXD. 0 управление линией SSPTXD в ведомом режиме разрешено 1 управление линией SSPTXD в ведомом режиме запрещено

Продолжение таблицы 1.702

Биты	Наименование	Назначение
[2]	MS	Выбор ведущего или ведомого режима работы. Изменение данного бита может производиться только при деактивированном модуле PrimeCell SSP, когда SSE=0: 0 Конфигурирование устройства в качестве ведущего (по умолчанию) 1 Конфигурирование устройства в качестве ведомого
[1]	SSE	Активация порта синхронной последовательной связи: 0 Работа модуля SSP запрещена 1 Работа модуля SSP разрешена
[0]	LBM	Тестирование по шлейфу: 0 Нормальный режим работы 1 Выход регистра сдвига передатчика соединен с входом сдвигового регистра приемника

1.3.12.2.5.3.3.3 Регистр данных SSPDR

Регистр данных SSPDR имеет разрядность 16 бит. При чтении доступ к данным буфера FIFO приемника обеспечивается указателем чтения FIFO приемника. По мере извлечения данных из входящего кадра данных блоком приемника модуля PrimeCell SSP они помещаются в буфер FIFO приемника указателем записи FIFO приемника.

При осуществлении записи в регистр SSPDR производится запись данных, выделенных указателем записи, в буфер FIFO передатчика. Значения данных удаляются из буфера FIFO передатчика блоком передатчика по одному. Они загружаются в сдвиговый регистр передатчика, откуда последовательно выдаются на линию SSPTXD с заданной скоростью информационного обмена.

В случае если выбран размер информационного слова менее 16 бит, перед записью в буфер FIFO приемника необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова размером менее 16 бит автоматически выравниваются по правой границе в буфере приемника.

Таблица 1.703 – Назначение разрядов регистра SSPDR

Биты	Наименование	Назначение
[15:0]	DATA	Буфер FIFO передатчика/приемника: Чтение Буфер FIFO приемника Запись Буфер FIFO передатчика В случае, если модуль PrimeCell SSP запрограммирован на размер информационного слова менее 16 бит, перед записью необходимо обеспечить выравнивание данных по правой границе. Блок передатчика игнорирует неиспользуемые биты. Принятые информационные слова автоматически выравниваются по правой границе в блоке приемника.

1.3.12.2.5.3.3.4 Регистр состояния SSPSR

Регистр состояния SSPSR доступен только для чтения и содержит информацию о состоянии буферов FIFO приемника и передатчика и занятости модуля PrimeCell SSP.

Таблица 1.704 – Назначение разрядов регистра SSPSR

Биты	Наименование	Назначение
[15:5]	-	Резерв, при чтении результат не определен, при записи следует устанавливать в «0».
[4]	BSY	Флаг занятости модуля PrimeCell SSP, только для чтения: 0 Модуль SSP неактивен 1 модуль SSP в настоящее время передает и/или принимает данные, либо в буфере FIFO передатчика находятся данные
[3]	RFF	Буфер FIFO приемника заполнен, только для чтения: 0 Буфер FIFO приемника не заполнен 1 Буфер FIFO приемника заполнен
[2]	RNE	Буфер FIFO приемника не пуст, только для чтения: 0 Буфер FIFO приемника пуст 1 Буфер FIFO приемника не пуст
[1]	TNF	Буфер FIFO передатчика не заполнен, только для чтения: 0 Буфер FIFO передатчика заполнен 1 Буфер FIFO передатчика не заполнен
[0]	TFE	Буфер FIFO передатчика пуст, только для чтения: 0 Буфер FIFO передатчика не пуст 1 Буфер FIFO передатчика пуст

1.3.12.2.5.3.3.5 Регистр предварительного делителя частоты SSPCPSR

Регистр SSPCPSR является предварительным регистром тактовой частоты и используется для задания коэффициента деления, на который должен делиться входной сигнал SSPCLK перед дальнейшим использованием.

Программируемое в данном регистре значение должно быть целым числом в диапазоне от 2 до 254. Младший значащий разряд регистра принудительно устанавливается в ноль. Если записать в регистр SSPCPSR нечетное число, его последующее чтение даст результатом это число, но с установленным в ноль младшим битом.

Таблица 1.705 – Назначение разрядов регистра SSPCPSR

Биты	Наименование	Назначение
[15:8]	-	Резерв, при чтении результат не определен, при записи следует устанавливать в «0».
[7:0]	CPSDVSR	Коэффициент деления тактовой частоты. Записываемое значение должно быть целым числом в диапазоне от 2 до 254, в зависимости от частоты сигнала SSPCLK. Младший значащий разряд регистра принудительно устанавливается в ноль.

1.3.12.2.5.3.3.6 Регистр установки и сброса маски прерывания SSPIMSC

Регистр SSPIMSC является регистром установки и сброса маски прерывания. Данный регистр доступен для чтения и записи.

При чтении выдается текущее значение маски. При этом запись «1» в разряд разрешает соответствующее прерывание. Запись «0» производит сброс соответствующей маски.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Таблица 1.706– Назначение разрядов регистра SSPIMSC

Биты	Наименование	Назначение
[15:4]	Reserved	Резерв. Не подлежит модификации. При чтении выдаются нули
[3]	TXIM	Маска прерывания по заполнению буфера FIFO передатчика: 0 При заполнении на 50% и менее буфера FIFO передатчика производится маскирование. 1 При заполнении на 50% и менее буфера FIFO передатчика маскирование отсутствует.
[2]	RXIM	Маска прерывания по заполнению буфера FIFO приемника: 0 При заполнении на 50% и менее буфера FIFO приемника производится маскирование. 1 При заполнении на 50% и менее буфера FIFO приемника маскирование не производится.
[1]	RTIM	Маска прерывания по времени ожидания буфера FIFO приемника: 0 Буфер FIFO приемника не пуст, и не было попыток его чтения в течение времени ожидания с маскированием 1 Буфер FIFO приемника не пуст, и не было попыток его чтения в течение времени ожидания без маскирования
[0]	RORIM	Маска прерывания по переполнению буфера приемника: 0 Маскирование при переполнении буфера FIFO приемника. 1 При переполнении буфера FIFO приемника маскирование не производится.

1.3.12.2.5.3.3.7 Регистр состояния прерываний SSPRIS

Регистр SSPRIS содержит текущее состояние прерываний без учета маскирования. Данный регистр доступен только для чтения.

При чтении данного регистра выдается текущее исходное значение состояния соответствующего прерывания до события маскирования. Данные, записываемые в регистр, игнорируются.

Таблица 1.707 – Назначение разрядов регистра SSPRIS

Биты	Наименование	Назначение
[15:4]	Reserved	Резерв. Не подлежит модификации. При чтении выдаются нули
[3]	TXRIS	Состояние до маскирования прерывания SSPTXINTR
[2]	RXRIS	Состояние до маскирования прерывания SSPRXINTR
[1]	RTRIS	Состояние до маскирования прерывания SSPRTINTR
[0]	RORRIS	Состояние до маскирования прерывания SSPRORINTR

1.3.12.2.5.3.3.8 Регистр состояния маскированных прерываний SSPMIS

Регистр SSPMIS содержит текущее состояние прерываний с учетом маскирования. Данный регистр доступен только для чтения. При чтении данного регистра выдается текущее значение маскированного состояния соответствующего прерывания. Данные, записываемые в регистр, игнорируются.

Таблица 1.708 – Назначение разрядов регистра SSPMIS

Биты	Наименование	Назначение
[15:4]	Reserved	Резерв. Не подлежит модификации. При чтении выдаются нули
[3]	TXMIS	Присваивает буферу FIFO передатчика состояние маскированного прерывания SSPTXINTR после маскирования
[2]	RXMIS	Присваивает буферу FIFO приемника состояние маскированного прерывания SSPRXINTR после маскирования
[1]	RTMIS	Присваивает времени ожидания приемника значение, соответствующее состоянию маскированного прерывания SSPRTINTR после маскирования
[0]	RORMIS	Присваивает переполненному приемнику состояние маскированного прерывания SSPRORINTR после маскирования

1.3.12.2.5.3.3.9 Регистр сброса прерываний SSPICR

Регистр SSPICR является регистром сброса прерываний и доступен только для записи. Сброс признака прерывания по заданному событию производится путем записи «1» в соответствующий бит. Запись в любой из разрядов регистра «0» игнорируется.

Таблица 1.709 – Назначение разрядов регистра SSPICR

Биты	Наименование	Назначение
[15:2]	Reserved	Резерв. Не подлежит модификации. При чтении выдаются нули
[1]	RTIC	Сброс прерывания SSPRTINTR
[0]	RORIC	Сброс прерывания SSPRORINTR

1.3.12.2.5.3.3.10 Регистр управления прямым доступом к памяти SSPDMACR

Регистр SSPDMACR является регистром управления прямым доступом к памяти. Данный регистр доступен для чтения и записи.

После сброса все биты регистра обнуляются.

Таблица 1.710 – Назначение разрядов регистра SSPDMACR

Биты	Наименование	Назначение
[15:2]	Reserved	Резерв. Не подлежит модификации. При чтении выдаются нули
[1]	TXDMAE	Использование ПДП при передаче. Если бит установлен в «1», разрешено формирование запросов ПДП для обслуживания буфера FIFO передатчика
[0]	RXDMAE	Использование ПДП при приеме. Если бит установлен в «1», разрешено формирование запросов ПДП для обслуживания буфера FIFO приемника

1.3.12.2.5.3.3.11 Регистры идентификации периферийных устройств SSPPeriphID0-3

Регистры SSPPeriphID0-3 – это четыре восьмибитовых регистра, охватывающие адреса от 0xFE0 до 0xFEC. Регистры, в принципе, могут рассматриваться как один 32-битный регистр. Регистры, доступные только для чтения, предоставляют следующие опции для периферийного устройства:

PartNumber[11:0] (Серийный номер устройства)

Используется для идентификации периферийного устройства. Используется трехразрядный код устройства 0x022.

Designer ID[19:12] (Идентификатор разработчика)

Идентификатор разработчика. Идентификатором компании ARM Ltd является «0x41, ASCII A».

Revision[23:20] (Версия)

Номер версии периферийного устройства. Номер начинается с «0» и зависит от версии.

Configuration[31:24] (Конфигурация)

Конфигурация периферийного устройства. Значением конфигурации является «0».

Ограничения по использованию **Отсутствуют.**

Конфигурации Все конфигурации SSP.

Характеристики См. Таблицу 1.700.

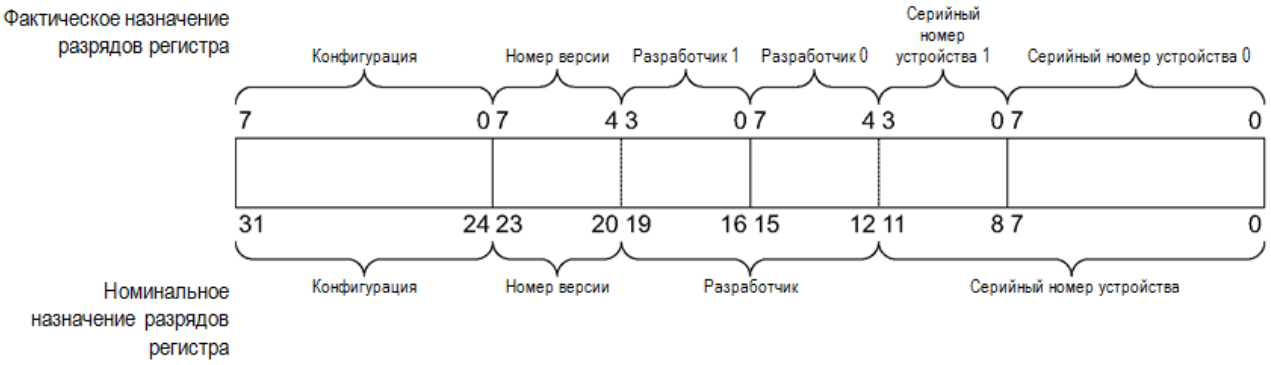


Рисунок 1.247 – Назначение разрядов регистра идентификации периферийных устройств

Примечание

При разработке карты распределения памяти необходимо помнить, что объем памяти регистра составляет 4 КБ. Доступ к памяти всех регистров идентификации периферийных устройств должен быть 32-битным; доступ осуществляется согласно инструкциям LDR и STR.

Регистр SSPPeriphID0

В регистр SSPPeriphID0 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.711 – Назначение разрядов регистра SSPPeriphID0

Биты	Наименование	Назначение
[15:8]	Reserved	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
[7:0]	PartNumber0	Результатом повторного считывания данных битов будет «0x22»

Регистр SSPPeriphID1

В регистр SSPPeriphID1 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.712– Назначение разрядов регистра SSPPeriphID1

Биты	Наименование	Назначение
[15:8]	Reserved	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
[7:4]	Designer0	Результатом повторного считывания данных битов будет «0x1»
[3:0]	PartNumber1	Результатом повторного считывания данных битов будет «0x0»

Регистр SSPPeriphID2

В регистр SSPPeriphID2 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.713– Назначение разрядов регистра SSPPeriphID2

Биты	Наименование	Назначение
[15:8]	Reserved	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
[7:4]	Revision	Данные биты возвращают номер версии периферийного устройства [3:0]
[3:0]	Designer1	Результатом повторного считывания данных битов будет «0x4»

Регистр SSPPeriphID3

В регистр SSPPeriphID3 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.714 – Назначение разрядов регистра SSPPeriphID3

Биты	Наименование	Назначение
[15:8]	Reserved	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
[7:0]	Configuration	Результатом повторного считывания данных битов будет «0x00»

1.3.12.2.5.3.3.12 Регистры идентификации модуля PrimeCell SSPPCellID0-3

Регистры SSPPCellID0-3 – это четыре восьмибитовых регистра, охватывающие адреса от 0xFF0 до 0xFFC. Регистры, в принципе, могут рассматриваться как один 32-битный регистр. Данный регистр используется в качестве стандартной системы идентификации периферийных устройств. Установленным значением регистра SSPPCellID является «0xB105F00D».

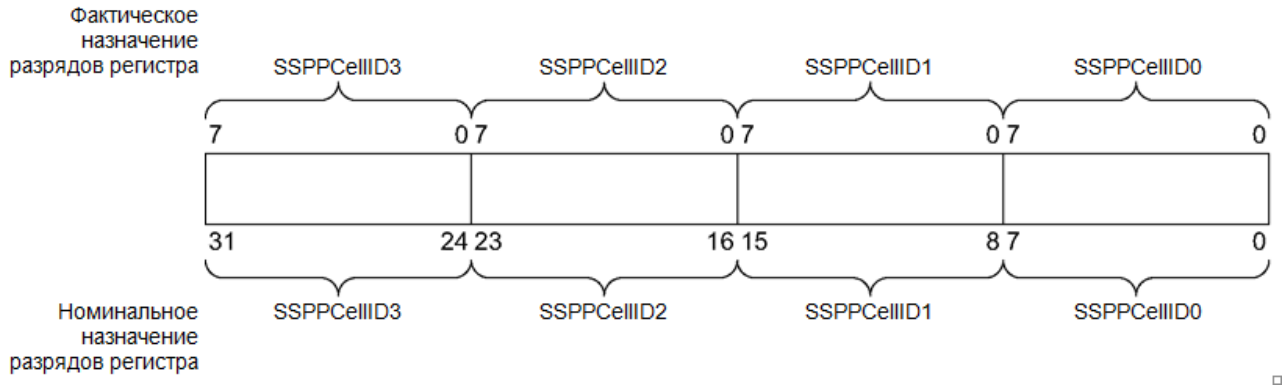


Рисунок 1.248– Назначение разрядов регистра идентификации модуля PrimeCell

Регистр SSPPCellID0

В регистр SSPPCellID0 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.715 – Назначение разрядов регистра SSPPCellID0

Биты	Наименование	Назначение
[15:8]	Reserved	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
[7:0]	SSPPCellID0	Результатом повторного считывания данных битов будет «0x0D»

Регистр SSPPCellID1

В регистр SSPPCellID1 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.716– Назначение разрядов регистра SSPPCellID1

Биты	Наименование	Назначение
[15:8]	Reserved	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
[7:0]	SSPPCellID1	Результатом повторного считывания данных битов будет «0xF0»

Регистр SSPPCellID2

В регистр SSPPCellID2 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.717 – Назначение разрядов регистра SSPPCellID2

Биты	Наименование	Назначение
[15:8]	Reserved	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
[7:0]	SSPPCellID2	Результатом повторного считывания данных битов будет «0x05»

Регистр SSPPCellID3

В регистр SSPPCellID3 записано фиксированное значение, и значения полей регистра определяют параметр сброса.

Таблица 1.718 – Назначение разрядов регистра SSPPCellID3

Биты	Наименование	Назначение
[15:8]	Reserved	Резерв, при чтении неопределенного бита его значение должно интерпретироваться как нулевое
[7:0]	SSPPCellID3	Результатом повторного считывания данных битов будет «0xB1»

-

1.3.12.2.5.3.4 Прерывания

Модуль PrimeCell SSP генерирует пять линий запроса на прерывание. Четыре из них являются независимыми маскируемыми с активным высоким логическим уровнем:

- SSPRXINTR запрос на обслуживание буфера FIFO приемника. См. «SSPRXINTR».
- SSPTXINTR запрос на обслуживание буфера FIFO передатчика модуля PrimeCell SSP. См. «SSPTXINTR».

- SSPRORINTR переполнение буфера FIFO приемника модуля PrimeCell SSP. См. «SSPRORINTR».
- SSPRTINTRвремя ожидания модуля PrimeCell SSP См. «SSPRTINTR».

Пятая линия является объединенной линией запроса на прерывание SSPINTR. См. «SSPINTR».

Каждый из четырех независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре маски SSPIMSC. Установка соответствующего бита маски на высокий уровень разрешает прерывание.

Доступность как индивидуальных, так и общей линии запроса позволяет организовать обслуживание прерываний в системе, как путем применения глобальной процедуры обработки, так и с помощью драйверов устройств, построенных по модульному принципу.

Предусмотрены независимые линии запроса на прерывание потока принимаемых и передаваемых динамических данных SSPTXINTR и SSPRXINTR, отличные от линии прерываний состояний, что позволяет производить считывание или запись данных в соответствии с уровнями срабатывания буферов FIFO приемника или передатчика.

Признаки возникновения каждого из условий прерывания можно считать из регистров SSPRIS и SSPMIS.

1.3.12.2.5.3.4.1 SSPRXINTR

Прерывание по заполнению буфера FIFO приемника формируется в случае, если буфер приемника содержит четыре или более допустимых слов данных.

1.3.12.2.5.3.4.2 SSPTXINTR

Прерывание по заполнению буфера FIFO передатчика формируется в случае, если буфер передатчика содержит четыре или менее допустимых слов данных. Состояние прерывания SSPTXINTR не зависит от значения сигнала разрешения работы модуля PrimeCell SSP, что позволяет организовать взаимодействие программного обеспечения с передатчиком одним из двух способов.

- запись данных в буфер FIFO передатчика заблаговременно перед активизацией модуля PrimeCell SSP и разрешения прерываний
- предварительное разрешение работы модуля PrimeCell SSP и формирования прерываний и заполнение буфера передатчика FIFO в ходе проведения процедуры обслуживания прерываний

1.3.12.2.5.3.4.3 SSPRORINTR

Прерывание по переполнению буфера FIFO приемника SSPORINTR формируется в случае, если буфер уже заполнен, и блоком приемника осуществлена попытка записать в него дополнительный кадр данных. При этом данные перезаписываются в регистре сдвига приемника, но в буфер FIFO приемника не заносятся.

1.3.12.2.5.3.4.4 SSPRTINTR

Прерывание по времени ожидания FIFO приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход модуля PrimeCell SSP не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Данный механизм обеспечивает осведомленность пользователя о наличии в буфере FIFO приемника необработанных данных. Прерывание по времени ожидания снимается либо после считывания данных из буфера FIFO приемника до его опустошения, либо после приема новых данных по входной линии SSPRXD. Кроме того, данное прерывание может быть снято путем внесения значения в бит RTIC регистра SSPICR.

1.3.12.2.5.3.4.5 SSPINTR

Все описанные сигналы запроса на прерывание также объединяются в общую линию SSPINTR путем объединения по схеме ИЛИ отдельных сигналов с учетом маскирования. Общий выход может быть подключен к системному контроллеру прерывания, что позволит обеспечить дополнительное маскирование запросов на уровне периферийных устройств.

Объединенный сигнал запроса на прерывание формируется в случае формирования и активации одного из четырех независимых вышеперечисленных сигналов.

1.3.12.2.6 Контроллер GSPI/SDIO

1.3.12.2.6.1 Общее описание контролера SPI/SDIO

Контроллер SPI/SDIO состоит из двух независимых контроллеров, внешние порты которых мультиплексируются на общие буфера ввода-вывода микросхемы.

Описание каждого из контроллеров приведено ниже.

1.3.12.2.6.2 Структурная схема контролера SPI/SDIO

Структурная схема контроллера SPI/SDIO представлена на рисунке1.249.

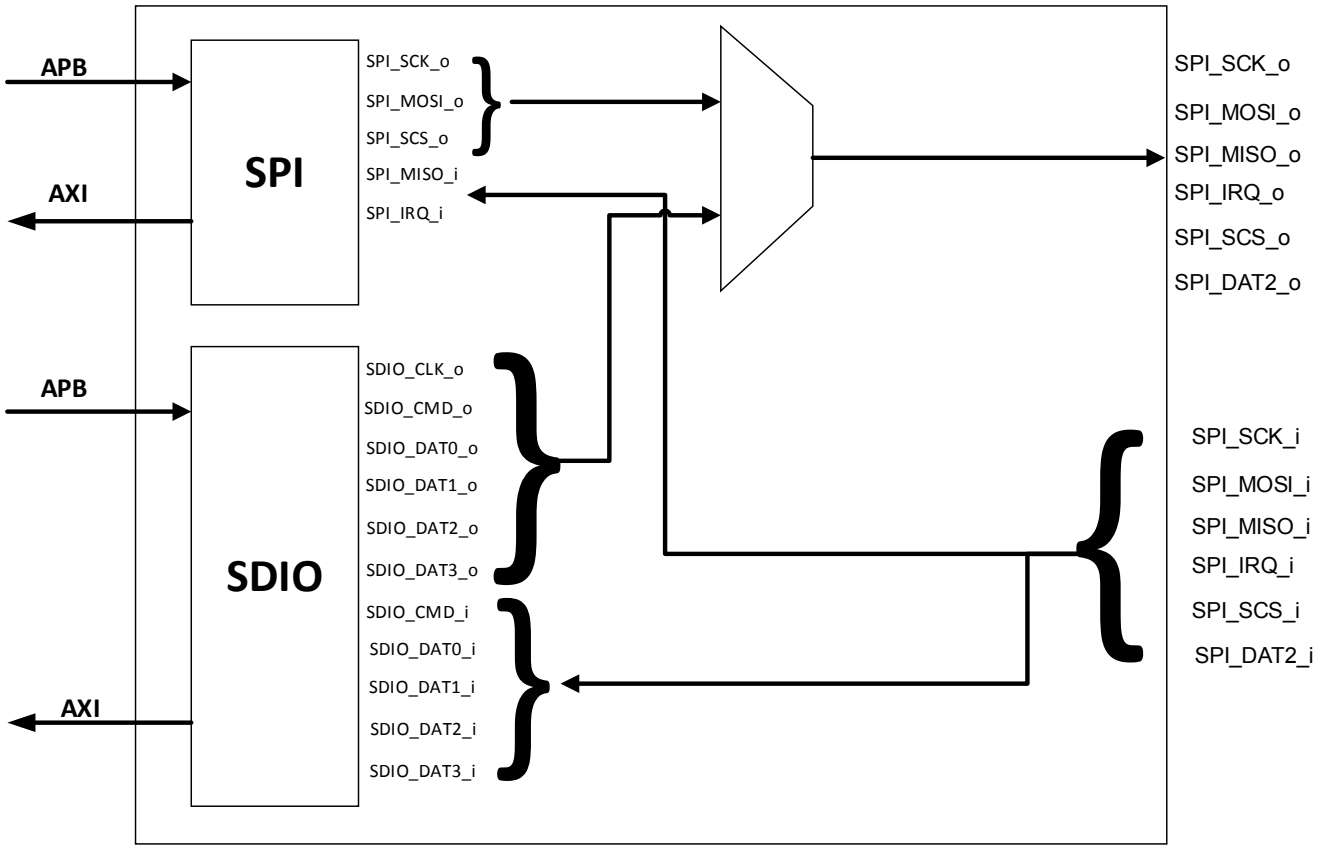


Рисунок 1.249 – Структурная схема контроллера SPI/SDIO

1.3.12.2.6.3 Общее описание контроллера SPI

Контроллер SPI расширяет функционал СФ-блока ARM SSP021, добавляя поддержку DMA операций, внешнего сигнала прерывания GSPI, а также программного управления сигналом Slave Select.

1.3.12.2.6.3.1 Структурная схема контроллера SPI

Структурная схема контроллера SPI представлена на 1.250.

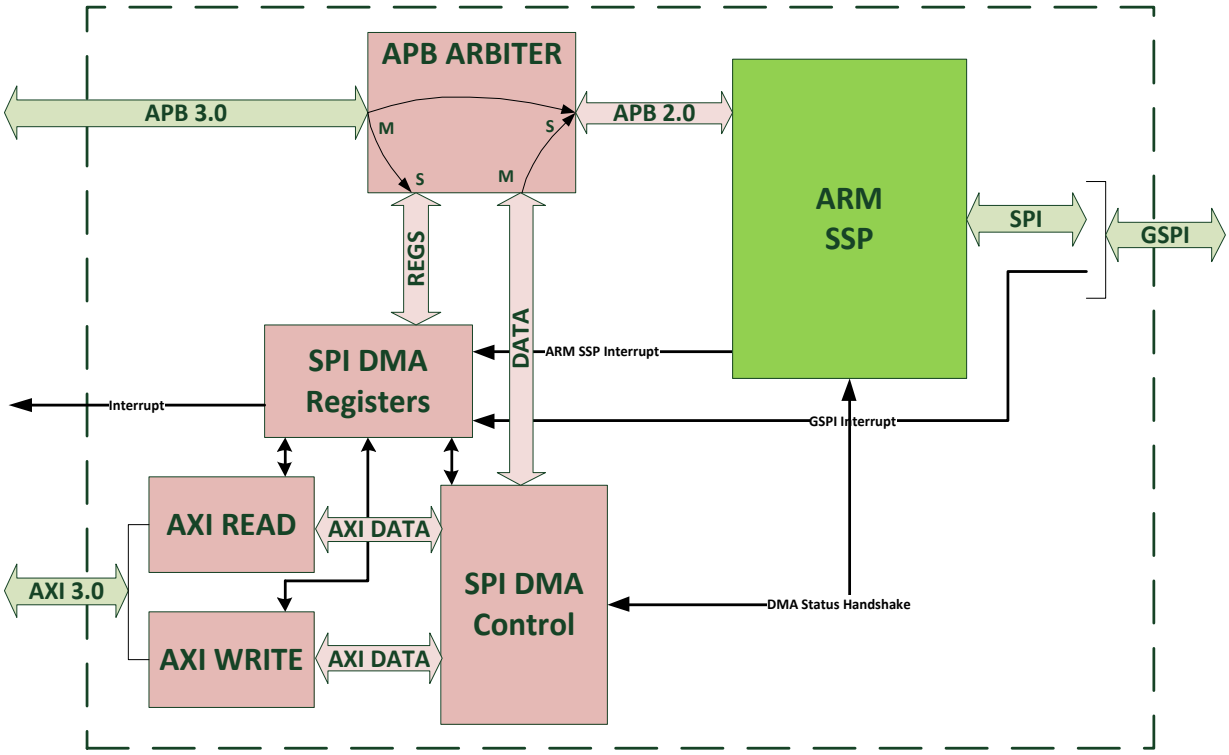


Рисунок 1.250 – Структурная схема контроллера SPI

1.3.12.2.6.3.2 Принципы функционирования контроллера SPI

1.3.12.2.6.3.2.1 Работа с контроллером ARM SSP без использования DMA

Работать без использования DMA нужно только при длине SPI пересылки в 1 слово. При длине пересылки в 2 слова и больше возможно и рекомендуется использование DMA.

1. Замаскировать все прерывания, кроме прерывания от ARM SSP (регистр SPI_IRQMASKS, бит 0).
2. Работать с контроллером ARM SSP в соответствии с документацией (при этом НЕ включать поддержку DMA). Статусный регистр SPI_STATUS при этом можно не использовать.

1.3.12.2.6.3.2.2 Работа с контроллером ARM SSP в режиме DMA

В режиме DMA работа возможна при длине SPI пересылки от 2-х слов и выше. Длина слова может составлять 8 либо 16 бит, при других длинах (от 4 до 16 бит по спецификации) необходима специальная обработка входных и выходных DMA буферов.

1. Настроить маски прерываний. Основным рабочим прерыванием в данном режиме является прерывание по концу DMA буфера канала записи AXI (бит 2 регистров SPI_IRQMASKS и SPI_STATUS).

2. Настроить параметры SPI пересылки контроллера ARM SSP в соответствии с документацией (частоту работы SPI, захват/передачу данных по фронту/спаду синхросигнала, длину слова SPI и другие), а также включить контроллер ARM SSP.

3. Включить поддержку DMA контроллера ARM SSP (как RX, так и TX, регистр SSPDMACR).

4. Выбрать длину слова данных SPI – 8 либо 16 бит.

5. Настроить параметры DMA каналов чтения и записи (начальные и конечные адреса буферов). При этом размеры буферов на чтение и запись должны совпадать. Настроить параметры шины AXI.

6. Разрешить работу DMA канала чтения (регистр SPI_AXIR_BUFENA).

7. Дождаться поступления прерывания по концу DMA буфера канала записи AXI. Крайне не рекомендуется обращаться к регистрам контроллера ARM SSP во время выполнения DMA операции (после разрешения работы DMA канала чтения и до поступления прерывания по концу DMA буфера канала записи AXI), поскольку была замечена нестабильная работа ARM SSP. Операция завершена.

1.3.12.2.6.3.2.3 Особенности работы с прерываниями

При обработке прерываний (всех, кроме внешнего прерывания GSPI и прерывания от контроллера ARM SSP) для сброса сигнала прерывания достаточно прочитать регистр статуса SPI_STATUS.

Для сброса сигнала прерывания от контроллера ARM SSP необходимо записать нужное значение в регистр SSPICR контроллера ARM SSP. При этом после сброса прерывания бит 0 регистра SPI_STATUS останется в «1», для сброса данного бита необходимо считать регистр SPI_STATUS.

Внешнее прерывание GSPI транслируются напрямую от внешнего источников, сбрасывается оно также на стороне внешнего источника.

1.3.12.2.6.3.2.4 Программный сброс

Записать «1» в регистр SPI_SOFTRST. Программный сброс считается завершенным, если из данного регистра считывается нулевое значение.

1.3.12.2.6.3.2.5 Программное управление сигналом Slave Select

С помощью регистра SPI_SOFTSS можно программно управлять сигналом Slave Select интерфейса SPI.

1.3.12.2.6.4 Общее описание контроллера SDIO

Контроллер интерфейса SDIO реализует хост-интерфейс стандарта SD/SDHC/SDIO 2.0 и позволяет взаимодействовать с запоминающими устройствами, поддерживающими интерфейс стандарта SD/SDHC 2.0 и устройствами ввода-вывода, поддерживающими интерфейс стандарта SDIO 2.0.

Ниже перечислены основные функции и особенности контроллера SDIO:

- Полная поддержка спецификации SDIO 2.0 и спецификации SDHC2.0 (без CPRM);
- Встроенный контроллер ПДП;
- Поддержка шины данных шириной 4 бита с возможностью работать в режиме 1-битной шины;

- Поддержка блоков данных размером до 512 байт;
- Поддержка команд чтения и записи нескольких блоков данных;
- Полный список поддерживаемых команд: CMD0, CMD2, CMD3, CMD4, CMD5, CMD6, CMD7, CMD8, CMD9, CMD10, CMD12, CMD13, CMD15, CMD16, CMD17, CMD18, CMD24, CMD25, CMD27, CMD28, CMD29, CMD30, CMD32, CMD33, CMD38, CMD42, CMD52, CMD53, CMD55, CMD56, ACMD6, ACMD13, ACMD22, ACMD23, ACMD41, ACMD42, ACMD51.

1.3.12.2.6.4.1 Структурная схема контроллера SDIO

Структура контроллера SDIO показана на рисунке 1.251.

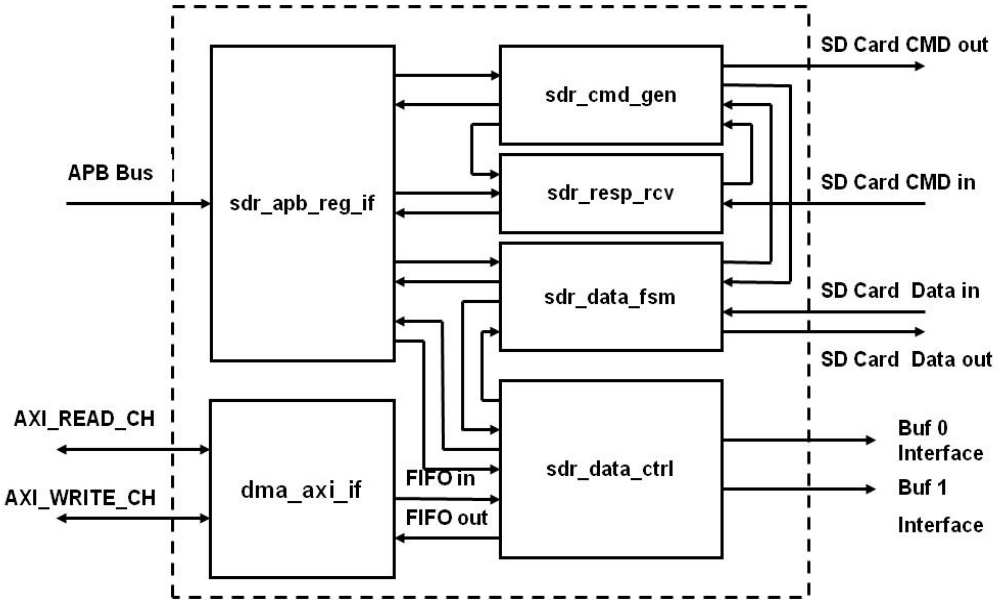


Рисунок 1.251 – Структура контроллера SDIO

1.3.12.2.6.4.2 Принципы функционирования контроллера SDIO

В данном разделе описаны алгоритмы работы с регистрами контроллера SDIO как в режиме взаимодействия с SD картой памяти, так и в режиме взаимодействия с устройством ввода-вывода SDIO

1.3.12.2.6.4.2.1 Алгоритм инициализации карты SD

SD хост-контроллер в начале работы должен проинициализировать SD карту в соответствии со спецификацией SD Card. Для этого необходимо настроить регистры блока контроллера SDIO так, как это показано на Рисунок 1.. Перед началом инициализации карты в контроллере SDIO необходимо включить обработку ошибок, установив в 1 биты EN_CMD_TO, EN_CMD_CRC, EN_CMD_EB, EN_CMD_IDX, EN_DAT_CRC, и EN_DAT_EB регистра SDR_Error_Enable_REG (смещение адреса = 0x14).

Для того, чтобы послать команду SD карте, необходимо записать аргумент команды в регистр SDR_COMD_ARGUMENT_REG (смещение адреса 0x08), а затем в другие атрибуты команды в регистр SDR_CTRL_REG (смещение адреса 0x04).

Например, для того, чтобы послать команду CMD8 необходимо сначала записать в регистр аргумента SDR_COMD_ARGUMENT_REG (смещение адреса 0x08) значение 0x000001aa (биты [7:0]=10101010b – это стандартная проверочная последовательность, а биты [11:8]=0001b сигнализируют о том, что карта способна работать при поданном на нее напряжении от 2,7В до 3,6В). Затем необходимо записать в регистр SDR_CTRL_REG (смещение адреса 0x04) значение 0x00083811 (где биты [21:16]=001000b - это индекс команды 8), и данная запись в регистр инициирует посылку команды карте SD.

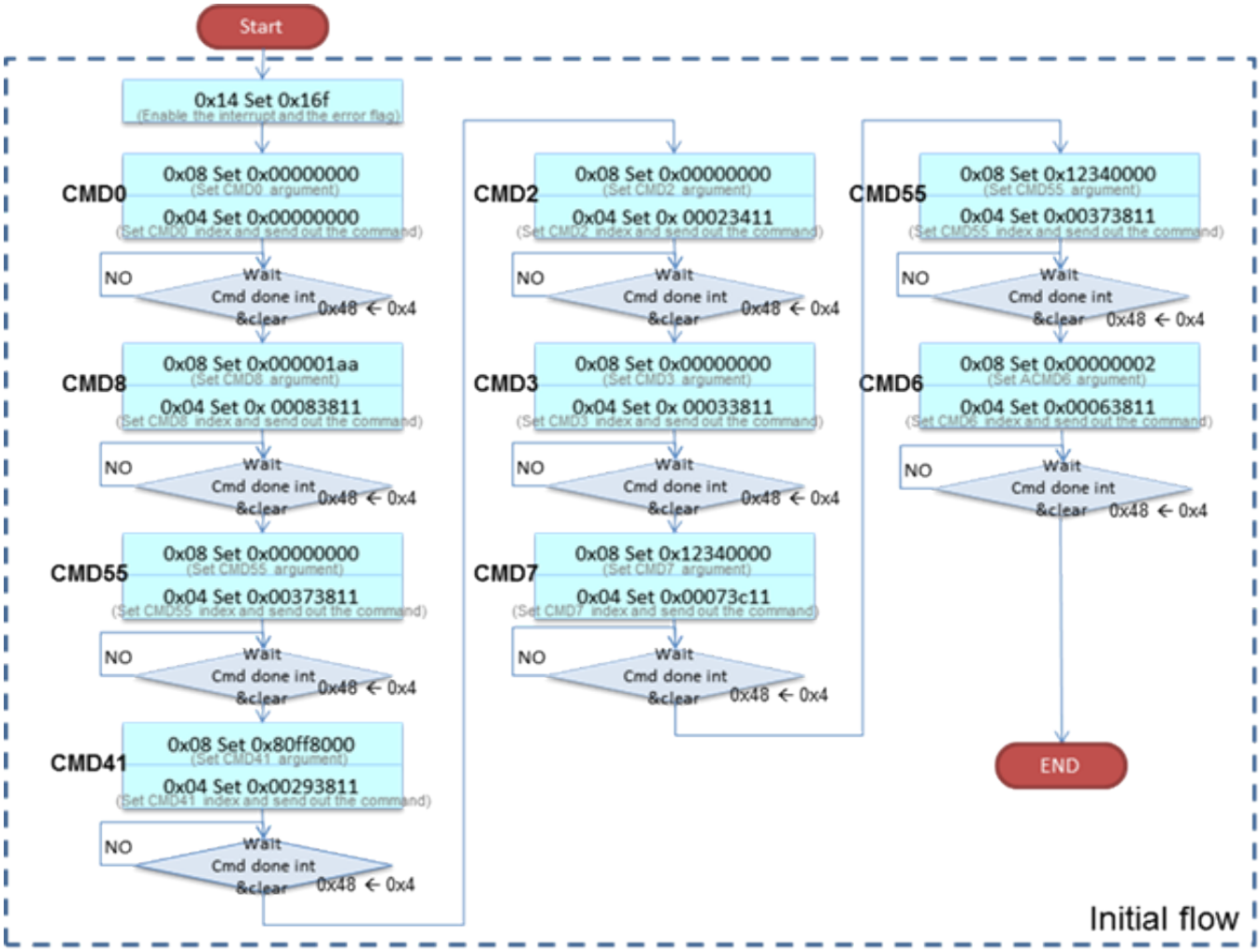


Рисунок 1.252 – Настройка регистров контроллера SDIO

1.3.12.2.6.4.2.2 Алгоритм чтения из карты SD одного блока данных

На рисунке 1.253 показана схема поочередной работы буферов данных. На данном рисунке в состоянии 1 оба буфера пусты и в буфер 0 начинают поступать считанные данные из SD карты. В состоянии 2 буфер 0 полон и считанные данные из SD карты начинают поступать в буфер 1, одновременно с этим данные из буфера 0 пересылаются через интерфейс AXI в системную память. После опустошения буфера 0 и заполнения буфера 1 буфера меняются местами – в буфер 0 снова записываются считанные данные из SD карты, а данные из буфера 1 пересылаются через интерфейс AXI в системную память. Так происходит до получения последней порции данных, которая записывается из буфера 1 через интерфейс AXI в системную память (состояние 3 на рисунке 1.253).

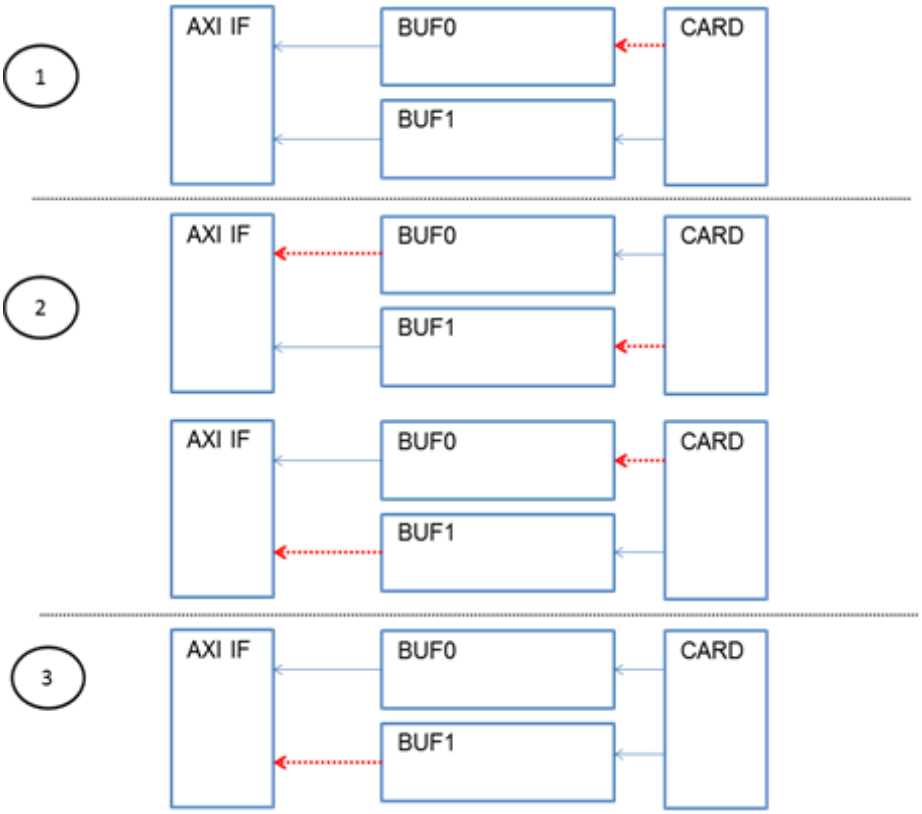


Рисунок 1.253 – Схема поочередной работы буферов данных

На рисунке 1.254 представлена последовательность программирования регистров блока контроллера SDIO для чтения одного или нескольких одиночных блоков данных из карты SD. При посылке первой команды чтения одного блока данных CMD17 буфера данных пусты и в буфер 0 будут считаны данные из карты SD. При посылке последующих команд CMD17 необходимо чередовать работы буферов 0 и 1 так, как это было показано на рисунке 1.253 (данные из SD карты считываются в один из буферов одновременно с отправкой данных по интерфейсу AXI в системную память из другого). Регистры, которые в соответствии с алгоритмом на рисункеРисунок 1. необходимо программировать одновременно, можно программировать последовательно (например, в таком порядке: 0x0C, 0x00, 0x08, 0x04, 0x50, 0x44, 0x40, и последним 0x38).

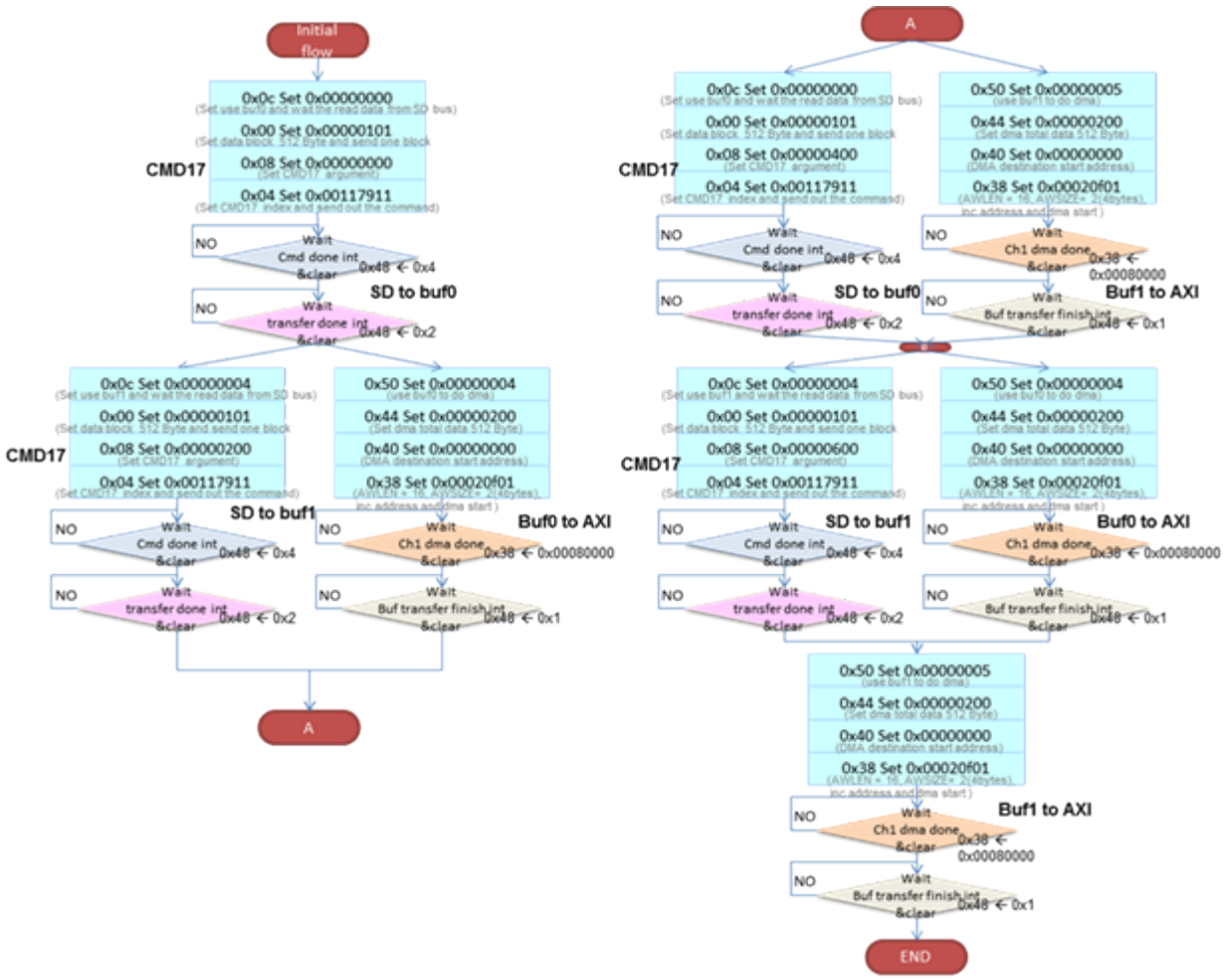


Рисунок 1.254 – Последовательность программирования регистров блока контроллера SDIO для чтения одного или нескольких одиночных блоков данных их карты SD

1.3.12.2.6.4.2.3 Алгоритм записи в SD карту одного блока данных

На рисунке 1.255 показана схема поочередной работы буферов данных. На данном рисунке в состоянии 1 оба буфера пусты и в буфер 0 начинают поступать считанные данные из системной памяти через интерфейс AXI. В состоянии 2 буфер 0 полон и считанные из системной памяти данные начинают передаваться в SD карту, одновременно с этим данные из системной памяти через интерфейс AXI записываются в буфера 1. После опустошения буфера 0 и заполнения буфера 1 буфера меняются местами – в буфер 0 снова записываются считанные данные из системной памяти, а данные из буфера 1 передаются в SD карту. Так происходит до отправки последней порции данных, которая записывается из буфера 1 в SD карту (состояние 3 на рисунке 1.253.

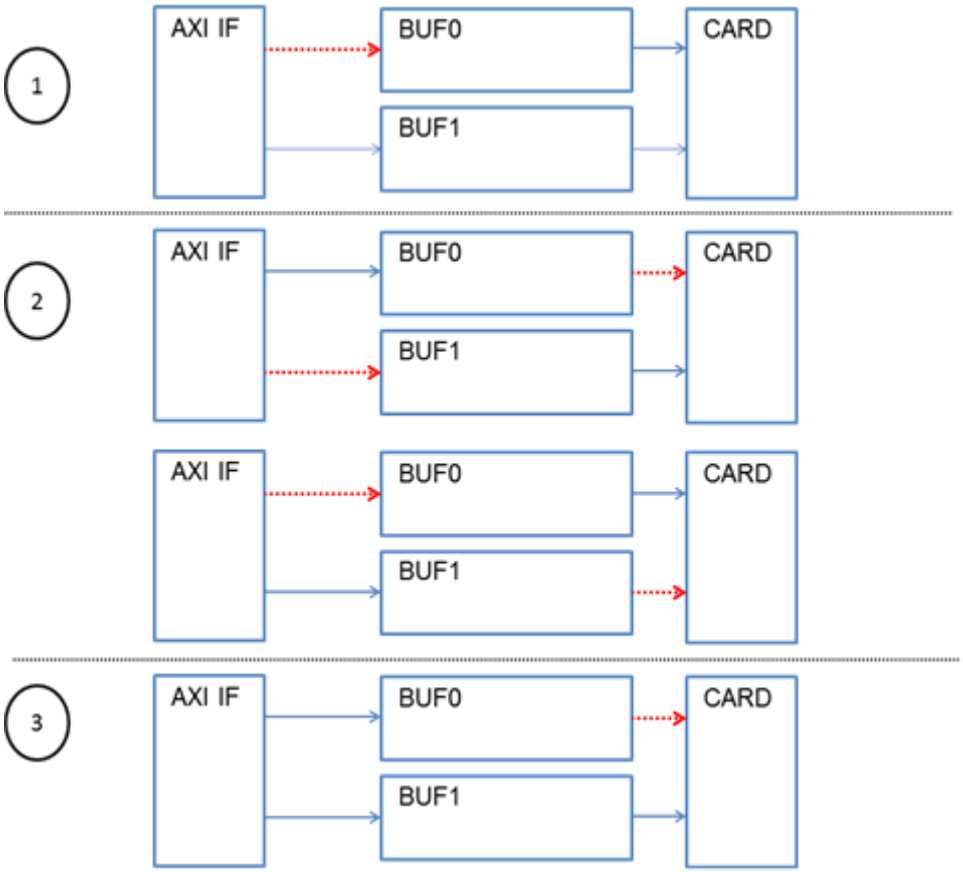


Рисунок 1.255 – Схема почередной работы буферов данных

На рисунке 1.256 представлена последовательность программирования регистров блока контроллера SDIO для записи одного или нескольких одиночных блоков данных в карту SD с использованием буферов данных 0 и 1.

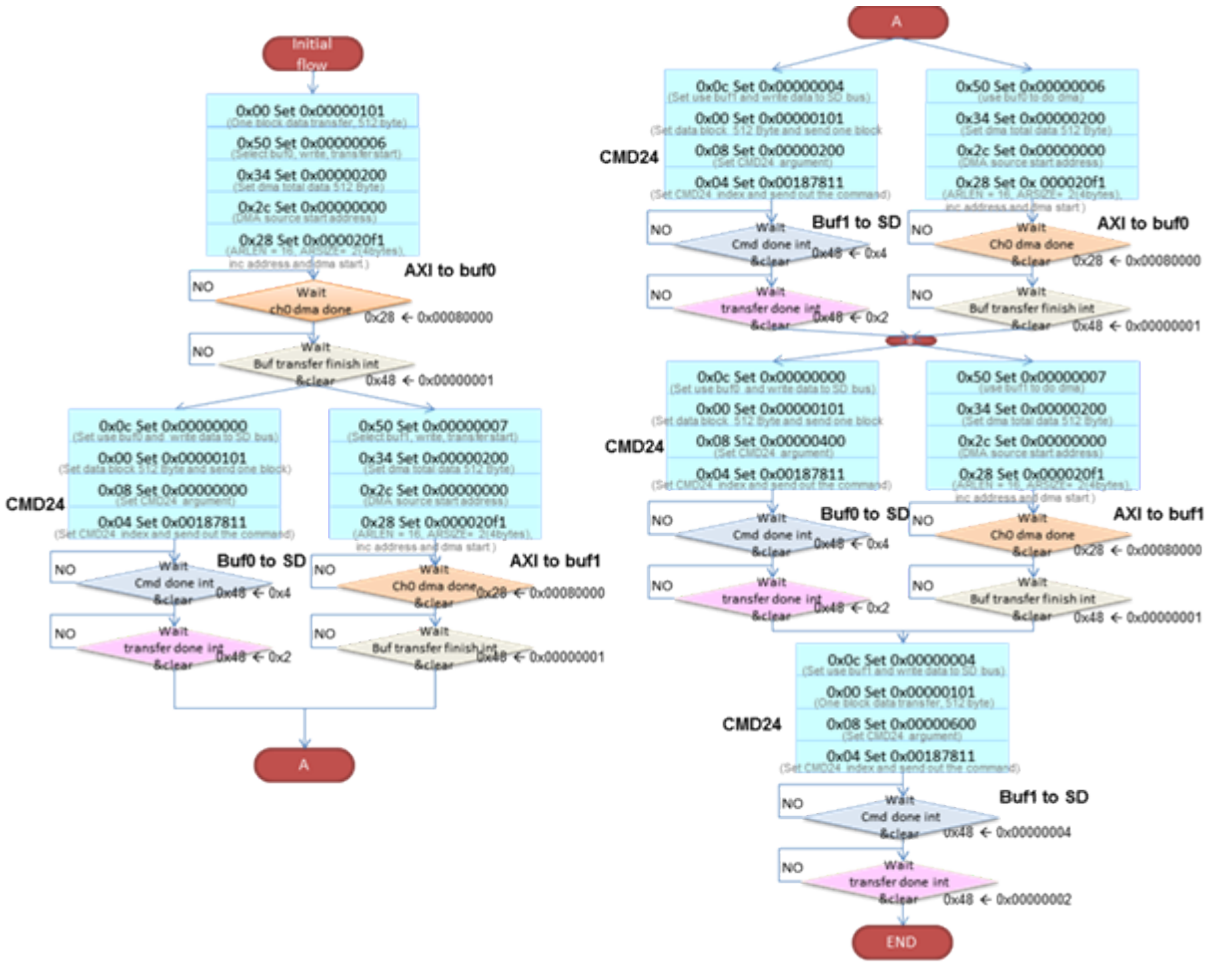


Рисунок 1.256 – Последовательность программирования регистров блока контроллера SDIO для записи одного или нескольких одиночных блоков данных в карту SD

1.3.12.2.6.4.2.4 Алгоритм чтения из SD карты нескольких блоков данных одной командой

Чтение нескольких блоков данных из SD карты может быть реализовано одной командой CMD18. Блоки данных будут последовательно считываться из карты до поступления команды STOP_TRANSMISSION (CMD12). Передача данных прекратиться сразу после передачи последнего бита команды CMD12. Контроллер автоматически пошлет команду CMD12 после считывания требуемого количества данных. Например, для считывания 8 блоков данных драйверу необходимо дождаться поступления 7 прерываний о завершении копирования данных в буфер (data_bound_int) и одного прерывания о завершении передачи (tran_done) после копирования последнего блока данных. Для установки прерывания о завершении копирования данных в буфер необходимо записать 0x8 в регистр SDR_BUF_TRAN_RESP_REG (смещение адреса 0x48). Для установки прерывания о завершении передачи необходимо записать 0x2 в регистр SDR_BUF_TRAN_RESP_REG (смещение адреса 0x48). На рисунке 1.257 представлена последовательность программирования регистров блока контроллера SDIO для чтения из SD карты нескольких блоков данных одной командой.

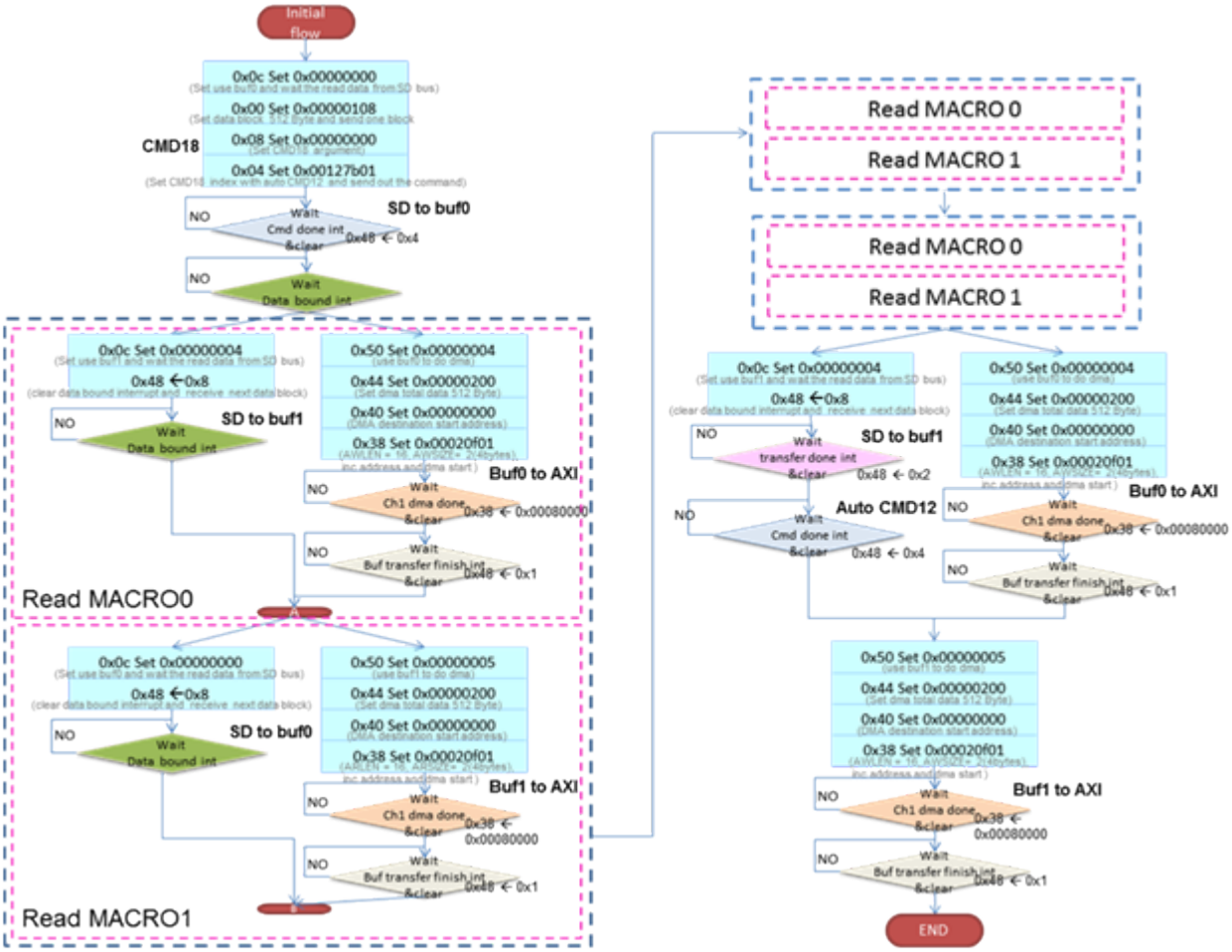


Рисунок 1.257 – Последовательность программирования регистров блока контроллера SDIO для чтения из SD карты нескольких блоков данных одной командой

1.3.12.2.6.4.2.5 Алгоритм записи в SD карту нескольких блоков данных одной командой

Запись нескольких блоков данных в SD карту может быть реализовано одной командой CMD25. Передача данных прекратиться сразу после передачи последнего бита команды CMD12. Контроллер автоматически пошлет команду CMD12 после записи требуемого количества данных. Программному обеспечению драйвера также необходимо следить за количеством переданных данных, чтобы корректно обрабатывать прерывания о завершении копирования данных в буфер и о завершении передачи данных. На рисунке 1.258 представлена последовательность программирования регистров блока контроллера SDIO для записи в SD карту нескольких блоков данных одной командой.

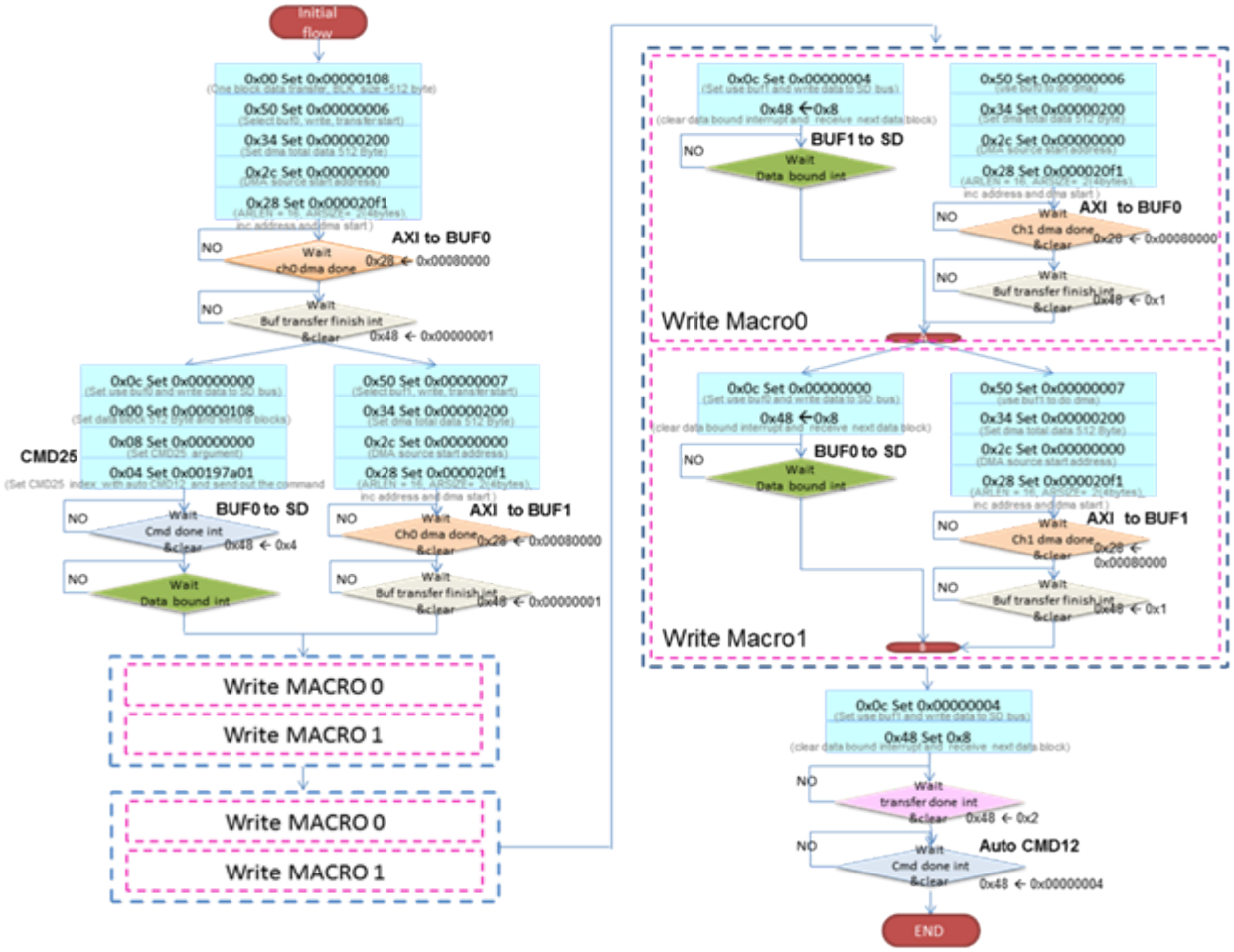


Рисунок 1.258 – Последовательность программирования регистров блока контроллера SDIO для записи в SD карту нескольких блоков данных одной командой

1.3.12.2.6.4.2.6 Алгоритм инициализации устройства SDIO

Алгоритм инициализации устройства SDIO отличается от алгоритма инициализации SD карты. В соответствии с требованиями стандарта SDIO подключение устройства SDIO к контроллеру, который не может работать с SDIO устройствами (только с SD картами), не должно приводить к отказу контроллера. Для этого были внесены изменения в процесс идентификации карты. Вместо команды ACMD41 для SDIO устройств была введена новая команда (IO_SEND_OP_COND, CMD5). После подачи питания или сброса все функции ввода-вывода у SDIO устройства должны быть отключены, и часть устройства, которая отвечает за операции ввода-вывода, должна бездействовать, за исключением ответа на команды CMD5 и CMD0 (бит CS должен быть в состоянии 0). На рисунке 1.259 представлена последовательность программирования регистров блока контроллера SDIO для инициализации устройства SDIO.

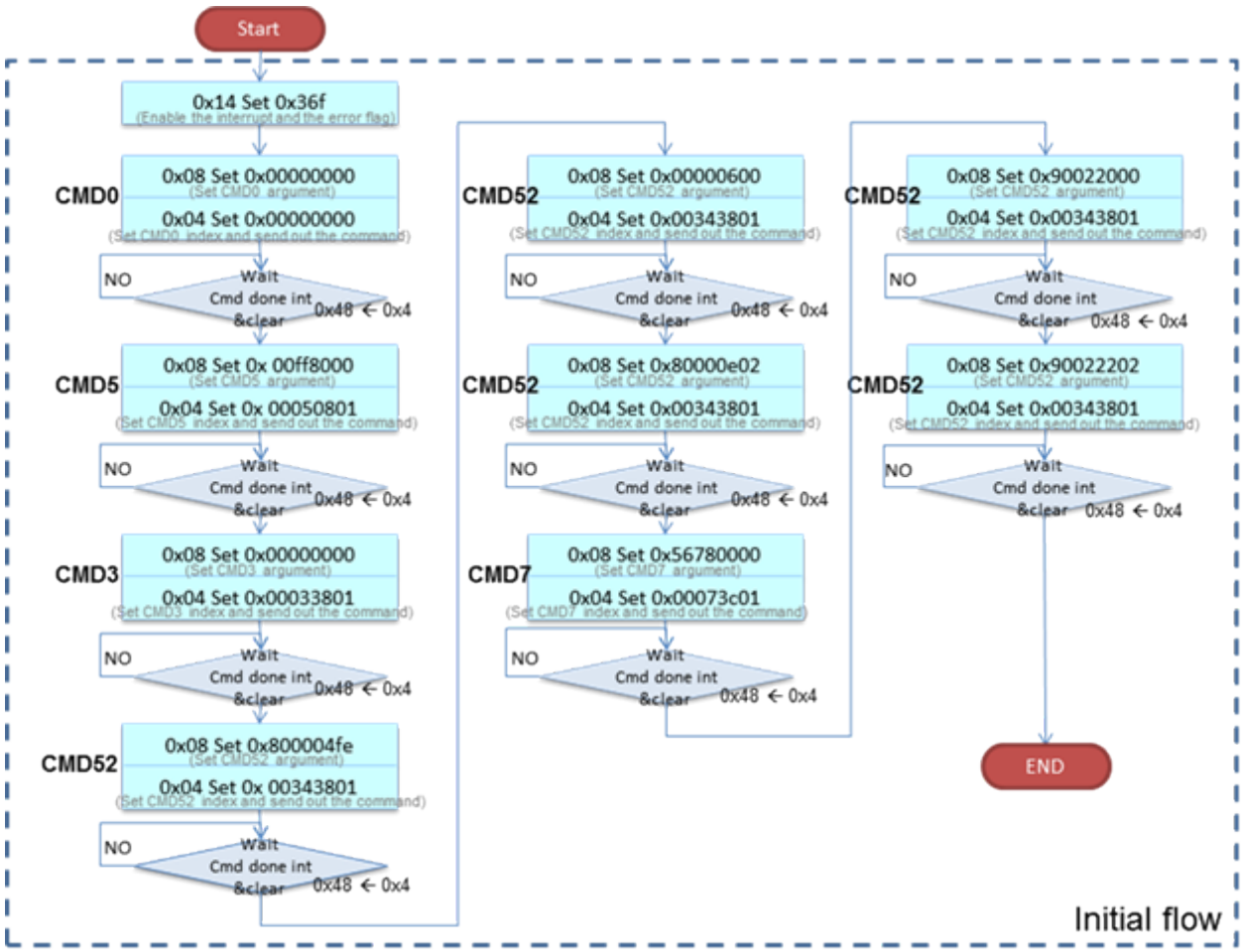


Рисунок 1.259 – Полследовательность программирования регистров блока контролера SDIO для инициализации устройства SDIO

1.3.12.2.6.4.2.7 Обмен одиночными блоками данных с SDIO устройством (IO_RW_EXTENDED)

С помощью команды CMD53 можно обмениваться одиночными блоками данных с SDIO устройством. На рисунке 1.260 и рисунке 1.261 представлена последовательность программирования регистров блока контроллера SDIO соответственно для чтения и записи одиночных блоков данных.

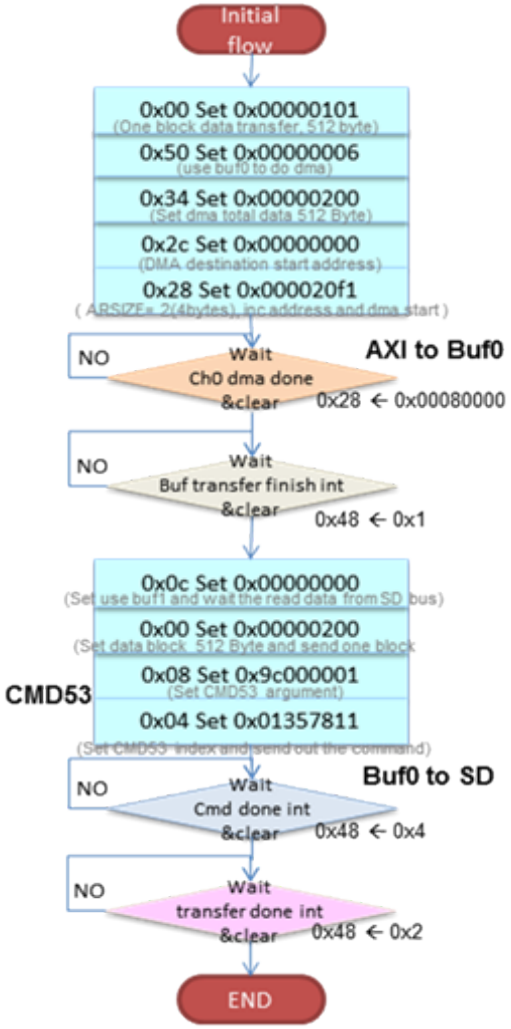


Рисунок 1.260 – Последовательность программирования регистров блока контроллера SDIO для чтения одиночных блоков данных

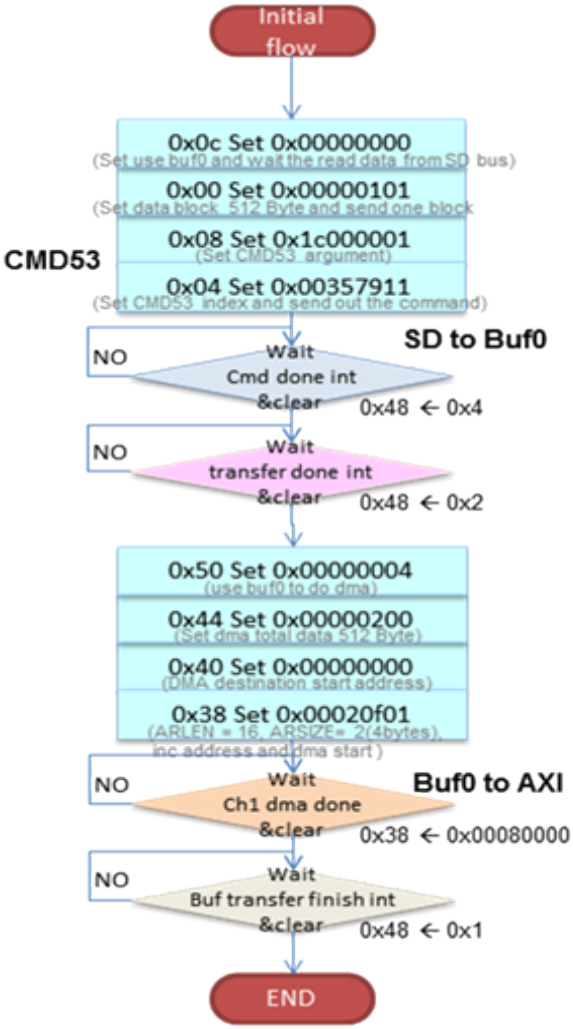


Рисунок 1.261 – Полседовательность программирования регистров блока контроллера SDIO для записи одиночных блоков данных

На рисунке 1.262 и рисунке 1.263 представлена последовательность программирования регистров блока контроллера SDIO соответственно для чтения и записи нескольких одиночных блоков данных подряд. При этом поочередно, по мере заполнения, используются буфера данных 0 и 1.

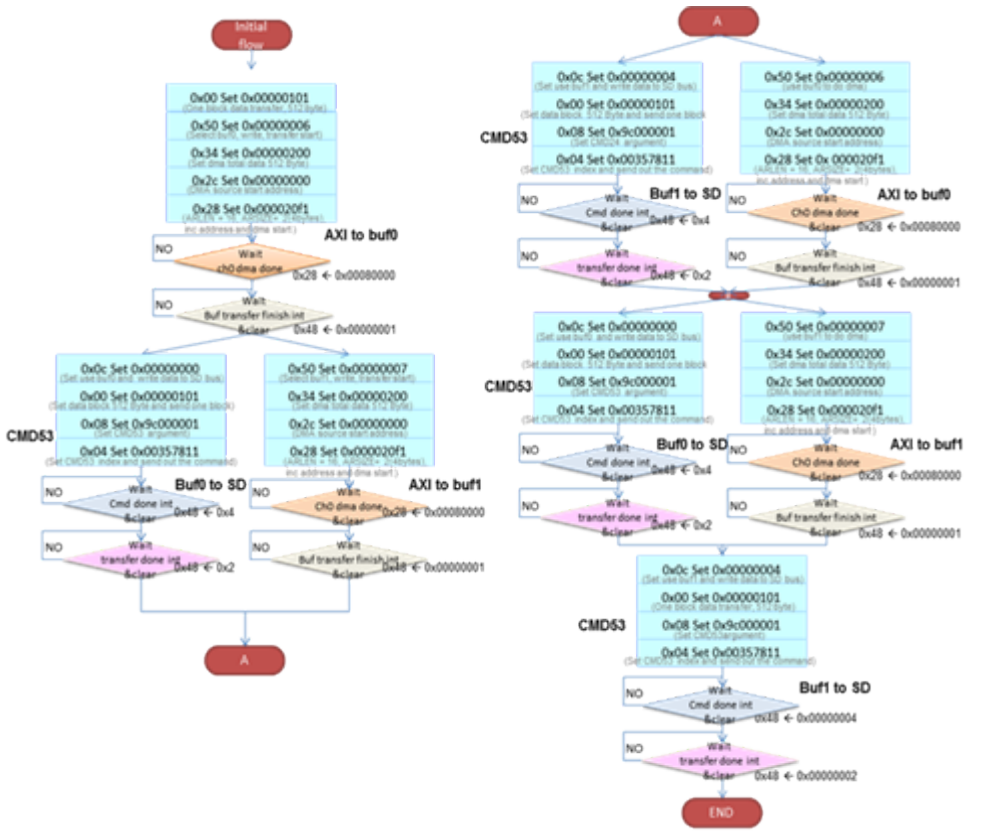


Рисунок 1.262 – Последовательность программирования регистров блока контроллера SDIO для чтения нескольких одиночных блоков данных подряд

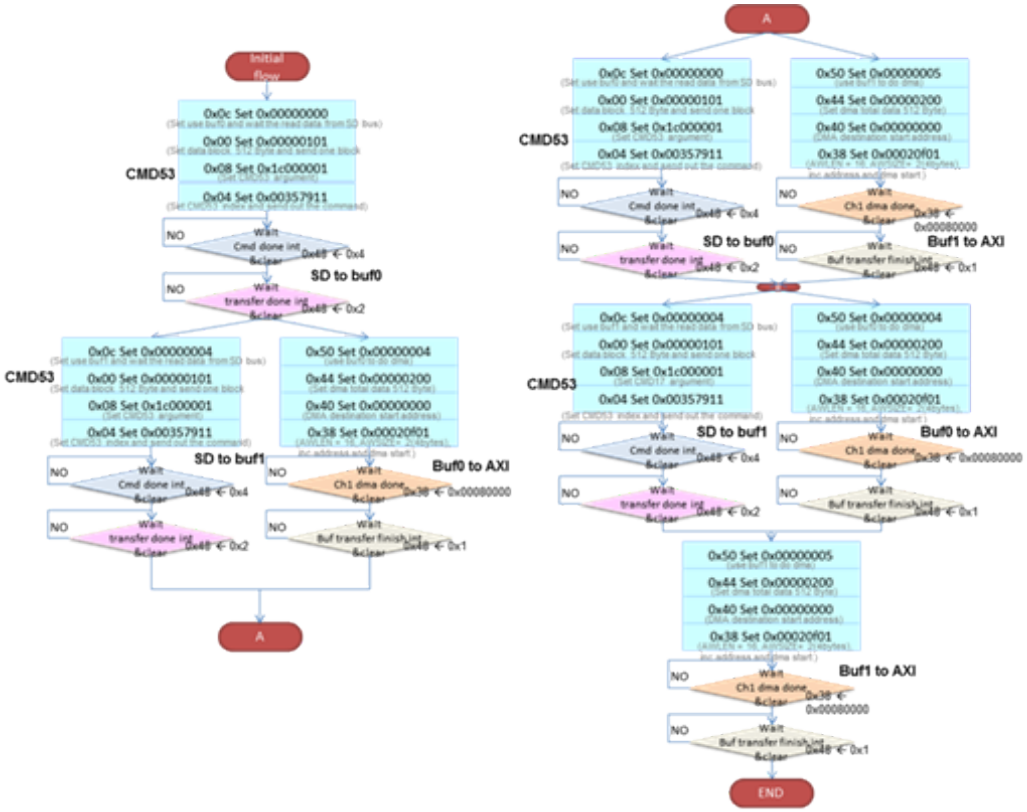


Рисунок 1.263 – Последовательность программирования регистров блока контроллера SDIO для записи нескольких одиночных блоков данных подряд

1.3.12.2.6.4.2.8 Обмен несколькими блоками данных с использованием одной команды (SDIO IO_RW_EXTENDED)

Запись нескольких блоков данных в SD карту может быть реализовано одной командой CMD53. Программному обеспечению драйвера также необходимо следить за количеством переданных данных, чтобы корректно обрабатывать прерывания о завершении копирования данных в буфер и о завершении передачи данных. На рисунках 1.260 и 1.261 представлена последовательность программирования регистров блока контроллера SDIO соответственно для чтения и записи в SDIO устройство нескольких блоков данных одной командой.

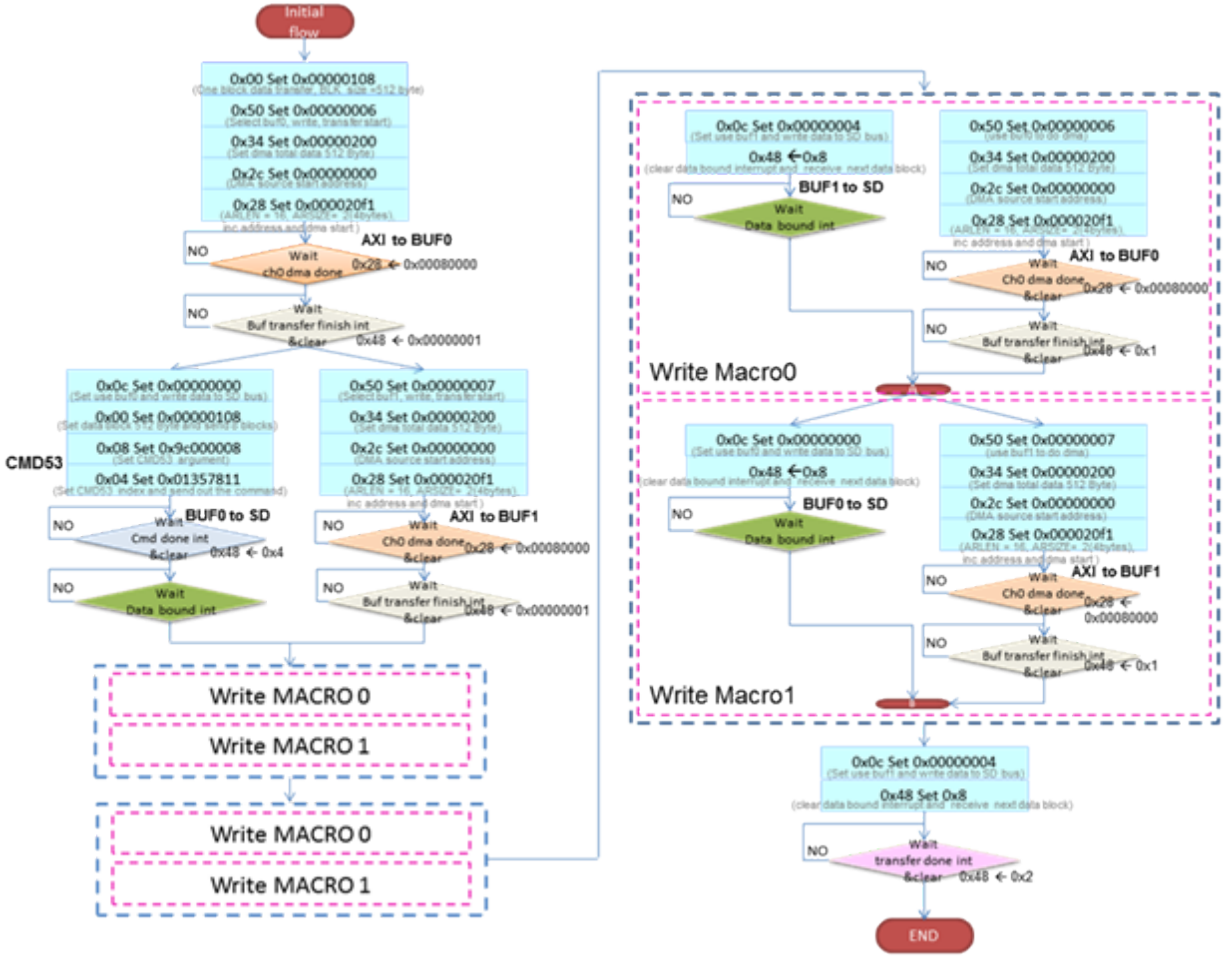


Рисунок 1.260 – Последовательность програмирования регистров контролера SDIO для чтения в SDIO устройство нескольких блоков данных одной командой

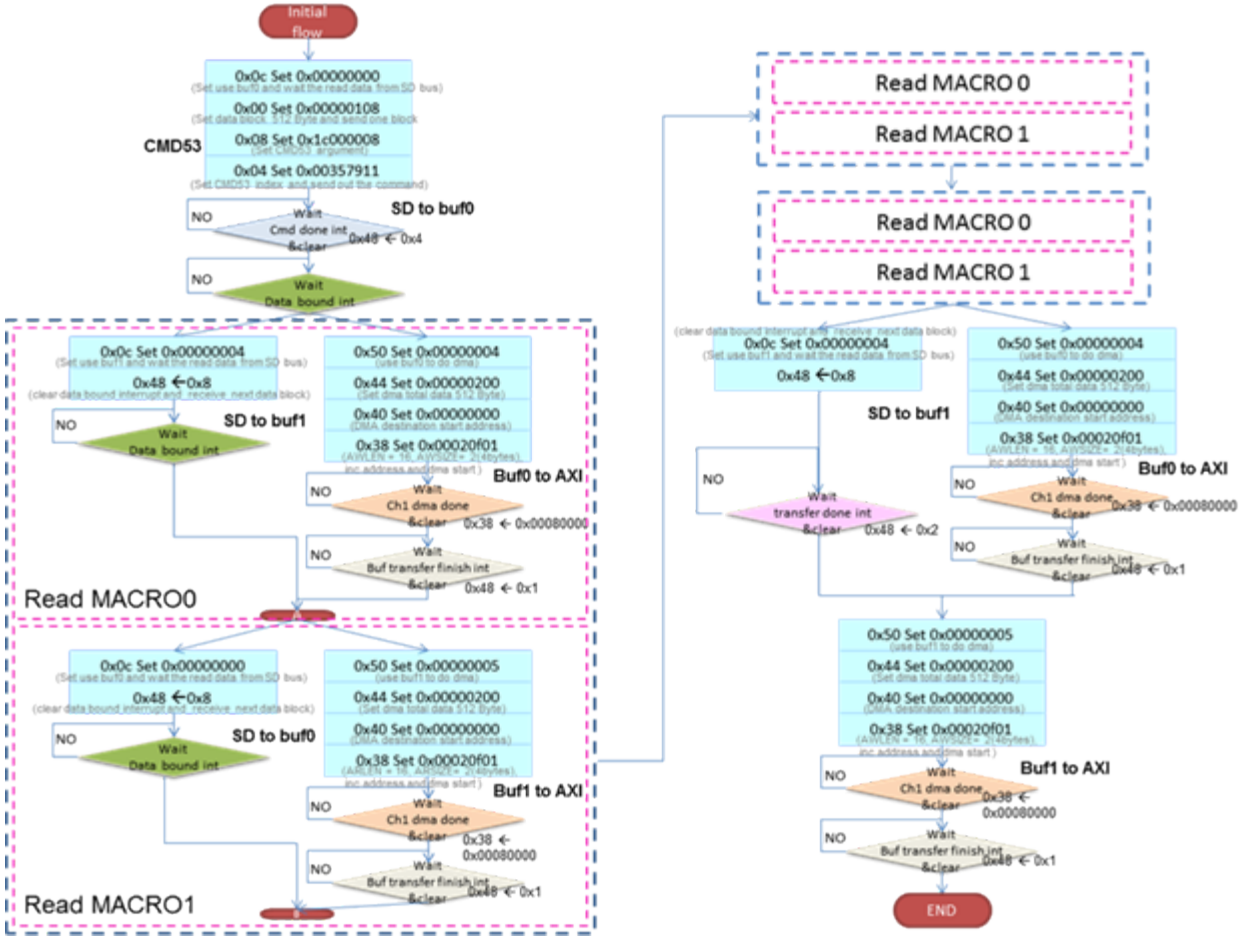


Рисунок 1.261 - Последовательность програмирования регистров контролера SDIO для записи в SDIO устройство нескольких блоков данных одной командой

1.3.12.2.6.4.2.9 Обмен с SDIO устройством несколькими блоками данных одной командой – остановка обмена данными

Для остановки обмена несколькими блоками данных с SD картой используется команда CMD12, которая может быть послана контроллером в любой момент. Для остановки обмена с SDIO устройством вместо команды CMD12 необходимо использовать запись в биты ASx регистра CCCR. Обычно такой способ используют для остановки «бесконечного» обмена данными (с числом блоков block count=0 в теле команды). Если заранее известно, сколько блоков данных нужно передать, рекомендуется указывать требуемое количество блоков в теле команды вместо «бесконечного» обмена данными с остановкой в нужный момент времени. Передача данных начинается во время исполнения команды CMD53, внутренний автомат состояний находится в состоянии пересылки данных, и для выхода из этого состояния необходимо послать команду CMD52. Таким образом, хост-контроллер должен послать команду CMD52 для остановки обмена данными.

На рисунках 1.262 и 1.263 представлена последовательность программирования регистров блока контроллера SDIO соответственно для остановки записи и чтения из SDIO устройства нескольких блоков данных.

Для остановки записи данных в SDIO устройство хост-контроллер может послать команду на запись в регистр CCCR в произвольный момент времени в промежутках между операциями с блоками данных. При этом передача последнего блока данных (включая получение контрольной суммы от устройства SDIO) должна быть уже завершена. Для выполнения этого требования

необходимо ,чтобы последний бит команды отмены обмена данными был передан как минимум за два такта до получения контроллером последнего бита ответной контрольной суммы последнего блока данных. Контроллер не должен посылать команду отмены в середине передачи блока данных. После того, как команда отмена передана SDIO устройству, SDIO устройство активирует сигнал занятости (линия DAT[0] переходит в состояние логического 0) и удерживает его до тех пор, пока не закончит обработку последнего блока принятых данных. Во время активности сигнала занятости хост-контроллер может освободить шину, запрограммировав бит BR регистра CCCR.

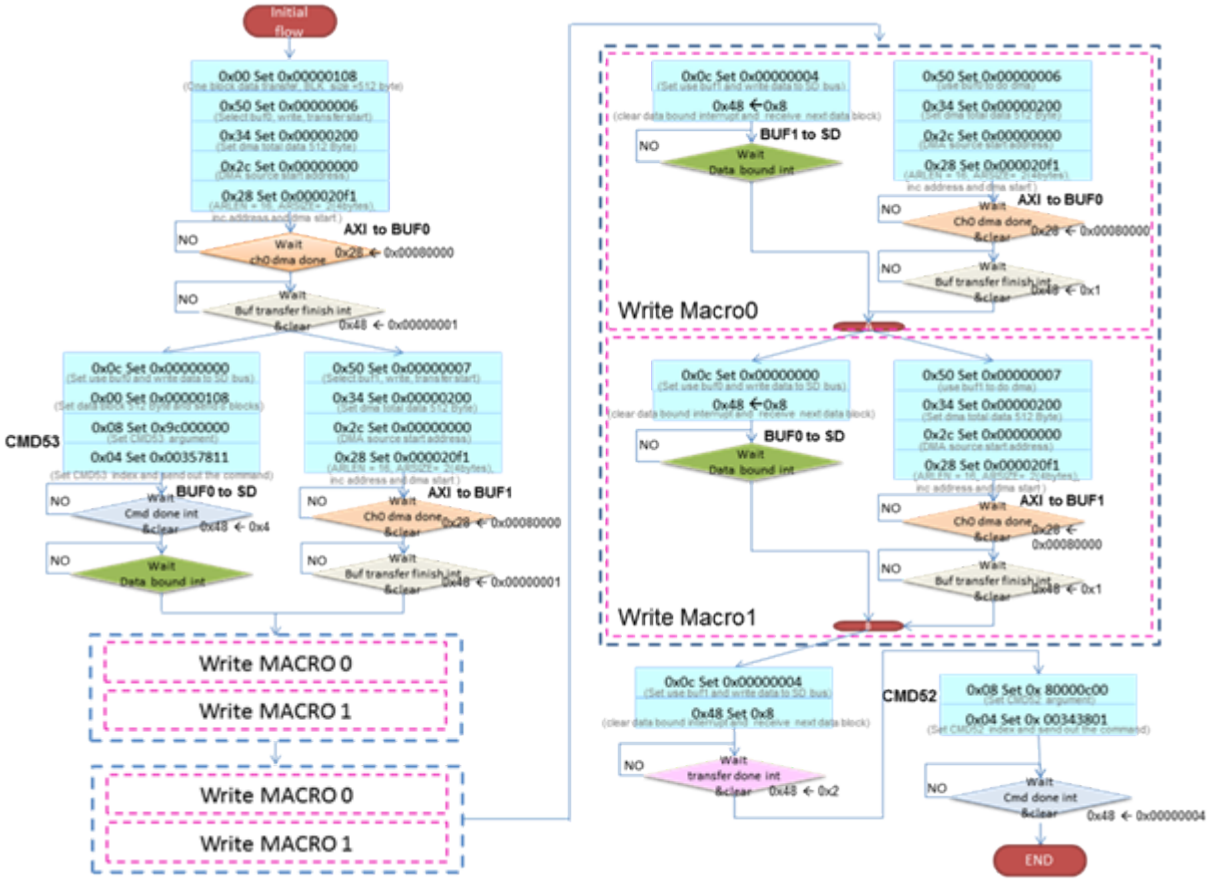


Рисунок 1.262 - Последовательность программирования регистров блока контроллера SDIO для остановки записи из SDIO устройства нескольких блоков данных

Для остановки чтения данных из SDIO устройства хост-контроллер может послать команду на запись в регистр CCCR в произвольный момент времени. Передача данных остановится через два такта после передачи последнего бита команды отмены даже в том случае, если SDIO устройство уже начало передачу блока данных. При этом данные из прерванного блока данных будет утеряны.

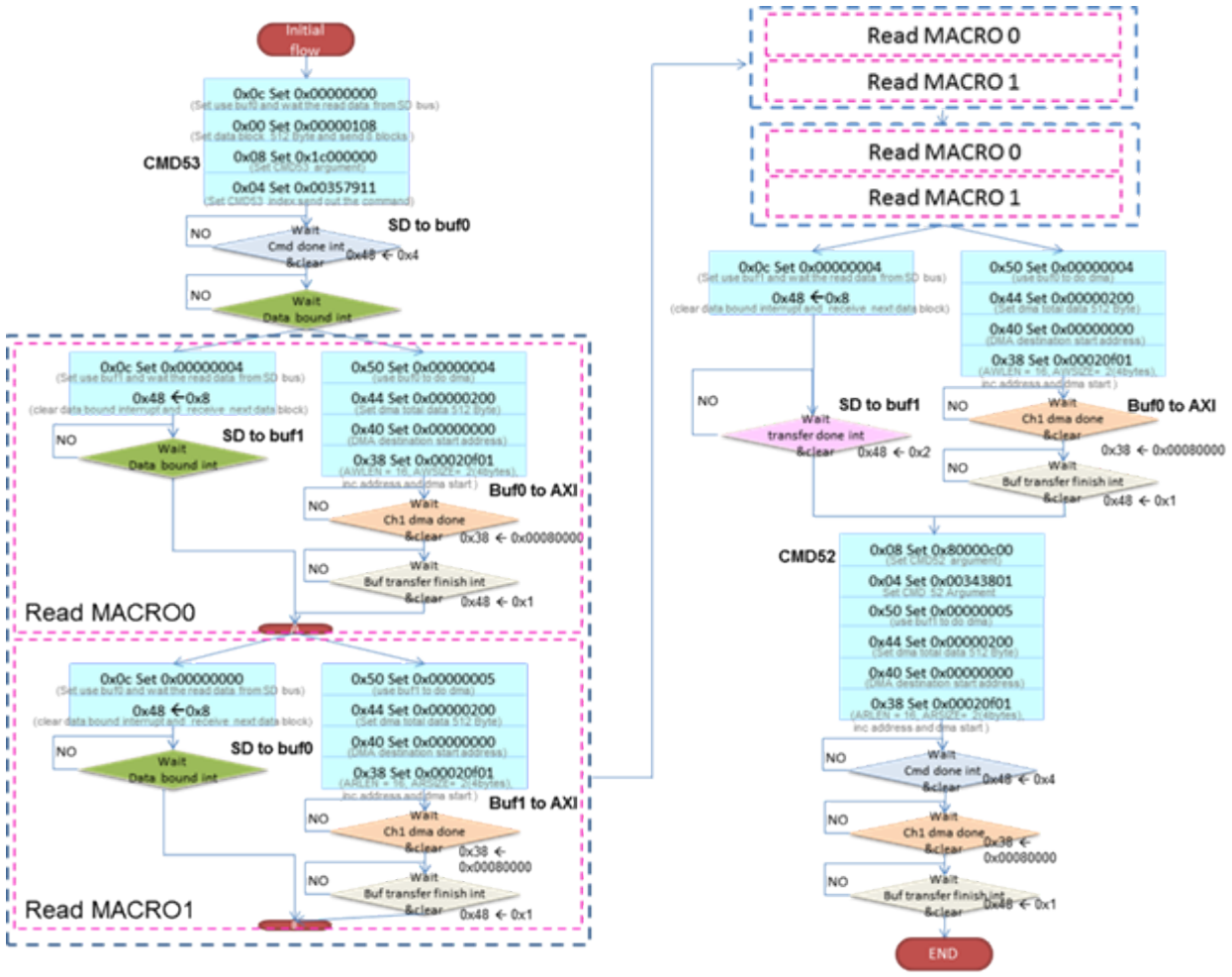


Рисунок 1.263 - Последовательность программирования регистров блока контроллера SDIO остановки чтения из SDIO устройства нескольких блоков данных.

1.3.12.2.7 Контроллер USB

1.3.12.2.7.1 Введение

MUSBMHDRC - это многофункциональный блок, который совмещает в одном ядре:

- контроллер периферийных устройств USB с высокой (high-speed)/средней (full-speed) скоростью соединения;
- контроллер ‘Dual-role’ USB (с возможностью работы как в режиме ведущего, так и ведомого устройства) для связи ‘On-The-Go’ (OTG) типа точка-точка с другим устройством USB с высокой, средней или низкой (low-speed) скоростью соединения; и
- (при подключении к хабу) хост-контроллер для системы USB с множеством ведомых устройств.

- что позволяет устройству, в котором используется ядро MUSBMHDRC, при необходимости переключаться между этими режимами.

Ядро совместимо со стандартом USB 2.0 для высокой и средней скорости соединения, а также с расширением On-The-Go к спецификации USB 2.0. Спецификация On-The-Go для USB была введена для обеспечения простого соединения портативных потребительских устройств, таких как мобильные телефоны, карманные компьютеры, цифровые фотоаппараты и MP3 плееры. Устройства, работающие только в режиме ведомого, могут инициировать траффик USB

по протоколу установки соединения (Session Request Protocol, SRP) тогда как устройства двойного назначения поддерживают как SRP, так и протокол согласования ведущего устройства (Host Negotiation Protocol, HNP) и может по необходимости выступать в роли, как ведущего, так и ведомого устройства. MUSBMHDRC также поддерживает разделенные транзакции, что в свою очередь позволяет использовать устройства со средней и низкой пропускной способностью с хабом USB 2.0. Ядро также содержит поддержку отключения питания для не используемых портативных устройств.

MUSBMHDRC конфигурируется пользователем и кроме к оконечной точки 0 может включать до 15 оконечных точек ‘передатчика’ и/или до 15 оконечных точек ‘приемника’. (Использование этих оконечных точек для входных транзакций и выходных транзакций зависит от того, используется ли MUSBMHDRC как ведомое или как ведущее устройство. При использовании в качестве ведомого устройства входные транзакции обрабатываются через оконечные точки передатчика, а выходные транзакции обрабатываются через оконечные точки приемника. При использовании в качестве ведущего устройства входные транзакции обрабатываются через оконечные точки приемника, а выходные транзакции обрабатываются через оконечные точки передатчика.) Эти дополнительные оконечные точки могут конфигурироваться в программном обеспечении индивидуально, чтобы выполнять поточные (Bulk) (что также позволяет выполнять пересылки по прерыванию (Interrupt)), изохронные (Isochronous) или управляющие (Control) пересылки. Более того, в процессе работы оконечные точки также могут быть распределены по разным функциям целевого устройства– это максимизирует количество устройств, которые могут поддерживаться одновременно.

Каждая оконечная точка требует ассоциированного с ним FIFO буфера. MUSBMHDRC имеет интерфейс RAM для подключения к одному блоку синхронного однопортового блока памяти (RAM), который используется для FIFO буферов оконечных точек. (Сам блок RAM должен добавляться пользователем.)

FIFO буфер для оконечной точки 0 должен иметь глубину 64 байта и буферировать 1 пакет. Интерфейс RAM может конфигурироваться с учетом FIFO буферов других оконечных точек; буфер может иметь размер от 8 до 8192 байт и буферировать 1 или 2 пакета. С каждой оконечной точкой может ассоциироваться отдельный FIFO буфер; или же оконечная точка передатчика и оконечная точка приемника с одним и тем же номером могут быть сконфигурированы для использования одного FIFO буфера, например, с целью уменьшения размера необходимого блока RAM – при условии, что эти оконечные точки никогда не будут активными одновременно.

MUSBMHDRC снабжен 32-битовым синхронным интерфейсом CPU, предназначенным для подключения к шине AMBA AHB1. Интерфейс поддерживает работу с шиной AHB в широком диапазоне частот шины. Также поддерживаются многоуровневые (Multi-layer) операции на шине AHB. MUSBMHDRC легко подключается к ряду других стандартных шин посредством добавления соответствующей оболочки / моста.

Также поддерживается доступ DMA к FIFO буферам оконечных точек.

The MUSBMHDRC предоставляет UTMI+ Level 3-совместимый интерфейс для подключения к соответствующему USB трансиверу с высокой и средней скоростью соединения. Для ULPI-совместимых PHY имеется дополнительный ULPI Link Wrapper (описан в документации musbhdrc_ulpi_an.pdf, в директории musbmhdrc/docs). Предоставляется также альтернативный интерфейс, который позволяет использовать PHY USB 1.1 средней пропускной способности совместно с ядром, но только для транзакций со средней и низкой скоростью соединения.

MUSBMHDRC обеспечивает все кодирование, декодирование, проверку и повторный запрос, необходимые для передачи и приема пакетов USB – прерывание CPU генерируется только после успешной пересылки данных оконечной точки.

При работе в качестве ведущего устройства MUSBMHDRC дополнительно поддерживает счетчик фреймов и автоматически распределяет пересылки - SOF, изохронные, по прерыванию, и поточные. Имеется также поддержка для протокола установки соединения (Session Request Protocol) и протокола согласования ведущего устройства (Host Negotiation Protocol), используемых для связи точка-точка, подробное описание которой приводится в расширении USB On-The-Go к спецификации USB 2.0.

MUSBMHDRC предлагает ряд тестовых режимов – в первую очередь четыре тестовых режима для высокой скорости соединения, описанной в спецификации USB 2.0. Имеются также опции, которые позволяют принудительно перевести MUSBMHDRC в режим со средней скоростью соединения, с высокой скоростью соединения или в режим ведущего устройства. Последний может быть полезен для отладки проблем PHY в аппаратных средствах.

Скрипты графического интерфейса пользователя служат для конфигурирования ядра под требования пользователя. Используемый скрипт зависит от выбранного интерфейса CPU. Примечание: В данный момент ядро доступно только в Verilog.

Данную спецификацию следует читать совместно со спецификацией USB On-The-Go, в которой также приводятся требования к питанию, уровням напряжения, коннекторам и т. д.

1.3.12.2.7.2 Описание функций

1.3.12.2.7.2.1 Режимы работы

MUSBMHDRC имеет два режима работы – режим ведомого устройства и режим ведущего устройства.

В режиме ведомого устройства MUSBMHDRC кодирует, декодирует, проверяет и направляет все пакеты USB, отправляемые и принимаемые. Входные транзакции обрабатываются с помощью FIFO буферов передатчика, выходные транзакции обрабатываются с помощью FIFO буферов приемника. Поддерживаются управляющие, поточные, изохронные пересылки и пересылки по прерыванию.

В режиме ведущего устройства работа MUSBMHDRC зависит от того, подключен ли он к другому устройству USB соединением типа точка-точка, или к хабу. При подключении к другому устройству USB MUSBMHDRC предлагает ряд возможностей, необходимых для того, чтобы выполнять роль ведущего устройства в соединении типа точка-точка с этим устройством USB. При подключении к хабу обеспечиваются функции, необходимые для того, чтобы выполнять роль ведущего устройства для нескольких ведомых устройств, которые поддерживаются одновременно.

При работе в режиме ведущего устройства и при использовании для соединения типа точка-точка с одним устройством USB (с высокой, средней или низкой скоростью соединения) MUSBMHDRC способен поддерживать управляющие, поточные, изохронные транзакции и транзакции по прерыванию. Входные транзакции обрабатываются с помощью FIFO буферов приемника, выходные транзакции обрабатываются с помощью FIFO буферов передатчика. Кроме кодирования, декодирования и проверки принимаемых и отправляемых пакетов USB, MUSBMHDRC также автоматически распределяет изохронные оконечные точки и оконечные точки по прерыванию, чтобы выполнять одну транзакцию каждые n фреймов /микрофреймов

(или до трех транзакций, если выбрана опция высокой пропускной способности), где n – это запрограммированный для оконечной точки интервал опроса. Оставшаяся полоса пропускания шины равномерно распределяется между управляющими и поточными оконечными точками (см. раздел 8.5.4, Переупорядочивание транзакций).

При подключении к хабу MUSBMHDRC также обеспечивает все перечисленные выше функции, но дополнительно нуждается в программировании следующих параметров:

- Адрес функции целевого устройства.
- Быстродействие целевого устройства (чтобы можно было выполнить соответствующее преобразование скорости).
- Если целевое устройство обеспечивает среднюю или низкую скорость соединения, но подключено через хаб с высокой скоростью соединения, то оконечная точка нуждается в дополнительном программировании функционального адреса и номера порта хаба.
-

При подключении в роли устройства ‘A’ (источник питания и ведущее устройство по умолчанию) от MUSBMHDRC может потребоваться питать VBus напряжением 5В; в роли устройства ‘B’ (ведомое устройство по умолчанию) MUSBMHDRC должен иметь возможность активизировать устройство ‘A’ установкой напряжения 2В на VBus. Выходные сигналы MUSBMHDRC указывают, когда требуются эти опции питания.

Начальное включение режима ведущего устройства или ведомого устройства зависит от того, используется ли MUSBMHDRC в качестве устройства ‘A’ или устройства ‘B’, что в свою очередь зависит от уровня входного сигнала IDDIG. Когда MUSBMHDRC работает в качестве устройства ‘A’, то при начальной конфигурации он переводится в режим ведущего устройства. При работе в качестве устройства ‘B’ MUSBMHDRC при начальной конфигурации переводится в режим ведомого устройства. Однако в регистре DevCtl имеется бит “Host Req”, посредством которого CPU может послать запрос на переключение устройства ‘B’ в режим ведущего устройства, в момент отсутствия активности на шине USB.

Входной сигнал IDDIG отражает состояние контакта ID разъема мини-AB устройства; низкий уровень IDDIG указывает на подключение ‘A’, то есть, работу в качестве устройства ‘A’, а высокий уровень указывает на подключение ‘B’ и работу в качестве устройства ‘B’.

Информация о том, работает ли в качестве устройства ‘A’ или устройства ‘B’, а также информация о скорости соединения также записывается в регистре DevCtl, вместе с информацией об уровне VBus относительно порогов высокого и низкого напряжения, используемых для указания начала и окончания сеанса.

Процедуры для запроса соединения и для распределения ролей ведущий/ведомый между устройствами на обеих концах соединения описаны в разделах 14 и 15 соответственно. Все пересылки соответствуют стандартным протоколам передачи данных USB.

1.3.12.2.7.2.2 Блок-схема

На приведенной на рисунке 1.264 блок-схеме показаны главные функциональные блоки, из которых состоит MUSBMHDRC. (Блок-схема любого моста, предназначенного для работы с ядром, приводится в отдельной спецификации для соответствующего моста, которая находится в директории musbmhdrc/docs).

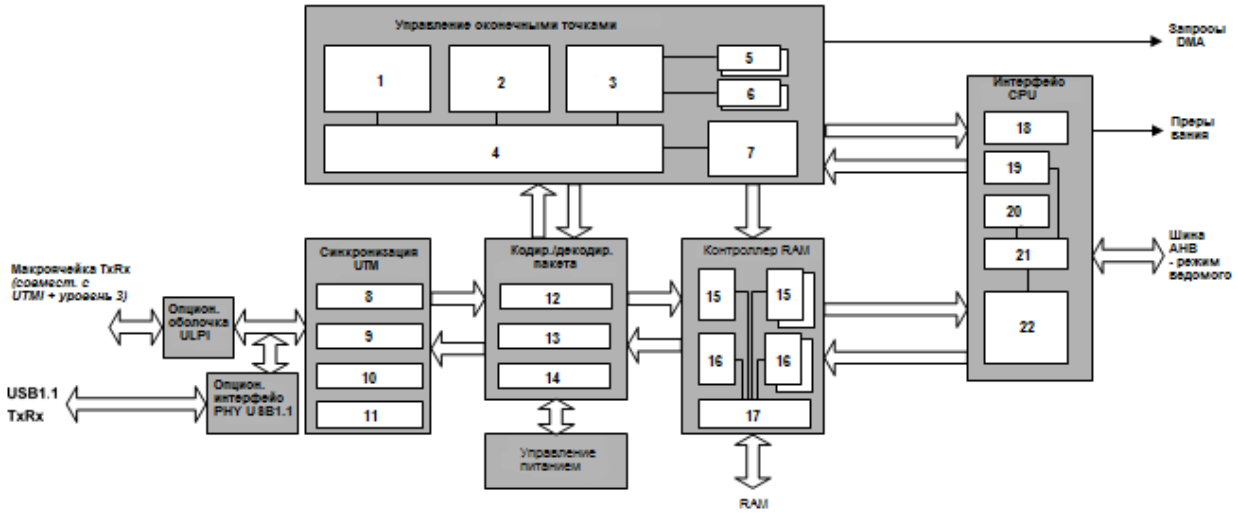


Рисунок 1.264 - Блок-схема MUSBMHDRC

1. Управление оконечной точкой 0 – ведущее устройство
2. Управление оконечной точкой 0 – ведомое устройство
3. Управление оконечными точками 1 – 15
4. Общее управление оконечными точками
5. Передатчик
6. Приемник
7. Планировщик транзакций ведущего устройства
8. Синхронизация данных
9. Согласование ведущего устройства
10. Протокол согласования ведущего устройства (HNP) и протокол установки соединения (SRP)
11. Таймеры
12. Кодирование пакета
13. Декодирование пакета
14. Генерация/проверка контрольной суммы
15. Буфер приемника
16. Буфер передатчика
17. Контроль цикла
18. Контроль прерываний
19. Декодер контроля прерываний
20. Общие регистры
21. Контроль цикла
22. Декодер FIFO

1.3.12.2.7.2.3 Синхронизация UT M

Блок синхронизации UTM предназначен для ресинхронизации между частотным доменом 60МГц блока передатчика и системной тактовой частотой контроллера, которая подается на остальное ядро, в том числе на интерфейс CPU. Это позволяет остальным частям MUSBMHDRC работать на частоте шины CPU без дополнительной синхронизации. Блок также выполняет обмен сигналами для определения возможности установки соединения на высокой скорости и

поддерживает протоколы HNP и SRP при соединении точка-точка с другим устройством USB OTG.

Используя восьмибитовый интерфейс, блок сначала преобразует данные в 16-битовый формат – для этого ядро должно работать с тактовой частотой не менее 30MHz. В реальности минимальная частота системного тактового сигнала, необходимая для правильной передачи данных через границу домена, зависит от технологии реализации

1.3.12.2.7.2.4 Кодирование / декодирование пакета

Блок кодирования/декодирования пакета генерирует заголовки пакетов, предназначенных для пересылки, и декодирует заголовки принятых пакетов. Он также генерирует CRC для пакетов, предназначенных для пересылки, и проверяет Контрольную сумму (CRC) принятых пакетов.

1.3.12.2.7.2.5 Контроллеры оконечных точек

Используются два управляющих конечных автомата: один для управления пересылками через оконечную точку 0, и один для поточных, изохронных, управляющих транзакций и транзакций по прерыванию через оконечные точки от 1 до 15.

1.3.12.2.7.2.6 Интерфейс CPU

Интерфейс CPU обеспечивает доступ к управляющим регистрам, регистрам состояния и FIFO буферу. Он также генерирует прерывания для CPU при успешном приеме или передаче пакетов, а также когда ядро переходит в спящий режим или выходит из спящего режима.

Интерфейс, обеспечиваемый MUSBMHDRC, представляет собой 32-битовый синхронный интерфейс, соответствующий спецификации интерфейсов для шины AMBA AHB. Интерфейс для других стандартов шины может быть получен добавлением к ядру соответствующей оболочки.

1.3.12.2.7.2.7 Контроллер RAM

Контроллер RAM обеспечивает интерфейс для одного блока синхронного однопортового блока памяти, используемого для буферизации пакетов между CPU и USB. Контроллер берет указатели FIFO буферов контроллеров оконечных точек, преобразует в указатели адреса внутри блока RAM и генерирует управляющие сигналы доступа к RAM.

1.3.12.2.7.2.8 Поддержка контроллера DMA

При необходимости MUSBMHDRC может включать контроллер многоканального DMA, предназначенный для эффективной загрузки/выгрузки FIFO буферов оконечных точек. Этот контроллер DMA может конфигурироваться для поддержки до 8 каналов.

Контроллер DMA имеет собственный блок управляющих регистров и собственный контроллер прерываний. Он поддерживает два режима работы, и каждый канал может быть независимо запрограммирован на любой режим.

В качестве альтернативы контроллер MUSBMHDRC может быть объединен с внешним контроллером DMA с целью эффективной загрузки/выгрузки FIFO буферов оконечных точек. MUSBMHDRC вырабатывает сигналы запроса DMA для каждой оконечной точки.

1.3.12.2.7.3 Описание регистров

1.3.12.2.7.3.1 РЕГИСТРОВАЯ ПАМЯТЬ MUS B M H D RC

Регистровая память MUSBMHDRC разбита на следующие разделы:

- Общие регистры USB (00h–0Fh) – Эти регистры хранят управляющую информацию и информацию состояния всего ядра.
- Индексные регистры управления/состояния (CSR) оконечных точек (10h–1Fh) – Эти регистры хранят управляющую информацию и информацию состояния выбранной в данный момент оконечной точки. Регистры включаются в этот раздел в зависимости от того, как каком режиме работает ядро, ведомого устройства (DevCtl.D2=0) или ведущего устройства (DevCtl.D2=1), а также значения регистра Index.
- FIFO (20h–5Fh) – Этот диапазон адресов предоставляет доступ к FIFO оконечных точек.
- Дополнительные управляющие регистры и регистры конфигурации (60h–7Fh) – Эти регистры хранят дополнительную информацию управления и состояния устройства.
- Управляющие регистры целевой оконечной точки (80h–FFh) – Если в конфигурации GUI разрешен режим с множеством ведомых устройств, эти регистры хранят информацию о целевой функции и адресе хаба для каждой из целевых точек. Доступ к этим регистрам возможен только при разрешении режима с множеством ведомых устройств.
- Неиндексные регистры управления/состояния оконечных точек (100h и выше) – Регистры с адресами 10h–1Fh доступны независимо от состояния регистра Index. 100h–10Fh – это регистры оконечной точки 0, 110h–11Fh - регистры оконечной точки 1; 120h–12Fh - регистры оконечной точки 2, и так далее.
- Управляющие регистры DMA (200h и выше) – Эти регистры появляются только в том случае, когда синтезируемый модуль включает опциональный контроллер DMA (см. раздел 17).
- Регистры RqPktCount (304h – 33Ch) – Эти регистры используются в режиме ведущего устройства совместно с AutoReq (см. раздел 8.5.2).
- Регистры DPktBufDis (340h – 343h) – Эти регистры обеспечивают пользователю прямое управление запретом двойной буферизации пакетов.
- Регистры C_T_UCH (344h – 345h) – Эти регистры устанавливают таймер времени ожидания ЛЧМ-сигнала (Chirp).
- Регистры C_T_HSRTN (346h – 347h) – Эти регистры устанавливают задержку относительно окончания выхода из спящего режима при высокой скорости соединения, разрешая нормальный режим работы UTM.

Схема распределения регистровой памяти представлена на рисунке 1.265.

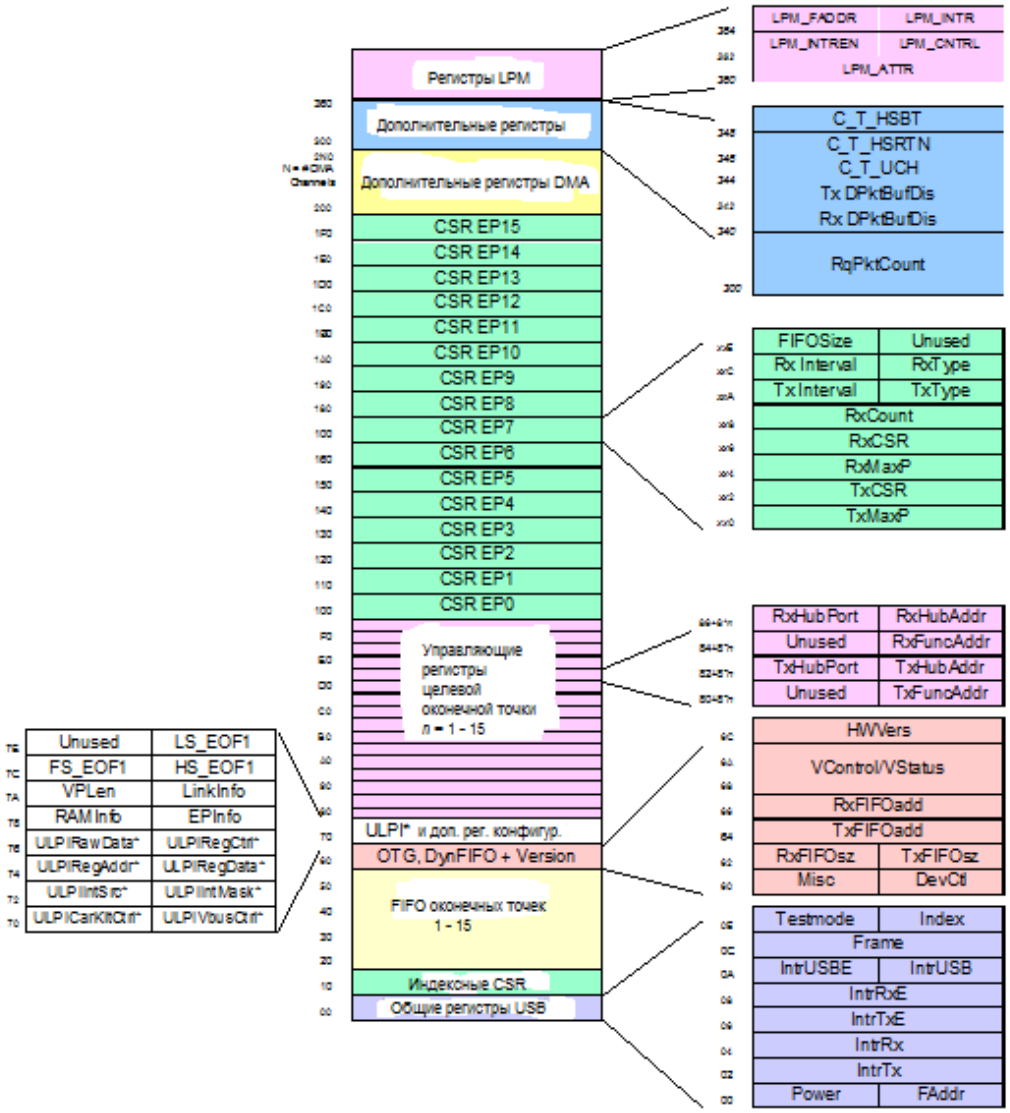


Рисунок 1.265 - Схема распределения регистровой памяти

Примечание: Дополнительные регистры, ассоциированные с любым мостом, предоставленным для использования с ядром MUSBMHDRC, или любые изменения следующих регистров, которые являются результатом использования этого моста, описаны в отдельной спецификации на мост, которая находится в директории musbmhdrc/docs.

Таблица 1.719 – Регистровая память MUSBMHDRC

АДР.	НАИМ.	ОПИСАНИЕ
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Общие регистры USB (00h – 0Fh)		
00	FAddr	Регистр адреса функции.
01	Power	Регистр управления питанием.
02, 03	IntrTx	Регистр прерывания для оконечного устройства 0 и оконечных устройств передатчика от 1 до 15.
04 ,05	IntrRx	Регистр прерывания для оконечных устройств приемника от 1 до 15.
06, 07	IntrTxE	Регистр разрешения прерывания для IntrTx.
08, 09	IntrRxE	Регистр разрешения прерывания для IntrRx.
0A	IntrUSB	Регистр прерывания для общих прерываний USB.
0B	IntrUSBE	Регистр разрешения прерывания для IntrUSB.
0C, 0D	Frame	Номер фрейма.

Продолжение таблицы 1.719

АДР.	НАИМ.	ОПИСАНИЕ
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Общие регистры USB (00h – 0Fh)		
0E	Index	Индексный режим для выбора регистров состояния и управляющих регистров оконечной точки.
0F	Testmode	Разрешает тестовые режимы USB 2.0.
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Индексные регистры – режим ведомого устройства (10h – 1Fh)		
10, 11	TxMaxP	Максимальный размер пакета для оконечной точки передатчика ведомого устройства. (Регистр Index установлен только для выбора оконечных точек 1 – 15)
12, 13	CSR0L/H	Управляющий регистр состояния для оконечной точки 0. (Регистр Index установлен для выбора оконечной точки 0)
	TxCSRL/H	Управляющий регистр состояния для оконечной точки передатчика ведомого устройства. (Регистр Index установлен для выбора оконечных точек 1 – 15)
14, 15	RxMaxP	Максимальный размер пакета для оконечной точки приемника ведомого устройства. (Регистр Index установлен только для выбора оконечных точек 1 – 15)
16, 17	RxCSRL/H	Управляющий регистр и регистр состояния для оконечной точки приемника ведомого устройства. (Индексный регистр только для выбора оконечных точек 1 – 15)
18, 19	Count0	Количество принятых байт в FIFO оконечной точки 0. (Регистр Index установлен только для выбора оконечной точки 0)
	RxCount	Количество байт, которой будет считано из FIFO оконечной точки приемника ведомого устройства. (Регистр Index установлен для выбора оконечных точек 1 – 15)
1A–1B	–	Зарезервировано. Считываемое значение зависит от использования в режиме ведущего устройства (см. следующую стр.).
1C–1E	–	Не используется, всегда считывается 0.
1F	ConfigData	Информация о конфигурации ядра. (Регистр Index установлен для выбора оконечной точки 0.)
	FIFOSize	Размер выбранных FIFO буферов приемника и FIFO буферов передатчика (Только оконечные точки 1 – 15).
10,11	TxMaxP	Максимальный размер пакета для оконечной точки передатчика ведущего устройства. (Регистр Index установлен только для выбора оконечных точек 1 – 15)
12,13	CSR0L/H	Управляющий регистр и регистр состояния для оконечной точки 0. (Регистр Index установлен для выбора оконечной точки 0)
	TxCSRL/H	Управляющий регистр и регистр состояния для оконечной точки передатчика ведущего устройства. (Регистр Index установлен для выбора оконечных точек 1 – 15)
14,15	RxMaxP	Максимальный размер пакета для оконечной точки приемника ведущего устройства. (Регистр Index установлен только для выбора оконечных точек 1 – 15)
16,17	RxCSRL/H	Управляющий регистр и регистр состояния для оконечной точки приемника ведущего устройства. (Регистр Index установлен только для выбора оконечных точек 1 – 15)
18,19	Count0	Количество принятых байт в FIFO оконечной точки 0. (Регистр Index установлен для выбора оконечной точки 0)
	RxCount	Количество байт, которой будет считано из FIFO оконечной точки приемника ведущего устройства. (Регистр Index установлен для выбора оконечных точек 1 – 15)
1A	Type0	Определяет скорость оконечной точки 0. (Индексный регистр для выбора оконечной точки 0)
	TxType	Устанавливает протокол транзакции, скорость и номер оконечной точки ведомого устройства для оконечной точки передатчика ведущего устройства. (Индексный регистр для выбора оконечных точек 1 – 15)

Продолжение таблицы 1.719

АДР.	НАИМ.	ОПИСАНИЕ	
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Индексные регистры – режим ведомого устройства (10h – 1Fh)			
1B	NAKLimit0	Устанавливает время ожидания ответного пакета NAK на оконечной точке 0. (Регистр Index установлен для выбора оконечной точки 0)	
	TxInterval	Устанавливает интервал опроса для транзакций по прерыванию и изохронных транзакций, или время ожидания ответного пакета NAK при поточных транзакциях для оконечной точки передатчика ведущего устройства. (Регистр Index установлен только для выбора оконечных точек 1 – 15)	
1C	RxType	Устанавливает протокол транзакции, скорость и номер оконечной точки ведомого устройства для оконечной точки приемника ведущего устройства. (Регистр Index установлен только для выбора оконечных точек 1 – 15)	
1D	RxInterval	Устанавливает интервал опроса для транзакций по прерыванию и изохронных транзакций, или время ожидания ответного пакета NAK при поточных транзакциях для оконечной точки приемника ведущего устройства. (Регистр Index установлен только для выбора оконечных точек 1 – 15)	
1E	–	Не используется, всегда считывается 0.	
1F	ConfigData	Информация о конфигурации ядра. (Регистр Index установлен для выбора оконечной точки 0.)	
	FIFOSize	Размер выбранных FIFO буферов Rx и FIFO буферов TX (Только оконечные точки 1 – 15).	
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: FIFO (20h – 5Fh)			
20 – 5F	FIFOx	FIFO для оконечных точек 0 – 15.	
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Дополнительные управляющие регистры и регистры конфигурации (60h – 7Fh)			
60	DevCtl	Управляющий регистр устройства OTG.	
61	MISC	Общий регистр	
62	TxFIFOsz	Размер FIFO буфера оконечной точки передатчика	Используется только при выбранной опции динамического определения размера FIFO буфера. В противном случае считывается 0.
63	RxFIFOsz	Размер FIFO буфера оконечной точки приемника	
64,65	TxFIFOadd	Адрес FIFO буфера оконечной точки передатчика	
66,67	RxFIFOadd	Адрес FIFO буфера оконечной точки приемника	
68–6B	VControl/ VStatus	UTMI+ PHY Регистры производителя	
6C,6D	HWVers	Аппаратная версия регистра числа	
6E,6F	–	Не используется	
70 – 77	–	Регистры ULPI, имплементируемые только о при использовании ULPI Link Wrapper. См. ULPI Application Note musbhdrc_ulpi_an.pdf.	
78	EPInfo	Информация о количестве оконечных точек передатчика и Rx.	
79	RAMInfo	Информация о ширине RAM и количестве каналов DMA.	
7A	LinkInfo	Информация о применяемых задержках.	
7B	VPLen	Длительность импульса накачки питания (VBus pulsing).	
7C	HS_EOF1	Буфер для работы в реальном масштабе времени, доступный при транзакциях с высокой скоростью передачи.	
7D	FS_EOF1	Буфер для работы в реальном масштабе времени, доступный при транзакциях со средней скоростью передачи.	
7E	LS_EOF1	Буфер для работы в реальном масштабе времени, доступный при транзакциях с низкой скоростью передачи.	
7F	SOFT_RST	Сброс программы.	

Продолжение таблицы 1.719

АДР.	НАИМ.	ОПИСАНИЕ	
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Адресные регистры целевого устройства (80h – FFh) (Эти регистры действительны только если в конфигурации GUI разрешен режим с множеством ведомых устройств)			
80+8*n	TxFuncAddr	Адрес функции передающей оконечной точки n (Только режим ведущего устройства)	Только в режиме с множеством ведомых устройств
81+8*n	–	Не используется, всегда считывается 0.	Только в режиме с множеством ведомых устройств
82+8*n	TxHubAddr	Адрес хаба передающей оконечной точки n (Только режим ведущего устройства)	Только в режиме с множеством ведомых устройств
83+8*n	TxHubPort	Порт хаба передающей оконечной точки n (Только режим ведущего устройства)	Только в режиме с множеством ведомых устройств
84+8*n	RxFuncAddr	Адрес функции принимающей оконечной точки n (Только режим ведущего устройства)	Только в режиме с множеством ведомых устройств
85+8*n	–	Не используется, всегда считывается 0.	Только в режиме с множеством ведомых устройств
86+8*n	RxHubAddr	Адрес хаба принимающей оконечной точки n (Только режим ведущего устройства)	Только в режиме с множеством ведомых устройств
87+8*n	RxHubPort	Порт хаба принимающей оконечной точки n (Только режим ведущего устройства)	Только в режиме с множеством ведомых устройств
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Дополнительные регистры (200h – 27Ch)			
200h	DMA_INTR	Регистр прерывания DMA.	Регистры DMA доступны только при условии, что MUSBMHDRC сконфигурирован для использования как минимум одного внутреннего DMA. У каждого канала имеется свой набор регистров.
204h+ (n-1)*10h	DMA_CNTL	Управляющий регистр DMA для канала DMA с номером n. (каналы от 1 до 8).	
208h + (n-1)*10h	DMA_ADDR	Адресный регистр DMA для канала DMA с номером n. (каналы от 1 до 8).	
20Ch + (n-1)*10h	DMA_COUNT	Регистр-счетчик DMA для канала DMA с номером n. (каналы от 1 до 8).	
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Дополнительные регистры (304h – 347h)			
300+4*n	RqPktCount	Количество запрашиваемых пакетов для приемной оконечной точки n (Только оконечные точки 1 – 15)	
340, 341	Rx DPktBufDis	Регистр запрета двойной буферизации пакета для оконечных точек приемника от 1 до 15	
342, 343	TX DPktBufDis	Регистр запрета двойной буферизации пакета для оконечных точек передатчика от 1 до 15	
344, 345	C_T_UCH	Этот регистр устанавливает таймер времени ожидания ЛЧМ-сигнала.	
346, 347	C_T_HSRTN	Этот регистр устанавливает задержку относительно окончания выхода из спящего режима при высокой скорости соединения, разрешая стандартный режим работы UTM.	
РЕГИСТРОВАЯ ПАМЯТЬ MUSBMHDRC: Регистры LPM (360h – 365h)			
360	LPM_ATTR	Регистр атрибутов LPM	
362	LPM_CNTRL	Управляющий регистр LPM	
363	LPM_INTREN	Регистр разрешения прерывания LPM	
364	LPM_INTR	Регистр прерывания LPM	
365	LPM_FADDR	Регистр адреса функции LPM	

Примечание: в приведенных ниже описаниях битов:

‘r’ означает, что бит предназначен только для чтения, ‘rw’ – что бит предназначен как для чтения, так и для записи, ‘set’ – бит может быть только установлен, ‘r/set’ – бит может быть только прочитан или установлен, но не может быть сброшен, ‘clear’ – бит может быть только сброшен, ‘r/clear’ - бит может быть только прочитан или сброшен, но не может быть установлен, ‘self-clearing’ означает, что бит будет автоматически сброшен после выполнения ассоциированного с ним действия.

1.3.12.2.7.3.2 Общие регистры

- описаны в порядке возрастания адреса

1.3.12.2.7.3.2.1 F A DDR

FAddr – это 8-битовый регистр, в который должен записываться 7-битовый адрес ведомой части транзакции.

При использовании MUSBMHDRC в режиме ведомого устройства (DevCtl.D2=0) в этот регистр должен записываться адрес, полученный посредством команды SET_ADDRESS, который затем будет использоваться для декодирования адреса функции в последующих пакетах маркера (token packets).

Примечание: Только режим ведомого устройства!

Конфигурация ядра с поддержкой режима множества ведомых устройств: Этот регистр используется только в операциях, выполняемых MUSBMHDRC в режиме ведомого устройства. В режиме ведущего устройства регистр игнорируется.

Конфигурация ядра с поддержкой режима множества ведомых устройств: Этот регистр используется в операциях, выполняемых MUSBMHDRC в режиме как ведущего, так и ведомого устройства.

Адрес: 00h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	0	Адрес функции						
От CPU	r	rw	rw	rw	rw	rw	rw	rw
От USB	-	r	R	r	r	r	r	r

Бит	Наимен.	Функция
D7	-	Не используется, всегда считывается 0.
D6 – D0	Func Addr	Адрес функции.

1.3.12.2.7.3.2.2 P O WE R

Power - это 8-битовый регистр, который используется для управления сигналами входа в спящий режим и выхода из спящего режима, а также некоторыми основными аспектами работы MUSBMHDRC.

		D7	D6	D5	D4	D3	D2	D1	D0
		ISO Update	Soft Conn	HS Enab	HS Mode	Reset	Resume	Suspend Mode	Enable SuspendM
Режим ведомого устройства	От CPU	rw	rw	rw	r	r	rw	r	rw
	От USB	r	r	r	rw	rw	r	rw	r
Режим ведущего устройства	От CPU	-	-	rw	r	rw	rw	set	rw
	От USB	-	-	r	rw	rw	r/set	clear	r

Бит	Наимен.	Функция
D7	ISO Update	Если этот бит установлен CPU, то MUSBMHDRC после установки TxPktRdy будет ожидать маркера SOF, прежде чем отправить пакет. Если маркер чтения получен раньше маркера SOF, тогда отправляется пакет данных нулевой длины. Примечание: Действует только в режиме ведомого устройства. Этот бит также влияет на выполнение оконечными точками изохронной передачи.
D6	Soft Conn	При разрешенной функции программного соединения/отсоединения (Soft Connect/Disconnect) линии USB D+/D- разрешаются, когда CPU устанавливает этот бит, и находятся в третьем состоянии, когда CPU сбрасывает этот бит. (См. раздел 8.2.) Примечание: Действует только в режиме ведомого устройства.
D5	HS Enab	Если этот бит установлен CPU, то после сброса устройства хабом MUSBMHDRC пошлет запрос на высокую скорость соединения. Если бит не установлен, устройство будет работать только в режиме средней скорости соединения.
D4	HS Mode	Если этот бит «только для чтения», установлен, он указывает на успешное согласование режима высокой скорости соединения во время сброса USB. В режиме ведомого устройства действителен после завершения сброса USB (что указывается прерыванием сброса USB). В режиме ведущего устройства действителен после сброса бита Reset. Остается действительным на все время сеанса. Примечание: Исключение сделано для ЛЧМ-сигнала типа J (Tiny-J) при выборе скорости передачи.
D3	Reset	Этот бит устанавливается, когда на шине присутствует сигнал сброса. Исключение сделано для ЛЧМ-сигнала типа J. Примечание: Этот бит позволяет выполнять чтение и запись от CPU в режиме ведущего устройства, но только чтение в режиме ведомого устройства.
D2	Resume	Устанавливается CPU для генерации сигнала выхода из спящего режима, когда устройство находится в спящем режиме. В режиме ведомого устройства CPU должно сбросить этот бит по истечение 10 мс (максимум 15 мс), чтобы завершить сигнал выхода из спящего режима. В режиме ведущего устройства CPU должен сбросить этот бит через 20 мс.
D1	Suspend Mode	В режиме ведущего устройства этот бит устанавливается CPU для входа в спящий режим. В режиме ведомого устройства этот бит устанавливается после входа в спящий режим. Бит сбрасывается после считывания CPU регистра прерываний или установки бита Resume (предыдущего).
D0	Enable SuspendM	Устанавливается CPU для разрешения выходного сигнала SUSPENDM.

1.3.12.2.7.3.2.3 I N T R T X

IntrTx – это 16-битовый регистр «только для чтения», который указывает текущие активные прерывания для оконечной точки 0 и оконечных точек передатчика от 1 до 15. Примечание: Биты, относящиеся к оконечным точкам, которые не были сконфигурированы, всегда считываются как 0. Следует также отметить, что все активные прерывания сбрасываются при считывании регистра.

Адрес: 02h; Значение после сброса: 16’h0000

	D15	D14	D13	D12	D11	D10	D9	D8
	EP15 Tx	EP14 Tx	EP13 Tx	EP12 Tx	EP11 Tx	EP10 Tx	EP9 Tx	EP8 Tx
От CPU	r	r	r	r	r	r	r	r
От USB	set	set	set	set	set	set	set	set
	D7	D6	D5	D4	D3	D2	D1	D0
	EP7 Tx	EP6 Tx	EP5 Tx	EP4 Tx	EP3 Tx	EP2 Tx	EP1 Tx	EP0 Tx
От CPU	r	r	r	r	r	r	r	r
От USB	set	set	set	set	set	set	set	set

Бит	Наимен.	Функция
D15	EP15 TX	Прерывание оконечной точки 15 передатчика.
D14	EP14 TX	Прерывание оконечной точки 14 передатчика.
D13	EP13 TX	Прерывание оконечной точки 13 передатчика.
D12	EP12 TX	Прерывание оконечной точки 12 передатчика.
D11	EP11 TX	Прерывание оконечной точки 11 передатчика.
D10	EP10 TX	Прерывание оконечной точки 10 передатчика.
D9	EP9 TX	Прерывание оконечной точки 9 передатчика.
D8	EP8 TX	Прерывание оконечной точки 8 передатчика.
D7	EP7 TX	Прерывание оконечной точки 7 передатчика.
D6	EP6 TX	Прерывание оконечной точки 6 передатчика.
D5	EP5 TX	Прерывание оконечной точки 5 передатчика.
D4	EP4 TX	Прерывание оконечной точки 4 передатчика.
D3	EP3 TX	Прерывание оконечной точки 3 передатчика.
D2	EP2 TX	Прерывание оконечной точки 2 передатчика.
D1	EP1 TX	Прерывание оконечной точки 1 передатчика.
D0	EP0	Прерывание оконечной точки 0.

Бит	Наимен.	Описание
D7	B-Device	Этот бит «только для чтения» указывает, работает ли MUSBMHDRC в роли устройства ‘A’ или устройства ‘B’. 0 ⇒ устройство ‘A’; 1 ⇒ устройство ‘B’. Действителен только во время сеанса. Для определения роли в отсутствии сеанса необходимо установить бит сеанса, а затем считать этот бит. Примечание: Если ядро находится в режиме Force_Host (то есть, сеанс была начат с Testmode.D7 = 1), этот бит будет указывать на состояние входного сигнала HOSTDISCON от PHY.
D6	FSDev	Этот бит «только для чтения» устанавливается при обнаружении устройства со средней или высокой скоростью соединения, подключенного к порту. (Устройства с высокой скоростью соединения отличают от устройств со средней скоростью соединения при помощи проверки на высокоскоростный ЛЧМ-сигнал при сбросе устройства.) Действителен только в режиме ведущего устройства.
D5	LSDev	Этот бит «только для чтения» устанавливается при обнаружении устройства с низкой скоростью соединения, подключенного к порту. Действителен только в режиме ведущего устройства.
D4–D3	VBus[1:0]	Эти биты «только для чтения» кодируют текущий уровень VBus следующим образом:

D4	D3	Значение
0	0	Ниже SessionEnd
0	1	Выше SessionEnd, ниже AValid
1	0	Выше AValid, ниже VBusValid
1	1	Выше VBusValid

D2	Host Mode	Этот бит «только для чтения» устанавливается, когда MUSBMHDRC выступает в роли ведущего устройства.
D1	Host Req	При установке этого бита MUSBMHDRC инициирует согласование ведущего устройства при переходе в спящий режим. Сбрасывается по завершении согласования ведущего устройства. См. Раздел 15 (Только устройство ‘B’)
D0	Session	При работе в роли устройства ‘A’ этот бит устанавливается или сбрасывается CPU для начала и окончания сеанса. При работе в роли устройства ‘B’ этот бит устанавливается/сбрасывается MUSBMHDRC при начале/окончании сеанса. Он также устанавливается CPU для инициации протокола установки соединения. Когда MUSBMHDRC находится в спящем режиме, бит может быть сброшен CPU, чтобы выполнить программное отключение. Примечание: Сброс этого бита, когда ядро не находится в спящем режиме, приведет к неопределенному поседению.

1.3.12.2.7.3.2.4 M I SC

MISC - это 8-битовый регистр, который содержит различные общие биты конфигурации. Эти биты включают биты разрешения DMA Rx/TX Early. Биты конфигурации содержатся в регистре в соответствии с приведенной ниже таблицей.

Адрес: 61h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	Unused	Unused	Unused	Unused	Unused	Unused	tx_edma	rx_edma
От CPU	R	r	r	r	r	r	rw	rw
От USB	r	r	r	r	r	r	rw	rw

Бит	Наимен.	Функция
D7-D2	Unused	Эти биты зарезервированы.
D1	tx_edma	1’b0: сигнал DMA_REQ всех оконечных точек передатчика будет деактивирован после записи байтов MAXP в оконечную точку. Это режим отсроченной отправки. 1’b1: сигнал DMA_REQ всех оконечных точек передатчика будет деактивирован после записи байтов MAXP-8 в оконечную точку. Это режим досрочной отправки.
D0	rx_edma	1’b0: сигнал DMA_REQ всех оконечных точек приемника будет деактивирован после записи байтов MAXP в оконечную точку. Это режим отсроченной отправки. 1’b1: сигнал DMA_REQ всех оконечных точек приемника будет деактивирован после записи байтов MAXP-8 в оконечную точку. Это режим досрочной отправки.

1.3.12.2.7.3.3 ИНДЕКСНЫЕ РЕГИСТРЫ

Примечание: Действие этих регистров, когда выбранная оконечная точка не сконфигурирована, не определено.

1.3.12.2.7.3.3.1 C S R0L

CSR0L – это 8-битовый регистр, который содержит управляющие биты и биты состояния для оконечной точки 0.

CSR0L режиме ведомого устройства:
Адрес: 12h (с индексным регистром, установленным в 0); Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	Serviced SetupEnd	Serviced RxPktRdy	SendStall	SetupEnd	DataEnd	SentStall	TxPktRdy	RxPktRdy
От CPU	set	set	set	r	set	r/clear	r/set	r
От USB	r	r	r	set	r	set	rw	set

Бит	Наимен.	Функция в режиме ведомого устройства
D7	ServicedSetupEnd	CPU записывает 1 в этот бит, чтобы сбросить бит SetupEnd. Сбрасывается автоматически.
D6	ServicedRxPktRdy	CPU записывает 1 в этот бит, чтобы сбросить бит RxPktRdy. Сбрасывается автоматически.
D5	SendStall	CPU записывает 1 в этот бит, чтобы завершить текущую транзакцию. После передачи пакета STALL этот бит автоматически сбрасывается.
D4	SetupEnd	Этот бит устанавливается, когда управляющая транзакция заканчивается до установки бита DataEnd. Генерируется прерывание, и в это время FIFO очищается. Бит сбрасывается CPU посредством записи 1 в бит ServicedSetupEnd
D3	DataEnd	CPU устанавливает этот бит: При установке TxPktRdy для последнего пакета данных. При сбросе RxPktRdy после выгрузки последнего пакета данных. При установке TxPktRdy для пакета данных нулевой длины. Сбрасывается автоматически.
D2	SentStall	Этот бит устанавливается при передаче пакета STALL. Этот бит должен сбрасывать CPU.

Бит	Наимен.	Функция в режиме ведомого устройства
D1	TxPktRdy	CPU устанавливает этот бит после загрузки пакета данных в FIFO. Сбрасывается автоматически после передачи пакета данных. В этот момент также генерируется прерывание (если разрешено).
D0	RxPktRdy	Этот бит устанавливается после приема пакета данных. При установке этого бита генерируется прерывание. CPU сбрасывает этот бит посредством установки бита ServicedRxPktRdy.

CSR0L режиме ведущего устройства:

Адрес: 12h (с индексным регистром, установленным в 0); Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	NAK Timeout	StatusPkt	ReqPkt	Error	SetupPkt	RxStall	TxPktRdy	RxPktRdy
От CPU	r/clear	set	set	r/clear	r/clear	r/clear	r/set	r/clear
От USB	set	r	rw	set	r	set	clear	rw

Бит	Наимен.	Функция в режиме ведущего устройства
D7	Timeout NAK	Этот бит устанавливается, когда оконечная точка 0 остановлена после приема ответных пакетов NAK в течение времени большего, чем установлено регистром NAKLimit0. CPU должен сбросить этот бит, чтобы позволить оконечной точке продолжить работу.
D6	StatusPkt	CPU устанавливает этот бит одновременно с установкой бита TxPktRdy или ReqPkt, чтобы выполнить транзакцию отправки статуса. Установка этого бита подтверждает, что флаг данных (Data Toggle) установлен в 1, так что для транзакции статуса используется пакет DATA1.
D5	ReqPkt	CPU устанавливает этот бит в ответ на транзакцию чтения. Сбрасывается при установке бита RxPktRdy.
D4	Error	Этот бит устанавливается после трех попыток выполнить транзакцию, во время которых не был получен ответ ведомого устройства. Этот бит сбрасывается CPU. При установке этого бита генерируется прерывание.
D3	SetupPkt	CPU устанавливает этот бит одновременно с установкой бита TxPktRdy, чтобы отправить для транзакции маркер SETUP вместо маркера записи. Примечание: Установка этого бита также сбрасывает Data Toggle.
D2	RxStall	Этот бит устанавливается при передаче пакета STALL. Этот бит должен сбрасывать CPU.
D1	TxPktRdy	CPU устанавливает этот бит после загрузки пакета данных в FIFO. Сбрасывается автоматически после передачи пакета данных. В этот момент также генерируется прерывание (если разрешено).
D0	RxPktRdy	Этот бит устанавливается после приема пакета данных. При установке этого бита генерируется прерывание (если разрешено). CPU должен сбросить этот бит после считывания пакета из FIFO.

1.3.12.2.7.3.3.2 C S R0H

CSR0H – это 8-битовый регистр, содержащий управляющие биты и биты состояния для оконечной точки 0. Примечание: Интерпретация регистра зависит от режима работы MUSBMHDRC – в качестве ведомого или ведущего устройства. Пользователь должен помнить, что значение регистра при считывании отражает присвоенный статус, то есть, результат записи в регистр.

CSR0H режиме ведомого устройства:

Адрес: 13h (с индексным регистром, установленным в 0); Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	-	-	-	-	-	-	-	FlushFIFO (сбрасывается автоматически)
От CPU	r	r	r	r	r	r	r	set
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция в режиме ведомого устройства
D7 – D1	–	Не используется. Считывается 0.
D0	FlushFIFO	CPU записывает 1 в этот бит, чтобы очистить следующий пакет, который будет передаваться/считываться, из FIFO оконечной точки 0. Указатель FIFO сбрасывается, и бит TxPktRdy/RxPktRdy (ниже) также сбрасывается. Примечание: FlushFIFO следует использовать только при установленном бите TxPktRdy/RxPktRdy. В противном случае это может привести к искажению данных.

CSR0H режиме ведущего устройства:

Адрес: 13h (с индексным регистром, установленным в 0); Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	-	-	-	-	Dis Ping	Data Toggle Write Enable	Data Toggle	FlushFIFO
От CPU	r	r	r	r	rw	set	rw	set
От USB	r	r	r	r	r	r	rw	r

Бит	Наимен.	Функция в режиме ведущего устройства
D7 – D4	–	Не используется. Считывается 0.
D3	Dis Ping	CPU записывает 1 в этот бит, чтобы указать ядру не посылать маркеры PING в фазах данных и состояния высокоскоростной управляющей пересылки (для использования с устройствами, которые не отвечают на PING).
D2	Data Toggle Write Enable	CPU записывает 1 в этот бит, чтобы разрешить запись текущего состояния флага данных оконечной точки 0 (см. бит Data Toggle, ниже). Этот бит автоматически сбрасывается при записи нового значения.
D1	Data Toggle	При чтении этот бит указывает на текущее состояние флага данных оконечной точки 0. Если D10 имеет высокий уровень, этот бит может быть записан с установкой флага данных. Если D10 имеет низкий уровень, любое значение, записанное в бит, игнорируется.
D0	FlushFIFO	CPU записывает 1 в этот бит, чтобы очистить следующий пакет, который будет передаваться/считываться, из FIFO оконечной точки 0. Указатель FIFO сбрасывается, и бит TxPktRdy/RxPktRdy (ниже) тоже сбрасывается. Примечание: FlushFIFO следует использовать только при установленном бите TxPktRdy/RxPktRdy. В противном случае это может привести к искажению данных.

1.3.12.2.7.3.3.3 CO UNT 0

Count0 – это 7-битовый регистр «только для чтения», который указывает количество полученных байт в FIFO оконечной точки 0. Считываемое значение меняется в соответствии с изменением содержимого FIFO и действительно только при установленном RxPktRdy (CSR0.D0)

Адрес: 18h (с индексным регистром, установленным в 0); Значение после сброса: 7’b0000000

	D6	D5	D4	D3	D2	D1	D0
	(MSB)	Счетчик Rx оконечной точки 0					(LSB)
От CPU	r	r	r	r	r	r	r
От USB	w	w	w	w	w	w	w

1.3.12.2.7.3.3.4 T Y PE 0

ПРИМЕЧАНИЕ: ТОЛЬКО РЕЖИМ ВЕДУЩЕГО УСТРОЙСТВА!

Эти биты применяются только тогда, когда в конфигурации GUI включена опция работы с множеством ведомых устройств. Если работа с множеством ведомых устройств разрешена, то Type0 – это 8-битовый регистр, в котором имплементированы только биты 6 и 7. Эти биты должны записываться в соответствии с рабочей скоростью целевого устройства.

Адрес: 1Ah (с индексным регистром, установленным в 0); Значение после сброса: 2’b00

	D6	D5
	Speed	
От CPU	rw	rw
От USB	r	r

Бит	Наимен.	Функциия
D7– D6	Speed	Рабочая скорость целевого устройства: 00: Не используется (Примечание: В этом случае предполагается, что целевое устройство будет использовать ту же скорость, что и ядро.) 01: Высокая 10: Средняя 11: Низкая

1.3.12.2.7.3.3.5 CO NF IGD A T A

ConfigData – это 8-битовый регистр «только для чтения», хранящий информацию о выбранной конфигурации ядра.

Адрес: 1Fh(с индексным регистром, установленным в 0; Значение после сброса: Зависит от конфигурации

	D7	D6	D5	D4	D3	D2	D1	D0
	MP RxE	MPTxE	BigEndian	HBRxE	HBTxE	DynFIFO Sizing	SoftConE	UTMI DataWidth
От CPU	r	r	r	r	r	r	r	r
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7	MPRxE	При установке в ‘1’ выбирается автоматическое объединение пакетов поточной передачи (см. раздел 22)
D6	MPTxE	При установке в ‘1’ выбирается автоматическое разделение пакетов поточной передачи (см. раздел 22))
D5	BigEndian	Всегда “0”. Указывает на обратный порядок адресации.
D4	HBRxE	При установке в ‘1’ указывает, что выбран режим поддержки изохронных оконечных точек приемника с высокой пропускной способностью (High-bandwidth Rx ISO Endpoint Support).
D3	HBTxE	При установке в ‘1’ указывает, что выбран режим поддержки изохронных оконечных точек передатчика с высокой пропускной способностью.
D2	DynFIFO Sizing	При установке в ‘1’ указывает, что выбрана опция динамического определения размера FIFO.
D1	SoftConE	Всегда ‘1’. Указывает на программное подключение/отключение.
D0	UTMI DataWidth	Указывает на выбранную ширину данных UTMI+. Всегда 0, указывает на 8 бит.

1.3.12.2.7.3.3.6 N A KLI M IT 0

ПРИМЕЧАНИЕ: ТОЛЬКО РЕЖИМ ВЕДУЩЕГО УСТРОЙСТВА!

NAKLimit0 – это 5-битовый регистр, который устанавливает количество фреймов/микрофреймов (при высокой скорости соединения), после которых оконечная точка 0 должна перейти в режим ожидания последовательности ответных пакетов NAK. (Аналогичные установки для других оконечных точек могут быть сделаны в их регистрах TxInterval и RxInterval: см. разделы 3.3.15 и 3.3.17.)

Количество выбранных фреймов/микрофреймов равняется 2(m-1) (где m записанное регистр значение, действительные значения 2 – 16). Если ведущее устройство получает от целевого устройства ответные пакеты NAK для большего количества фреймов, чем число, указаное в этом регистре, оконечная точка будет остановлена. Примечание: Значение 0 или 1 запрещают функцию ожидания ответного пакета NAK.

Адрес: 1Bh (с индексным регистром, установленным в 0); Значение после сброса: 5’b00000

	D4	…	D0
	(MSB) Ограничение NAK (m) оконечной точки 0 (LSB)		
От CPU	rw	…	rw
От USB	w	…	r

1.3.12.2.7.3.3.7 T X M A XP

Регистр TxMaxP определяет максимальный объем данных, который может быть передан через выбранную оконечную точку передатчика за одну операцию. Регистр TxMaxP есть у каждой оконечной точки передатчика (за исключением оконечной точки 0).

Адрес: 10h; Значение после сброса: 11/13/16’h0000

	D12/15 D11	D10 ... D0
	m – 1	(MSB) Максимальное содержимое/транзакция (LSB)
От CPU	rw … rw	rw ... rw
От USB	r … r	r ... r

Биты 10:0 определяют максимальное содержимое (в байтах) одной транзакции. Может устанавливаться значение до 1024 байт, но на нее накладываются ограничения, указанные в спецификации USB, на размер пакетов для поточных, изохронных пересылок и пересылок по прерыванию в операциях со средней и высокой скоростью соединения.

Если при конфигурации ядра выбрана опция высокой скорости соединения для оконечных точек с изохронной/по прерыванию передачей или разделение пакета для оконечных точек с поточной передачей, регистр содержит либо 2, либо 5 дополнительных бит, которые определяют множитель m, чтобы получить значение больше записанного.

В случае оконечных точек потоковой пересылки с разрешенной опцией разделения пакетов множитель m может принимать значение до 32 и определяет максимальное число пакетов ‘USB’ (то есть, пакетов для передачи по USB) указанного содержимого, на которые должен разделяться один пакет данных, помещенных в FIFO. (Если опция разделения пакетов не включена, D15–D13 не имплементированы, а D12–D11(если включены) игнорируются.)

Примечание: Пакет данных должен быть кратным максимальному содержимому, указанному битами 10:0, которое, в свою очередь, может принимать значение 8, 16, 32, 64 или (в случае пересылок с высокой скоростью) 512 байт.

Для оконечных точек изохронной/по прерыванию пересылки в режиме высокой скорости соединения при включенной опции высокой пропускной способности, множитель m может принимать значение 2 или 3 (установленный бит 11 или бит 12 соответственно) и указывает максимальное количество таких транзакций, которые могут иметь место в одном микрофрейме. Если бит 11 или бит 12 не равен нулю, MUSBMHDRC автоматически разделит записанный в FIFO пакет данных на 2 или 3 пакета ‘USB’, каждый из которых будет содержать указанное содержимое (или меньшее). Максимальное содержимое для каждой транзакции составляет 1024 байта, так что в каждом микрофрейме разрешено передавать до 3072 байт.

Значение, записанное в битах 10:0 (умноженное на m в случае изохронных пересылок с высокой пропускной способностью) должно совпадать со значением, приведенным в поле wMaxPacketSize стандартного дескриптора для соответствующей оконечной точки (см. USB Specification Версия 2.0, Глава 9). Несовпадение может привести к непредсказуемому результату.

Общий объем данных, который определяется числом, записанным в этом регистре (равен содержимому x m) не должен превышать размер FIFO для оконечной точки передатчика, а также не должен превышать размер половины FIFO, если требуется двойная буферизация.

Если содержимое этого регистра изменяется после отправки пакетов из оконечной точки, FIFO оконечной точки FIFO должен быть полностью очищен (при помощи бита FlushFIFO в TxCSRL) после записи нового значения в этот регистр.

Примечание: В режиме 1 DMA для правильной генерации прерывания в регистр TxMaxP должно быть записано четное число байт.

1.3.12.2.7.3.3.8 T X C S R L

CSR0L – это 8-битовый регистр, который содержит управляющие биты и биты состояния для оконечной точки 0.

В режиме ведомого устройства:
Адрес: 12h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	IncompTx	ClrDataTog	SentStall	SendStall	FlushFIFO	UnderRun	FIFONotEmpty	TxPktRdy
От CPU	r/clear	set	r/clear	rw	set	r/clear	r/clear	r/set
От USB	set	r/clear	set	r	r	set	set	Clear

Бит	Наимен.	Функция в режиме ведомого устройства
D7	IncompTx	При использовании оконечной точки для изохронной пересылки с высокой пропускной способностью, этот бит устанавливается для указания ситуации, когда большой пакет разделяется на 2 или 3 пакета для передачи, но было получено недостаточно маркеров чтения для передачи всех частей. Примечание: Во всех других пересылках, кроме изохронных, этот бит всегда будет считываться 0.
D6	ClrDataTog	CPU записывает 1 в этот бит, чтобы сбросить в 0 флаг данных оконечной точки.
D5	SentStall	Этот бит устанавливается при передаче пакета STALL. FIFO очищается, а бит TxPktRdy сбрасывается (см. ниже). CPU должен сбросить этот бит.
D4	SendStall	CPU записывает 1 в этот бит, чтобы выдать пакета STALL на маркер чтения. CPU сбрасывает этот бит для выхода из состояния останова. Примечание: Этот бит не оказывает влияния, если оконечная точка используется для изохронных пересылок.
D3	FlushFIFO	CPU записывает 1 в этот бит, чтобы очистить последний пакет из FIFO оконечной точки передатчика. Указатель FIFO сбрасывается, бит TxPktRdy (ниже) сбрасывается, и генерируется прерывание. Может устанавливаться одновременно с TxPktRdy, чтобы выгрузить пакет, загруженный в FIFO. Примечание: FlushFIFO должен использоваться только при установленном TxPktRdy. В противном случае это может вызвать искажение данных. Также следует помнить, что если FIFO используется для двойной буферизации, то возможно, FlushFIFO потребуется устанавливать дважды, чтобы полностью очистить FIFO.
D2	UnderRun	USB устанавливает этот бит, если получен маркер чтения, при не установленном TxPktRdy. CPU должен сбросить этот бит.
D1	FIFONotEmpty	USB устанавливает этот бит, когда в FIFO буфере передатчика содержится как минимум 1 пакет.

Бит	Наимен.	Функция в режиме ведомого устройства
D0	TxPktRdy	CPU устанавливает этот бит после загрузки пакета данных в FIFO. Он сбрасывается автоматически после передачи пакета данных. В этот момент также генерируется прерывание (если разрешено). TxPktRdy также автоматически сбрасывается до загрузки второго пакета в двойной буфер FIFO.

В режиме ведущего устройства:
Адрес: 12h; Значение после сброса: 8'h00

	D7	D6	D5	D4	D3	D2	D1	D0
	NAK Timeout/ IncompTx	ClrDataTog	RxStall	SetupPkt	FlushFIFO	Error	FIFONotEmpty	TxPktRdy
От CPU	r/clear	set	r/clear	rw	set	r/clear	r/clear	r/set
От USB	set	r/clear	set	r	r	set	set	Clear

Бит	Наимен.	Функция в режиме ведущего устройства
D7	NAK Timeout IncompTx	Только оконечные точки в режиме поточной пересылки: Этот бит будет установлен, когда оконечная точка передатчика остановлена после приема ответных пакетов NAK в течение времени, большего, чем установлено как NAK Limit регистром TxInterval. CPU должен сбросить этот бит, чтобы разрешить оконечной точке продолжить работу. Только оконечные точки в режиме пересылки по прерыванию с высокой пропускной способностью: Этот бит будет установлен при отсутствии ответа от устройства, которому отправляется пакет.
D6	ClrDataTog	CPU записывает 1 в этот бит, чтобы сбросить в 0 флаг данных оконечной точки.
D5	RxStall	Этот бит устанавливается при приеме пакета STALL. Когда этот бит установлен, любой исполняемый в текущий момент запрос DMA прерывается, FIFO полностью очищается, и бит TxPktRdy сбрасывается (см. ниже). CPU должен сбросить этот бит.
D4	SetupPkt	CPU устанавливает этот бит одновременно с битом TxPktRdy, чтобы послать для транзакции маркер SETUP вместо выходного маркера. Примечание: Установка этого бита также сбрасывает Data Toggle.
D3	FlushFIFO	CPU записывает 1 в этот бит, чтобы очистить последний пакет из FIFO оконечной точки передатчика. Указатель FIFO сбрасывается, бит TxPktRdy (ниже) сбрасывается, и генерируется прерывание. Может устанавливаться одновременно с TxPktRdy, чтобы выгрузить пакет, загруженный в FIFO. Примечание: FlushFIFO должен использоваться только при установленном TxPktRdy. В противном случае это может вызвать искажение данных. Также следует помнить, что если FIFO используется для двойной буферизации, то возможно, FlushFIFO потребуется устанавливать дважды, чтобы полностью очистить FIFO.
D2	Error	USB устанавливает этот бит после 3 попыток отправить пакет, когда не было получено пакета согласования. При установке этого бита генерируется прерывание, сбрасывается TxPktRdy, и FIFO полностью очищается. CPU должен сбросить этот бит. Действителен только когда оконечная точка работает в режиме поточной пересылки или пересылки по прерыванию.
D1	FIFONotEmpty	USB устанавливает этот бит, когда в FIFO буфере передатчика содержится как минимум 1 пакет.
D0	TxPktRdy	CPU устанавливает этот бит после загрузки пакета данных в FIFO. Он сбрасывается автоматически после передачи пакета данных. В этот момент также генерируется прерывание (если разрешено). TxPktRdy также автоматически сбрасывается до загрузки второго пакета в двойной буфер FIFO.

1.3.12.2.7.3.3.9 TX CS RH

TxCSRH – это 8-битовый регистр, обеспечивающий дополнительное управление для пересылок через выбранную в данный момент оконечную точку передатчика. Регистр TxCSRH имеется у каждой сконфигурированной оконечной точки передатчика. Примечание: Интерпретация регистра зависит от режима работы MUSBMHDRC – в качестве ведомого или ведущего устройства. Пользователь должен помнить, что значение регистра при считывании отражает присвоенный статус, то есть, результат записи в регистр.

В режиме ведомого устройства:

Адрес: 13h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	AutoSet	ISO	Mode	DMAReqEnab	FrcDataTog	DMAReqMode	-	-
От CPU	rw	rw	rw	rw	rw	rw	r	r
От USB	r	r	r	r	r	r	r	r

Бит	Наим.	Функция в режиме ведомого устройства
D7	AutoSet	Если CPU устанавливает этот бит, TxPktRdy будет автоматически установлен, когда данные пакета максимального размера (значение в TxMaxP) загружаются в FIFO буфер передатчика. Если загружен пакет меньше максимального, то TxPktRdy необходимо установить вручную. Примечание: Не следует устанавливать для оконечных точек в изохронном режиме пересылки с высокой пропускной способностью и для оконечных точек в режиме пересылки по прерыванию с высокой пропускной способностью.
D6	ISO	CPU устанавливает этот бит для оконечной точки передатчика для изохронных пересылок, и сбрасывает его для переключения оконечной точки передатчика на поточные пересылки и пересылки по прерыванию. Примечание: Этот бит оказывает влияние только в режиме ведомого устройства. В режиме ведущего устройства всегда считывается ноль.
D5	Mode	CPU устанавливает этот бит, чтобы определить направление оконечной точки как передатчика, и сбрасывает этот бит, чтобы определить направление оконечной точки как приемника. Примечание: Этот бит оказывает влияние, когда один и тот же FIFO оконечной точки используется как для транзакций записи, так и для транзакций чтения.
D4	DMAReqEnab	CPU устанавливает этот бит, чтобы разрешить запрос DMA для оконечной точки передатчика.
D3	FrcDataTog	CPU устанавливает этот бит для принудительного переключения флага данных оконечной точки и очистки пакета данных из FIFO, независимо от того, получен ли пакет ACK. Бит может быть использован оконечными точками передатчика в режиме пересылок по прерыванию, которые используются для обратной связи по скорости для оконечных точек в режиме изохронных пересылок.
D2	DMAReqMode	CPU устанавливает этот бит, чтобы выбрать режим 1 запроса DMA, и сбрасывает, чтобы выбрать режим 0 запроса DMA. Примечание: Этот бит не должен сбрасываться ни раньше, ни в том же цикле, в котором сбрасывается бит DMAReqEnab (см. выше).
D1 – D0	–	Не используется, всегда считывается 0.

В режиме ведущего устройства:

Адрес: 13h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	AutoSet	-	Mode	DMAReqEnab	FrcDataTog	DMAReqMode	Data Toggle Wr. Enable (автоматический сброс)	Data Toggle
От CPU	rw	r	rw	rw	rw	rw	set	rw
От USB	r	r	r	r	r	r	r	rw

Бит	Наим.	Функция в режиме ведущего устройства
D7	AutoSet	Если CPU устанавливает этот бит, TxPktRdy будет автоматически установлен, когда данные пакета максимального размера (значение в TxMaxP) загружаются в FIFO буфер передатчика. Если загружен пакет меньше максимального, то TxPktRdy необходимо установить вручную. Примечание: Не следует устанавливать для оконечных точек в изохронном режиме пересылки с высокой пропускной способностью и для оконечных точек в режиме пересылки по прерыванию с высокой пропускной способностью.
D6	–	Не используется, всегда считывается 0.
D5	Mode	CPU устанавливает этот бит, чтобы определить направление оконечной точки как передатчика, и сбрасывает этот бит, чтобы установить направление как приемника. Примечание: Этот бит оказывает влияние, когда тот же FIFO оконечной точки используется как для транзакций записи, так и для транзакций чтения.
D4	DMAReqEnab	CPU устанавливает этот бит, чтобы разрешить запрос DMA для оконечной точки передатчика.
D3	FrcDataTog	CPU устанавливает этот бит для принудительного переключения флага данных оконечной точки и очистки пакета данных из FIFO, независимо от того, получен ли пакет ACK. Он может быть использован оконечными точками передатчика в режиме пересылок по прерыванию, которые используются для обратной связи по скорости для оконечных точек в режиме изохронных пересылок.
D2	DMAReqMode	CPU устанавливает этот бит, чтобы выбрать режим 1 запроса DMA, и сбрасывает, чтобы выбрать режим 0 запроса DMA. Примечание: Этот бит не должен сбрасываться ни раньше, ни в том же цикле, в котором сбрасывается бит DMAReqEnab (см. выше).
D1	Data Toggle Write Enable	CPU записывает в этот бит 1, чтобы разрешить запись текущего состояния флага данных оконечной точки передатчика (см. бит Data Toggle, ниже). Этот бит автоматически сбрасывается при записи нового значения.
D0	Data Toggle	При чтении этот бит указывает текущее состояние флага данных оконечной точки передатчика. Если D1 имеет высокий уровень, это может быть записано требуемой установкой флага данных. Если D1 имеет низкий уровень, то любое значение этого бита игнорируется.

1.3.12.2.7.3.3.10 R X M A XP

Регистр RxMaxP определяет максимальный объем данных, который может быть переслан через выбранную оконечную точку приемника за одну операцию. Регистр RxMaxP есть для каждой оконечной точки приемника (за исключением оконечной точки 0).

Адрес: 10h; Значение после сброса: 11/13/16’h0000

	D12/15 D11	D10 ... D0
	m – 1	(MSB) Максимальное содержимое/транзакция (LSB)
От CPU	rw … rw	rw ... rw
От USB	r … r	r ... r

Биты 10:0 определяют (в байтах) максимальное содержимое в одной транзакции. Может устанавливаться величина до 1024 байт, но на нее накладываются ограничения, указанные в

спецификации USB, на размер пакетов для поточных, изохронных пересылок и пересылок по прерыванию в операциях со средней и высокой пропускной способностью.

Если при конфигурации ядра выбрана опция высокой пропускной способности для оконечных точек с изохронной/по прерыванию передачей или разделение пакета для оконечных точек с поточной передачей, регистр содержит либо 2, либо 5 дополнительных бит, которые определяют множитель m, чтобы получить значение больше записанного.

В случае оконечных точек с потоковой передачей с разрешенной опцией разделения пакетов множитель m может принимать значение до 32 и определяет максимальное число пакетов USB указанного содержимого, которые должны соединяться в один пакет данных, помещенных в FIFO. (Если опция разделения пакетов не включена, D15–D13 не имплементированы, а D12–D11(если включены) игнорируются.)

Для оконечных точек с изохронной/по прерыванию передачей в режиме высокой скорости соединения при включенной опции высокой пропускной способности, множитель m может принимать значение 2 или 3 (установленный бит 11 или бит 12 соответственно) и указывает максимальное количество таких транзакций, которые могут иметь место в одном микрофрейме. Если бит 11 или бит 12 не равен нулю, MUSBMHDRC автоматически соединит отдельные пакеты USB, полученные в любом микрофрейме, в один пакет в FIFO приемника. Максимальное содержимое для каждой транзакции составляет 1024 байта, так что в каждом микрофрейме разрешено передавать до 3072 байт. (Для изохронных пересылок со средней скоростью соединения или если высокая пропускная способность не включена, биты 11 и 12 игнорируются.)

Значение, записанное в битах 10:0 (умноженное на m в случае изохронных пересылок с высокой пропускной способностью) должно совпадать со значением, приведенным в поле wMaxPacketSize стандартного дескриптора для соответствующей оконечной точки (см. USB Specification Версия 2.0, Глава 9). Несовпадение может привести к непредсказуемому результату.

Общий объем данных, который определяется числом, записанным в этом регистре (равен содержимому x m) не должен превышать размер FIFO для оконечной точки приемника, а также не должен превышать размер половины FIFO, если требуется двойная буферизация.

Примечание: В режиме 1 DMA для правильной генерации прерывания в регистр RxMaxP должно быть записано четное число байт.

1.3.12.2.7.3.3.11 R X CS RL

RxCSRL – это 8-битовый регистр, который содержит управляющие биты и биты состояния для пересылок через выбранную в данный момент оконечную точку приемника. Регистр RxCSRL имеется у каждой сконфигурированной оконечной точки приемника. Примечание: Интерпретация регистра зависит от режима работы MUSBMHDRC – в качестве ведомого или ведущего устройства. Пользователь должен помнить, что значение регистра при считывании отражает присвоенный статус, то есть, результат записи в регистр.

В режиме ведомого устройства:
Адрес: 16h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	ClrDataTog	SentStall	SendStall	FlushFIFO (автоматический сброс)	DataError	OverRun	FIFOFull (автоматический сброс)	RxPktRdy
От CPU	set	r/clear	rw	set	r	r/clear	r	r/clear
От USB	r/clear	set	r	r	set	set	set	set

Бит	Наим.	Функция в режиме ведомого устройства
D7	ClrDataTog	CPU записывает 1 в этот бит, чтобы сбросить в 0 флаг данных оконечной точки.
D6	SentStall	Этот бит устанавливается при передаче пакета STALL. CPU должен сбросить этот бит.
D5	SendStall	CPU записывает 1 в этот бит, чтобы выдать согласование пакета STALL. CPU сбрасывает этот бит для выхода из состояния останова. Примечание: Этот бит не оказывает влияния, если оконечная точка используется для изохронных пересылок.
D4	FlushFIFO	CPU записывает 1 в этот бит, чтобы очистить следующий считываемый пакет из FIFO оконечной точки приемника. Указатель FIFO сбрасывается, бит TxPktRdy (ниже) сбрасывается. Примечание: FlushFIFO должен использоваться только при установленном TxPktRdy. В противном случае это может вызвать искажение данных. Также следует помнить, что если FIFO используется для двойной буферизации, то возможно, FlushFIFO потребуется устанавливать дважды, чтобы полностью очистить FIFO.
D3	DataError	Этот бит устанавливается при установленном RxPktRdy, если пакет данных имеет ошибку CRC или битстаффинга. Бит сбрасывается при сбросе RxPktRdy. Примечание: Этот бит действителен, если оконечная точка работает в режиме изохронных пересылок. В режиме поточных пересылок всегда считывается нулевое значение.
D2	OverRun	Этот бит устанавливается, если выходной пакет не может быть загружен в FIFO буфер приемника. CPU должен сбросить этот бит. Примечание: Этот бит действителен, если оконечная точка работает в режиме изохронных пересылок. В режиме поточных пересылок всегда считывается нулевое значение.
D1	FIFOFull	Этот бит устанавливается, когда больше невозможно загрузить пакеты в FIFO буфер приемника.
D0	RxPktRdy	Этот бит устанавливается после получения пакета данных. CPU должен сбросить этот бит после выгрузки пакета из FIFO буфера приемника. При установке бита генерируется прерывание.

В режиме ведущего устройства:
Адрес: 16h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	ClrDataTog	RxStall	ReqPkt	FlushFIFO (сбрасывается автоматически)	DataError/ NAK Timeout	Error	FIFOFull (сбрасывается автоматически)	RxPktRdy
От CPU	set	r/clear	rw	set	r(/clear)	r/clear	r	r/clear
От USB	r/clear	set	rw	r	set	set	set	set

Бит	Наимен.	Функция в режиме ведомого устройства
D7	ClrDataTog	CPU записывает 1 в этот бит, чтобы сбросить в 0 флаг данных оконечной точки.
D6	RxStall	Этот бит устанавливается при приеме согласования пакета STALL; генерируется прерывание. CPU должен сбросить этот бит.
D5	ReqPkt	CPU записывает 1 в этот бит для запроса на транзакцию чтения. Бит сбрасывается после установки RxPktRdy.
D4	FlushFIFO	CPU записывает 1 в этот бит, чтобы очистить следующий считываемый пакет из FIFO оконечной точки приемника. Указатель FIFO сбрасывается, бит TxPktRdy (ниже) сбрасывается. Примечание: FlushFIFO должен использоваться только при установленном TxPktRdy. В противном случае это может вызвать искажение данных. Также следует помнить, что если FIFO используется для двойной буферизации, то возможно, FlushFIFO потребуется устанавливать дважды, чтобы полностью очистить FIFO.

Бит	Наимен.	Функция в режиме ведомого устройства
D3	DataError/ NAK Timeout	При работе в режиме изохронной пересылки этот бит устанавливается при установленном бите RxPktRdy, если пакет данных имеет ошибку CRC или битстаффинга, и сбрасывается при сбросе RxPktRdy. В режиме поточной пересылки этот бит устанавливается, когда оконечная точка приемника остановлена после приема ответных пакетов NAK в течение времени, большего, чем установлено как NAK Limit регистром RxInterval. CPU должен сбросить этот бит, чтобы разрешить оконечной точке продолжить работу. Однако при разрешении двойной буферизации пакетов одного этого недостаточно для продолжения работы. При сбросе этого бита необходимо в том же цикле установить бит ReqPkt.
D2	Error	USB устанавливает этот бит после 3 неудачных попыток получения пакета данных. CPU должен сбросить этот бит. При установке этого бита генерируется прерывание. Примечание: Этот бит действителен только когда оконечная точка работает в режиме поточной пересылки или пересылки по прерыванию. В режиме изохронной пересылки всегда считывается нулевое значение.
D1	FIFOFull	Этот бит устанавливается, когда больше невозможно загрузить пакеты в FIFO буфер приемника.
D0	RxPktRdy	Этот бит устанавливается после получения пакета данных. CPU должен сбросить этот бит после выгрузки пакета из FIFO буфера приемника. При установке бита генерируется прерывание.

1.3.12.2.7.3.3.12 R X CS RH

RxCSRH – это 8-битовый регистр, обеспечивающий дополнительное управление для пересылок через выбранную в данный момент оконечную точку приемника. Регистр RxCSRH имеется у каждой сконфигурированной оконечной точки приемника (за исключением оконечной точки 0). Примечание: Интерпретация регистра зависит от режима работы MUSBMHDRC – в качестве ведомого или ведущего устройства. Пользователь должен помнить, что значение регистра при считывании отражает присвоенный статус, то есть, результат записи в регистр.

В режиме ведомого устройства:

Адрес: 17h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	AutoClear	ISO	DMAReqEnab	DisNyet/ PID Error	DMAReqMode	-	-	
От CPU	rw	rw	rw	rw	rw	rw	r	r
От USB	r	r	r	r	r	r	r	r

<table>
<tr><th>Бит</th><th>Наимен.</th><th>Функция в режиме ведомого устройства</th></tr>
<tr><td>D7</td><td>AutoClear</td><td>Если CPU устанавливает этот бит, то RxPktRdy автоматически сбрасывается, когда пакет размером RxMaxP байт выгружается из FIFO буфера приемника. Когда выгружаются пакеты меньше максимального размера, то RxPktRdy следует сбрасывать вручную. При использовании DMA для выгрузки FIFO буфера приемника данные считываются из FIFO буфера Rx 4-байтными частями независимо от RxMaxP. Следовательно, RxPktRdy будет сбрасываться следующим образом:

<table>
<tr><th>Остаток (RxMaxP/4)</th><th>Считанные байты</th><th>Размеры пакетов, сбрасывающие RxPktRdy.</th></tr>
<tr><td>0 (т.е. RXMaxP = 64 байта)</td><td>RXMAXP</td><td>RXMAXP, RXMAXP-1, RXMAXP-2, RXMAXP-3</td></tr>
<tr><td>3 (т.е. RXMaxP = 63 байта)</td><td>RXMAXP+ 1</td><td>RXMAXP, RXMAXP-1, RXMAXP-2</td></tr>
<tr><td>2 (т.е. RXMaxP = 62</td><td>RXMAXP+</td><td>RXMAXP, RXMAXP-1</td></tr>
<tr><td>1 (т.е. RXMaxP = 61</td><td>RXMAXP+</td><td>RXMAXP</td></tr>
</table>

Примечание: Не следует устанавливать для оконечных точек в режиме изохронной пересылки с высокой пропускной способностью.</td></tr>
<tr><td>D6</td><td>ISO</td><td>CPU устанавливает этот бит для изохронных пересылок и сбрасывает для разрешения оконечной точке поточных пересылок и пересылок по прерыванию.</td></tr>
<tr><td>D5</td><td>DMAReqEnab</td><td>CPU устанавливает этот бит для разрешения запроса DMA для оконечной точки приемника.</td></tr>
<tr><td>D4</td><td>DisNyet

PID Error</td><td>Поточные пересылки/Пересылки по прерыванию: CPU устанавливает этот бит, чтобы запретить посылку пакетов NYET. После установки все успешно принятые пакеты чтения являются пакетами ACK, в том числе в момент заполнения FIFO.
Примечание: Этот бит оказывает влияние только в режиме высокой скорости соединения, и в этом режиме он должен быть установлен для всех оконечных точек в режиме пересылок по прерыванию.
Изохронные пересылки: Ядро устанавливает этот бит для указания на ошибку PID в принятом пакете.</td></tr>
<tr><td>D3</td><td>DMAReqMode</td><td>CPU устанавливает этот бит, чтобы выбрать режим 1 запроса DMA, и сбрасывает, чтобы выбрать режим 0 запроса DMA.</td></tr>
<tr><td>D2–D1</td><td>–</td><td>Не используется, всегда считывается 0.</td></tr>
<tr><td>D0</td><td>IncompRx</td><td>Этот бит устанавливается при изохронной пересылке или пересылке по прерыванию с высокой пропускной способностью, если пакет в FIFO буфере приемника неполон из-за того, что часть данных не была получена. Сбрасывается при сбросе RxPktRdy.
Примечание: Во всех других пересылках, кроме изохронных, этот бит всегда будет считываться 0.</td></tr>
</table>

В режиме ведущего устройства:
Адрес: 17h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	AutoClear	AutoReq	DMAReqEnab	PID Error	DMAReqMode	Data Toggle Wr. Enable	Data Toggle	Incomp Rx
От CPU	rw	rw	rw	r	rw	r	r	r/clear
От USB	r	r	r	rw	r	r	r	set

<table>
<tr><th>Бит</th><th>Наимен.</th><th>Функция в режиме ведущего устройства</th></tr>
<tr><td>D7</td><td>AutoClear</td><td>Если CPU устанавливает этот бит, то RxPktRdy автоматически сбрасывается, когда пакет из RxMaxP байт выгружается из FIFO буфера приемника. Когда выгружаются пакеты меньше максимального размера, то RxPktRdy следует сбрасывать вручную. При использовании DMA для выгрузки FIFO буфера приемника данные считываются из FIFO буфера Rx 4-байтными частями независимо от RxMaxP. Следовательно, RxPktRdy будет сбрасываться следующим образом:

<table>
<tr><th>Остаток (RxMaxP/4)</th><th>Считанные байты</th><th>Размеры пакетов, сбрасывающие RxPktRdy.</th></tr>
<tr><td>0 (т.е. RXMaxP = 64 байта)</td><td>RXMAXP</td><td>RXMAXP, RXMAXP-1, RXMAXP-2, RXMAXP-3</td></tr>
<tr><td>3 (т.е. RXMaxP = 63 байта)</td><td>RXMAXP+1</td><td>RXMAXP, RXMAXP-1, RXMAXP-2</td></tr>
<tr><td>2 (т.е. RXMaxP = 62</td><td>RXMAXP+2</td><td>RXMAXP, RXMAXP-1</td></tr>
<tr><td>1 (т.е. RXMaxP = 61</td><td>RXMAXP+3</td><td>RXMAXP</td></tr>
</table>

Примечание: Не следует устанавливать для оконечных точек в режиме изохронной пересылки с высокой пропускной способностью.</td></tr>
<tr><td>D6</td><td>AutoReq</td><td>Если CPU устанавливает этот бит, то ReqPkt будет автоматически установлен при сбросе бита RxPktRdy.
Примечание: Этот бит автоматически сбрасывается при приеме короткого пакета.</td></tr>
<tr><td>D5</td><td>DMAReqEnab</td><td>CPU устанавливает этот бит для разрешения запроса DMA для оконечной точки приемника.</td></tr>
<tr><td>D4</td><td>PID Error</td><td>Только изохронные пересылки: Ядро устанавливает этот бит для указания на ошибку PID в принятом пакете.
Поточные пересылки и пересылки по прерыванию: Установка этого бита игнорируется.</td></tr>
<tr><td>D3</td><td>DMAReqMode</td><td>CPU устанавливает этот бит, чтобы выбрать режим 1 запроса DMA, и сбрасывает, чтобы выбрать режим 0 запроса DMA.</td></tr>
<tr><td>D2</td><td>Data Toggle Write Enable</td><td>CPU записывает в этот бит 1, чтобы разрешить запись текущего состояния флага данных оконечной точки 0 (см. бит Data Toggle, ниже). Этот бит автоматически сбрасывается при записи нового значения.</td></tr>
<tr><td>D1</td><td>Data Toggle</td><td>При чтении этот бит указывает текущее состояние флага данных оконечной точки 0. Если D10 имеет высокий уровень, это может быть записано установкой флага данных. Если D10 имеет низкий уровень, то любое значение этого бита игнорируется.</td></tr>
<tr><td>D0</td><td>IncompRx</td><td>Этот бит устанавливается при изохронной пересылке или пересылке по прерыванию с высокой пропускной способностью, если пакет в FIFO буфере приемника неполон из-за того, что часть данных не была получена. Сбрасывается при сбросе RxPktRdy.
Примечание: При корректном выполнении протокола USB этот бит никогда не устанавливается. Установка этого бита указывает на неправильную работу соответствующего ведомого устройства. (Во всех других пересылках, кроме изохронных, этот бит всегда будет считываться 0.)</td></tr>
</table>

1.3.12.2.7.3.3.13 R X COUNT

RxCount – это 14-битовый регистр «только для чтения», который указывает количество байт в текущем пакете для чтения из FIFO буфера. Если пакет был передан в виде нескольких пакетов поточной передачи, значение соответствует объединенному пакету.

Примечание: Считываемое значение меняется по мере выгрузки FIFO и действительно только при установленном RxPktRdy (RxCSR.D0).

Адрес: 18h; Значение после сброса: 14’h0000

	D13	…	D0
	(MSB)	Счетчик Rx оконечной точки 0	(LSB)
От CPU	r	…	r
От USB	w	…	w

1.3.12.2.7.3.3.14 T X T Y PE

ПРИМЕЧАНИЕ: ТОЛЬКО РЕЖИМ ВЕДУЩЕГО УСТРОЙСТВА!

TxType – это 8-битовый регистр, в который должен быть записан номер, присвоенный оконечной точке, протокол транзакции для выбранной в текущий момент оконечной точки передатчика, а также рабочая скорость. Для каждой сконфигурированной оконечной точки передатчика имеется регистр TxType (за исключением оконечной точки 0, для которой выделен отдельный регистр Type с адресом 1Ah). D6-D7 действительны только когда ядро сконфигурировано для работы с множеством ведомых устройств.

Примечание: Значение изменяется по мере выгрузке FIFO и действительно только при установленном бите RxPktRdy (RxCSR.D0)

Адрес: 1Ah; Значение после сброса: 8'h00

	D7	D6	D5	D4	D3	D2	D1	D0
	Speed		Protocol		Target Endpoint Number			
От CPU	Rw	rw	rw	rw	rw	rw	rw	rw
От USB	R	r	r	r	r	r	r	r

Примечание: D6-D7 действительны только при выборе режима с множеством ведомых устройств; в противном случае биты не используются.

Бит	Наимен.	Функция в режиме ведущего устройства
D7–D6	Speed	Рабочая скорость целевого устройства, когда ядро сконфигурировано для работы с множеством ведомых устройств: 00: Не используется (Примечание: В этом случае предполагается, что целевое устройство будет использовать ту же скорость, что и ядро.) 01: Высокая 10: Средняя 11: Низкая Когда ядро не сконфигурировано для работы с множеством ведомых устройств, эти биты недоступны.
D5–D4	Protocol	CPU следует установить эти биты для выбора требуемого протокола оконечной точки передатчика: 00: Управляющий 01: Изохронный 10: Потоковый 11: По прерыванию
D3 – D0	Target Endpoint Number	CPU следует установить это значение, равное номеру оконечной точки, содержащемуся в дескрипторе оконечной точки передатчика и сообщенное MUSBMHDRC во время нумерации устройства.

1.3.12.2.7.3.3.15 T X INT E R V A L

Адрес: 1Bh; Значение после сброса: 8'b00000000

	D7	...	D0
	Tx Polling Interval/NAK Limit (m)		
От CPU	rw	...	RW
От USB	r	...	R

ПРИМЕЧАНИЕ: ТОЛЬКО РЕЖИМ ВЕДУЩЕГО УСТРОЙСТВА!

TxInterval – это 8-битовый регистр, который для пересылок по прерыванию и изохронных пересылок определяет интервал опроса для выбранной в текущий момент оконечной точки передатчика. Для оконечных точек в режиме поточных пересылок этот регистр устанавливает количество фреймов/микрофреймов после которых оконечная точка должна прекратить ожидание последовательности ответных пакетов NAK.

Для каждой сконфигурированной оконечной точки передатчика имеется регистр TxInterval (за исключением оконечной точки 0). В каждом случае установленное значение определяет количество фреймов/микрофреймов следующим образом:

Тип пересылки	Скорость	Допустимые значения (m)	Интерпретация
По прерыванию	Низкая или средняя	1 – 255	Интервал опроса m фреймов.
	Высокая	1 – 16	Интервал опроса 2(m-1) микрофреймов.
Изохронный	Средняя или высокая	1 – 16	Интервал опроса 2(m-1) фреймов/микрофреймов.
Поточный	Средняя или высокая	2 – 16	NAK Limit составляет 2(m-1) фреймов/микрофреймов. Примечание: Значение 0 и 1 запрещает функцию интервала ожидания NAK.

1.3.12.2.7.3.3.16 R X T Y P E

ПРИМЕЧАНИЕ: ТОЛЬКО РЕЖИМ ВЕДУЩЕГО УСТРОЙСТВА!

RxType – это 8-битовый регистр, в который должен быть записан номер, присвоенный оконечной точке, который будет использовать протокол транзакции для выбранной в текущий момент оконечной точки приемника, а также рабочая скорость. Для каждой сконфигурированной оконечной точки приемника имеется регистр RxType (за исключением оконечной точки 0, для которой выделен отдельный регистр Type с адресом 1Ah. D6-D7 действительны только когда ядро сконфигурировано для работы с множеством ведомых устройств.

Адрес: 1Ch; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
		*Speed		Protocol		Target Endpoint Number		
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция в режиме ведущего устройства
D7–D6	Speed	Рабочая скорость целевого устройства, когда ядро сконфигурировано для работы с множеством ведомых устройств: 00: Не используется (Примечание: В таком случае предполагается, что целевое устройство будет использовать ту же скорость, что и ядро.) 01: Высокая 10: Средняя 11: Низкая Когда ядро не сконфигурировано для работы с множеством ведомых устройств, эти биты недоступны.
D5–D4	Protocol	CPU следует установить эти биты для выбора требуемого протокола оконечной точки приемника: 00: Управляющий 01: Изохронный 10: Потоковый 11: По прерыванию
D3 – D0	Target Endpoint Number	CPU следует установить это значение, равное номеру оконечной точки, содержащемуся в дескрипторе оконечной точки приемника и сообщенное MUSBMHDRC во время нумерации устройства.

Примечание: D6-D7 действительны только при выборе режима с множеством ведомых устройств; в противном случае биты не используются.

1.3.12.2.7.3.3.17 R X INT E R V A L

ПРИМЕЧАНИЕ: ТОЛЬКО РЕЖИМ ВЕДУЩЕГО УСТРОЙСТВА!

RxInterval – это 8-битовый регистр, который для пересылок по прерыванию и изохронных пересылок определяет интервал опроса для выбранной в текущий момент оконечной точки передатчика. Для оконечных точек в режиме поточных пересылок этот регистр устанавливает количество фреймов/микрофреймов после которых оконечная точка должна прекратить ожидание последовательности ответных пакетов NAK. Для каждой сконфигурированной оконечной точки приемника имеется регистр RxInterval (за исключением оконечной точки 0). В каждом случае установленное значение определяет количество фреймов/микрофреймов (при высокой скорости соединения) следующим образом:

Адрес: 1Dh; Значение после сброса: 8’b00000000

	D7 ... D0
	Tx Polling Interval/NAK Limit (m)
От CPU	rw ... rw
От USB	r ... r

Тип пересылки	Скорость	Допустимые значения	Интерпретация
По прерыванию	Низкая или средняя	1 – 255	Интервал опроса m фреймов.
	Высокая	1 – 16	Интервал опроса 2(m-1) микрофреймов.
Изохронный	Средняя или высокая	1 – 16	Интервал опроса 2(m-1) фреймов/микрофреймов.
Поточный	Средняя или высокая	2 – 16	NAK Limit составляет 2(m-1) фреймов/микрофреймов. Примечание: Значение 0 и 1 запрещает функцию интервала ожидания NAK.

1.3.12.2.7.3.3.18 F I FO SI ZE

FIFOSize – это 8-битовый регистр «только для чтения», который хранит размеры FIFO буферов, соответствующих выбранным дополнительным оконечным точкам передатчика/приемника. Младший полубайт кодирует размер FIFO выбранных оконечных точек передатчика, старший полубайт кодирует размер FIFO выбранных оконечных точек приемника. Значения 3 – 13 соответствуют размеру FIFO 2n байт (8 – 8192 байт). Если оконечная точка не сконфигурирована, будет считываться значение 0. Если оконечные точки передатчика и приемника используют один и тот же FIFO буфер, то размер FIFO приемника будет кодироваться как 0xF.

Примечание: Регистр интерпретируется таким образом только если в регистре Index выбрана одна из оконечных точек 1 – 15 и не выбрана функция динамического определения размера FIFO. Он интерпретируется особенным образом, если в регистре Index выбрана оконечная точка 0 (см. раздел 1.3.12.2.7.3.3.4); при использовании динамического определения размера FIFO результат считывания регистра не действителен.

Address: 1Fh (в регистре Index выбрана только одна из оконечных точек 1 – 15); Reset value: Зависит от конфигурации

	D7 ... D4	D3 ... D0
	Rx FIFO Size	Tx FIFO Size
От CPU	r ... r	r ... r
От USB	r ... r	r ... r

1.3.12.2.7.3.4 F I F O (Адреса 20h – 5Fh)

Этот диапазон адресов включает 16 адресов для доступа CPU к FIFO буферу каждой оконечной точки. Запись по этим адресам загружает данные в FIFO передатчика соответствующей оконечной точки. Чтение по этим адресам выгружает данные из FIFO приемника соответствующей оконечной точки.

Диапазон адресов 20h – 5Fh, и FIFO расположены на 32-битовых двухсловных границах (оконечная точка 0 по адресу 20h, оконечная точка 1 - 24h ... оконечная точка 15 - 5Ch).

Примечание: (i) Пересылки в FIFO буфер могут быть 8-битовыми, 16-битовыми или 32-битовыми, и допустима любая комбинация доступа при условии непрерывности данных, к которым предоставляется доступ. Тем не мене, все пересылки, соответствующие одному пакету,

должны иметь одинаковую ширину, так чтобы данные были разделены на байты, слова или слова двойной длины. Однако последняя пересылка может содержать меньшее число байтов, чем предыдущие, чтобы завершить пересылку нечетного количества байтов или слов.

(ii) В зависимости от размера FIFO буфера и ожидаемого максимального размера пакета, FIFO поддерживает буферизацию либо одного, либо двух пакетов. Однако групповая запись множества пакетов не поддерживается, поскольку после записи каждого пакета требуется установка флагов.

(iii) После ответа STALL или ошибки TX Strike Out на оконечной точке 1 – 15 соответствующий FIFO буфер полностью очищается.

1.3.12.2.7.3.5 ДОПОЛНИТЕЛЬНЫЕ УПРАВЛЯЮЩИЕ РЕГИСТРЫ И РЕГИСТРЫ СОСТОЯНИЯ В РЕЖИМЕ МНОЖЕСТВА ВЕДОМЫХ УСТРОЙСТВ

Следующие подразделы описывают дополнительные управляющие регистры и регистры состояния, действительные только если в конфигурации GUI включена опция множества ведомых устройств. Если опция множества ведомых устройств отключена, эти регистры недоступны.

1.3.12.2.7.3.5.1 T X F UNC A D D R / R XFU N C A DDR

ПРИМЕЧАНИЕ: ТРЕБУЕТСЯ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА!

TxFuncAddr и RxFuncAddr – это 7-битовые регистры чтения/записи, в которых хранится адрес целевой функции, доступной через соответствующую оконечную точку (EPn). TxFuncAddr должен быть определен для каждой используемой оконечной точки передатчика; RxFuncAddr должен быть определен для каждой используемой оконечной точки приемника.

Примечание: TxFuncAddr должен быть определен для оконечной точки 0. Регистр RxFuncAddr для оконечной точки 0 не существует.

Адрес: 80+8*nh и 84+8*nh соответственно; Значение после сброса: 7'h00

	D6	...	D0
	Адрес целевой функции		
От CPU	rw	...	rw
От USB	r	...	r

1.3.12.2.7.3.5.2 T X H U B A DD R / R X HUB A D DR

ПРИМЕЧАНИЕ: ТОЛЬКО В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА!

TxFuncAddr и RxFuncAddr – это 8-битовые регистры чтения/записи, которые подобно TxHubPort и RxHubPort должны быть записаны только в том случае, когда устройство с низкой или средней скоростью соединения подключено к оконечной точке передатчика/приемника EPn через высокоскоростной хаб USB 2.0, который выполняет необходимую трансляцию транзакции между передачей с высокой скоростью и передачей с низкой/средней скоростью. В таких обстоятельствах:

младшие 7 бит должны содержать адрес этого хаба USB 2.0

старший бит должен указывать, имеет ли хаб несколько трансляторов транзакций (устанавливается в ‘0’ при одном трансляторе транзакции; устанавливается в ‘1’ при нескольких трансляторов транзакции).

Примечание: Если к хабу подключена оконечная точка 0, то для EP0 должен быть определен TxHubAddr. Регистр RxFuncAddr для оконечной точки 0 не существует.

Адрес: 80+8*nh и 84+8*nh соответственно; Значение после сброса: 8'h00

	D7	D6	...	D0
		Адрес хаба с несколькими трансляторами		
От CPU	rw	rw	...	rw
От USB	r	r	...	r

1.3.12.2.7.3.5.3 TXHUBPORT/RXHUBPORT

ПРИМЕЧАНИЕ: ТОЛЬКО В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА!

TxHubPort и RxHubPort должны быть записаны только в том случае, когда устройство с низкой или средней скоростью соединения подключено к оконечной точке передатчика/приемника EPn через высокоскоростной хаб USB 2.0, который выполняет необходимую трансляцию транзакции. В таком случае эти 7-битовые регистры чтения/записи необходимо использовать для записи порта того хаба USB 2.0, через который осуществляется доступ к целевой функции, ассоциированной с оконечной точкой.

Примечание: TxHubPort должен быть определен для оконечной точки 0. Регстр RxHubPort для оконечной точки 0 не существует.

Адрес: 83+8*nh и 87+8*nh соответственно; Значение после сброса: 7'h00

	D6	...	D0
	Порт хаба		
От CPU	rw	...	rw
От USB	r	...	r

Эта информация вместе с адресом хаба (см. выше) позволяет MUSBMHDRC поддерживать разделенные транзакции.

1.3.12.2.7.3.6 ДОПОЛНИТЕЛЬНЫЕ УПРАВЛЯЮЩИЕ РЕГИСТРЫ И РЕГИСТРЫ СОСТОЯНИЯ

1.3.12.2.7.3.6.1 VCONROL

ПРИМЕЧАНИЕ: ТОЛЬКО ДЛЯ ЗАПИСИ!

VControl – это UTMI+ PHY регистр производителя, который может быть опционально включен в ядро при конфигурации. Его размер также указывается при конфигурации и может составлять до 32 бит. Структуру регистра определяет разработчик системы, хотя пользователи должны иметь в виду, что спецификация UTMI+ определяет 4-битовый регистр VControl.

Регистр может быть доступен по адресу 68h.

Пользователям также следует иметь в виду:

(i) Если указан регистр длиной более 8 бит, любая запись должна производиться в весь регистр, с использованием соответственно 16-битовой или 32-битовой записи. Запись отдельных битов не поддерживается.

(ii) Время ожидания записи (измеренное между передним фронтом CLK в конце AHB цикла записи и положительным фронтом XCLK, когда управляющий регистр UTMI+ PHY VControl загружен) составляет от Hc + 3Xc до Hc + 4Xc, где Hc – это период CLK, а Xc - период XCLK. Таким образом, минимальный интервал между успешными записями в управляющий регистр ядра VControl должен составлять Hc + 4Xc, чтобы значение на искажалось во время синхронизации с частотным доменом XCLK.

1.3.12.2.7.3.6.2 VSTATUS

ПРИМЕЧАНИЕ: ТОЛЬКО ДЛЯ ЧТЕНИЯ!

VStatus – это UTMI+ PHY регистр производителя, который может быть опционально включен в ядро при конфигурации. Его размер также указывается при конфигурации и может составлять до 32 бит. Структуру регистра определяет разработчик системы, хотя пользователи должны иметь в виду, что спецификация UTMI+ определяет 8-битовый регистр VStatus.

Регистр может быть доступен по адресу 68h.

Пользователям следует также иметь в виду:

(i) Входная шина VSTATUS считывается каждые 6 циклов XCLK.

(ii) Время ожидания записи (измеренное между передним фронтом CLK в конце AHB цикла записи и положительным фронтом XCLK, когда управляющий регистр UTMI+ PHY VControl загружен) составляет от 2Hc + Xc до 3Hc + 6Xc, где Hc – это период CLK, а Xc - период XCLK.

1.3.12.2.7.3.6.3 H W V E R S

HWVers – это 16- битовый регистр «только для чтения», хранящий информации о версии RTL, из которой было сгенерировано ядро, в частности, номер версии RTL (vxx.yyy).

Address: 6Ch; Reset value: Зависит от версии

	D15	D14 ... D10	D9 ... D0
	RC	xx	yyy
От CPU	r	r ... r	r ... r
От USB	r	r ... r	r ... r

Бит	Наимен.	Функция
D15	RC	Устанавливается в ‘1’, если RTL устанавливается из предвыпускной версии, а не из полной версии ядра.
D14 – D10	xx	Старшие цифры версии (Диапазон 0 – 31).
D9 – D0	yyy	Младшие цифры версии (Диапазон 0 – 999).

1.3.12.2.7.3.7 ДОПОЛНИТЕЛЬНЫЕ РЕГИСТРЫ КОНФИГУРАЦИИ

1.3.12.2.7.3.7.1 E P I N F O

Этот 8-битовый регистр «только для чтения» позволяет считывает количество оконечных точек передатчика и приемнике, включенных в блок.

Адрес: 78h; Значение после сброса: Зависит от имплементации

	D7	D6	D5	D4	D3	D2	D1	D0
	RxEndPoints				TxEndPoints			
От CPU	r	r	r	r	r	r	r	r
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7 – D4	RxEndPoints	Количество оконечных точек приемника, имплементированных в блок.
D3 – D0	TxEndPoints	Количество оконечных точек передатчика, имплементированных в блок.

1.3.12.2.7.3.7.2 R A MIN FO

Этот 8-битовый регистр «только для чтения» содержит информацию о ширине RAM.

Адрес: 79h; Значение после сброса: Зависит от имплементации

	D7	D6	D5	D4	D3	D2	D1	D0
	DMAChans				RamBits			
От CPU	r	r	r	r	r	r	r	r
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7 – D4	DMAChans	Количество каналов DMA, имплементированных в блок.
D3 – D0	RamBits	Ширина адресной шины RAM.

1.3.12.2.7.3.7.3 L I N K I N FO

Этот 8-битовый регистр позволяет определить некоторые задержки.

Адрес: 7Ah; Значение после сброса: 8'h5C

	D7	D6	D5	D4	D3	D2	D1	D0
	WTCON				WTID			
От CPU	r	r	r	r	r	r	r	r
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7 – D4	WTCON	Устанавливает ожидание для фильтра пользователя подключения/отключения в 533,3 нс блоках. (Установка по умолчанию соответствует 2,667 мкс.)
D3 – D0	WTID	Устанавливает задержку от момента активизации сигнала IDPULLUP до подтверждения сигнала IDDIG в 4,369 мс блоках. (Установка по умолчанию соответствует 52,43 мс.)

1.3.12.2.7.3.7.4 VPLEN

Этот 8-битовый регистр устанавливает длительность импульса накачки питания.

Адрес: 7Bh; Значение после сброса: 8'h3C

	D7	D6	D5	D4	D3	D2	D1	D0
	(msb)			VPLEN				(lsb)
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7 – D0	VPLEN	Устанавливает длительность накачки питания в 546,1 мкс блоках. (Установка по умолчанию соответствует 32,77 мс.)

1.3.12.2.7.3.7.5 H S _EO F 1

Этот 8-битовый регистр устанавливает минимально допустимый временной интервал между началом последней транзакции и EOF для транзакций с высокой скоростью соединения.

Адрес: 7Ch; Значение после сброса: 8'h80

	D7	D6	D5	D4	D3	D2	D1	D0
	(msb)			_EOF1				(lsb)
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7 – D0	HS_EOF 1	Для транзакций с высокой скоростью соединения устанавливает время до EOF, для предотвращения начала новой транзакции, в 133,3 нс блоках. (Установка по умолчанию соответствует 17,07 мкс.)

1.3.12.2.7.3.7.6 FS_EOF1

Этот 8-битовый регистр устанавливает минимально допустимый временной интервал между началом последней транзакции и EOF для транзакций со средней скоростью соединения.

Адрес: 7Dh; Значение после сброса: 8'h77

	D7	D6	D5	D4	D3	D2	D1	D0
	(msb)			FS_EOF1				(lsb)
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7 – D0	FS_EOF 1	Для транзакций со средней скоростью соединения устанавливает время до EOF, для предотвращения начала новой транзакции в 533,3 нс блоках. (Установка по умолчанию соответствует 63,46 мкс.)

1.3.12.2.7.3.7.7 LS_EOF1

Этот 8-битовый регистр устанавливает минимально допустимый временной интервал между началом последней транзакции и EOF для транзакций с низкой скоростью соединения.

Адрес: 7Eh; Значение после сброса: 8'h72

	D7	D6	D5	D4	D3	D2	D1	D0
	(msb)				LS_EOF1			(lsb)
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7 – D0	LS_EOF 1	Для транзакций с низкой скоростью соединения устанавливает время до EOF, для предотвращения начала новой транзакции в 1,067 мкс блоках. (Установка по умолчанию соответствует 121,6 мкс.)

1.3.12.2.7.3.7.8 S O FT _R ST

Этот 8-битовый регистр устанавливает в состояние с низким уровнем выходные сигналы сброса NRSTO и NRSTOX. Этот регистр очищается автоматически и сбрасывается входным сигналом NRST.

Адрес: 7Fh; Значение после сброса: 8'h00

		D1	D0
	(msb) SOFT_RST (lsb)		
От CPU		rw	rw
От USB		r	r

Бит	Наимен.	Функция
D7 – D2	-	Не используется, всегда считывается ноль.
D1	NRSTX	По умолчанию значение этого бита 1’b0. Когда в этот бит записывается 1, выходной сигнал NRSTXO будет активирован (низкий уровень) через минимальную задержку в 7 циклов входного сигнала CLK. Выходной сигнал NRSTXO будет асинхронно активирован и синхронно деактивирован в соответствии с XCLK. Этот регистр очищается автоматически и сбрасывается входным сигналом NRST.
D0	NRST	По умолчанию значение этого бита 1’b0. Когда в этот бит записывается 1, выходной сигнал NRSTO будет активирован (низкий уровень) через минимальную задержку в 7 циклов входного сигнала CLK. Выходной сигнал NRSTO будет асинхронно активирован и синхронно деактивирован в соответствии с XCLK. Этот регистр очищается автоматически и сбрасывается входным сигналом NRST.

1.3.12.2.7.3.8 ДОПОЛНИТЕЛЬНЫЕ РЕГИСТРЫ

Следующие подразделы содержат подробную информацию о дополнительных регистрах, которые управляют работой и влияют на работу MUSBMHDRC.

1.3.12.2.7.3.8.1 RQ PK TCOU NT

ПРИМЕЧАНИЕ: ТОЛЬКО В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА!

Для каждой оконечной точки 1 – 15 приемника MUSBMHDRC предоставляет 16-битовый регистр RqPktCoun. Этот регистр чтения-записи используется в режиме ведущего устройства для указания количества пакетов, которые должны быть переданы в поблочной передаче одного или нескольких пакетов поточной пересылки длины MaxP к оконечной точке приемника с номером n. Ядро использует записанное в этом регистре значение для того, чтобы определить количество запросов, которые необходимо выдать, когда установлена (в регистре RxCSR) опция AutoReq.

Дополнительную информацию см. в разделе 1.3.12.2.7.3.13.1.

Примечание: Несколько пакетов, объединенные в один пакет поточной передачи внутри FIFO, считаются одним пакетом.

Адрес: 300 + 4*n; Значение после сброса: 16'h0000

	D15 ... D0
	RqPktCount
От CPU	rw ... rw
От USB	rw ... rw

Бит	Наимен.	Функция
D15 – D0	RqPktCount	Устанавливает количество пакетов размера MaxP, которые должны быть переданы в поблочной передаче. Используется только в режиме ведущего устройства с установленной опцией AutoReq. Не оказывает влияния в режиме ведомого устройства или с не установленной опцией AutoReq.

1.3.12.2.7.3.8.2 ЗАПРЕТ ДВОЙНОЙ БУФЕРИЗАЦИИ ПАКЕТОВ

Для каждой оконечной точки приемника и передатчика с номерами 1 – 15 MUSBMHDRC предоставляет бит DPktBufDis. Эти биты хранятся в общем наборе регистров DPktBufDis. Один набор регистров ассоциирован с оконечными точками приемника, другой – с оконечными точками передатчика. Эти биты чтения/записи используются для управления двойной буферизацией пакета для каждой оконечной точки. Бит игнорируется при разрешенном динамическом определении размера FIFO. Активированный бит (DPktBufDis равен 1’b1) запретит двойную буферизацию пакета для соответствующей оконечной точки независимо от соотношения размера FIFO буфера оконечнойq точки и размера INMAXP. Неактивированный бит (DPktBufDis равен 1’b0) НЕ обязательно разрешает двойную буферизацию пакета – скорее разрешает включить двойную буферизацию пакета в зависимости от соотношения размера FIFO буфера оконечнойq точки и размера INMAXP. Более подробную информацию см. в разделе 1.3.12.2.7.3.13.

1.3.12.2.7.3.8.2.1 R X D P KT BU FD IS

Rx DPktBufDis – это 16-битовый регистр, который указывает, какая из оконечных точек приемника запретила возможность двойной буферизации пакета, описанную в разделе 1.3.12.2.7.3.13.1.2 спецификации MUSBMHDRC.

Примечание: Биты, относящиеся к точкам, которые не были сконфигурированы, могут быть активированы записью ‘1’ в соответствующий регистр; тем не менее, бит запрета не будет иметь видимого эффекта.

Адрес: 340h; Значение после сброса: 16'h0000

	D15	D14	D13	D12	D11	D10	D9	D8
	EP15 RxDis	EP14 RxDis	EP13 RxDis	EP12 RxDis	EP11 RxDis	EP10 RxDis	EP9 RxDis	EP8 RxDis
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	r	r	r	R	r	r	r	r
	D7	D6	D5	D4	D3	D2	D1	D0
	EP7 RxDis	EP6 RxDis	EP5 RxDis	EP4 RxDis	EP3 RxDis	EP2 RxDis	EP1 RxDis	Unused
От CPU	rw	rw	rw	rw	rw	rw	rw	r
От USB	r	r	r	r	r	r	r	r

Бит	Наим.	Функция
D15	EP15 RxDis	Двойная буферизация пакета запрещена для оконечной точки 15 приемника.
D14	EP14 RxDis	Двойная буферизация пакета запрещена для оконечной точки 14 приемника.
D13	EP13 RxDis	Двойная буферизация пакета запрещена для оконечной точки 13 приемника.
D12	EP12 RxDis	Двойная буферизация пакета запрещена для оконечной точки 12 приемника.
D11	EP11 RxDis	Двойная буферизация пакета запрещена для оконечной точки 11 приемника.
D10	EP10 RxDis	Двойная буферизация пакета запрещена для оконечной точки 10 приемника.
D9	EP9 RxDis	Двойная буферизация пакета запрещена для оконечной точки 9 приемника.
D8	EP8 RxDis	Двойная буферизация пакета запрещена для оконечной точки 8 приемника.
D7	EP7 RxDis	Двойная буферизация пакета запрещена для оконечной точки 7 приемника.
D6	EP6 RxDis	Двойная буферизация пакета запрещена для оконечной точки 6 приемника.
D5	EP5 RxDis	Двойная буферизация пакета запрещена для оконечной точки 5 приемника.
D4	EP4 RxDis	Двойная буферизация пакета запрещена для оконечной точки 4 приемника.
D3	EP3 RxDis	Двойная буферизация пакета запрещена для оконечной точки 3 приемника.
D2	EP2 RxDis	Двойная буферизация пакета запрещена для оконечной точки 2 приемника.
D1	EP1 RxDis	Двойная буферизация пакета запрещена для оконечной точки 1 приемника.
D0	Unused	Зарезервировано

1.3.12.2.7.3.8.2.2 T X D P KT BU FD IS

Tx DPktBufDis – это 16-битовый регистр, который указывает, какая из оконечных точек передатчика запретила возможность двойной буферизации пакета, описанную в разделе 1.3.12.2.7.3.13.1.2 спецификации MUSBMHDRC.

Примечание: Биты, относящиеся к точкам, которые не были сконфигурированы, могут быть активированы записью ‘1’ в соответствующий регистр; тем не менее, бит запрета не будет иметь видимого эффекта.

Адрес: 340h; Значение после сброса: 16'h0000

	D15	D14	D13	D12	D11	D10	D9	D8
	EP15 TxDis	EP14 TxDis	EP13 TxDis	EP12 TxDis	EP11 TxDis	EP10 TxDis	EP9 TxDis	EP8 TxDis
От CPU	rw	rw	Rw	rw	Rw	rw	rw	rw
От USB	r	r	R	r	R	r	r	r
	D7	D6	D5	D4	D3	D2	D1	D0
	EP7 TxDis	EP6 TxDis	EP5 TxDis	EP4 TxDis	EP3 TxDis	EP2 TxDis	EP1 TxDis	Unused
От CPU	rw	rw	rw	rw	Rw	rw	rw	r
От USB	r	r	R	r	R	r	r	r

Бит	Наим.	Функция
D15	EP15 TxDis	Двойная буферизация пакета запрещена для оконечной точки 15 передатчика.
D14	EP14 TxDis	Двойная буферизация пакета запрещена для оконечной точки 14 передатчика.
D13	EP13 TxDis	Двойная буферизация пакета запрещена для оконечной точки 13 передатчика.
D12	EP12 TxDis	Двойная буферизация пакета запрещена для оконечной точки 12 передатчика.
D11	EP11 TxDis	Двойная буферизация пакета запрещена для оконечной точки 11 передатчика.
D10	EP10 TxDis	Двойная буферизация пакета запрещена для оконечной точки 10 передатчика.
D9	EP9 TxDis	Двойная буферизация пакета запрещена для оконечной точки 9 передатчика.
D8	EP8 TxDis	Двойная буферизация пакета запрещена для оконечной точки 8 передатчика.
D7	EP7 TxDis	Двойная буферизация пакета запрещена для оконечной точки 7 передатчика.
D6	EP6 TxDis	Двойная буферизация пакета запрещена для оконечной точки 6 передатчика.
D5	EP5 TxDis	Двойная буферизация пакета запрещена для оконечной точки 5 передатчика.
D4	EP4 TxDis	Двойная буферизация пакета запрещена для оконечной точки 4 передатчика.
D3	EP3 TxDis	Двойная буферизация пакета запрещена для оконечной точки 3 передатчика.
D2	EP2 TxDis	Двойная буферизация пакета запрещена для оконечной точки 2 передатчика.
D1	EP1 TxDis	Двойная буферизация пакета запрещена для оконечной точки 1 передатчика.
D0	Unused	Зарезервировано

1.3.12.2.7.3.8.3 C_T_U CH

Этот регистр устанавливает время ожидания ЛЧМ-сигнала. Значение, умноженное на 4, определяет количество циклов XCLK до окончания ожидания. То есть, если частота XCLK составляет 30 Мгц, это значение определяет количество интервалов длительностью 133 нс до окончания ожидания. При частоте 60 Мгц это значение определяет количество интервалов длительностью 67 нс до окончания ожидания. Этот бит записывается ведущим устройством в частотном домене CLK, но счетчик использует данное значение в частотном домене XCLK. Время пересечения границы домена не установлено, поскольку значение регистра статично. Значение по умолчанию – это значение директивы компилятора с этим же именем в файле конфигурации musbmhdrc_cfg.v.

Адрес: 344h; Значение после сброса: Разное.

	D15	D14	D13	D12	D11	D10	D9	D8
C_T_UCH[15:8]								
От CPU	rw	rw	Rw	rw	Rw	rw	rw	rw
От USB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	D7	D6	D5	D4	D3	D2	D1	D0
C_T_UCH[7:0]								
От CPU	rw	rw	Rw	rw	Rw	rw	rw	rw
От USB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Бит	Наим.	Функция
D15-D0	C_T_UCH	Настраиваемый таймер время ожидания ЛЧМ-сигнала. Значение по умолчанию определяется директивой компилятора с этим же именем в файле конфигурации musbmhdrc_cfg.v. Значение по умолчанию 203Ah, если ширина данных PHY ведущего устройства равна 16 бит (XCLK 30МГц), и 4074h, если ширина данных PHY ведущего устройства равна 8 бит (XCLK 60МГц), что соответствует задержке 1,1 мс.

1.3.12.2.7.3.8.4 C_T_HH S R T N

Этот регистр устанавливает задержку после окончания сигнала выхода из спящего режима при высокой скорости соединения (ведущее устройство) до разрешения стандартного режима работы UTM. Значение, умноженное на 4, определяет количество циклов XCLK до окончания ожидания. То есть, если частота XCLK составляет 30 Мгц, это значение определяет количество

интервалов длительностью 33,3 нс до окончания задержки. При частоте 60 Мгц это значение определяет количество интервалов длительностью 16,7 нс до окончания задержки. Этот бит записывается ведущим устройством в частотном домене CLK, но счетчик использует данное значение в частотном домене XCLK. Время пересечения границы домена не установлено, поскольку значение регистра статично. Значение по умолчанию – это значение директивы компилятора с этим же именем в файле конфигурации musbmhdrc_cfg.v.

Адрес: 346h; Значение после сброса: Разное.

	D15	D14	D13	D12	D11	D10	D9	D8
C_T_HHSRTN[15:8]								
От CPU	rw	rw	Rw	rw	Rw	rw	rw	rw
От USB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	D7	D6	D5	D4	D3	D2	D1	D0
C_T_HHSRTN[7:0]								
От CPU	rw	rw	Rw	rw	Rw	rw	rw	rw
От USB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Бит	Наим.	Функция
D15-D0	C_T_HHSRTN	Задержка после окончания сигнала выхода из спящего режима при высокой скорости соединения до разрешения стандартного режима работы UTM. Значение по умолчанию определяется директивой компилятора с этим же именем в файле конфигурации musbhsfc_xcfg.v. Значение по умолчанию 2F3h, если ширина данных PHY ведущего устройства равна 16 бит (XCLK 30МГц), и 5E6h, если ширина данных PHY ведущего устройства равна 8 бит (XCLK 60МГц), что соответствует задержке 100 мкс.

1.3.12.2.7.3.8.5 C_T_H S BT

Согласно спецификации USB 2.0, раздел 7.1.19.2, ведущее устройство с высокой скоростью соединения или устройство, ожидающее отклика на пересылку, не должно прерывать транзакцию, если задержка между пакетами меньше 736 интервалов передачи бита, и должен прерывать транзакцию, если не получено отклика по истечении 816 интервалов передачи бита. Регистр указывает на величину, которую следует прибавить к минимальному периоду ожидания при высокой скорости соединения, то есть, 736 интервалов передачи бита. Время ожидания может увеличиваться дискретно по 64 интервала передачи бита при высокой скорости соединения (133 нс). Имеется 16 возможных значений. По умолчанию прибавляется 0, то есть, устанавливается минимальное время ожидания при высокой скорости соединения. Использование этого регистра позволяет установить время ожидания при высокой скорости соединения больше, чем максимум, указанный в спецификации USB 2.0, что делает MUSBMHDRC несовместимым со стандартом.

Адрес: 348h; Значение после сброса: 4’b0000

	D3	D2	D1	D0
	(msb) HS Timeout Adder (lsb)			
От CPU	rw	rw	rw	rw
От USB	r	r	r	r

Бит	Наимен.	Функция
D3-D0	C_T_HSBT	Величина прибавляется к минимальному периоду ожидания при высокой скорости соединения (736 интервалов передачи бита) дискретно по 64 интервала передачи бита при высокой скорости соединения. Это позволяет установить 16 возможных значений общего времени ожидания:

Значение в регистре	Общее время ожидания при высокой скорости соединения (интервалов передачи бита)	Общее время ожидания при высокой скорости соединения (мкс)
0	736	1.534
1	800	1.667
2	864	1.801
3	928	1.934
4	992	2.067
5	1056	2.201
6	1120	2.334
7	1184	2.467
8	1248	2.601
9	1312	2.734
10	1376	2.868
11	1440	3.001
12	1504	3.134
13	1568	3.268

1.3.12.2.7.3.9 РЕГИСТРЫ DMA

Регистры DMA доступны только в том случае, если MUSBMHDRC сконфигурирован для использования, как минимум, одного канала DMA. Ниже приведен один набор регистров на канал.

1.3.12.2.7.3.9.1 D M A_INT R

Этот регистр обеспечивает прерывание для каждого канала DMA. Регистр прерываний очищается при чтении. При установке любого бита этого регистра активизируется (низкий уровень) выходной сигнал DMA_NINT. События, вызывающие прерывания, описаны в разделе 17 (Опционный контроллер DMA.) Биты этого регистра могут устанавливаться только при установленном бите разрешения прерывания для соответствующего канала (регистр DMA_CNTL.D3).

Адрес: 200h; Значение после сброса: 00h

	D7	D6	D5	D4	D3	D2	D1	D0
DMA_INTR[7:0]								
От CPU	rw	rw	w	rw	rw	r	r	R
От USB	set	set	set	set	set	set	set	set

Бит	Наимен.	Функция
D7	CH8 DMA_INTR	Прерывание канала 8 DMA
D6	CH7 DMA_INTR	Прерывание канала 7 DMA
D5	CH6 DMA_INTR	Прерывание канала 6 DMA
D4	CH5 DMA_INTR	Прерывание канала 5 DMA
D3	CH4 DMA_INTR	Прерывание канала 4 DMA
D2	CH3 DMA_INTR	Прерывание канала 3 DMA
D1	CH2 DMA_INTR	Прерывание канала 2 DMA
D0	CH1 DMA_INTR	Прерывание канала 1 DMA

1.3.12.2.7.3.9.2 D M A_CNTL

Этот регистр доступен только в том случае, если MUSBMHDRC сконфигурирован для использования, как минимум, одного канала DMA. Этот регистр обеспечивает управление передачей для каждого канала DMA. Регистр управляет разрешением передачи, направлением передачи, режимом передачи и пакетными режимами.

Адрес: 204h + (n-1)*10h;n=номер канала от 1 до 8; Значение после сброса: 00h

						D10	D9	D8
						DMA_BRSTM		DMA_ERR
От CPU						rw	rw	rw
От USB						r	r	rw
	D7	D6	D5	D4	D3	D2	D1	D0
	DMAEP				DMAIE	DMAMODE	DMA_DIR	DMA_EN
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D10-D9	DMA_BRSTM	Пакетный режим 00 = Пакетный режим 0: Пакеты неопределенной длины 01 = Пакетный режим 1: INCR4 или неопределенная длина 10 = Пакетный режим 2: INCR8, INCR4 или неопределенная длина 11 = Пакетный режим 3: INCR16, INCR8, INCR4 или неопределенная длина
D8	DMA_ERR	Бит ошибки шины. Указывает, что обнаружена ошибка шины на входном сигнале AHB_HRESPM[1:0]. Этот бит сбрасывается программой.
D7-D4	DMAEP	Номер оконечной точки, с которым ассоциирован канал.
D3	DMAIE	Разрешение прерывания DMA.
D2	DMAMODE	Этот бит выбирает режим передачи DMA. 0 = режим 0 передачи DMA 1 = режим 1 передачи DMA
D1	DMA_DIR	Этот бит выбирает направление передачи DMA. 0 = запись DMA (оконечная точка приемника) 1 = чтение DMA (оконечная точка передатчика)
D0	DMA_ENAB	Этот бит разрешает передачу DMA и инициирует начало передачи.

1.3.12.2.7.3.9.3 D M A_A D D R

Этот регистр указывает текущий адрес памяти соответствующего канала DMA. Начальный адрес памяти, записанный в этот регистр, должен иметь такое значение, чтобы его величина по модулю 4 равнялось 0. То есть, DMA_ADDR[1:0] должен быть равен 2’b00. Два младших бита

этого регистра предназначены только для чтения и не могут быть установлены программой. При пересылке DMA адрес памяти увеличивается по мере передачи байт.

Адрес: 208h + (n-1)*10h;n=номер канала от 1 до 8; Значение после сброса: 00h

	D31	D30	D29	D28	D27	D26	D25	D24
DMA_ADDR[31:24]								
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	rw	rw	rw	rw	rw	rw	rw	rw
	D23	D22	D21	D20	D19	D18	D17	D16
DMA_ADDR[23:16]								
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	rw	rw	rw	rw	rw	rw	rw	rw
	D15	D14	D13	D12	D11	D10	D9	D8
DMA_ADDR[15:8]								
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	rw	rw	rw	rw	rw	rw	rw	rw
	D7	D6	D5	D4	D3	D2	D1	D0
DMA_ADDR[7:0]								
От CPU	rw	rw	rw	rw	rw	rw	r	r
От USB	rw	rw	rw	rw	rw	rw	rw	rw

Бит	Наимен.	Функция
D31 – D0	DMA_ADDR	Адрес памяти DMA. Обратите внимание, что начальный адрес, записанный в этот регистр, должен иметь значение, которое по модулю 4 равно 0. То есть, DMA_ADDR[1:0] должен быть равен 2'b00. Два младших бита этого регистра предназначены только для чтения и не могут быть установлены программой.

1.3.12.2.7.3.9.4 . D M A_COUN T

Этот регистр содержит текущий счетчик DMA для данной пересылки. Программа задает начальное значение счетчика, которое определяет длину всей пересылки. При пересылке каждого байта значение этого счетчика уменьшается.

Адрес: 20Ch + (n-1)*10h;n=номер канала от 1 до 8; Значение после сброса: 00h

	D31	D30	D29	D28	D27	D26	D25	D24
DMA_COUNT[31:24]								
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	rw	rw	rw	rw	rw	rw	rw	rw
	D23	D22	D21	D20	D19	D18	D17	D16
DMA_COUNT[23:16]								
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	rw	rw	rw	rw	rw	rw	rw	rw
	D15	D14	D13	D12	D11	D10	D9	D8
DMA_COUNT[15:8]								
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	rw	rw	rw	rw	rw	rw	rw	rw
	D7	D6	D5	D4	D3	D2	D1	D0
DMA_COUNT[7:0]								
От CPU	rw	rw	rw	rw	rw	rw	rw	rw
От USB	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	Function
D31 – D0	DMA_COUNT	Адрес памяти DMA для соответствующего канала DMA. Примечание: если DMA разрешен при значении счетчика, равном 0, шина не будет запрошена, и сгенерируется прерывание DMA.

1.3.12.2.7.3.10 РЕГИСТРЫ ДИНАМИЧЕСКОГО FIFO

Регистры динамического FIFO доступны лишь при условии, что MUSBMHDRC сконфигурирован для использования функции динамического определения размера FIFO. Для каждой оконечной точки, за исключением оконечной точки 0, имеется набор регистров. Это индексные регистры, и поэтому для доступа к ним должен быть установлен регистр INDEX по адресу 0Eh для соответствующие оконечной точки. Ограничения и использование регистров динамического FIFO описаны в разделе 19.

1.3.12.2.7.3.10.1 T X F I FO SZ

TxFIFOsz – это 5-битовый регистр, который управляет размером FIFO выбранной оконечной точки передатчика.

Адрес: 62h; Значение после сброса: 5’h00

	D4	D3	D2	D1	D0
	DPB	SZ3	SZ2	SZ1	SZ0
От CPU	rw	rw	rw	rw	rw
От USB	r	r	r	r	r

Бит	Наимен.	Функция
D4	DPB	Определяет поддержку двойной буферизации пакета. При ‘1’ поддерживается двойная буферизация пакета. При ‘0’ поддерживается только буферизация одиночного пакета.
D3 – D0	SZ[3:0]	Максимальный допустимый размер пакета (перед любым разделением внутри FIFO пакетов поточной пересылки или пересылки с высокой скоростью соединения до их передачи – см. разделы 8.4.1.3, 8.4.1.4 и 8.5.3)

SZ[3:0]				Размен пакета (Байты)
0	0	0	0	8
0	0	0	1	16
0	0	1	0	32
0	0	1	1	64
0	1	0	0	128
0	1	0	1	256
0	1	1	0	512
0	1	1	1	1024
1	0	0	0	2048
1	0	0	1	4096

Если DPB = 0, FIFO имеет указанный размер; если DPB = 1, FIFO имеет удвоенный размер.

1.3.12.2.7.3.10.2 R X F I FO SZ

RxFIFOsz – это 5-битовый регистр, который управляет размером FIFO выбранной оконечной точки приемника.

Адрес: 63h; Значение после сброса: 5’h00

	D4	D3	D2	D1	D0
	DPB	SZ3	SZ2	SZ1	SZ0
От CPU	rw	rw	rw	rw	rw
От USB	r	r	r	r	r

Бит	Наимен.	Функция
D4	DPB	Определяет поддержку двойной буферизации пакета. При ‘1’ поддерживается двойной буферизации пакета. При ‘0’ поддерживается только буферизация одиночного пакета.
D3 – D0	SZ[3:0]	Максимальный допустимый размер пакета (после любого объединения внутри FIFO пакетов поточной пересылки или пересылки с высокой скоростью соединения после их получения – см. разделы 8.4.1.3, 8.4.1.4 и 8.5.3)

SZ[3:0]				Размен пакета (Байты)
0	0	0	0	8
0	0	0	1	16
0	0	1	0	32
0	0	1	1	64
0	1	0	0	128
0	1	0	1	256
0	1	1	0	512
0	1	1	1	1024
1	0	0	0	2048
1	0	0	1	4096

Если DPB = 0, FIFO имеет указанный размер; если DPB = 1, FIFO имеет удвоенный размер.

1.3.12.2.7.3.10.3 T X F I FO A D D

TxFIFOadd – это 14-битовый регистр, который управляет начальным адресом FIFO выбранной оконечной точки передатчика.

Адрес: 64h; Значение после сброса: 14’h0000

	D13	D12 ... D0
		AD[12:0]
От CPU	rw	rw ... rw
От USB	r	r ... r

Бит	Наимен.	Функция
D13	–	Зарезервировано для будущего использования.
D12 – D0	AD[12:0]	Начальный адрес FIFO оконечной точки в блоках по 8 байт:

AD[12:0]				Start Address
0	0	0	0	0000
0	0	0	1	0008
0	0	0	2	0010
…				…
1	F	F	F	FFF8

1.3.12.2.7.3.10.4 R X F I FOA D D

RxFIFOadd – это 14-битовый регистр, который управляет начальным адресом FIFO выбранной оконечной точки приемника.

Адрес: 66h; Значение после сброса: 14’h0000

	D13	D12 ... D0
		AD[12:0]
От CPU	rw	rw ... rw
От USB	r	r ... r

Бит	Наимен.	Функция					
D13	–	Зарезервировано для будущего использования.					
D12 – D0	AD[12:0]	Начальный адрес FIFO оконечной точки в блоках по 8 байт:					
			AD[12:0]				Start Address
			0	0	0	0	0000
			0	0	0	1	0008
			0	0	0	2	0010
			…				…
			1	F	F	F	FFF8

1.3.12.2.7.3.11 РЕГИСТРЫ УПРАВЛЕНИЯ ПИТАНИЕМ (LPM)

1.3.12.2.7.3.11.1 L P M_ A TTR

Этот регистр используется для определения атрибутов транзакции LPM и режима сна. В режиме ведущего и в режиме ведомого устройства значение регистра одинаково, но источник данных разный:

В режиме ведомого устройства:

В режиме ведомого устройства значения в этом регистре будут содержать эквиваленты атрибутов, которые были получены в последней принятой транзакции LPM. Этот регистр дополняется содержимым пакета LPM, если ответом на транзакцию был пакет ACK. Этот регистр может изменяться программой. Во всех других случаях этот регистр будет сохранять текущее значение.

В режиме ведущего устройства:

В режиме ведущего устройства программа установит значения в этом регистре, чтобы определить следующую транзакцию LPM, которая будет передана. Эти значения будут включены в содержимое следующей транзакции LPM.

Адрес: 360h; Значение после сброса: 00h

	D15	D14	D13	D12	D11	D10	D9	D8
	EndPoint				Reserved			RmtWak
От CPU	r	r	r	r	r	r	r	r
От USB	r	r	r	r	rw	r	r	rw
	D7	D6	D5	D4	D3	D2	D1	D0
	HIRD				LinkState			
От CPU	r	r	r	r	r	r	r	r
От USB	rw	rw	rw	rw	rw	rw	rw	rw

Бит	Наимен.	Функция
D15-D12	EndPnt	Это EndPnt, содержащаяся в пакете маркера транзакции LPM.
D11 – D9	Reserved	Зарезервировано; всегда считывается 0.
D8	RmtWak	Это бит разрешения удаленной активизации. RmtWak = 1’b0: удаленная активизация запрещена. RmtWak = 1’b1: удаленная активизация разрешена. Этот бит применим только временно и только к текущему спящему режиму. Установленная возможность удаленной активизации будет действовать после текущего цикла спящего режима.
D7-D4	HIRD	Это длительность сигнала выхода из спящего режима, инициированного ведущим устройством. Эта величина определяет минимальное время, которое управляющее устройство будет поддерживать на шине сигнал выхода из спящего режима. Значение регистра преобразуется в длительность временного интервала следующим образом: Длительность сигнала выхода из спящего режима = 50 мкс + HIRD*75 мкс. Результат лежит в диапазоне 50 мкс до 1200 мкс.
D3-D0	LinkState	Это значение, устанавливаемое ведущим устройством, чтобы указать ведомому устройству, в какое состояние оно должно перейти после приема и подтверждения транзакции LPM. LinkState = 4’h0 – Зарезервировано LinkState = 4’h1 – Спящий режим (L1) LinkState = 4’h2 – Зарезервировано LinkState = 4’h3 – Зарезервировано

1.3.12.2.7.3.11.2 L P M_C N TRL

В режиме ведомого устройства:

Адрес: 362h; Значение после сброса: 00h

	D7	D6	D5	D4	D3	D2	D1	D0
				LPMNAK	LPMEN		LPMRES	LPMXMT
От CPU	r	r	r	r	rw	rw	rw	rw
От USB	r	r	r	r	r	r	clear	clear

Бит	Наименов.	Функция
D7 – D5	Reserved	Зарезервировано – всегда считывается 0x00.
D4	LPMNAK	Этот бит используется для перевода всех оконечных точек в такое состояние, чтобы ответом на все транзакции, отличные от транзакций LPM, был пакет NAK. Этот бит будет оказывать воздействие только после того, как MUSBMHDRC был переведен в спящий режим посредством LPM. В данном случае MUSBMHDRC продолжит отправку ответных пакетов NAK, пока этот бит не будет сброшен программой.
D3 – D2	LPMEN	Этот регистр используется для разрешения LPM в MUSBMHDRC. Различают три уровня разрешения LPM, что определяет ответ MUSBMHDRC на транзакции LPM. Эти уровни: 2'b00, 2'b10 – Не поддерживаются транзакции LPM и расширенной (Extended) транзакции. В данном случае MUSBMHDRC не будет отвечать на транзакции LPM, и транзакции будут прерваны. 2'b01 – LPM не поддерживается, но расширенные транзакции поддерживаются. В данном случае MUSBMHDRC будет отвечать на транзакцию LPM пакетом STALL. 2'b11 – MUSBMHDRC поддерживает LPM и расширенные транзакции. В данном случае MUSBMHDRC будет отвечать пакетом NYET или ACK, в соответствии со значением LPMXMT и другими условиями.
D1	LPMRES	Этот бит используется для инициирования выхода из спящего режима (дистанционная активизация). Этот бит отличается от стандартного бита RESUME в регистре POWER (адрес 0x01.2) тем, что время подачи сигнала RESUME управляется аппаратной частью. Когда программа записывает этот бит, сигнал выхода из спящего режима будет активен в течение 50 мкс. Этот бит сбрасывается автоматически.
D0	LPMXMT	Этот бит устанавливается программой, чтобы указать MUSBMHDRC на переход в состояние L1 после приема следующей транзакции LPM. Этот бит действует только если LPMEN установлен в состояние 2'b11. Этот бит может быть установлен в том же цикле, что и LPMEN. Если этот бит установлен в 1'b1 и LPMEN = 2'b11, то MUSBMHDRC может отвечать следующим образом: Если нет ожидающих передачи данных (все FIFO передатчика пусты), MUSBMHDRC ответит пакетом ACK. В данном случае этот бит автоматически сбросится, и будет сгенерировано программное прерывание. Если данные ожидают передачи (данные присутствуют по крайней мере в одном FIFO передатчика), MUSBMHDRC ответит пакетом NYET. В данном случае этот бит автоматически НЕ сбросится, но программное прерывание будет сгенерировано.

В режиме ведущего устройства:

Адрес: 362h; Значение после сброса: 00h

	D7	D6	D5	D4	D3	D2	D1	D0
	Reserved						LPMRES	LPMXMT
От CPU	r	r	r	r	r	r	rw	rw
От USB	r	r	r	r	r	r	rw	rw

Бит	Наименов.	Функция
D7 – D2	Reserved	Зарезервировано – всегда считывается 0x0.
D1	LPMRES	Этот бит используется для инициирования выхода из состояния L1. Этот бит отличается от стандартного бита RESUME в регистре POWER (адрес 0x01.2) тем, что время подачи сигнала RESUME управляется аппаратной частью. Когда программа записывает этот бит, сигнал выхода из состояния L1 будет активен в течение времени, определяемого полем HIRD в регистре LPM_ATTR. Этот бит сбрасывается автоматически.
D0	LPMXMT	Программа должна установить этот бит для передачи транзакции LPM. Этот бит сбрасывается автоматически. Этот бит будет немедленно сброшен после получения любого маркера или истечения трех периодов ожидания.

1.3.12.2.7.3.11.3 L P M_INT R E N

LPM_INTREN – это 6-битовый регистр, который содержит биты разрешения для прерываний в регистре LPM_INTR. Если бит в этом регистре установлен в 1, то сигнал MC_NINT будет активирован (низкий уровень), когда установлено соответствующее прерывание в регистре LPM_INTR. Если бит в этом регистре установлен в 0, соответствующий бит в регистре LPM_INTR тоже устанавливается, но сигнал MC_NINT не будет активирован (низкий уровень). При сбросе все биты этого регистра устанавливаются в 0.

Адрес: 363h; Значение после сброса: 00h

	D7	D6	D5	D4	D3	D2	D1	D0
	RESERVED		LPMERREN	LPMRESEN	LPMACKEN	LPMNYEN	LPMSTEN	LPMTOEN
От CPU	r	r	r	r	rw	rw	rw	rw
От USB	r	r	r	r	r	r	r	r

Бит	Наименов.	Функция
D7-D6	Reserved	Зарезервировано – всегда считывается 0x0.
D5	LPMERREN	1’b0 : Запретить прерывание LPMERR 1’b1 : Разрешить прерывание LPMERR
D4	LPMRESEN	1’b0 : Запретить прерывание LPMRES 1’b1 : Разрешить прерывание LPMRES
D3	LPMNCEN	1’b0 : Запретить прерывание LPMNC 1’b1 : Разрешить прерывание LPMNC
D2	LPMACKEN	1’b0 : Запретить прерывание LPMACK 1’b1 : Разрешить прерывание LPMACK
D1	LPMNYEN	1’b0 : Запретить прерывание LPMNY 1’b1 : Разрешить прерывание LPMNY
D0	LPMSTEN	1’b0 : Запретить прерывание LPMST 1’b1 : Разрешить прерывание LPMST

1.3.12.2.7.3.11.4 L P M_INT R

LPM_INTR - это 7-битовый регистр, который хранит состояние системы управления питанием (LPM).

В режиме ведомого устройства:

Адрес: 364h; Значение после сброса: 00h

	D7	D6	D5	D4	D3	D2	D1	D0
	RESERVED		LPMERR	LPMRES	LPMNC	LPMACK	LPMNY	LPMST
От CPU	r	r	r	r	r	r	r	r
От USB	r	r	set	set	set	set	set	set

Бит	Наименов.	Функция
D7-D6	Reserved	Зарезервировано – всегда считывается 0x0.
D5	LkPMERR	Этот бит устанавливается, если в принятой транзакции присутствует поле LinkState, которое не поддерживается. В данном случае ответом на транзакцию будет пакет STALL. Тем не менее, регистр LPM_ATTR будет изменен, чтобы программа могла отметить не соответствующее стандарту содержимое пакета LPM.
D4	LPMRES	Этот бит устанавливается, если MUSBMHDRC было выведено из спящего режима - по любой причине. Этот бит взаимоисключающий с битом RESUME в регистре Power (адрес 0x01).
D3	LPMNC	Этот бит устанавливается, если транзакция LPM получена, и MUSBMDHRC отвечает пакетом NYET вследствие ожидающих в FIFO приемника данных. Это может произойти только при следующем условии: Поле LPMRESP в регистре LMRESP установлено в значение 2’b11, поле LPMXMT установлено в значение 1’b1, и данные ожидают в TX FIFO буферах передатчиков MUSBMHDRC.
D2	LPMACK	Этот бит устанавливается, если транзакция LPM получена, и MUSBMDHRC отвечает пакетом ACK. Это может произойти только при следующем условии: Поле LPMRESP в регистре LMRESP установлено в значение 2’b11, поле LPMXMT установлено в значение 1’b1, и в FIFO буферах передатчиков MUSBMHDRC нет ожидающих данных.
D1	LPMNY	Этот бит устанавливается, если транзакция LPM получена, и MUSBMDHRC отвечает пакетом NYET. Это может произойти только при следующем условии: Поле LPMRESP в регистре LMRESP установлено в значение 2’b11, поле LPMXMT установлено в значение 1’b0.
D0	LPMST	Этот бит устанавливается, если транзакция LPM получена, и MUSBMDHRC отвечает пакетом STALL. Это может произойти только при следующем условии: Поле LPMRESP в регистре LMRESP установлено в значение 2’b01.

В режиме ведущего устройства:

Адрес: 364h; Значение после сброса: 00h

	D7	D6	D5	D4	D3	D2	D1	D0
	RESERVED			LPMRES	LPMNC	LPMACK	LPMNY	LPMST
От CPU	r	r	r	r	r	r	r	r
От USB	r	r	r	set	set	set	set	set

Бит	Наименов.	Функция
D7-D6	Reserved	Зарезервировано – всегда считывается 0x0.
D5	LPMERR	Этот бит устанавливается, если транзакция LPM получена с ошибкой Bit Stuff или с ошибкой PID. В данном случае отсутствует переход в спящий режим, и состояние устройства неизвестно.
D4	LPMRES	Этот бит устанавливается, если MUSBMHDRC было выведено из спящего режима - по любой причине. Этот бит взаимоисключающий с битом RESUME в регистре Power (адрес 0x01).
D3	LPMNC	Этот бит устанавливается, если транзакция LPM передана, но не завершена. Транзакция не выполнена из-за истечения времени ожидания или наличия ошибок в разрядах в ответе во всех трех попытках.
D2	LPMACK	Этот бит устанавливается, если транзакция LPM передана, и устройство отвечает пакетом ACK.
D1	LPMNY	Этот бит устанавливается, если транзакция LPM передана, и устройство отвечает пакетом NYET.
D0	LPMST	Этот бит устанавливается, если транзакция LPM передана, и устройство отвечает пакетом STALL.

1.3.12.2.7.3.11.5 L P M_F A DDR

Действительно только в режиме ведущего устройства

LPM_FADDR – это адрес функции, который будет включен в содержимое транзакции LPM.

Адрес: 365h; Значение после сброса: 8’h00

	D7	D6	D5	D4	D3	D2	D1	D0
	0	Адрес функции						
От CPU	r	rw	rw	rw	rw	rw	rw	rw
От USB	r	r	r	r	r	r	r	r

Бит	Наимен.	Функция
D7	-	Не используется, всегда считывается 0.
D6 – D0	LPM FADDR	Адрес функции LPM.

ПРОГРАММНОЕ ПОДКЛЮЧЕНИЕ/ОТКЛЮЧЕНИЕ

ПРИМЕЧАНИЕ: ТОЛЬКО В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА!

MUSBMHDRC может разрешить, чтобы его подключение к USB управлялось программой

Когда MUSBMHDRC работает в режиме ведомого устройства, UTMI+-соответствующее PHY, используемое вместе с MUSBMHDRC, может переключаться между стандартным режимом и режимом запрета передачи посредством установки/сброса бита 6 регистра Power (бит Soft Conn). Когда бит Soft Conn устанавливается в 1, PHY переходит в стандартный режим, и линии D+/D- шины USB разрешены. Когда функция программного подключения/отключения включена, и бит Soft Conn сбрасывается в 0, PHY переходит в режим запрета передачи (OPMODE[1:0] установлен в 01b), и линии D+ и D- находятся в третьем состоянии. MUSBMHDRC будет отключен от остальных устройств на шине USB.

После аппаратного сброса (NRST = 0) Soft Conn сбрасывается в 0. Таким образом MUSBMHDRC будет отключен, пока программа не установит бит Soft Conn в 1. То есть, прикладная программа может выбирать, когда переводить PHY в стандартный режим. Системы с продолжительной процедурой инициализации могут использовать эту возможность, чтобы убедиться, что инициализация завершена, и система готова к работе, прежде чем подключаться к USB.

1.3.12.2.7.3.12 РАБОТА В КАЧЕСТВЕ ВЕДУЩЕГО ИЛИ ВЕДОМОГО УСТРОЙСТВА

MUSBMHDRC может использоваться в разных системах. Он может использоваться в качестве «периферийного» устройства с высокой или средней скоростью соединения, подключенного к обычному ведущему устройству USB (например, персональному компьютеру). Он может использоваться в качестве ведущего или ведомого устройства при обмене точка-точка с другим «периферийным» устройством – или, если другое устройство также содержит Dual-Role контроллер, два устройства могут при необходимости меняться ролями. (Это второе устройство может иметь функцию USB с высокой, средней или низкой скоростью соединения.) MUSBMHDRC может также использоваться в качестве ведущего для нескольких периферийных устройств в системе с множеством ведомых устройств.

Во всех случаях между MUSBMHDRC и устройствами, с которыми он соединен, поддерживаются управляющие, поточные, изохронные транзакции, а также транзакции по прерыванию.

Режим работы MUSBMHDRC, в роли ведущего или ведомого устройства, зависит от способа соединения устройств. Каждый кабель USB имеет конец ‘A’ и конец ‘B’. Если конец ‘A’ подключен к устройству, содержащему MUSBMHDRC, то MUSBMHDRC возьмет на себя роль ведущего устройства и перейдет в режим «ведущего» (бит Host Mode (DevCtl.D2) будет установлен в ‘1’). Если подключен конец ‘B’ кабеля, то MUSBMHDRC перейдет в режим «ведомого», и бит Host Mode будет установлен в ‘0’.

Если MUSBMHDRC подключен к одному устройству, и это устройство содержит Dual-Role контроллер, то для того, чтобы два устройства менялись ролями, могут использоваться сигналы – без необходимости менять подключение кабеля к устройствам. Условия, при которых MUSBMHDRC может переключаться из режима ведомого устройства в режим ведущего устройства и наоборот, описаны в разделе 15: Согласование ведущего устройства.

Примечание: Возможность MUSBMHDRC работать с множеством ведомых устройств связана с рядом регистров, в которых записано распределение функций устройства по оконечным точкам ядра MUSBMHDRC, а также функциональные характеристики устройства, такие как количество оконечных точек, рабочая скорость и тип транзакций между оконечными точками. Эти регистры предназначены, в основном, для работы ядра в качестве ведущего для нескольких ведомых устройств, однако их также необходимо устанавливать, когда ядро используется в качестве ведущего для одного целевого устройства.

1.3.12.2.7.3.13 РАБОТА В КАЧЕСТВЕ ВЕДОМОГО УСТРОЙСТВА

Когда бит Host Mode (DevCtl.D2) сброшен, MUSBMHDRC работает как ведомое устройство.

В данном разделе рассматриваются действия ядра в отношении оконечных точек передатчика и оконечных точек приемника, вход в спящий режим и выход из него, а также распознавание начала фрейма, когда MUSBMHDRC используется в качестве ведомого устройства.

Условия, при которых MUSBMHDRC работает в режиме ведомого устройства, описаны в разделе 15: Согласование ведущего устройства.

1.3.12.2.7.3.13.1 ОБРАБОТКА ТРАНЗАКЦИИ ЧТЕНИЯ В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Когда MUSBMHDRC работает в режиме ведомого устройства, данные для транзакции чтения обрабатываются с помощью FIFO буферов передатчиков MUSBMHDRC.

Размеры FIFO буферов передатчиков для оконечных точек от 1 до 15 определяются либо константами конфигурации в файле конфигурации MUSBMHDRC, либо, когда выбрано динамическое определение размера FIFO, при помощи регистра TxFIFO2. Максимальный размер пакета данных, который может быть помещен для передачи в FIFO буфер оконечной точки передатчика, задается программой и определяется значением, записанным в регистр TxMaxP для данной оконечной точки (максимальное содержимое □□ количество транзакций/(микрофреймов (где применимо)).

Если динамическое определение размера FIFO не используется, то когда максимальный размер пакета установлен меньшим или равным половине размера FIFO, для транзакций чтения разрешена двойная буферизация пакета, а когда максимальный размер пакета больше половины размера FIFO, разрешена одиночная буферизация пакета. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакета является частью спецификации для FIFO оконечной точки – см. раздел 19). Когда разрешена двойная буферизация пакета, в FIFO могут буферироваться два пакета данных; когда разрешена одиночная буферизация, в этом случае может буферироваться только один пакет, даже если его размер меньше половины размера FIFO.

Примечание: Максимальный размен пакета, установленный для каждой оконечной точки, не должен превышать размер FIFO. Следует также иметь в виду, что регистр TxMaxP не должен записываться, пока в FIFO присутствуют данные – результат такого действия будет неопределенным.

1.3.12.2.7.3.13.1.1 ОДИНОЧНАЯ БУФЕРИЗАЦИЯ ПАКЕТА

Если размер FIFO оконечной точки передатчика меньше удвоенного размера максимального пакета для этой конечной точки (что установлено регистром TxMaxP, или, при использовании динамического определения размера FIFO, регистром TxFIFO2) в FIFO может буферироваться только один пакет, и разрешена одиночная буферизация пакета.

При каждой загрузке отправляемого пакета в FIFO передатчика должен устанавливаться бит TxPktRdy в регистре TxCSR. Если в TxCSR установлен бит AutoSet, то бит TxPktRdy будет автоматически устанавливаться при загрузке в FIFO пакета максимального размера. Для пакетов размером меньше максимального, и в случаях, когда AutoSet может не использоваться

(высокоскоростные изохронные транзакции и транзакции по прерыванию) TxPktRdy должен устанавливаться вручную (то есть, CPU).

Когда бит TxPktRdy установлен, вручную или автоматически, пакет считается готовым к отправке. Бит FIFONotEmpty в TxCSR также установлен.

После успешной отправки пакета оба бита, TxPktRdy и FIFONotEmpty, будут сброшены, и сгенерировано прерывание соответствующей конечной точки передатчика (если разрешено). Затем в FIFO может быть загружен следующий пакет.

1.3.12.2.7.3.13.1.2 ДВОЙНАЯ БУФЕРИЗАЦИЯ ПАКЕТА

Примечание: Двойная буферизация пакета запрещена, если активирован (равен 1’b1) соответствующий бит DPktBufDis оконечной точки в регистре Tx DPktBufDis (подробности см. в п. 1.3.12.2.7.3.8.2). По умолчанию этот бит сброшен (равен 1’b0).

Если размер FIFO оконечной точки передатчика как минимум в два раза превышает удвоенный размер максимального пакета для этой конечной точки (что установлено регистром TxMaxP, или, при использовании динамического определения размера FIFO, регистром TxFIFO2) в FIFO может буферироваться два пакета, и разрешена двойная буферизация пакета.

При каждой загрузке отправленного пакета в FIFO передатчика должен устанавливаться бит TxPktRdy в TxCSR. Если в TxCSR установлен бит AutoSet, то бит TxPktRdy будет автоматически устанавливаться при загрузке в FIFO пакета максимального размера. Для пакетов размером меньше максимального и в случаях, когда AutoSet может не использоваться (высокоскоростные изохронные транзакции и транзакции по прерыванию) TxPktRdy должен устанавливаться вручную (то есть, CPU).

Когда бит TxPktRdy установлен, вручную или автоматически, пакет считается готовым к отправке. Бит FIFONotEmpty в TxCSR также установлен.

После загрузки первого пакета TxPktRdy немедленно сбрасывается, и вырабатывается прерывание. В FIFO передатчика может загружаться второй пакет, и бит TxPktRdy снова устанавливается (вручную или автоматически, если пакет максимального размера). Теперь оба пакета готовы к отправке.

После успешной отправки пакета оба бита, TxPktRdy и FIFONotEmpty, будут сброшены, и сгенерировано прерывание соответствующей конечной точки передатчика (если разрешено), чтобы указать на возможность загрузки следующего пакета в Tx FIFO. В данный момент состояние бита FIFONotEmpty указывает, сколько пакетов было загружено. Если бит FIFONotEmpty установлен, то в FIFO уже есть пакет, и в него может быть загружен только один пакет. Если бит FIFONotEmpty сброшен, то в FIFO нет пакетов, и в него можно загружать еще два пакета.

1.3.12.2.7.3.13.1.3 ИЗОХРОННЫЕ / ПО ПРЕРЫВАНИЮ ОКОНЕЧНЫЕ ТОЧКИ С ВЫСОКОЙ ПРОПУСКНОЙ СПОСОБНОСТЬЮ

В режиме высокой скорости соединения оконечные точки передатчика, настроенные на изохронные/по прерыванию транзакции с высокой пропускной способностью, могут передавать до трех пакетов ‘USB’ в любом микрофрейме с содержимым до 1024 байта в каждом пакете, что соответствует скорости передачи до 3072 байт на микрофрейм.

MUSBMHDRC поддерживает эту возможность, разрешая пользователю за одну транзакцию загружать в соответствующее FIFO пакеты данных до 3072 байт (то есть, 3 □□1024 байт). С точки зрения программного обеспечения в CPU далее процедура в точности соответствует

описанной выше для одиночной и двойной буферизации пакета (соответственно), за исключением того, что бит TxPktRdy всегда должен устанавливаться вручную (то есть, CPU), поскольку AutoSet не работает с изохронными/по прерыванию транзакциями с высокой пропускной способностью.

Любой загруженный в FIFO пакет данных, который больше максимального содержимого, для передачи по USB автоматически разделяется на ‘USB’ пакеты максимального содержимого, или меньшие по размеру. Количество пакетов USB, переданных в одном микрофрейме, и максимальное содержимое в каждом пакете определяется при помощи регистра TxMaxP. Биты 10-0 регистра TxMaxP определяют максимальное содержимое в пакете USB, а биты 12, 11 определяют максимальное количество таких пакетов, которые могут быть переданы в одном микрофрейме (2 или 3). Вместе они определяют максимальный размер пакета, который может быть загружен в FIFO.

Всегда отправляется как минимум один пакет: количество оставшихся пакетов, отправляемых в том же микрофрейме, будет зависеть от объема данных, загруженных в FIFO. Бит TxPktRdy будет сброшен, и прерывание сгенерировано только после отправки всех пакетов.

Каждый пакет USB отправляется в ответ на маркер чтения. Если, в конце микрофрейма MUSBMHDRC не получил достаточного количества маркеров для отправки всех пакетов по USB (например, из-за искажения одного из полученных маркеров чтения), оставшиеся в FIFO данные будут очищены. Затем бит TxPktRdy будет сброшен, и бит IncompTx в регистре TxCSR установлен, чтобы указать, что не все загруженные в FIFO данные были отправлены.

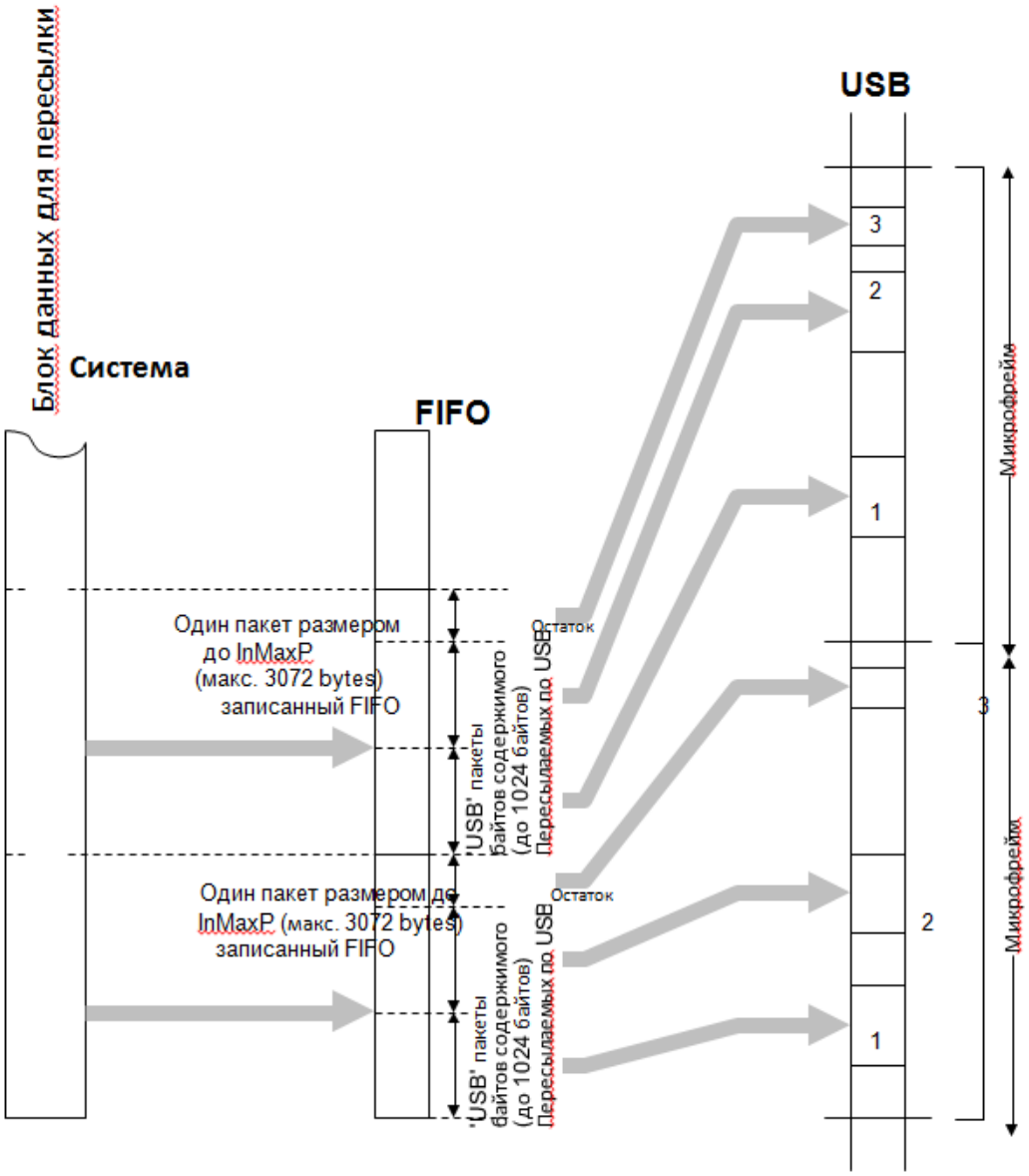


Рисунок 1.266

1.3.12.2.7.3.13.1.4 ОПЦИЯ СПЕЦИАЛЬНОЙ ОБРАБОТКИ ПАКЕТОВ

Пакеты, передаваемые при поточных операциях, определяются спецификацией USB и бывают размером в 8, 16, 32, 64 и 512 байт, причем опция 512 байт применима только к высокоскоростной пересылке. В некоторых системах, однако, прикладной программе удобнее за одну операцию записывать в оконечную точку большие массивы данных, чем может быть передано за одну операцию USB. В частности, такая ситуация возникает, когда одна и та же оконечная точка в одних случаях используется для высокоскоростной передачи 512 байт, а в других – для передач со средней скоростью. При работе со средней скоростью за одну операцию передается максимум 64 байта.

Для разрешения таких ситуаций в MUSBMHDRC имеется опция ‘разделение поточного пакета передатчика’, выбор которой позволяет записывать пакеты данных большего размера в поточные оконечные точки передатчика; затем данные разделяются на пакеты соответствующего (указанного) размера для передачи по шине USB. (Подобная опция имеется для чтения из поточных оконечных точек приемника данных большего размера, чем отдельные пакеты USB – см. раздел 1.3.12.2.7.3.13.1.4).

Если эта опция выбрана, регистр TxMaxP данной оконечной точки увеличивается на 16 бит, и младшие 11 бит регистра определяют содержимое для каждой отдельной пересылки, а старшие 5 бит определяют множитель. Прикладная программа затем может записать в FIFO пакеты данных размером содержимое □□множитель, которые MUSBMHDRC разделит на отдельные пакеты указанного размера содержимого для передачи по USB. С точки зрения программного обеспечения результирующая операция не будет отличаться от передачи одиночного пакета USB, за исключением размера записанного пакета.

Эта возможность предлагается как опция конфигурации, а не как стандартная характеристика, поскольку она существенно увеличивает количество используемых логических вентилей.

Примечание: эта функция используется только с поточными оконечными точками, и согласно спецификации USB размер содержимого должен составлять 8, 16, 32, 64 и 512 байт, причем опция 512 байт применима только к высокоскоростной передаче. Размер содержимого, записанный в регистре TxMaxP, должен также совпадать с полем wMaxPacketSize стандартного дескриптора для данной оконечной точки. Соответствующий FIFO буфер также должен быть достаточно большим, чтобы вместить пакет данных до его разделения.

1.3.12.2.7.3.13.2 ОБРАБОТКА ТРАНЗАКЦИЙ ЗАПИСИ В РОЛИ ВЕДОМОГО УСТРОЙСТВА

Когда MUSBMHDRC работает в режиме ведомого устройства, данные для транзакций записи обрабатываются с помощью FIFO буферов приемников MUSBMHDRC.

Размеры FIFO буферов приемников для оконечных точек от 1 до 15 определяются либо константами конфигурации в файле конфигурации MUSBMHDRC, либо, когда выбрано динамическое определение размера FIFO, при помощи регистра RxFIFO2. Максимальный размер пакета данных, который принимается оконечной точкой приемника в любом фрейме/микрофрейме (в высокоскоростном режиме), задается программой и определяется значением, записанным в регистр RxMaxP для данной оконечной точки (максимальное содержимое □□количество транзакций/(микрофреймов (где применимо)).

Если не используется динамическое определение размера FIFO, то когда максимальный размер пакета установлен меньшим или равным половине размера FIFO, для транзакций записи разрешена двойная буферизация пакета, а когда максимальный размер пакета больше половины размера FIFO, разрешена одиночная буферизация пакета. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакета является частью спецификации для FIFO оконечной точки – см. раздел 19). Когда разрешена двойная буферизация пакета, в FIFO могут буферироваться два пакета данных; когда разрешена одиночная буферизация, в этом случае может буферироваться только один пакет, даже если его размер меньше половины размера FIFO.

Примечание: Максимальный размен пакета не должен превышать размер FIFO.

1.3.12.2.7.3.13.2.1 ОДИНОЧНАЯ БУФЕРИЗАЦИЯ ПАКЕТА

Если размер FIFO оконечной точки приемника меньше удвоенного размера максимального пакета для этой конечной точки (что установлено регистром RxMaxP, или, при использовании

динамического определения размера FIFO, регистром RxFIFO2) в FIFO может буферироваться только один пакет, и разрешена одиночная буферизация пакета.

Когда пакет получен и помещен в FIFO приемника, в регистре RxCSR устанавливаются бит RxPktRdy (D0) и бит FIFOFull (D1), и генерируется прерывание (если разрешено) соответствующей оконечной точки приемника, сигнализируя, что пакет можно выгружать из FIFO.

После выгрузки пакета необходимо сбросить бит RxPktRdy, чтобы разрешить прием следующих пакетов. Если в регистре RxCSR (D15) установлен бит AutoClear, и из FIFO выгружен пакет максимального размера, то бит RxPktRdy сбрасывается автоматически. Бит FIFOFull также сбрасывается. Для пакетов, размер которых меньше максимального, RxPktRdy должен сбрасываться вручную (то есть, CPU) (исключения см. в описании регистра).

1.3.12.2.7.3.13.2.2 ДВОЙНАЯ БУФЕРИЗАЦИЯ ПАКЕТА

Примечание: Двойная буферизация пакета запрещена, если активирован (равен 1’b1) соответствующий бит DPktBufDis оконечной точки в регистре Rx DPktBufDis (подробности см. в п. 1.3.12.2.7.3.8.2). По умолчанию этот бит сброшен (равен 1’b0).

Если размер FIFO оконечной точки приемника как минимум в два раза превышает удвоенный размер максимального пакета для этой конечной точки (что установлено регистром RxMaxP, или, при использовании динамического определения размера FIFO, регистром RxFIFO2) в FIFO может буферироваться два пакета, и разрешена двойная буферизация пакета.

Когда первый пакет получен и помещен в FIFO приемника, в регистре RxCSR устанавливается бит RxPktRdy, и генерируется прерывание (если разрешено) соответствующей оконечной точки приемника. Примечание: Бит FIFOFull в регистре RxCSR на данном этапе не устанавливается: он устанавливается после получения и загрузки в FIFO приемника второго пакета.

После выгрузки каждого пакета необходимо сбросить бит RxPktRdy, чтобы разрешить прием следующих пакетов. Если в регистре RxCSR установлен бит AutoClr, и из FIFO выгружен пакет максимального размера, то бит RxPktRdy сбрасывается автоматически. Для пакетов, размер которых меньше максимального, RxPktRdy должен сбрасываться вручную (то есть, CPU).

Если бит FIFOFull при сбросе RxPktRdy был установлен в 1, то MUSBMHDRC сначала сбросит бит FIFOFull. Затем снова будет установлен бит RxPktRdy, чтобы указать, что в FIFO ожидает еще один пакет.

1.3.12.2.7.3.13.2.3 ИЗОХРОННЫЕ / ПО ПРЕРЫВАНИЮ ОКОНЕЧНЫЕ ТОЧКИ С ВЫСОКОЙ ПРОПУСКНОЙ СПОСОБНОСТЬЮ

В режиме высокой скорости соединения оконечные точки приемника, настроенные на изохронные/по прерыванию транзакции с высокой пропускной способностью, могут принимать до трех пакетов ‘USB’ в любом микрофрейме с содержимым до 1024 байта в каждом пакете, что соответствует скорости передачи до 3072 байт на микрофрейм. Транзакции по прерыванию с высокой пропускной способностью могут также приниматься в режиме ведущего устройства, однако в режиме ведомого устройства транзакции по прерыванию с высокой пропускной способностью не поддерживаются.

MUSBMHDRC поддерживает эту возможность, автоматически объединяя пакеты USB, полученные в микрофрейме, в один пакет размером до 3072 байт (то есть, 3 □□1024 байт) в FIFO приемника. С точки зрения программного обеспечения в CPU далее процедура в точности соответствует описанной выше для одиночной и двойной буферизации пакета (соответственно), за исключением того, что бит RxPktRdy всегда должен устанавливаться вручную (то есть, CPU), поскольку AutoClear не работает с изохронными транзакциями с высокой пропускной способностью.

Максимальное количество пакетов USB, переданных в одном микрофрейме, и максимальное содержимое в каждом пакете отделяется при помощи регистра RxMaxP. Биты 10-0 регистра RxMaxP определяют максимальное содержимое в пакете USB, а биты 12, 11 определяют максимальное количество таких пакетов, которые могут быть приняты в одном микрофрейме (2 или 3).

Количество пакетов USB, отправленных в любом микрофрейме, будет зависеть от объема данных, которые необходимо передать, и указывается посредством PID для отдельных пакетов. Если указанное число пакетов не получено к концу микрофрейма, в регистре RxCSR будет установлен бит IncompRx, указывая, что данные в FIFO неполны. Аналогичным образом, при получении пакета с неверным типом данных в регистре RxCSR будет установлен бит PID Error. Тем не менее, в каждом из этих случаев будет генерироваться прерывание, чтобы разрешить считывание полученных данных из FIFO.

Примечание: Условия, при которых сообщается об ошибке PID Error или IncompRx, зависят от точной последовательности принятых пакетов. Когда ядро работает в режиме ведомого устройства, эти условия приведены в следующей таблице. (Отдельную таблицу для режима ведущего устройства см. в разделе 1.3.12.2.7.3.14.2.)

Таблица 1.720

No. пакета(ов) ожидаем.	Данные пакета(ов) полученные	Ответ	No. пакета(ов) ожидаем.	Данные пакета(ов) полученные	Ответ
1	DATA0 (‘D0’)	OK	3	D0	OK
	DATA1 (‘D1’)	PID Error устан.		D1	IncompRx устан.
	DATA2 (‘D2’)	PID Error устан.		D2	IncompRx устан.
	MDATA (‘DM’)	PID Error устан.		DM	IncompRx устан.
2	D0	OK		DM D0	PID Error устан.
	D1	IncompRx устан.		DM D1	OK
	D2	IncompRx устан. + PID Error устан.		DM D2	IncompRx устан.
	DM	IncompRx устан.		DM DM	IncompRx устан.
	DM D0	PID Error устан.		DM DM D0	PID Error устан.
	DM D1	OK		DM DM D1	PID Error устан.
	DM D2	PID Error устан.		DM DM D2	OK
	DM DM	PID Error устан.		DM DM DM	PID Error устан.

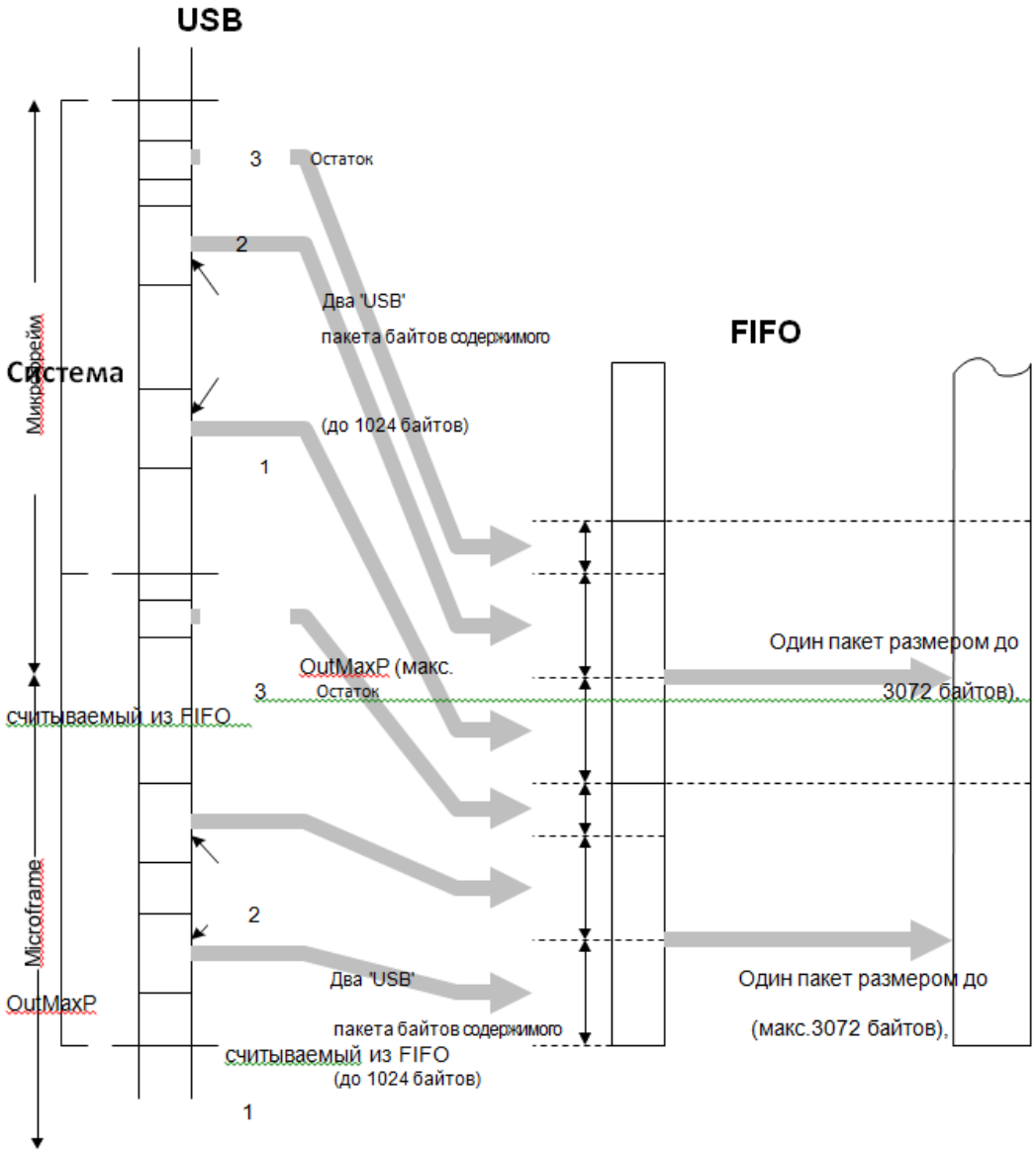


Рисунок 1.267

1.3.12.2.7.3.13.2.4 ОПЦИЯ СПЕЦИАЛЬНОЙ ОБРАБОТКИ ПАКЕТОВ

Пакеты, передаваемые при поточных операциях, определяются спецификацией USB и бывают размером в 8, 16, 32, 64 и 512 байт, причем опция 512 байт применима только к высокоскоростной пересылке. В некоторых системах, однако, прикладной программе удобнее за одну операцию читать из оконечной точки большие массивы данных, чем может быть передано за одну операцию USB. В частности, такая ситуация возникает, когда одна и та же оконечная точка в одних случаях используется для высокоскоростной передачи 512 байт, а в других – для передач со средней скоростью. При работе со средней скоростью за одну операцию передается максимум 64 байта.

Для разрешения таких ситуаций в MUSBMHDRC имеется опция ‘объединение поточного пакета приемника’, выбор которой позволяет соединять пакеты данных, полученные по шине USB, в пакеты данных большего размера до чтения их программой. (Подобная опция имеется для записи в поточные оконечные точки передатчика данных большего размера, чем отдельные пакеты USB – см. раздел 8.4.1.4).

Если эта опция выбрана, регистр RxMaxP данной оконечной точки увеличивается на 16 бит, и младшие 11 бит регистра определяют содержимое для каждой отдельной пересылки, а старшие 5 бит определяют множитель. MUSBMHDRC затем объединит соответствующее количество принимаемых пакетов USB в один пакет данных размером содержимое □□множитель внутри FIFO, прежде чем установить бит RxPktRdy, указывая прикладной программе, что в FIFO присутствует готовый для чтения пакет.

Эта возможность предлагается как опция конфигурации, а не как стандартная характеристика, поскольку она существенно увеличивает количество используемых логических вентилей.

Примечание: эта функция используется только с поточными оконечными точками, и согласно спецификации USB размер содержимого должен составлять 8, 16, 32, 64 и 512 байт, причем опция 512 байт применима только к высокоскоростной передаче. Размер содержимого, записанный в регистре RxMaxP, должен также совпадать с полем wMaxPacketSize стандартного дескриптора для данной оконечной точки. Соответствующий FIFO буфер также должен быть достаточно большим, чтобы вместить объединенный пакет данных.

Следует также отметить, что бит RxPktRdy устанавливается только тогда, когда было принято указанное число пакетов, или получен “короткий” пакет USB (то есть, пакет меньшего размера, чем указанное для оконечной точки содержимое). Если используется протокол, при котором оконечная точка принимает поточные пересылки, кратные указанному размеру содержимого, без завершающего короткого пакета, то регистр RxMaxP не следует программировать на ожидание большего числа пакетов, чем присутствует в пересылке (в противном случае выполнение программы не будет прервано в конце пересылки).

1.3.12.2.7.3.13.3 ДОПОЛНИТЕЛЬНЫЕ ДЕЙСТВИЯ

Ядро MUSBMHDRC автоматически реагирует на некоторые состояния шины USB или на действия ведущего устройства. Подробности приводятся ниже.

ПАКЕТ STALL ДЛЯ УПРАВЛЯЮЩЕЙ ПЕРЕСЫЛКИ

Ядро MUSBMHDRC автоматически выдает пакет STALL для управляющей пересылки при следующих условиях:

- Ведущее устройство во время фазы отправки данных (OUT Data) управляющей пересылки передает больше данных, чем было указано в запросе ведомого устройства в фазе SETUP.

Это состояние определяется MUSBMHDRC, когда ведущее устройство отправляет маркер записи (вместо маркера чтения) после того, как CPU выгрузило последний пакет вывода и установило бит DataEnd.

- Ведущее устройство в фазе приема данных (IN data) управляющей пересылки запрашивает больше данных, чем было указано в запросе ведомого устройства в фазе SETUP.

Это состояние определяется MUSBMHDRC, когда ведущее устройство отправляет маркер чтения (вместо маркера записи) после того, как CPU сбросило TxPktRdy и установило DataEnd в ответ на пакет ACK, выданный ведущим устройством в ответ на пакет, который должен быть последним.

- Ведущее устройство передает больше данных, чем MaxP с маркером записи.
- Ведущее устройство передает пакет данных больше нулевой длины для фазы отправки статуса (OUT Status).

ПАКЕТЫ ВЫХОДНЫХ ДАННЫХ НУЛЕВОЙ ДЛИНЫ В УПРАВЛЯЮЩИХ ПЕРЕСЫЛКАХ

Пакет выходных данных нулевой длины используется для того, чтобы указать на окончание управляющей пересылки. При нормальной работе такие пакеты должны приниматься только после того, как переданы все данные, длина которых была указана в запросе ведомого устройства (то есть, после того, как CPU установило бит DataEnd). Если же ведущее устройство передает пакет выходных данных нулевой длины раньше, чем переданы все данные, длина которых была указана в запросе ведомого устройства, эти сигналы приведут к преждевременному окончанию пересылки. В таком случае MUSBMHDRC автоматически очистит в FIFO любой маркер чтения, загруженный CPU, готовым к фазе приема данных, и установит бит SetupEnd.

1.3.12.2.7.3.13.4 СПЯЩИЙ РЕЖИМ ВЕДОМОГО УСТРОЙСТВА

Когда на шине USB отсутствует активность в течение 3 мс, MUSBMHDRC переходит в спящий режим. Если прерывание спящего режима разрешено, то в этот момент будет сгенерировано прерывание. В спящем режиме установится низкий уровень выходного сигнала SUSPENDM (если разрешено); этот сигнал может использоваться для перевода PHY в спящий режим. Кроме того, активируется сигнал POWERDWN: он может использоваться для остановки CLK и уменьшения энергопотребления в спящем режиме. POWERDWN остается в активном состоянии пока с шины не убирается питание (что свидетельствует об отсоединении ведомого устройства), или пока на шине не появляется сигнал выхода из спящего режима или сигнал сброса.

При обнаружении сигнала выхода из спящего режима MUSBMHDRC выходит из спящего режима и устанавливает высокий уровень сигнала SUSPENDM. В ответ на это из спящего режима выходит PHY. Если прерывание выхода из спящего режима разрешено, то в этот момент будет сгенерировано прерывание. CPU может также принудительно вывести MUSBMHDRC из спящего режима, установив бит Resume в регистре Power. После установки этого бита MUSBMHDRC выйдет из спящего режима и выставит на шину сигнал выхода из спящего режима. CPU должно сбросить этот бит через 10 мс (максимум 15 мс), чтобы снять сигнал выхода из спящего режима.

При выходе из спящего режима по команде CPU прерывание выхода из спящего режима не генерируется.

1.3.12.2.7.3.13.5 НАЧАЛО ФРЕЙМА

Когда MUSBMHDRC работает в режиме ведомого устройства, он должен получать от ведущего устройства пакет начала фрейма (SOF) один раз в миллисекунду при средней скорости соединения и один раз каждые 125 микросекунд при высокой скорости соединения.

После получения пакета SOF 11-битовый номер фрейма, содержащийся в пакете, записывается в регистр Frame, и на SOF_PULSE генерируется импульс длительностью в один интервал побитовой передачи CLK (если разрешено в регистре IntrUSBE).

После того, как MUSBMHDRC начал получать пакеты SOF, он ожидает их поступления каждую миллисекунду (или каждые 125 микросекунд). Если пакет SOF не получен по истечении 1,00358 мс (или 125,125 мкс), пакет считается потерянным; SOF_PULSE генерируется, но регистр Frame не изменяется. MUSBMHDRC продолжит генерировать импульс SOF_PULSE один раз в миллисекунду (или 125 микросекунд) и повторно синхронизирует эти импульсы с принятыми пакетами SOF после возобновления успешного приема этих пакетов.

1.3.12.2.7.3.14 РАБОТА В КАЧЕСТВЕ ВЕДУЩЕГО УСТРОЙСТВА

Когда бит Host Mode (DevCtl.D2) установлен в ‘1’, MUSBMHDRC работает как ведущее устройство либо для связи точка-точка с другим устройством USB, либо - при подключении к хабу - для связи с несколькими устройствами в режиме множества ведомых устройств. Поддерживаются функции USB высокой, средней и низкой скорости соединения как для связи точка-точка, так и для связи через хаб. (При необходимости ядро автоматически выполняет нужную транзакцию, чтобы позволить ведомому устройству с низкой или средней скоростью соединения использоваться с хабом USB 2.0.)

Поддерживаются управляющие, поточные, изохронные транзакции и транзакции по прерыванию.

В этом разделе описываются действия ядра в отношении оконечных точек передатчика и оконечных точек приемника, переупорядочивания транзакций, входа в спящий режим и выхода из него, а также сброса – доступные при использовании MUSBMHDRC в качестве ведущего устройства. Режим ведущего устройства автоматически выбирается при подключении ядра к хабу. Условия, при которых MUSBMHDRC работает в режиме ведущего устройства при соединении точка-точка, описаны в разделе 15: Согласование ведущего устройства.

1.3.12.2.7.3.14.1 НАЧАЛЬНАЯ НАСТРОЙКА ДЛЯ КОНФИГУРАЦИИ РЕЖИМА СМНОЖЕСТВА ВЕДОМЫХ УСТРОЙСТВ

Следующие требования начальной настройки применимы к ядру только если в конфигурации GUI разрешен режим множества ведомых устройств. Если опция режима множества ведомых устройств отключена, начальная настройка не выполняется.

Прежде чем получить доступ к ведомому устройству в роли ведущего устройства – для связи точка-точка или для связи с несколькими ведомыми устройствами через хаб – необходимо установить соответствующие регистры RxFuncAddr или TxFuncAddr для каждой используемой оконечной точки приемника или передатчика, чтобы записать адрес функции данного устройства. Если к MUSBMHDRC через высокоскоростной хаб USB 2.0 подключено устройство со средней или низкой скоростью соединения, необходимо также записать адрес хаба и адрес порта в соответствующие регистры RxHubAddr /TxHubAddr и RxHubPort/TxHubPort. Это позволит ядру поддерживать разделенные транзакции.

Кроме того, необходимо записать рабочую скорость устройства (высокая, средняя или низкая) в регистры Type0 (оконечная точка 0), TxType или RxType для каждой оконечной точки, к которой получает доступ ведущее устройство.

RxFuncAddr, TxFuncAddr, RxHubAddr, TxHubAddr, RxHubPort, TxHubPort – это 7-битные регистры чтения-записи. Дополнительная информация об этих регистрах приведена в разделе 1.3.12.2.7.3.5.2. Информация о регистрах Type0, TxType и RxType приведена в разделах 1.3.12.2.7.3.3.4, 1.3.12.2.7.3.3.14 и 1.3.12.2.7.3.3.16 соответственно.

Следует отметить, что при связи с множеством ведомых устройств значения в этих регистрах отражают текущее распределение оконечных точек ядра в соответствии с функциями, которые ассоциированы с подключенными устройствами. Для максимизации количества поддерживаемых устройств MUSBMHDRC позволяет динамически изменять это распределение – просто изменяя информацию об адресе и скорости, записанную в этих регистрах.

Любое изменение в распределении оконечных точек в соответствии с функциями устройств необходимо производить после завершения текущих транзакций на данных оконечных точках.

Дополнительная информация о распределении оконечных точек в соответствии с функциями устройств и о переключении между различными распределениями приведена в разделе 1.3.12.2.7.6.1.

1.3.12.2.7.3.14.2 ОБРАБОТКА ТРАНЗАКЦИИ ЧТЕНИЯ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Когда MUSBMHDRC работает в режиме ведущего устройства, транзакции чтения обрабатываются аналогично транзакциям записи в режиме ведомого устройства, за исключением транзакций, которые сначала необходимо инициировать установкой бита ReqPkt в регистре RxCSR. Этот бит указывает планировщику транзакций на активную транзакцию в данной оконечной точке. Затем планировщик транзакций посылает целевой функции маркер чтения.

После получения пакета и загрузки его в FIFO приемника устанавливается бит RxPktRdy в регистре RxCSR, и генерируется прерывание соответствующей оконечной точки приемника (если разрешено), сигнализируя, что пакет можно выгружать из FIFO.

После выгрузки пакета бит RxPktRdy должен быть сброшен. Бит AutoClear в регистре RxCSR может быть использован для автоматического сброса RxPktRdy, когда из FIFO выгружается пакет максимального размера (исключения см. в описании регистра).

Бит AutoReq в регистре RxCSR определяет автоматическую установку бита ReqPkt при сбросе бита RxPktRdy. Биты AutoClear и AutoReq могут использоваться вместе с контроллером DMA, чтобы выполнять полные поточные пересылки без вмешательства CPU. При передаче известного количества MaxP пакетов это количество должно быть записано в регистр RqPktCount соответствующей оконечной точки (см. раздел 1.3.12.2.7.3.8.1). Ядро уменьшает значение регистра RqPktCount после каждого запроса. Когда значение уменьшается от 1 до 0, бит AutoReq сбрасывается, чтобы предотвратить попытки дальнейших транзакций. В тех случаях, когда размер пересылки неизвестен, значение RqPktCount устанавливается равным нулю. Тогда бит AutoReq будет установлен до того, как его сбросит прием короткого пакета (то есть, меньшего, чем MaxP), что может произойти в конце поточной пересылки.

Если целевая функция отвечает на маркер чтения поточной пересылки или пересылки по прерыванию пакетом NAK, MUSBMHDRC продолжит попытки транзакции, пока не будет достигнут установленный NAK Limit. Если целевая функция отвечает пакетом STALL, MUSBMHDRC не будет повторять транзакцию, а сгенерирует прерывание CPU посредством установки бита RxStall в регистре RxCSR. Если целевая функция не отвечает на маркер чтения в

течение установленного времени (или в пакете присутствует ошибка CRC или согласующего бита) MUSBMHDRC повторит транзакцию. Если после трех попыток целевая функция по-прежнему не отвечает, MUSBMHDRC сбросит бит ReqPkt и сгенерирует прерывание CPU посредством установки бита Error в регистре RxCSR. Примечание: При транзакциях по прерыванию с высокой пропускной способностью ведущее устройство предпримет попытку 2 или 3 транзакций в одном микрофрейме и сгенерирует прерывание после получения всех пакетов. Если любая из этих транзакций не подтверждена целевым устройством, попыток осуществить остальные транзакции в этом же микрофрейме предприниматься не будет.

Примечание: Количество пакетов USB, отправленных в любом микрофрейме, будет зависеть от объема данных, которые необходимо передать, и указывается посредством PID для отдельных пакетов. Если указанное число пакетов не получено к концу микрофрейма, в регистре RxCSR будет установлен бит IncompRx, указывая, что данные в FIFO неполны. Аналогичным образом, при получении пакета с неверным типом данных в регистре RxCSR будет установлен бит PID Error. Тем не менее, в каждом из этих случаев будет генерироваться прерывание, чтобы разрешить считывание полученных данных из FIFO.

Условия, при которых сообщается об ошибке PID Error или IncompRx, зависят от точной последовательности принятых пакетов. Когда ядро работает в режиме ведущего устройства, эти условия приведены в следующей таблице. (Отдельную таблицу для режима ведомого устройства см. в разделе 1.3.12.2.7.3.13.1.3.)

No. пакета(ов) ожидаем.	Данные пакета(ов) полученные	Ответ	No. пакета(ов) ожидаем.	Данные пакета(ов) полученные	Ответ
1	DATA0 (‘D0’)	OK	2	D1 DM	PID Error устан.
	DATA1 (‘D1’)	PID Error устан.		D1 NR	IncompRx устан.
	DATA2 (‘D2’)	PID Error устан.		D2 D0	PID Error устан.
	MDATA (‘DM’)	PID Error устан.		D2 D1	PID Error устан.
	No response (‘NR’)	IncompRx устан.*		D2 D2	PID Error устан.
2	D0	OK		D2 DM	PID Error устан.
	D1 D0	OK		D2 NR	IncompRx устан. + PID Error устан.
	D1 D1	PID Error устан.		DM	PID Error устан.
	D1 D2	PID Error устан.		NR	IncompRx устан.*
No. пакета(ов) ожидаем.	Данные пакета(ов) полученные	Ответ	No. пакета(ов) ожидаем.	Данные пакета(ов) полученные	Ответ
3	D0	OK	3	D2 D2 D1	PID Error устан.
	D1 D0	OK		D2 D2 D2	PID Error устан.
	D1 D1	PID Error устан.		D2 D2 DM	PID Error устан.
	D1 D2	PID Error устан.		D2 D2 NR	IncompRx устан. + PID Error устан.
	D1 DM	PID Error устан.		D2 DM D0	PID Error устан.
	D1 NR	IncompRx устан.		D2 DM D1	PID Error устан.
	D2 D0	PID Error устан.		D2 DM D2	PID Error устан.
	D2 D1 D0	OK		D2 DM DM	PID Error устан.
	D2 D1 D1	PID Error устан.		D2 DM NR	IncompRx устан. + PID Error устан.
	D2 D1 D2	PID Error устан.		D2 NR	IncompRx устан.
	D2 D1 DM	PID Error устан.		DM	PID Error устан.
	D2 D1 NR	IncompRx устан.		NR	IncompRx устан.*
	D2 D2 D0	PID Error устан.			

* В этих случаях прерывание будет сгенерировано, но бит RxPktRdy не будет установлен, поскольку данные не были помещены в FIFO.

1.3.12.2.7.3.14.3 ОБРАБОТКА ТРАНЗАКЦИИ ЗАПАИСИ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Когда MUSBMHDRC работает в режиме ведущего устройства, транзакции записи обрабатываются аналогично транзакциям чтения в режиме ведомого устройства.

Бит TxPktRdy в регистре TxCSR необходимо устанавливать при загрузке каждого пакета в FIFO передатчика. Точно так же, установка бита AutoSet в регистре TxCSR (где применимо) вызовет автоматическую установку бита TxPktRdy при загрузке в FIFO пакета максимального размера. Более того, AutoSet может использоваться совместно с контроллером DMA, чтобы осуществлять полные поточные пересылки без вмешательства CPU.

Если целевая функция отвечает на маркер записи пакетом NAK, MUSBMHDRC продолжит попытки транзакции, пока не будет достигнут установленный NAK Limit. Если целевая функция отвечает пакетом STALL, MUSBMHDRC не будет повторять транзакцию, а сгенерирует прерывание CPU посредством установки бита RxStall в регистре TxCSR. Если целевая функция не отвечает на маркер чтения в течение установленного времени (или в пакете присутствует ошибка CRC или согласующего бита) MUSBMHDRC повторит транзакцию. Если после трех попыток целевая функция по-прежнему не отвечает, MUSBMHDRC очистит FIFO и сгенерирует прерывание CPU посредством установки бита Error в регистре TxCSR.

1.3.12.2.7.3.14.4 . ПЕРЕУПОРЯДОЧИВАНИЕ ТРАНЗАКЦИЙ

При работе в качестве ведущего устройства MUSBMHDRC поддерживает счетчик фреймов/микрофреймов. Если целевая функция представляет собой устройство со средней или высокой скоростью соединения, MUSBMHDRC автоматически будет передавать пакет SOF/uSOF в начале каждого фрейма/микрофрейма. Если целевая функция представляет собой устройство с низкой скоростью соединения, то по шине будет передаваться состояние ‘K’ для поддержания устройства в “активном состоянии”, чтобы устройство с низкой скоростью соединения не переходило в спящий режим.

После передачи пакета SOF/uSOF MUSBMHDRC циклически просмотрит все сконфигурированные оконечные точки в поисках активных транзакций. Активная транзакция определяется как оконечная точка приемника, для которой установлен бит ReqPkt, или оконечная точка передатчика, для которой установлен бит TxPktRdy и/или установлен бит FIFONotEmpt.

Активные изохронные транзакции или транзакции по прерыванию будут начаты только если они обнаружены в первом цикле планировщика транзакций фрейма/микрофрейма, и если интервальный счетчик для этой оконечной точки уменьшился до нуля. Это значит, что за n фреймов/микрофреймов для одной оконечной точки будет выполнена только одна изохронная транзакция или транзакция по прерыванию (до трех транзакций, если разрешена поддержка высокоскоростного режима), где n – интервал, установленный с помощью регистра для данной оконечной точки – см. разделы 1.3.12.2.7.3.3.15 и 1.3.12.2.7.3.3.17.

Активная поточная транзакция начнется немедленно, если во фрейме/микрофрейме осталось достаточно времени, чтобы завершить транзакцию до прихода следующего пакета SOF/uSOF. Если транзакцию требуется повторить (например, был принят пакет NAK, или целевая функция не отвечает), то транзакция повторится только после того, как планировщик

транзакций проверит все остальные точки на активные транзакции. Таким образом, оконечная точка, отправляющая много пакетов NAK, не блокирует другие транзакции на шине. Ядро также позволяет пользователю указать максимальное время, в течение которого пакеты NAK могут приниматься от определенного целевого устройства (см. разделы 1.3.12.2.7.3.3.4, 1.3.12.2.7.3.3.15и 1.3.12.2.7.3.3.17).

1.3.12.2.7.3.14.5 ШУМ

MUSBMHDRC не начнет транзакцию, пока шина не будет находиться в неактивном состоянии, как минимум в течение одного межпакетного интервала. Транзакция также не начнется, если ее невозможно закончить до конца фрейма. Если в конце фрейма шина по-прежнему активна, MUSBMHDRC будет считать, что подключенная функция неисправна, остановит все транзакции и сгенерирует прерывание шума.

1.3.12.2.7.3.14.6 СПЯЩИЙ РЕЖИМ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Если в регистре Power установлен бит SuspendMode, то MUSBMHDRC завершит текущую транзакцию, затем остановит планировщик транзакций и счетчик фреймов. Не будут начаты следующие транзакции, и не будут генерироваться пакеты SOF.

Для выхода из спящего режима CPU должен установить бит Resume и сбросить бит Suspend в регистре Power. Пока бит Resume установлен, MUSBMHDRC будет генерировать на шине сигнал выхода из спящего режима. Через 20 мс CPU сбросит бит Resume, поле чего начнется работа счетчика фреймов и планировщика транзакций.

В спящем режиме выходной сигнал SUSPENDM также перейдет в состояние с низким уровнем (если разрешено); это может использоваться для перевода в режим пониженного энергопотребления драйверов USB. Кроме того, активизируется сигнал POWERDWN: он может использоваться для остановки CLK и уменьшения энергопотребления в этом режиме. Однако если необходимо обеспечить возможность дистанционного выхода из спящего режима, должно поддерживаться питание PHY, чтобы MUSBMHDRC мог обнаружить на шине сигнал выхода из спящего режима.

1.3.12.2.7.4 СБРОС USB

1.3.12.2.7.4.1 В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Когда MUSBMHDRC работает как ведомое устройство, и на шине USB детектируется состояние сброса, устройство выполняет следующие действия:

Устанавливает FAddr в 0.

Устанавливает Index в 0.

Очищает все FIFO оконечных точек.

Очищает все управляющие регистры и регистры состояния.

Разрешает прерывания всех оконечных точек.

Генерирует прерывание сброса.

Если бит HS Enab в регистре Power (D5) был установлен, то MUSBMHDRC также пытается согласовать работу в режиме высокой скорости соединения. На выбор высокой скорости соединения указывает бит HS Mode (Power.D4).

Когда прикладная программа, управляющая MUSBMHDRC, получает прерывание сброса Reset, она должна закрыть все открытые каналы и ждать начала нумерации шины.

1.3.12.2.7.4.2 В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Если бит Reset в регистре Power устанавливается, когда MUSBMHDRC работает в режиме ведущего устройства, то MUSBMHDRC генерирует на шине сигнал сброса. Если был установлен бит HS Enab в регистре Power (D5), то MUSBMHDRC также пытается согласовать работу в режиме высокой скорости соединения.

CPU должно сохранять установленный бит Reset не менее 20 мс, чтобы обеспечить надежный сброс целевого устройства.

После того, как CPU сбросит этот бит, начнется работа счетчика фреймов и планировщика транзакций. На выбор высокой скорости соединения указывает бит HS Mode (Power.D4).

1.3.12.2.7.5 СПЯЩИЙ РЕЖИМ, ВХОД И ВЫХОД

Технология управления питанием (Link Power Management) дает две возможности для перевода MUSBMHDRC в спящий режим и вывода из спящего режима. Эти два метода проиллюстрированы ниже, на схеме основной транзакции LPM.

Процедура входа MUSBMHDRC в спящий режим и выхода из него зависит от режима работы ядра, в качестве ведомого или ведущего устройства, а также от желательного метода.

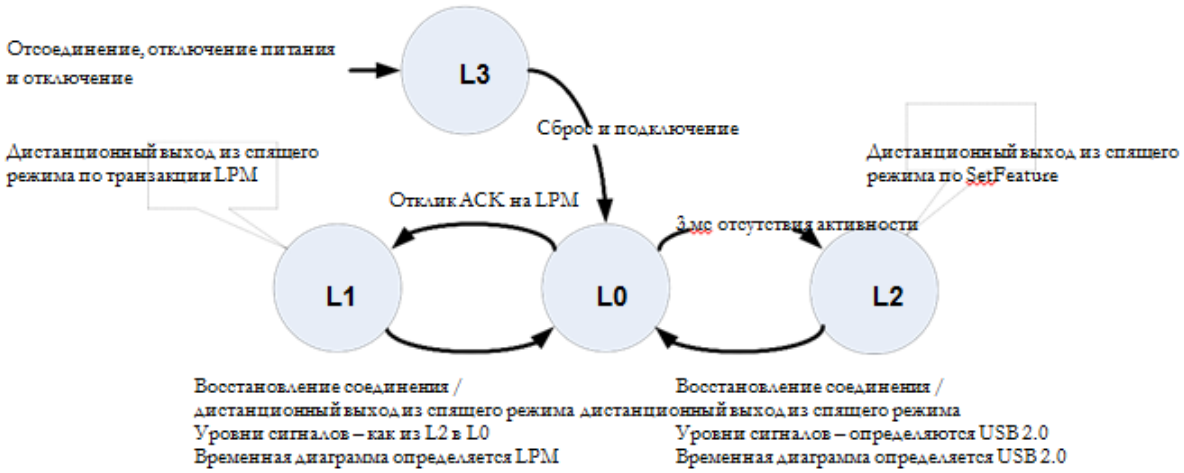


Рисунок 1.268

1.3.12.2.7.5.1 СПЯЩИЙ РЕЖИМ, ВХОД/ВЫХОД С ПОМОЩЬЮ НЕАКТИВНОСТИ НА ШИНЕ USB (ИЗ СОСТОЯНИЯ L0 В СОСТОЯНИЕ L1)

1.3.12.2.7.5.1.1 MUSBMHDRC В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Вход в спящий режим. При работе в режиме ведомого устройства MUSBMHDRC отслеживает активность на шине USB, и при отсутствии активности в течение 3 мс переходит в

спящий режим. Если прерывание спящего режима разрешено, в этот момент будет сгенерировано прерывание. Выходной сигнал SUSPENDM также перейдет в состояние с низким уровнем (если разрешено).

В этот момент также активизируется сигнал POWERDWN, указывая на возможность остановки CLK, чтобы уменьшить энергопотребление. POWERDWN остается в активном состоянии пока с шины не убирается питание (что свидетельствует об отсоединении ведомого устройства), или пока на шине не появляется сигнал выхода из спящего режима или сигнал сброса.

- Когда на шине появляется сигнал выхода из спящего режима, сначала включается CLK (при необходимости). Затем MUSBMHDRC автоматически выходит из спящего режима. Если разрешено прерывание выхода из спящего режима, то генерируется прерывание
- Инициирование дистанционного выхода из спящего режима. Если программа хочет инициировать дистанционный выход из спящего режима, в котором находится MUSBMHDRC, она должна установить в ‘1’ бит Resume (D2) регистра Power (Примечание: Если CLK было отключено, его необходимо вновь запустить до начала записи в регистр.) Затем программа должна оставить этот бит установленным в течение приблизительно 10 мс (минимум 2 мс, максимум 10 мс), прежде чем сбросить его в 0. К этому времени хаб должен выставить на шину USB сигнал выхода из спящего режима.

Примечание: Если программа инициирует дистанционный выход из спящего режима, прерывание не генерируется.

1.3.12.2.7.5.1.2 MUSBMHDRC В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

- Вход в спящий режим. При работе в режиме ведущего устройства MUSBMHDRC может быть принудительно переведен в спящий режим установкой бита Suspend регистра Power. После установки этого бита MUSBMHDRC завершит текущую транзакцию, затем остановит планировщик транзакций и счетчик фреймов. Не будут начаты следующие транзакции, и не будут генерироваться пакеты SOF.
- Если установлен бит Enable SuspendM (Power.D0), то после перехода MUSBMHDRC в спящий режим UTMI+ PHY переключится в режим пониженного энергопотребления и отключит XCLK.
- Передача сигнала выхода из спящего режима. Когда прикладной программе требуется, чтобы MUSBMHDRC вышел из спящего режима, она должна сбросить бит Suspend регистра Power, установить бит Resume и оставить его установленным в течение 20 мс. Пока бит Resume установлен, MUSBMHDRC будет генерировать на шину USB сигнал выхода из спящего режима. Через 20 мс CPU сбросит бит Resume, поле чего начнется работа счетчика фреймов и планировщика транзакций.
- Реакция на дистанционный выход из спящего режима. При обнаружении сигнала выхода из спящего режима от целевого устройства, когда MUSBMHDRC находится в спящем режиме, UTMI+ PHY будет выведено из режима пониженного энергопотребления, и возобновится XCLK. Затем MUSBMHDRC выйдет из спящего режима и автоматически установит в ‘1’ бит Resume регистра Power (D2), чтобы снять генерацию сигнала выхода из спящего режима от целевого устройства. Если прерывание выхода из спящего режима разрешено, будет сгенерировано прерывание.

1.3.12.2.7.5.2 СПЯЩИЙ РЕЖИМ, ВХОД/ВЫХОД С ПОМОЩЬЮ ТРАНЗАКЦИИ LPM (ИЗ СОСТОЯНИЯ L0 В СОСТОЯНИЕ L1)

1.3.12.2.7.5.2.1 MUSBMHDRC В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

- Вход в спящий режим.

При работе в режиме ведомого устройства MUSBMHDRC никогда не инициирует переход в спящий режим при помощи транзакции LPM (переход из стояния L0 в состояние L1). MUSBMHDRC лишь переходит в спящий режим по запросу хоста. Однако для этого функция LPM должна быть разрешена соответствующий установкой регистра LPM_CNTRL (адрес 0x362). Поле LPMEN (в регистре LPM_CNTRL) используется для разрешения и поддержки расширенных транзакций и транзакций LPM. Поле LPMXMT используется для того, чтобы указать аппаратной части на готовность перейти в спящий режим и ответить на следующую транзакцию LPM пакетом ACK. В этом случае MUSBMHDRC ответит на следующую транзакцию LPM пакетом ACK при выполнении всех остальных условий. Ответ MUSBMHDRC на транзакцию LPM приведен в таблице 1.721:

Таблица 1.721

LPMXMT	LPMCNTRL	Ожидание данных (Данные находятся в буферах FIFO передатчика)	Ответ на следующую транзакцию LPM.
1’b0, 1’b0 1’b1 1’b1	2’b00, 2’b10 2’b00 2’b10	Не имеет значения	Тайм-аут
1’b0, 1’b1	2’b01	Не имеет значения	STALL
1’b0	2’b11	Не имеет значения	NYET
1’b1	2’b11	Да	NYET
1’b1	2’b11	Нет	ACK

Для всех указанных выше случаев, когда MUSBMHDRC отвечает на транзакцию LPM (не истекло время ожидания), в регистре LPM_INTR (адрес 0x364) генерируется прерывание LPM. Следует отметить, что MUSBMHDRC ответит пакетом ACK только в том случае, если ни в одном из FIFO оконечных точек передатчика нет данных, ожидающих пересылки. При наличии ожидающих данных MUSBMHDRC ответит пакетом NYET.

После успешного приема транзакции LPM произойдут три события:

- В регистр LPM_ATTR (адрес 0x360) будут записаны значения из только что полученной транзакции LPM. Регистр LPM_ATTR содержит следующие поля:

- LinkState – Это поле указывает MUSBMHDRC, в какое состояние перейти. Единственное допустимое значение этого поля 4’h1 указывает, что ядро должно перейти в спящий режим. На любое другое значение MUSBMHDRC ответит пакетом STALL, и будет сгенерировано соответствующее прерывание. Однако в этом случае регистр LPM_ATTR будет изменен, чтобы программа могла обнаружить неправильное содержимое пакета LPM. Кроме того, будет сгенерировано прерывание LPMERR, указывающее программе о неподдерживаемой транзакции LPM.
- HIRD – Это поле сообщает MUSBMHDRC минимальный интервал времени, по истечению которого ведущее устройство может выставлять на шину сигнал выхода из спящего режима. Поле позволяет определить этот интервал в диапазоне от 50 мкс до 1200 мкс. Для следующих транзакций LPM это значение может быть другим.
- RmtWak – Это 1-битовое поле, указывающее, разрешен ли для MUSBMHDRC дистанционный выход из спящего режима. Для следующих транзакций LPM значение этого бита может быть другим. Этот бит не заменяет информацию о возможности выхода из спящего режима ранее установленной при нумерации MUSBMHDRC.

- The MUSBMHDRC перейдет в спящий режим через 9 мкс после передачи пакета ACK. Сигнал восстановления соединения может быть выставлен ведущим устройством или MUSBMHDRC через 50 мкс после этого события. В течение этих 9 мкс ведущее устройство может продолжать передачу транзакции LPM. В данном случае MUSBMHDRC ответит пакетом ACK независимо от значения LPMXMT в регистре LPM_CNTRL.

- Будет сгенерировано прерывание, сообщающее программе об ответе (в данном случае ACK). Ответ ACK указывает программе, что MUSBMHDRC перешел в спящий режим.

Поскольку основная цель спящего режима – экономия энергии, то программа считает регистр LPM_ATTR, чтобы определить атрибуты спящего режима. Программа должна в соответствии с этими атрибутами определить, можно ли применить в системе дополнительные меры энергосбережения. При этом следует иметь в виду, что если ведущее устройство инициирует сигнал выхода из спящего режима, MUSBMHDRC должен ответит на пакетные передачи через время, равное HIRD + 10 мкс.

- Появление на шине сигнала восстановления соединения.

Когда ведущее устройство активизирует шину, оно выставляет сигнал восстановления соединения через минимальный интервал времени, указанный в поле HIRD регистра LPM_ATTR (адрес 0x360). MUSBMHDRC должен быть способен ответить на трафик через промежуток времени, равный HIRD + 10us. MUSBMHDRC автоматически прейдет в нормальный режим работы, и в регистре LPM_INTR (адрес 0x364.4) будет сгенерировано прерывание. Однако для этого должны быть доступны входные сигналы CLK и XCLK. Чтобы удовлетворить требованиям к временным характеристикам восстановления соединения, в регистре LPM_CNTRL (адрес 0x362.4) имеется дополнительная функция LPMNAK. Если LPMNAK установлено в 1’b1, все оконечные точки отвечают на любую транзакцию (отличную от LPM) пакетом NAK. Этот бит будет оказывать влияние только после того, как MUSBMHDRC был переведен в спящий режим транзакцией LPM. Обычно этот бит активируется, когда также активировано поле LPMXMT. Использование этой функции может упростить выполнение требований к временным характеристикам восстановления соединения, поскольку для ответа MUSBMHDRC на трафик (пакетом NAK) необходим только сигнал XCLK. Программа может продолжать восстановление нормального режима работы системы, пока MUSBMHDRC отвечает на все транзакции пакетом NAK. После того, как система будет полностью восстановлена, программа может очистить поле LPMNAK в регистре LPM_CNTRL.

- Инициирование удаленного выхода из спящего режима.

Если программа хочет инициировать дистанционный выход из спящего режима, в котором находится MUSBMHDRC, она должна записать 1’b1 в бит LPMRES регистра LPM_CNTRL (address 0x362.1). Этот бит сбрасывается автоматически. Запись 1’b1 вызовет передачу на шину сигнала восстановления соединения в течение 50 мкс. Ведущее устройство ответит передачей сигнала восстановления соединения длительностью от 60 мкс до 990 мкс. Через 10 мкс после того, как ведущее устройство прекратит передачу сигнала восстановления соединения, MUSBMHDRC перейдет в нормальный режим работы и будет готов для передачи пакетов. В регистре LPM_INTR (адрес 0x364.4) будет сгенерировано прерывание выхода из спящего режима.

1.3.12.2.7.5.2.2 MUSBMHDRC В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

- Вход в спящий режим.

При работе в режиме ведущего устройство MUSBMHDRC вызывает переход в спящий режим при помощи транзакции LPM (переход из стояния L0 в состояние L1), инициируя транзакцию LPM следующим образом:

- Программа установит требуемые атрибуты спящего режима в регистре LPM_ATTR (адрес 0x360). Разрешение дистанционного выхода из режима и большой HIRD дадут ведомым устройствам больше возможностей для экономии энергии.
- - В регистре LPM_INTEN (0x363) должны быть разрешены все прерывания LPM.
- Программа должна инициировать транзакцию записью 0x01 в регистр LPM_CNTRL (адрес 0x362).
- Чтобы сообщить программе об ответе на транзакцию LPM, генерируется прерывание. Если получен пакет ACK, то через 8 мкс MUSBMHDRC автоматически перейдет в спящий режим.

Если ответ от ведомого устройства содержит ошибку битстаффинга (bit stuffing) или ошибку PID, то генерируется прерывание LPMERR. Аппаратная часть немедленно попытается выполнить транзакцию LPM еще два раза. Устройство перейдет в спящий режим через 8 мкс после начального пакета LPM, и поэтому способно ответить на обе эти последующие транзакции LPM. Если тайм-аут LPM повторится три раза, то будут установлены прерывания LPMNC и LPMERR. В этот момент программа не знает о состоянии устройства и должна определять его другими способами.

- Передача сигнала восстановления соединения.

Сигнал восстановления соединения генерируется программой следующим образом:

- В регистре LPM_INTEN должны быть разрешены все прерывания LPM.
- Программа должна записать бит LPMRES в регистр LPM_CNTRL. Этот бит сбрасывается автоматически. При этом по шине будет передаваться сигнал восстановления соединения в течение времени, указанного в поле HIRD регистра LPM_ATTR. Аппаратная часть предполагает, что это значение было использовано в последней транзакции LPM, которая вызвала переход в спящий режим.
- По истечение времени HIRD + 10 мкс, MUSBMHDRC перейдет в нормальный режим работы и будет готов для передачи пакетов. В регистре LPM_INTR (адрес 0x364.4) будет сгенерировано прерывание выхода из спящего режима.

Примечание: До выхода из спящего режима программа должна убедиться, что система полностью вышла из состояния пониженного энергопотребления, и входные сигналы CLK и XCLK доступны.

-
- Ответ на дистанционный выход из спящего режима.

Если в транзакции LPM, инициировавшей переход в спящий режим, разрешена функция дистанционного выхода из спящего режима, то устройство может выставлять на шину сигнал восстановления соединения. В этом случае устройство будет подавать на шину сигнал восстановления соединения (resume signaling) в течение 50 мкс. MUSBMHDRC немедленно начнет подавать на шину сигнал восстановления соединения в течение 60 мкс. Через 10 мкс после окончания сигнала выхода из спящего режима MUSBMHDRC перейдет в нормальный режим работы и будет готов для передачи пакетов. В регистре LPM_INTR (адрес 0x364.4) будет сгенерировано прерывание выхода из спящего режима.

1.3.12.2.7.6 ПОДДЕРЖКА МНОЖЕСТВА ВЕДОМЫХ УСТРОЙСТВ

Этот раздел применим только если в конфигурации GUI включена опция работы MUSBMHDRC с множеством ведомых устройств. В режиме ведущего устройства MUSBMHDRC имеет возможность выступать в качестве ведущего для разных ведомых устройств – с высокой, средней и низкой скоростью соединения – если эти устройства подключены к MUSBMHDRC через хаб USB.

Главная особенность поддержки ядром множества ведомых устройств – это возможность индивидуального распределения функций целевых устройств по различным оконечным точками передатчика и приемника, имплементированных в ядро MUSBMHDRC. Более того, это распределение может осуществляться динамически, что позволяет использовать в разных сочетаниях ведомые устройства из целевого списка. Тем не менее, сочетания ведомых устройств, которые могут использоваться совместно, ограничены количеством оконечных точек передатчика и приемника, имплементированных в ядро. Дополнительные устройства могут быть добавлены только при наличии доступных оконечных точек.

1.3.12.2.7.6.1 РАСПРЕДЕЛЕНИЕ УСТРОЙСТВ ПО ОКОНЕЧНЫМ ТОЧКАМ

Отдельные функции подключенных устройств распределяются по оконечным точкам внутри ядра MUSBMHDRC при помощи группы из трех регистров, которые назначаются каждой из оконечных точек приемника или передатчика, имплементированных в ядро (включая оконечную точку 0).

Эти регистры - Tx/RxFuncAddr, Tx/RxHubAddr and Tx/RxHubPort. (Расположение этих регистров зависит от конкретной оконечной точки MUSBMHDRC.

Информация, которую необходимо записать в регистр Tx/RxFuncAddr, представляет собой адрес целевой функции, к которой будет осуществляться доступ через выбранную оконечную точку. Эту информацию следует записывать отдельно для каждой используемой оконечной точки передатчика или приемника. Для оконечной точки 0 необходимо записать значения как в TxFuncAddr, так и в RxFuncAddr.

Регистры Tx/RxHubAddr и Tx/RxHubPort необходимы в том случае, когда ведомое устройство со средней или низкой скоростью соединения подключается к MUSBMHDRC через высокоскоростной хаб USB 2.0, который выполняет необходимые транзакции трансляции между пересылкой с низкой/средней скоростью и пересылкой с высокой скоростью. Регистры Tx/RxHubAddr и Tx/RxHubPort нужны для записи адреса хаба, выполняющего транзакцию трансляции, и порта этого хаба, через который соответствующая оконечная точка передатчика/приемника получает доступ к устройству. Примечание: Если к хабу подключена оконечная точка 0, то для нее требуется установить оба регистра, передатчика и приемника.

Регистр Tx/RxHubAddr также используется для записи информации, обеспечивает ли хаб несколько трансляторов транзакций, или всего один транслятор транзакций. Это оказывает существенное влияние на общую пропускную способность, которой можно достичь.

Подробное описание этих регистров приводится в разделах 1.3.12.2.7.3.5.2 и 1.3.12.2.7.3.5.3.

Кроме адреса и целевой функции, записываемых в этих трех регистрах, необходимо также записать номер оконечной точки, рабочую скорость целевого устройства и тип транзакции, которая будет выполняться.

Для оконечной точки передатчика эта информация должна быть записана в регистре TxType по адресу 1Ah, когда в регистре Index выбрана требуемая оконечная точка. Для оконечной точки приемника эта информация должна быть записана в регистр RxType по адресу 1Ch, когда в регистре Index выбрана требуемая оконечная точка. В обоих случаях номер оконечной точки записывается в битах D3 – D0, тип транзакции выбирается с помощью битов D5 – D4, а рабочая скорость определяется битами D7 – D6.

В случае оконечной точки 0 необходимо задать только скорость (эта оконечная точка предназначена для обработки управляющих транзакций и поэтому всегда ассоциируется с оконечной точкой 0 устройства). Скорость задается битами D7 – D6 регистра Type 0, который расположен по адресу 1Ah, когда в регистре Index установлено значение 0.

Подробная информация о регистрах Type0, TxType и RxType приведена в разделах 1.3.12.2.7.3.3.4, 1.3.12.2.7.3.3.14 и 1.3.12.2.7.3.3.16 соответственно.

1.3.12.2.7.6.2 РАБОТА

После распределения функций по оконечным точкам и записи скорости целевого устройства, как описано в разделе 11.1, большинство операций первоначальной настройки в режиме поддержки множества ведомых устройств не отличаются от аналогичных действий, когда ядро подключено к одному устройству. Их подробное описание приводится в других разделах руководства.

Тем не менее, в следующих случаях требуются дополнительные действия:

- когда выбрана опция динамического перераспределения функций по оконечным точкам (то есть, поддержка более широкого диапазона устройств)
- когда управляющие пакеты, обычно ассоциированные с оконечной точкой 0, обрабатываются через другие точки.

При использовании динамического распределения пользователь должен следить за текущим состоянием флага данных, ассоциированного с оконечной точкой и с каждым из устройств, которые назначены для данной точки. Знание этого состояние необходимо для того, чтобы пользователь мог выбирать правильное состояние флага данных при переключении между устройствами. (За это действие отвечает пользователь, поскольку ядро не может определить ожидаемое состояние флага данных, когда функция подключается и отключается).

Состояние флага данных может быть изменено записью в соответствующий регистр TxCSRL/RxCSRL, где устанавливаются биты Data Toggle Write Enable и Data Toggle, которые присутствуют в этих регистрах в режиме ведущего устройства. (См. разделы 0 и 1.3.12.2.7.3.3.11)

Примечание: При работе ядра в режиме ведущего устройства в регистр CSR0 также включены биты Data Toggle Write Enable и Data Toggle. Однако управляющие операции, выполняемые через оконечную точку 0 ядра, обычно должны оставлять флаг данных в известном состоянии.

Когда управляющие пакеты обрабатываются не через оконечную точку 0, пользователь должен дополнительно запрашивать отправку каждого маркера Setup. Для этого требуется наряду с битом TkPktRdy установить бит SetupPkt, включенный в регистр TxCSR при работе

ядра в режиме ведущего устройства. Если бит SetupPkt не установлен, будет отправляться маркер записи.

В общем случае рекомендуется использовать оконечную точку 0 ядра для обработки управляющих пакетов для всех устройств, подключенных к ядру, и соответствующим образом переключать назначение этой оконечной точки. Затем следует позаботиться о посылке правильного маркера, что обеспечивает корректную отправку флага данных для этой оконечной точки.

Для этой функции, как указано выше, можно использовать другую оконечную точку, однако при это следует учитывать следующее:

- управляющая функция должна быть назначена паре оконечных точек передатчик/приемник (то есть, с одним и тем же номером).
- выбранные оконечные точки должны быть ассоциированы с FIFO буферами, способными вместить размер пакета для транзакций оконечной точки 0 при выбранной рабочей скорости (то есть, минимум 8 байт для транзакций с низкой или средней скоростью и 64 байта для высокоскоростных транзакций).

1.3.12.2.7.6.3 ПРОПУСКНАЯ СПОСОБНОСТЬ

Способность системы с множеством ведомых устройств справляться с изохронными (особенно) транзакциями определяется доступной пропускной способностью.

После первоначальной настройки оконечной точки все переупорядочивание транзакций выполняется аппаратной частью. Однако, как и в случае ведущих устройств EHCI/OHCI/UHCI на базе персонального компьютера, перед открытием периодического канала (periodic pipe) программа должна определить, доступна ли требуемая пропускная способность. Затем, если периодический канал открыт для устройства со средней скоростью соединения через высокоскоростной хаб, программа должна подтвердить, доступна ли требуемая пропускная способность как на логической высокоскоростной шине, так и на шине со средней скоростью, генерируемой транслятором транзакций хаба.

Пропускная способность, необходимая для разных транзакций, может быть определена при помощи алгоритмов, аналогичных алгоритмам для ведущих устройств на основе персонального компьютера (подробно описаны в разделе 5.11.3 спецификации USB 2.0).

Очевидно, что пропускная способность будет больше в том случае, когда используемый хаб поддерживает несколько трансляторов транзакций.

1.3.12.2.7.7 ПОДКЛЮЧЕНИЕ/ОТКЛЮЧЕНИЕ

Поведение MUSBMHDRC при подключении и отключении зависит от того, в каком режиме оно используется - ведущего устройства, или ведомого устройства при связи между равноправными устройствами.

1.3.12.2.7.7.1 РЕЖИМ ВЕДУЩЕГО УСТРОЙСТВА

Если MUSBMHDRC работает в режиме ведущего устройства, CPU начинает сеанс с установки бита Session (DevCtl.D0). Затем на VBus подается питание, и ядро ждет подключения устройства.

Когда устройство обнаружено, вырабатывается прерывание Connect (то есть устанавливается бит IntrUSB.D4). Скорость подключенного устройства может быть определена считыванием регистра DevCtl, в котором для устройства с высокой/средней скоростью соединения будет установлен бит FSDev (D6), а для устройства с низкой скоростью соединения будет установлен бит LSDev (D5). Затем CPU должно сбросить устройство. Если установлены оба бита, FSDev и HS Enab (Power.D5), то MUSBMHDRC попытается согласовать высокую скорость соединения. Успешное согласование будет отмечено установкой бита HS Mode (Power.D4).

CPU должно сохранять установленный бит Reset в течение 20 мс, чтобы обеспечить сброс целевого устройства. Затем можно начинать нумерацию устройства.

Если устройство отключается во время сеанса, то генерируется прерывание Disconnect (то есть, устанавливается бит IntrUSB.D5).

1.3.12.2.7.7.2 РЕЖИМ ВЕДОМОГО УСТРОЙСТВА

Если MUSBMHDRC работает в режиме ведомого устройства, то при подключении к ведущему устройству прерывание не генерируется. Однако если ведущее устройство прерывает сеанс, то генерируется прерывание Disconnect (IntrUSB.D5)

1.3.12.2.7.8 ПРОГРАММИРОВАНИЕ

В этом и следующих разделах описываются действия, которые должно выполнить устройство, управляющее ядром MUSBMHDRC, а также все аспекты работы ядра, которые влияют на эти действия.

В дальнейшем предполагается, что устройство управления представляет собой микроконтроллер, реализованный аппаратно, но это может быть и специализированный аппаратный логический блок.

1.3.12.2.7.8.1 ПРОГРАММНОЕ ПОДКЛЮЧЕНИЕ/ОТКЛЮЧЕНИЕ

При необходимости MUSBMHDRC позволяет программному обеспечению управлять подключением к шине USB.

Если выбрано программное подключение/отключение, то при работе MUSBHHDRC в режиме ведомого устройства UTMI+- соответствующий PHY, используемые с MUSBMHDRC, могут переключаться из нормального режима в режим запрета передачи и наоборот установкой/сбросом бита 6 регистра Power (этот бит обозначен как Soft Conn). Когда бит Soft Conn установлен в 1, PHY переводится в нормальный режим, и линии D+/D- шины USB разрешены. Одновременно MUSBMHDRC переводится в состояние, при котором он не будет реагировать на любые сигналы USB, за исключением сброса USB.

Когда эта функция включена, и бит Soft Conn сбрасывается в 0, PHY переключается в режим запрета передачи, D+ and D- переводятся в третье состояние, и другие устройств на шине USB воспринимают MUSBMHDRC как отключенный.

После аппаратного сброса (NRST = 0), Soft Conn сбрасывается в 0. Таким образом, MUSBMHDRC будет оставаться отключенным до тех пор, пока программа не установит Soft Conn в 1. Прикладная программа затем может определять, когда переводить PHY в нормальный режим. Системы с длительной процедурой инициализации могут использовать это

обстоятельство для того, чтобы убедиться, что инициализация закончена, и система готова выполнить нумерацию, прежде чем подключать ее к шине USB.

После установки бита Soft Conn в 1 программа также может имитировать отключение, сбросив этот бит в 0.

1.3.12.2.7.8.2 ОБРАБОТКА ПРЕРЫВАНИЙ USB

При поступлении прерывания от USB, CPU должно считать регистр состояния прерываний, чтобы определить оконечную точку(и), сгенерировавшую прерывание, и перейти к соответствующей подпрограмме. Если прерывание сгенерировали несколько оконечных точек, то первой необходимо обслужить оконечную точку 0, а затем остальные.

Алгоритм подпрограммы обслуживания прерывания USB изображен на приведенном ниже рисунке.

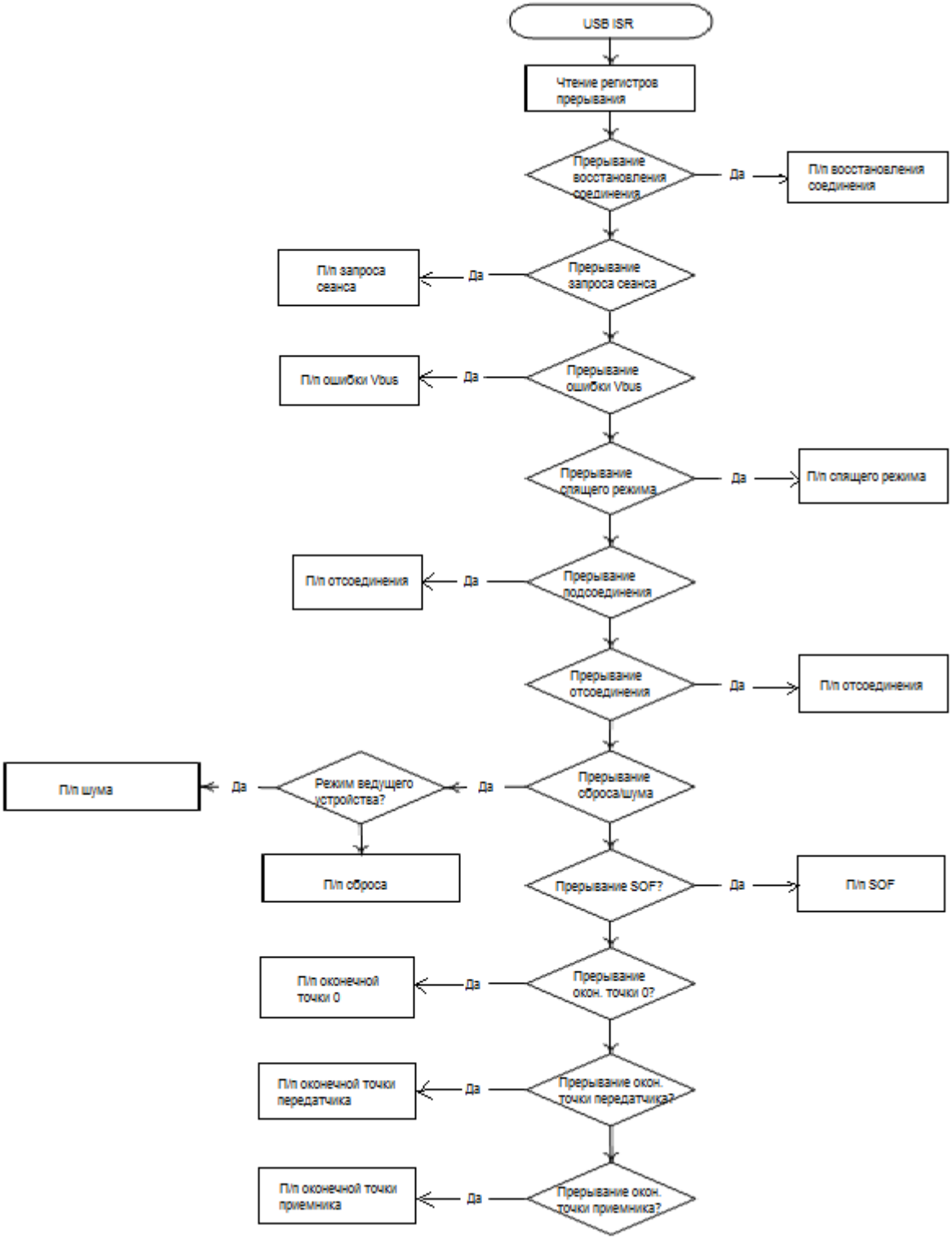


Рисунок 1.269

1.3.12.2.7.9 ЗАПРОС СЕАНСА OTG

В целях энергосбережения дополнение USB On-The-Go позволяет подавать питание на VBus только при необходимости и выключать, когда шина не используется. MUSBMHDRC обеспечивает дополнительную экономию энергии, позволяя в отсутствие сеанса остановить CLK. CLK может быть остановлен при активизации POWERDWN.

VBus всегда питается от устройства ‘A’, подключенного к шине. MUSBMHDRC определяет, является ли он устройством ‘A’ или ‘B’, проверяя входной сигнал IDDIG от UTMI+ PHY. Этот сигнал переводится в состояние с низким уровнем при обнаружении разъема ‘A-типа’ (указывая, что MUSBMHDRC является устройством ‘A’), или в состояние с высоким уровнем при обнаружении разъема ‘B-типа’ (указывая, что MUSBMHDRC является устройством ‘B’).

1.3.12.2.7.9.1 НАЧАЛО СЕАНСА

Когда устройству, содержащему MUSBMHDRC, требуется начать сеанс, CPU должно установить бит Session регистра DevCtl (D0). MUSBMHDRC затем разрешит контроль сигнала ID. В результате уровень входного сигнала установится низким, если обнаружено соединение A-типа, или высоким, если обнаружено соединение B-типа. Также устанавливается бит B-Device в регистре DevCtl (D7), указывающий, работает ли MUSBMHDRC в роли устройства ‘A’ или устройства ‘B’.

Если MUSBMHDRC является устройством ‘A’: MUSBMHDRC перейдет в режим ведущего устройства (устройство ‘A’ по умолчанию всегда ведущее), включит VBus и будет ожидать, когда VBus превысит порог VBus Valid, на что укажет высокий уровень входного сигнала VBUSVALID. (Это событие также приведет к установке значения 11b битов Vbus[1:0] регистра DevCtl (D4 – D3)).

Затем MUSBMHDRC будет ожидать подключения ведомого устройства. После обнаружения ведомого устройства генерируется прерывание Connect (IntrUSB.D4) (если разрешено), и в регистре DevCtl будет установлен бит FSDev или LSDev (D6/D5 соответственно) в зависимости от типа ведомого устройства – с высокой/средней или низкой скоростью соединения.

Далее CPU должен сбросить это ведомое устройство. Если оба бита FSDev и HSEnab (Power.D5) установлены, MUSBMHDRC будет отслеживать сигнал LINESTATE во время сброса, чтобы обнаружить поступивший от ведомого устройства ЛЧМ-сигнал режима высокой скорости соединения. На получение этого сигнала MUSBMHDRC отвечает ЛЧМ-сигналами режима высокой скорости соединения и переключается в режим с высокой скоростью соединения.

Для окончания сеанса необходимо сбросить бит Session (DevCtl.D0). MUSBMHDRC также автоматически завершит сеанс при обнаружении помехи.

Если MUSBMHDRC является устройством ‘B’: MUSBMHDRC подаст запрос на сеанс при помощи протокола запроса сеанса, определенного в приложении USB On-The-Go, то есть, сначала активирует DISCHRGVBUS, чтобы снять питание с VBus. Затем, когда VBus опустится ниже порога Session End (на что укажет высокий уровень входного сигнала SESSEND и биты Vbus[1:0] регистра DevCtl (D4 – D3), установленные в 00b) – и состояние линии SE0 сохранялось > 2 мс – MUSBMHDRC сначала подает импульс на линию данных, а затем на VBus (установив высокий уровень CHRGVBUS).

В конце сеанса сбрасывается бит Session – обычно это делает MUSBMHDRC, но может также CPU, если прикладная программа хочет выполнить программное отключение (см. описание DevCtl в разделе **Error! Reference source not found.**). MUSBMHDRC затем переключит сигнал TERMSEL, который заставит PHY отключить нагрузочный резистор на D+. Это укажет устройству ‘A’, чтобы оно завершило сеанс.

1.3.12.2.7.9.2 ОБНАРУЖЕНИЕ АКТИВНОСТИ

Когда другое устройство соединения OTG желает начать сеанс, оно либо повышает уровень VBus выше порога Session Valid, если оно является устройством ‘A’ (на что указывает высокий уровень входного сигнала AVALID и запись в биты Vbus[1:0] регистра DevCtl (D4 – D3) значения 10b), либо если оно является устройством ‘B’, то сначала подает импульс на линию данных, а затем на VBus. В зависимости от этого MUSBMHDRC определяет тип текущего соединения устройства, ‘A’ или ‘B’, и выполняет соответствующие действия:

Если VBus превысило порог Session Valid: В этом случае MUSBMHDRC является устройством ‘B’. Тогда MUSBMHDRC устанавливает бит Session в регистре DevCtl (D0). При обнаружении на шине сигнала сброса, генерируется прерывание Reset (IntrUSB.D2) (если разрешено), которое CPU интерпретирует ка начало сеанса.

MUSBMHDRC будет работать в режиме ведомого устройства, поскольку устройство типа ‘B’ по умолчанию считается ведомым.

В конце сеанса устройство ‘A’ выключит питание VBus. Когда уровень VBus опустится ниже порога Session Valid (на что указывает низкий уровень входного сигнала AVALID и запись в биты Vbus[1:0] регистра DevCtl (D4 – D3) значения 01b), MUSBMHDRC обнаружит это и сбросит бит Session, указывая на завершение сеанса. Также будет сгенерировано прерывание Disconnect (IntrUSB.D5) (если разрешено).

Если обнаружена подача импульса на линию данных/VBus: В этом случае MUSBMHDRC является устройством ‘A’. Оно сгенерирует прерывание Session Request (IntrUSB.D6 – если разрешено), чтобы указать, что устройство ‘B’ запрашивает сеанс. CPU затем должно начать сеанс, установив бит Session (DevCtl.D0).

1.3.12.2.7.10 СОГЛАСОВАНИЕ ВЕДУЩЕГО УСТРОЙСТВА

Если MUSBMHDRC является устройством ‘A’ (IDDIG низкий, B-Device (DevCtl.D7) = 0), то после начала сеанса оно автоматически перейдет в режим ведущего устройства.

Если MUSBMHDRC является устройством ‘B’:(IDDIG высокий, B-Device (DevCtl.D7) = 1), то после начала сеанса оно автоматически перейдет в режим ведомого устройства. Тем не менее CPU может потребовать переключения MUSBMHDRC в режим ведущего устройства, установив бит Host Req в регистре DevCtl (D1). Этот бит может устанавливаться либо одновременно с запросом сеанса установкой бита Session (DevCtl.D0), либо в любое время после начала сеанса. Когда MUSBMHDRC затем переходит в спящий режим (отсутствие активности на шине более 3 мс), то считая, что бит Host Req останется установленным, он перейдет в режим ведущего устройства и начнет согласование ведущего устройства (в соответствии с приложением USB On-The-Go), переключив сигнал TERMSEL, который заставит PHY отключить нагрузочный резистор на D+. Это заставит устройство ‘A’ перейти в режим ведомого устройства и подключить собственный нагрузочный резистор. Обнаружив это, MUSBMHDRC генерирует прерывание Connect (IntrUSB.D4), если оно разрешено. Кроме того, установится бит Reset в регистре Power (D3), чтобы начать сброс устройства ‘A’. (MUSBMHDRC автоматически начинает последовательность сброса, чтобы сброс начался через 1 мс после того, как устройство ‘A’ подключит свой нагрузочный резистор). CPU должно подождать не менее 20 мс, а затем сбросить бит Reset и пронумеровать устройство ‘A’.

Когда устройство ‘B’ она основе MUSBMHDRC закончило пользоваться шиной, CPU должно перевести его в спящий режим, установив бит Suspend Mode в регистре Power (D1). Устройство ‘A’ должно это обнаружить и также завершить сеанс или переключиться в режим ведущего устройства. Если ‘A’ построено на основе MUSBMHDRC, оно сгенерирует прерывание Disconnect (IntrUSB.D5), если оно разрешено.

1.3.12.2.7.11 ОБЩАЯ ПОДДЕРЖКА DMA

MUSBMHDRC поддерживает прямой доступ FIFO буферам для оконечных точек передатчика 1 – 15 и оконечных точек приемника 1 – 15 с помощью либо встроенного контроллера DMA (если он имплементирован), либо внешнего контроллера DMA.

Основой поддержки как встроенного, так и внешнего контроллера DMA служит отдельная линия запроса для каждой оконечной точки приемника и каждой оконечной точки передатчика. Если определено общее количество N оконечных точек передатчика и общее количество M оконечных точек приемника (в дополнение к оконечной точке 0), то с оконечными точками передатчика 1 ... N ассоциируются DMA_REQ[0] ... DMA_REQ[N–1], а с оконечными точками приемника 1 ... M - DMA_REQ[N+M–1]. Эти линии запроса обрабатываются внутри ядра, если используется встроенный контроллер DMA, но также доступны на верхнем уровне ядра, что позволяет использовать их для внешнего контроллера DMA.

Линии запроса DMA разрешаются индивидуально битом DMAReqEnab соответствующего управляющего регистра TxCSR (D12) или управляющего регистра RxCSR (D13) и работают в двух режимах, которые называются режимом 0 запроса DMA и режимом 1 запроса DMA. Режим работы выбирается с помощью бита DMAReqMode (TxCSR.D10 для оконечных точек передатчика и RxCSR.D11 для оконечных точек приемника). Требуемый режим запроса необходимо выбрать при использовании как встроенного, так и внешнего контроллера DMA.

Для оконечных точек приемника, работающих в режиме 0 запроса DMA, линия запроса DMA переходит в состояние с высоким уровнем, когда пакет данных доступен в FIFO оконечной точки, и обычно переходит в состояние с низким уровнем в конце цикла, в котором начинает обрабатываться 8-ой байт с конца (что происходит за две пересылки минус один период CLK до пересылки, содержащей этот байт). Если используется внешний контроллер DMA, и не выбрана опция Early DMA Assert (см. раздел 1.3.12.2.7.3.2.4), то линия запроса переходит в состояние с низким уровнем в конце цикла, в котором идентифицирован конец пересылки (что происходит через один цикл CLK после предпоследней пересылки). Линия запроса также переходит в состояние с низким уровнем, если CPU сбрасывает бит RxPktRdy (RxCSRL.D0).

Поведение линий запроса DMA для оконечных точек приемника, работающих в режиме 1 запроса DMA, аналогично, за исключением того, что линия запроса DMA переходит в состояние с высоким уровнем только когда был принят пакет максимального размера (этот размер установлен в регистре RxMaxP). Если принятый пакет имеет другой размер, уровень линии запроса DMA останется низким. Следует, однако, иметь в виду, что при переключении режима запроса DMA из режима 1 в режим 0 линия запроса будет активирована при наличии пакета в FIFO, чтобы этот ‘предварительно полученный’ пакет мог быть загружен.

Для оконечных точек передатчика, работающих как в режиме 0 запроса DMA, так и в режиме 1 запроса DMA, линия запроса переходит в состояние с высоким уровнем, когда FIFO оконечной точки может принять пакет данных. Обычно линия переходит в состояние с низким уровнем через один цикл CLK после того, как 8-ой с конца байт TxMaxP загружен в FIFO, но если используется внешний контроллер DMA, и не выбрана опция Early DMA Assert (см. раздел 1.3.12.2.7.3.2.4), то линия запроса переходит в состояние с низким уровнем через один цикл CLK

после загрузки всех байт TxMaxP. Линия запроса также переходит в состояние с низким уровнем, если CPU сбрасывает бит TxPktRdy (TxCSRL.D0).

Примечание: При работе в режиме ведущего устройства если устанавливается бит RxStall (TxCSRL.D5) или бит Error (TxCSRL.D2) после трех неудачных попыток передать пакет, линия запроса DMA будет блокирована до сброса бита RxStall/Error.

Выбранный режим также влияет на генерацию прерываний оконечных точек (если они разрешены). В режиме 0 запроса DMA прерывание не генерируется, когда пакеты получены, но генерируется прерывание соответствующей оконечной точки, чтобы загрузить все пакеты. В режиме 1 запроса DMA прерывание оконечной точки блокируется, за исключением прерывания после получения короткого пакета (то есть, меньшего, чем RxMaxP байт).

Условия, при которых генерируются прерывания оконечных точек передатчика и приемника, приведены в таблицах:

Таблица 1.721

EPInterrupt при установке RxPktRdy		
DMAReqEnab	DMAReqMode	Прерывание генерируется?
0	X	Да
1	0	Нет
1	1	Только при коротком пакете

Таблица 1.722

EPInterrupt при установке TxPktRdy		
DMAReqEnab	DMAReqMode	Прерывание генерируется?
0	X	Да
1	0	Да
1	1	Нет

Режим 0 запроса DMA может с равным успехом использоваться для поточных, изохронных пересылок и пересылок по прерыванию. Если оконечная точка сконфигурирована для изохронных пересылок, при использовании DMA всегда будет выбираться режим 0 запроса DMA.

Режим 1 запроса DMA особенно эффективен при пересылке больших блоков данных к поточной оконечной точке. Протокол USB требует, чтобы такие пакеты разделялись на серии пакетов максимальной длины, допустимой для оконечной точки (512 для высокой скорости, 64 байта для средней скорости соединения). Следует учитывать, что для правильной генерации прерываний в Tx/RxMaxP следует указывать четное число байт. Режим 1 запроса DMA может использоваться – с должным образом запрограммированным контроллером DMA – для того, чтобы исключить издержки от необходимости прерывания процессора после каждого отдельного пакета: вместо этого процессор прерывается только после завершения пересылки. В некоторых случаях блок пересылаемых данных будет содержать предварительно определенное число этих пакетов, которые управляющая программа ‘считает’ в процессе пересылки. В других случаях размер последнего пакета серии меньше максимального, и приемник может использовать этот ‘короткий’ пакет для указания конца пересылки. (Если общий размер пересылки в точности кратен максимальному размеру пакета, программа пересылки должна передать нулевой пакет, который обнаружит приемник.)

Примечание: В режиме 1 запроса DMA для правильной генерации прерываний в Tx/RxMaxP следует указывать четное число байт.

Дополнительная информация по использованию DMA для поточных передач приведена в разделе 22.3.

Пересылки DMA могут быть 8-битовыми, 16-битовыми или 32-битовыми – по требованию. Однако все пересылки, относящиеся к одному пакету (за исключением последней) должны иметь одинаковую ширину, чтобы данные были организованы по байтам, словам или словам удвоенной длины. Последняя пересылка может содержать меньше байт, чем предыдущие, чтобы завершить пересылку с нечетным количеством байт или слов.

Примечание: Перед переключением режима запроса DMA запрос DMA должен быть запрещен. В частности, бит DMAReqMode в регистре TxCSRH не следует сбрасывать в ноль ни в том же цикле, когда сбрасывается соответствующий бит DMAReqEnab, ни раньше.

1.3.12.2.7.12 ОПЦИОНАЛЬНЫЙ КОНТРОЛЛЕР DMA

Опционально MUSBMHDRC может включать многоканальный контроллер DMA, в котором конфигурируются до 8 каналов.

Этот контроллер DMA поддерживает два режима работы, которые называются режимом 0 DMA и режимом 1 DMA, и может обрабатывать пакет размером до 8К (для использования совместно с опциями разделения поточного пакета приемника и объединения поточного пакета передатчика, описанных в разделах (8.4.1.4 и 1.3.12.2.7.3.13.1.4). Кроме того, контроллер можно запрограммировать на выполнение пересылок с использованием инкрементальных пакетов (incrementing burst) из 4/8/16 слов данных INCR4, INCR8 и INCR16 вместо пакетов неопределенной длины.

При работе в режиме 0 DMA контроллер DMA можно запрограммировать только на загрузку/выгрузку одного пакета, так что вмешательство процессора требуется для каждого пакета, пересылаемого по USB. Этот режим может использоваться с любой оконечной точкой при контрольных, поточных, изохронных транзакциях и транзакциях по прерыванию (то есть, включая оконечную точку 0).

При работе в режиме 1 DMA контроллер DMA можно запрограммировать на загрузку/выгрузку полного пакета поточной пересылки (который может состоять из множества пакетов). После начальной настройки контроллер DMA будет загружать/выгружать все пакеты пересылки, прерывая процессор только по завершению пересылки. Режим 1 DMA может применяться только в тех оконечных точках, которые используют поточные транзакции.

Режим работы каждого канала может программироваться индивидуально.

1.3.12.2.7.12.1 РЕГИСТРЫ DMA

Контроллер DMA имеет один регистр прерывания, указывающий, какой канал ожидает прерывания, и набор из трех управляющих регистров для каждого указанного в конфигурации канала. Полное описание регистров DMA приведено в разделе описания регистров 1.3.12.2.7.3.8.5.

Адрес*	Регистр	Описание
200h	DMA_INTR	Регистр прерывания DMA.
204h+ (n-1)*10h	DMA_CNTL	Управляющий регистр DMA.
208h + (n-1)*10h	DMA_ADDR	Адресный регистр DMA.
20Ch + (n-1)*10h	DMA_COUNT	Регистр счетчик DMA.

*n = номер канала от 1 до 8

1.3.12.2.7.12.2 ЦИКЛЫ ШИНЫ DMA

Контроллер DMA использует инкрементные пакеты на шине AHB. Он начинает новый пакет при первом захвате шины (либо в начале пакета USB, либо в процессе восстановления прерванной пересылки пакета), когда адрес AHB начинает новый 1K блок. Примечание: Требование начала нового пакета на границах 1K выполняется контроллером автоматически. Пользователю нет нужды учитывать эти границы при программировании контроллера DMA.

Эти пакеты могут состоять из 4, 8, 16 слов или иметь неопределенную длину, в зависимости от того, как запрограммирован канал DMA, от размера пересылаемого пакета и от расположения относительно следующей границы 1K. Биты D10–9 регистра CNTL выбирают пакетный режим 0 – 3, который в свою очередь определяет используемые типы пакетов (см. выше, раздел 17.1). Например, выбор пакетного режима 2 позволяет использовать пакеты из 8 слов (INCR8), из 4 слов (INCR4) и пакеты неопределенной длины, но не из 16 слов (INCR16).

Не установлено ограничений на выбор пакетного режима как в режиме 1 DMA, так и в режиме 0 DMA.

Каждая пересылка пакета обычно выполняется при помощи пересылки слов (32 бита), но в конце пакетной пересылки может присутствовать дополнительный байт или половина слова. Поскольку начальный адрес (записанный в регистр DMA ADDR (n)) должен указывать на начало слова, пакетная пересылка начнется с пересылки слова, но в конце может быть добавлена пересылка байта или половины слова, чтобы переслать остаток. Шина AHB поддерживает разделение транзакций и повторные попытки.

1.3.12.2.7.12.3 ОШИБКИ ШИНЫ

Если во время обращения контроллера DMA к памяти возникает ошибка шины, контроллер немедленно завершает пересылку DMA и генерирует прерывание процессора установкой бита Bus Error (D8) регистра CNTL.

Примечание: На генерирование прерывания не влияет установка бита Interrupt Enable в регистре DMA CNTL (бит 3). Прерывание будет сгенерировано и при CNTL.D3 = 0.

1.3.12.2.7.12.4 ПЕРЕСЫЛКА ПАКЕТОВ

Использование встроенного контроллера DMA для доступа к FIFO буферам MUSBMHDRC требует соответствующего программирования как контроллера DMA, так и оконечной точки MUSBMHDRC. Возможно большое количество вариантов. В следующих разделах подробно описаны стандартные начальные настройки, используемые для основных действий по пересылке отдельных пакетов и группы пакетов.

1.3.12.2.7.12.4.1 ОТДЕЛЬНЫЙ ПАКЕТ: ОКОНЕЧНАЯ ТОЧКА ПРИЕМНИКА

Пересылка отдельных пакетов обычно выполняется в режиме 0 DMA.

Для этого оконечная точка приемника должна быть запрограммирована следующим образом:

- Соответствующий бит разрешения прерывания в регистре IntrRxE установлен в 1.
- Бит DMAReqEnab (D13) соответствующего регистра RxCSR установлен в 0. (Примечание: Для данной операции нет необходимости устанавливать MUSBMHDRC в режим поддержки DMA).

При получении пакета MUSBMHDRC генерирует прерывание соответствующей оконечной точки. Далее процессор должен запрограммировать выбранный канал контроллера DMA следующим образом:

- ADDR: Адрес памяти для хранения пакета
- COUNT: Размер пакета (определяемый чтением регистра MUSBMHDRC RxCount)
- CNT: DMA Enable (D0) =1; Direction (D1) =0; DMA Mode (D2) =0; Interrupt Enable (D3) =1; Required Burst Mode (D10–9).

Затем контроллер DMA запросит захват шины и перешлет пакет в память. Закончив пересылку, он сгенерирует прерывание DMA (установив низкий уровень сигнала DMA_NINT). Затем процессор должен сбросить бит RxPktRdy в регистре RxCSR MUSBMHDRC.

1.3.12.2.7.12.4.2 ОТДЕЛЬНЫЙ ПАКЕТ: ОКОНЕЧНАЯ ТОЧКА ПЕРЕДАТЧИКА

Для выполнения этой операции в режиме 0 DMA оконечная точка передатчика должна быть запрограммирована следующим образом:

- Соответствующий бит разрешения прерывания в регистре IntrTxE установлен в 1.
- Бит DMAReqEnab (D12) соответствующего регистра TxCSR установлен в 0. (Примечание: Для данной операции нет необходимости устанавливать MUSBMHDRC в режим поддержки DMA).

Когда FIFO буфер MUSBMHDRC становится доступен, MUSBMHDRC прерывает работу процессора, сгенерировав прерывание соответствующей оконечной точки передатчика. Далее процессор должен запрограммировать контроллер DMA следующим образом:

- ADDR: Адрес буфера для передачи пакета
- COUNT: Размер передаваемого пакета
- CNTL: DMA Enable (D0) =1; Direction (D1) =1; DMA Mode (D2) =0; Interrupt Enable (D3) =1; Required Burst Mode (D10–9).

Затем контроллер DMA запросит захват шины и перешлет пакет в FIFO буфер MUSBMHDRC. Закончив пересылку, он сгенерирует прерывание DMA (установив низкий уровень сигнала DMA_NINT). Затем процессор должен сбросить бит TxPktRdy в регистре TxCSR MUSBMHDRC.

1.3.12.2.7.12.4.3 ГРУППА ПАКЕТОВ: ОКОНЕЧНАЯ ТОЧКА ПРИЕМНИКА

Пересылка группы пакетов обычно выполняется в режиме 1 DMA.

Для приема группы пакетов в режиме 1 DMA контроллер DMA должен быть запрограммирован следующим образом:

- ADDR: Адрес буфера для хранения пересылки
- COUNT: Максимальный размер буфера данных
- CNTL: DMA Enable (D0) =1; Direction (D1) =0; DMA Mode (D2) =1; Interrupt Enable (D3) =1; Required Burst Mode (D10–9).

Оконечная точка приемника MUSBMHDRC должна быть запрограммирована следующим образом:

- Соответствующий бит разрешения прерывания в регистре IntrRxE установлен в 1.
- Установлены в 1 биты AutoClear (D15), DMAReqEnab (D13) и DMAReqMode (D11) соответствующего регистра RxCSR. В режиме ведущего устройства бит AutoReq (D14) также должен быть установлен в 1, а в регистре RqPktCount необходимо запрограммировать количество пакетов в пересылке.

При получении MUSBMHDRC каждого пакета контроллер DMA запросит захват шины и перешлет пакет в память. При установленном бите AutoClear MUSBMHDRC автоматически сбросит бит RxPktRdy.

В режиме ведомого устройства, или когда значение RqPktCount равно нулю, этот процесс будет продолжаться автоматически, пока MUSBMHDRC не получит ‘короткий пакет’ (меньше, чем пакет максимального размера для оконечной точки), указывающий на конец пересылки. Этот ‘короткий пакет’ не будет пересылаться контроллером DMA: вместо этого MUSBMHDRC прервет работу процессора, сгенерировав прерывание соответствующей оконечной точки. Затем процессор может считать регистр MUSBMHDRC RxCount, чтобы определить размер ‘короткого пакета’ и либо выгрузить его вручную, либо перепрограммировать контроллер DMA в режим 0, чтобы выгрузить этот пакет.

В режиме ведущего устройства, когда AutoReq установлен, а значение RqPktCount не равно нулю, ядро может уменьшать значение в регистре RqPktCount после каждого запроса. Когда значение уменьшается с 1 до 0, бит AutoReq сбрасывается, чтобы предотвратить любые следующие попытки пересылки.

Регистр ADDR контроллера DMA должен инкрементироваться по мере выгрузки пакетов, чтобы процессор мог определить размер пересылки, сравнивая текущее значение ADDR с начальным адресом буфера памяти.

Примечание: Если размер пересылки превышает размер буфера данных, контроллер DMA остановит выгрузку FIFO и прервет работу процессора при помощи линии DMA_NINT.

1.3.12.2.7.12.4.4 ГРУППА ПАКЕТОВ: ОКОНЕЧНАЯ ТОЧКА ПЕРЕДАТЧИКА

Для выполнения этой операции в режиме 1 DMA контроллер DMA должен быть запрограммирован следующим образом:

- ADDR : Адрес буфера для передачи блока
- COUNT : Размер блока данных
- CNTL: DMA Enable (D0) =1; Direction (D1) =1; DMA Mode (D2) =1; Interrupt Enable (D3) =1; Required Burst Mode (D10–9).

Оконечная точка передатчика MUSBMHDRC должна быть запрограммирована следующим образом:

- Соответствующий бит разрешения прерывания в регистре IntrRxE установлен в 1 (просто для того, чтобы могли быть обнаружены ошибки).
- Установлены в 1 биты AutoSet (D15), DMAReqEnab (D12) и DMAReqMode (D10) соответствующего регистра TxCSR.

Когда FIFO буфер MUSBMHDRC становится доступен, MUSBMHDRC DMA запросит захват шины и перешлет пакет в FIFO. При установленном бите With AutoSet MUSBMHDRC автоматически установит бит TxPktRdy. Этот процесс продолжится, пока весь блок данных не будет передан в MUSBMHDRC. Затем контроллер DMA прервет работу процессора, установив низкий уровень сигнала DMA_NINT. Если последний пакет, который необходимо загрузить, меньше, чем пакет максимального размера для оконечной точки, бит TxPktRdy для этого пакета установлен не будет: поэтому процессор должен ответить на прерывание DMA установкой бита TxPktRdy, чтобы разрешить отправку последнего ‘короткого пакета’. Если последний пакет, который необходимо загрузить, имеет максимальный размер, то дальнейшие действия зависят от того, управляется ли пересылка приложением, таким как подпрограмма массовой памяти в системе Windows, которая ведет подсчет отдельных отправленных пакетов. Если пересылка не управляется таким способом, на прерывание DMA процессор все равно должен ответить установкой бита TxPktRdy. Это приводит к отправке нулевого пакета, который программа приемника интерпретирует как указание на конец пересылки.

1.3.12.2.7.13 СОБЫТИЯ VBUS

Спецификация USB On-The-Go определяет ряд порогов, на которые должны реагировать устройства, связанные между собой в режиме точка-точка.

- VBus Valid (должно быть больше, чем 4,4 and 4,75 В)
- Session Valid для устройства ‘A’ (должно быть между 0,8 В and 2,1 В)
- Session End (должно быть между 0,2 В and 0,8 В)
-

(Реальные пороги, используемые в конкретном устройстве, задаются серией внешних по отношению к ядру MUSBMHDRC компараторов, которые устанавливают высокий или низкий уровень соответствующих входных сигналов VBUSVALID, AVALID и SESSEND в зависимости от уровня VBus.)

Какие из этих порогов являются критичными, а также необходимая реакция CPU, управляющего MUSBMHDRC, зависит от роли устройства, ‘A’ или ‘B’, и обстоятельств, при которых произошло это событие. Необходимые действия описаны ниже.

1.3.12.2.7.13.1 ДЕЙСТВИЯ УСТРОЙСТВА ‘A’

VBus > VBus Valid во время сеанса, инициированного MUSBMHDRC (т. е. Vbus[1:0] (DevCtl.[D4:D3]) = 11b, бит Session (DevCtl.D0) установлен). Когда VBus превышает VBus Valid, MUSBMHDRC выбирает режим ведущего устройства и ждет подключения ведомого устройства. Затем он генерирует прерывание Connect (IntrUSB.D4). CPU должен сбросить подключенное устройство ‘B’ и выполнить его нумерацию.

VBus > Session Valid во время сеанса, инициированного устройством ‘B’ (т. е. Vbus[1:0] (DevCtl.[D4:D3]) = 10b, бит Session (DevCtl.D0) сброшен). Когда VBus превысит Session Valid, MUSBMHDRC сгенерирует прерывание Session Request (IntrUSB.D6). CPU должно

установить бит Session. MUSBMHDRC затем либо останется в режиме ведущего устройства, либо переключится в режим ведомого устройства, в зависимости от того, подключен ли нагрузочный резистор в устройстве ‘B’.

VBus ниже VBus Valid при установленном бите Session (т. е. Vbus[1:0] (DevCtl.[D4:D3]) □□11b, бит Session (DevCtl.D0) установлен). Это указывает на проблему с уровнем питания VBus. Например, напряжение аккумулятора может быть слишком низким, чтобы поддерживать VBus Valid. Или устройство ‘B’ может потреблять больший ток, чем способно обеспечить устройство ‘A’. В обоих случаях MUSBMHDRC автоматически прервет сеанс и сгенерирует прерывание VBus Error (IntrUSB.D7).

1.3.12.2.7.13.2 ДЕЙСТВИЯ УСТРОЙСТВА ‘B’

VBus > Session Valid (т. е. Vbus[1:0] (DevCtl.[D4:D3]) = 10b, бит Session (DevCtl.D0) сброшен). MUSBMHDRC установит бит Session и установит низкий уровень выходного сигнала DPPULLDOWN, чтобы отключить нагрузочный резистор на линии D+.

VBus < Session Valid при установленном бите Session (т. е. Vbus[1:0] (DevCtl.[D4:D3]) = 01b, бит Session (DevCtl.D0) установлен). Это событие указывает на отключение питания устройства ‘A’ (или его отсоединение). MUSBMHDRC сбросит бит Session (DevCtl.D0) и сгенерирует прерывание Disconnect (IntrUSB.D5). CPU должно завершить сеанс.

VBus < Session End (т. е. Vbus[1:0] (DevCtl.[D4:D3]) = 00b). Это состояние, при котором устройство ‘B’ может инициировать запрос сеанса. Если бит Session (DevCtl.D0) установлен, то через 2 мс присутствия на шине SE0 MUSBMHDRC начнет SRP подачей импульса на линии данных, а затем на VBus (установив высокий уровень CHRGVBUS).

1.3.12.2.7.14 ДИНАМИЧЕСКОЕ ОПРЕДЕЛЕНИЕ РАЗМЕРА FIFO

При необходимости MUSBMHDRC можно сконфигурировать так, чтобы он имел один общий FIFO буфер размером 128, 256, 512, 1K … 64K байт, разные области которого затем в процессе инициализации MUSBMHDRC будут распределены по оконечным точкам. (См. раздел 1.3.12.2.7.3.3.18).

Примечание: Настоятельно рекомендуется использовать только этот режим, когда MUSBMHDRC используется в составе устройства, которому в разных ситуациях требуется разный размер FIFO. Если размеры FIFO буферов не будут изменяться, то лучше установить их размер посредством стандартных опций конфигурации, поскольку опция динамического определения размера FIFO существенно увеличивает размер ядра и требует более сложной аппаратной части для ее поддержки.

Распределение области FIFO по разным оконечным точкам требует, чтобы для каждой оконечной точки приемника и передатчика были указаны:

- Начальный адрес FIFO в блоке памяти RAM
- Максимальный размер пакета, который будет поддерживаться
- Требуется ли двойная буферизация

(Два последних пункта вместе определяют размер области, которую требуется выделить в FIFO.)

Эта информация может указываться при помощи следующих четырех регистров, которые добавляются к области индексных регистров MUSBMHDRC при выборе опции динамического определения размера FIFO:

РЕГИСТРЫ MUSBMHDRC		
АДРЕС	НАИМЕНОВАНИЕ	ОПИСАНИЕ
62	TxFIFOsz	Размер FIFO оконечной точки передатчика
63	RxFIFOsz	Размер FIFO оконечной точки приемника
64,65	TxFIFOadd	Адрес FIFO оконечной точки передатчика
66,67	RxFIFOadd	Адрес FIFO оконечной точки приемника

Подробное описание этих регистров приводится в разделе 1.3.12.2.7.3.10.

Примечание: (i) Опция динамического определения размеров FIFO применима только к оконечным точкам 1 – 15. FIFO оконечной точки 0 имеет фиксированный размер (64 байта) и фиксированное расположение (начальный адрес 0). (ii) Аппаратная часть (и разработчик системы) должны позаботиться о том, чтобы все оконечные точки передатчика и приемника, активные в текущей конфигурации USB, имели блок памяти RAM, выделенный исключительно для них и имеющий размер не меньше максимального размера пакета, установленного для оконечной точки.

1.3.12.2.7.14.1 УПРАВЛЕНИЕ СЕАНСОМ

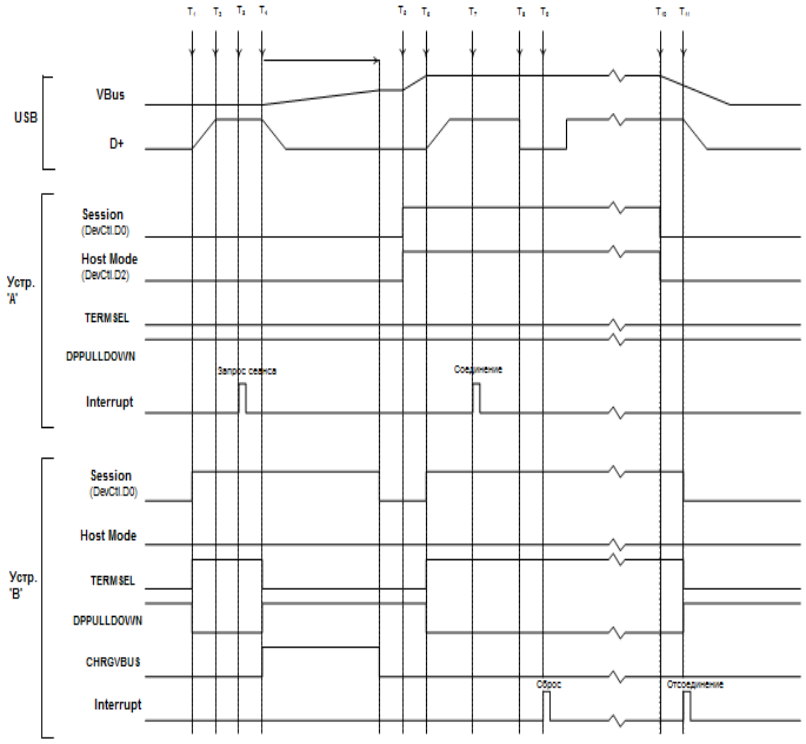


Рисунок 1.270

T1 Аппаратная часть устройства ‘B’ запрашивает начало сеанса, установив бит Session (DevCtl.D0).

T2 Устройство ‘B’ подает импульс линию данных.

T3 Устройство ‘A’ обнаруживает импульс на линии данных и генерирует прерывание Session Request (IntrUSB.D6).

T4 Устройство ‘B’ подает импульс на VBus (устанавливая высокий уровень сигнала CHRGVBUS).

T5 Аппаратная часть устройства ‘A’ начинает сеанс, устанавливая Session (DevCtl.D0).

T6 Устройство ‘B’ обнаруживает начало сеанса и устанавливает высокий уровень сигнала TERMSEL (может использоваться для подключения нагрузочного резистора к D+).

T7 Устройство ‘A’ обнаруживает подключение нагрузочного резистора и генерирует прерывание Connect (IntrUSB.D4).

T8 Аппаратная часть устройства ‘A’ сбрасывает устройство ‘B’, а затем начинает транзакции.

T9 Устройство ‘B’ обнаруживает сигнал сброса и генерирует прерывание Reset (IntrUSB.D2).

T10 Аппаратная часть устройства ‘A’ заканчивает сеанс, сбрасывая бит Session.

T11 Устройство ‘B’ обнаруживает окончание сеанса и генерирует прерывание Disconnect (IntrUSB.D5).

1.3.12.2.7.14.2 СОГЛАСОВАНИЕ ВЕДУЩЕГО УСТРОЙСТВА

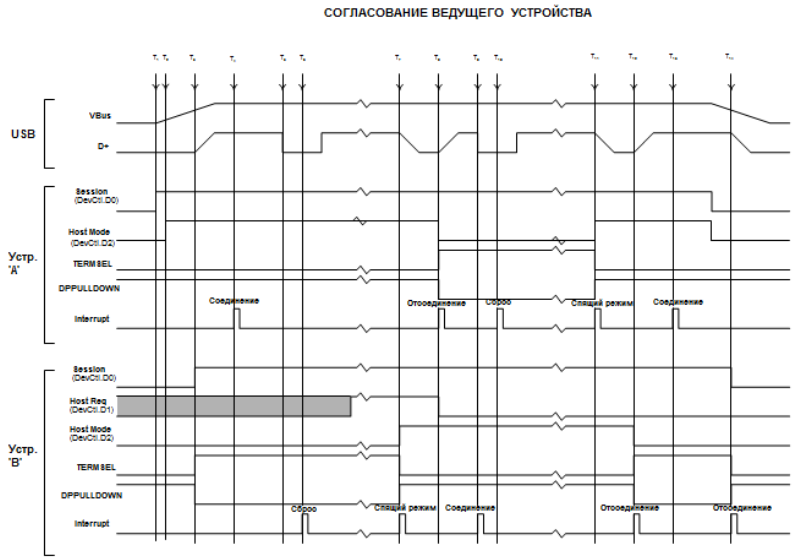


Рисунок 1.271

T1 Аппаратная часть устройства ‘A’ начинает сеанс, установив бит Session (DevCtl.D0).

T2 Устройство ‘A’ проверяет IDDIG (после ожидания WTID (регистр LinkInfo)) и устанавливает бит HostMode (DevCtl.D3).

T3 Устройство ‘B’ обнаруживает начало сеанса и устанавливает высокий уровень TERMSEL (может использоваться для подключения нагрузочного резистора к D+).

T4 Устройство ‘A’ обнаруживает присутствие устройства ‘B’ и генерирует прерывание Connect (IntrUSB.D4).

T5 Аппаратная часть устройства ‘A’ сбрасывает устройство ‘B’, а затем начинает транзакции.

T6 Устройство ‘B’ обнаруживает сигнал сброса и вырабатывает прерывание Reset (IntrUSB.D2).

T7 Устройство ‘B’ обнаруживает на шине сигнал спящего режима, когда установлен бит Host Req (DevCtl.D1), генерирует прерывание Suspend (IntrUSB.D0) и устанавливает низкий уровень TERMSEL, отключая нагрузочный резистор от D+.

T8 Устройство ‘A’ обнаруживает отключение, генерирует прерывание Disconnect (IntrUSB.D5) и устанавливает высокий уровень TERMSEL (что снова может использоваться для подключения нагрузочного резистора к D+).

T9 Устройство ‘B’ обнаруживает устройство ‘A’, генерирует прерывание Connect (IntrUSB.D4) и инициирует сброс устройства ‘A’. (Аппаратная часть устройства ‘B’ может сбросить бит Reset (Power.D3) через 20 мс и начать транзакции.)

T10 Устройство ‘A’ обнаруживает сигнал сброса и генерирует прерывание Reset (IntrUSB.D2).

T11 Устройство ‘A’ обнаруживает на шине сигнал спящего режима, генерирует прерывание Suspend (IntrUSB.D0) и устанавливает низкий уровень TERMSEL, отключая нагрузочный резистор.

T12 Устройство ‘B’ обнаруживает отключение, генерирует прерывание Disconnect (IntrUSB.D5) устанавливает высокий уровень своего сигнала TERMSEL, подключая свой нагрузочный регистр к D+.

T13 Устройство ‘A’ обнаруживает устройство ‘B’, затем завершает сеанс, сбрасывая бит Session.

T14 Устройство ‘B’ обнаруживает окончание сеанса и генерирует прерывание Disconnect (IntrUSB.D5).

1.3.12.2.7.15 УПРАВЛЯЮЩИЕ ТРАНЗАКЦИИ

1.3.12.2.7.15.1 УПРАВЛЯЮЩИЕ ТРАНЗАКЦИИ В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Оконечная точка 0 – это главная управляющая оконечная точка ядра. Поэтому подпрограммы обслуживания оконечной точки 0 гораздо сложнее, чем для остальных оконечных точек.

Программное обеспечение должно обрабатывать все запросы стандартного устройства, которые могут быть отправлены или получены через оконечную точку 0. Они описаны в Universal Serial Bus Specification, версия 2.0, глава 9. Протокол этих запросов устройства предполагает в пересылке различное количество и типы транзакций. Для учета этого CPU должен работать в режиме управляющего конечного автомата, чтобы обеспечить их декодирование и обработку.

Запросы стандартного устройства, получаемые ведомым устройством USB, могут быть разделены на три категории: запросы с данными нулевой длины (в которых информация включена в команду), запросы записи (в которых за командой следуют дополнительные данные)

и запросы чтения (в которых от ведомого устройства требуется отправить данные ведущему устройству).

В этом разделе рассматривается последовательность действий, которые должна выполнить программа, чтобы обработать эти разные типы запросов устройства.

Примечание: Пакет установки любого запроса стандартного устройства должен включать 8-битовую команду. Любой пакет установки, содержащий поле команды, отличное от 8 бит, будет автоматически отвергнут ядром MUSBMHDRC.

1.3.12.2.7.15.1.1 ЗАПРОСЫ С ДАННЫМИ НУЛЕВОЙ ДЛИНЫ

В запросах с данными нулевой длины вся информация содержится в 8-байтовой команде и не требуют передачи дополнительных данных. Примеры запросов стандартного устройства «с нулевыми данными»: SET_FEATURE, CLEAR_FEATURE, SET_ADDRESS, SET_CONFIGURATION, SET_INTERFACE.

Последовательность событий, как и при всех запросах, начинается после того, как программа получит прерывание оконечной точки 0. Бит RxPktRdy (CSR0L.D0) также должен быть установлен. 8-байтовая команда затем должна быть считана из FIFO оконечной точки 0, расшифрована и соответствующим образом исполнена. Например, по команде SET_ADDRESS 7-битовое значение адреса, содержащееся в команде, должно быть записано в регистр FAddr.

После этого должен быть записан регистр CSR0, чтобы установить бит ServicedRxPktRdy (D6) (указывающий, что команда была считана из FIFO), а также установить бит DataEnd (D3) (указывающий, что для этого запроса больше не ожидается данных).

Когда ведущее устройство переходит в стадию статуса запроса, генерируется второе прерывание оконечной точки 0, указывающее на завершение запроса. От программы больше не требуется никаких действий: второе прерывание просто служит подтверждением, что запрос успешно выполнен.

При поступлении нераспознаваемой команды, или если по какой-либо причине команда не может быть исполнена после расшифровки, необходимо выполнить запись в регистр CSR0, чтобы установить бит ServicedRxPktRdy (D6) и установить бит SendStall (D5). Когда ведущее устройство переходит в стадию статуса запроса, MUSBMHDRC вырабатывает сигнал STALL, чтобы указать ведущему устройству, что запрос не выполнен. Генерируется второе прерывание оконечной точки 0, и устанавливается бит SentStall (CSR0L.D2).

Если ведущее устройство передает данные после установки бита DataEnd, то MUSBMHDRC вырабатывает сигнал STALL. Генерируется прерывание оконечной точки 0, и устанавливается бит SentStall (CSR0L.D2).

1.3.12.2.7.15.1.2 ЗАПРОСЫ ЗАПИСИ

Запросы записи включают дополнительные пакет (или пакеты) данных, передаваемые от ведущего устройства после 8-байтовой команды. Пример такого запроса стандартного устройства: SET_DESCRIPTOR.

Последовательность событий, как и при всех запросах, начинается после того, как программа получит прерывание оконечной точки 0. Бит RxPktRdy (CSR0L.D0) также должен

быть установлен. 8-байтовая команда затем должна быть считана из FIFO оконечной точки 0 и расшифрована.

Как и в случае запроса с нулевыми данными, после этого должен быть записан регистр CSR0, чтобы установить бит ServicedRxPktRdy (D6) (указывающий, что команда была считана из FIFO), но в данном случае бит DataEnd (D3) не устанавливается (что указывает на ожидание дополнительных данных).

После прихода второго прерывания оконечной точки 0 необходимо считать регистр CSR0, чтобы проверить состояние оконечной точки. Бит The RxPktRdy (CSR0L.D0) должен быть установлен, чтобы указать на принятый пакет данных. Для определения размера этого пакета требуется считать регистр COUNT0. Затем пакет данных может быть считан из FIFO оконечной точки 0.

Если длина данных, связанных с запросом (указана в поле команды wLength) больше максимального размера пакета для оконечной точки 0, то будут переданы следующие пакеты данных. В этом случае необходимо выполнить запись в регистр CSR0, чтобы установить бит ServicedRxPktRdy, но бит DataEnd устанавливать не следует.

После получения всех ожидаемых пакетов данных необходимо выполнить запись в регистр CSR0, чтобы установить бит ServicedRxPktRdy и установить бит DataEnd (указывающий, что больше данных не ожидается).

Когда ведущее устройство переходит в стадию статуса запроса, генерируется еще одно прерывание оконечной точки 0, указывающее на завершение запроса. От программы больше не требуется никаких действий: второе прерывание просто служит подтверждение, что запрос успешно выполнен.

При поступлении нераспознаваемой команды, или если по какой-либо причине команда не может быть исполнена после расшифровки, необходимо выполнить запись в регистр CSR0, чтобы установить бит ServicedRxPktRdy (D6) и установить бит SendStall (D5). Когда ведущее устройство отправит следующие данные, MUSBMHDRC вырабатывает сигнал STALL, чтобы указать ведущему устройству, что запрос не выполнен. Генерируется прерывание оконечной точки 0, и устанавливается бит SentStall (CSR0L.D2).

Если ведущее устройство передает данные после установки бита DataEnd, то MUSBMHDRC вырабатывает сигнал STALL. Генерируется прерывание оконечной точки 0, и устанавливается бит SentStall (CSR0L.D2).

1.3.12.2.7.15.1.3 ЗАПРОСЫ ЧТЕНИЯ

Запросы чтения включают дополнительные пакет (или пакеты) данных, передаваемые от функции ведущему устройству после 8-байтовой команды. Примеры таких запросов стандартного устройства: GET_CONFIGURATION, GET_INTERFACE, GET_DESCRIPTOR, GET_STATUS, SYNCH_FRAME.

Последовательность событий, как и при всех запросах, начинается после того, как программа получит прерывание оконечной точки 0. Бит RxPktRdy (CSR0L.D0) также должен быть установлен. 8-байтовая команда затем должна быть считана из FIFO оконечной точки 0 и расшифрована. После этого должен быть записан регистр CSR0, чтобы установить бит ServicedRxPktRdy (D6) (указывающий, что команда была считана из FIFO).

Данные которые должны быть переданы ведущему устройству, затем записываются в FIFO оконечной точки. Если длина данных, предназначенных для передачи, больше максимального размера пакета для оконечной точки 0, то в FIFO должен быть записан только пакет максимальной длины. Далее необходимо выполнить запись в регистр CSR0, чтобы установить бит TxPktRdy (D1) (указывающий наличие в FIFO пакета для передачи). После отправки пакета ведущему устройству будет сгенерировано еще одно прерывание оконечной точки 0, и следующий пакет данных может быть записан в FIFO.

После записи в FIFO последнего пакета необходимо выполнить запись в регистр CSR0, чтобы установить бит ServicedRxPktRdy и установить бит DataEnd (D3) (указывающий, что после этого пакета больше нет данных).

Когда ведущее устройство переходит в стадию статуса запроса, генерируется еще одно прерывание оконечной точки 0, указывающее на завершение запроса. От программы больше не требуется никаких действий: второе прерывание просто служит подтверждение, что запрос успешно выполнен.

При поступлении нераспознаваемой команды, или если по какой-либо причине команда не может быть исполнена после расшифровки, необходимо выполнить запись в регистр CSR0, чтобы установить бит ServicedRxPktRdy (D6) и установить бит SendStall (D5). Когда ведущее устройство получит данные, MUSBMHDRC вырабатывает сигнал STALL, чтобы указать ведущему устройству, что запрос не выполнен. Генерируется прерывание оконечной точки 0, и устанавливается бит SentStall (CSR0L.D2).

Если ведущее устройство передает данные после установки бита DataEnd (D3), то MUSBMHDRC вырабатывает сигнал STALL. Генерируется прерывание оконечной точки 0, и устанавливается бит SentStall (CSR0L.D2).

1.3.12.2.7.15.1.4 СОСТОЯНИЯ ОКОНЕЧНОЙ ТОЧКИ 0

Когда MUSBMHDRC работает в режиме ведомого устройства, для управления оконечной точкой 0 требуется три режима - IDLE, TX и RX – в соответствии с тремя разными фазами управляющей пересылки и состояниями, в который переходит оконечная точка 0 в разных фазах пересылки.

По умолчанию состояние после включения питания или сброса – IDLE.

Бит RxPktRdy (CSR0L.D0) устанавливается после того, как оконечная точка 0 в состоянии IDLE указывает на новый запрос устройства. После того, как запрос устройства выгрузится из FIFO, MUSBMHDRC расшифровывает дескриптор, чтобы выяснить наличие фазы пересылки данных, и если она присутствует, то направление фазы данных управляющей пересылки (чтобы установить направление FIFO).

В зависимости от направления фазы данных оконечная точка 0 переходит либо в состояние передатчика, либо в состояние приемника. Если фаза данных отсутствует, оконечная точка остается в состоянии IDLE, чтобы иметь возможность принять следующий запрос устройства.

Действия, которые должно выполнить CPU в разных фазах возможных пересылок (то есть, загрузка FIFO, установка TxPktRdy) представлены на диаграмме на следующей странице.

Следует учитывать, что MUSBMHDRC изменяет направление FIFO в зависимости от направления фазы данных независимо от CPU.

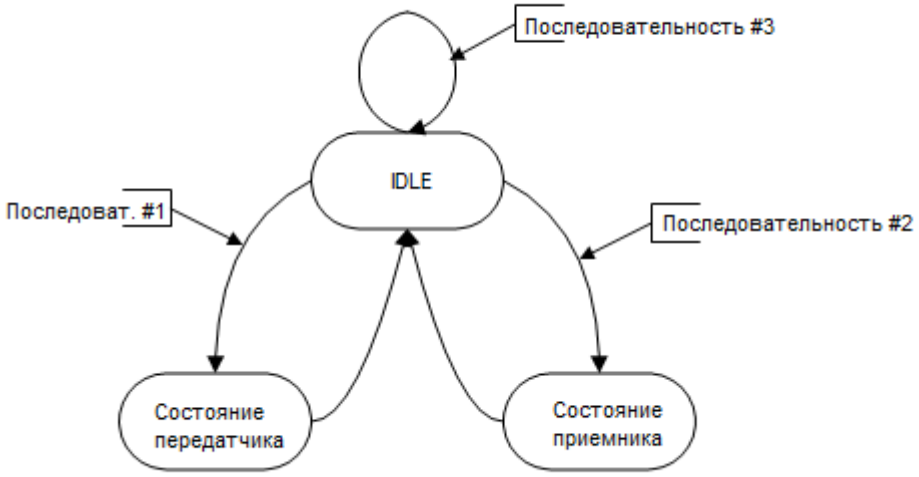
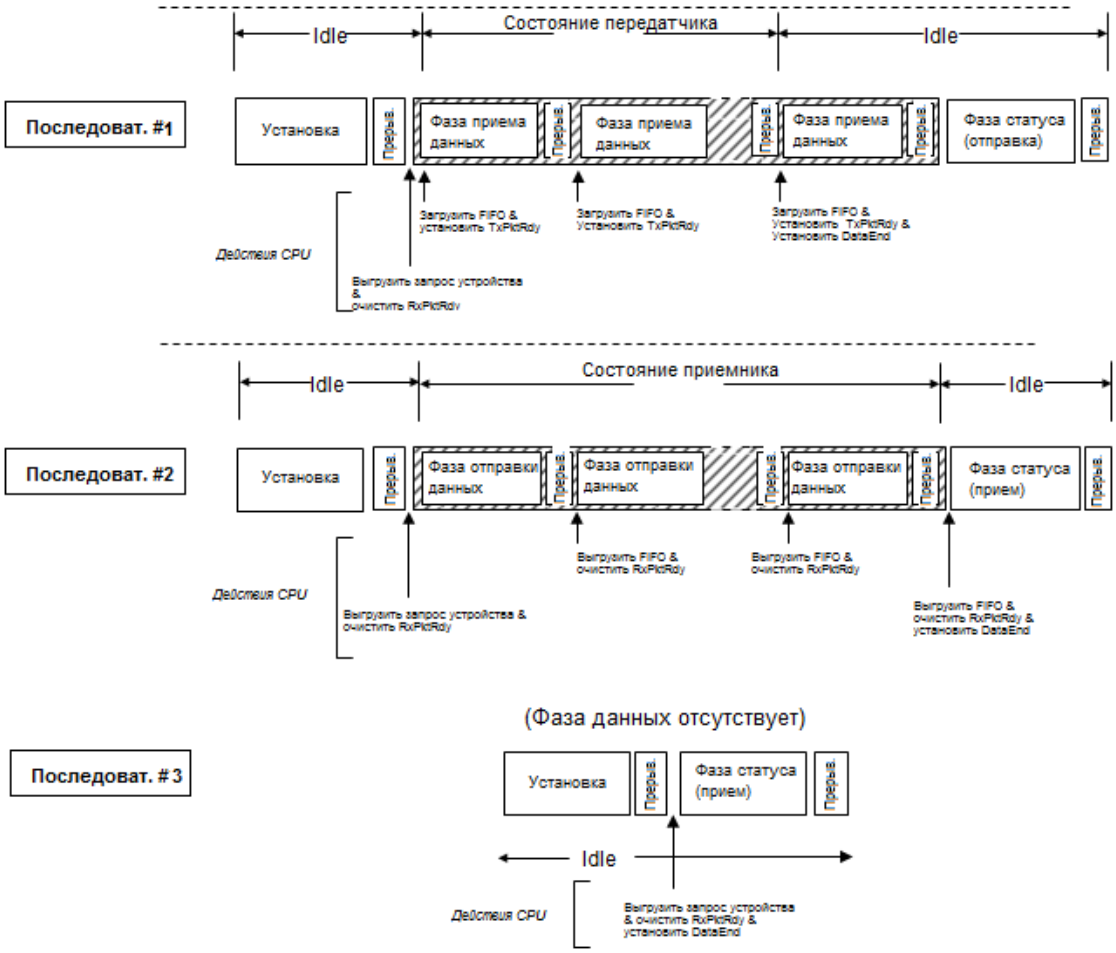


Рисунок 1.272 - Состояния оконечной точки 0 в режиме ведомого устройства



1.3.12.2.7.15.1.5 СЕРВИСНАЯ ПРОГРАММА ОКОНЕЧНОЙ ТОЧКИ 0 В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Прерывание оконечной точки 0 генерируется в следующих случаях:

- Ядро устанавливает бит RxPktRdy (CSR0L.D0) после получения допустимого маркера и записи данных в FIFO.
- Ядро сбрасывает бит TxPktRdy bit (CSR0L.D1) после того, как пакет данных из FIFO успешно передан ведущему устройству.
- Ядро устанавливает бит SentStall (CSR0L.D2) после завершения управляющей транзакции вследствие нарушения протокола.
- Ядро устанавливает бит SetupEnd (CSR0L.D4), поскольку управляющая пересылка завершилась до установки бита DataEnd (CSR0L.D3).

После начала сервисной программы оконечной точки 0 аппаратная часть должна сначала проверить, не закончилась ли текущая управляющая пересылка закончилась в результате состояния STALL или преждевременного прекращения управляющей пересылки. Если управляющая пересылка закончилась в результате состояния STALL, то должен быть установлен бит SentStall. Если управляющая пересылка закончилась в результате преждевременного прекращения управляющей пересылки, то должен быть установлен бит SetupEnd. В обоих

случаях аппаратная часть должна прервать обработку текущей управляющей пересылки и установить состояние IDLE.

После того, как аппаратная часть определила, что прерывание не было сгенерировано запрещенным состоянием шины, следующее действие определяется состоянием оконечной точки.

Если оконечная точка 0 находится в состоянии IDLE, единственная допустимая причина прерывания может быть результатом того, что ядро получает данные с шины USB. Сервисная программа должна проверить это, считав бит RxPktRdy (CSR0L.D0). Если этот бит установлен, то ядро получило пакет SETUP. Он должен быть выгружен из FIFO и расшифрован, чтобы определить необходимые действия ядра.

В зависимости от команды, содержащейся в пакете SETUP, оконечная точка перейдет в одно из трех состояний:

- Если команда представляет собой транзакцию одного пакета (SET_ADDRESS, SET_INTERFACE и т. п.) без фазы данных, то оконечная точка останется в состоянии IDLE.
- Если команда имеет фазу отправки данных (SET_DESCRIPTOR и т. п.), то оконечная точка перейдет в состояние приемника.
- Если команда имеет фазу данных IN (GET_DESCRIPTOR и т. п.), то оконечная точка перейдет в состояние передатчика.

Если оконечная точка 0 находится в состоянии передатчика, прерывание указывает, что ядро получило маркер чтения, и данные из FIFO были отправлены. Аппаратная часть должна отреагировать на это либо помещением новых данных в FIFO, если ведущее устройство по-прежнему ожидает данных2, либо установкой бита DataEnd, указывающего, что фаза данных завершена. После завершения фазы данных транзакции оконечная точка должна вернуться в состояние IDLE и ждать следующей управляющей транзакции.

Если оконечная точка 0 находится в состоянии приемника, прерывание указывает на полученный пакет данных. Аппаратная часть должна отреагировать на это выгрузкой полученных данных из FIFO. Затем аппаратная часть должна определить, все ли ожидаемые данные получены2. Если да, то аппаратная часть устанавливает бит DataEnd и возвращает оконечную точку 0 в состояние IDLE. Если получены не все данные, аппаратная часть должна установить бит ServicedRxPktRdy (CSR0L.D6), чтобы указать, что данные считаны из FIFO, а оконечная точка осталась в состоянии приемника.

Все управляющие транзакции включают поле, которое указывает объем данных, которые управляющее устройство ожидает принять или собирается отправить.

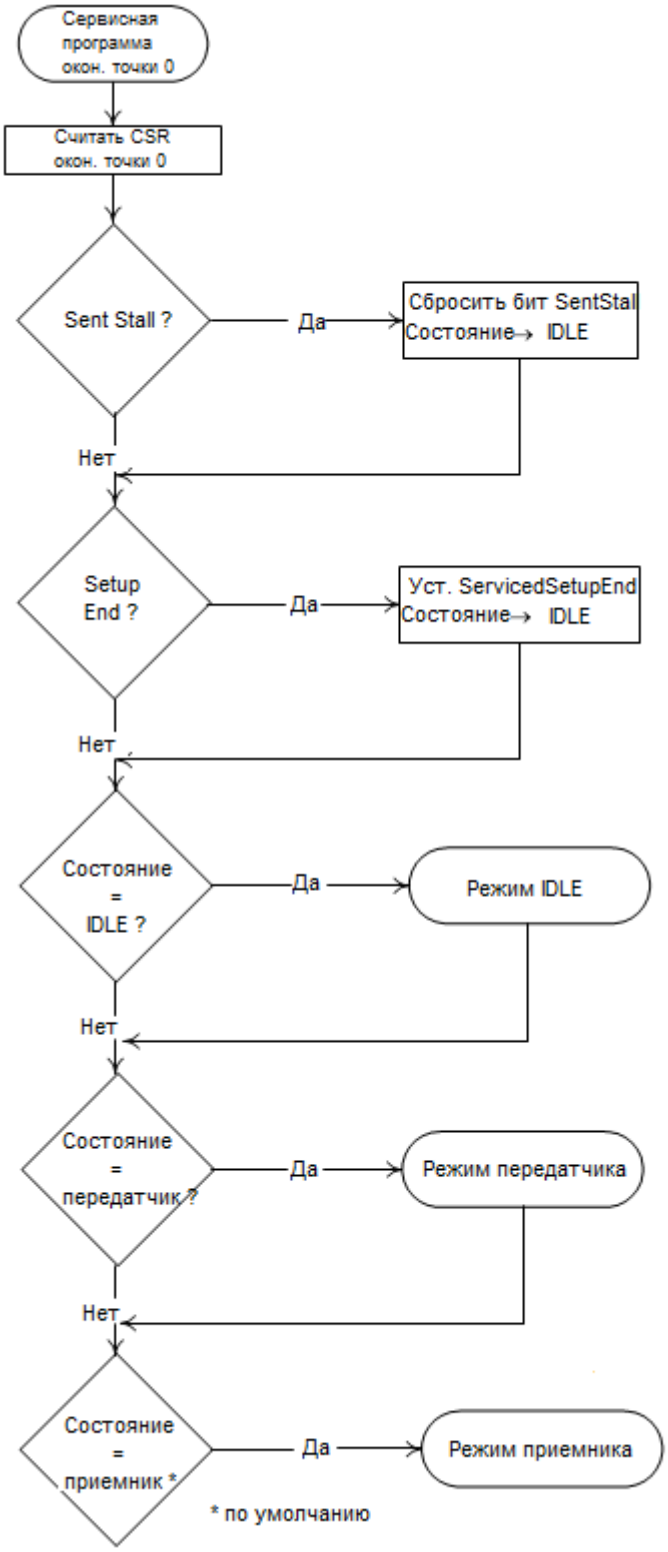


Рисунок 1.272

1.3.12.2.7.15.1.5.1 РЕЖИМ IDLE

Режим IDLE – это режим, который контроллер оконечной точки 0 должен выбрать при включении питания или сбросе, а также режим, в который контроллер оконечной точки 0 должен вернуться при выходе из режима приемника или передатчика.

В этом режиме также обрабатывается фаза SETUP управляющей пересылки (как указано на следующем рисунке).

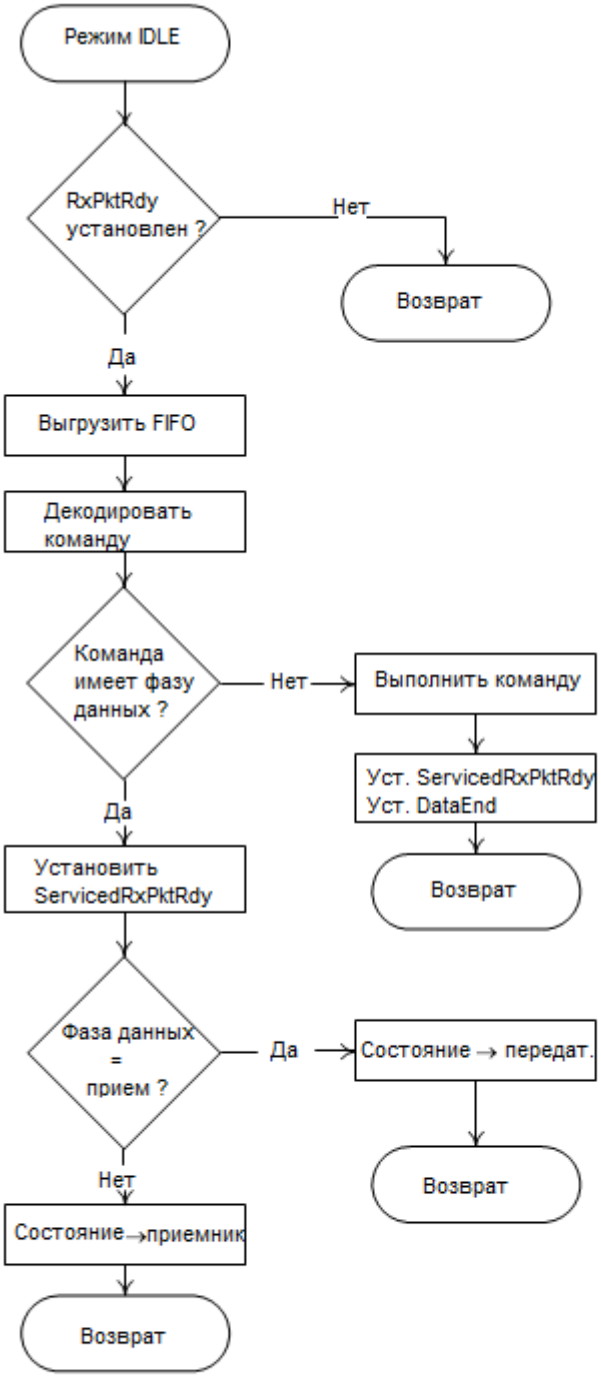


Рисунок 1.273 - Алгоритм для фазы SETUP управляющей пересылки, когда MUSBMHDRC работает в режиме ведомого устройства

1.3.12.2.7.15.1.5.2 РЕЖИМ ПЕРЕДАТЧИКА

Когда оконечная точка находится в состоянии передатчика, все поступающие маркеры чтения должны обрабатываться как часть фазы данных, пока необходимый объем данных не будет отправлен ведущему устройству. Получение маркера записи или маркера SETUP, когда оконечная точка находится в состоянии передатчика, вызовет состояние SetupEnd, поскольку ядро ожидает только маркеры чтения.

Выход из режима передатчика до отправки ожидаемого объема данных могут вызвать три события:

- Ведущее устройство посылает недопустимый маркер, вызывающий состояние SetupEnd (CSR0L.D4 установлен)
- Аппаратная часть отправляет пакет, размер которого меньше максимального, установленного до оконечной точки 0 (MaxP)
- Аппаратная часть отправляет пустой пакет данных.

До окончания транзакции аппаратной части нужно просто загрузить FIFO по приходу прерывания, которое указывает, что пакет был отправлен из FIFO. (Прерывание генерируется при сбросе TxPktRdy.)

Когда аппаратная часть принудительно завершает пересылку (отправляя короткий или пустой пакет данных), она должна установить бит DataEnd(CSR0L.D3), чтобы указать ядру, что фаза данных завершена, и что ядро должно получить пакет подтверждения.

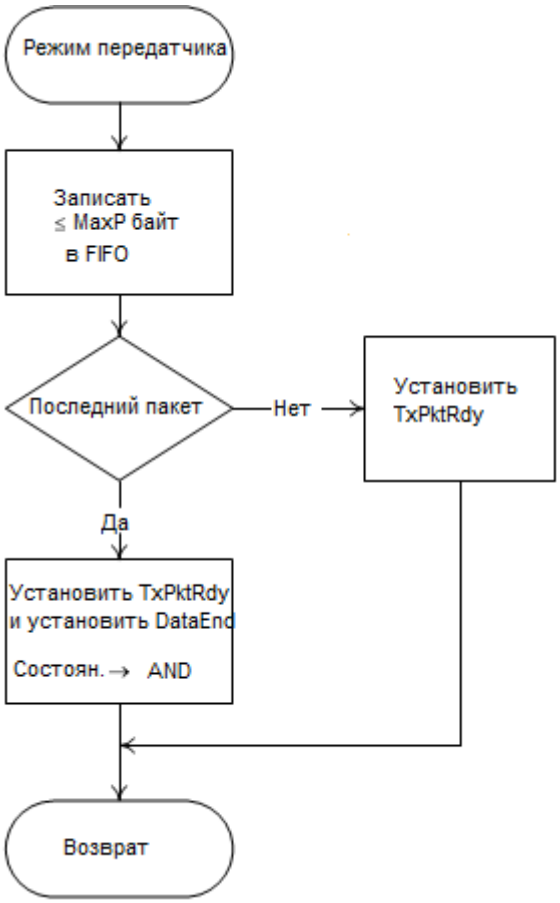


Рисунок 1.274 - Алгоритм для фазы приема данных управляющей пересылки, когда MUSBMHDRC работает в режиме ведомого устройства

1.3.12.2.7.15.1.5.3 . РЕЖИМ ПРИЕМНИКА

В режиме приемника все поступающие данные должны обрабатываться как часть фазы данных, пока необходимый объем данных не будет получен. Получение маркера чтения или маркера SETUP, когда оконечная точка находится в состоянии приемника, вызовет состояние SetupEnd, поскольку ядро ожидает только маркеры записи.

Выход из режима приемника до получения ожидаемого объема данных могут вызвать три события:

- Ведущее устройство посылает недопустимый маркер, вызывающий состояние SetupEnd (CSR0L.D4 установлен)
- Ведущее устройство отправляет пакет, размер которого меньше максимального, установленного до оконечной точки 0 (MaxP)
- Ведущее устройство отправляет пустой пакет данных.

До окончания транзакции аппаратной части нужно просто выгрузить FIFO по приходу прерывания, которое указывает на поступление новых данных (RxPktRdy (CSR0L.D0) установлен), и сбросить RxPktRdy, установив бит ServicedRxPktRdy (CSR0L.D6).

Когда аппаратная часть обнаруживает прекращение пересылки (получив ожидаемый объем данных или пустой пакет данных), она должна установить бит DataEnd(CSR0L.D3), чтобы указать ядру, что фаза данных завершена, и что ядро должно получить пакет подтверждения.

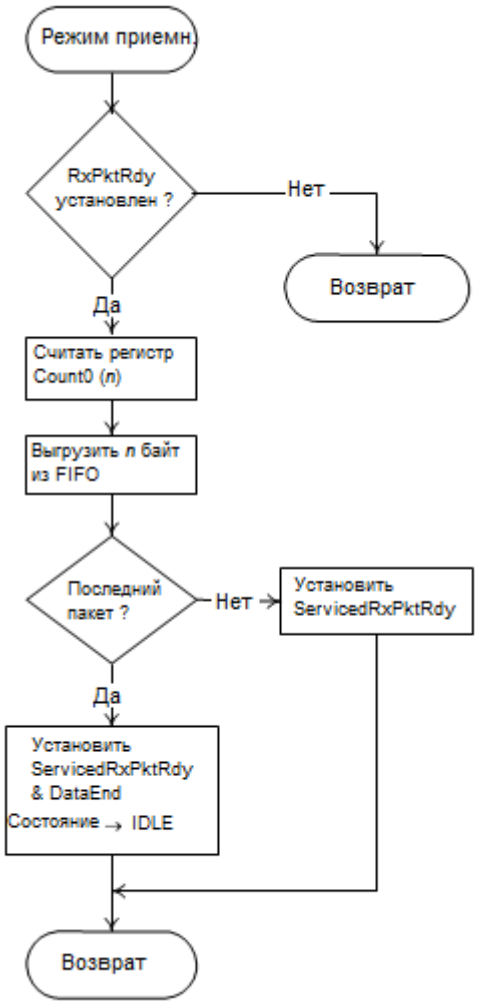


Рисунок 1.274 - Алгоритм для фазы отправки данных управляющей пересылки, когда MUSBMHDRC работает в режиме ведомого устройства

1.3.12.2.7.15.1.6 ОБРАБОТКА ОШИБОК В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Управляющая пересылка может быть прервана вследствие ошибки протокола на USB, вследствие того, что ведущее устройство преждевременно завершает пересылку, или если управляющая программа функции желает прервать пересылку (например, из-за того, что не может выполнить команду).

MUSBMHDRC автоматически обнаруживает ошибки протокола и передает пакет STALL ведущему устройству при следующих условиях:

-
- Ведущее устройство во время фазы отправки данных запроса чтения запрашивает больше данных, чем было указано в команде. Это состояние детектируется, когда ведущее устройство передает маркер записи после того, как был установлен бит DataEnd в регистре CSR0.
- Ведущее устройство во время фазы приема данных запроса записи посылает больше данных, чем было указано в команде. Это состояние детектируется, когда ведущее устройство передает маркер чтения после того, как был установлен бит DataEnd в регистре CSR0.

- Ведущее устройство отправляет в пакете больше данных, чем MaxP.
- Ведущее устройство отправляет пакет ненулевой длины DATA1 во время фазы STATUS запроса чтения.

После отправки MUSBMHDRC пакета STALL устанавливается бит SentStall (CSR0L.D2) и генерируется прерывание. Когда программа получает прерывание оконечной точки 0 с установленным битом SentStall, она должна прервать текущую пересылку, сбросить бит SentStall и вернуться с состояние IDLE.

Если ведущее устройство преждевременно заканчивает пересылку, переходя в фазу STATUS до того, как были переданы все запрошенные данные, или отправляя пакет SETUP до завершения текущей пересылки, то будет установлен бит SetupEnd (CSR0L.D4), и сгенерировано прерывание оконечной точки 0. Когда программа получает прерывание оконечной точки 0 с установленным битом SetupEnd, она должна прервать текущую пересылку, сбросить бит ServicedSetupEnd (CSR0L.D7) и вернуться с состояние IDLE. Если бит RxPktRdy (CSR0L.D0) установлен, это указывает, что ведущее устройство отправило еще один пакет SETUP, и программа должна обработать эту команду.

Если программа желает прервать текущую пересылку из-за того, что не может обработать команду, или произошла внутренняя ошибка, то она должна установить бит SendStall (CSR0L.D5). Тогда MUSBMHDRC отправит пакет STALL ведущему устройству, установит бит SentStall (CSR0L.D2) и сгенерирует прерывание оконечной точки 0.

1.3.12.2.7.15.1.7 ДОПОЛНИТЕЛЬНЫЕ ДЕЙСТВИЯ

В режиме ведомого устройства ядро MUSBMHDRC автоматически реагирует на определенные состояния шины USB или действия ведущего устройства. Подробная информация приводится ниже:

ПАКЕТ STALL В УПРАВЛЯЮЩЕЙ ПЕРЕСЫЛКЕ

Ядро MUSBMHDRC автоматически отправит согласование STALL управляющей пересылке при следующих условиях:

- Ведущее устройство во время фазы отправки данных управляющей пересылки посылает больше данных, чем указано в запросе устройства во время фазы SETUP.

Это состояние детектируется MUSBMHDRC, когда ведущее устройство передает маркер записи (вместо маркера чтения) после того, как CPU выгрузило последний пакет для отправки и установило DataEnd.

- Ведущее устройство во время фазы приема данных управляющей пересылки запрашивает больше данных, чем указано в запросе устройства во время фазы SETUP.
-

Это состояние детектируется MUSBMHDRC, когда ведущее устройство передает маркер чтения (вместо маркера записи) после того, как CPU сбросило бит TxPktRdy и установило бит DataEnd в ответ на пакет ACK, отправленный ведущим устройством в ответ на пакет, который должен быть последним.

- Ведущее устройство посылает с маркером записи данные большего объема чем MaxP.

- Ведущее устройство посылает пакет данных ненулевой длины для фазы отправки статуса.

ПАКЕТЫ ДАННЫХ OUT НУЛЕВОЙ ДЛИНЫ В УПРАВЛЯЮЩИХ ПЕРЕСЫЛКАХ

Пакет отправляемых данных нулевой длины используется для указания конца управляющей пересылки. При нормальной работе такие пакеты могут быть получены только после пересылки всей длины запроса устройства (то есть, после того, как CPU установило бит DataEnd). Если же ведущее устройство посылает пакет данных нулевой длины прежде чем был переслана вся длина запроса устройства, это указывает на преждевременный конец пересылки. В таком случае MUSBMHDRC автоматически очистит из FIFO любой маркер чтения, загруженный CPU, готовым к фазе данных, и устанавливает бит SetupEnd (CSR0L.D4).

1.3.12.2.7.15.2 УПРАВЛЯЮЩИЕ ТРАНЗАКЦИИ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Управляющие транзакции ведущего устройства выполняются через оконечную точку 0, и программное обеспечение должно обрабатывать все запросы стандартного устройства, которые могут быть отправлены или получены через оконечную точку 0 (как описано в Universal Serial Bus Specification, версия 2.0, глава 9).

Точно так же, как и для ведомого устройства USB, запросы стандартного устройства могут быть разделены на три категории: запросы с данными нулевой длины (в которых информация включена в команду), запросы записи (в которых за командой следуют дополнительные данные) и запросы чтения (в которых от ведомого устройства требуется отправить данные ведущему устройству).

- Запросы с данными нулевой длины содержат команду SETUP, за которой следует фаза ожидания статуса (IN Status).
- Запросы записи содержат команду SETUP, за которой следует фаза отправки данных, а за ней фаза ожидания статуса.
- Запросы чтения содержат команду SETUP, за которой следует фаза приема данных, а за ней фаза отправки статуса.

Может быть установлено время ожидания, чтобы ограничить период, в течение которого MUSBMHDRC будет пытаться повторить транзакцию, на которую целевое устройство отвечает пакетом NAK. Это ограничение может находиться в диапазоне от 2 до 215 фреймов/микрофреймов и устанавливается с помощью регистра NAKLimit0 (см. раздел 1.3.12.2.7.3.3.6).

В следующих разделах описаны действия, которые должно выполнить CPU для отправки разных типов запросов – на разных фазах управляющей транзакции.

Примечание: Прежде чем инициировать любую транзакцию в режиме ведущего устройства, необходимо записать в регистр FAddr адрес ведомого устройства. При первом подключении значение FAddr должно быть сброшено в ноль. После отправки команды SET_ADDRESS в регистр FAddr следует записать новый адрес целевого устройства.

1.3.12.2.7.15.2.1 21.2.1. ФАЗА SETUP В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Для фазы SETUP управляющей транзакции CPU, управляющий работой ведущего устройства, должно:

- Загрузить 8 байт требуемой команды запроса устройства в FIFO оконечной точки 0
- Затем установить SetupPkt и TxPtRdy (биты CSR0L.D3 и CSR0L.D1, соответственно).

После этого MUSBMHDRC отправляет маркер SETUP, за которым следует 8-байтовая команда оконечной точке 0 адресуемого устройства – при необходимости попытки повторяются. (Подробно эта операция описана в разделе 19.1.)

-
- В конце попытки отправить данные MUSBMHDRC генерирует прерывание оконечной точки 0 (то есть, устанавливает бит IntrTx.D0). Затем CPU должно считать регистр CSR0, чтобы определить состояние битов RxStall (D2), Error (D4) и NAK Timeout (D7).

Если установлен бит RxStall, это указывает, что цель не приняла команду (например, потому что она не поддерживается целевым устройством) и выдала ответ STALL.

Если установлен бит Error, это значит, что MUSBMHDRC пытался три раза отправить пакет SETUP и следующий за ним пакет данных, но не получил ответа.

Если установлен бит NAK Timeout, это значит, что MUSBMHDRC получил ответ NAK на каждую попытку отправить пакет SETUP в течение времени, превышающего установленный в регистре NAKLimit0 предел. В этом случае MUSBMHDRC может получить указание продолжить попытки выполнить транзакцию (пока снова не истечет установленное время), что достигается сбросом бита NAK Timeout, либо прервать транзакцию, очистив FIFO и затем сбросив бит NAK Timeout.

- Если не установлен ни один из битов RxStall, Error и NAK Timeout, то на фазу SETUP получен ответный пакет подтверждения ACK, и CPU должен перейти к следующей фазе, необходимой для данного запроса стандартного устройства - приема данных, отправки данных или ожидания статуса.

1.3.12.2.7.15.2.2 ФАЗА ПРИЕМА ДАННЫХ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Для фазы приема данных управляющей транзакции CPU, управляющий работой ведущего устройства, должно:

- Установить бит Set ReqPkt (CSR0L.D5).
- Ожидать, когда MUSBMHDRC отправит маркер чтения и получит в ответ требуемые данные.
- Когда MUSBMHDRC MUSBMHDRC генерирует прерывание оконечной точки 0 (то есть, устанавливает бит IntrTx.D0), CPU должно считать регистр CSR0, чтобы определить состояние битов RxStall (D2), Error (D4), NAK Timeout (D7) и RxPktRdy (D0)

Если установлен бит RxStall, это указывает, что цель выдала ответ STALL.

Если установлен бит Error, это значит, что MUSBMHDRC пытался три раза отправить пакет IN и следующий за ним пакет данных, но не получил ответа.

Если установлен бит NAK Timeout, это значит, что MUSBMHDRC получил ответ NAK на каждую попытку отправить маркер чтения в течение времени, превышающего установленный в регистре NAKLimit0 предел. В этом случае MUSBMHDRC может получить указание продолжить

попытки выполнить транзакцию (пока снова не истечет установленное время), что достигается сбросом бита NAK Timeout, либо прервать транзакцию, сбросив ReqPkt, а затем сбросив бит NAK Timeout.

- Если установлен бит RxPktRdy, CPU должно считать данные из FIFO оконечной точки 0, затем сбросить бит RxPktRdy.
- Если не все данные приняты, CPU должен повторить шаги 1 – 4.

После успешного приема всех данных CPU должен перейти к фазе отправки статуса управляющей транзакции.

1.3.12.2.7.15.2.3 ФАЗА ОТПРАВКИ ДАННЫХ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Для фазы отправки данных управляющей транзакции CPU, управляющий работой ведущего устройства, должно:

- Загрузить в FIFO оконечной точки 0 данные для пересылки.
- Установить бит TxPtRdy bit (CSR0L.D1).

Затем MUSBMHDRC отправляет оконечной точке 0 адресуемого устройства маркер записи, за которым следуют данные из FIFO – при необходимости попытки повторяются.

- В конце попытки отправить данные MUSBMHDRC генерирует прерывание оконечной точки 0 (то есть, устанавливает бит IntrTx.D0). Затем CPU должно считать регистр CSR0, чтобы определить состояние битов RxStall (D2), Error (D4) и NAK Timeout (D7).

Если установлен бит RxStall, это указывает, что цель выдала ответ STALL.

Если установлен бит Error, это значит, что MUSBMHDRC пытался три раза отправить пакет данных и следующий за ним пакет данных, но не получил ответа.

Если установлен бит NAK Timeout, это значит, что MUSBMHDRC получил ответ NAK на каждую попытку отправить пакет данных в течение времени, превышающего установленный в регистре NAKLimit0 предел. В этом случае MUSBMHDRC может получить указание продолжить попытки выполнить транзакцию (пока снова не истечет установленное время), что достигается сбросом бита NAK Timeout, либо прервать транзакцию, очистив FIFO и затем сбросив бит NAK Timeout.

Если не установлен ни один из битов RxStall, Error и NAK Timeout, то на фазу отправки данных получен ответный пакет подтверждения ACK.

- Если не все данные переданы, CPU должен повторить шаги 1 – 4.

После успешной отправки всех данных CPU должен перейти к фазе ожидания статуса управляющей транзакции.

1.3.12.2.7.15.2.4 ФАЗА ОЖИДАНИЯ СТАТУСА В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

(ПОСЛЕ ФАЗЫ SETUP ИЛИ ФАЗЫ ОТПРАВКИ ДАННЫХ)

Для фазы ожидания статуса управляющей транзакции CPU, управляющее работой ведущего устройства, должно:

Установить StatusPkt и ReqPkt (бит CSR0L.D6 и CSR0L.D5, соответственно). Примечание: эти биты должны быть установлены одновременно в одной операции записи.

Ожидать, когда MUSBMHDRC отправит маркер чтения и получит ответ от ведомого устройства. (Подробно эта операция описана в разделе 19.1.)

Когда MUSBMHDRC MUSBMHDRC генерирует прерывание оконечной точки 0 (то есть, устанавливает бит IntrTx.D0), CPU должно считать регистр CSR0, чтобы определить состояние битов RxStall (D2), Error (D4), NAK Timeout (D7) и RxPktRdy (D0)

- Если установлен бит RxStall, это указывает, что цель не может выполнить команду и выдает ответ STALL.
-
- Если установлен бит Error, это значит, что MUSBMHDRC пытался три раза отправить маркер чтения, но не получил ответа.
-
- Если установлен бит NAK Timeout, это значит, что MUSBMHDRC получил ответ NAK на каждую попытку отправить маркер чтения в течение времени, превышающего установленный в регистре NAKLimit0 предел. В этом случае MUSBMHDRC может получить указание продолжить попытки выполнить транзакцию (пока снова не истечет установленное время), что достигается сбросом бита NAK Timeout, либо прервать транзакцию, сбросив ReqPkt and StatusPkt, а затем сбросив бит NAK Timeout.

- CPU должен сбросить бит StatusPkt – одновременно (в одной операции записи) с битом RxPktRdy, если он был установлен.

1.3.12.2.7.15.2.5 ФАЗА ОТПРАВКИ СТАТУСА В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

(ПОСЛЕ ФАЗЫ ПРИЕМА ДАННЫХ)

Для фазы отправки статуса управляющей транзакции CPU, управляющий работой ведущего устройства, должно:

- Установить StatusPkt и TxPktRdy (бит CSR0L.D6 и CSR0L.D1, соответственно). Примечание: эти биты должны быть установлены одновременно.
- Ожидать, когда MUSBMHDRC отправит маркер записи и пакет DATA1 нулевой длины.
- В конце попытки отправить данные MUSBMHDRC генерирует прерывание оконечной точки 0 (то есть, устанавливает бит IntrTx.D0). Затем CPU должно считать регистр CSR0, чтобы определить состояние битов RxStall (D2), Error (D4) и NAK Timeout (D7).

Если установлен бит RxStall, это указывает, что цель не может выполнить команду и выдает ответ STALL.

Если установлен бит Error, это значит, что MUSBMHDRC пытался три раза отправить пакет статуса и следующий за ним пакет данных, но не получил ответа.

Если установлен бит NAK Timeout, это значит, что MUSBMHDRC получил ответ NAK на каждую попытку отправить маркер чтения в течение времени, превышающего установленный в регистре NAKLimit0 предел. В этом случае MUSBMHDRC может получить указание продолжить попытки выполнить транзакцию (пока снова не истечет установленное время), что достигается сбросом бита NAK Timeout, либо прервать транзакцию, очистив FIFO и затем сбросив бит NAK Timeout.

- Если не установлен ни один из битов RxStall, Error и NAK Timeout, то на фазу статуса получен ответный пакет подтверждения ACK.

1.3.12.2.7.16 ПОТОЧНЫЕ ТРАНЗАКЦИИ

1.3.12.2.7.16.1 ПОТОЧНЫЕ ТРАНЗАКЦИИ В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

1.3.12.2.7.16.1.1 ПОТОЧНЫЕ ТРАНЗАКЦИИ ЧТЕНИЯ

Поточные транзакции чтения используются для пересылки непериодических данных от контроллера функции к ведущему устройству.

Для использования оконечной точки передатчика в режиме ведомого устройства для поточных транзакций чтения доступны четыре дополнительные опции:

- Двойная буферизация пакетов

За исключением случаев, когда используется динамическое определение размера FIFO, если значение, записанное в регистр TxMaxP, меньше или равно половине размера FIFO, выделенного для оконечной точки, то автоматически разрешается двойная буферизация пакетов. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакетов является частью спецификации для FIFO оконечной точки – см. раздел 1.3.12.2.7.3.3.18). При включенной опции в FIFO могут храниться до двух пакетов, ожидающих пересылки к ведомому устройству.

- DMA

Если для оконечной точки разрешен DMA, то должен вырабатываться запрос DMA, когда оконечная точка готова принять следующий пакет в свой FIFO. Эта функция может быть использована для разрешения контроллеру DMA (например, опционально включаемому в блок MUSBMHDRC) загружать пакеты в FIFO без участия процессора. Если используется режим 1 DMA, то для правильной генерации прерывания в регистре TxMaxP[D10:0] должно быть записано четное число.

- AutoSet

При разрешенной функции AutoSet бит TxPktRdy (TxCSRL.D0) автоматически устанавливается, когда в FIFO загружен пакет размером TxMaxP байт. Эта функция особенно полезна, если для загрузки FIFO используется DMA, поскольку позволяет избежать необходимости участия процессора в загрузке отдельных пакетов при объемной поточной пересылке.

- Автоматическое разделение пакетов

В некоторых системах прикладной программе бывает удобно за одну операцию записывать в оконечную точку объемы данных большие, чем можно переслать за одну операцию USB. В особенности это актуально, когда одна и та же оконечная точка в одних обстоятельствах используется для высокоскоростных пересылок 512 байт, а в других для пересылок со средней скоростью. При работе на средней скорости максимальный объем данных одной пересылки составляет всего 64 байта. Для этого у MUSBMHDRC имеется опция, которая позволяет записывать в оконечные точки поточной пересылки большие объемы данных, которые затем разделяются на пакеты соответствующего (указанного) размера для пересылки по шине USB. Наличие этой опции определяется по установленному биту MPTxE (D6) регистра ConfigData (см. раздел 1.3.12.2.7.3.3.5).

Информация о необходимом размере пакета устанавливается в регистре TxMaxP (см. раздел 1.3.12.2.7.3.3.7).

1.3.12.2.7.16.1.1.1 НАЧАЛЬНАЯ НАСТРОЙКА

При конфигурировании оконечной точки передатчика для поточных транзакций в регистр TxMaxP следует записать максимальный размер пакета (в байтах) для оконечной точки. Это значение должно совпадать со значением поля wMaxPacketSize стандартного дескриптора оконечной точки для данной оконечной точки. Кроме того, в регистре IntrTxE должен быть установлен в ‘1’ соответствующий бит разрешения прерывания (если для данной точки требуется прерывание), а старший байт регистра TxCSR должен быть установлен в соответствии с приведенной ниже таблицей (биты D9 – D8 не используются):

D15	AutoSet	0/1	Устанавливается в 1, если необходима функция AutoSet.
D14	ISO	0	Устанавливается в 0 для разрешения протокола поточной передачи.
D13	Mode	1	Устанавливается в 1 для разрешения FIFO (необходимо только в том случае, когда FIFO используется совместно с оконечной точкой приемника t).
D12	DMAReqEnab	0/1	Устанавливается в 1, если требуются запросы DMA. Примечание: Если установлен в 1, то необходимо также выбрать DMAReqMode (TxCSRH.D2).
D11	FrcDataTog	0	Устанавливается в 0 для разрешения работы флага данных в стандартном режиме.

При начальном конфигурировании оконечной точки (после команды SET_CONFIGURATION или SET_INTERFACE на оконечной точке 0) следует записать младший байт TxCSR, чтобы установить бит ClrDataTog (D6). Это обеспечивает правильную начальную установку флага данных (управляется MUSBMHDRC автоматически). Кроме того, если в FIFO присутствуют пакеты данных (на что указывает установленный бит FIFONotEmpty (TxCSRL.D1)), они должны быть очищены, при помощи установки бита FlushFIFO (TxCSRL.D3). Примечание: Возможно, этот бит придется устанавливать два раза подряд, если разрешена двойная буферизация пакетов.

1.3.12.2.7.16.1.1.2 РАБОТА

При необходимости пересылки данных по входному поточному каналу, пакет данных необходимо загрузить в FIFO и выполнить запись в регистр TxCSR, чтобы установить бит

TxPktRdy (D0). После отправки пакета MUSBMHDRC сбрасывает бит TxPktRdy, и генерируется прерывание, чтобы можно было загрузить в FIFO следующий пакет. Если разрешена двойная буферизация пакетов, то после загрузки первого пакета и установки бита TxPktRdy бит TxPktRdy будет немедленно сброшен MUSBMHDRC, и сгенерировано прерывание, чтобы в FIFO можно было загрузить второй пакет. Программа должна действовать одинаково, загружая пакет после получения прерывания, независимо от того, разрешена двойная буферизация пакетов, или нет.

В общем случае размер пакета не должен превышать размер, указанный младшими 11 битами регистра TxMaxP эта часть регистра определяет содержимое (размер пакета) для пересылок по USB и согласно спецификации USB может составлять 8, 16, 32, 64 (средняя или высокая скорость соединения) или 512 байт (только высокая скорость соединения). Если необходимо передать больший объем данных, они должны пересылаться как группа пакетов USB, каждый из которых имеет размер TxMaxP[D10:D0], за исключением последнего пакета, содержащего остаток данных.

Исключение из этого правила имеет место в том случае, когда при конфигурации ядра выбрана опция автоматического разделения пакетов поточной пересылки. (Это можно определить по установленному биту MPTxE (D6) регистра ConfigData.) При выборе этой опции в FIFO могут быть записаны пакеты в 32 раза превышающие размер, указанный в TxMaxP[D10:D0] (при условии достаточно большого объема FIFO, чтобы вместить эти большие пакеты), которые затем разделяются ядром на пакеты соответствующего размера для пересылки по USB. Размер пакетов, записанных в FIFO, вычисляется как m □□USB-содержимое, где TxMaxP[D15:D11] = m – 1. Чтобы воспользоваться этой функцией, прикладной программе нужно лишь записать нужное значения в регистр TxMaxP (и убедиться, что значение битов 10:0 совпадает со значением поля wMaxPacketSize стандартного дескриптора для данной оконечной точки). С точки зрения прикладной программы процесс пересылки этих больших пакетов не отличается от пересылки пакета стандартного размера в режиме поточной пересылки.

Ведущее устройство может определить, что были отправлены все предназначенные для пересылки данные, зная общий объем ожидаемых данных. Или оно может считать, что были отправлены все данные, если получит пакет меньшего размера, чем установленное содержимое (TxMaxP[D10:D0]). В последнем случае если общий объем блока данных кратен этому содержимому, то после пересылки всех данных функция должна отправить нулевой пакет. Это делается посредством установки бита TxPktRdy, когда придет следующее прерывание, но без загрузки данных в FIFO.

При пересылке больших блоков данных можно избежать необходимости вызывать сервисную подпрограмму для загрузки каждого пакета, если использовать обработку ошибок DMAError.

Если программа желает закрыть входной поточный канал (Bulk IN pipe), она должна установить бит SendStall (TxCSRL.D4). Когда MUSBMHDRC получит следующий маркер чтения, он отправит сигнал STALL ведущему устройству, установит бит SentStall (TxCSRL.D5) и сгенерирует прерывание.

Когда программа получает прерывание с установленным битом SentStall (TxCSRL.D5), она должна сбросить бит SentStall. Тем не менее, она должна оставить установленным бит SendStall (TxCSRL.D4) до тех пор, пока не сможет снова открыть входной поточный канал. Примечание: Если ведущее устройство по какой-то причине не смогло получить пакет STALL, оно отправит еще один маркер ввода, и поэтому рекомендуется не сбрасывать бит SendStall до тех пор, пока программа не будет готова снова открыть входной поточный канал. После повторного открытия

канала последовательность флагов данных должна быть перезапущена установкой бита ClrDataTog в регистре TxCSR (D6).

1.3.12.2.7.16.1.2 ПОТОЧНЫЕ ТРАНЗАКЦИИ ЗАПИСИ В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Поточные транзакции записи используются для пересылки непериодических данных от ведущего устройства к контроллеру функции.

Для использования оконечной точки передатчика в режиме ведомого устройства для поточных транзакций записи доступны четыре дополнительные опции:

- Двойная буферизация пакетов

За исключением случаев, когда используется динамическое определение размера FIFO, если значение, записанное в регистр RxMaxP, меньше или равно половине размера FIFO, выделенного для оконечной точки, то автоматически разрешается двойная буферизация пакетов. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакетов является частью спецификации для FIFO оконечной точки – см. раздел 3.3.18). При включенной опции в FIFO могут храниться до двух пакетов.

- DMA

Если для оконечной точки разрешен DMA, то должен вырабатываться запрос DMA, когда оконечная точка содержит пакет данных в своем FIFO. Эта функция может быть использована для разрешения DMA без участия процессора. Если используется режим 1 DMA, то для правильной генерации прерывания в регистре RxMaxP[D10:0] должно быть записано четное число.

- AutoClear

При разрешенной функции AutoClear бит RxPktRdy (RxCSRL.D0) автоматически сбрасывается, когда из FIFO выгружен пакет размером RxMaxP байт (исключения см. в описании регистра). Эта функция особенно полезна, если для выгрузки FIFO используется DMA, поскольку позволяет избежать необходимости участия процессора в выгрузке отдельных пакетов при объемной поточной пересылке.

- Автоматическое объединение пакетов

В некоторых системах прикладной программе бывает удобно за одну операцию считывать с оконечной точки объемы данных большие, чем можно переслать за одну операцию USB. В особенности это актуально, когда одна и та же оконечная точка в одних обстоятельствах используется для высокоскоростных пересылок 512 байт, а в других для пересылок со средней скоростью. При работе на средней скорости максимальный объем данных одной пересылки составляет всего 64 байта. Для этого у MUSBMHDRC имеется опция конфигурации, которая позволяет объединять полученные по USB пакеты в большие пакеты данных, которые затем считываются прикладной программой. Наличие этой опции определяется по установленному биту MPRxE (D7) регистра ConfigData register (см. раздел 1.3.12.2.7.3.3.5). Информация о необходимом размере пакета устанавливается в регистре RxMaxP (см. раздел 1.3.12.2.7.3.3.10), а размер текущего объединенного пакета хранится в регистре RxCount (см. раздел 1.3.12.2.7.3.3.13).

1.3.12.2.7.16.1.2.1 НАЧАЛЬНАЯ НАСТРОЙКА

При конфигурировании оконечной точки приемника для поточных транзакций в регистр RxMaxP следует записать максимальный размер пакета (в байтах) для оконечной точки. Это значение должно совпадать со значением поля wMaxPacketSize стандартного дескриптора для данной оконечной точки. Кроме того, в регистре IntrRxE должен быть установлен в ‘1’ соответствующий бит разрешения прерывания (если для данной точки требуется прерывание), а старший байт регистра RxCSR должен быть установлен в соответствии с приведенной ниже таблицей (биты D10 – D8 не используются):

D15	AutoClear	0/1	Устанавливается в 1, если необходима функция AutoClear.
D14	ISO	0	Устанавливается в 0 для разрешения протокола поточной передачи.
D13	DMAReqEnab	0/1	Устанавливается в 1, если требуются запросы DMA. Примечание: Если установлен в 1, то необходимо также выбрать DMAReqMode (RxCSRH.D3).
D12	DisNyet	0	Устанавливается в 0 для разрешения стандартного протокола PING управления потоком данных (PING flow control).

При начальном конфигурировании оконечной точки (после команды SET_CONFIGURATION или SET_INTERFACE на оконечной точке 0) следует записать младший байт RxCSR, чтобы установить бит ClrDataTog (D6). Это обеспечивает правильную начальную установку флага данных (управляется MUSBMHDRC автоматически). Кроме того, если в FIFO присутствуют пакеты данных (на что указывает установленный бит RxPktRdy (RxCSRL.D0)), они должны быть очищены, при помощи установки бита FlushFIFO (RxCSRL.D4). Примечание: Возможно, этот бит придется устанавливать два раза подряд, если разрешена двойная буферизация пакетов.

1.3.12.2.7.16.1.2.2 РАБОТА

При получении оконечной точкой приемника пакета данных в режиме поточной пересылки устанавливается бит RxPktRdy (RxCSRL.D0) и генерируется прерывание. Программа должна считать регистр RxCount соответствующей оконечной точки, чтобы определить размер пакета. Пакет данных должен быть считан из FIFO, после чего сбрасывается RxPktRdy. Если бит FIFOFull был установлен в 1, когда сбрасывался RxPktRdyd, то MUSBMHDRC сначала сбросит бит FIFOFull. Затем MUSBMHDRC снова установит бит RxPktRdy, чтобы указать, что следующий пакет ждет выгрузки из FIFO.

Размер принятых пакетов не должен превышать размер, указанный в регистре RxMaxP (значение, установленное в поле wMaxPacketSize дескриптора оконечной точки, отправленного ведущему устройству). Если необходимо переслать функции блок данных, размер которого больше wMaxPacketSize, он будет отправлен в виде группы пакетов. Все пакеты будут иметь размер wMaxPacketSize, за исключение последнего пакета, в котором будет содержаться остаток данных. Программа может использовать собственный метод определения общего размера блока и приема последнего пакета. Или программа может считать, что принят весь блок данных, когда получает пакет меньшей длины, чем wMaxPacketSize. (Если общий размер блока данных кратен wMaxPacketSize, после пересылки данных будет отправлен пакет с нулевыми данными, который укажет на окончание пересылки).

В общем случае прикладная программа должна считывать каждый пакет из FIFO отдельно. Исключение из этого правила имеет место в том случае, когда при конфигурации ядра выбрана опция автоматического объединения пакетов поточной пересылки. (Это можно определить по

установленному биту MPRxE (D7) регистра ConfigData.) Если была выбрана эта опция, ядро может принять одновременно до 32 пакетов и объединить их в один пакет в FIFO (при условии, что FIFO буфер достаточно велик, чтобы вместить эти большие пакеты). Размер пакетов, записанных в FIFO, вычисляется как m □□wMaxPacketSize, где RxMaxP[D15:D11] = m – 1.

Чтобы воспользоваться этой функцией, прикладной программе нужно лишь записать нужное значения в регистр RxMaxP (и убедиться, что значение битов 10:0 совпадает со значением поля wMaxPacketSize стандартного дескриптора для данной оконечной точки). С точки зрения прикладной программы процесс пересылки этих больших пакетов не отличается от пересылки пакета стандартного размера в режиме поточной пересылки.

При пересылке больших блоков данных использование DMA позволяет избежать необходимости вызывать сервисную подпрограмму для выгрузки каждого пакета.

1.3.12.2.7.16.1.2.3 ОБРАБОТКА ОШИБОК

Если программа желает закрыть выходной поточный канал (Bulk OUT pipe), она должна установить бит SendStall (RxCSRL.D5). Когда MUSBMHDRC получит следующий пакет, он отправит сигнал STALL ведущему устройству, установит бит SentStall (RxCSRL.D6) и сгенерирует прерывание.

Когда программа получает прерывание с установленным битом SentStall (RxCSRL.D6), она должна сбросить этот бит. Тем не менее, она должна оставить установленным бит SendStall (RxCSRL.D5) до тех пор, пока не сможет снова открыть выходной поточный канал. Примечание: Если ведущее устройство по какой-то причине не смогло получить пакет STALL, оно отправит еще один пакет, и поэтому рекомендуется не сбрасывать бит SendStall до тех пор, пока программа не будет готова снова открыть выходной поточный канал. После повторного открытия канала последовательность флагов данных должна быть перезапущена установкой бита ClrDataTog в регистре RxCSR (D7).

1.3.12.2.7.16.2 ПОТОЧНЫЕ ТРАНЗАКЦИИ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

1.3.12.2.7.16.2.1 . ПОТОЧНЫЕ ТРАНЗАКЦИИ ЧТЕНИЯ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Поточные транзакции чтения используются для пересылки непериодических данных от контроллера функции к ведущему устройству.

Для использования оконечной точки приемника в режиме ведущего устройства для поточных транзакций чтения доступны пять дополнительных опций:

Двойная буферизация пакетов

При разрешенной двойной буферизации пакетов один пакет может приниматься, а второй одновременно считываться. За исключением случаев, когда используется динамическое определение размера FIFO, если значение, записанное в регистр RxMaxP, меньше или равно половине размера FIFO, выделенного для оконечной точки, то автоматически разрешается двойная буферизация пакетов. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакетов является частью спецификации для FIFO оконечной точки – см. раздел 1.3.12.2.7.3.13.1.2).

DMA

Если для оконечной точки разрешен DMA, то должен вырабатываться запрос DMA, когда оконечная точка содержит пакет данных в своем FIFO. Эта функция может быть использована для разрешения DMA без участия процессора.

AutoReq(uest)

При разрешенной функции AutoReq(uest) бит ReqPkt (RxCSRL.D5) будет автоматически установлен при сбросе бита RxPktRdy. Эта функция может быть использована совместно с регистром RqPktCount для запроса требуемого количества пакетов максимального размера.

AutoClear

При разрешенной функции AutoClear бит RxPktRdy (RxCSRL.D0) автоматически сбрасывается, когда из FIFO выгружен пакет размером RxMaxP байт (исключения см. в описании регистра). Эта функция особенно полезна совместно с функцией AutoRequest, когда для выгрузки FIFO используется DMA, поскольку позволяет избежать необходимости участия процессора в выгрузке отдельных пакетов при объемной поточной пересылке.

- Автоматическое объединение пакетов

В некоторых системах прикладной программе бывает удобно за одну операцию считывать с оконечной точки объемы данных большие, чем можно переслать за одну операцию USB. В особенности это актуально, когда одна и та же оконечная точка в одних обстоятельствах используется для высокоскоростных пересылок 512 байт, а в других для пересылок со средней скоростью. При работе на средней скорости максимальный объем данных одной пересылки составляет всего 64 байта. Для этого у MUSBMHDRC имеется опция конфигурации, которая позволяет объединять полученные по USB пакеты в большие пакеты данных, которые затем считываются прикладной программой. Наличие этой опции определяется по установленному биту MPRxE (D7) регистра ConfigData register (см. раздел 3.3.5). Информация о необходимом размере пакета устанавливается в регистре RxMaxP (см. раздел 3.3.10), а размер текущего объединенного пакета, который будет считываться, хранится в регистре RxCount (см. раздел 1.3.12.2.7.3.3.13).

1.3.12.2.7.16.2.1.1 . НАЧАЛЬНАЯ НАСТРОЙКА

Прежде чем инициировать поточную транзакцию чтения:

- Должен быть установлен адрес целевой функции, как описано в разделе 1.3.12.2.7.3.14.1.
- Для оконечной точки MUSBMHDRC, которая будет использоваться, должен быть записан Регистр RxType - биты D7, D6 определяют рабочую скорость, биты D5, D4 = 10 (для выбора поточной пересылки, а биты D3 – D0 содержат номер оконечной точки, содержащийся в дескрипторе оконечной точки приемника, который передается MUSBMHDRC во время нумерации устройства (см. раздел 3.3.16).
- В регистр RxMaxP для оконечной точки MUSBMHDRC должен быть записан максимальный размер пакета (в байтах) для пересылки. Значение должно совпадать со значением поля wMaxPacketSize стандартного дескриптора для целевой оконечной точки.
- В регистр RxInterval необходимо записать требуемое значение NAK Limit (2 – 215 фреймов/микрофреймов), или нулевое значение, если функция NAK Timeout не требуется.

- В регистре IntrRxE необходимо установить в ‘1’ соответствующий бит разрешения прерывания (если для данной оконечной точки требуется прерывание).
- Биты регистра RxCSR должны быть установлены в соответствии с приведенной ниже таблицей:

D15	AutoClear	0/1	Устанавливается в 1, если необходима функция AutoClear.
D14	AutoReq	0/1	Устанавливается в 1, если необходима функция AutoRequest.
D13	DMAReqEnab	0/1	Устанавливается в 1, если требуется запрос DMA. Примечание: Если установлен в 1, то необходимо также выбрать DMAReqMode (RxCSRH.D3).
D12	DisNyet	0	Устанавливается в 0 для стандартного протокола PING управления потоком данных.

При начальном конфигурировании необходимо сбросить в 0 флаг данных либо с помощью битов Data Toggle Write Enable и Data Toggle (RxCSRH.D2 и D9), чтобы переключить текущее состояние, либо записью младшего байта регистра RxCSR чтобы установить бит ClrDataTog (D7). Это обеспечивает правильную начальную установку флага данных (управляется MUSBMHDRC автоматически). Кроме того, если в FIFO присутствуют пакеты данных (на что указывает установленный бит RxPktRdy (RxCSRL.D0), они должны быть очищены, при помощи установки бита FlushFIFO (RxCSRL.D4). Примечание: Возможно, этот бит придется устанавливать два раза подряд, если разрешена двойная буферизация пакетов.

1.3.12.2.7.16.2.1.2 РАБОТА

Когда от ведомого устройства USB запрашиваются поточные данные, CPU должно установить бит ReqPkt в соответствующем регистре RxCSR (D5). MUSBMHDRC затем отправит маркер чтения выбранной оконечной точке ведомого устройства и будет ждать пересылки данных.

При корректном получении данных устанавливается бит RxPktRdy (RxCSRL.D0). Если ведомое устройство USB отвечает пакетом STALL, то устанавливается бит RxStall (RxCSRL.D6). Если получен пакет NAK, MUSBMHDRC предпринимает еще одну попытку – и повторяет попытки либо до успешной транзакции, либо до тех пор, пока не будет исчерпано время ожидания NAKLimit, установленное в регистре RxInterval. Если не получено вообще никакого ответа, MUSBMHDRC предпринимает еще две попытки, а затем сообщает об ошибке (установкой бита RxCSRL.D2).

Затем MUSBMHDRC генерирует прерывание соответствующей оконечной точки, по которому CPU должно считать соответствующий регистр RxCSR, чтобы определить, установлены ли биты RxPktRdy, RxStall, Error или NAK Timeout, после чего выполнить соответствующие действия. (Если установлен бит NAK Timeout, то MUSBMHDRC может получить указание продолжить попытки выполнить транзакцию (пока снова не истечет установленное время), что достигается сбросом бита NAK Timeout, либо прервать транзакцию, сбросив ReqPkt, а затем сбросив бит NAK Timeout.

Размер принятых пакетов не должен превышать размер, указанный в регистре RxMaxP (значение, установленное в поле wMaxPacketSize дескриптора оконечной точки, отправленного ведущему устройству).

Если необходимо переслать функции блок данных, размер которого больше wMaxPacketSize, он будет отправлен в виде группы пакетов. Все пакеты будут иметь размер wMaxPacketSize, за исключение последнего пакета, в котором будет содержаться остаток

данных. Если известен размер блока данных, который необходимо переслать, то количество пакетов для пересылки wMaxPacketSize можно записать в регистр RqPktCount и включить опцию AutoReq. При запросе каждого пакета значение в регистре будет уменьшаться на единицу. Когда значение в регистре RqPktCount уменьшится с 1 до 0, бит AutoReq сбрасывается, что исключает дальнейшие запросы. Другой вариант – считать, что получен весь блок, после получения пакета размером меньше, чем wMaxPacketSize. (Если общий размер блока данных кратен wMaxPacketSize, последний пакет также будет меньше размером, чем wMaxPacketSize, поскольку после пересылки данных часто отправляется пакет с нулевыми данными, который укажет на окончание пересылки). В этом случае значение RqPktCount должно быть оставлено нулевым. Если установлен бит AutoReq, то после получения короткого пакета он автоматически сбрасывается.

Выше описан общий случай, когда прикладная программа считывает каждый пакет из FIFO отдельно. Алгоритм немного меняется, когда при конфигурации ядра выбрана опция автоматического объединения пакетов поточной пересылки. (Это можно определить по установленному биту MPRxE (D7) регистра ConfigData.) Если была выбрана эта опция, ядро может принять одновременно до 32 пакетов и объединить их в один пакет в FIFO (при условии, что FIFO буфер достаточно велик, чтобы вместить эти большие пакеты). Размер пакетов, записанных в FIFO, вычисляется как m □□wMaxPacketSize, где RxMaxP[D15:D11] = m – 1.

Чтобы воспользоваться этой функцией, прикладной программе нужно лишь записать нужное значения в регистр RxMaxP (и убедиться, что значение битов 10:0 совпадает со значением поля wMaxPacketSize стандартного дескриптора оконечной точки для данной оконечной точки). Затем в регистр RqPktCount заносится количество пакетов, вычисленное исходя из величины m □□wMaxPacketSize, и далее процесс пересылки этих больших пакетов не отличается от пересылки пакета стандартного размера в режиме поточной пересылки.

При пересылке больших блоков данных использование DMA позволяет избежать необходимости вызывать сервисную подпрограмму для выгрузки каждого пакета.

1.3.12.2.7.16.2.1.3 . ОБРАБОТКА ОШИБОК

Если программа желает закрыть входной поточный канал, она отправит пакет STALL в ответ на маркер чтения. Это приведет к установке бита RxStall (RxCSRL.D6).

1.3.12.2.7.16.2.2 ПОТОЧНЫЕ ТРАНЗАКЦИИ ЗАПИСИ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Поточные транзакции записи используются для пересылки непериодических данных от ведущего устройства к контроллеру функции.

Для использования оконечной точки приемника в режиме ведущего устройства для поточных транзакций чтения доступны четыре дополнительные опции:

- Двойная буферизация пакетов

За исключением случаев, когда используется динамическое определение размера FIFO, если значение, записанное в регистр TxMaxP, меньше или равно половине размера FIFO, выделенного для оконечной точки, то автоматически разрешается двойная буферизация пакетов. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакетов является частью спецификации для FIFO оконечной точки – см. раздел

8.4.1.2). При включенной опции в FIFO могут храниться до двух пакетов, ожидающих пересылки к ведомому устройству.

- DMA

Если для оконечной точки разрешен DMA, то должен вырабатываться запрос DMA, когда оконечная точка готова принять следующий пакет в свой FIFO. Эта функция может быть использована для разрешения контроллеру DMA загружать пакеты в FIFO без участия процессора.

AutoSet

При разрешенной функции AutoSet бит TxPktRdy (TxCSRL.D0) автоматически устанавливается, когда в FIFO загружен пакет размером TxMaxP байт. Эта функция особенно полезна, если для загрузки FIFO используется DMA, поскольку позволяет избежать необходимости участия процессора в загрузке отдельных пакетов при объемной поточной пересылке.

- Автоматическое разделение пакетов

В некоторых системах прикладной программе бывает удобно за одну операцию записывать в оконечную точку объемы данных большие, чем можно переслать за одну операцию USB. В особенности это актуально, когда одна и та же оконечная точка в одних обстоятельствах используется для высокоскоростных пересылок 512 байт, а в других для пересылок со средней скоростью. При работе на средней скорости максимальный объем данных одной пересылки составляет всего 64 байта. Для этого у MUSBMHDRC имеется опция, которая позволяем записывать в оконечные точки поточной пересылки большие объемы данных, которые затем разделяются на пакеты соответствующего (указанного) размера для пересылки по шине USB. Наличие этой опции определяется по установленному биту MPTxE (D6) регистра ConfigData (см. раздел 1.3.12.2.7.3.3.5).

Информация о необходимом размере пакета устанавливается в регистре TxMaxP (см. раздел 1.3.12.2.7.3.3.7).

1.3.12.2.7.16.2.2.1 НАЧАЛЬНАЯ НАСТРОЙКА

Прежде чем инициировать поточную транзакцию записи:

- Должен быть установлен адрес целевой функции, как описано в разделе 1.3.12.2.7.3.14.1.
- Для оконечной точки MUSBMHDRC, которая будет использоваться, должен быть записан регистр TxType - биты D7, D6 определяют рабочую скорость, биты D5, D4 = 10 (для выбора поточной пересылки), а биты D3 – D0 содержат номер оконечной точки, содержащийся в дескрипторе оконечной точки передатчика, который передается MUSBMHDRC во время нумерации устройства (см. раздел 3.3.14).
- В регистр TxMaxP для оконечной точки MUSBMHDRC должен быть записан максимальный размер пакета (в байтах) для пересылки. Значение должно совпадать со значением поля wMaxPacketSize стандартного дескриптора для целевой оконечной точки.
- В регистр TxInterval необходимо записать требуемое значение NAK Limit (2 – 215 фреймов/микрофреймов), или нулевое значение, если функция NAK Timeout не требуется.

- В регистре IntrTxE необходимо установить в ‘1’ соответствующий бит разрешения прерывания (если для данной оконечной точки требуется прерывание).
- Следующие биты регистра TxCSR должны быть установлены в соответствии с приведенной ниже таблицей:

D15	AutoSet	0/1	Устанавливается в 1, если необходима функция AutoSet.
D13	Mode	1	Устанавливается в 1 для разрешения FIFO (необходимо только в том случае, когда FIFO используется совместно с оконечной точкой приемника t).
D12	DMAReqEnab	0/1	Устанавливается в 1, если для этой оконечной точки требуются запросы DMA. Примечание: Если установлен в 1, то необходимо также выбрать DMAReqMode (TxCSRH.D2).
D11	FrcDataTog	0	Устанавливается в 0 для разрешения работы флага данных в стандартном режиме.

- При начальном конфигурировании необходимо сбросить в 0 флаг данных либо с помощью битов Data Toggle Write Enable и Data Toggle (TxCSRH.D1и D0), чтобы переключить текущее состояние, либо записью младшего байта регистра TxCSR чтобы установить бит ClrDataTog (D6). Это обеспечивает правильную начальную установку флага данных (управляется MUSBMHDRC автоматически). Кроме того, если в FIFO присутствуют пакеты данных (на что указывает установленный бит FIFONotEmpty (TxCSRL.D1)), они должны быть очищены, при помощи установки бита FlushFIFO (TxCSRL.D3). Примечание: Возможно, этот бит придется устанавливать два раза подряд, если разрешена двойная буферизация пакетов.

1.3.12.2.7.16.2.2.2 . РАБОТА

При необходимости пересылки данных ведомому устройству USB необходимо загрузить в FIFO первый пакет данных (или два пакета при двойной буферизации) и установить бит TxPktRdy в соответствующем регистре TxCSR (D0). Затем MUSBMHDRC отправит маркер записи выбранной оконечной точке приема, а за ним первый пакет данных из FIFO.

При корректном получении данных в ответ приходит пакет ACK, послед чего MUSBMHDRC сбросит бит TxPktRdy (TxCSRL.D0). Если ведомое устройство USB отвечает пакетом STALL, то устанавливается бит RxStall (TxCSRL.D5). Если получен пакет NAK, MUSBMHDRC предпринимает еще одну попытку – и повторяет попытки либо до успешной транзакции, либо до тех пор, пока не будет исчерпано время ожидания NAKLimit, установленное в регистре TxInterval. Если не получено вообще никакого ответа, MUSBMHDRC предпринимает еще две попытки, а затем сообщает об ошибке (установкой бита TxCSRL.D2).

Затем MUSBMHDRC генерирует прерывание соответствующей оконечной точки, по которому CPU должно считать соответствующий регистр TxCSR, чтобы определить, установлены ли биты RxStall (D5), Error (D2) или NAK Timeout (D7), после чего выполнить соответствующие действия. (Если установлен бит NAK Timeout, то MUSBMHDRC может получить указание продолжить попытки выполнить транзакцию (пока снова не истечет установленное время), что достигается сбросом бита NAK Timeout, либо прервать транзакцию, очистив FIFO, а затем сбросив бит NAK Timeout.)

В общем случае размер пакета не должен превышать размер, указанный младшими 11 битами регистра TxMaxP (в который должно быть записано то же значение, что и в поле wMaxPacketSize дескриптора соответствующей оконечной точки. Если необходимо передать блок данных, больший TxMaxP, он должен пересылаться как группа пакетов - каждый как

описано выше. Эти пакеты должны иметь размер TxMaxP[D10:D0], за исключением последнего пакета, содержащего остаток данных.

Исключение из этого правила имеет место в том случае, когда при конфигурации ядра выбрана опция автоматического разделения пакетов поточной пересылки. (Это можно определить по установленному биту MPTxE (D6) регистра ConfigData.) При выборе этой опции в FIFO могут быть записаны пакеты в 32 раза превышающие размер, указанный в TxMaxP[D10:D0] (при условии достаточно большого объема FIFO, чтобы вместить эти большие пакеты), которые затем разделяются ядром на пакеты соответствующего размера для пересылки по USB. Размер пакетов, записанных в FIFO, вычисляется как m □□USB-содержимое, где TxMaxP[D15:D11] = m – 1. Чтобы воспользоваться этой функцией, прикладной программе нужно лишь записать нужное значения в регистр TxMaxP. С точки зрения прикладной программы процесс пересылки этих больших пакетов не отличается от пересылки пакета стандартного размера в режиме поточной пересылки.

Если общий размер блока данных кратен TxMaxP, то ведущее устройство может после пересылки всех данных отправить нулевой пакет. Это делается посредством установки бита TxPktRdy, когда придет следующее прерывание, но без загрузки данных в FIFO.

При пересылке больших блоков данных использование DMA позволяет избежать необходимости вызывать сервисную подпрограмму для выгрузки каждого пакета.

1.3.12.2.7.16.2.2.3 ОБРАБОТКА ОШИБОК

Если целевое устройство желает закрыть выходной поточный канал, оно отправит в ответ пакет STALL. На это укажет установка бита RxStall (TxCSRL.D5).

1.3.12.2.7.16.3 ИСПОЛЬЗОВАНИЕ DMA

Преимущество DMA состоит в том, что оно повышает эффективность работы шины и процессора при загрузке и выгрузке FIFO.

DMA можно использовать при любом типе пересылки, но полезнее всего эта функция при необходимости передавать большие блоки данных через поточную оконечную точку. Протокол USB требует, чтобы большие блоки данных передавались серией пакетов максимальной длины для данной оконечной точки (512 байт для высокой скорости соединения, 64 байта для средней скорости). Последний пакет в серии может быть меньше максимального. Приемник может использовать прием этого «короткого» пакета как сигнал окончания пересылки (в конце серии может передаваться пакет с нулевыми данными, если размер блока данных кратен максимальному размеру пакета).

Возможности DMA, предоставляемые MUSBMHDRC, могут использоваться как в режиме ведущего, таки и в режиме ведомого устройства, чтобы лишний раз не прерывать работу процессора после каждого отдельного пакета - прерывание поступит только после завершения всей пересылки.

В следующих разделах описываются основные действия, необходимые для использования DMA при некоторых стандартных типах пересылок через поточные оконечные точки передатчика и приемника. Эти действия могут выполняться и при использовании как встроенного контроллера DMA (если он имплементирован в ядро), таки внешнего контроллера DMA.

1.3.12.2.7.16.3.1 ИСПОЛЬЗОВАНИЕ DMA С ПОТОЧНЫМИ ОКОНЕЧНЫМИ ТОЧКАМИ ПЕРЕДАТЧИКА

Для оконечных точек передатчика линия запроса DMA переходит в состояние с высоким уровнем, когда FIFO оконечной точки имеет возможность принять пакет данных, и в состояние с низким уровнем после загрузки TxMaxP байт в FIFO. Линия запроса также перейдет в состояние с низким уровнем при установке бита TxPktRdy в регистре TxCSR.

Для использования DMA при передаче больших блоков данных к ведущему устройству USB рекомендуется следующая первоначальная настройка контроллера DMA и MUSBMHDRC.

Контроллер DMA следует запрограммировать для выполнения пакетного чтения DMA пакетов максимального для оконечной точки размера (512 байт при высокой скорости соединения, 64 байта при низкой скорости), когда линия запроса DMA для пересылок оконечной точки переходит из состояния с низким уровнем в состояние с высоким уровнем. Подробная информация по начальной настройке встроенного DMA приводится в разделе 17 спецификации на MUSBMHDRC. Контроллер должен выполнять чтение пакета в ответ на каждый запрос DMA, пока не будет переслан весь блок данных. (Последний пакет может иметь размер меньше максимального.) Затем он должен прервать работу CPU.

При программировании MUSBMHDRC необходимо разрешить AutoSet и режим 1 запроса DMA, установив биты AutoSet, DMAReqEnab и DMAReqMode в регистре TxCSR (биты D15, D12 и D10 соответственно).

Запрограммированный таким образом MUSBMHDRC переведет линию запроса DMA в состояние высокого уровня, когда в FIFO имеется достаточно места для приема пакета. После того, как контроллер DMA загрузит в FIFO пакет максимального размера, будет автоматически установлен бит TxPktRdy. Теперь пакет готов для пересылки ведущему устройству. Загрузив последний пакет, контроллер DMA должен прервать работу процессора. (Встроенный контроллер для этого активизирует сигнал DMA_NINT.) Если последний загруженный пакет имеет размер меньше максимального, бит TxPktRdy не устанавливается, и поэтому разрешение на пересылку последнего пакета должно быть выдано вручную (то есть, CPU). Бит TxPktRdy также необходимо устанавливать вручную в том случае, когда последний пакет имел максимальный размер, и требуется отправить нулевой пакет, чтобы указать на окончание пересылки.

Примечание: Если при работе в режиме ведущего устройства ядро не сможет успешно передать пакет три раза подряд, то в регистре TxCSR будет установлен бит Error (TxCSR.D2), и линия запроса DMA останется запрещенной до тех пор, пока этот бит не будет сброшен. Необходимо помнить, что бит DMAReqMode в регистре TxCSR не следует сбрасывать ни раньше, ни во время того цикла, когда сбрасывается бит DMAReqEnab.

1.3.12.2.7.16.3.2 ИСПОЛЬЗОВАНИЕ DMA С ПОТОЧНЫМИ ОКОНЕЧНЫМИ ТОЧКАМИ ПРИЕМНИКА

Поведение линии запроса DMA для оконечной точки приемника зависит от режима запроса DMA, выбранного при помощи регистра RxCSR (D11). В режиме 0 линия запроса DMA для оконечной точки приемника переходит в состояние с высоким уровнем, когда в FIFO оконечной точки доступен пакет данных, и в состояние с низким уровнем после прочтения последнего

байта в пакете данных – или при сбросе бита RxPktRdy в регистре RxCSR. В режиме 1 линия запроса DMA для оконечной точки приемника переходит в состояние с высоким уровнем только если полученный пакет имеет максимальный размер (установленный в регистре RxMaxP). Если полученный пакет имеет другой размер, линия запроса DMA остается в состоянии с низким уровнем, в результате чего пакет остается в FIFO при установленном бите RxPktRdy. В этом случае генерируется прерывание оконечной точки приемника (если разрешено).

Режимы запроса DMA изначально предназначены для пересылки больших пакетов данных к поточной оконечной точке. Протокол USB требует, чтобы такие пакеты разбивались на группы пакетов максимального размера (512 для высокой скорости соединения, 64 для средней). Последний пакет группы может иметь размер меньше максимального (или нулевой, если общий размер пересылки кратен максимальному размеру пакета), и приемник тогда может интерпретировать этот «короткий» пакет как сигнал окончания пересылки. Режим 1 запроса DMA может использоваться с соответственно запрограммированным контроллером для того, чтобы лишний раз не прерывать работу процессора после каждого отдельного пакета - прерывание поступит только после завершения всей пересылки. Примечание: Если режим запроса DMA переключается с 1 на 0, то если в FIFO имеется пакет данных, то линия запроса будет активирована, чтобы разрешить загрузку этого ‘полученного ранее’ пакета.

1.3.12.2.7.17 ТРАНЗАКЦИИ ПО ПРЕРЫВАНИЮ ПРИ СРЕДНЕЙ СКОРОСТИ СОЕДЕНИНИЯ / НИЗКОЙ ПРОПУСКНОЙ СПОСОБНОСТИ

1.3.12.2.7.17.1 ТРАНЗАКЦИИ ПО ПРЕРЫВАНИЮ В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Транзакции чтения по прерыванию используют тот же протокол, что и поточные транзакции чтения (описанные в разделе 1.3.12.2.7.16.1.1), и их можно использовать аналогичным образом. Транзакции записи по прерыванию используют почти такой же протокол, как и поточные транзакции записи (описанные в разделе 1.3.12.2.7.16.2), и их можно использовать аналогичным образом.

Тем не менее, следует помнит, что оконечные точки записи в MUSBMHDRC, работающем в режиме ведомого устройства, поддерживают одну функцию для транзакций записи по прерыванию, не поддерживаемую для поточных транзакций записи –непрерывное переключение бита флага данных. Эта функция включается установкой бита FrcDataTog в регистре TxCSR (D11). Когда этот бит установлен в ‘1’, the MUSBMHDRC будет считать, что пакет успешно отправлен, и переключит бит данных для оконечной точки, независимо от того, получен ли от ведущего устройства пакет подтверждения ACK.

Еще одно отличие заключается в том, что оконечные точки в режиме пересылки по прерыванию не поддерживают протокол PING управления потоком данных. Это означает, что MUSBMHDRC не должен отвечать согласованием NYET – только пакетами ACK/NAK/STALL. Для этого необходимо установить в ‘1’ бит DisNyet в регистре RxCSR (D12), чтобы запретить передачу согласования NYET в режиме с высокой скоростью соединения.

Для работы с оконечными точками в режиме пересылки по прерыванию можно использовать DMA, но это не дает особого выигрыша, поскольку в режиме пересылки по прерыванию оконечные точки обычно пересылают все данные одним пакетом.

1.3.12.2.7.17.2 ТРАНЗАКЦИИ ПО ПРЕРЫВАНИЮ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

Когда MUSBMHDRC выступает в роли ведущего устройства, транзакции с оконечной точкой в режиме пересылки по прерыванию очень похожи на аналогичные транзакции в режиме поточной пересылки (описаны в разделах 1.3.12.2.7.16.1.1 и 1.3.12.2.7.16.2соответственно) – за исключением того, что поддерживаются высокоскоростные транзакции по прерыванию.

Главное отличие в отношении порядка работы заключается в том, что в биты RxType[5:4] и TxType[5:4] необходимо записать 11 (что указывает на транзакцию по прерыванию), а не 10.

Требуемый интервал опроса необходимо записать в регистры RxInterval/TxInterval (см. разделы 1.3.12.2.7.3.3.17 и 1.3.12.2.7.3.3.15).

1.3.12.2.7.18 ИЗОХРОННЫЕ ТРАНЗАКЦИИ ПРИ СРЕДНЕЙ СКОРОСТИ СОЕДЕНИНИЯ / НИЗКОЙ ПРОПУСКНОЙ СПОСОБНОСТИ

1.3.12.2.7.18.1 ИЗОХРОННЫЕ ТРАНЗАКЦИИ В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

1.3.12.2.7.18.1.1 ИЗОХРОННЫЕ ТРАНЗАКЦИИ ЧТЕНИЯ

Изохронные транзакции чтения используются для пересылки периодических данных от контроллера функции к ведущему устройству. В этом разделе представлено использование режима ведомого устройства изохронных оконечных точек передатчика со средней скоростью соединения и изохронных оконечных точек передатчика с высокой скоростью соединения и низкой пропускной способностью (1 пакет на микрофрейм). Оконечные точки передатчика с высокой скоростью соединения и высокой пропускной способностью (> 8 Мб/с) описаны в следующем разделе.

Для использования оконечной точки передатчика в режиме ведомого устройства для изохронных транзакций чтения доступны три дополнительные опции:

- Двойная буферизация пакетов

За исключением случаев, когда используется динамическое определение размера FIFO, если значение, записанное в регистр TxMaxP, меньше или равно половине размера FIFO, выделенного для оконечной точки, то автоматически разрешается двойная буферизация пакетов. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакетов является частью спецификации для FIFO оконечной точки – см. раздел 1.3.12.2.7.3.13.2.4). При включенной опции в FIFO могут храниться до двух пакетов, ожидающих пересылки к ведомому устройству. Примечание: Для изохронных транзакций обычно рекомендуется двойное буферирование пакетов, чтобы избежать ошибок опустошения буфера (см. раздел ‘Работа’ ниже).

- DMA

Если для оконечной точки разрешен DMA, то должен вырабатываться запрос DMA, когда оконечная точка готова принять следующий пакет в свой FIFO. Эта функция может быть использована для разрешения контроллеру DMA загружать пакеты в FIFO без участия

процессора. Тем не менее, эта функция не очень эффективна при работе с изохронными оконечными точками, поскольку передаваемые пакеты часто имеют размер, отличающийся от максимального, и после каждого пакета необходимо обращаться к регистру TxCSR, чтобы проверить наличие ошибок опустошения буфера.

- AutoSet

При разрешенной функции AutoSet для оконечной точки с низкой пропускной способностью бит TxPktRdy (TxCSRL.D0) автоматически устанавливается, когда в FIFO загружен пакет размером TxMaxP байт. Тем не менее, эта функция не очень эффективна при работе с изохронными оконечными точками, поскольку передаваемые пакеты часто имеют размер, отличающийся от максимального, и после каждого пакета необходимо обращаться к регистру TxCSR, чтобы проверить наличие ошибок опустошения буфера.

1.3.12.2.7.18.1.1.1 НАЧАЛЬНАЯ НАСТРОЙКА

При конфигурировании оконечной точки передатчика для изохронных транзакций в регистр TxMaxP следует записать максимальный размер пакета (в байтах) для оконечной точки. Это значение должно совпадать со значением поля wMaxPacketSize стандартного дескриптора для данной оконечной точки. Кроме того, в регистре IntrTxE должен быть установлен в ‘1’ соответствующий бит разрешения прерывания (если для данной точки требуется прерывание), а старший байт регистра TxCSR должен быть установлен в соответствии с приведенной ниже таблицей (биты D9 – D8 не используются):

D15	AutoSet	0/1	Устанавливается в 1, если необходима функция AutoSet.
D14	ISO	0	Устанавливается в 0 для разрешения протокола изохронной пересылки.
D13	Mode	1	Устанавливается в 1 для разрешения FIFO (необходимо только в том случае, когда FIFO используется совместно с оконечной точкой приемника).
D12	DMAReqEnab	0/1	Устанавливается в 1, если для данной оконечной точки требуется запрос DMA. Примечание: Если установлен в 1, то необходимо также выбрать DMAReqMode (TxCSRH.D2).
D11	FrcDataTog	0	Игнорируется в изохронном режиме.

1.3.12.2.7.18.1.1.2 РАБОТА

Изохронная оконечная точка не поддерживает повторных попыток пересылки данных, и поэтому во избежание опустошения буфера данных отправляемые ведомому устройству данные должны загружаться в буфер до того, как был принят маркер чтения. Ведущее устройство будет присылать один маркер чтения на фрейм (или микрофрейм при высокой скорости соединения), но время внутри фрейма может быть разным. Если маркер чтения принят ближе к концу одного фрейма, а затем в начале следующего, то для перезагрузки FIFO останется мало времени.

Функция AutoSet может использоваться при работе изохронной оконечной точки передатчика с низкой пропускной способностью точно так же, как при работе поточной оконечной точки передатчика. Однако если данные поступают от источника не с постоянной скоростью, синхронизированной с частотой фреймов ведущего устройства, то размер пакетов, отправляемых ведущему устройству, будет увеличиваться или уменьшаться от фрейма к фрейму (или от микрофрейма к микрофрейму) в соответствии со скоростью источника данных. Это означает, что реальный размер пакета не всегда будет совпадать по размеру с TxMaxP, что сделает бесполезной функцию AutoSet.

При отправке пакета ведущему устройству генерируется прерывание, и программа может использовать его для загрузки в FIFO следующего пакета и установки бита TxPktRdy в регистре TxCSR (D0) точно так же, как для поточной оконечной точки передатчика. Поскольку прерывание может прийти практически в любой момент внутри фрейма (микрофрейма), в зависимости от того, когда ведущее устройство запланировало транзакцию, это может привести к иррегулярным запросам на загрузку FIFO. Если источником данных для оконечной точки служит какое-либо внешнее оборудование, возможно, удобнее дожидаться окончания каждого фрейма (микрофрейма), а затем загружать FIFO, поскольку это минимизирует потребность в дополнительной буферизации. Для этого можно использовать либо прерывание SOF (IntrUSB.D3), либо внешний сигнал SOF_PULSE от MUSBMHDRC – по их приходу загружать следующий пакет. Сигнал SOF_PULSE генерируется один раз за фрейм (микрофрейм) при приеме пакета SOF. (MUSBMHDRC также поддерживает счетчик внешних фреймов (микрофреймов), так что у него сохраняется возможность генерировать SOF_PULSE при потере пакета SOF.) Прерывания могут использоваться для установки бита TxPktRdy в регистре TxCSR (D0) и для проверки переполнения/опустошения буфера данных (см. раздел ‘Обработка ошибок’ ниже).

Открытие изохронного канала чтения с двойной буферизацией может стать источником проблем. Двойная буферизация требует, чтобы пакет данных не пересылался до фрейма (микрофрейма), следующего за загрузкой этого пакета. Если функция загружает первый пакет данных по меньшей мере за один фрейм (микрофрейм) до того, как ведущее устройство откроет канал (и начнет отправлять маркеры чтения), проблем не возникает. Но если к моменту загрузки пакета ведущее устройство уже начало отправлять маркеры чтения, пакет может быть отправлен во время того же фрейма (микрофрейма), когда он был загружен, в зависимости от того, когда был получен маркер чтения, раньше или позже. Эту потенциальную проблему можно исключить установкой бита ISO Update в регистре Power (D7). Когда этот бит установлен в 1, то любой пакет данных, загруженный в FIFO оконечной точки передатчика, не будет передаваться до получения следующего пакета SOF, что предотвратит преждевременную пересылку пакета.

1.3.12.2.7.18.1.1.3 ОБРАБОТКА ОШИБОК

Если при получении маркера чтения в FIFO оконечной точки нет данных, то оконечная точка отправляет ведущему устройству пакет нулевых данных и устанавливает бит UnderRun в регистре TxCSR (D2). Это свидетельствует, что программа не успевает с нужной скоростью поставлять данные для ведущего устройства. Дальнейшая обработка этой ошибки определяется прикладной программой.

Если программа загружает один пакет за фрейм (микрофрейм) и обнаруживает установленный бит TxPktRdy в регистре TxCSR (D0), когда хочет загрузить следующий пакет, это указывает что пакет данных не был отправлен (возможно, из-за поврежденного маркера чтения). Далее прикладная программа должна отреагировать на это состояние: она может очистить неотправленный пакет, установив бит FlushFIFO в регистре TxCSR (D3), или пропустить текущий пакет.

1.3.12.2.7.18.1.2 ИЗОХРОННЫЕ ТРАНЗАКЦИИ ЗАПИСИ

Изохронные транзакции записи используются для пересылки периодических данных от ведущего устройства к контроллеру функции. В этом разделе представлено использование

режима ведомого устройства изохронных оконечных точек приемника со средней скоростью соединения и изохронных оконечных точек приемника с высокой скоростью соединения и низкой пропускной способностью (1 пакет на микрофрейм). Оконечные точки приемника с высокой скоростью соединения и высокой пропускной способностью (> 8 Мб/с) описаны в следующем разделе.

Для использования оконечной точки приемника в режиме ведомого устройства для изохронных транзакций записи доступны три дополнительные опции:

- Двойная буферизация пакетов

За исключением случаев, когда используется динамическое определение размера FIFO, если значение, записанное в регистр RxMaxP, меньше или равно половине размера FIFO, выделенного для оконечной точки, то автоматически разрешается двойная буферизация пакетов. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакетов является частью спецификации для FIFO оконечной точки – см. раздел 8.4.2.2). При включенной опции в FIFO могут храниться до двух пакетов, ожидающих пересылки к ведущему устройству. Примечание: Для изохронных транзакций обычно рекомендуется двойное буферирование пакетов, чтобы избежать ошибок переполнения буфера (см. раздел ‘Работа’ ниже).

- DMA

Если для оконечной точки разрешен DMA, то запрос DMA вырабатывается, когда в FIFO оконечной точки имеется пакет данных. Эта функция может быть использована для разрешения контроллеру DMA выгружать пакеты из FIFO без участия процессора. Тем не менее, эта функция не очень эффективна при работе с изохронными оконечными точками, поскольку передаваемые пакеты часто имеют размер, отличающийся от максимального, и после каждого пакета необходимо обращаться к регистру RxCSR, чтобы проверить наличие ошибок переполнения буфера или CRC.

- AutoClear

При разрешенной функции AutoClear бит RxPktRdy (RxCSRL.D0) автоматически сбрасывается, когда из FIFO выгружен пакет размером RxMaxP байт (исключения см. в описании регистра). Тем не менее, эта функция не очень эффективна при работе с изохронными оконечными точками, поскольку передаваемые пакеты часто имеют размер, отличающийся от максимального, и после каждого пакета необходимо обращаться к регистру RxCSR, чтобы проверить наличие ошибок переполнения буфера или CRC.

1.3.12.2.7.18.1.2.1 НАЧАЛЬНАЯ НАСТРОЙКА

При конфигурировании оконечной точки передатчика для изохронных транзакций в регистр RxMaxP следует записать максимальный размер пакета (в байтах) для оконечной точки. Это значение должно совпадать со значением поля wMaxPacketSize стандартного дескриптора для данной оконечной точки. Кроме того, в регистре IntrRxE должен быть установлен в ‘1’ соответствующий бит разрешения прерывания (если для данной точки требуется прерывание), а старший байт регистра RxCSR должен быть установлен в соответствии с приведенной ниже таблицей (биты D10 – D8 не используются/только для чтения):

D15	AutoClear	0/1	Устанавливается в 1, если необходима функция AutoClear.
D14	ISO	1	Устанавливается в 1 для разрешения протокола изохронной пересылки.

D13	DMAReqEnab	0/1	Устанавливается в 1, если для данной оконечной точки требуется запрос DMA. Примечание: Если установлен в 1, то необходимо также выбрать DMAReqMode (RxCSRH.D3).
D12	DisNyet	0	Игнорируется в изохронном режиме.

1.3.12.2.7.18.1.2.2 РАБОТА

Изохронная оконечная точка не поддерживает повторных попыток пересылки данных, и поэтому во избежание переполнения буфера данных в FIFO должно быть место для приема пакета. Ведущее устройство будет присылать один пакет на фрейм (или микрофрейм при высокой скорости соединения), но время внутри фрейма может быть разным. Если пакет принят ближе к концу одного фрейма, а затем в начале следующего, то для выгрузки FIFO останется мало времени. По этой причине обычно требуется двойная буферизация пакетов.

Функция AutoClear может использоваться при работе изохронной оконечной точки приемника точно так же, как при работе с поточной оконечной точкой приемника. Однако если данные не принимаются с постоянной скоростью, синхронизированной с частотой фреймов ведущего устройства, размер пакетов, отправляемых ведущим устройством, будет увеличиваться или уменьшаться от фрейма к фрейму (или от микрофрейма к микрофрейму) в соответствии с требуемой скоростью. Это означает, что реальный размер пакета не всегда будет совпадать по размеру с RxMaxP, что сделает бесполезной функцию AutoClear.

При приеме пакета от ведущего устройства генерируется прерывание, и программа может использовать его для выгрузки пакета из FIFO и установки бита RxPktRdy в регистре RxCSR (D0) точно так же, как для поточной оконечной точки приемника. Поскольку прерывание может прийти практически в любой момент внутри фрейма (микрофрейма), в зависимости от того, когда ведущее устройство запланировало транзакцию, это может привести к иррегулярным запросам на выгрузку FIFO. Если приемником данных для оконечной точки служит какое-либо внешнее оборудование, возможно, удобнее дожидаться окончания каждого фрейма (микрофрейма), а затем выгружать FIFO. Для этого можно использовать либо прерывание SOF (IntrUSB.D3), либо внешний сигнал SOF_PULSE от MUSBMHDRC – по их приходу выгружать пакет данных. Сигнал SOF_PULSE генерируется один раз за фрейм (микрофрейм) при приеме пакета SOF. (MUSBMHDRC также поддерживает счетчик внешних фреймов (микрофреймов), так что у него сохраняется возможность генерировать SOF_PULSE при потере пакета SOF.) Прерывания могут использоваться для сброса бита RxPktRdy в регистре RxCSR (D0) и для проверки переполнения/опустошения буфера данных (см. раздел ‘Обработка ошибок’ ниже).

1.3.12.2.7.18.1.2.3 ОБРАБОТКА ОШИБОК

Если в FIFO нет места для хранения пакета, полученного от ведущего устройства, то устанавливает бит OverRun в регистре RxCSR (D2). Это свидетельствует, что программа не успевает с нужной для ведущего устройства скоростью выгружать данные. Дальнейшая обработка этой ошибки определяется прикладной программой.

Если программа обнаруживает в принятом пакете ошибку CRC, то она все равно сохранит пакет в FIFO, затем установит бит RxPktRdy (RxCSR.D0) и бит DataError (RxCSRL.D3). Дальнейшая обработка этой ошибки определяется прикладной программой.

1.3.12.2.7.18.2 ИЗОХРОННЫЕ ТРАНЗАКЦИИ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

1.3.12.2.7.18.2.1 ИЗОХРОННЫЕ ТРАНЗАКЦИИ ЧТЕНИЯ

Изохронные транзакции чтения используются для пересылки периодических данных от контроллера функции к ведущему устройству. В этом разделе представлено использование режима ведущего устройства изохронных оконечных точек приемника со средней скоростью соединения и изохронных оконечных точек приемника с высокой скоростью соединения и низкой пропускной способностью (1 пакет на микрофрейм). Оконечные точки с высокой скоростью соединения и высокой пропускной способностью (> 8 Мб/с) описаны в следующем разделе.

Для использования оконечной точки приемника в режиме ведомого устройства для изохронных транзакций чтения доступны четыре дополнительные опции:

- Двойная буферизация пакетов

При разрешенной двойной буферизации пакетов один пакет может приниматься, а другой в это время считываться. За исключением случаев, когда используется динамическое определение размера FIFO, если значение, записанное в регистр RxMaxP, меньше или равно половине размера FIFO, выделенного для оконечной точки, то автоматически разрешается двойная буферизация пакетов. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакетов является частью спецификации для FIFO оконечной точки – см. раздел 1.3.12.2.7.3.13.2.4). Примечание: Для изохронных транзакций обычно рекомендуется двойное буферирование пакетов, чтобы избежать переполнения буфера данных (см. раздел ‘Работа’ ниже).

- DMA

Если для оконечной точки разрешен DMA, то запрос DMA вырабатывается, когда в FIFO оконечной точки имеется пакет данных. Эта функция может быть использована для разрешения контроллеру DMA выгружать пакеты из FIFO без участия процессора. Тем не менее, эта функция не очень эффективна при работе с изохронными оконечными точками, поскольку передаваемые пакеты часто имеют размер, отличающийся от максимального.

- AutoClear

При разрешенной функции AutoClear бит RxPktRdy (RxCSRL.D0) автоматически сбрасывается, когда из FIFO выгружен пакет размером RxMaxP байт (исключения см. в описании регистра). Тем не менее, эта функция не очень эффективна при работе с изохронными оконечными точками, поскольку передаваемые пакеты часто имеют размер, отличающийся от максимального, и после каждого пакета необходимо обращаться к регистру RxCSR, чтобы проверить наличие ошибок переполнения буфера или CRC.

- AutoReq(uest)

При разрешенной функции AutoReq(uest) будет автоматически устанавливаться бит ReqPkt (RxCSRL.D5), кода сбрасывается бит RxPktRdy.

1.3.12.2.7.18.2.1.1 НАЧАЛЬНАЯ НАСТРОЙКА

Прежде чем инициировать изохронную транзакцию чтения:

Должен быть установлен адрес целевой функции, как описано в разделе 1.3.12.2.7.3.14.1.

- Для оконечной точки MUSBMHDRC, которая будет использоваться, должен быть записан регистр RxType - биты D7, D6 определяют рабочую скорость, биты D5, D4 = 01 (для выбора изохронной пересылки, а биты D3 – D0 содержат номер оконечной точки, который содержится в дескрипторе оконечной точки приемника, который передается в MUSBMHDRC во время нумерации устройства (см. раздел 1.3.12.2.7.3.3.16).
- В регистр RxInterval для оконечной точки MUSBMHDRC следует записать требуемый интервал транзакции (обычно одна транзакция на каждый фрейм /микрофрейм) – см. раздел 1.3.12.2.7.3.3.17.
- В регистр RxMaxP для оконечной точки MUSBMHDRC должен быть записан максимальный размер пакета (в байтах) для пересылки. Значение должно совпадать со значением поля wMaxPacketSize стандартного дескриптора для целевой оконечной точки.
- В регистре IntrRxE необходимо установить в ‘1’ соответствующий бит разрешения прерывания (если для данной оконечной точки требуется прерывание).
- Следующие биты регистра RxCSR должны быть установлены в соответствии с приведенной ниже таблицей:
-

D15	AutoClear	0/1	Устанавливается в 1, если необходима функция AutoClear.
D14	AutoReq	0/1	Устанавливается в 1, если необходима функция AutoRequest.
D13	DMAReqEnab	0/1	Устанавливается в 1, если для этой точки требуется запрос DMA. Примечание: Если установлен в 1, то необходимо также выбрать DMAReqMode (RxCSRH.D3).
D12	DisNyet	0	Игнорируется в изохронном режиме.

-

1.3.12.2.7.18.2.1.2 РАБОТА

Работа начинается с того, что CPU устанавливает бит ReqPkt (RxCSRL.D5). В ответ MUSBMHDRC отправляет цели маркер чтения.

При приеме пакета генерируется прерывание, которое может быть использовано программой для выгрузки пакета из FIFO и сброса бита RxPktRdy в регистре RxCSR (D0) точно так же, как для поточной оконечной точки приемника. Поскольку прерывание может прийти практически в любой момент внутри фрейма (микрофрейма), это может привести к иррегулярным запросам на выгрузку FIFO. Если приемником данных для оконечной точки служит какое-либо внешнее оборудование, возможно, удобнее дожидаться окончания каждого фрейма (микрофрейма), а затем выгружать FIFO. Для этого можно использовать сигнал SOF_PULSE от MUSBMHDRC – по его приходу выгружать пакет данных. Сигнал SOF_PULSE генерируется один раз за фрейм (микрофрейм). Прерывания могут также использоваться для сброса бита RxPktRdy в регистре RxCSR

Функция AutoClear может использоваться при работе с изохронной оконечной точкой приемника точно так же, как при работе с поточной оконечной точкой приемника. Однако если данные не принимаются с постоянной скоростью, синхронизированной с частотой фреймов MUSBMHDRC, размер пакетов, отправляемых ведущим устройством, будет увеличиваться или

уменьшаться от фрейма к фрейму (или от микрофрейма к микрофрейму) в соответствии с требуемой скоростью. Это означает, что реальный размер пакета не всегда будет совпадать по размеру с RxMaxP, что сделает бесполезной функцию AutoClear.

1.3.12.2.7.18.2.1.3 ОБРАБОТКА ОШИБОК

Если при приеме пакета обнаружится ошибка CRC или битстаффинга, то пакет все равно сохранится в FIFO, но будет установлен бит, который указывает на возможное искажение данных.

1.3.12.2.7.18.2.2 ИЗОХРОННЫЕ ТРАНЗАКЦИИ ЗАПИСИ

Изохронные транзакции записи используются для пересылки периодических данных от ведущего устройства к контроллеру функции. В этом разделе описано использование изохронных оконечных точек передатчика со средней скоростью соединения и изохронных оконечных точек передатчика с высокой скоростью соединения и низкой пропускной способностью (1 пакет на микрофрейм). Оконечные точки с высокой скоростью соединения и высокой пропускной способностью (> 8 Мб/с) описаны в следующем разделе.

Для использования оконечной точки передатчика в режиме ведущего устройства для пересылки данных доступны три дополнительные опции:

- Двойная буферизация пакетов

За исключением случаев, когда используется динамическое определение размера FIFO, если значение, записанное в регистр TxMaxP, меньше или равно половине размера FIFO, выделенного для оконечной точки, то автоматически разрешается двойная буферизация пакетов. (При выборе динамического определения размера FIFO использование одиночной или двойной буферизации пакетов является частью спецификации для FIFO оконечной точки – см. раздел 8.4.2.2). При включенной опции в FIFO могут храниться до двух пакетов, ожидающих пересылки к ведомому устройству.

- DMA

Если для оконечной точки разрешен DMA, то должен вырабатываться запрос DMA, когда оконечная точка готова принять следующий пакет в свой FIFO. Тем не менее, эта функция не очень эффективна при работе с изохронными оконечными точками, поскольку передаваемые пакеты часто имеют размер, отличающийся от максимального.

- AutoSet

При разрешенной функции AutoSet для изохронной оконечной точки с низкой пропускной способностью бит TxPktRdy (TxCSRL.D0) автоматически устанавливается, когда в FIFO загружен пакет размером TxMaxP байт. Тем не менее, эта функция не очень эффективна при работе с изохронными оконечными точками, поскольку передаваемые пакеты часто имеют размер, отличающийся от максимального.

1.3.12.2.7.18.2.2.1 НАЧАЛЬНАЯ НАСТРОЙКА

Прежде чем инициировать изохронную транзакцию записи:

- Должен быть установлен адрес целевой функции, как описано в разделе 1.3.12.2.7.3.14.1.
- Для оконечной точки MUSBMHDRC, которая будет использоваться, должен быть записан регистр TxType - биты D7, D6 определяют рабочую скорость, биты D5, D4 = 01 (для выбора изохронной пересылки), а биты D3 – D0 содержат номер оконечной точки, который содержится в дескрипторе оконечной точки передатчика, который передается в MUSBMHDRC во время нумерации устройства (см. раздел 1.3.12.2.7.3.3.14).
- В регистр TxInterval для оконечной точки MUSBMHDRC следует записать требуемый интервал транзакции (обычно одна транзакция на каждый фрейм /микрофрейм) – см. раздел 1.3.12.2.7.3.3.15.
- В регистр TxMaxP для оконечной точки MUSBMHDRC должен быть записан максимальный размер пакета (в байтах) для пересылки. Значение должно совпадать со значением поля wMaxPacketSize стандартного дескриптора для целевой оконечной точки.
- В регистре IntrTxE необходимо установить в ‘1’ соответствующий бит разрешения прерывания (если для данной оконечной точки требуется прерывание).
- Следующие биты регистра TxCSR должны быть установлены в соответствии с приведенной ниже таблицей:
-

D15	AutoSet	0/1	Устанавливается в 1, если необходима функция AutoSet.
D13	Mode	1	Устанавливается в 1 для разрешения FIFO (необходимо только в том случае, когда FIFO используется совместно с оконечной точкой приемника).
D12	DMAReqEnab	0/1	Устанавливается в 1, если для данной оконечной точки требуется запрос DMA. Примечание: Если установлен в 1, то необходимо также выбрать DMAReqMode (TxCSRH.D2).
D11	FrcDataTog	0	Игнорируется в изохронном режиме.

1.3.12.2.7.18.2.2.2 РАБОТА

Работа начинается с того, что CPU записывает данные в FIFO и устанавливает бит TxPktRdy (TxCSRL.D0). В ответ MUSBMHDRC отправляет цели маркер записи, а затем первый пакет данных из FIFO.

При передаче пакета генерируется прерывание, которое может быть использовано программой для загрузки следующего пакета в FIFO и установки бита TxPktRdy в регистре TxCSR (D0) точно так же, как для поточной оконечной точки передатчика. Поскольку прерывание может прийти практически в любой момент внутри фрейма, в зависимости от того, когда ведущее устройство запланировало транзакцию, это может привести к иррегулярным запросам на загрузку FIFO. Если источником данных для оконечной точки служит какое-либо внешнее оборудование, возможно, удобнее дожидаться окончания каждого фрейма (микрофрейма), а затем загружать FIFO, что минимизирует потребность в дополнительной буферизации. Для этого можно использовать сигнал SOF_PULSE от MUSBMHDRC – по его приходу загружать следующий пакет данных. Сигнал SOF_PULSE генерируется один раз за фрейм (микрофрейм). Прерывания могут также использоваться для установки бита TxPktRdy в регистре TxCSR

Функция AutoSet может использоваться при работе с изохронной оконечной точкой передатчика точно так же, как при работе с поточной оконечной точкой передатчика. Однако

если данные не поступают от источника с постоянной скоростью, синхронизированной с частотой фреймов MUSBMHDRC, размер пакетов будет увеличиваться или уменьшаться от фрейма к фрейму (или от микрофрейма к микрофрейму) в соответствии со скоростью данных источника. Это означает, что реальный размер пакета не всегда будет совпадать по размеру с TxMaxP, что сделает бесполезной функцию AutoSet.

1.3.12.2.7.19 ИЗОХРОННЫЕ ТРАНЗАКЦИИ / ТРАНЗАКЦИИ ПО ПРЕРЫВАНИЮ С ВЫСОКОЙ ПРОПУСКНОЙ СПОСОБНОСТЬЮ

Изохронные транзакции / транзакции по прерыванию с высокой пропускной способностью используют, в основном, тот же протокол, что и другие изохронные транзакции / транзакции по прерыванию. Однако имеются и некоторые особенности.

-
- Изохронные транзакции / транзакции по прерыванию с высокой пропускной способностью могут выполняться только в том случае, когда ядро MUSBMHDRC сконфигурировано для поддержки этих транзакций (см. раздел три руководства пользователя MUSBMHDRC).

Ядро также необходимо сконфигурировать таким образом, чтобы оконечные точки, используемые для транзакций с высокой пропускной способностью, имели FIFO достаточного размера для помещения данных по меньшей мере одного пакета высокой пропускной способности (более 1 кБ и до 3 кБ).

- При установке максимального размера пакет для оконечной точки в регистре TxMaxP/RxMaxP необходимо также установить максимальное число транзакций на микрофрейм при помощи битов D11 and D12 этого же регистра.

Это максимальное количество транзакций (2 или 3) также отражает максимальное число частей, которыми может быть передан любой пакет высокой пропускной способности, что в свою очередь определяет максимальный размер пакета как в 2 или 3 раза больший, чем максимальное содержимое, указанное для оконечной точки в том же регистре.

Примечание: Максимальное содержимое, которое может быть передано за одну трансакцию, составляет 1 килобайт.

- При отправке пакетов прикладная программа должна устанавливать бит TxPktRdy. Аналогичным образом, при выгрузке пакетов из FIFO оконечной точки приемника прикладная программа должна сбрасывать бит RxPktRdy.

При транзакциях с высокой пропускной способностью для установки и сброса этих битов невозможно использовать функции AutoSet и AutoClear.

- Пересылка пакетов несколькими частями способствует появлению еще одного типа ошибок – передачи неполных пакетов.

Для оконечных точек передатчика в режиме ведомого устройства ошибка возникает, когда MUSBMHDRC не получает нужного количества маркеров записи от ведомого устройства, чтобы переслать все части пакета данных. Для транзакций по прерыванию с высокой пропускной способностью в режиме ведущего устройства ошибка возникает, когда ядро не получает ответа от устройства, которому был отправлен пакет. В обоих случаях MUSBMHDRC установит бит IncompTx в регистре TxCSR (D7).

Для оконечных точек приемника проблема возникает в том случае, когда PID принятых частей пакета данных указывают, что один или несколько частей пакета не были приняты. Тогда MUSBMHDRC устанавливает бит IncompRx в регистре RxCSR (D8). В основном этот бит устанавливается только если ядро работает в режиме ведомого устройства, однако он может устанавливаться и в режиме ведущего устройства, но только в том случае, когда устройство, с которым связан MUSBMHDRC, не отвечает в соответствии с протоколом USB.

1.3.12.2.7.20 АЛГОРИТМЫ ТРАНЗАКЦИЙ В РЕЖИМЕ ВЕДОМОГО УСТРОЙСТВА

Примечание: Действия ведущего устройства показаны на белом фоне, действия MUSBMHDRC - на сером.

1.3.12.2.7.20.1 УПРАВЛЯЮЩИЕ ТРАНЗАКЦИ

1.3.12.2.7.20.1.1 ФАЗА SET UP

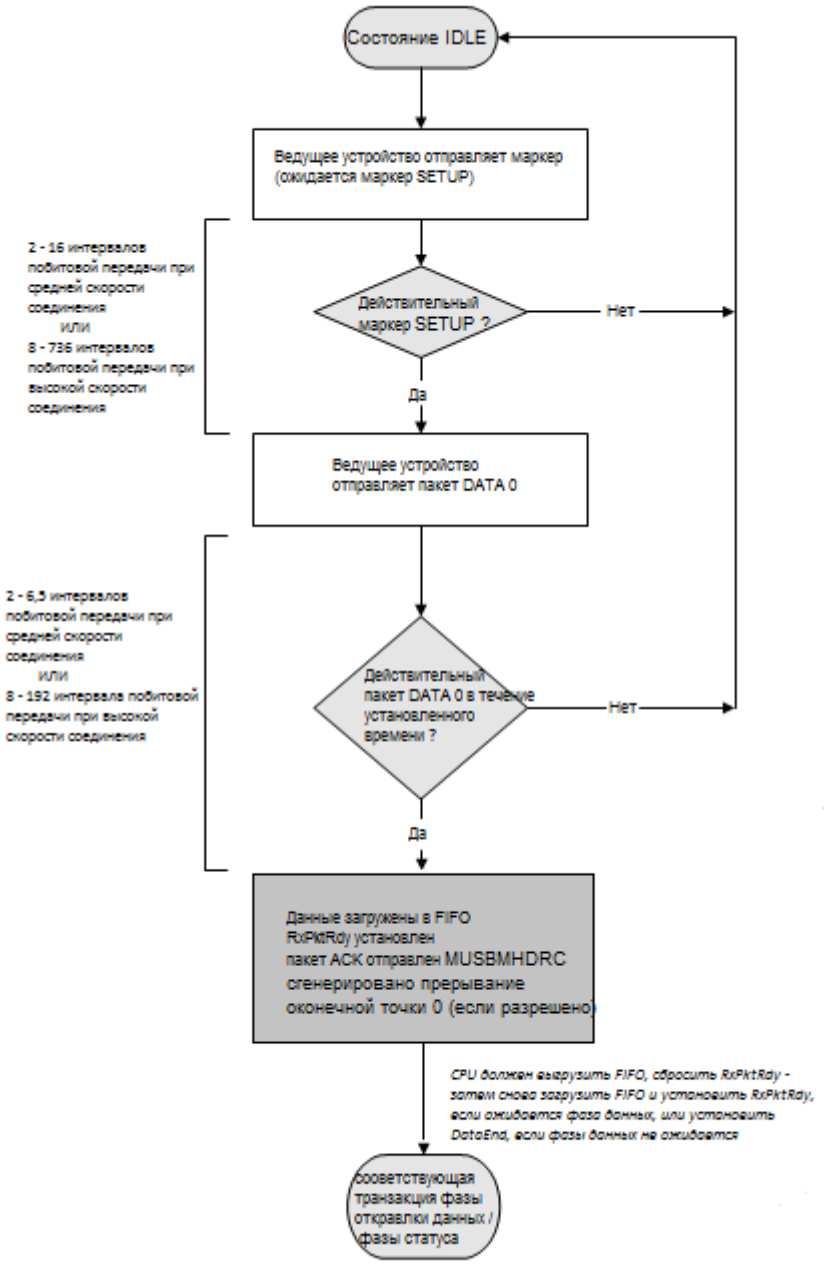


Рисунок 1.274

1.3.12.2.7.20.1.2 ФАЗА ПРИЕМА ДАННЫХ

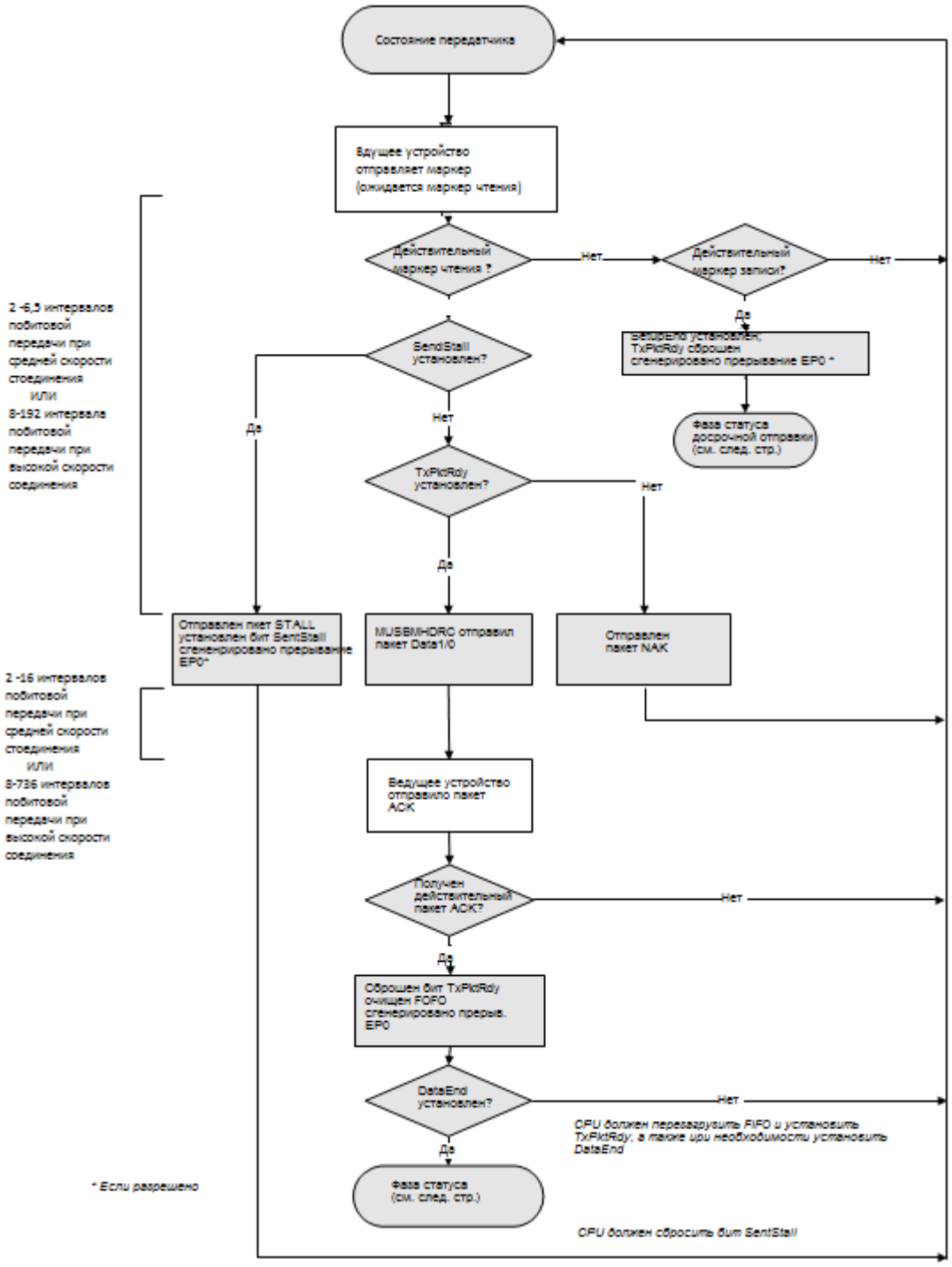


Рисунок 1.275

1.3.12.2.7.20.1.3 ПОСЛЕ ФАЗЫ СТАТУСА

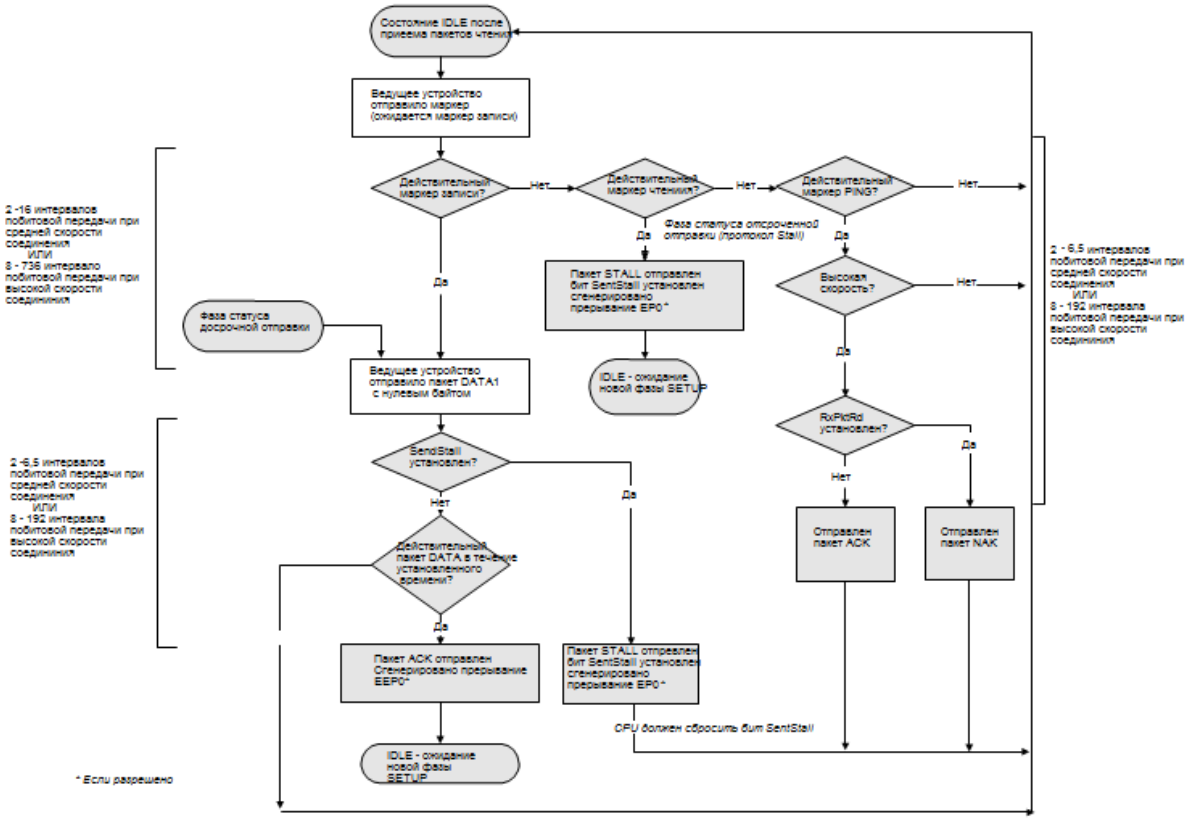


Рисунок 1.276

1.3.12.2.7.20.1.4 ФАЗА ОТПРАВКИ ДАННЫХ

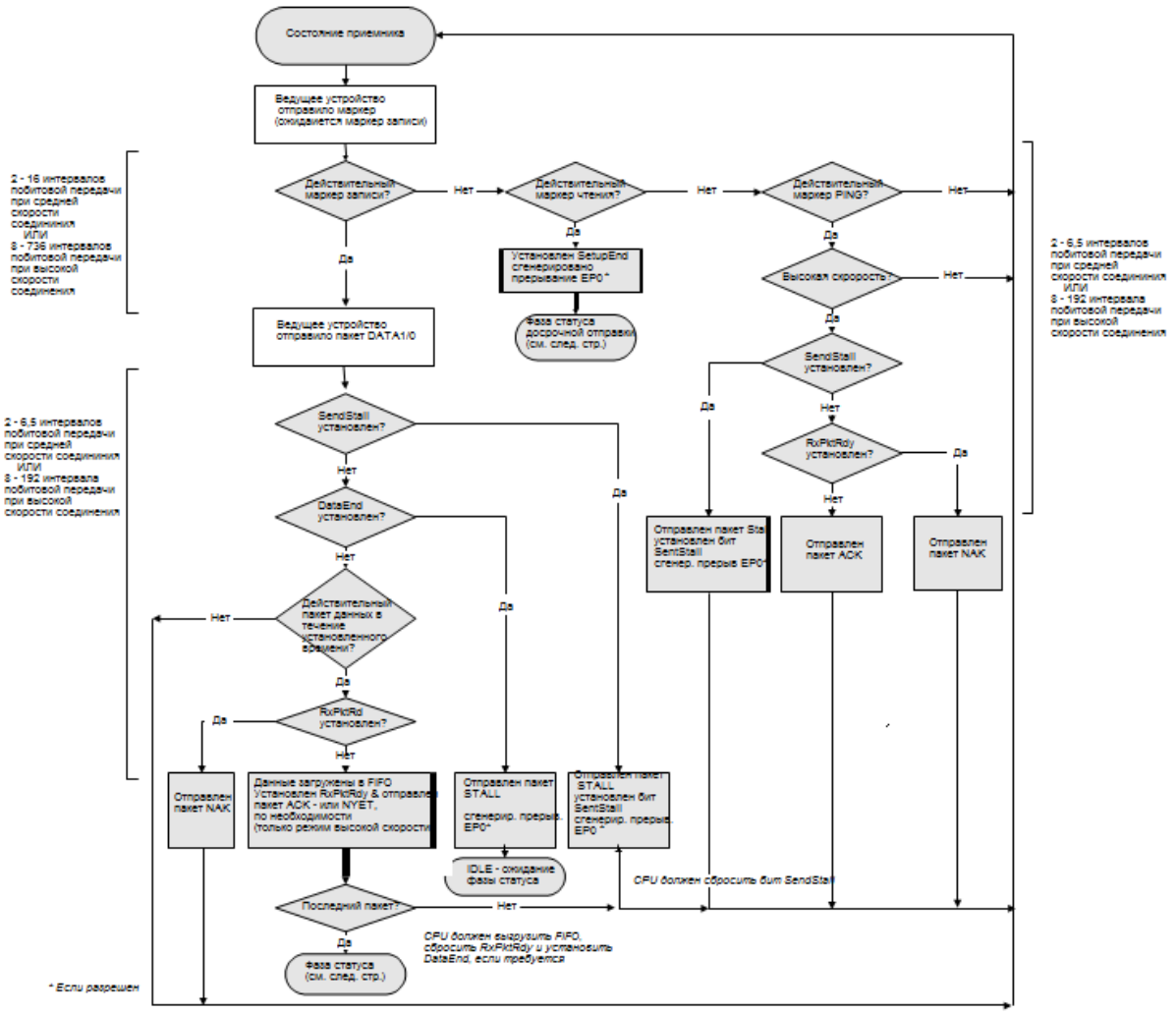


Рисунок 1.277

1.3.12.2.7.20.1.5 ПОСЛЕ ФАЗЫ СТАТУСА

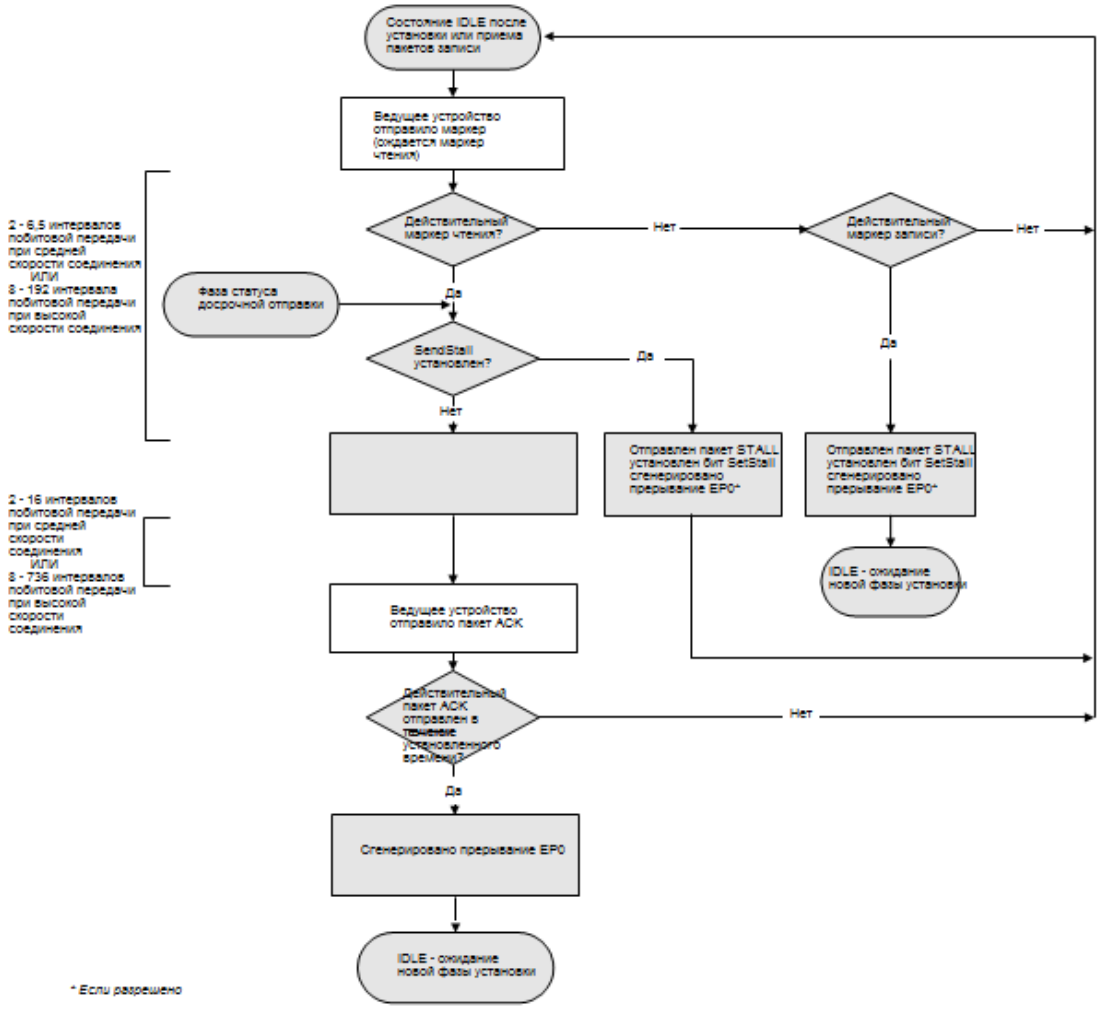


Рисунок 1.278

1.3.12.2.7.20.2 ПОТОЧНЫЕ ТРАНЗАКЦИИ / ТРАНЗАКЦИИ ПО ПРЕРЫВАНИЮ С НИЗКОЙ ПРОПУСКНОЙ СПОСОБНОСТЬЮ

1.3.12.2.7.20.2.1 ТРАНЗАКЦИЯ ЧТЕНИЯ

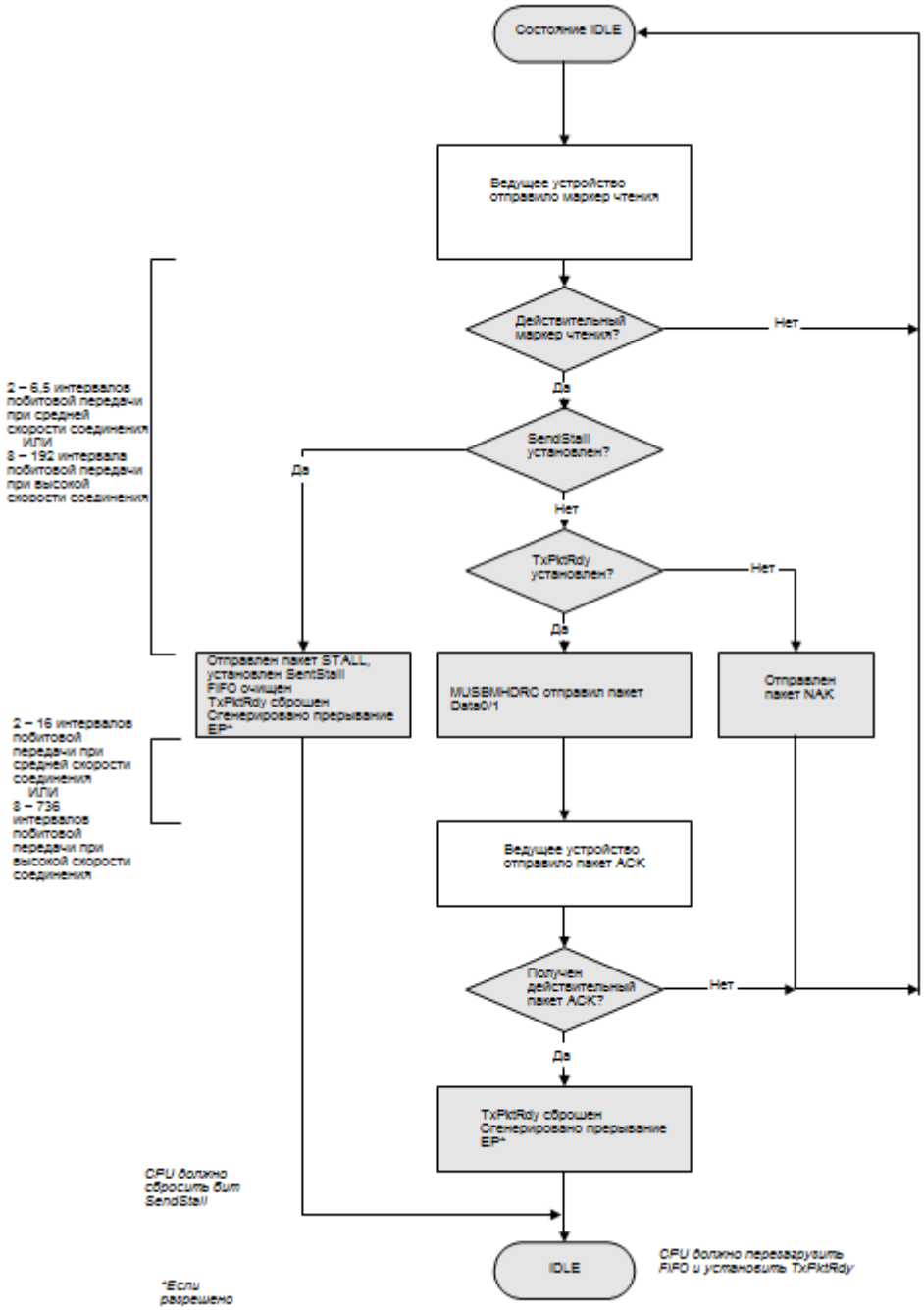


Рисунок 1.279

1.3.12.2.7.20.2.2 ТРАНЗАКЦИЯ ЗАПИСИ

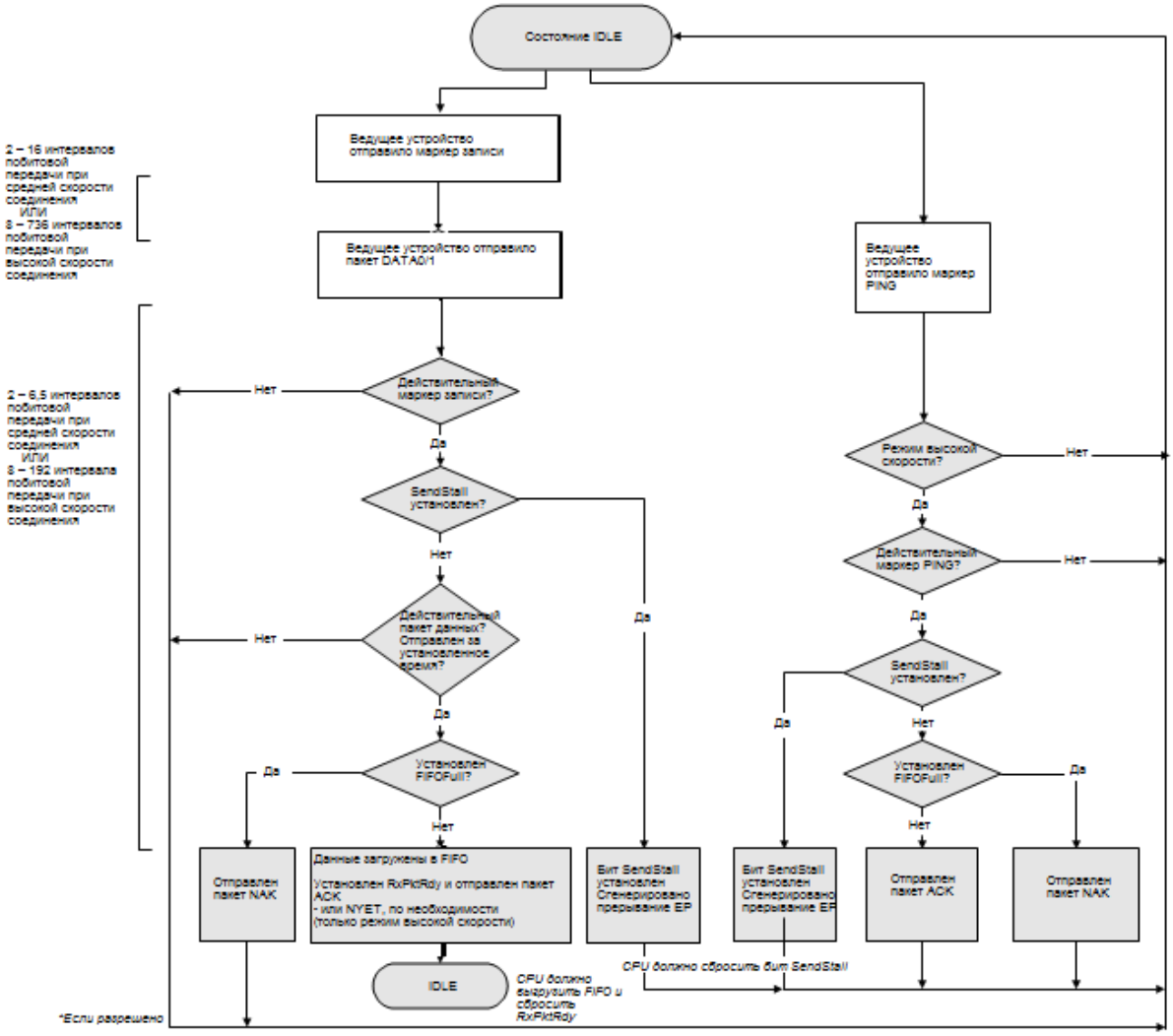


Рисунок 1.280

1.3.12.2.7.20.3 . ИЗОХРОННЫЕ ТРАНЗАКЦИИ СО СРЕДНЕЙ СКОРОСТЬЮ СОЕДИНЕНИЯ / НИЗКОЙ ПРОПУСКНОЙ СПОСОБНОСТЬЮ

1.3.12.2.7.20.3.1 ТРАНЗАКЦИЯ ЧТЕНИЯ

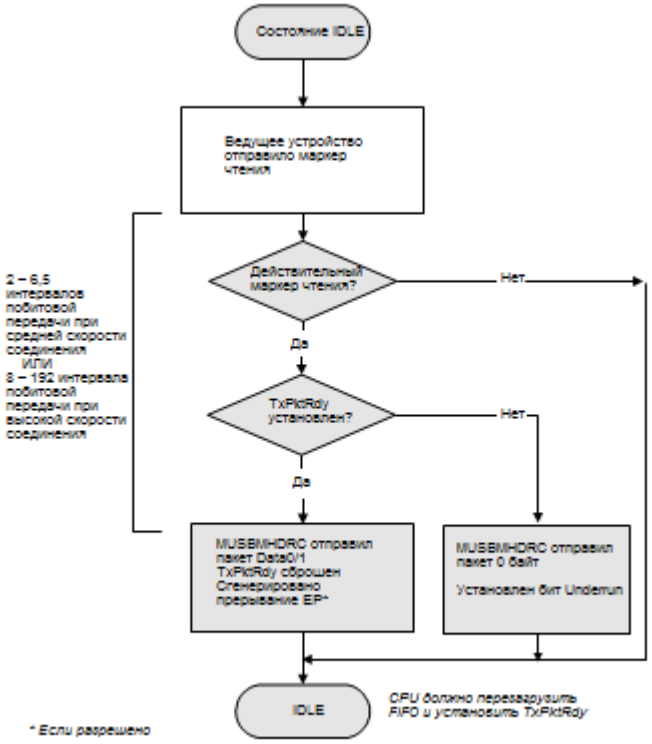


Рисунок 1.281

1.3.12.2.7.20.3.2 . ТРАНЗАКЦИЯ ЗАПИСИ

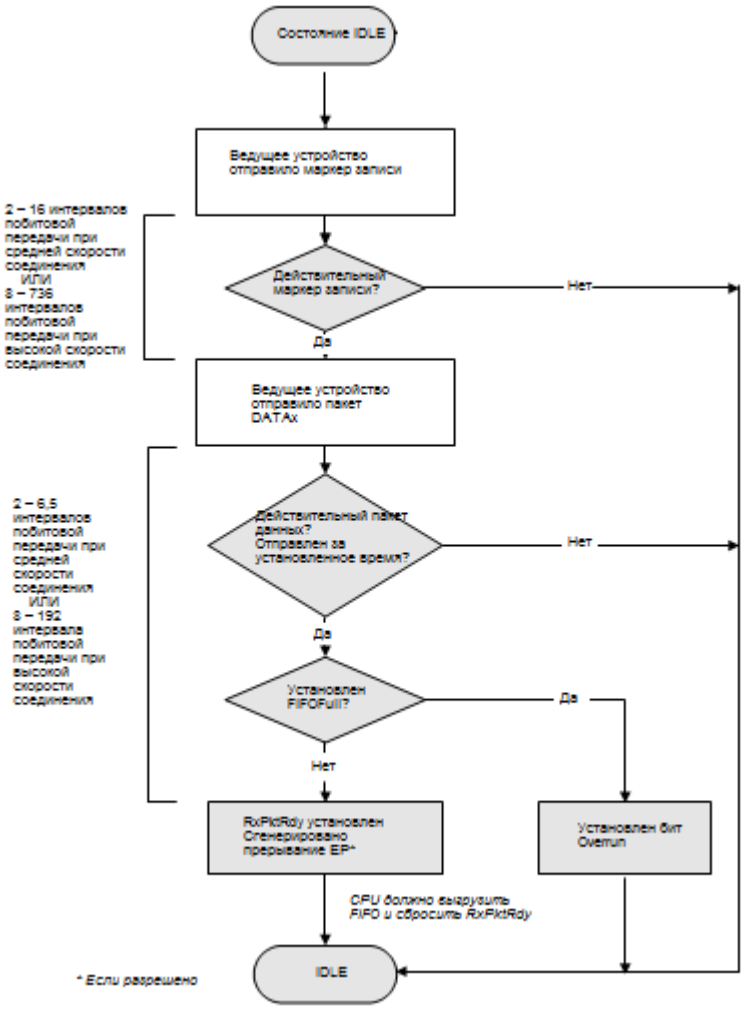


Рисунок 1.282

1.3.12.2.7.20.4 ТРАНЗАКЦИИ С ВЫСОКОЙ ПРОПУСКНОЙ СПОСОБНОСТЬЮ (ИЗОХРОННЫЕ /ПО ПРЕРЫВАНИЮ)

1.3.12.2.7.20.4.1 ТРАНЗАКЦИЯ ЧТЕНИЯ

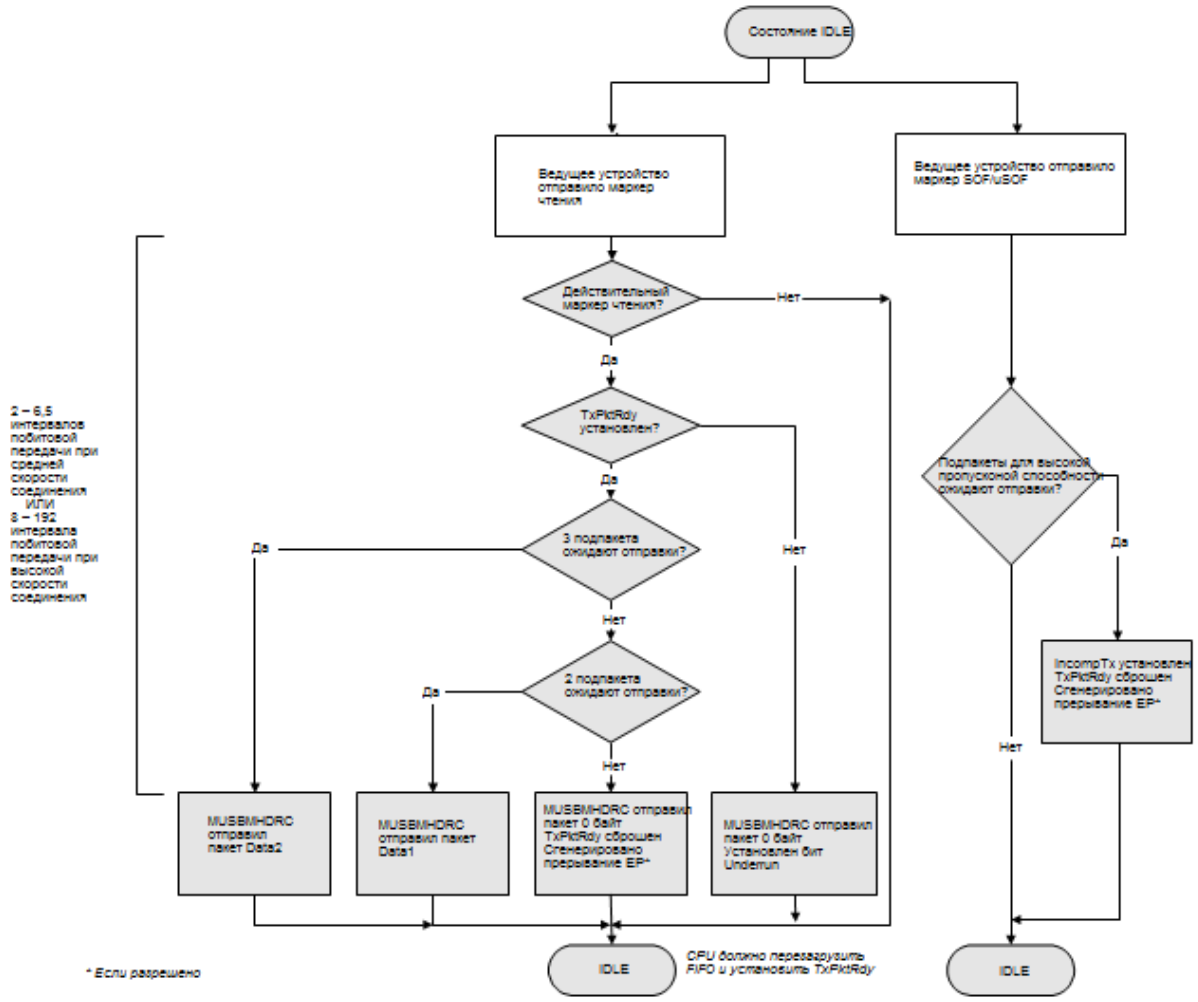


Рисунок 1.283

1.3.12.2.7.20.4.2 ТРАНЗАКЦИЯ ЗАПИСИ

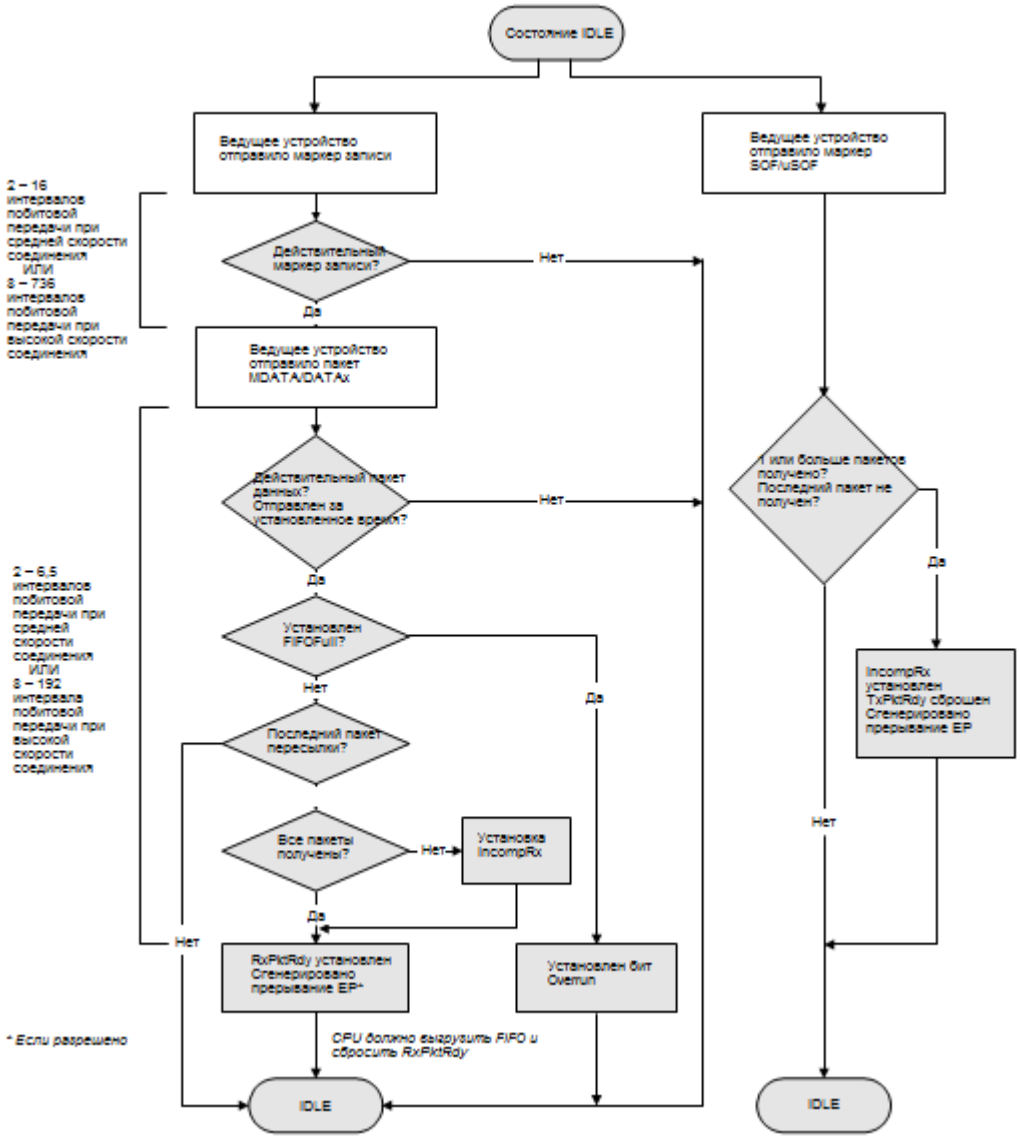


Рисунок 1.284

1.3.12.2.7.21 АЛГОРИТМЫ ТРАНЗАКЦИЙ В РЕЖИМЕ ВЕДУЩЕГО УСТРОЙСТВА

1.3.12.2.7.21.1 УПРАВЛЯЮЩИЕ ТРАНЗАКЦИИ

1.3.12.2.7.21.1.1 ФАЗА SET UP

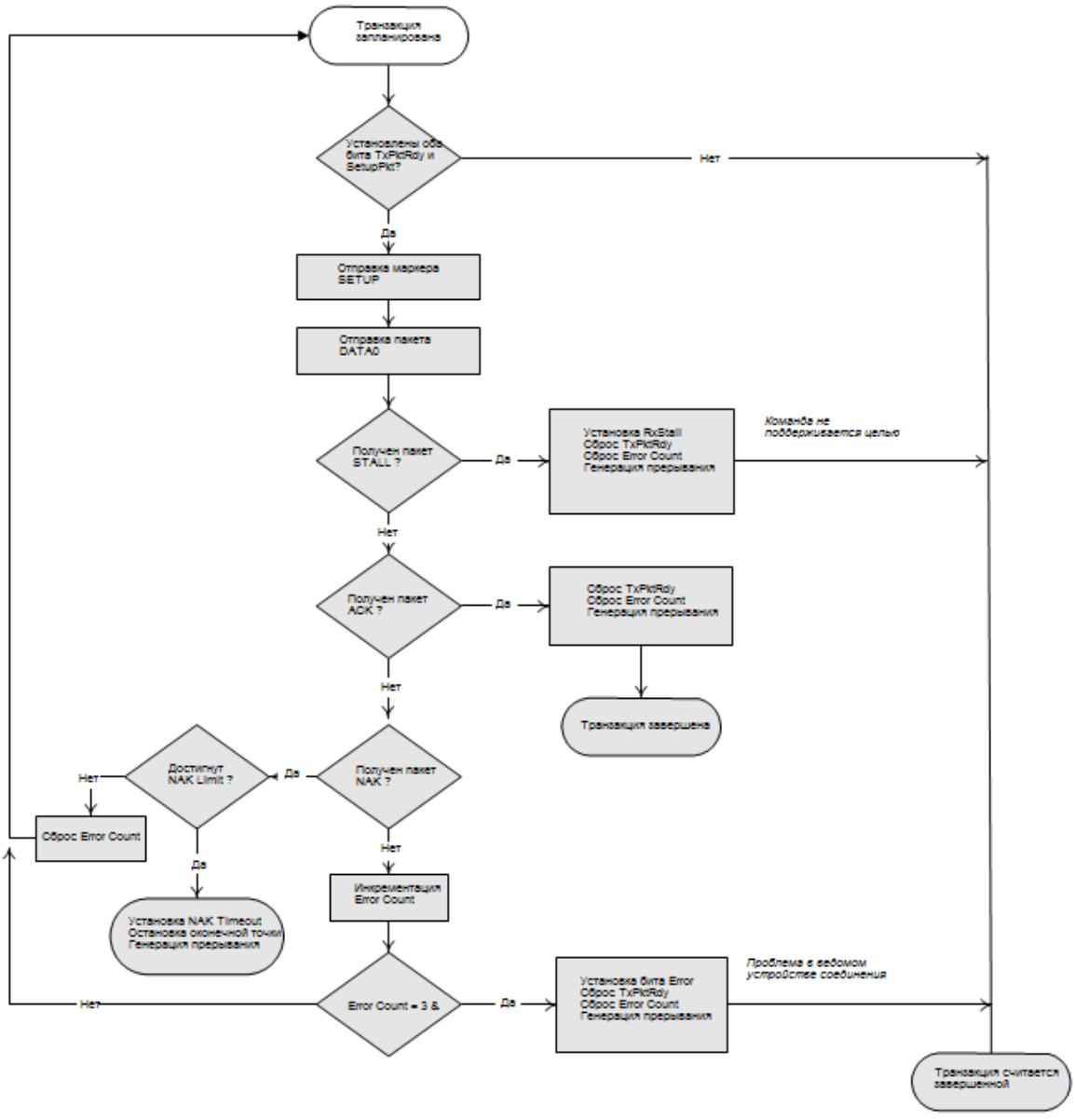


Рисунок 1.285

1.3.12.2.7.21.1.2 ФАЗА ПРИЕМА ДАННЫХ

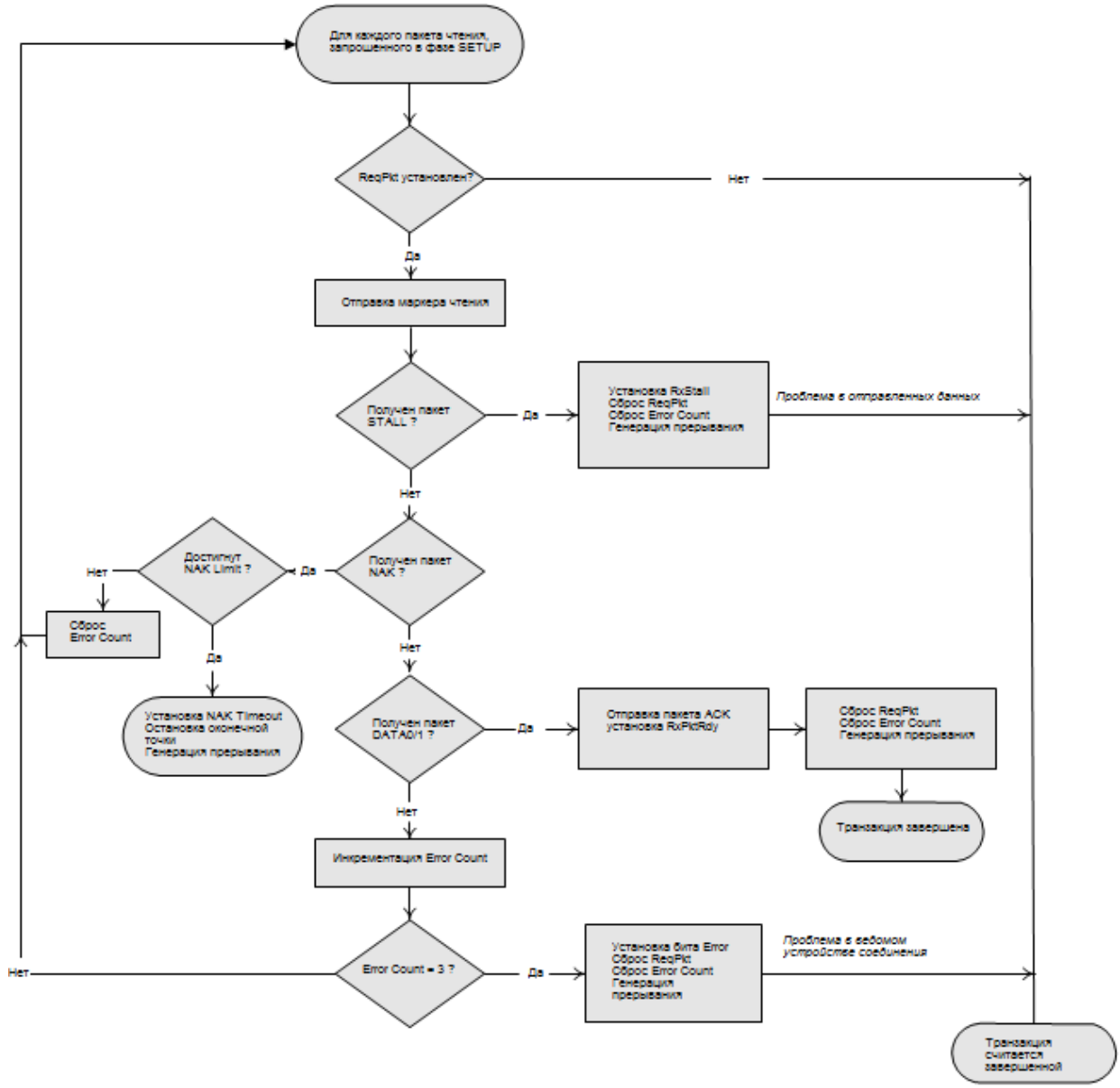


Рисунок 1.286

1.3.12.2.7.21.1.3 ПОСЛЕ ФАЗЫ S T A T U S

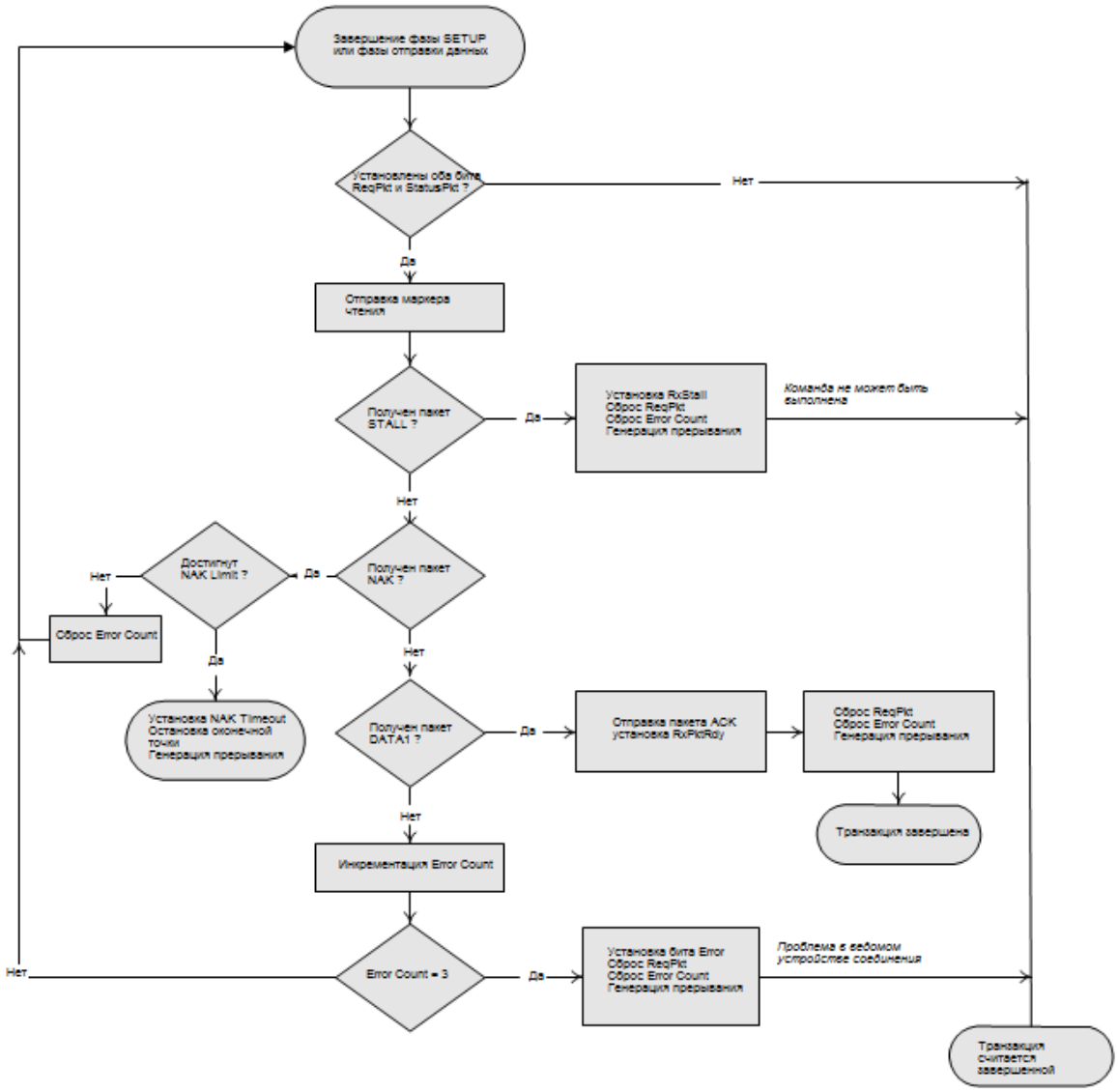


Рисунок 1.287

1.3.12.2.7.21.1.4 ФАЗА ОТПРАВКИ ДАННЫХ

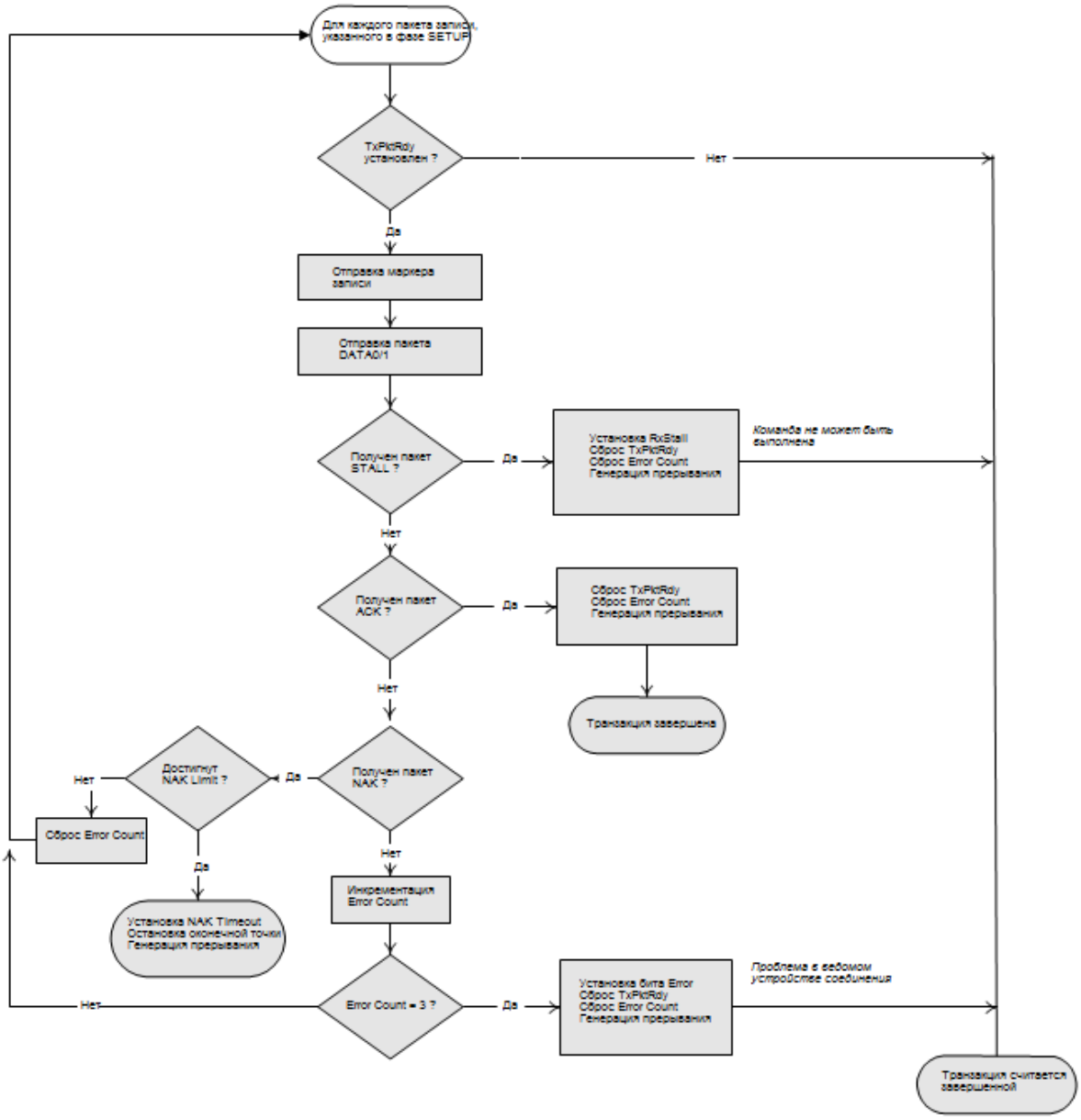


Рисунок 1.288

1.3.12.2.7.21.1.5 ПОСЛЕ ФАЗЫ S T A T U S

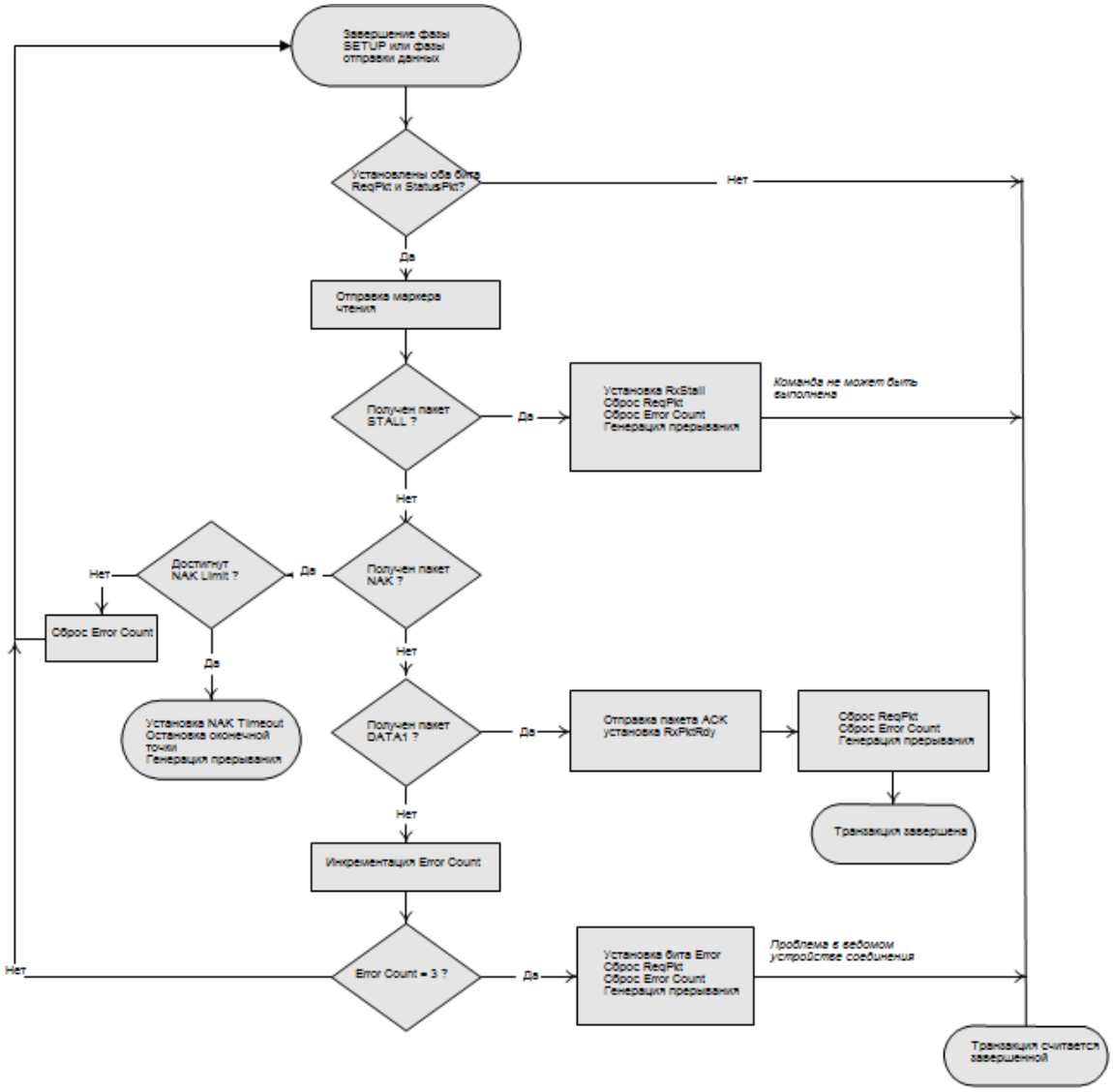


Рисунок 1.289

1.3.12.2.7.21.2 ПОТОЧНЫЕ ТРАНЗАКЦИИ / ТРАНЗАКЦИИ ПО ПРЕРЫВАНИЮ С НИЗКОЙ ПРОПУСКНОЙ СПОСОБНОСТЬЮ

1.3.12.2.7.21.2.1 ТРАНЗАКЦИЯ ЧТЕНИЯ

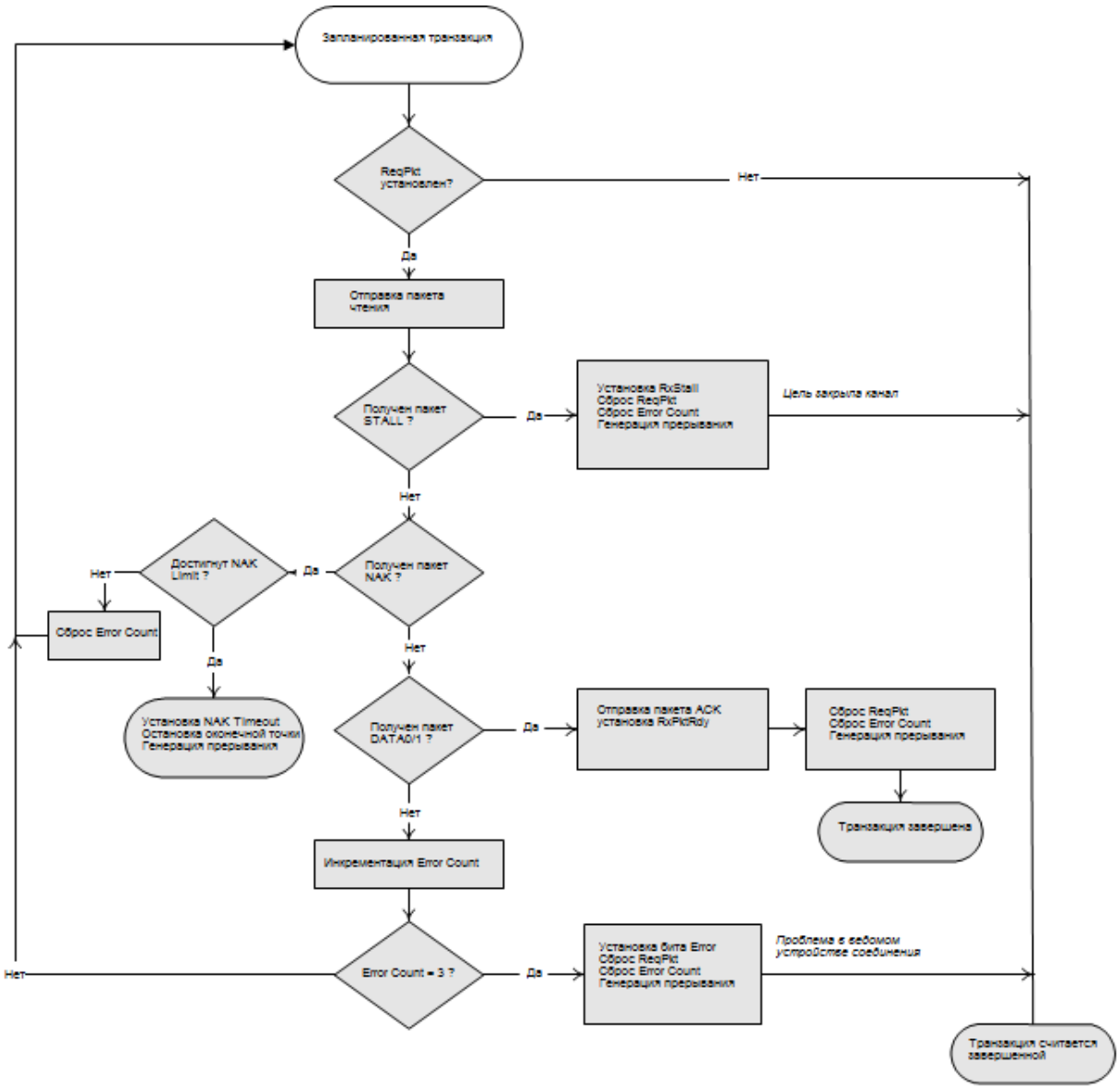


Рисунок 1.290

1.3.12.2.7.21.2.2 ТРАНЗАКЦИЯ ЗАПИСИ

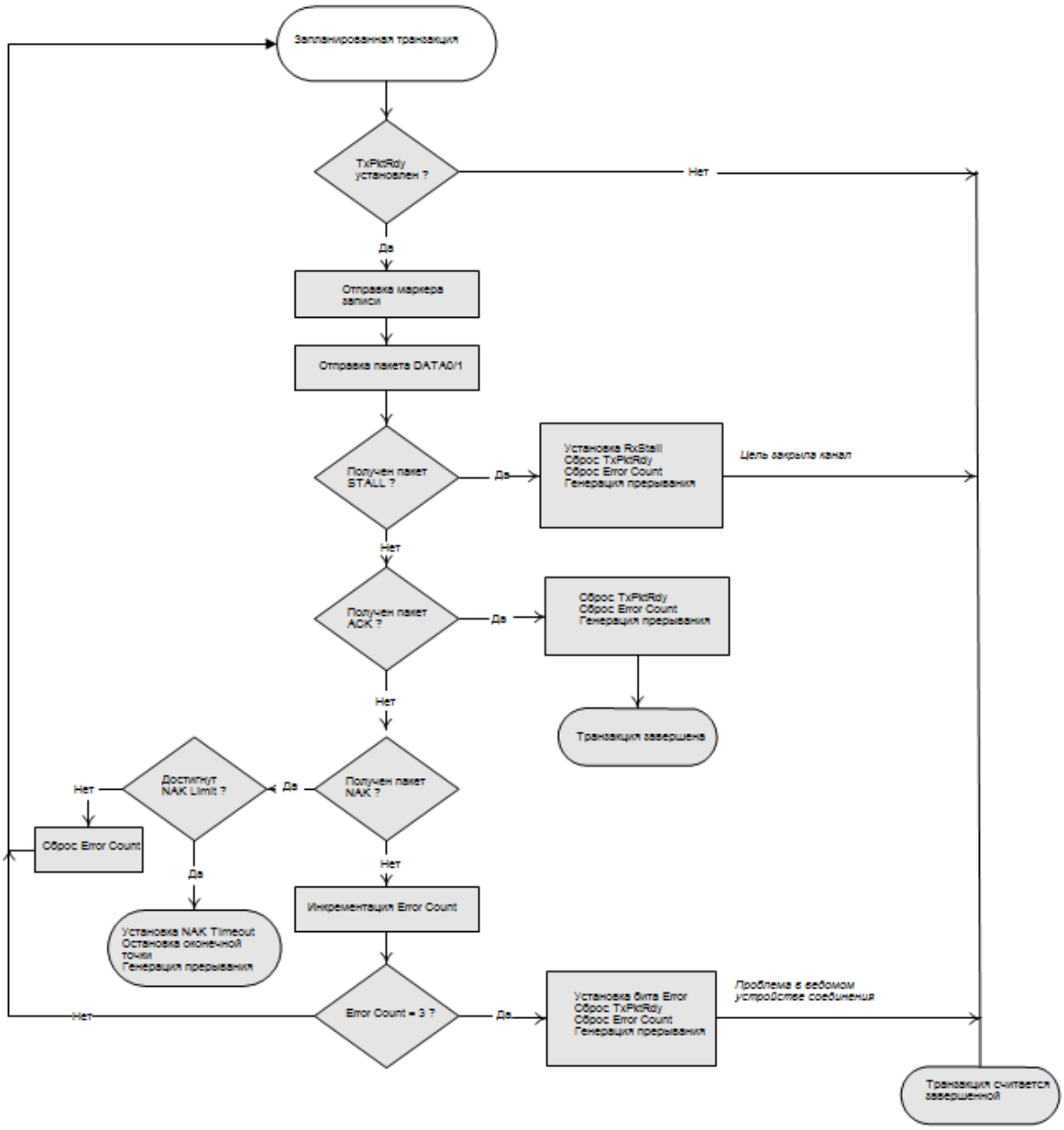


Рисунок 1.291

27 .3 . ИЗОХРОННЫЕ ТРАНЗАКЦИИ ПРИ СРЕДНЕЙ СКОРОСТИ СОЕДИНЕНИЯ / НИЗКОЙ ПРОПУСКНОЙ СПОСОБНОСТИ

1.3.12.2.7.21.2.3 ТРАНЗАКЦИЯ ЧТЕНИЯ

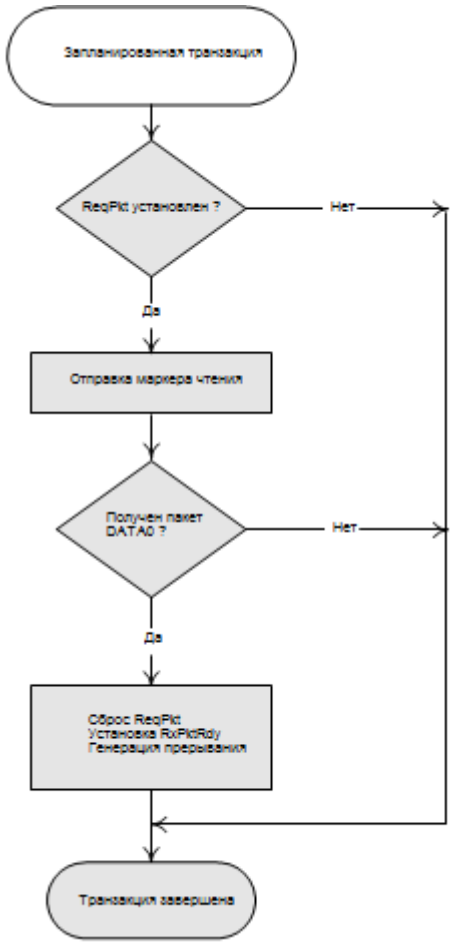


Рисунок 1.292

1.3.12.2.7.21.2.4 ТРАНЗАКЦИЯ ЗАПИСИ

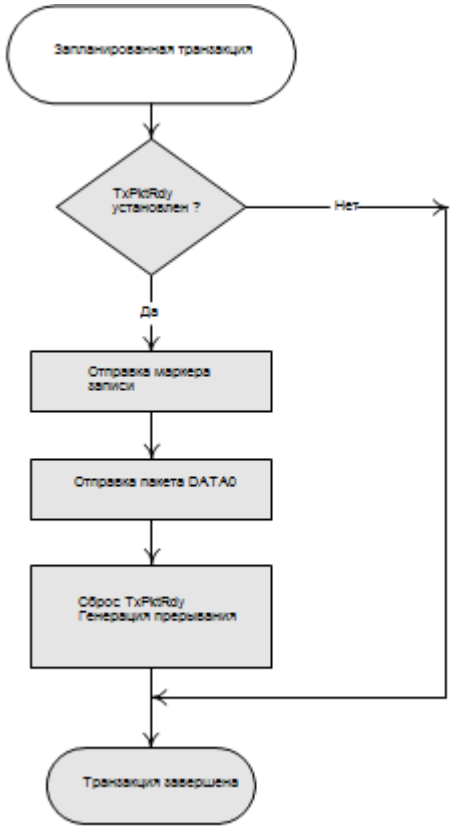


Рисунок 1.293

1.3.12.2.7.21.3 ТРАНЗАКЦИИ С ВЫСОКОЙ ПРОПУСКНОЙ СПОСОБНОСТЬЮ (ИЗОХРОННЫЕ / ПО ПРЕРЫВАНИЮ)

1.3.12.2.7.21.3.1 ВХОДНАЯ ТРАНЗАКЦИЯ

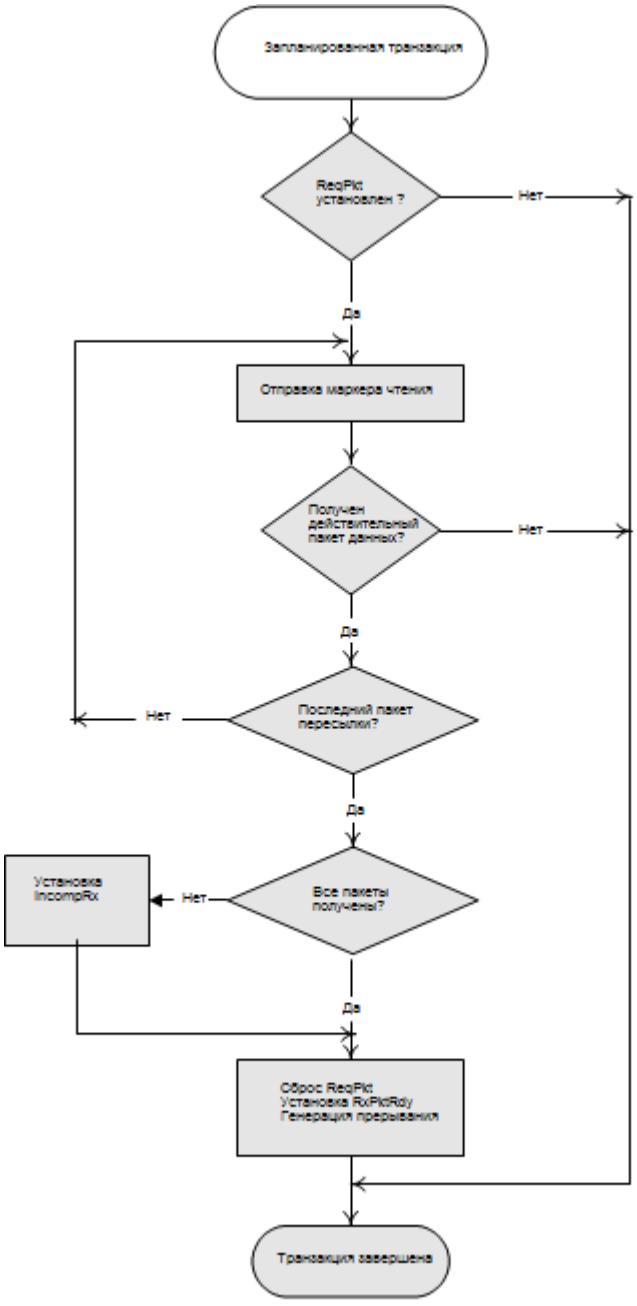


Рисунок 1.294

1.3.12.2.7.21.3.2 ТРАНЗАКЦИЯ ЗАПИСИ

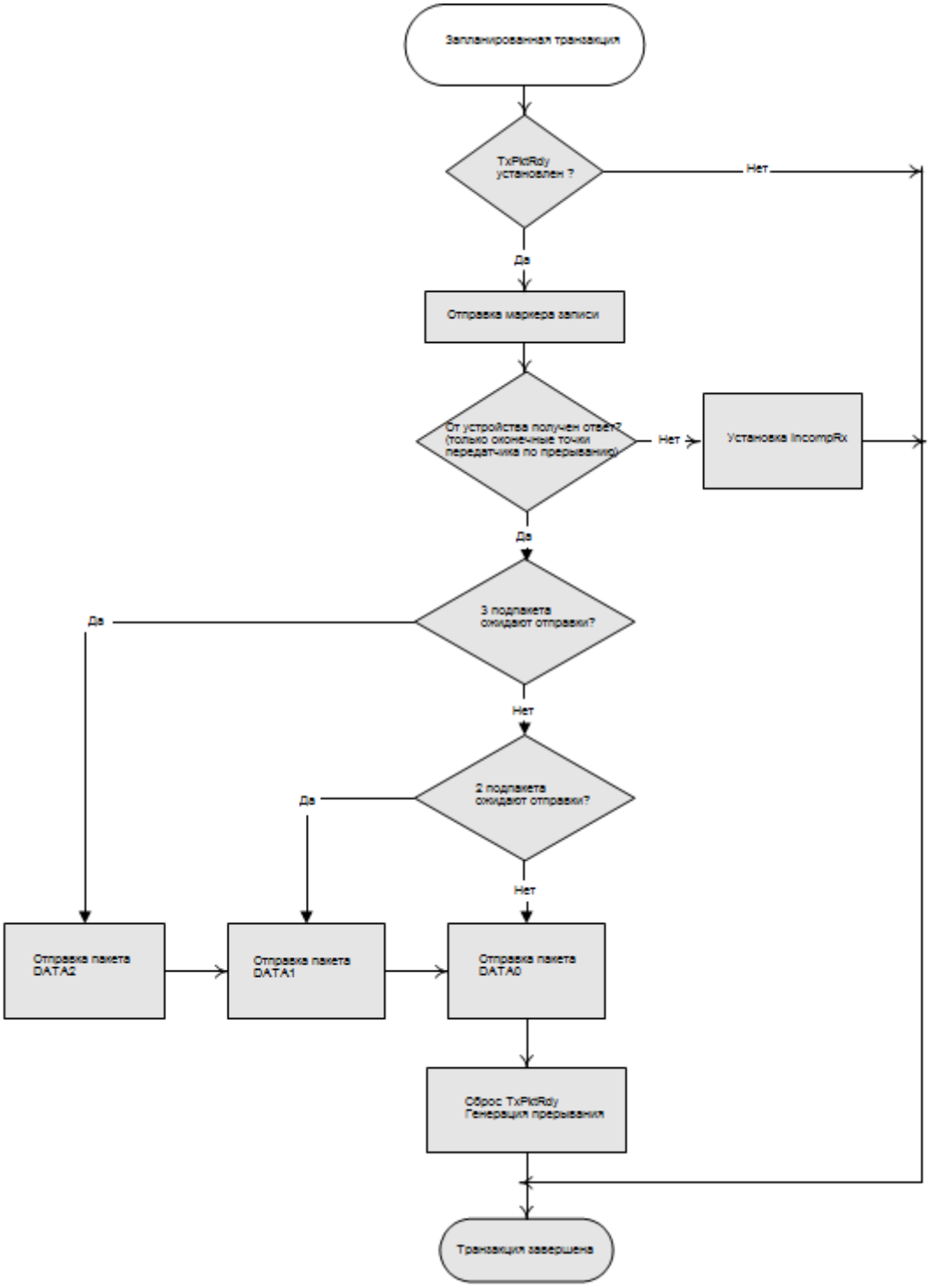


Рисунок 1.295

1.3.12.2.7.21.4 ОПЕРАЦИИ DMA (С ВСТРОЕННЫМ КОНТРОЛЛЕРОМ DMA)

1.3.12.2.7.21.4.1 ОДИН ПАКЕТ ПЕРЕДАТЧИКА

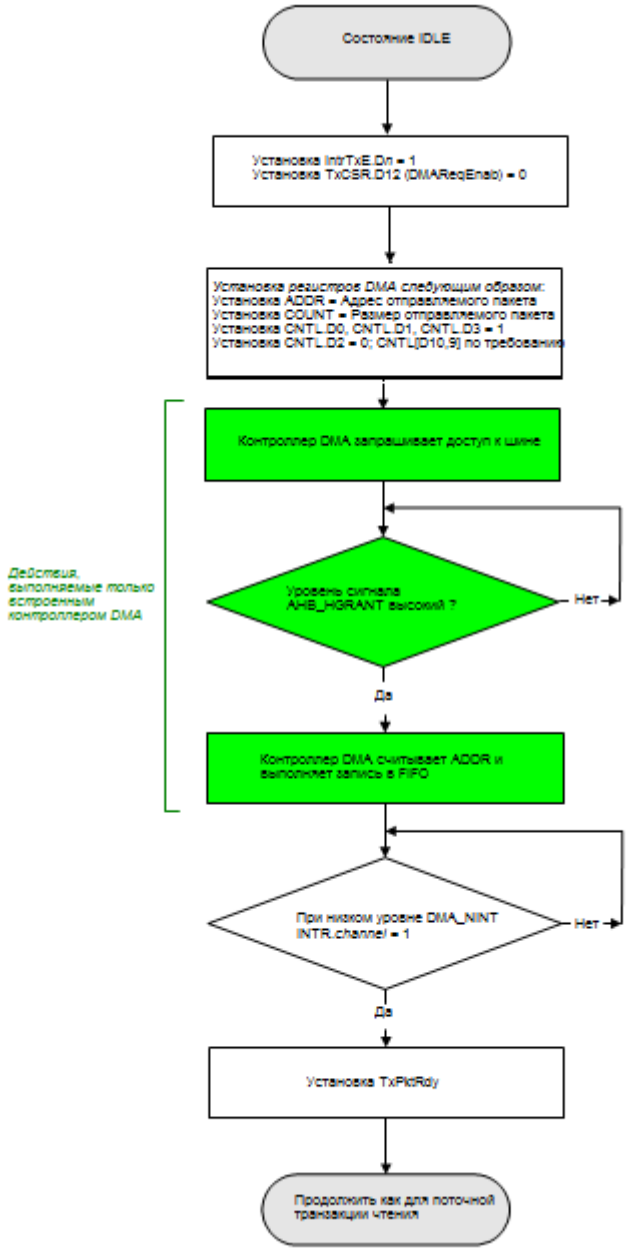


Рисунок 1.296

1.3.12.2.7.21.4.2 ОДИН ПАКЕТ ПРИЕМНИКА

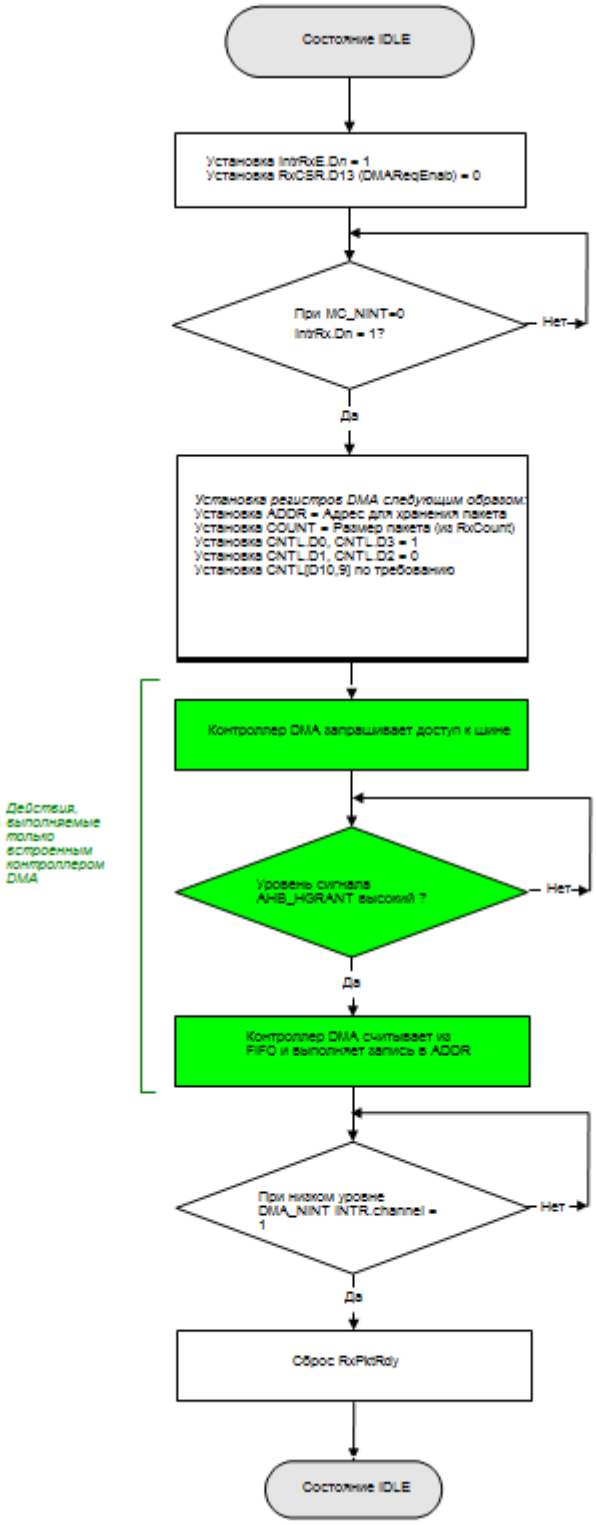


Рисунок 1.297

1.3.12.2.7.21.4.3 ГРУППА ПАКЕТОВ ПЕРЕДАТЧИКА

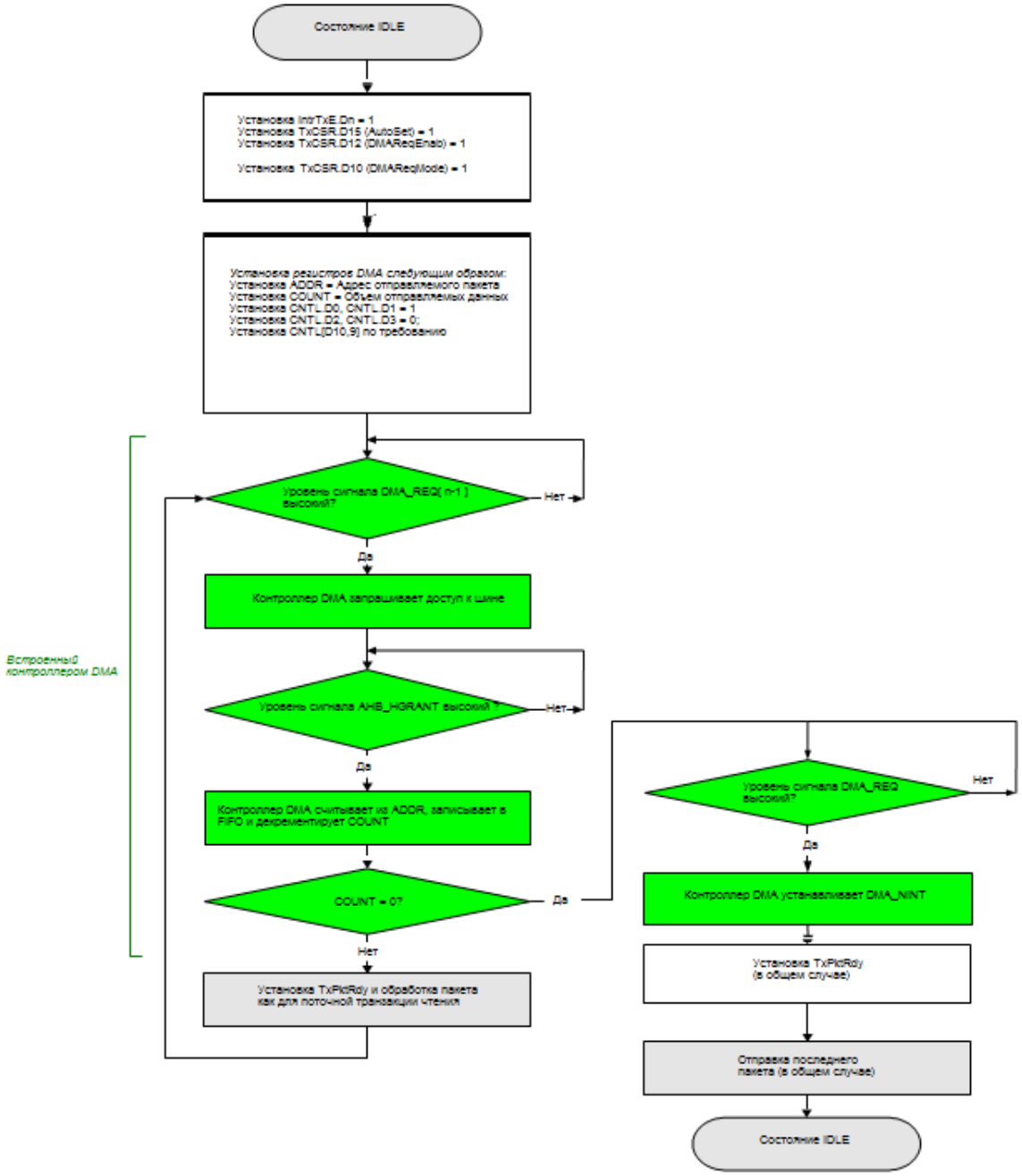


Рисунок 1.298

1.3.12.2.7.21.4.4 ГРУППА ПАКЕТОВ ПРИЕМНИКА

Если размер блока данных известен:

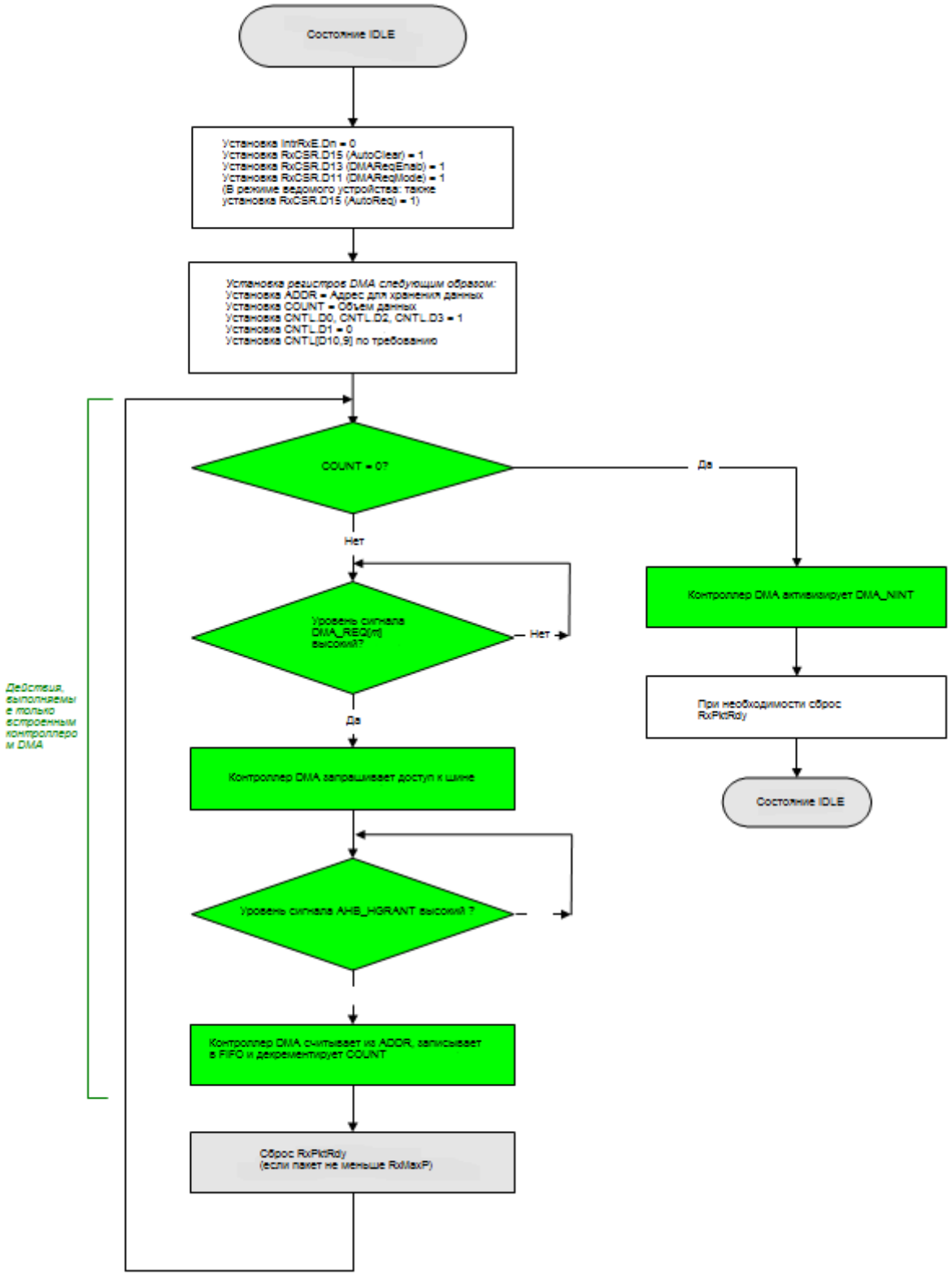


Рисунок 1.299

Если размер блока данных неизвестен:

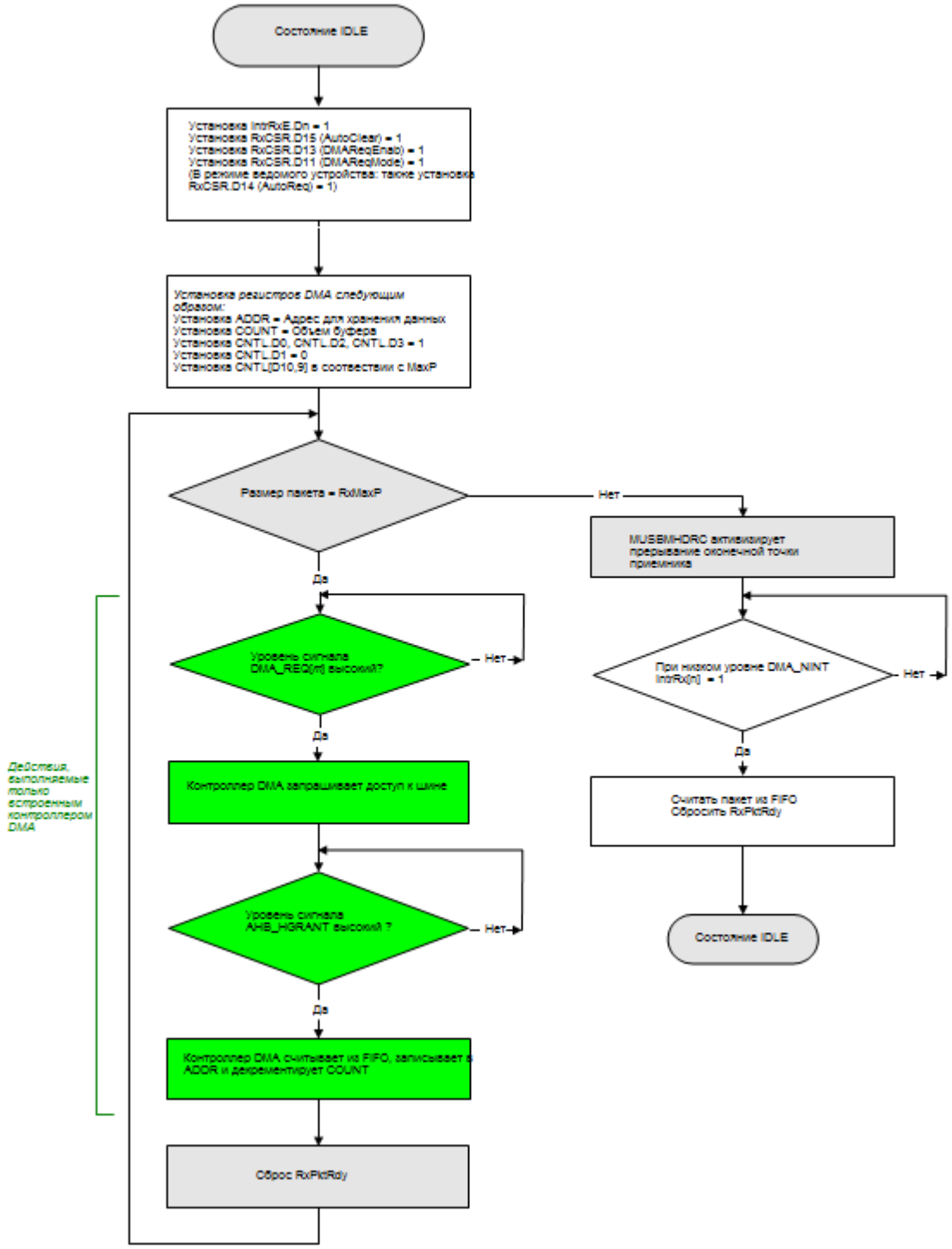


Рисунок 1.300

1.3.12.2.7.22 ТЕСТОВЫЕ РЕЖИМЫ

MUSBMHDRC поддерживает четыре тестовых режима USB 2.0, установленных для высокоскоростных функций. Он также поддерживает дополнительный тестовый режим ‘доступа к FIFO’, который можно использовать для проверки работы интерфейса CPU и блока памяти.

Переход в тестовый режим осуществляется с помощью записи в регистр TestMode (address 0Fh). Тестовый режим обычно запрашивается ведущим устройством, которое отправляет запрос SET_FEATURE оконечной точке 0. Получив запрос, программа должна подождать завершения пересылки оконечной точки 0 (когда она получает прерывание оконечной точки 0, указывающее, что фаза статуса окончилась), а затем выполнить запись в регистр TestMode.

Примечание: При обычной работе эти тестовые режимы не нужны.

1.3.12.2.7.22.1 TEST_SE0_ N A K

Для перехода в тестовый режим Test_SE0_NAK программа должна установить бит Test_SE0_NAK, записав значение 8’h01 в регистр TestMode. MUSBMHDRC перейдет в режим, в котором будет отвечать на любой действительный маркер чтения пакетом NAK.

1.3.12.2.7.22.2 TEST_J

Для перехода в тестовый режим Test_J программа должна установить бит Test_J, записав значение 8’h02 в регистр TestMode. MUSBMHDRC перейдет в режим, в котором будет непрерывно передавать на шину состояние J.

1.3.12.2.7.22.3 TEST_K

Для перехода в тестовый режим Test_K программа должна установить бит Test_K, записав значение 8’h04 в регистр TestMode. MUSBMHDRC перейдет в режим, в котором будет непрерывно передавать на шину состояние K.

1.3.12.2.7.22.4 TEST_P A C KET

Для выполнения теста Test_Packet программа должна:

Начать сеанс (если ядро используется в режиме ведущего устройства).
Записать стандартный тестовый пакет (представленный ниже) в FIFO оконечной точки.
Записать значение 8’h08 в регистр TestMode, чтобы перейти в режим Test_Packet.
Установить бит TxPktRdy в регистре CSR0 (D1).

Тестовый пакет для загрузки в FIFO имеет следующий вид (все байты в шестнадцатеричной системе). Тестовый пакет необходимо загрузить один раз; MUSBMHDRC будет непрерывно отправлять этот пакет без участия программы.

00	00	00	00	00	00	00	00
00	AA	AA	AA	AA	AA	AA	AA
AA	EE	EE	EE	EE	EE	EE	EE
EE	FE	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	7F	BF	DF
EF	F7	FB	FD	FC	7E	BF	DF
EF	F7	FB	FD	7E			

Эта последовательность данных определена в Universal Serial Bus Specification Версия 2.0, раздел 7.1.20. В начале последовательности данных MUSBMHDRC добавит DATA0 PID, а в конце CRC.

1.3.12.2.7.22.5 FIFO_ A CC ESS

Тестовый режим доступа к FIFO позволяет пользователю проверять работу интерфейса CPU и блока памяти, загружая в FIFO оконечной точки 0 пакет размерном до 64 байт, а затем считывая его. Оконечная точка 0 используется потому, что она является двунаправленной и использует одну область памяти для FIFO приемника и передатчика.

Примечание: Для выполнения этого теста ядро не нужно подключать к шине USB. Если оно подключено к шине, то при запуске теста не должно быть сеанса связи.

Процедура выполнения теста:

- Загрузить пакет размером до 64 байт в FIFO передатчика оконечной точки 0 с помощью доступа CPU.
- Установить бит TxPktRdy (CSR0L.D1).
- Записать значение 8’h40 в регистр Testmode.
- Выгрузить пакет из FIFO приемника оконечной точки 0 с помощью доступа CPU.
- Установить бит Set ServicedRxPktRdy (CSR0L.D6).

Запись значения 8’h40 в регистр Testmode вызывает следующую цепочку событий:

- Указатель CPU для оконечной точки 0 (в котором записывается количество байт для передачи) копируется в указатель USB для оконечной точки 0 (в котором записывается количество принятых байт).
- Сбрасывается указатель CPU для оконечной точки 0.
- Сбрасывается бит TxPktRdy (CSR0L.D1).
- Устанавливается бит RxPktRdy (CSR0L.D0).
- Генерируется прерывание оконечной точки 0 (если разрешено).
-

Смысл этих шагов заключается в том, чтобы заставить контроллер оконечной точки 0 действовать так, как будто загруженные в FIFO передатчика пакет был очищен, а тот же самый пакет был принят по шине USB. Данные, загруженные в FIFO передатчика, теперь могут быть считаны из FIFO приемника.

1.3.12.2.7.22.6 FORCE_HOST

Режим Force_Host позволяет пользователю перевести ядро в режим ведущего устройства независимо от того, подключено ли оно к какому-либо ведомому устройству, то есть, игнорируя состояние входного сигнала CID и сигналы LINESTATE и HOSTDISCON. (В этом режиме состояние сигнала HOSTDISCON может быть считано с бита 7 регистра DevCtl.)

Этот режим, который выбирается установкой бита 7 в регистре Testmode, разрешает имплементацию USB Test_Force_Enable (7.1.20). Он также может использоваться для поиска и устранения ошибок PHY в аппаратной части.

При установленном бите Force_Host ядро переходит в режим ведущего устройства, когда установлен бит Session, и остается в режиме ведущего устройства до сброса этого бита, даже если ведомое устройство отсоединяется во время сеанса. Рабочая скорость в этом режиме определяется битами Force_HS и Force_FS регистра Testmode, как описано в разделе **Error! Reference source not found.**.

1.3.12.2.8 Блок физического инетрфейса USB (USBPHY)

Блок физического интерфейса USB реализует физический уровень спецификации USB 2.0. Блок MUSB подключается к блоку USBPHY (см.рисунок 1.301) по интерфейсу UTMI для передачи-приема данных по шине USB.

Кроме этого, блок USBPHY имеет ряд специальных входов, состояние которых задается при помощи регистра usb0_reset_reg (см.раздел 1.3.12.2.8) и которые предназначены для реализации последовательностей управления питанием блока USBPHY.

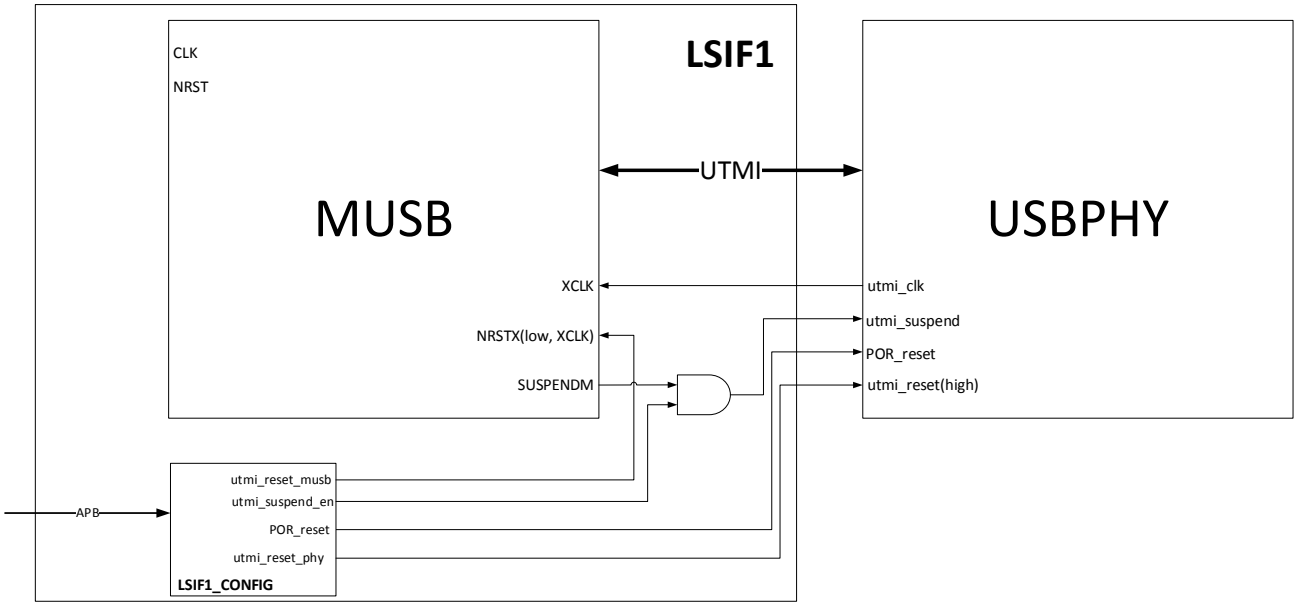


Рисунок 1.301

1.3.12.2.8.1 Последовательность POWER SUPPLY BRING UP

При подаче напряжения рекомендуется выдерживать приведенную на рис.1.302 и описанную в таблице 1.723 последовательность.

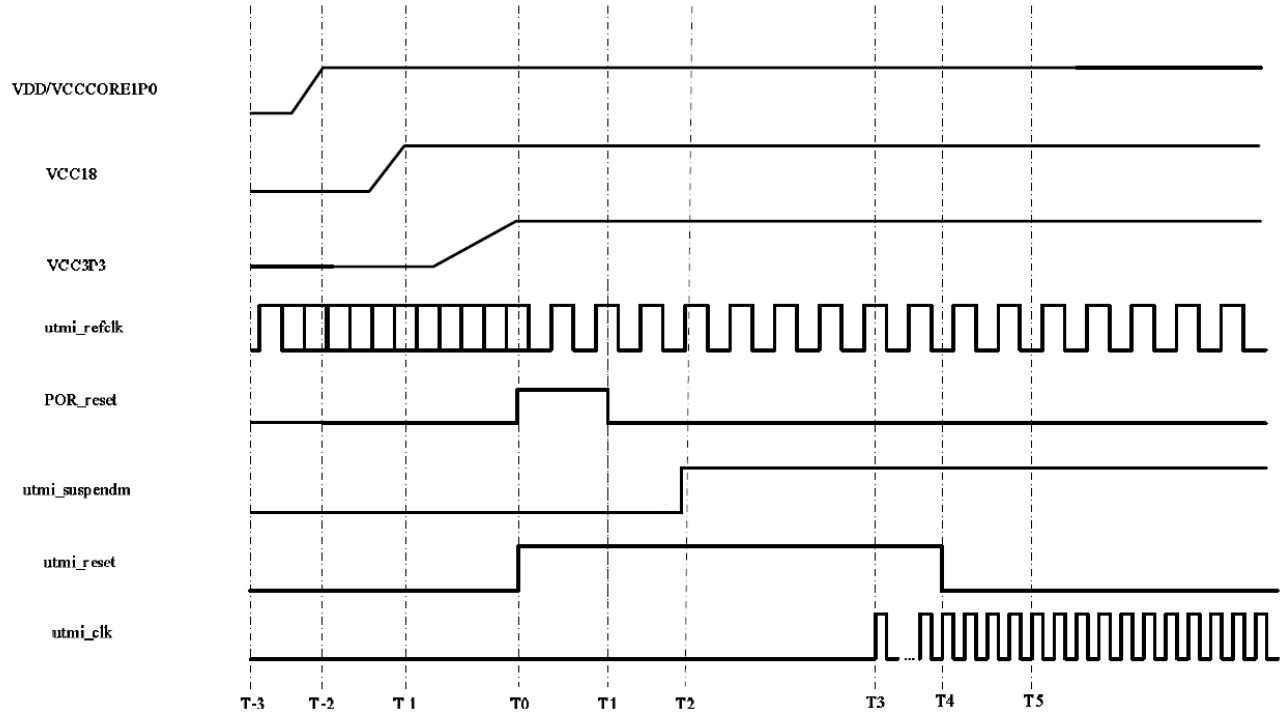


Рисунок 1.302

Таблица 1.723

Состояние	Описание	Длительность до след.состояния
T-3	Контроллер USB должен находиться в состоянии Reset. Входные сигналы POR_reset, utmi_suspendm, utmi_reset блока USBPHY находятся в состоянии 0, т.к. в момент сброса в состоянии 0 находятся все биты регистра usb0_reset_reg, которые задают состояние перечисленных выше сигналов. Подача на напряжения питания на VCCCORE1P0.	Максимум 2 мс
T-2	Выход на заданный уровень напряжения питания на VCCCORE1P0. Подача на VCC18 и выход на заданный уровень напряжения питания	Максимум 2 мс
T-l	Выход на заданный уровень напряжения питания на VCC18. Подача на VCC3P3 и выход на заданный уровень напряжения питания.	Максимум 10 мс
T0	К моменту T0 напряжение питания на блоке USBPHY должно быть стабильно и должен быть подан синхросигнал utmi_refclk (через входной контакт USB_CLK). Сигнал POR_reset устанавливается в состояние 1 при помощи установки в состояние 1 бита POR_reset регистра usb0_reset_reg. Сигнал utmi_reset устанавливается в состояние 1 при помощи установки в состояние 1 бита utmi_reset регистра usb0_reset_reg.	Не менее 10 тактов utmi_refclk
Tl	Сигнал POR_reset устанавливается в состояние 0 при помощи установки в состояние 0 бита POR_reset регистра usb0_reset_reg.	Минимум 10 мкс
T2	Сигнал utmi_suspendm устанавливается в состояние 1 при помощи установки в состояние 1 бита utmi_suspendm_en регистра usb0_reset_reg.	

Продолжение таблицы 1.723

Состояние	Описание	Длительность до след.состояния
T3	К моменту, минимум через 500 мкс, T3 стабилизируется встроенный в USBPHY блок ФАПЧ и на выходе utmi_clk блока USBPHY появляется синхросигнал, тактирующий обмен между контроллером USB и блоком USBPHY по интерфейсу UTMI.	
T4	К моменту T4 синхросигнал utmi_clk стабилизируется. Сигнал utmi_reset устанавливается в состояние 0 при помощи установки в состояние 0 бита.	Минимум - 10 тактов синхросигнала utmi_clk
T5	К моменту T5 блок USBPHY переходит в рабочее состояние. Входной сигнал сброса XRSTX контроллера USB устанавливается в состояние 1 при помощи установки в состояние 1 бита utmi_reset_musb регистра usb0_reset_reg.	Минимум 10 мс

1.3.12.2.8.2 Последовательность SUSPEND SEQUENCE и снятие электропитания

Последовательность SUSPEND SEQUENCE используется для перевода блока USBPHY в режим приостановки. Переведенный в данный режим блок USBPHY перестает генерировать синхросигнал utmi_clk, кроме того, в нем переключются только несколько триггеров по синхросигналу utmi_refclk, в то время как синхросигнал на большинство триггеров не подается.

Режиму приостановки характерно пониженное потребление электропитания блоком USBPHY. Но, при необходимости, после перевода блока в режим приостановки он может быть оключен от электропитания.

Варианты отключения электропитания и требования к ним приведены в табл.1.724.

Таблица 1.724

Напряжение	Состояние	Требования
VDD* VCCCORE1P0 VCCA3P3 VCC18	off on on on	Вход VDD должен быть подключен к «земле».
VDD VCCCORE1P0 VCCA3P3 VCC18	on off off off	Входы VCCCORE1P0/VCC18/VCCA3P3 должны быть подключены к «земле».
VDD VCCCORE1P0 VCCA3P3 VCC18	off on off off	Входы VDD/VCC18/VCCA3P3 должны быть подключены к «земле».
VDD VCCCORE1P0 VCCA3P3 VCC18	on on off off	Входы VCCA3P3/VCC18 должны быть подключены к «земле».

Примечание: * - напряжение VDD – это напряжение питания цифрового ядра микросхемы, подаваемое на блок USBPHY.

Блок USBPHY может быть выведен из режима приостановки при помощи последовательности RESUME SEQUENCE, описанной в разделе 1.3.12.2.8.3, если он не отключался от электропитания, и при помощи последовательности POWER SUPPLY BRING UP, описанной в разделе 1.3.12.2.8.1, если электропитание от него отключалось.

Временная диаграмма перевода блока USBPHY в режим приостановки приведена на рисунке 1.303.

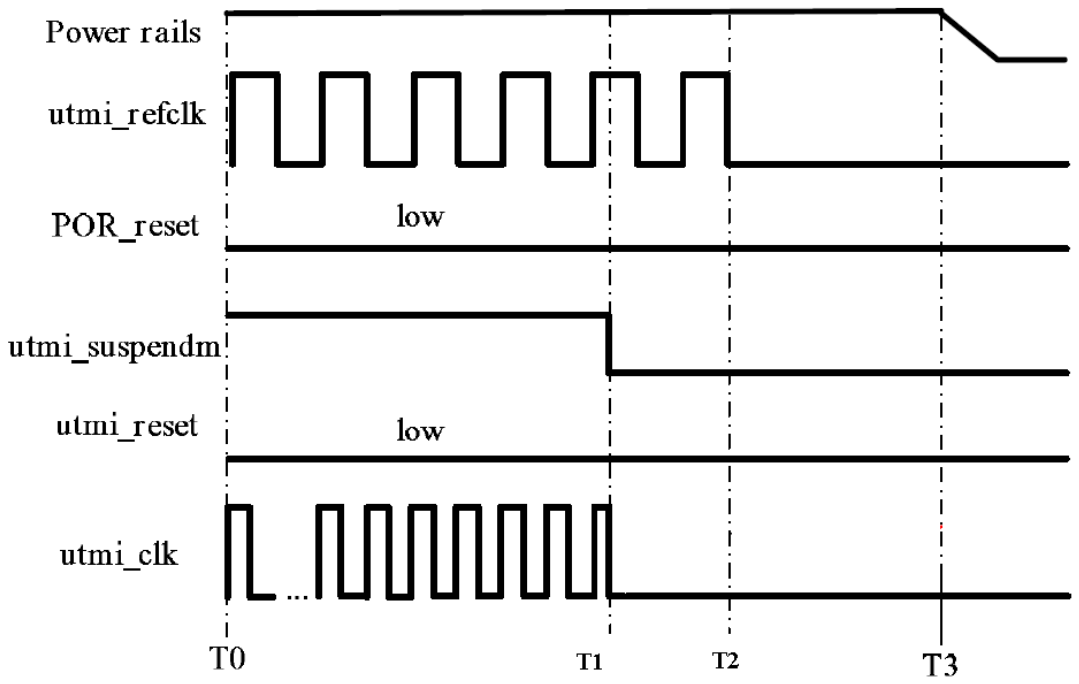


Рисунок 1.303 - Описание перевода блока USBPHY в режим приостановки приведено в табл.Таблица 1..

Таблица 1.725

Состояние	Описание	Длительность до след.состояния
T0	Входные сигналы POR_reset и utmi_reset блока USBPHY должны находится в состоянии 0 (биты POR_reset и utmi_reset_phy регистра usb0_reset_reg соответсвенно). Все сигналы интерфейса UTMI между контроллером USB и блоком USBPHY не должны меняться в этот момент.	-
T1	Сигнал utmi_suspendm устанавливается в состояние 0 при помощи установки в состояние 0 бита utmi_suspendm_en регистра usb0_reset_reg.	Минимум 1 мкс
T2	Блок USBPHY переходит в состояние присотановки. Сигнал utmi_refclk (USB_CLK) может быть снят с блока USBPHY.	Минимум 1 мс
T3	Момент безопасного отключения блока USBPHY от электропитания (опционально).	

1.3.12.2.8.3 Последовательность RESUME SEQUENCE/REMOTE WAKEUP SEQUENCE

Последовательность RESUME SEQUENCE предназначена для вывода блока USBPHY, работающего с контроллером USB , находящимся в состоянии USB-Device, из режима приостановки.

Последовательность REMOTE WAKEUP SEQUENCE предназначена для вывода блока USBPHY, работающего с контроллером USB , находящимся в состоянии USB-Host, из режима приостановки.

Временная диаграмма перевода блока USBPHY в режим приостановки приведена на рисунке 1.304.

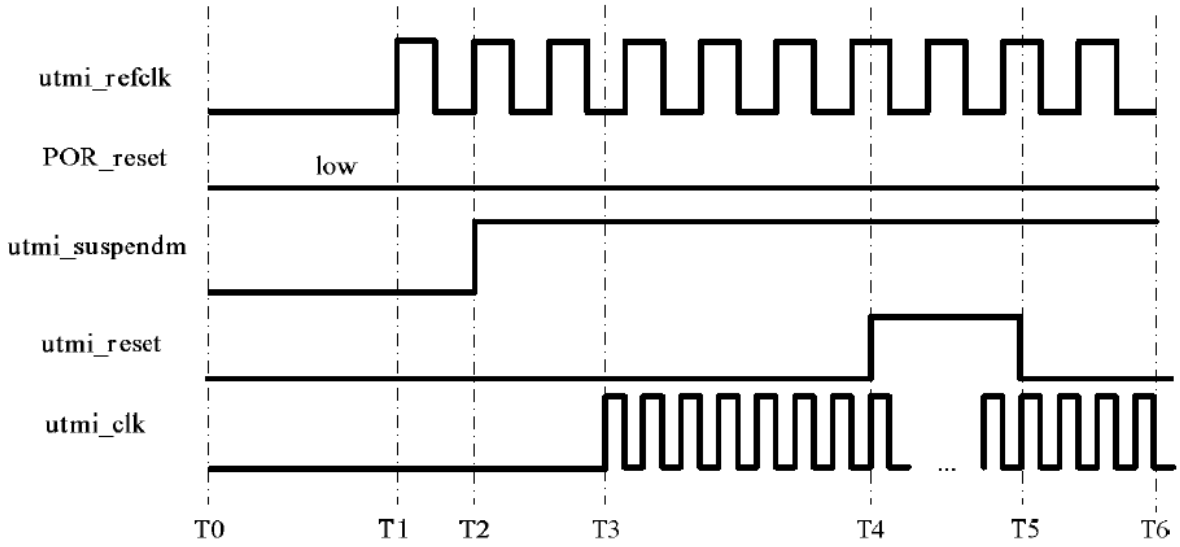


Рисунок 1.304 - Описание вывода блока USBPHY из режима приостановки приведено в таблице 1.726.

Таблица 1.726

Состояние	Описание	Длительность до след.состояния
T0	Контроллер USB должен находиться в состоянии Reset. Блок USBPHY обнаруживает последовательность Chirp k, генерируемую подключенной к нему по шине USB ответному USB-Device (или USB-Host).	-
T1	Момент стабилизации синхросигнала utmi_refclk.	Сразу после стабилизации синхросигнала utmi_refclk
T2	Сигнал utmi_suspendm устанавливается в состояние 1 при помощи установки в состояние 1 бита utmi_suspendm_en регистра usb0_reset_reg.	Минимум 500 мкс
T3	Момент начала генерации блоком USBPHY синхросигнала utmi_clk	Минимум 10 тактов utmi_clk
T4	Сигнал utmi_reset устанавливается в состояние 1 при помощи установки в состояние 1 бита utmi_reset регистра usb0_reset_reg.	Минимум 10 тактов utmi_clk
T5	Сигнал utmi_reset устанавливается в состояние 0 при помощи установки в состояние 0 бита utmi_reset регистра usb0_reset_reg. Блок USBPHY переходит в нормальный режим функционированияю	Минимум 1 мкс
T6	Контроллер USB переходит в нормальный режим функционирования.	

1.3.13 Подсистема общего управления микросхемы

1.3.13.1 Системный контроллер SCTL

Системный контроллер микросхемы представляет собой набор программно-доступных регистров, которые обеспечивают функции управления и статуса, которые не реализованны в других устройствах микросхемы.

1.3.13.1.1 Список регистров

Таблица 1.727 – Список регистров блока SCTL

Адрес	Название	Описание
0x000	BOOT_CFG	Состояние внешних выводов SYS_BOOT[7..0]
0x004	PLL_STATE	Статус блоков PLL
0x008	SYS_CONF0	Регистр конфигурации системы
0x00C		зарезервировано
0x010	RST_MON	регистр отслеживания сбросов
0x014	RST_CTRL	программный сброс системы/микросхемы
0x018	PPC_SLP_CTRL	управление режимом сна процессорных ядер PPC470S
0x01C	SRAM_NOR_CTRL	Управление интерфейсом SRAM_NOR
0x020	BISR_IM0	Управление механизмом самовосстановления памяти IM0
0x024	BISR_L2C_0	Управление механизмом самовосстановления памяти L2C_0
0x028	BISR_L2C_1	Управление механизмом самовосстановления памяти L2C_1
0x02C	BISR_DSP0_IM1_0	управление механизмом самовосстановления памяти DSP0_IM1_0
0x030	BISR_DSP0_IM1_1	управление механизмом самовосстановления памяти DSP0_IM1_1
0x034	BISR_DSP0_L2C	управление механизмом самовосстановления памяти DSP0_L2C
0x038	BISR_DSP1_IM1_0	управление механизмом самовосстановления памяти DSP1_IM1_0
0x03C	BISR_DSP1_IM1_1	управление механизмом самовосстановления памяти DSP1_IM1_1
0x040	BISR_DSP1_L2C	управление механизмом самовосстановления памяти DSP1_L2C
…		зарезервировано для других регистров управления функцией redundancy
0x060		зарезервировано для других регистров управления буферами в/в
0x064		зарезервировано для других регистров управления буферами в/в
0x068	IFSYS_IO_CONF2	Конфигурация GPIO буферов интерфейсных подсистем
0x06C	IFSYS_IO_CONF3	Конфигурация GPIO буферов интерфейсных подсистем
0x070	IFSYS_IO_CONF4	Конфигурация GPIO буферов интерфейсных подсистем
0x074	IFSYS_IO_CONF5	Конфигурация GPIO буферов интерфейсных подсистем
0x078	IFSYS_IO_CONF6	Конфигурация GPIO буферов интерфейсных подсистем
0x07C	IFSYS_IO_CONF7	Конфигурация GPIO буферов интерфейсных подсистем
0x080	IFSYS_IO_CONF8	Конфигурация GPIO буферов интерфейсных подсистем
0x084	IFSYS_IO_CONF9	Конфигурация GPIO буферов интерфейсных подсистем
0x088	IFSYS_IO_CONF10	Конфигурация GPIO буферов интерфейсных подсистем
0x08C	IFSYS_IO_CONF11	Конфигурация GPIO буферов интерфейсных подсистем
0x090	IFSYS_IO_CONF12	Конфигурация GPIO буферов интерфейсных подсистем
0x094	IFSYS_IO_CONF13	Конфигурация GPIO буферов интерфейсных подсистем
0x098	IFSYS_IO_CONF14	Конфигурация GPIO буферов интерфейсных подсистем
0x09C		зарезервировано для других регистров управления буферами в/в
0x0A0		зарезервировано для других регистров управления буферами в/в
0x0A4		зарезервировано для других регистров управления буферами в/в
0x0A8		зарезервировано для других регистров управления буферами в/в
0x0AC	IFSYS_IO_CONF19	Конфигурация GMII буферов интерфейсных подсистем
0x0B0	IFSYS_IO_CONF20	Конфигурация GMII буферов интерфейсных подсистем
0x0B4	IFSYS_IO_CONF21	Конфигурация GMII буферов интерфейсных подсистем
0x0B8	IFSYS_IO_CONF22	Конфигурация GMII буферов интерфейсных подсистем

Продолжение таблицы 1.727

Адрес	Название	Описание
0x0BC		зарезервировано для других регистров управления буферами в/в
0x0C0		зарезервировано для других регистров управления буферами в/в
0x0C4		зарезервировано для других регистров управления буферами в/в
0x0C8		зарезервировано для других регистров управления буферами в/в
0x0CC		зарезервировано для других регистров управления буферами в/в
0x0D0		зарезервировано для других регистров управления буферами в/в
0x0D4		зарезервировано для других регистров управления буферами в/в
0x0D8		зарезервировано для других регистров управления буферами в/в
0x0DC		зарезервировано для других регистров управления буферами в/в
0x0E0		зарезервировано для других регистров управления буферами в/в
0x0E4		зарезервировано для других регистров управления буферами в/в
0x0E8		зарезервировано для других регистров управления буферами в/в
0x0EC	IFSYS_IO_CONF35	Конфигурация I2C буферов
0x0F0		зарезервировано для других регистров управления буферами в/в
0x0F4		зарезервировано для других регистров управления буферами в/в
0x0F8		зарезервировано для других регистров управления буферами в/в
0x0FC		зарезервировано для других регистров управления буферами в/в
0x100	MLT_IO_CONF0	Конфигурация GPIO буферов мультимедийных подсистем
0x104	MLT_IO_CONF1	Конфигурация GPIO буферов мультимедийных подсистем
0x108	MLT_IO_CONF2	Конфигурация GPIO буферов мультимедийных подсистем
0x10C	MLT_IO_CONF3	Конфигурация GPIO буферов мультимедийных подсистем
0x110	MLT_IO_CONF4	Конфигурация GPIO буферов мультимедийных подсистем
0x114	MLT_IO_CONF5	Конфигурация GPIO буферов мультимедийных подсистем
0x118	MLT_IO_CONF6	Конфигурация GPIO буферов мультимедийных подсистем
0x11C	MLT_IO_CONF7	Конфигурация GPIO буферов мультимедийных подсистем
0x120	MLT_IO_CONF8	Конфигурация GPIO буферов мультимедийных подсистем
0x124	MLT_IO_CONF9	Конфигурация GPIO буферов мультимедийных подсистем
0x128	MLT_IO_CONF10	Конфигурация GPIO буферов мультимедийных подсистем
0x12C	MLT_IO_CONF11	Конфигурация GPIO буферов мультимедийных подсистем
0x130	MLT_IO_CONF12	Конфигурация GPIO буферов мультимедийных подсистем
0x134	MLT_IO_CONF13	Конфигурация GPIO буферов мультимедийных подсистем
0x138	MLT_IO_CONF14	Конфигурация GPIO буферов мультимедийных подсистем
0x13C	MLT_IO_CONF15	Конфигурация GPIO буферов мультимедийных подсистем
0x140	MLT_IO_CONF16	Конфигурация GPIO буферов мультимедийных подсистем
0x144	MLT_IO_CONF17	Конфигурация GPIO буферов мультимедийных подсистем
0x148	MLT_IO_CONF18	Конфигурация GPIO буферов мультимедийных подсистем
0x14C	MLT_IO_CONF19	Конфигурация GPIO буферов мультимедийных подсистем
0x150	MLT_IO_CONF20	Конфигурация GPIO буферов мультимедийных подсистем
0x154	MLT_IO_CONF21	Конфигурация GPIO буферов мультимедийных подсистем
0x158	MLT_IO_CONF22	Конфигурация GPIO буферов мультимедийных подсистем
0x15C	MLT_IO_CONF23	Конфигурация GPIO буферов мультимедийных подсистем
0x160	MLT_IO_CONF24	Конфигурация GPIO буферов мультимедийных подсистем
0x164	MLT_IO_CONF25	Конфигурация GPIO буферов мультимедийных подсистем
0x168	MLT_IO_CONF26	Конфигурация GPIO буферов мультимедийных подсистем
0x16C	MLT_IO_CONF27	Конфигурация GPIO буферов мультимедийных подсистем
0x170	MLT_IO_CONF28	Конфигурация GPIO буферов мультимедийных подсистем
0x174	MLT_IO_CONF29	Конфигурация GPIO буферов мультимедийных подсистем
0x178	MLT_IO_CONF30	Конфигурация GPIO буферов мультимедийных подсистем
0x17C	MLT_IO_CONF31	Конфигурация GPIO буферов мультимедийных подсистем
0x180	MLT_IO_CONF32	Конфигурация GPIO буферов мультимедийных подсистем
0x184	MLT_IO_CONF33	Конфигурация GPIO буферов мультимедийных подсистем
…		зарезервировано для других регистров управления буферами в/в
0x190	MISC_IO_CONF0	Конфигурация GPIO буферов общесистемного назначения
…		зарезервировано для других регистров управления буферами в/в
0x1E0	USB_BIST_CNTRL	Управление режимом самотестирования блока USB_PHY

Продолжение таблицы 1.727

Адрес	Название	Описание
0x1E4	EM0_BIST_CNTRL	Управление режимом самотестирования блока EM0_DDR3_PHY
0x1E8	EM1_BIST_CNTRL	Управление режимом самотестирования блока EM1_DDR3_PHY
0x1EC		зарезервировано для других регистров самотестирования
0x1F0		зарезервировано для других регистров самотестирования
0x1F4		зарезервировано для других регистров самотестирования
0x1F8		зарезервировано для других регистров самотестирования
0x1FC		зарезервировано для других регистров самотестирования
0x200		
…		
0x240	INTSTA_DSP2CPU	Состояние прерываний от ядер DSP к CPU подсистеме
0x244	INTCLR_DSP2CPU	Сброс прерываний от ядер DSP к CPU подсистеме
0x248	INTREQ_CPU2DSP	Генерация прерываний к ядрам DSP
0x24C		
…		
0x290	MLT_ADR_MSB0	задание старших бит адреса master-устройств multimedia
0x294	MLT_ADR_MSB1	задание старших бит адреса master-устройств multimedia
0x298		
0x29C		

Ниже приведено детальное описание каждого регистра.

Зарезервированные поля читаются как нули, запись в них игнорируется.

Если значение после сброса не указано, то это означает что регистр является монитором и его значение всегда равно текущему значению отслеживаемого им сигнала.

1.3.13.1.2 Общесистемные регистры

Таблица 1.728 - Регистр BOOT_CFG – статус внешних выводов SYS_BOOT

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7]	SYS_BOOT7	-	R/W	Состояние вывода GPIO0_7/SYS_BOOT7
[6]	SYS_BOOT6	-	R/W	Состояние вывода GPIO0_6/SYS_BOOT6
[5]	SYS_BOOT5	-	R/W	Состояние вывода GPIO0_5/SYS_BOOT5
[4]	SYS_BOOT4	-	R	Состояние вывода GPIO0_4/SYS_BOOT4
[3]	SYS_BOOT3	-	R/W	Состояние вывода GPIO0_3/SYS_BOOT3
[2]	SYS_BOOT2	-	R/W	Состояние вывода GPIO0_2/SYS_BOOT2
[1]	SYS_BOOT1	-	R	Состояние вывода GPIO0_1/SYS_BOOT1
[0]	SYS_BOOT0	-	R	Состояние вывода GPIO0_0/SYS_BOOT0

Таблица 1.729 - Регистр PLL_STATE – состояние блоков PLL

Биты	Название	Сброс	Доступ	Описание
[31:6]	-	-	R0	зарезервировано
[5]	PLL_AUDIO_RDY	-	R	Статус PLL_AUDIO 1 – стабильна 0 – нестабильна
[4]	PLL_VIDEO_RDY	-	R	Статус PLL_VIDEO 1 – стабильна 0 – нестабильна
[3]	PLL_PHY_EM1_RDY	-	R	Статус PLL_PHY_EM1 1 - стабильна 0 – нестабильна
[2]	PLL_PHY_EM0_RDY	-	R	Статус PLL_PHY_EM0 1 – стабильна 0 – нестабильна

Продолжение таблицы 1.729

Биты	Название	Сброс	Доступ	Описание
[1]	PLL_DDR_RDY	-	R	Статус PLL_DDR 1 - стабильна 0 - нестабильна
[0]	PLL_CPU_RDY	-	R	Статус PLL_CPU 1 - стабильна 0 - нестабильна

Таблица 1.730 - Регистр SYS_CONF0 – регистр конфигурации системы

Биты	Название	Сброс	Доступ	Описание
[31:14]	-	-	R0	зарезервировано
[13]	SEL25_EXT_MEM	1'b0	R/W	Управление сигналами SEL25 GMII-буферов интерфейса EXT_MEM
[12]	SEL25_MII	1'b0	R/W	Управление сигналами SEL25 GMII-буферов интерфейса MII
[11]	PPC_TMRCLK_SEL	1'b0	R/W	Выбор внешнего источника синхросигнала для таймеров ядер PPC: 0 - синхросигнал с CRG_CPU 1 – внешний синхросигнал с буфера CLK_TMR_MPIC
[10]	USBPHY_REFCLK_SEL	1'b1	R/W	Выбор входной частоты USB PHY 0 - utmi_refclk =12 MHz 1 - utmi_refclk =24 MHz
[9]	MPIC_TMRCLK_SEL	1'b0	R/W	Выбор источника синхросигнала для таймеров MPIC: 0 – системный синхросигнал (частота шины DCR) 1 – внешний синхросигнал с буфера CLK_TMR_MPIC
[8]	-	-	R0	зарезервировано
[7:4]	GRETH_GBIT1_EDCL_ADDR	4'h0	R/W	Поле задает 4 младших разряда IP-адреса контроллера GRETH_GBIT1_EDCL
[3:0]	GRETH2_EDCL_ADDR	4'h0	R/W	Поле задает 4 младших разряда IP-адреса контроллера GRETH2_EDCL

Таблица 1.731 - Регистр SRAM_NOR_CTRL – управление интерфейсом SRAM_NOR

Биты	Название	Сброс	Доступ	Описание
[31:4]	-	-	R0	зарезервировано
[3:1]	mcif_ce_manage[2:0]	0	R/W	Управление CE_MANAGE для контроллера SRAM_NOR MCIF
[0]	sram_nor_mux_ctrl	0	R/W	Выбор контроллера SRAM_NOR: 0 – с интерфейсом AXI (LSIF0) 1 – с интерфейсом MCIF (HLB_CPU)

Таблица 1.732 - Регистр RST_MON – регистр отслеживания сбросов

Биты	Название	Сброс	Доступ	Описание
[11:10]	PPC1_CORERST	2’b00	R	Отслеживание сброса ядра PPC1: 2’b00 – не было сброса ядра после сброса микросхемы/системы 2'b01 – ядро сброшено по своему запросу 2'b10 – ядро сброшено через регистр MPIC 2’b11 – ядро сброшено через регистр CRG
[9:8]	PPC0_CORERST	2’b00	R	Отслеживание сброса ядра PPC0: 2’b00 – не было сброса ядра после сброса микросхемы/системы 2'b01 – ядро сброшено по своему запросу 2'b10 – ядро сброшено через регистр MPIC 2’b11 – ядро сброшено через регистр CRG
[7:6]	-	-	R0	зарезервировано
[5:4]	SYS_LRT	2’b10	R	Отображает тип последнего системного сброса: 2’b00 – резерв. 2'b01 – CHIP_RST 2’b10 – SYS_RST 2’b11 – резерв.
[3]	-	-	R0	зарезервировано
[2:0]	SYS_LRS	3’b001	R	Отображает источник последнего системного сброса: 3'b000 – резерв. 3'b001 – EXT 3'b010 – PPC0 3’b011 – PPC1 3’b100 – SCTL 3’b101 – WD_APB 3’b110 – POR 3'b111 – резерв.

Таблица 1.733 - Регистр RST_CTRL – программный сброс системы/микросхемы

Биты	Название	Сброс	Доступ	Описание
[31:24]	WKEY	8’h01	R0	Блокировка записи. Для записи других полей регистра в поле WKEY нужно записать 8'h57. Запись любого другого значения запрещает запись других полей. Чтение возвращает статус блокировки: 8'h00 – запись разблокирована 8'h01 – запись заблокирована
[23:4]	SYS_RST_DUR	20'h061A8	R/W	Длительность системного сброса, вызываемого программным запросом, в периодах синхросигнала CLK_CPU
[3:2]	-	-	R0	зарезервировано
[1:0]	SYS_RST_REQ	2’b00	R0/W	Программный сброс: 2'b00 – не имеет эффекта 2'b01 – вызывает сброс микросхемы 2'b10 – вызывает сброс системы 2'b11 – не имеет эффекта

Таблица 1.734 - Регистр PPC_SLP_CTRL – управление режимами сна процессорных ядер PPC470S

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:6]	PPC1_SLP_TYPE	2'b00	R/W	Выбор типа сна ядра PPC1 при приходе активного сигнала CORESLEEPREQ: 2'b00 – CPU_DOZE 2'b01 – CPU_SLP 2'b10 – CPU_L2C_SLP 2'b11 - зарезервировано
[5:4]	PPC0_SLP_TYPE	2'b00	R/W	Выбор типа сна ядра PPC0 при приходе активного сигнала CORESLEEPREQ: 2'b00 – CPU_DOZE 2'b01 – CPU_SLP 2'b10 – CPU_L2C_SLP 2'b11 - зарезервировано
[3]	PPC1_WU_INT	1'b0	R/W	Программное прерывание для пробуждения ядра PPC1. MPIC128[72].
[2]	PPC0_WU_INT	1'b0	R/W	Программное прерывание для пробуждения ядра PPC0. MPIC128[71].
[1]	PPC1_SLP_STATUS	1'b0	R	Статус режима сна ядра PPC1: Устанавливается в 1 когда ядро находится в режиме сна указанном в регистре PPC1_SLP_TYPE
[0]	PPC0_SLP_STATUS	1'b0	R	Статус режима сна ядра PPC0: Устанавливается в 1 когда ядро находится в режиме сна указанном в регистре PPC0_SLP_TYPE

1.3.13.1.3 Управление процедурой самовосстановления блоков памяти с функцией redundancy.

Таблица 1.735 - Регистр BISR_IM0 – управление механизмом самовосстановления памяти IM0 с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1'b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1'b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1'b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления

Таблица 1.736 - Регистр BISR_L2C_0 – управление механизмом самовосстановления памяти L2C_0 с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1’b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1’b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1’b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1’b0	R/W	установка в 1 инициирует запуск операции самовосстановления

Таблица 1.737 - Регистр BISR_L2C_1 – управление механизмом самовосстановления памяти L2C_1 с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1’b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1’b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1’b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1’b0	R/W	установка в 1 инициирует запуск операции самовосстановления

Таблица 1.738 - Регистр BISR_DSP0_IM1_0 – управление механизмом самовосстановления памяти DSP0_IM1_0 с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1’b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1’b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1’b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1’b0	R/W	установка в 1 инициирует запуск операции самовосстановления

Таблица 1.739 - Регистр BISR_DSP0_IM1_1 – управление механизмом самовосстановления памяти DSP0_IM1_1 с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1’b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1’b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1’b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1’b0	R/W	установка в 1 инициирует запуск операции самовосстановления

Таблица 1.740 - Регистр BISR_DSP0_L2C – управление механизмом самовосстановления памяти DSP0_L2C с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1’b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1’b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1’b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1’b0	R/W	установка в 1 инициирует запуск операции самовосстановления

Таблица 1.741 - Регистр BISR_DSP1_IM1_0 – управление механизмом самовосстановления памяти DSP1_IM1_0 с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1’b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1’b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1’b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1’b0	R/W	установка в 1 инициирует запуск операции самовосстановления

Таблица 1.742 - Регистр BISR_DSP1_IM1_1 – управление механизмом самовосстановления памяти DSP1_IM1_1 с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1'b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1'b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1'b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления

Таблица 1.743 - Регистр BISR_DSP1_L2C – управление механизмом самовосстановления памяти DSP1_L2C с redundancy

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	REDCORR	1'b0	R	установка в 1 сигнализирует о корректной работе резервных элементов и возможности дальнейшего переконфигурирования памяти (если память работает корректно, но с использованием резервных элементов, это означает, что в будущем не будет возможности новой коррекции, в этом случае бит будет равным 0)
[3]	RSLT	1'b0	R	установка в 1 сигнализирует об успешном выполнении операции самовосстановления
[2]	FINISH	1'b0	R	установка в 1 сигнализирует о завершении операции самовосстановления
[1]	PROG_START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления в программном режиме
[0]	START	1'b0	R/W	установка в 1 инициирует запуск операции самовосстановления

1.3.13.1.4 Управление характеристиками буферов ввода-вывода

В регистрах определяются характеристики буферов ввода-вывода. Мощность буфера, режим отключения питания и так далее.

Регистры IFSYS_IO_CONF2- IFSYS_IO_CONF14 задают конфигурацию для IO-буферов типа SRP_BI_SDS_33V_STB.

Поля каждого регистра соответствуют одноименным IO-буферам, приведенным в документе MM7705 IO list.xls и подключаются к конфигурационным входам IO-буферов типа SRP_BI_SDS_33V_STB следующим образом(биты 2 и 3 зарезервированы и не используются):

SRP_BI_SDS_33V_STB/SMT – бит [0]
SRP_BI_SDS_33V_STB/SR– бит [1]
SRP_BI_SDS_33V_STB/E – бит [5:4]
SRP_BI_SDS_33V_STB/P – бит [7:6]

Регистры IFSYS_IO_CONF019 - IFSYS_IO_CONF22 задают конфигурацию для IO-буферов типа MIP_BI_SDS_33V_NC.

Поля каждого регистра соответствуют одноименным IO-буферам, приведенным в документе MM7705 IO list.xls и подключаются к конфигурационным входам IO-буферов типа MIP_BI_SDS_33V_NC следующим образом:

MIP_BI_SDS_33V_NC/SR – бит [0]

Регистр IFSYS_IO_CONF34 задает конфигурацию для IO-буферов типа I2P_ON_003_33V_NC

Поля каждого регистра соответствуют одноименным IO-буферам, приведенным в документе MM7705 IO list.xls и подключаются к конфигурационным входам IO-буферов типа I2P_ON_003_33V_NC следующим образом:

I2P_ON_003_33V_NC/SMT – бит [0]

Таблица 1.744

Название регистра	Тип доступа	Поля регистра		Значение по умолчанию
IFSYS_IO_CONF0	R/W	Резерв	[0]	1'b0
		Резерв	[1]	1'b0
		Резерв	[2]	1'b0
		Резерв	[3]	1'b0
		Резерв	[4]	1'b0
		Резерв	[5]	1'b0
		Резерв	[6]	1'b0
		Резерв	[7]	1'b0
		Резерв	[8]	1'b0
		Резерв	[9]	1'b0
		Резерв	[10]	1'b0
		Резерв	[11]	1'b0
		Резерв	[12]	1'b0
		Резерв	[13]	1'b0
		Резерв	[14]	1'b0
		Резерв	[15]	1'b0
		Резерв	[16]	1'b0
		Резерв	[17]	1'b0
		Резерв	[18]	1'b0
		Резерв	[19]	1'b0
		Резерв	[20]	1'b0
		Резерв	[21]	1'b0
		Резерв	[22]	1'b0
		Резерв	[23]	1'b0
		Резерв	[24]	1'b0
		Резерв	[25]	1'b0
		Резерв	[26]	1'b0
		Резерв	[27]	1'b0
		Резерв	[28]	1'b0
		Резерв	[29]	1'b0
		Резерв	[30]	1'b0
		Резерв	[31]	1'b0

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра					Значение по умолчанию
IFSYS_IO_CONF1	R/W	Резерв	[	0	]		1'b0
		Резерв	[	1	]		1'b0
		Резерв	[	2	]		1'b0
		Резерв	[	3	]		1'b0
		Резерв	[	4	]		1'b0
		Резерв	[	5	]		1'b0
		Резерв	[	6	]		1'b0
		Резерв	[	7	]		1'b0
		Резерв	[	8	]		1'b0
		Резерв	[	9	]		1'b0
		Резерв	[	10	]		1'b0
		Резерв	[	11	]		1'b0
		Резерв	[	12	]		1'b0
		Резерв	[	13	]		1'b0
		Резерв	[	14	]		1'b0
		Резерв	[	15	]		1'b0
		Резерв	[	16	]		1'b0
		Резерв	[	17	]		1'b0
		Резерв	[	18	]		1'b0
		Резерв	[	19	]		1'b0
		Резерв	[	20	]		1'b0
		Резерв	[	21	]		1'b0
		Резерв	[	22	]		1'b0
		Резерв	[	23	]		1'b0
		Резерв	[	24	]		1'b0
		Резерв	[	25	]		1'b0
		Резерв	[	26	]		1'b0
		Резерв	[	27	]		1'b0
		Резерв	[	28	]		1'b0
		Резерв	[	29	]		1'b0
		Резерв	[	30	]		1'b0
		Резерв	[	31	]		1'b0
IFSYS_IO_CONF2	R/W	LSCB0_CLK_IN_0	[	7	:	0]	8'b00010010
		LSCB0_TXA0	[	15	:	8]	8'b00010010
		LSCB0_nTXA0	[	23	:	16]	8'b00010010
		LSCB0_RXA0	[	31	:	24]	8'b00010010
IFSYS_IO_CONF3	R/W	LSCB0_nRXA0	[	7	:	0]	8'b00010010
		LSCB0_TX_INH_A0	[	15	:	8]	8'b00010010
		LSCB0_TXB0	[	23	:	16]	8'b00010010
		LSCB0_nTXB0	[	31	:	24]	8'b00010010
IFSYS_IO_CONF4	R/W	LSCB0_RXB0	[	7	:	0]	8'b00010010
		LSCB0_nRXB0	[	15	:	8]	8'b00010010
		LSCB0_TX_INH_B0	[	23	:	16]	8'b00010010
		LSCB1_CLK_IN_1	[	31	:	24]	8'b00010010
IFSYS_IO_CONF5	R/W	LSCB1_TXA1	[	7	:	0]	8'b00010010
		LSCB1_nTXA1	[	15	:	8]	8'b00010010
		LSCB1_RXA1	[	23	:	16]	8'b00010010
		LSCB1_nRXA1	[	31	:	24]	8'b00010010
IFSYS_IO_CONF6	R/W	LSCB1_TX_INH_A1	[	7	:	0]	8'b00010010
		LSCB1_TXB1	[	15	:	8]	8'b00010010
		LSCB1_nTXB1	[	23	:	16]	8'b00010010
		LSCB1_RXB1	[	31	:	24]	8'b00010010

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра					Значение по умолчанию
IFSYS_IO_CONF7	R/W	LSCB1_nRXB1	[	7	:	0]	8'b00010010
		LSCB1_TX_INH_B1	[	15	:	8]	8'b00010010
		GPIO0_0	[	23	:	16]	8'b00010010
		GPIO0_1	[	31	:	24]	8'b00010010
IFSYS_IO_CONF8	R/W	GPIO0_2	[	7	:	0]	8'b00010010
		GPIO0_3	[	15	:	8]	8'b00010010
		GPIO0_4	[	23	:	16]	8'b00010010
		GPIO0_5	[	31	:	24]	8'b00010010
IFSYS_IO_CONF9	R/W	GPIO0_6	[	7	:	0]	8'b00010010
		GPIO0_7	[	15	:	8]	8'b00010010
		GPIO1_0	[	23	:	16]	8'b00010010
		GPIO1_1	[	31	:	24]	8'b00010010
IFSYS_IO_CONF10	R/W	GPIO1_2	[	7	:	0]	8'b00010010
		GPIO1_3	[	15	:	8]	8'b00010010
		GPIO1_4	[	23	:	16]	8'b00010010
		GPIO1_5	[	31	:	24]	8'b00010010
IFSYS_IO_CONF11	R/W	GPIO1_6	[	7	:	0]	8'b00010010
		GPIO1_7	[	15	:	8]	8'b00010010
		SPI0_SCLK	[	23	:	16]	8'b00010010
		SPI0_MOSI	[	31	:	24]	8'b00010010
IFSYS_IO_CONF12	R/W	SPI0_MISO	[	7	:	0]	8'b00010010
		SPI1_SCLK	[	15	:	8]	8'b00010010
		SPI1_MOSI	[	23	:	16]	8'b00010010
		SPI1_MISO	[	31	:	24]	8'b00010010
IFSYS_IO_CONF13	R/W	UART0_RXD	[	7	:	0]	8'b00010010
		UART0_TXD	[	15	:	8]	8'b00010010
		UART1_RXD	[	23	:	16]	8'b00010010
		UART1_TXD	[	31	:	24]	8'b00010010
IFSYS_IO_CONF14	R/W	UART2_RXD	[	7	:	0]	8'b00010010
		UART2_TXD	[	15	:	8]	8'b00010010
		резерв	[	23	:	16]	8'b00000000
		резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF15	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF16	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF17	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF18	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра		Значение по умолчанию
IFSYS_IO_CONF19	R/W	MII_RX_CLK	[0]	1'b0
		MII_RX_COL	[1]	1'b0
		MII_RX_CRS	[2]	1'b0
		MII_RXD0	[3]	1'b0
		MII_RXD1	[4]	1'b0
		MII_RXD2	[5]	1'b0
		MII_RXD3	[6]	1'b0
		MII_RX_ER	[7]	1'b0
		MII_RX_DV	[8]	1'b0
		MII_TX_CLK	[9]	1'b0
		MII_TX_EN	[10]	1'b0
		MII_TX_ER	[11]	1'b0
		MII_TXD0	[12]	1'b0
		MII_TXD1	[13]	1'b0
		MII_TXD2	[14]	1'b0
		MII_TXD3	[15]	1'b0
		MII_MDC	[16]	1'b0
		MII_MDIO	[17]	1'b0
		Резерв	[18]	1'b0
		Резерв	[19]	1'b0
		SDIO_CLK	[20]	1'b0
		SDIO_CMD	[21]	1'b0
		SDIO_D0	[22]	1'b0
		SDIO_D1	[23]	1'b0
		SDIO_D2	[24]	1'b0
		SDIO_D3	[25]	1'b0
		Резерв	[26]	1'b0
		Резерв	[27]	1'b0
		Резерв	[28]	1'b0
		Резерв	[29]	1'b0
		Резерв	[30]	1'b0
		Резерв	[31]	1'b0

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра		Значение по умолчанию
IFSYS_IO_CONF20	R/W	EXT_MEM_D39	[0]	1'b0
		EXT_MEM_D38	[1]	1'b0
		EXT_MEM_D37	[2]	1'b0
		EXT_MEM_D36	[3]	1'b0
		EXT_MEM_D35	[4]	1'b0
		EXT_MEM_D34	[5]	1'b0
		EXT_MEM_D33	[6]	1'b0
		EXT_MEM_D32	[7]	1'b0
		EXT_MEM_D31	[8]	1'b0
		EXT_MEM_D30	[9]	1'b0
		EXT_MEM_D29	[10]	1'b0
		EXT_MEM_D28	[11]	1'b0
		EXT_MEM_D27	[12]	1'b0
		EXT_MEM_D26	[13]	1'b0
		EXT_MEM_D25	[14]	1'b0
		EXT_MEM_D24	[15]	1'b0
		EXT_MEM_D23	[16]	1'b0
		EXT_MEM_D22	[17]	1'b0
		EXT_MEM_D21	[18]	1'b0
		EXT_MEM_D20	[19]	1'b0
		EXT_MEM_D19	[20]	1'b0
		EXT_MEM_D18	[21]	1'b0
		EXT_MEM_D17	[22]	1'b0
		EXT_MEM_D16	[23]	1'b0
		EXT_MEM_D15	[24]	1'b0
		EXT_MEM_D14	[25]	1'b0
		EXT_MEM_D13	[26]	1'b0
		EXT_MEM_D12	[27]	1'b0
		EXT_MEM_D11	[28]	1'b0
		EXT_MEM_D10	[29]	1'b0
		EXT_MEM_D9	[30]	1'b0
		EXT_MEM_D8	[31]	1'b0

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра		Значение по умолчанию
IFSYS_IO_CONF21	R/W	EXT_MEM_D7	[0]	1'b0
		EXT_MEM_D6	[1]	1'b0
		EXT_MEM_D5	[2]	1'b0
		EXT_MEM_D4	[3]	1'b0
		EXT_MEM_D3	[4]	1'b0
		EXT_MEM_D2	[5]	1'b0
		EXT_MEM_D1	[6]	1'b0
		EXT_MEM_D0	[7]	1'b0
		EXT_MEM_BHE_H	[8]	1'b0
		EXT_MEM_BHE_L	[9]	1'b0
		EXT_MEM_BLE_H	[10]	1'b0
		EXT_MEM_BLE_L	[11]	1'b0
		EXT_MEM_OE	[12]	1'b0
		EXT_MEM_WE	[13]	1'b0
		EXT_MEM_A25	[14]	1'b0
		EXT_MEM_A24	[15]	1'b0
		EXT_MEM_A23	[16]	1'b0
		EXT_MEM_A22	[17]	1'b0
		EXT_MEM_A21	[18]	1'b0
		EXT_MEM_A20	[19]	1'b0
		EXT_MEM_A19	[20]	1'b0
		EXT_MEM_A18	[21]	1'b0
		EXT_MEM_A17	[22]	1'b0
		EXT_MEM_A16	[23]	1'b0
		EXT_MEM_A15	[24]	1'b0
		EXT_MEM_A14	[25]	1'b0
		EXT_MEM_A13	[26]	1'b0
		EXT_MEM_A12	[27]	1'b0
		EXT_MEM_A11	[28]	1'b0
		EXT_MEM_A10	[29]	1'b0
		EXT_MEM_A9	[30]	1'b0
		EXT_MEM_A8	[31]	1'b0

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра		Значение по умолчанию
IFSYS_IO_CONF22	R/W	EXT_MEM_A7	[0]	1'b0
		EXT_MEM_A6	[1]	1'b0
		EXT_MEM_A5	[2]	1'b0
		EXT_MEM_A4	[3]	1'b0
		EXT_MEM_A3	[4]	1'b0
		EXT_MEM_A2	[5]	1'b0
		EXT_MEM_A1	[6]	1'b0
		EXT_MEM_A0	[7]	1'b0
		EXT_MEM_CE5	[8]	1'b0
		EXT_MEM_CE4	[9]	1'b0
		EXT_MEM_CE3	[10]	1'b0
		EXT_MEM_CE2	[11]	1'b0
		EXT_MEM_CE1	[12]	1'b0
		EXT_MEM_CE0	[13]	1'b0
		Резерв	[14]	1'b0
		Резерв	[15]	1'b0
		Резерв	[16]	1'b0
		Резерв	[17]	1'b0
		Резерв	[18]	1'b0
		Резерв	[19]	1'b0
		Резерв	[20]	1'b0
		Резерв	[21]	1'b0
		Резерв	[22]	1'b0
		Резерв	[23]	1'b0
		Резерв	[24]	1'b0
		Резерв	[25]	1'b0
		Резерв	[26]	1'b0
		Резерв	[27]	1'b0
		Резерв	[28]	1'b0
		Резерв	[29]	1'b0
		Резерв	[30]	1'b0
		Резерв	[31]	1'b0
IFSYS_IO_CONF23	R/W	Резерв	[7 : 0]	8'b00000000
		Резерв	[15 : 8]	8'b00000000
		Резерв	[23 : 16]	8'b00000000
		Резерв	[31 : 24]	8'b00000000
IFSYS_IO_CONF24	R/W	Резерв	[7 : 0]	8'b00000000
		Резерв	[15 : 8]	8'b00000000
		Резерв	[23 : 16]	8'b00000000
		Резерв	[31 : 24]	8'b00000000
IFSYS_IO_CONF25	R/W	Резерв	[7 : 0]	8'b00000000
		Резерв	[15 : 8]	8'b00000000
		Резерв	[23 : 16]	8'b00000000
		Резерв	[31 : 24]	8'b00000000
IFSYS_IO_CONF26	R/W	Резерв	[7 : 0]	8'b00000000
		Резерв	[15 : 8]	8'b00000000
		Резерв	[23 : 16]	8'b00000000
		Резерв	[31 : 24]	8'b00000000
IFSYS_IO_CONF27	R/W	Резерв	[7 : 0]	8'b00000000
		Резерв	[15 : 8]	8'b00000000
		Резерв	[23 : 16]	8'b00000000
		Резерв	[31 : 24]	8'b00000000

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра					Значение по умолчанию
IFSYS_IO_CONF28	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF29	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF30	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF31	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF32	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF33	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF34	R/W	Резерв	[	7	:	0]	8'b00000000
		Резерв	[	15	:	8]	8'b00000000
		Резерв	[	23	:	16]	8'b00000000
		Резерв	[	31	:	24]	8'b00000000
IFSYS_IO_CONF35	R/W	I2C0_SDA			[	0]	1’b0
		I2C0_SCL			[	1]	1’b0
		I2C1_SDA			[	2]	1’b0
		I2C1_SCL			[	3]	1’b0
		I2C2_SDA			[	4]	1’b0
		I2C2_SCL			[	5]	1’b0
MLT_IO_CONF0	R/W	VIN0_D23	[	7	:	0]	8'b00010010
		VIN0_D22	[	15	:	8]	8'b00010010
		VIN0_D21	[	23	:	16]	8'b00010010
		VIN0_D20	[	31	:	24]	8'b00010010
MLT_IO_CONF1	R/W	VIN0_D19	[	7	:	0]	8'b00010010
		VIN0_D18	[	15	:	8]	8'b00010010
		VIN0_D17	[	23	:	16]	8'b00010010
		VIN0_D16	[	31	:	24]	8'b00010010
MLT_IO_CONF2	R/W	VIN0_D15	[	7	:	0]	8'b00010010
		VIN0_D14	[	15	:	8]	8'b00010010
		VIN0_D13	[	23	:	16]	8'b00010010
		VIN0_D12	[	31	:	24]	8'b00010010
MLT_IO_CONF3	R/W	VIN0_D11	[	7	:	0]	8'b00010010
		VIN0_D10	[	15	:	8]	8'b00010010
		VIN0_D9	[	23	:	16]	8'b00010010
		VIN0_D8	[	31	:	24]	8'b00010010

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра					Значение по умолчанию
MLT_IO_CONF4	R/W	VIN0_D7	[	7	:	0]	8'b00010010
		VIN0_D6	[	15	:	8]	8'b00010010
		VIN0_D5	[	23	:	16]	8'b00010010
		VIN0_D4	[	31	:	24]	8'b00010010
MLT_IO_CONF5	R/W	VIN0_D3	[	7	:	0]	8'b00010010
		VIN0_D2	[	15	:	8]	8'b00010010
		VIN0_D1	[	23	:	16]	8'b00010010
		VIN0_D0	[	31	:	24]	8'b00010010
MLT_IO_CONF6	R/W	VIN0_HSYNC	[	7	:	0]	8'b00010010
		VIN0_VSYNC	[	15	:	8]	8'b00010010
		VIN0_DE	[	23	:	16]	8'b00010010
		VIN0_FIELD	[	31	:	24]	8'b00010010
MLT_IO_CONF7	R/W		[	7	:	0]	8'b00000000
			[	15	:	8]	8'b00000000
			[	23	:	16]	8'b00000000
			[	31	:	24]	8'b00000000
MLT_IO_CONF8	R/W	VIN1_D23	[	7	:	0]	8'b00010010
		VIN1_D22	[	15	:	8]	8'b00010010
		VIN1_D21	[	23	:	16]	8'b00010010
		VIN1_D20	[	31	:	24]	8'b00010010
MLT_IO_CONF9	R/W	VIN1_D19	[	7	:	0]	8'b00010010
		VIN1_D18	[	15	:	8]	8'b00010010
		VIN1_D17	[	23	:	16]	8'b00010010
		VIN1_D16	[	31	:	24]	8'b00010010
MLT_IO_CONF10	R/W	VIN1_D15	[	7	:	0]	8'b00010010
		VIN1_D14	[	15	:	8]	8'b00010010
		VIN1_D13	[	23	:	16]	8'b00010010
		VIN1_D12	[	31	:	24]	8'b00010010
MLT_IO_CONF11	R/W	VIN1_D11	[	7	:	0]	8'b00010010
		VIN1_D10	[	15	:	8]	8'b00010010
		VIN1_D9	[	23	:	16]	8'b00010010
		VIN1_D8	[	31	:	24]	8'b00010010
MLT_IO_CONF12	R/W	VIN1_D7	[	7	:	0]	8'b00010010
		VIN1_D6	[	15	:	8]	8'b00010010
		VIN1_D5	[	23	:	16]	8'b00010010
		VIN1_D4	[	31	:	24]	8'b00010010
MLT_IO_CONF13	R/W	VIN1_D3	[	7	:	0]	8'b00010010
		VIN1_D2	[	15	:	8]	8'b00010010
		VIN1_D1	[	23	:	16]	8'b00010010
		VIN1_D0	[	31	:	24]	8'b00010010
MLT_IO_CONF14	R/W	VIN1_HSYNC	[	7	:	0]	8'b00010010
		VIN1_VSYNC	[	15	:	8]	8'b00010010
		VIN1_DE	[	23	:	16]	8'b00010010
		VIN1_FIELD	[	31	:	24]	8'b00010010
MLT_IO_CONF15	R/W		[	7	:	0]	8'b00000000
			[	15	:	8]	8'b00000000
			[	23	:	16]	8'b00000000
			[	31	:	24]	8'b00000000
MLT_IO_CONF16	R/W	VOUT0_D23	[	7	:	0]	8'b00010010
		VOUT0_D22	[	15	:	8]	8'b00010010
		VOUT0_D21	[	23	:	16]	8'b00010010
		VOUT0_D20	[	31	:	24]	8'b00010010

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра					Значение по умолчанию
MLT_IO_CONF17	R/W	VOUT0_D19	[	7	:	0]	8'b00010010
		VOUT0_D18	[	15	:	8]	8'b00010010
		VOUT0_D17	[	23	:	16]	8'b00010010
		VOUT0_D16	[	31	:	24]	8'b00010010
MLT_IO_CONF18	R/W	VOUT0_D15	[	7	:	0]	8'b00010010
		VOUT0_D14	[	15	:	8]	8'b00010010
		VOUT0_D13	[	23	:	16]	8'b00010010
		VOUT0_D12	[	31	:	24]	8'b00010010
MLT_IO_CONF19	R/W	VOUT0_D11	[	7	:	0]	8'b00010010
		VOUT0_D10	[	15	:	8]	8'b00010010
		VOUT0_D9	[	23	:	16]	8'b00010010
		VOUT0_D8	[	31	:	24]	8'b00010010
MLT_IO_CONF20	R/W	VOUT0_D7	[	7	:	0]	8'b00010010
		VOUT0_D6	[	15	:	8]	8'b00010010
		VOUT0_D5	[	23	:	16]	8'b00010010
		VOUT0_D4	[	31	:	24]	8'b00010010
MLT_IO_CONF21	R/W	VOUT0_D3	[	7	:	0]	8'b00010010
		VOUT0_D2	[	15	:	8]	8'b00010010
		VOUT0_D1	[	23	:	16]	8'b00010010
		VOUT0_D0	[	31	:	24]	8'b00010010
MLT_IO_CONF22	R/W	VOUT0_HSYNC	[	7	:	0]	8'b00010010
		VOUT0_VSYNC	[	15	:	8]	8'b00010010
		VOUT0_DE	[	23	:	16]	8'b00010010
		VOUT0_FIELD	[	31	:	24]	8'b00010010
MLT_IO_CONF23	R/W	VOUT0_CLKO	[	7	:	0]	8'b00010010
			[	15	:	8]	8'b00000000
			[	23	:	16]	8'b00000000
			[	31	:	24]	8'b00000000
MLT_IO_CONF24	R/W	VOUT1_D23	[	7	:	0]	8'b00010010
		VOUT1_D22	[	15	:	8]	8'b00010010
		VOUT1_D21	[	23	:	16]	8'b00010010
		VOUT1_D20	[	31	:	24]	8'b00010010
MLT_IO_CONF25	R/W	VOUT1_D19	[	7	:	0]	8'b00010010
		VOUT1_D18	[	15	:	8]	8'b00010010
		VOUT1_D17	[	23	:	16]	8'b00010010
		VOUT1_D16	[	31	:	24]	8'b00010010
MLT_IO_CONF26	R/W	VOUT1_D15	[	7	:	0]	8'b00010010
		VOUT1_D14	[	15	:	8]	8'b00010010
		VOUT1_D13	[	23	:	16]	8'b00010010
		VOUT1_D12	[	31	:	24]	8'b00010010
MLT_IO_CONF27	R/W	VOUT1_D11	[	7	:	0]	8'b00010010
		VOUT1_D10	[	15	:	8]	8'b00010010
		VOUT1_D9	[	23	:	16]	8'b00010010
		VOUT1_D8	[	31	:	24]	8'b00010010
MLT_IO_CONF28	R/W	VOUT1_D7	[	7	:	0]	8'b00010010
		VOUT1_D6	[	15	:	8]	8'b00010010
		VOUT1_D5	[	23	:	16]	8'b00010010
		VOUT1_D4	[	31	:	24]	8'b00010010
MLT_IO_CONF29	R/W	VOUT1_D3	[	7	:	0]	8'b00010010
		VOUT1_D2	[	15	:	8]	8'b00010010
		VOUT1_D1	[	23	:	16]	8'b00010010
		VOUT1_D0	[	31	:	24]	8'b00010010

Продолжение таблицы 1.744

Название регистра	Тип доступа	Поля регистра					Значение по умолчанию
MLT_IO_CONF30	R/W	VOUT1_HSYNC	[	7	:	0]	8'b00010010
		VOUT1_VSYNC	[	15	:	8]	8'b00010010
		VOUT1_DE	[	23	:	16]	8'b00010010
		VOUT1_FIELD	[	31	:	24]	8'b00010010
MLT_IO_CONF31	R/W	VOUT1_CLKO	[	7	:	0]	8'b00010010
			[	15	:	8]	8'b00000000
			[	23	:	16]	8'b00000000
			[	31	:	24]	8'b00000000
MLT_IO_CONF32	R/W	AUD_D3	[	7	:	0]	8'b00010010
		AUD_D2	[	15	:	8]	8'b00010010
		AUD_D1	[	23	:	16]	8'b00010010
		AUD_D0	[	31	:	24]	8'b00010010
MLT_IO_CONF33	R/W	AUD_WS	[	7	:	0]	8'b00010010
		AUD_SCLK	[	15	:	8]	8'b00010010
		SPDIF	[	23	:	16]	8'b00010010
			[	31	:	24]	8'b00000000
MISC_IO_CONF0	R/W	TDO	[	7	:	0]	8'b00010010
		MCHKOUT	[	15	:	8]	8'b00010010
			[	23	:	16]	8'b00000000
			[	31	:	24]	8'b00000000

1.3.13.1.5 Управление запуском задач на DSP ядрах

Регистр INTSTA_DSP2CPU – состояние прерываний от ядер DSP к CPU

Каждый бит регистра фиксирует по положительному фронту соответствующее прерывание от ядер DSP.

Прерывания могут быть сброшены через регистр INTCLR_DSP2CPU.

Таблица 1.745 - Регистр INTSTA_DSP2CPU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7]	INTHP_DSP1NMC1	1’b0	R	Состояние прерывания INTLP_DSP1NMC1 0 – прерывание не активно 1 – прерывание активно
[6]	INTLP_DSP1NMC1	1’b0	R	Состояние прерывания INTLP_DSP1NMC1 0 – прерывание не активно 1 – прерывание активно
[5]	INTHP_DSP1NMC0	1’b0	R	Состояние прерывания INTLP_DSP1NMC0 0 – прерывание не активно 1 – прерывание активно
[4]	INTLP_DSP1NMC0	1’b0	R	Состояние прерывания INTLP_DSP1NMC0 0 – прерывание не активно 1 – прерывание активно
[3]	INTHP_DSP0NMC1	1’b0	R	Состояние прерывания INTHP_DSP0NMC1 0 – прерывание не активно 1 – прерывание активно

Продолжение таблицы 1.745

Биты	Название	Сброс	Доступ	Описание
[2]	INTLP_DSP0NMC1	1’b0	R	Состояние прерывания INTLP_DSP0NMC1 0 – прерывание не активно 1 – прерывание активно
[1]	INTHP_DSP0NMC0	1'b0	R	Состояние прерывания INTHP_DSP0NMC0 0 – прерывание не активно 1 – прерывание активно
[0]	INTLP_DSP0NMC0	1’b0	R	Состояние прерывания INTLP_DSP0NMC0 0 – прерывание не активно 1 – прерывание активно

Регистр INTCLR_DSP2CPU – сброс прерываний от ядер DSP к CPU

Таблица 1.746 - Регистр INTCLR_DSP2CPU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7]	INTHP_DSP1NMC1_CLR	1’b0	W1	Сброс прерывания INTLP_DSP1NMC1 Запись 1 сбрасывает прерывание
[6]	INTLP_DSP1NMC1_CLR	1’b0	W1	Сброс прерывания INTLP_DSP1NMC1 Запись 1 сбрасывает прерывание
[5]	INTHP_DSP1NMC0_CLR	1’b0	W1	Сброс прерывания INTLP_DSP1NMC0 Запись 1 сбрасывает прерывание
[4]	INTLP_DSP1NMC0_CLR	1’b0	W1	Сброс прерывания INTLP_DSP1NMC0 Запись 1 сбрасывает прерывание
[3]	INTHP_DSP0NMC1_CLR	1’b0	W1	Сброс прерывания INTHP_DSP0NMC1 Запись 1 сбрасывает прерывание
[2]	INTLP_DSP0NMC1_CLR	1’b0	W1	Сброс прерывания INTLP_DSP0NMC1 Запись 1 сбрасывает прерывание
[1]	INTHP_DSP0NMC0_CLR	1'b0	W1	Сброс прерывания INTHP_DSP0NMC0 Запись 1 сбрасывает прерывание
[0]	INTLP_DSP0NMC0_CLR	1’b0	W1	Сброс прерывания INTLP_DSP0NMC0 Запись 1 сбрасывает прерывание

Регистр INTREQ_CPU2DSP – генерация прерываний от CPU к ядрам DSP

Запись 1 в значащий бит регистра генерирует соответствующее прерывание для ядер DSP. После записи 1 биты сбрасываются в 0 автоматически.

Таблица 1.747 - Регистр INTREQ_CPU2DSP

Биты	Название	Сброс	Доступ	Описание
[31:12]	-	-	R0	зарезервировано
[11]	DSP1NMC1_ INTNMI	1’b0	W1	Немаскируемое прерывание для ядра DSP1NMC1.
[10]	DSP1NMC1_ INTHP	1’b0	W1	Высокоприоритетное прерывание для ядра DSP1NMC1.
[9]	DSP1NMC1_ INTLP	1’b0	W1	Низкоприоритетное прерывание для ядра DSP1NMC1.
[8]	DSP1NMC0_ INTNMI	1’b0	W1	Немаскируемое прерывание для ядра DSP1NMC0.
[7]	DSP1NMC0_ INTHP	1’b0	W1	Высокоприоритетное прерывание для ядра DSP1NMC0.
[6]	DSP1NMC0_ INTLP	1’b0	W1	Низкоприоритетное прерывание для ядра DSP1NMC0.
[5]	DSP0NMC1_ INTNMI	1’b0	W1	Немаскируемое прерывание для ядра DSP0NMC1.

Продолжение таблицы 1.747

Биты	Название	Сброс	Доступ	Описание
[4]	DSP0NMC1_ INTHP	1’b0	W1	Высокоприоритетное прерывание для ядра DSP0NMC1.
[3]	DSP0NMC1_ INTLP	1’b0	W1	Низкоприоритетное прерывание для ядра DSP0NMC1.
[2]	DSP0NMC0_ INTNMI	1’b0	W1	Немаскируемое прерывание для ядра DSP0NMC0.
[1]	DSP0NMC0_ INTHP	1'b0	W1	Высокоприоритетное прерывание для ядра DSP0NMC0.
[0]	DSP0NMC0_ INTLP	1’b0	W1	Низкоприоритетное прерывание для ядра DSP0NMC0.

1.3.13.1.6 Управление режимом самотестирования BIST

Регистр USB_BIST_CNTRL контроль интерфейса BIST блока USB_PHY.

Таблица 1.748 - Регистр USB_BIST_CNTRL

Биты	Название	Сброс	Доступ	Описание
[31:11]	-	-	R0	зарезервировано
[10:9]	utmi_status	2’b00	R	Монитор сигнала utmi_status[1:0]
[8]	utmi_bistdone	1’b0	R	Монитор сигнала utmi_bistdone
[7]	utmi_suspendm	1’b0	R/W	Управляет сигналом utmi_suspendm
[6]	utmi_termselect	1’b0	R/W	Управляет сигналом utmi_termselect
[5:4]	utmi_xcvrselect	2’b00	R/W	Управляет сигналом utmi_xcrvselect[1:0]
[3:2]	utmi_testcontrol	2’b01	R/W	Управляет сигналом utmi_testcontrol[1:0]
[1]	utmi_biston	1'b0	R/W	Управляет сигналом utmi_biston
[0]	utmi_bisten	1’b0	R/W	Разрешает управление сигналами для процедуры MBIST USBPHY

Значение сигналов и порядок работы процедуры BIST см. в документе INNO_USB2_GF28SLP_1xport_RCM_20150324.pdf

Регистр EM0_BIST_CNTRL контроль интерфейса BIST блока DDR3_PHY_EM0.

Таблица 1.749 - Регистр EM0_BIST_CNTRL

Биты	Название	Сброс	Доступ	Описание
[31:7]	-	-	R0	зарезервировано
[6]	bist_error	1’b0	R	Состояние сигнала bist_error
[5]	bist_complete	1’b0	R	Состояние сигнала bist_complete
[4]	dfi_init_complete	1’b0	R	Состояние сигнала dfi_init_complete
[3]	dfi_init_start	1’b1	R/W	Управление сигналом dfi_init_start
[2]	bist_start	1’b0	R/W	Управление сигналом bist_start
[1]	bist_mux	1'b0	R/W	Управление сигналом bist_mux
[0]	bist_mode	1’b0	R/W	Управление сигналом bits_mode

Значение сигналов и порядок работы процедуры BIST см. в документе INNO_DDR_COMBO_PHY_IG_v2.76

Регистр EM1_BIST_CNTRL контроль интерфейса BIST блока DDR3_PHY_EM1.

Таблица 1.750 - Регистр EM1_BIST_CNTRL

Биты	Название	Сброс	Доступ	Описание
[31:7]	-	-	R0	зарезервировано
[6]	bist_error	1'b0	R	Состояние сигнала bist_error
[5]	bist_complete	1'b0	R	Состояние сигнала bist_complete
[4]	dfi_init_complete	1'b0	R	Состояние сигнала dfi_init_complete
[3]	dfi_init_start	1'b1	R/W	Управление сигналом dfi_init_start
[2]	bist_start	1'b0	R/W	Управление сигналом bist_start
[1]	bist_mux	1'b0	R/W	Управление сигналом bist_mux
[0]	bist_mode	1'b0	R/W	Управление сигналом bits_mode

Значение сигналов и порядок работы процедуры BIST см. в документе INNO_DDR_COMBO_PHY_IG_v2.76

1.3.13.1.7 Другие регистры

Регистр MLT_ADR_MSB0 – задание старших бит адреса master-устройств multimedia

Таблица 1.751 - Регистр MLT_ADR_MSB0

Биты	Название	Сброс	Доступ	Описание
[31:28]	ADR_MSB_CODA_PRI	4'h0	R/W	Биты [35:32] адреса hlb_codec.coda_pri
[27:24]	ADR_MSB_CODA_SEC	4'h0	R/W	Биты [35:32] адреса hlb_codec.coda_sec
[23:20]	ADR_MSB_OSD1_1	4'h0	R/W	Биты [35:32] адреса hlb_video1.osd1
[19:16]	ADR_MSB_OSD0_1	4'h0	R/W	Биты [35:32] адреса hlb_video1.osd0
[15:12]	ADR_MSB_MVL1	4'h0	R/W	Биты [35:32] адреса hlb_video1.mvl
[11:8]	ADR_MSB_OSD1_0	4'h0	R/W	Биты [35:32] адреса hlb_video0.osd1
[7:4]	ADR_MSB_OSD0_0	4'h0	R/W	Биты [35:32] адреса hlb_video0.osd0
[3:0]	ADR_MSB_MVL0	4'h0	R/W	Биты [35:32] адреса hlb_video0.mvl

Регистр MLT_ADR_MSB1 – задание старших бит адреса master-устройств multimedia

Таблица 1.752 - Регистр MLT_ADR_MSB1

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:4]	ADR_MSB_AUDMA	4'h0	R/W	Биты [35:32] адреса hlb_audio.audma
[3:0]	ADR_MSB_JPEG	4'h0	R/W	Биты [35:32] адреса hlb_codec.jpeg

1.3.13.2 Блок сдвоенных таймеров DIT

Блок сдвоенных таймеров имеет следующие характеристики:

- два 32/16 – разрядных таймера;
- для каждого таймера можно настроить режим его работы (свободный счет, периодичный, одиночный);
- каждый таймер имеет собственный выход прерывания;
- подключение к системной шине микросхемы через AMBA APB интерфейс, используются полный, 32 разрядный интерфейс шины данных.

1.3.13.2.1 Устройство блока сдвоенных таймеров

Блок не имеет сигналов, соединенных с внешними выводами микросхемы.
На рисунке 1.305 приведена структурная схема блока:

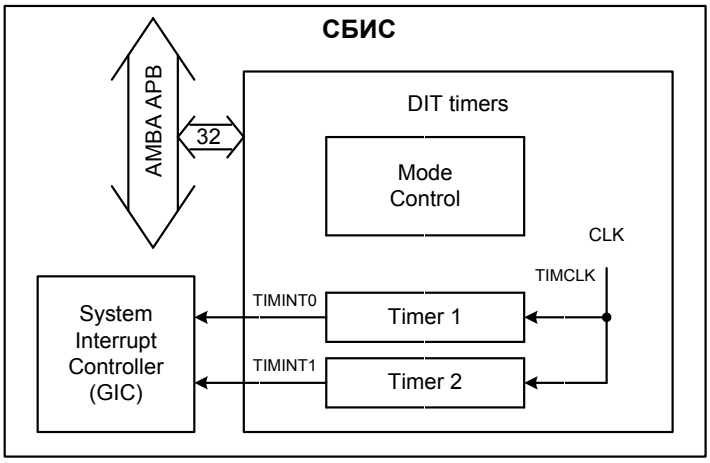


Рисунок 1.305 – Структурная схема блока сдвоенных таймеров

Блок состоит из двух идентичных блоков Timer 1 и Timer 2, каждый из которых может функционировать в 16 или 32-битном режиме. Внутри каждого блока находится счетчик FRC (Free Running Counter).

Оба таймера могут быть программно настроены на работу в одном из следующих режимов:

свободный счет (free-running) – счетчик таймера постоянно декрементируется, счет автоматически начинается с максимального значения после достижения нуля;

периодичный (periodic) – аналогично предыдущему, только после достижения нуля счет начинается со значения, предварительно загруженного в регистр TimerXLoad;

одиночный (one-shot) – счетчик начинает декрементироваться со значения, загруженного в регистр TimerXLoad, после достижения нуля счет останавливается.

Ниже приведена последовательность работы с таймерами:

Необходимо провести начальные настройки используемого таймера, т.е. прописать поля регистра TimerXControl.

Загрузить начальное значение счетчика в регистр TimerXLoad.

Разрешить счет путем установки бита TimerEn регистра TimerXControl. Если после установки этого бита записать новое значение в регистр TimerXLoad, то счетчик продолжит декрементироваться с нового значения.

Другой способ записать новое значение счетчика заключается в записи фонового регистра TimerXBGLoad. Эта запись не будет иметь мгновенного эффекта, вместо этого записанное значение перепишется в регистр TimerXLoad автоматически по достижению счетчиком нулевого значения.

Текущее значение счетчика может быть считано из регистра TimerXValue в любой момент времени.

Каждый раз, когда счетчик достигает нулевого значения, генерируется прерывание. Для сброса сгенерированного прерывания необходимо произвести запись в регистр TimerXIntClr. Генерация прерывания может быть замаскирована путем установки соответствующих бит регистра управления TimerXControl.

1.3.13.2.2 Регистровая модель блока сдвоенных таймеров

Программно доступные регистры блока сдвоенных таймеров DIT расположены в области памяти периферийных устройств кластера REG_CPU и имеют базовый адрес DIT Base = 0x000A_2000 и общий размер 4 Кб. Спецификация регистров представлена в таблице 1.753:

Таблица 1.753 - Спецификация регистров блока DIT

Адрес (байтовый доступ)	Тип	Разрядность	Начальное значение	Имя	Описание
0x000A_2000 hex	ЧТ/ЗП	32	0x00000000	Timer1Load	Регистр загрузки. TimerXLoad
0x000A_2004 hex	ЧТ	32	0xFFFFFFFF	Timer1Value	Регистр текущего значения. TimerXValue
0x000A_2008 hex	ЧТ/ЗП	8	0x20	Timer1Control	Регистр управления. TimerXControl
0x000A_200C hex	ЗП	-	-	Timer1IntClr	Регистр снятия прерываний. TimerXIntClr
0x000A_2010 hex	ЧТ	1	0x0	Timer1RIS	Регистр состояния прерывания до наложения маски. TimerXRIS
0x000A_2014 hex	ЧТ	1	0x0	Timer1MIS	Регистр состояния прерывания после наложения маски. TimerXMIS
0x000A_2018 hex	ЧТ/ЗП	32	0x00000000	Timer1BGLoad	Фоновый регистр загрузки. TimerXBGLoad
0x000A_2020 hex	ЧТ/ЗП	32	0x00000000	Timer2Load	Регистр загрузки. TimerXLoad
0x000A_2024 hex	ЧТ	32	0xFFFFFFFF	Timer2Value	Регистр текущего значения. TimerXValue
0x000A_2028 hex	ЧТ/ЗП	8	0x20	Timer2Control	Регистр управления. TimerXControl
0x000A_202C hex	ЗП	-	-	Timer2IntClr	Регистр снятия прерываний. TimerXIntClr
0x000A_2030 hex	ЧТ	1	0x0	Timer2RIS	Регистр состояния прерывания до наложения маски. TimerXRIS
0x000A_2034hex	ЧТ	1	0x0	Timer2MIS	Регистр состояния прерывания после наложения маски. TimerXMIS
0x000A_2038 hex	ЧТ/ЗП	32	0x00000000	Timer2BGLoad	Фоновый регистр загрузки. TimerXBGLoad
0x000A_203C hex - 0x000A_2EFC hex	-	-	-	-	Зарезервировано
0x000A_2F00 hex	ЧТ/ЗП	1	0x0	TimerITCR	Регистр управления тестом интеграции. TimerITCR
0x000A_2F04 hex	ЗП	2	0x0	TimerITOP	Регистр управления тестом интеграции. TimerITOP
0x000A_2F08 hex - 0x000A_2FDC hex	-	-	-	-	Зарезервировано
0x000A_2FE0 hex	ЧТ	8	0x04	TimerPeriphID0	Идентификатор периферийного устройства. TimerPeriphID0 биты [7:0]

Продолжение таблицы 1.753

Адрес (байтовый доступ)	Тип	Разрядность	Начальное значение	Имя	Описание
0x000A_2FE4 hex	ЧТ	8	0x18	TimerPeriphID1	Идентификатор периферийного устройства. TimerPeriphID1 биты [15:8]
0x000A_2FE8 hex	ЧТ	8	0x04	TimerPeriphID2	Идентификатор периферийного устройства. TimerPeriphID2 биты [23:16]
0x000A_2FEC hex	ЧТ	8	0x00	TimerPeriphID3	Идентификатор периферийного устройства. TimerPeriphID3 биты [31:24]
0x000A_2FF0 hex	ЧТ	8	0x0D	TimerPCellID0	PrimeCell идентификатор. TimerPCellID0 биты [7:0]
0x000A_2FF4 hex	ЧТ	8	0xF0	TimerPCellID1	PrimeCell идентификатор. TimerPCellID1 биты [15:8]
0x000A_2FF8 hex	ЧТ	8	0x05	TimerPCellID2	PrimeCell идентификатор. TimerPCellID2 биты [23:16]
0x000A_2FFC hex	ЧТ	8	0xB1	TimerPCellID3	PrimeCell идентификатор. TimerPCellID3 биты [31:24]

Далее приведено подробное описание каждого регистра. Заметим, что использование X в названии регистров означает, что регистр относится к Timer 1, либо к блоку Timer 2.

1.3.13.2.2.1 Регистр загрузки. TimerXLoad

Регистр TimerXLoad - это 32-х битный регистр, в который программно записывается значение, начиная с которого счетчик должен декрементироваться. Если периодический режим активен и текущее значение достигает нуля, то происходит перезагрузка счетчика.

При записи, текущее значение счетчика незамедлительно меняется на значение, записанное по активному фронту синхросигнала TIMCLK.

Отметим, что минимальным значением для TimerXLoad является 1. Если значение TimerXLoad установлено в 0, то прерывания вырабатываются незамедлительно.

Значение данного регистра всегда переписывается, если прописано значение регистра TimerXBGLoad. Но текущее значение счета изменяется не сразу. Если значения записываются сразу в оба регистра: TimerXLoad и TimerXBGLoad, до прихода активного фронта сигнала TIMCLK, тогда по следующему активному фронту сигнала TIMCLK значение, записанное в регистре TimerXLoad, заменяется текущим значением счета. После чего, каждый раз, когда счетчик достигает значения нуль, текущее значение устанавливается в значение, записанное в TimerXBGLoad.

Значение, считанное из TimerXLoad, - это значение, которое будет прописано в счетчике, как только он станет равным нулю в периодическом режиме.

1.3.13.2.2.2 Регистр текущего значения. TimerXValue

Регистр TimerXValue – 32-х битный регистр, предназначенный только для чтения, содержит текущее значение декрементирующего счетчика.

После процесса загрузки, когда прописывается новое значение в TimerXLoad, в регистре TimerXValue незамедлительно отражается новое загруженное значение

Отметим, что старшие 16 бит 32-х битного регистра TimerXValue не должны автоматически устанавливаться в нуль, когда используется 16-битный временной режим. Если TimerXValue находится в 16-битном режиме, тогда старшие 16 бит регистра TimerXValue должны иметь ненулевые значения, если перед этим TimerXValue был в 32-битном режиме, и произошла запись в регистр TimerXLoad, прежде чем перейти в 16-битный режим.

1.3.13.2.2.3 Регистр управления. TimerXControl

Формат регистра управления приведен в таблице 1.754.

Таблица 1.754 - Формат регистра TimerXControl

Биты	Название	Тип	Функции
[31:8]	-	-	Резервные биты, не изменяются, игнорируются при чтении
[7]	TimerEn	ЧТ/ЗП	Бит разрешения: 0 = Timer X FRC неактивен (по умолчанию) 1 = Timer X FRC активен
[6]	TimerMode	ЧТ/ЗП	Бит режима: 0 = Timer X FRC находится в режиме свободного доступа (по умолчанию) 1 = Timer X FRC находится в периодическом режиме.
[5]	IntEnable	ЧТ/ЗП	Бит разрешения прерываний: 0 = Timer X FRC Прерывание неактивно 1 = Timer X FRC Прерывание активно (по умолчанию).
[4]	-	-	Резервные биты, не изменяются, игнорируются при чтении
[3:2]	TimerPre	ЧТ/ЗП	Биты масштабирования частоты счета (делитель частоты): 00 = синхросигнал не делится (по умолчанию) 01 = синхросигнал делится на 16 10 = синхросигнал делится на 256 11 = не используется.
[1]	TimerSize	ЧТ/ЗП	Выбирает16/32 битный счетчик используется: 0 = 16-битный счетчик (по умолчанию) 1 = 32-битный счетчик.
[0]	OneShot	ЧТ/ЗП	Выбирает однократный или многократный режим счетчика: 0 = многократный режим (по умолчанию) 1 = однократный режим.

Заметим, что состояние счетчика, разрядность или делитель частоты - это установки, которые не должны меняться, пока Timer X FRC работает. Если необходима новая конфигурация, тогда Timer X FRC должен находиться в неактивном состоянии и значения новой конфигурации записываются в соответствующие регистры. Timer X FRC должен быть заново активирован, после чего изменения конфигурации заканчиваются. Отказ следовать данному порядку действий может послужить причиной непредсказуемого поведения блока.

1.3.13.2.2.4 Регистр снятия прерываний. TimerXIntClr

Любая запись в данный регистр снимает прерывание с выхода счетчика.

1.3.13.2.2.5 Регистр состояния прерывания до наложения маски. TimerXRIS

Регистр TimerXRIS указывает на состояние прерываний до наложения маски от счетчика. Формат регистра представлен в таблице 1.755.

Таблица 1.755 - Формат регистра TimerXRIS

Биты	Имя	Тип	Функция
[31:1]	-	-	Резервные биты, не изменяются, игнорируются при чтении
[0]	TimerXRIS	ЧТ	Состояние необработанных прерываний счетчика

1.3.13.2.2.6 Регистр состояния прерывания после наложения маски. TimerXMIS

Регистр TimerXMIS определяет прерывания после наложения маски от счетчика. Это значение представляется логическим элементом "И" между битом состояния необработанного прерывания (TimerXRIS) и битом разрешения прерываний (IntEnable) регистра управления, и

представляет аналогичное значение выходного сигнала TIMERINTX. Формат регистра представлен в таблице 1.756.

Таблица 1.756 - Формат регистра TimerXMIS

Биты	Имя	Тип	Функция
[31:1]	-	-	Резервные биты, не изменяются, игнорируются при чтении
[0]	TimerXMIS	ЧТ	Состояние бита, разрешающего прерывание от счетчика

1.3.13.2.2.7 Фоновый регистр загрузки. TimerXBGLoad

Регистр TimerXBGLoad – является 32-битным регистром, который содержит значение, с которого счетчик начинает декрементироваться. Данное значение используется для перезагрузки счетчика в периодическом режиме, и когда текущее значение счета достигает нуля.

Это предоставляет альтернативный метод доступа к регистру TimerXLoad. Отличием является то, что запись в TimerXBGLoad не означает, что счетчик устанавливает новое значение сразу.

Чтение из этого регистра вернет то же самое значение, возвратное из TimerXLoad. Для получения более полной информации смотри раздел «Регистр загрузки. TimerXLoad».

1.3.13.2.2.8 Идентификатор периферийного устройства. TimerPeriphID0-3

Регистр TimerPeriphID0-3 представляет четыре 8-битных регистра, предназначенных только для чтения, которые охватывают адресное пространство от 0xFE0 до 0xFEC. Регистры могут быть концептуально представлены как 32-битный регистр. В таблице 1.757 приведен формат полей этого регистра.

Таблица 1.757 - Поля идентификатора периферийного устройства TimerPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для DIT - 0x804
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM - 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока. Номер ревизии начинается с нуля
Configuration[31:24]	Является вариантом конфигурацией периферии. Для DIT 0x0

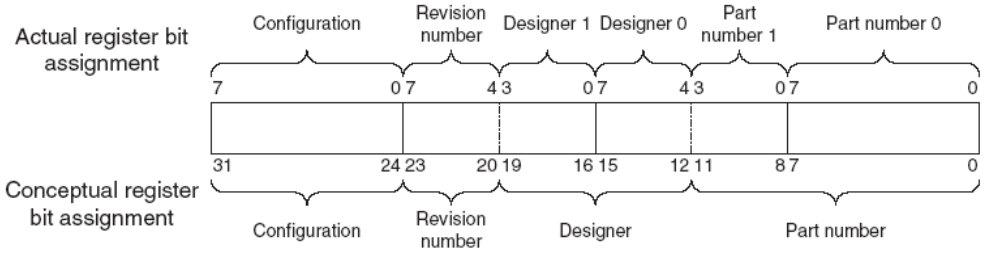


Рисунок 1.306 - Формат регистра TimerPeriphID0-3

Отметим, что при проектировании важно помнить, что периферийные блоки занимают 4 Кб адресного пространства. Обращение к периферийным регистрам должно быть 32-битным, независимо от их реальной разрядности.

1.3.13.2.2.9 PrimeCell идентификатор. TimerPCellID0-3

Регистры TimerPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, предназначенные только для чтения, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. Регистр TimerPCellID установлен в 0xB105F00D. На рисунке 1.307 приведен формат этого регистра.

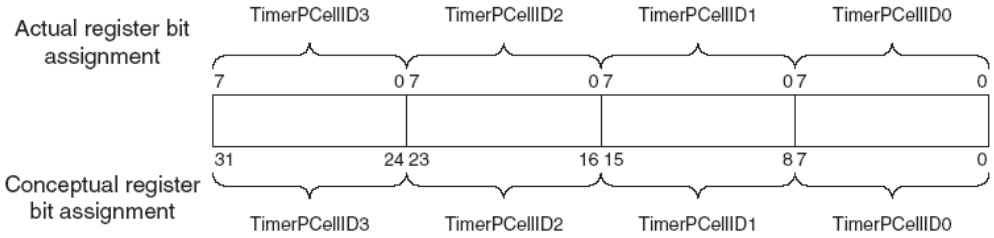


Рисунок 1.307 – Формат регистра TimerPCellID0-3

1.3.13.2.2.10 Регистр управления тестом интеграции. TimerITCR

Этот однобитный регистр используется для активации тестового режима проверки правильности интеграции периферийного блока в систему – далее «теста интеграции». Когда используется этот режим, маскированные сигналы прерывания напрямую контролируются регистром TimerITOP. Формат регистра приведен в таблице 1.758.

Таблица 1.758 - Формат регистра TimerITCR

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервированы, чтение не определено, должны быть записаны нули
[0]	ITEN	ЧТ/ЗП	Разрешение режима «теста интеграции». Когда этот бит установлен в 1, DIT переходит в режим теста интеграции, иначе находится в нормальном режиме.

1.3.13.2.2.11 Регистр управления выходными сигналами. TimerITOP

Когда используется режим теста интеграции, выходные сигналы прерывания непосредственно управляются битами этого регистра. Комбинированное прерывание TIMERINTC формируется по логическому "ИЛИ" между битами, установленными в регистре TimerITOP. Формат регистра приведен в таблице 1.759.

Таблица 1.759 - Формат регистра TimerITOP

Биты	Название	Тип	Выполняемая функция
[31:2]	-	-	Зарезервированы, чтение не определено, должны быть записаны нули
[0]	TIMERINT2	ЗП	Значение с выхода TIMERINT2, в режиме теста интеграции
[1]	TIMERINT1	ЗП	Значение с выхода TIMERINT1, в режиме теста интеграции

1.3.13.2.3 Прерывания

Блок имеет 2 прерывания, каждое из которых подключено к системному контроллеру прерываний (GIC).

1.3.13.3 Сторожевой таймер WATCHDOG

Сторожевой таймер служит индикатором попадания процессора в состояние “зависания” и сброса процессора в случае обнаружения данной ситуации. Управление блоком сторожевого таймера осуществляется по шине AMBA APB версии 2.0. Блок состоит из программируемого 32-разрядного таймера, работающего на тактовой частоте шины APB, схемы управления режимами работы таймера и схемы формирования контрольных и управляющих сигналов.

1.3.13.3.1 Организация работы сторожевого таймера

После снятия сигнала системного сброса, блок находится в выключенном состоянии. Если возникла необходимость воспользоваться сторожевым таймером, то программным образом надо задать интервал счета таймера, формируя тем самым квант времени работы сторожевого таймера. После записи интервала счета счетчик начинает декрементироваться.

Когда счетчик достигает нулевого значения, генерируется сигнал прерывания WDOGINT. Данный сигнал прерывания заводится на контроллеры прерываний обоих процессорных систем. Одновременно в счетчик заносится начальное значение интервала счета и декремент счетчика возобновляется. Если до достижения счетчиком нулевого значения, прерывание WDOGINT не будет обработано (сброшено), то будет сформирован сигнал WDOGRST, на основе которого внешним образом можно сгенерировать сигнал системного сброса процессора.

Для исключения случайного сброса процессора вследствие неправильной записи в регистры блока сторожевого таймера предусмотрена возможность программно заблокировать эти регистры для записи.

1.3.13.3.2 Программно доступные регистры блока сторожевого таймера

Программно доступные регистры блока сторожевого таймера WDT расположены в области памяти периферийных устройств REG_CCPU, и имеют начальный адрес WDT Base = 0x000A_8000 hex и общий размер 4 КБайт. Спецификация регистров блока представлена в таблице 1.760.

Таблица 1.760 - Спецификация регистров блока WDT

Адрес (байтовый доступ)	Тип	Раз-ряд-ность	Начальное значение	Имя	Описание
0x000A_8000 hex	ЧТ/ЗП	32	0xFFFFFFFF	WdogLoad	Регистр загрузки. WdogLoad
0x000A_8004 hex	ЧТ	32	0xFFFFFFFF	WdogValue	Регистр значения. WdogValue
0x000A_8008 hex	ЧТ/ЗП	2	0x0	WdogControl	Регистр управления. WdogControl
0x000A_800C hex	ЗП	-	-	WdogIntClr	Регистр сброса прерываний. WdogIntClr
0x000A_8010 hex	ЧТ	1	0x0	WdogRIS	Регистр состояния прерывания до наложения маски. WdogRIS
0x000A_8014 hex	ЧТ	1	0x0	WdogMIS	Регистр состояния прерывания после наложения маски. WdogMIS
0x000A_8018 hex - 0x000A_8BFF hex	-	-	-	-	Зарезервировано
0x000A_8C00 hex	ЧТ/ЗП	32	0x0	WdogLock	Регистр блокировки. WdogLock
0x000A_8C04 hex - 0x000A_8EFF hex	-	-	-	-	Зарезервировано
0x000A_8F00 hex	ЧТ/ЗП	1	0x0	WdogITCR	Регистр управления тестом интеграции. WdogITCR
0x000A_8F04 hex	ЗП	2	0x0	WdogITOP	See Integration Test Output Set Register, WdogITOP
0x000A_8F08 hex - 0x000A_8FDC hex	-	-	-	-	Зарезервировано

Продолжение таблицы 1.760

Адрес (байтовый доступ)	Тип	Раз-ряд-ность	Начальное значение	Имя	Описание
0x000A_8FE0 hex	ЧТ	8	0x05	WdogPeriphID0	Идентификатор периферийного устройства. WdogPeriphID0
0x000A_8FE4 hex 0x3F9	ЧТ	8	0x18	WdogPeriphID1	Идентификатор периферийного устройства. WdogPeriphID1
0x000A_8FE8 hex	ЧТ	8	0x14	WdogPeriphID2	Идентификатор периферийного устройства. WdogPeriphID2
0x000A_8FEC hex	ЧТ	8	0x00	WdogPeriphID3	Идентификатор периферийного устройства. WdogPeriphID3
0x000A_8FF0 hex	ЧТ	8	0x0D	WdogPCellID0	Идентификатор PrimeCell. WdogPCellID0
0x000A_8FF4 hex	ЧТ	8	0xF0	WdogPCellID1	Идентификатор PrimeCell. WdogPCellID1
0x000A_8FF8 hex	ЧТ	8	0x05	WdogPCellID2	Идентификатор PrimeCell. WdogPCellID2
0x000A_8FFC hex	ЧТ	8	0xB1	WdogPCellID3	Идентификатор PrimeCell. WdogPCellID3

WdogLoad - регистр загрузки -32-битный регистр, доступный по записи и по чтению. Данный регистр хранит значение интервала счета, с которого счетчик начинает декрементироваться. Если значение регистра WdogLoad установлено в нуль, то прерывания сторожевым таймером вырабатываются незамедлительно если содержимое регистра отлично от нуля, то это значение загружается в счетчик незамедлительно после записи регистра.

WdogValue – регистр значения счетчика - 32-битный регистр, доступный только на чтение, при чтении этого регистра выдается текущее значение декрементирующего счетчика.

WdogControl - регистр управления - 2-битный регистр, доступный на запись и чтение, который разрешает программному обеспечению (ПО) управлять сторожевым таймером. В таблице 1.761 приведен формат регистра WdogControl.

Таблица 1.761 - Формат регистра WdogControl

Биты	Название	Тип	Выполняемая функция
[31:2]	-	-	Зарезервировано
[1]	RESEN	ЧТ/ЗП	Разрешающий сигнал блока WDOG для сброса выхода, WDOGRST. Работает как маска для сброса выхода. Если установлен высокий уровень, то разрешен сброс, если низкий, то сброс неактивен.
[0]	INTEN	ЧТ/ЗП	Сигнал, разрешающий прерывание события, WDOGINT. Если установлен высокий уровень, то счетчик находится в активном состоянии и разрешены прерывания, если установлен низкий уровень, то счетчик и прерывания неактивны. Нужно перезагрузить счетчик значением, которое хранится в WdogLoad, если прерывание раннее было неактивным, но затем стало разрешенным.

WdogIntClr - регистр сброса прерываний Любая запись в данный регистр, снимает прерывание WDOG, и перезагружает счетчик значением регистра WdogLoad.

WdogRIS - регистр состояния прерывания до наложения маски Данный регистр определяет состояние прерываний, формируемых счетчиком, до наложения маски, Регистр WdogRIS возводится в единицу в момент, когда счетчик сторожевого таймера достигает нуля. В таблице 1.762 приведен формат регистра WdogRIS.

Таблица 1.762 - Формат регистра WdogRIS

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервировано
[0]	WDOGRIS	ЧТ	Состояние прерывания от счетчика до наложения маски.

WdogMIS - регистр состояния прерывания после наложения маски Данный регистр определяет состояние прерывания счетчика после наложения маски. Это состояние формируется логическим функцией «И» между битом WDOGRIS и битом INTEN регистра управления. В таблице 5.70 приведен формат регистра WdogMIS.

Таблица 1.763 - Формат регистра WdogMIS

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервировано
[0]	WDOGMIS	ЧТ	Состояние прерывания от счетчика после наложения маски.

WdogLock - регистр блокировки записи. Запись любого значения кроме 0x1ACCE551 в данный регистр блокирует программную запись во все другие регистры блока. Запись значения 0x1ACCE551 разблокирует программную запись во все регистры блока. Таким образом, можно защитить регистры WDOG от некорректно работающего программного обеспечения. Чтение из этого регистра возвращает состояние блокировки:

0 – доступ по записи разрешен (нет блокировки);

1 – доступ по записи запрещен (заблокирован).

В таблице 5.71 приведен формат регистра WdogLock.

Таблица 1.764 - Формат регистра WdogLock

Биты	Название	Тип	Выполняемая функция
[31:0]	WDOGLOCK	ЧТ/ЗП	Запись значения 0x1ACCE551 в этот регистр разрешает запись во все регистры. Запись любого другого значения делает невозможным доступ по записи ко всем регистрам. Чтение возвращает состояние блокировки: 0x00000000 – запись во все регистры разрешена 0x00000001 – запись во все регистры запрещена

WdogPeriphID0-3 – Идентификатор периферийного устройства Регистр TimerPeriphID0-3 представляет четыре 8-битных регистра, предназначенных только для чтения, которые охватывают адресное пространство от 0xFE0 до0xFEC. Регистры могут быть концептуально представлены как 32-битный регистр. В таблице 1.765 приведен формат этого регистра.

Таблица 1.765 - Формат регистра WdogPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для DIT 0x805
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока. Номер ревизии начинается с нуля
Configuration[31:24]	Является вариантом конфигурации периферии. Для DIT 0x0

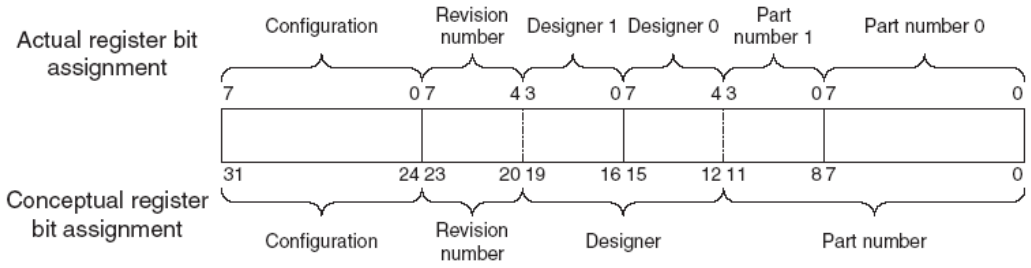


Рисунок 1.308 - Формат регистра WdogPeriphID0-3

Отметим, что при программировании важно помнить, что периферийные блоки занимают 4 Кбайт адресного пространства. Обращение к периферийным регистрам должно быть 32-битным, независимо от их реальной разрядности.

WdogPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, программно доступные только по чтению, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. Регистр WdogPCellID установлен в 0xB105F00D. На рисунке 1.309 приведен формат этого регистра.

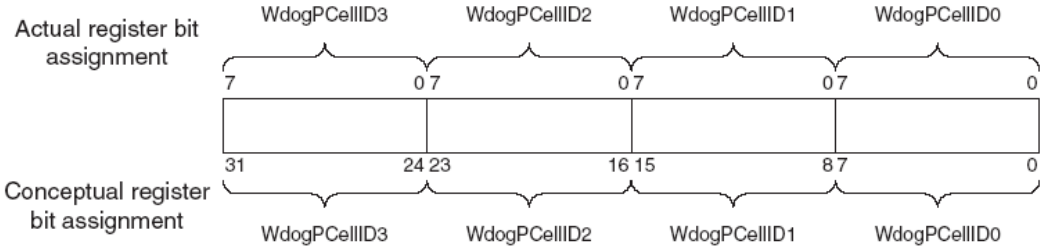


Рисунок 1.309 - Формат регистра WdogPCellID0-3

WdogITCR - регистр управления тестом интеграции. Данный однобитный регистр используется для активации тестового режима проверки правильности интеграции периферийного блока в систему – далее «теста интеграции». Когда используется этот режим, прерывание WDOGINT и сигнал сброса WDOGRST контролируются регистром WdogITOP. Формат регистра приведен в таблице 1.766.

Таблица 1.766- Формат регистра WdogITCR

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервированы
[0]	ITEN	ЧТ/ЗП	Разрешение режима «теста интеграции». Когда этот бит установлен в 1, WDOG переходит в режим теста интеграции, иначе находится в нормальном режиме.

WdogITOP - регистр управления выходными сигналами В режиме теста интеграции, на выходные сигналы прерывания подаются соответствующие разряды данного регистра. Формат регистра приведен в таблице 1.767.

Таблица 1.767 - Формат регистра WdogITOP

Биты	Название	Тип	Выполняемая функция
[31:2]	-	-	Зарезервированы, чтение не определено, должны быть записаны нули
[0]	WDOGINT	ЗП	Значение выхода WDOGINT, в режиме теста интеграции
[1]	WDOGRST	ЗП	Значение выхода WDOGRST, в режиме теста интеграции

1.3.13.4 Система синхронизации и сброса

1.3.13.4.1 Структура тактовых сигналов

Микросхема имеет внутреннюю систему синхронизации, основанную на блоках PLL. На системном уровне имеется 4 блока PLL:

PLL_CPU – служит для формирования тактовых сигналов процессорных ядер PPC и NMC3, и тактовых сигналов коммутационной среды.

PLL_DDR – служит для формирования тактовых сигналов интерфейсных подсистем, и опорных синхросигналов для DDR_PHY.

PLL_VDU – служит для формирования пиксельных частот устройств VDU0 и VDU1, и тактового сигнала CODEC.

PLL_AUD – служит для формирования базовой аудио частоты для устройств I2S и SPDIF

Кроме того, блоки PLL имеются в блоках физического уровня(PHY) интерфейсов DDR3, USB и SERDES, их описание приведено в соответствующих главах. На рисунках 1.310, 1.311, 1.312 и 1.313 представлены схемы генерации тактовых сигналов на основе системных PLL:

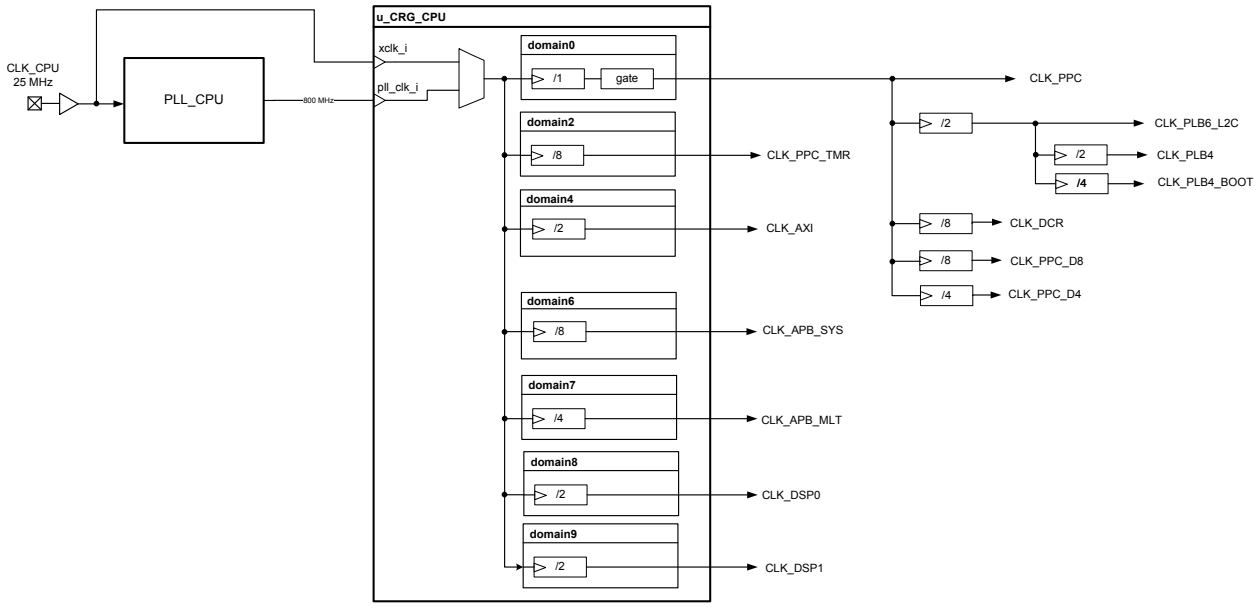


Рисунок 1.310 – схема тактовых сигналов формируемых PLL_CPU

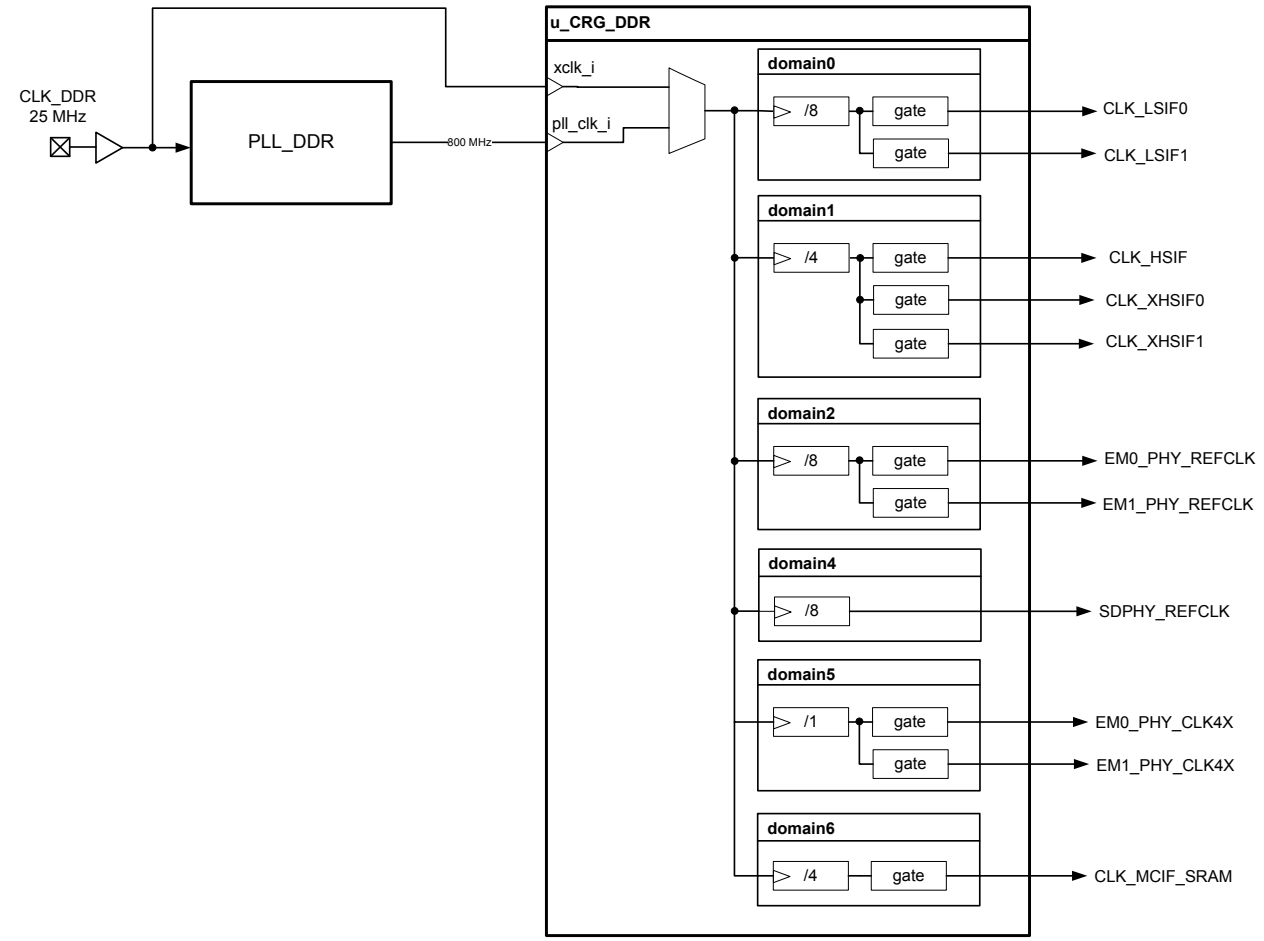


Рисунок 1.311 – схема тактовых сигналов формируемых PLL_DDR

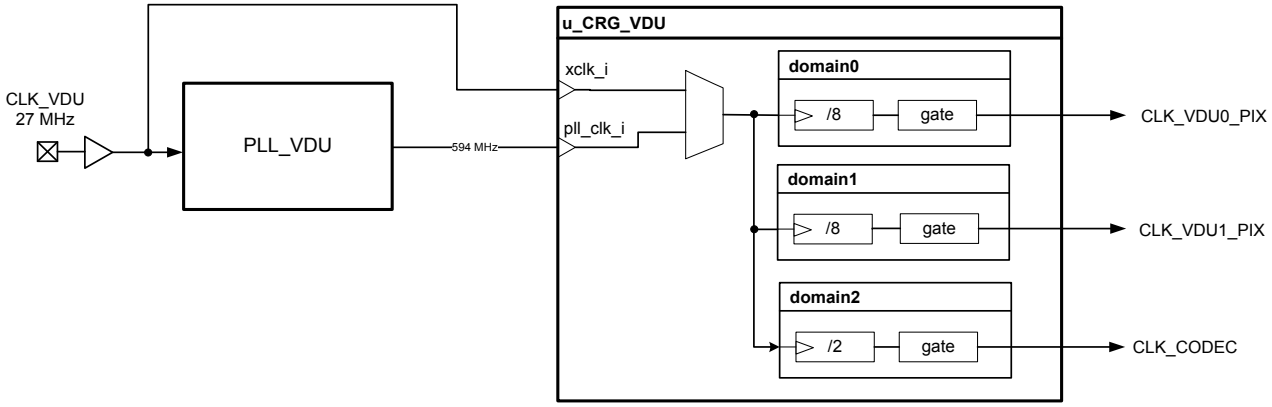


Рисунок 1.312 – схема тактовых сигналов формируемых PLL_VDU

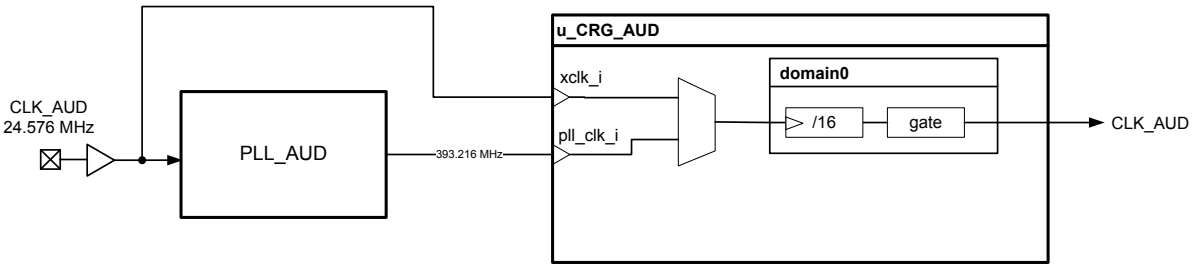


Рисунок 1.313 – схема тактовых сигналов формируемых PLL_AUD

1.3.13.4.2 Описание системных тактовых сигналов

Описание системных тактовых сигналов, формируемых блоками PLL_CPU, PLL_DDR, PLL_VDU и PLL_AUD, приведено в таблице 1.768.

Таблица 1.768 – описание системных тактовых сигналов микросхемы

Название	Описание	Максимальная частота при наихудших условиях, МГц	Источник
CLK_PPC	Тактовый сигнал процессорных ядер PPC	800	Программируемый делитель в CRG_CPU
CLK_PLB6_L2C	Тактовый сигнал контроллеров L2Cache и шины PLB6	400	Фиксированный делитель /2 от CLK_PPC
CLK_PLB4	Тактовый сигнал шины PLB4	200	Фиксированный делитель /2 от CLK_PLB6_L2C
CLK_PLB4_BOOT	Тактовый сигнал шины PLB4_BOOT	100	Фиксированный делитель /4 от CLK_PLB6_L2C
CLK_DCR	Тактовый сигнал конфигурационной шины DCR	100	Фиксированный делитель /8 от CLK_PPC
CLK_PPC_D4	Тактовый сигнал отладочной подсистемы ядер PPC	200	Фиксированный делитель /4 от CLK_PPC
CLK_PPC_D8	Тактовый сигнал трассировки инструкций ядер PPC	100	Фиксированный делитель /8 от CLK_PPC
CLK_PPC_TMR	Тактовый сигнал встроенных таймеров в ядрах PPC	100	Программируемый делитель в CRG_CPU
CLK_AXI	Тактовый сигнал коммутационной среды AXI	400	Программируемый делитель в CRG_CPU
CLK_APB_SYS	Тактовый сигнал системной конфигурационной шины APB	100	Программируемый делитель в CRG_CPU
CLK_APB_MLT	Тактовый сигнал мультимедийной конфигурационной шины APB	200	Программируемый делитель в CRG_CPU
CLK_DSP0	Тактовый сигнал подсистемы DSP0	400	Программируемый делитель в CRG_CPU

Продолжение таблицы 1.768

Название	Описание	Максимальная частота при наихудших условиях, МГц	Источник
CLK_DSP1	Тактовый сигнал подсистемы DSP1	400	Программируемый делитель в CRG_CPU
CLK_LSIF0	Тактовый сигнал интерфейсной подсистемы LSIF0	100	Программируемый делитель в CRG_DDR
CLK_LSIF1	Тактовый сигнал интерфейсной подсистемы LSIF1	100	Программируемый делитель в CRG_DDR
CLK_HSIF	Тактовый сигнал интерфейсной подсистемы HSIF	200	Программируемый делитель в CRG_DDR
CLK_XHSIF0	Тактовый сигнал интерфейсной подсистемы XHSIF0	200	Программируемый делитель в CRG_DDR
CLK_XHSIF1	Тактовый сигнал интерфейсной подсистемы XHSIF1	200	Программируемый делитель в CRG_DDR
EM0_PHY_REFCLK	Опорная частота PLL в DDR0_PHY	100	Программируемый делитель в CRG_DDR
EM1_PHY_REFCLK	Опорная частота PLL в DDR1_PHY	100	Программируемый делитель в CRG_DDR
SDPHY_REFCLK	Опорная частота SERDES_PHY	100	Программируемый делитель в CRG_DDR
EM0_PHY_CLK4X	Тактовый сигнал интерфейса DDR_EM0 (режим работы от системной PLL)	800	Программируемый делитель в CRG_DDR
EM1_PHY_CLK4X	Тактовый сигнал интерфейса DDR_EM1 (режим работы от системной PLL)	800	Программируемый делитель в CRG_DDR
CLK_MCIF_SRAM	Тактовая частота контроллера SRAM_MCIF	200	Программируемый делитель в CRG_DDR
CLK_VDU0_PIX	Пиксельная частота видеоконтроллера VDU0	74,25	Программируемый делитель в CRG_VDU
CLK_VDU1_PIX	Пиксельная частота видеоконтроллера VDU1	74,25	Программируемый делитель в CRG_VDU
CLK_CODEC	Системная частота CODEC	297	Программируемый делитель в CRG_VDU
CLK_AUD	Базовая аудио частота передатчиков I2S и SPDIF	24,576	Программируемый делитель в CRG_AUD

1.3.13.4.3 Характеристики блоков PLL

Системные блоки PLL идентичны друг другу, и имеют структуру, изображенную на рисунке 1.769.

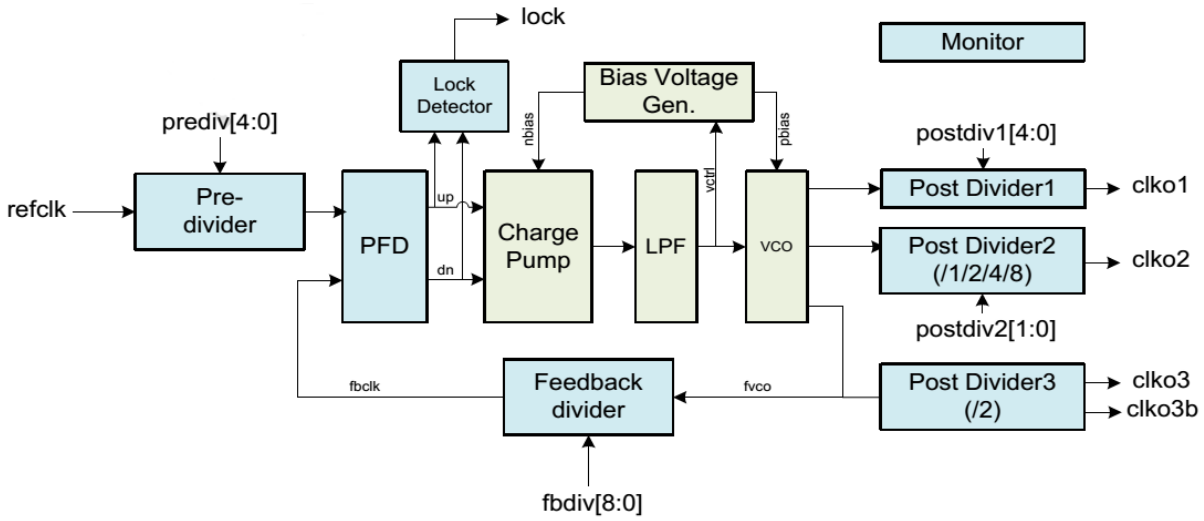


Рисунок 1.769

Во всех системных PLL используется выход clko2. Как видно из структуры, PLL имеет три управляемых делителя, определяющих выходную частоту:

входной делитель(prediv) – возможные коэффициенты деления 1..31

делитель обратной связи(fbdiv) – возможные коэффициенты деления 8...511

выходной делитель(postdiv2) – возможные коэффициенты деления 1,2,4,8

Частота на выходе блока PLL определяется по формуле:

$$F_{PLL} = \frac{F_{REF} \times FBDIV[8:0]}{PREDIV[4:0] \times POSTDIV2[1:0]}$$

Каждый блок PLL управляется своим контроллером CRG. Основные параметры системных блоков PLL приведены в таблице 1.770.

Таблица 1.770 – характеристики системных блоков PLL

Обозначение параметра	Описание параметра	Не менее	Не более
FREF	Частота на входе входного делителя	10 МГц	500 МГц
FPREDIV	Частота на выходе входного делителя	1 МГц	100 МГц
FVCO	Частота на выходе VCO	300	3000 МГц
TLOCK	Время стабилизации		1000 тактов частоты после входного делителя

1.3.13.4.4 Структура сигналов сброса

Обобщенная схема сигналов сброса показан на рисунке 1.314.

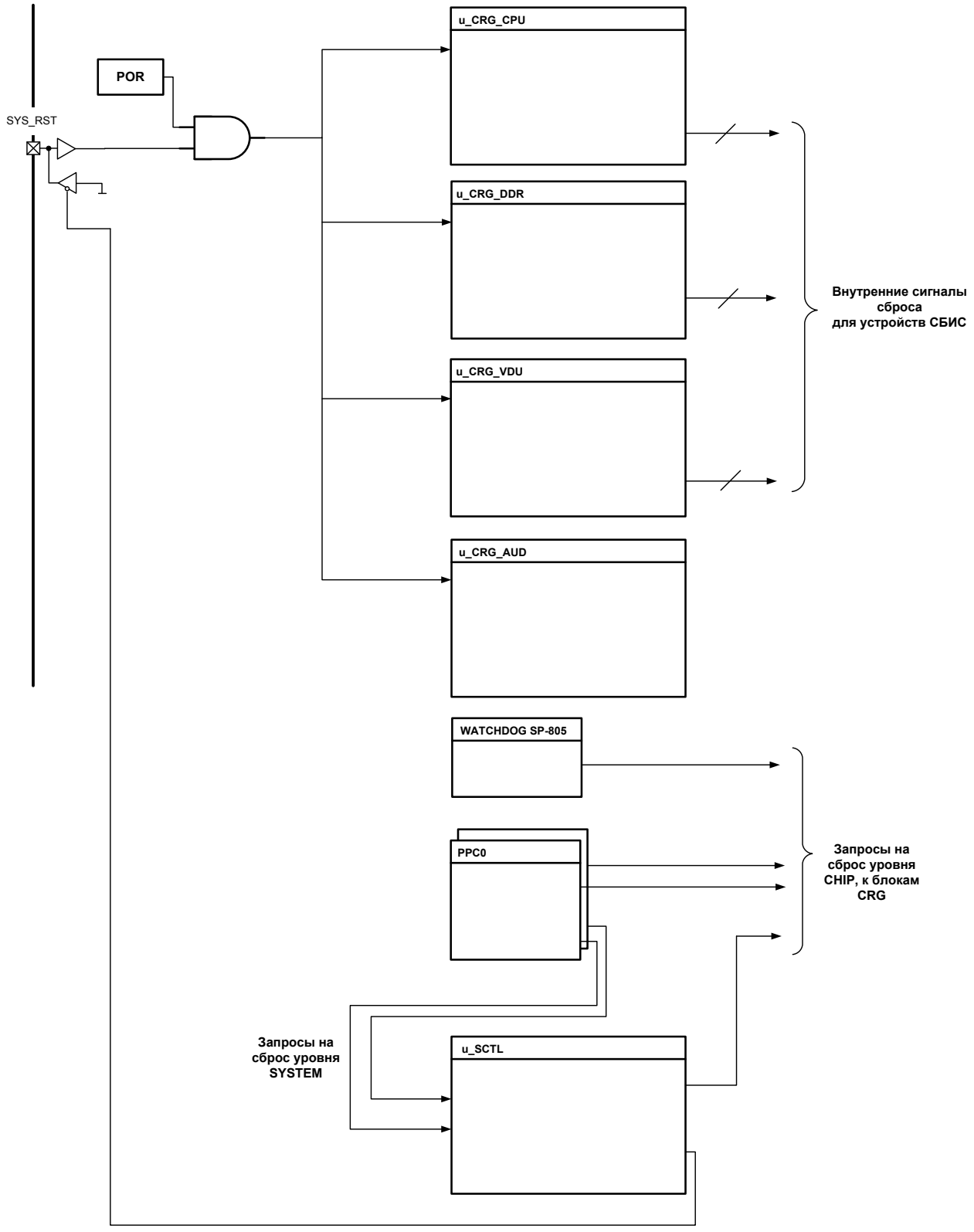


Рисунок 1.314 –Схема сигналов сброса микросхемы

Микросхема имеет следующие источники сброса:

Внутренняя схема сброса по включению питания (POR) – удерживает сигналы сброса в активном состоянии пока напряжения питания 1,0 В и 1,8 В не достигнут определенных пороговых значений.

Двунаправленный вывод SYS_RST. Низкий уровень на этом выводе приводит к сбросу микросхемы. Данный вывод может быть установлен в низкий уровень как снаружи, так и изнутри микросхемы.

Запросы на сброс уровня CHIP – приводят к сбросу микросхемы, не задействуя вывод SYS_RST. К таким запросам относятся: запросы CHIPRESETREQ от ядер PPC, запрос на сброс от сторожевого таймера WDT(ARM SP-805), запрос от системного контроллера SCTL.

Запросы на сброс уровня SYSTEM – приводят к сбросу микросхемы и внешних компонентов, путем установки вывода SYS_RST в низкий уровень. К таким запросам относятся: запросы SYSTEMRESETREQ от ядер PPC, запрос от системного контроллера.

1.3.13.4.5 Программная модель контроллеров CRG

1.3.13.4.5.1 Программная модель контроллера CRG_CPU

Таблица 1.770 – список регистров контроллера CRG_CPU

Адрес	Название	Описание
0x000	PLL_STATE	Регистр состояния блока умножения частоты (PLL)
0x004	PLL_CTRL	Управление PLL
0x008	-	Зарезервировано
0x00C	WR_LOCK	Регистр блокировки записи в другие регистры
0x010	RST_MON	Регистр отслеживания источника последнего сброса
0x014	RST_CTRL	Настройка поведения отработки входных сигналов сброса
0x018	RST_EN	Регистр программных сбросов
0x01C	UPD_CK	Активация новых значений регистров CKDIVx и CKENx
0x020	CKALIGN	Управление выравниваем синхросигналов по фазе
0x024	CKDIVSTAT	Признаки изменения регистров CKDIVMODEx
0x028	-	Зарезервировано
0x02C	-	Зарезервировано
0x030	PLL_PRDIV	Коэффициент входного делителя PLL
0x034	PLL_FBDIV	Коэффициент делителя обратной связи PLL
0x038	PLL_PSDIV	Коэффициент выходного делителя PLL
…	-	Зарезервировано
0x100	CKDIVMODE_CPU	Коэффициент деления центральной процессорной подсистемы
0x104	CKEN_CPU	Регистр разрешения тактирования ядер PPC470S
0x108	-	Зарезервировано
0x10C	-	Зарезервировано
0x110	CKDIVMODE_L2C_PLB6	Коэффициент деления для контроллеров L2C_0 и L2C_1, а также шины PLB6
0x114	CKEN_L2C_PLB6	Регистр разрешения тактирования контроллеров L2C_0 и L2C_1, а также PLB6-интерфейсов подсистем DDR3
0x118	-	зарезервировано
0x11C	-	зарезервировано
0x120	CKDIVMODE_PPC_TMR	Коэффициент деления для встроенных таймеров ядер PPC470S
0x124	CKEN_PPC_TMR	Регистр разрешения тактирования таймеров ядер PPC470S
…	-	зарезервировано
0x140	CKDIVMODE_AXI	Коэффициент деления коммутационной среды AXI
…	-	зарезервировано
0x160	CKDIVMODE_APB_SYS	Коэффициент деления для шины APB (системная)
0x164	-	зарезервировано
0x170	CKDIVMODE_APB_MLT	Коэффициент деления для шины APB (мультимедиа)
…	-	зарезервировано
0x180	CKDIVMODE_DSP0	Коэффициент деления для подсистемы DSP0

Таблица 1.770 – список регистров контроллера CRG_CPU

Адрес	Название	Описание
0x184	CKEN_DSP0	Регистр разрешения тактирования подсистемы DSP0
0x188	-	зарезервировано
0x18C	-	зарезервировано
0x190	CKDIVMODE_DSP1	Коэффициент деления для подсистемы DSP1
0x194	CKEN_DSP1	Регистр разрешения тактирования подсистемы DSP1

Таблица 1.771 - формат регистра PLL_STATE (0x000) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	PLL_RDY	1'b0	R	Флаг выхода в рабочий режим PLL_CPU 0 – PLL нестабильна 1 – PLL стабильна

Таблица 1.772 - формат регистра PLL_CTRL (0x004) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	PLL_BYPASS	1'b0	R/W	Выбор опорного синхросигнала 0 – выбран синхросигнал с выхода PLL_CPU 1 – выбран внешний синхросигнал CLK_CPU
[3:2]	-	-	R0	зарезервировано
[1]	PLL_ENABLE	1'b1	R/W	Включение/выключение PLL
[0]	PLL_RESTART	1'b0	R/W	Запись 1 запускает процедуру перезапуска блока PLL. Бит сбрасывается в 0 автоматически после выполнения процедуры перезапуска PLL.

Таблица 1.773 - формат регистра WR_LOCK (0x00C) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:0]	WR_LOCK	32'h0000_0001	R/W	Запись значения 0x1ACCE551 разрешает запись в другие регистры. Запись любого другого значения запрещает запись в другие регистры. Чтение возвращает статус блокировки: 0x0000_0000 – запись в регистры разрешена 0x0000_0001 – запись в регистры запрещена

Таблица 1.774 - формат регистра RST_MON (0x010) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:17]	-	-	R0	Зарезервировано
[16]	XRST_MON	1'b1	R	Бит отслеживает сигнал внешнего системного сброса вызванного активацией сигнала xrstn_i: 1 – внешний сброс был активирован 0 – внешний сброс не был активирован
[15:4]	-	-	R0	зарезервировано
[3:0]	RSTREQ_MON	4'h0	R	Каждый бит поля отслеживает соответствующий запрос на сброс rstreq_i 1 – последний сброс был по данному запросу 0 – последний сброс был по другой причине

Продолжение таблицы 1.775

Биты	Название	Сброс	Доступ	Описание
[31:20]	-	-	R0	зарезервировано
[19:16]	RSTREQ_MODE	4’h0		Каждый бит поля определяет поведение при приходе соответствующего сигнала запроса на сброс rstreq_i 1'b0 – сброс только внешних блоков. PLL не перезапускается, значения конфигурационных регистров CRG сохраняются. 1'b1 – сброс внешних блоков, перезапуск PLL, сброс конфигурационных регистров CRG.
[15:12]	-	-	R0	зарезервировано
[11:0]	RST_DUR	12'h0C8	R/W	Длительность внешних сигналов сброса, в периодах тактового сигнала на выходе PLL

Активный уровень битов сброса – низкий, т.е. обнуление бита активирует соответствующий сигнал сброса, установка в единицу – снимает сигнал сброса.

Таблица 1.776 - Формат регистра RST_EN (0x018) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:14]	-	-	R0	зарезервировано
[13]	HLB_AUDIO_RESET	1’b1	R/W	Сброс подсистемы HLB_AUDIO
[12]	HLB_VIDEO1_RESET	1’b1	R/W	Сброс подсистемы HLB_VIDEO1
[11]	HLB_VIDEO0_RESET	1’b1	R/W	Сброс подсистемы HLB_VIDEO0
[10]		1’b1	R/W	зарезервировано
[9]	DSP1_NMC1_RESET	1’b1	R/W	Сброс ядра NMC1 подсистемы DSP1
[8]	DSP1_NMC0_RESET	1’b1	R/W	Сброс ядра NMC0 подсистемы DSP1
[7]	DSP1_SYS_RESET	1’b1	R/W	Сброс системных блоков подсистемы DSP1
[6]	DSP0_NMC1_RESET	1’b1	R/W	Сброс ядра NMC1 подсистемы DSP0
[5]	DSP0_NMC0_RESET	1’b1	R/W	Сброс ядра NMC0 подсистемы DSP0
[4]	DSP0_SYS_RESET	1’b1	R/W	Сброс системных блоков подсистемы DSP0
[3]	PPC1_L2C_RESET	SYS_BOOT0| ~SYS_BOOT1	R/W	Сброс ядра PPC1 и контроллера L2C1
[2]	PPC0_L2C_RESET	~(SYS_BOOT0& SYS_BOOT1);	R/W	Сброс ядра PPC0 и контроллера L2C0
[1]	-	-	R0	зарезервировано
[0]	-	-	R0	зарезервировано

Таблица 1.777 - Формат регистра UPD_CK (0x01C) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	UPD_CKEN	1'b0	R/W	Применение разрешений выдачи тактовых сигналов. После настройки регистров CKENx необходимо установить этот бит в 1, чтобы новые значения этих регистров вступили в силу. Бит сбрасывается автоматически после применения изменений.
[3:1]	-	-	R0	зарезервировано
[0]	UPD_CKDIV	1'b0	R/W	Применение коэффициентов деления доменов синхросигналов. После настройки регистров CKDIVMODEx необходимо установить этот бит в 1, чтобы новые значения этих регистров вступили в силу. Бит сбрасывается автоматически после смены режима делителей.

Таблица 1.778 - формат регистра CKALIGN (0x020) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:10]	-	-	R0	зарезервировано
[9:0]	CKALIGN	10'd0	R/W	Управление выравниванием тактовых сигналов по фазе при смене коэффициентов делителей: CKALIGN[i]=1 – тактовые сигналы домена i будут выровнены по фазе с тактовыми сигналами тех доменов, для которых CKALIGN также равен 1 CKALIGN[i]=0 – тактовые сигналы домена i не будут выровнены по фазе с остальными

Таблица 1.779 - формат регистра CKDIVSTAT (0x024) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:10]	-	-	R0	зарезервировано
[9:0]	CKDIVSTAT	0	R	Каждый бит поля отражает статус изменения соответствующего регистра CKDIVMODEx: 0 – значение регистра CKDIVMODEx не менялось 1 – значение регистра CKDIVMODEx было изменено, но не вступило в силу

Таблица 1.780 - формат регистра PLL_PRDIV (0x030) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4:0]	PLL_PRDIV	5'd1	R/W	Коэффициент входного делителя PLL , корректными являются значения 1..31

Таблица 1.781 - формат регистра PLL_FBDIV (0x034) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:9]	-	-	R0	зарезервировано
[8:0]	PLL_FBDIV	9'd64	R/W	Коэффициент делителя обратной связи PLL, корректными являются значения 8..511

Таблица 1.89382 - формат регистра PLL_PSDIV (0x038) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:2]	-	-	R0	зарезервировано
[1:0]	PLL_PSDIV	2'b01	R/W	Коэффициент выходного делителя PLL: 2'b00: 1 2'b01: 2 2'b10: 4 2'b11: 8

Таблица 1.783 - Формат регистра CKDIVMODE_CPU(0x100) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h00	R/W	Коэффициент делителя для базового синхросигнала центральной процессорной подсистемы Fclk=Fpll/(DIVMODE+1)

Таблица 1.784 - Формат регистра CKEN_CPU (0x104) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:2]	-	-	R0	зарезервировано
[1]	PPC1_CLKEN	SYS_BOOT0 | ~SYS_BOOT1	R/W	Разрешение тактирования ядра PPC1: 1 – разрешено, 0 – запрещено
[0]	PPC0_CLKEN	~(SYS_BOOT0& SYS_BOOT1);	R/W	Разрешение тактирования ядра PPC0: 1 – разрешено, 0 - запрещено

Таблица 1.785 - формат регистра CKEN_L2C_PLB6 (0x114) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7]	DDR3_EM1_PLBCLKEN	1'b1	R/W	Разрешение тактирования интерфейса PLB6 подсистемы DDR3_EM1
[6]	DDR3_EM0_PLBCLKEN	1'b1	R/W	Разрешение тактирования интерфейса PLB6 подсистемы DDR3_EM0
[5]	L2C1_DCRCLKEN	SYS_BOOT0 | ~SYS_BOOT1	R/W	Разрешение тактирования интерфейса DCR контроллера L2C1: 1 – разрешено, 0 – запрещено
[4]	L2C1_PLBCLKEN	SYS_BOOT0 | ~SYS_BOOT1	R/W	Разрешение тактирования интерфейса PLB6 контроллера L2C1: 1 – разрешено, 0 – запрещено
[3]	L2C1_CLKEN	SYS_BOOT0 | ~SYS_BOOT1	R/W	Разрешение тактирования ядра контроллера L2C1 : 1 – разрешено, 0 - запрещено
[2]	L2C0_DCRCLKEN	~(SYS_BOOT0& SYS_BOOT1);	R/W	Разрешение тактирования интерфейса DCR контроллера L2C0: 1 – разрешено, 0 - запрещено
[1]	L2C0_PLBCLKEN	~(SYS_BOOT0& SYS_BOOT1);	R/W	Разрешение тактирования интерфейса PLB6 контроллера L2C0: 1 – разрешено, 0 - запрещено
[0]	L2C0_CLKEN	~(SYS_BOOT0& SYS_BOOT1);	R/W	Разрешение тактирования ядра контроллера L2C0 : 1 – разрешено, 0 - запрещено

Таблица 1.786 - Формат регистра CKDIVMODE_PPC_TMR (0x120) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h07	R/W	Коэффициент делителя для таймеров ядер PPC470S Fclk=Fpll/(DIVMODE+1) Другие значения запрещены и не должны устанавливаться.

Таблица 1.787 - формат регистра CKEN_PPC_TMR (0x124) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:2]	-	-	R0	зарезервировано
[1]	PPC1_TMR_CLKEN	1'b1	R/W	Разрешение тактирования таймеров ядра PPC1: 1 – разрешено, 0 – запрещено
[0]	PPC0_TMR_CLKEN	1'b1	R/W	Разрешение тактирования таймеров ядра PPC0: 1 – разрешено, 0 - запрещено

Таблица 1.788 - формат регистра CKDIVMODE_AXI (0x140) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h01	R/W	Коэффициент делителя для коммутационной среды AXI Fclk=Fpll/(DIVMODE+1)

Таблица 1.789 - Формат регистра CKDIVMODE_APB _SYS (0x160) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	Зарезервировано
[7:0]	DIVMODE[7:0]	8'h07	R/W	Коэффициент делителя для шины APB_SYS Fclk=Fpll/(DIVMODE+1)

Таблица 1.790 - Формат регистра CKDIVMODE_APB _MLT (0x170) в CRG_CPU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h03	R/W	Коэффициент делителя для шины APB_MLT Fclk=Fpll/(DIVMODE+1)

Таблица 1.791 - формат регистра CKDIVMODE_DSP0 (0x180)

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h01	R/W	Коэффициент делителя для подсистемы DSP0 Fclk=Fpll/(DIVMODE+1)

Таблица 1.792 - формат регистра CKEN_DSP0 (0x184)

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	DSP0_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы DSP0

Таблица 1.793 - формат регистра CKDIVMODE_DSP1 (0x190)

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h01	R/W	Коэффициент делителя для подсистемы DSP1 Fclk=Fpll/(DIVMODE+1)

Таблица 1.794 - формат регистра CKEN_DSP1 (0x194)

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	DSP1_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы DSP1

1.3.13.4.5.2 Программная модель контроллера CRG_DDR

Таблица 1.795– список регистров контроллера CRG_DDR

Адрес	Название	Описание
0x000	PLL_STATE	Регистр состояния блока умножения частоты (PLL)
0x004	PLL_CTRL	Управление PLL
0x008	-	Зарезервировано
0x00C	WR_LOCK	Регистр блокировки записи в другие регистры
0x010	-	Зарезервировано
0x014	RST_CTRL	Настройка поведения отработки входных сигналов сброса
0x018	RST_EN	Регистр программных сбросов
0x01C	UPD_CK	Активация новых значений регистров CKDIVx и CKENx
0x020	CKALIGN	Управление выравниваем синхросигналов по фазе
0x024	CKDIVSTAT	Признаки изменения регистров CKDIVMODEx
0x028	-	Зарезервировано
0x02C	-	Зарезервировано
0x030	PLL_PRDIV	Коэффициент входного делителя PLL
0x034	PLL_FBDIV	Коэффициент делителя обратной связи PLL
0x038	PLL_PSDIV	Коэффициент выходного делителя PLL
…	-	Зарезервировано
0x100	CKDIVMODE_LSIF	Коэффициент деления для подсистем LSIF0, LSIF1
0x104	CKEN_LSIF	Регистр разрешения тактирования подсистем LSIF0, LSIF1
0x108	-	Зарезервировано
0x10C	-	Зарезервировано
0x110	CKDIVMODE_HSIF	Коэффициент деления для подсистем HSIF, XHSIF0, XHSIF1
0x114	CKEN_HSIF	Регистр разрешения тактирования подсистем HSIF, XHSIF0, XHSIF1
0x118	-	зарезервировано
0x11C	-	зарезервировано
0x120	CKDIVMODE_DDR3_REFCLK	Коэффициент деления для входной частоты внутренней PLL DDR3 PHY
0x124	CKEN_DDR3_REFCLK	Регистр разрешения тактирования REFCLK DDR3 PHY
0x128	-	зарезервировано
0x12C	-	зарезервировано
…	-	зарезервировано
0x140	CKDIVMODE_SDPHY	Коэффициент деления опорной частоты блоков SERDES PHY
0x144	-	зарезервировано
0x148	-	зарезервировано
0x14C	-	зарезервировано
0x150	CKDIVMODE_DDR3_4X	Коэффициент деления для 4x-синхросигнала DDR3 PHY
0x154	CKEN_DDR3_4X	Регистр разрешения тактирования 4x DDR3 PHY
0x158	-	зарезервировано
0x15C	-	зарезервировано
0x160	CKDIVMODE_MCIF_SRAM	Коэффициент деления для контроллера SRAM MCIF
0x164	CKEN_MCIF_SRAM	Регистр разрешения тактирования для контроллера SRAM MCIF
0x168	-	зарезервировано
0x16C	-	зарезервировано

Таблица 1.796 - формат регистра PLL_STATE (0x000) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	Зарезервировано
[0]	PLL_RDY	1’b0	R	Флаг выхода в рабочий режим PLL_DDR 0 – PLL нестабильна 1 – PLL стабильна

Таблица 1.797 - формат регистра PLL_CTRL (0x004) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	PLL_BYPASS	1’b0	R/W	Выбор опорного синхросигнала 0 – выбран синхросигнал с выхода PLL_DDR 1 – выбран внешний синхросигнал CLK_DDR
[3:2]	-	-	R0	зарезервировано
[1]	PLL_ENABLE	1’b1	R/W	Включение/выключение PLL
[0]	PLL_RESTART	1’b0	R/W	Запись 1 запускает процедуру перезапуска блока PLL. Бит сбрасывается в 0 автоматически после выполнения процедуры перезапуска PLL.

Таблица 1.798 - формат регистра WR_LOCK (0x00C) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:0]	WR_LOCK	32’h0000_0001	R/W	Запись значения 0x1ACCE551 разрешает запись в другие регистры. Запись любого другого значения запрещает запись в другие регистры. Чтение возвращает статус блокировки: 0x0000_0000 – запись в регистры разрешена 0x0000_0001 – запись в регистры запрещена

Таблица 1.799 - Формат регистра RST_CTRL (0x014) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:20]	-	-	R0	зарезервировано
[19:16]	RSTREQ_MODE	4’h0		Каждый бит поля определяет поведение при приходе соответствующего сигнала запроса на сброс rstreq_i 1'b0 – сброс только внешних блоков. PLL не перезапускается, значения конфигурационных регистров CRG сохраняются. 1'b1 – сброс внешних блоков, перезапуск PLL, сброс конфигурационных регистров CRG.
[15:12]	-	-	R0	зарезервировано
[11:0]	RST_DUR	12'h12C	R/W	Длительность внешних сигналов сброса, в периодах тактового сигнала на выходе PLL

Таблица 1.800 - Формат регистра RST_EN (0x018) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:19]	-	-	R0	зарезервировано
[18]	EM1_DCR_RESET	1'b1	R/W	Сброс DCR-интерфейса подсистемы DDR3_EM1
[17]	EM1_PLB6_RESET	1'b1	R/W	Сброс PLB6-интерфейса подсистемы DDR3_EM1
[16]	EM0_DCR_RESET	1'b1	R/W	Сброс DCR-интерфейса подсистемы DDR3_EM0
[15]	EM0_PLB6_RESET	1'b1	R/W	Сброс PLB6-интерфейса подсистемы DDR3_EM0
[14]	XHSIF1_RST	1'b1	R/W	Сброс подсистемы XHSIF1
[13]	XHSIF0_RST	1'b1	R/W	Сброс подсистемы XHSIF0
[12]	HSIF_RST	1'b1	R/W	Сброс подсистемы HSIF
[11]	LSIF1_RST	1'b1	R/W	Сброс подсистемы LSIF1
[10]	LSIF0_RST	1'b1	R/W	Сброс подсистемы LSIF0
[9]	EM1_MC_REF_RESET	1'b1	R/W	Refresh-сброс контроллера DDR3_EM1_MC
[8]	EM1_MC_RECOV_RESET	1'b1	R/W	Recover-сброс контроллера DDR3_EM1_MC
[7]	EM1_MC_RESET	1'b1	R/W	Сброс контроллера DDR3_EM1_MC
[6]	EM1_PHY_SYSRESET	1'b1	R/W	Системный сброс блока DDR3_EM1_PHY
[5]	EM1_PHY_PRESET	1'b1	R/W	Сброс конфигурационных регистров DDR3_EM1_PHY
[4]	EM0_MC_REF_RESET	1'b1	R/W	Refresh-сброс контроллера DDR3_EM0_MC
[3]	EM0_MC_RECOV_RESET	1'b1	R/W	Recover-сброс контроллера DDR3_EM0_MC
[2]	EM0_MC_RESET	1'b1	R/W	Сброс контроллера DDR3_EM0_MC
[1]	EM0_PHY_SYSRESET	1'b1	R/W	Системный сброс блока DDR3_EM0_PHY
[0]	EM0_PHY_PRESET	1'b1	R/W	Сброс конфигурационных регистров DDR3_EM0_PHY

Таблица 1.801 - Формат регистра UPD_CK (0x01C) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	UPD_CKEN	1'b0	R/W	Применение разрешений выдачи тактовых сигналов. После настройки регистров CKENx необходимо установить этот бит в 1, чтобы новые значения этих регистров вступили в силу. Бит сбрасывается автоматически после применения изменений.
[3:1]	-	-	R0	зарезервировано
[0]	UPD_CKDIV	1'b0	R/W	Применение коэффициентов деления доменов синхросигналов. После настройки регистров CKDIVMODEx необходимо установить этот бит в 1, чтобы новые значения этих регистров вступили в силу. Бит сбрасывается автоматически после смены режима делителей.

Таблица 1.802 - формат регистра CKALIGN (0x020) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:10]	-	-	R0	зарезервировано
[5:0]	CKALIGN	6'b100100	R/W	Управление выравниванием тактовых сигналов по фазе при смене коэффициентов делителей: CKALIGN[i]=1 – тактовые сигналы домена i будут выровнены по фазе с тактовыми сигналами тех доменов, для которых CKALIGN также равен 1 CKALIGN[i]=0 – тактовые сигналы домена i не будут выровнены по фазе с остальными

Таблица 1.803 - Формат регистра CKDIVSTAT (0x024) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:10]	-	-	R0	зарезервировано
[5:0]	CKDIVSTAT	0	R	Каждый бит поля отражает статус изменения соответствующего регистра CKDIVMODEx: 0 – значение регистра CKDIVMODEx не менялось 1 – значение регистра CKDIVMODEx было изменено, но не вступило в силу

Таблица 1.804 - Формат регистра PLL_PRDIV (0x030) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4:0]	PLL_PRDIV	0x1	R/W	Коэффициент входного делителя PLL , корректными являются значения 1..31

Таблица 1.805 - формат регистра PLL_FBDIV (0x034) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:9]	-	-	R0	зарезервировано
[8:0]	PLL_FBDIV	0x40	R/W	Коэффициент делителя обратной связи PLL, корректными являются значения 8..511

Таблица 1.806 - Формат регистра PLL_PSDIV (0x038) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:2]	-	-	R0	зарезервировано
[1:0]	PLL_PSDIV	0x1	R/W	Коэффициент выходного делителя: 2’b00: 1 2’b01: 2 2’b10: 4 2’b11: 8

Таблица 1.807 - формат регистра CKDIVMODE_LSIF (0x100) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h07	R/W	Коэффициент делителя для подсистем LSIF0, LSIF1. Fclk=Fpll/(DIVMODE+1)

Таблица 1.808 - формат регистра CKEN_LSIF (0x104) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:2]	-	-	R0	зарезервировано
[1]	LSIF1_CLKEN	1’b1	R/W	Разрешение тактирования подсистемы LSIF1: 1 – разрешено, 0 - запрещено
[0]	LSIF0_CLKEN	1’b1	R/W	Разрешение тактирования подсистемы LSIF0: 1 – разрешено, 0 - запрещено

Таблица 1.809 - формат регистра CKDIVMODE_HSIF (0x110) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h03	R/W	Коэффициент делителя для подсистем HSIF, XHSIF0, XHSIF1. Fclk=Fpll/(DIVMODE+1)

Таблица 1.810 - формат регистра CKEN_HSIF (0x114) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:3]	-	-	R0	зарезервировано
[2]	XHSIF1_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы XHSIF: 1 – разрешено, 0 - запрещено
[1]	XHSIF0_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы XHSIF0: 1 – разрешено, 0 - запрещено
[0]	HSIF_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы HSIF: 1 – разрешено, 0 - запрещено

Таблица 1.811 - формат регистра CKDIVMODE_DDR_REFCLK (0x120) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h03	R/W	Коэффициент делителя для подсистем DDR3_EM0_PHY и DDR3_EM1_PHY(refclk и dfi_clk1x): Fclk=Fpll/(DIVMODE+1)

Таблица 1.812 - Формат регистра CKEN_DDR_REFCLK (0x124) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:4]	-	-	R0	зарезервировано
[3]	DDR3_EM1_MC_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы DDR3_EM1_MC (CLK1X) : 1 – разрешено, 0 - запрещено
[2]	DDR3_EM0_MC_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы DDR3_EM0_MC (CLK1X) : 1 – разрешено, 0 - запрещено
[1]	DDR3_EM1_PHY_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы DDR3_EM1_PHY(refclk и dfi_clk1x): 1 – разрешено, 0 - запрещено
[0]	DDR3_EM0_PHY_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы DDR3_EM0_PHY(refclk и dfi_clk1x): 1 – разрешено, 0 - запрещено

Таблица 1.813 - формат регистра CKDIVMODE_SDPHY (0x140) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h07	R/W	Коэффициент деления для опорной частоты блоков SERDES PHY Fclk=Fpll/(DIVMODE+1)

Таблица 1.814 - формат регистра CKDIVMODE_DDR_4X (0x150) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h00	R/W	Коэффициент делителя для подсистем DDR3_EM0_PHY и DDR3_EM1_PHY (dfi_clk4x): Fclk=Fpll/(DIVMODE+1)

Таблица 1.815 - формат регистра CKEN_DDR_4X (0x154) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:2]	-	-	R0	зарезервировано
[1]	DDR3_EM1_PHY_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы DDR3_EM1_PHY(dfi_clk4x): 1 – разрешено, 0 – запрещено
[0]	DDR3_EM0_PHY_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы DDR3_EM0_PHY(dfi_clk4x): 1 – разрешено, 0 - запрещено

Таблица 1.816 - формат регистра CKDIVMODE_MCIF_SRAM (0x160) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h03	R/W	Коэффициент делителя для контроллера MCIF_ SRAM

Таблица 1.817 - формат регистра CKEN_DDR_MCIF_SRAM (0x164) в CRG_DDR

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	MCIF_SRAM_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы контролера MCIF_SRAM

1.3.13.4.5.3 Программная модель контроллера CRG_VDU

Таблица 1.818 – список регистров контроллера CRG_VDU

Адрес	Название	Описание
0x000	PLL_STATE	Регистр состояния блока умножения частоты (PLL)
0x004	PLL_CTRL	Управление PLL
0x008	-	Зарезервировано
0x00C	WR_LOCK	Регистр блокировки записи в другие регистры
0x010	-	Зарезервировано
0x014	RST_CTRL	Настройка поведения отработки входных сигналов сброса
0x018	RST_EN	Регистр программных сбросов
0x01C	UPD_CK	Активация новых значений регистров CKDIVx и CKENx
0x020	CKALIGN	Управление выравниваем синхросигналов по фазе
0x024	CKDIVSTAT	Признаки изменения регистров CKDIVMODEx
0x028	-	Зарезервировано
0x02C	-	Зарезервировано
0x030	PLL_PRDIV	Коэффициент входного делителя PLL
0x034	PLL_FBDIV	Коэффициент делителя обратной связи PLL
0x038	PLL_PSDIV	Коэффициент выходного делителя PLL
…	-	Зарезервировано
0x100	CKDIVMODE_VOUT0	Коэффициент деления выходного видеоинтерфейса VOUT0
0x104	CKEN_VOUT0	Отключение/включение синхросигнала интерфейса VOUT0
0x108	-	Зарезервировано
0x10C	-	Зарезервировано
0x110	CKDIVMODE_VOUT1	Коэффициент деления выходного видеоинтерфейса VOUT1
0x114	CKEN_VOUT1	Отключение/включение синхросигнала интерфейса VOUT1
0x118	-	Зарезервировано
0x11C	-	Зарезервировано
0x120	CKDIVMODE_CODEC	Коэффициент деления для подсистемы HLB_CODEC
0x124	CKEN_CODEC	Отключение/включение синхросигнала HLB_CODEC
0x128	-	зарезервировано
0x12C	-	зарезервировано

Таблица 1.819 - Формат регистра PLL_STATE (0x000) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	PLL_RDY	1’b0	R	Флаг выхода в рабочий режим PLL_VDU 0 – PLL нестабильна 1 – PLL стабильна

Таблица 1.820 - формат регистра PLL_CTRL (0x004) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	PLL_BYPASS	1’b0	R/W	Выбор опорного синхросигнала 0 – выбран синхросигнал с выхода PLL_VDU 1 – выбран внешний синхросигнал CLK_VDU
[3:2]	-	-	R0	зарезервировано
[1]	PLL_ENABLE	1’b1	R/W	Включение/выключение PLL
[0]	PLL_RESTART	1’b0	R/W	Запись 1 запускает процедуру перезапуска блока PLL. Бит сбрасывается в 0 автоматически после выполнения процедуры перезапуска PLL.

Таблица 1.821 - формат регистра WR_LOCK (0x00C) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:0]	WR_LOCK	32’h0000_0001	R/W	Запись значения 0x1ACCE551 разрешает запись в другие регистры. Запись любого другого значения запрещает запись в другие регистры. Чтение возвращает статус блокировки: 0x0000_0000 – запись в регистры разрешена 0x0000_0001 – запись в регистры запрещена

Таблица 1.822 - формат регистра RST_CTRL (0x014) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:20]	-	-	R0	зарезервировано
[19:16]	RSTREQ_MODE	4’h0		Каждый бит поля определяет поведение при приходе соответствующего сигнала запроса на сброс rstreq_i 1'b0 – сброс только внешних блоков. PLL не перезапускается, значения конфигурационных регистров CRG сохраняются. 1'b1 – сброс внешних блоков, перезапуск PLL, сброс конфигурационных регистров CRG.
[15:12]	-	-	R0	зарезервировано
[11:0]	RST_DUR	12'h032	R/W	Длительность внешних сигналов сброса, в периодах тактового сигнала на выходе PLL

Таблица 1.823 - формат регистра RST_EN (0x018) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	RSTN_CODEC	1’b1	R/W	Сброс HLB_CODEC

Таблица 1.824 - формат регистра UPD_CK (0x01C) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	UPD_CKEN	1'b0	R/W	Применение разрешений выдачи тактовых сигналов. После настройки регистров CKENx необходимо установить этот бит в 1, чтобы новые значения этих регистров вступили в силу. Бит сбрасывается автоматически после применения изменений.
[3:1]	-	-	R0	зарезервировано
[0]	UPD_CKDIV	1'b0	R/W	Применение коэффициентов деления доменов синхросигналов. После настройки регистров CKDIVMODEx необходимо установить этот бит в 1, чтобы новые значения этих регистров вступили в силу. Бит сбрасывается автоматически после смены режима делителей.

Таблица 1.825 - формат регистра CKALIGN (0x020) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:3]	-	-	R0	зарезервировано
[2:0]	CKALIGN	3'b100	R/W	Управление выравниванием тактовых сигналов по фазе при смене коэффициентов делителей: CKALIGN[i]=1 – тактовые сигналы домена i будут выровнены по фазе с тактовыми сигналами тех доменов, для которых CKALIGN также равен 1 CKALIGN[i]=0 – тактовые сигналы домена i не будут выровнены по фазе с остальными

Таблица 1.826 - формат регистра CKDIVSTAT (0x024) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:3]	-	-	R0	зарезервировано
[2:0]	CKDIVSTAT	0	R	Каждый бит поля отражает статус изменения соответствующего регистра CKDIVMODEx: 0 – значение регистра CKDIVMODEx не менялось 1 – значение регистра CKDIVMODEx было изменено, но не вступило в силу

Таблица 1.827 - формат регистра PLL_PRDIV (0x030) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4:0]	PLL_PRDIV	5'd1	R/W	Коэффициент входного делителя PLL , корректными являются значения 1..31

Таблица 1.828 - формат регистра PLL_FBDIV (0x034) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:9]	-	-	R0	зарезервировано
[8:0]	PLL_FBDIV	9'd88	R/W	Коэффициент делителя обратной связи PLL, корректными являются значения 8..511

Таблица 1.829 - формат регистра PLL_PSDIV (0x038) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:2]	-	-	R0	зарезервировано
[1:0]	PLL_PSDIV	2'b10	R/W	Коэффициент выходного делителя: 2'b00: 1 2'b01: 2 2'b10: 4 2'b11: 8

Таблица 1.830 - формат регистра CKDIVMODE_VOUT0(0x100) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h15	R/W	Коэффициент деления для выходного цифрового видеоинтерфейса VOUT0. Fclk=Fpll/(DIVMODE+1)

Таблица 1.831 - формат регистра CKEN_VOUT0 (0x104) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	VOUT0_CLKEN	1'b1	R/W	Разрешение тактирования интерфейса VOUT0: 1 – разрешено, 0 - запрещено

Таблица 1.832 - формат регистра CKDIVMODE_VOUT1(0x110) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h15	R/W	Коэффициент деления для выходного цифрового видеоинтерфейса VOUT1. Fclk=Fpll/(DIVMODE+1)

Таблица 1.833 - формат регистра CKEN_VOUT1 (0x114) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	VOUT1_CLKEN	1'b1	R/W	Разрешение тактирования интерфейса VOUT1: 1 – разрешено, 0 - запрещено

Таблица 1.834 - формат регистра CKDIVMODE_CODEC (0x120) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h01	R/W	Коэффициент деления для подсистемы HLB_CODEC Fclk=Fpll/(DIVMODE+1)

Таблица 1.835 - формат регистра CKEN_CODEC (0x124) в CRG_VDU

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	CODEC_CLKEN	1'b1	R/W	Разрешение тактирования подсистемы HLB_CODEC: 1 – разрешено, 0 - запрещено

1.3.13.4.5.4 Программная модель контроллера CRG_AUD

Таблица 1.836 – список регистров контроллера CRG_AUD

Адрес	Название	Описание
0x000	PLL_STATE	Регистр состояния блока умножения частоты (PLL)
0x004	PLL_CTRL	Управление PLL
0x008	-	Зарезервировано
0x00C	WR_LOCK	Регистр блокировки записи в другие регистры
0x010	-	Зарезервировано
0x014	RST_CTRL	Настройка поведения отработки входных сигналов сброса
0x018	-	Зарезервировано
0x01C	UPD_CK	Активация новых значений регистров CKDIVx и CKENx
0x020	CKALIGN	Управление выравниваем синхросигналов по фазе
0x024	CKDIVSTAT	Признаки изменения регистров CKDIVMODEx
0x028	-	Зарезервировано
0x02C	-	Зарезервировано
0x030	PLL_PRDIV	Коэффициент входного делителя PLL
0x034	PLL_FBDIV	Коэффициент делителя обратной связи PLL
0x038	PLL_PSDIV	Коэффициент выходного делителя PLL
…	-	Зарезервировано
0x100	CKDIVMODE_AUD	Коэффициент деления базовой частоты аудиоинтерфейсов
0x104	CKEN_AUD	Отключение/включение синхросигнала аудиоинтерфейсов
0x108	-	Зарезервировано
0x10C	-	Зарезервировано

Таблица 1.837 - формат регистра PLL_STATE (0x000) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	PLL_RDY	1’b0	R	Флаг выхода в рабочий режим PLL_AUD 0 – PLL нестабильна 1 – PLL стабильна

Таблица 1.838 - формат регистра PLL_CTRL (0x004) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	PLL_BYPASS	1’b0	R/W	Выбор опорного синхросигнала 0 – выбран синхросигнал с выхода PLL_AUD 1 – выбран внешний синхросигнал CLK_AUD
[3:2]	-	-	R0	зарезервировано
[1]	PLL_ENABLE	1’b1	R/W	Включение/выключение PLL
[0]	PLL_RESTART	1’b0	R/W	Запись 1 запускает процедуру перезапуска блока PLL. Бит сбрасывается в 0 автоматически после выполнения процедуры перезапуска PLL.

Таблица 1.839 - Формат регистра WR_LOCK (0x00C) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:0]	WR_LOCK	32’h0000_0001	R/W	Запись значения 0x1ACCE551 разрешает запись в другие регистры. Запись любого другого значения запрещает запись в другие регистры. Чтение возвращает статус блокировки: 0x0000_0000 – запись в регистры разрешена 0x0000_0001 – запись в регистры запрещена

Таблица 1.840 - формат регистра RST_CTRL (0x014) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:20]	-	-	R0	зарезервировано
[19:16]	RSTREQ_MODE	4'h0		Каждый бит поля определяет поведение при приходе соответствующего сигнала запроса на сброс rstreq_i 1'b0 – сброс только внешних блоков. PLL не перезапускается, значения конфигурационных регистров CRG сохраняются. 1'b1 – сброс внешних блоков, перезапуск PLL, сброс конфигурационных регистров CRG.
[15:0]	-	-	R0	зарезервировано

Таблица 1.841 - формат регистра UPD_CK (0x01C) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4]	UPD_CKEN	1'b0	R/W	Применение разрешений выдачи тактовых сигналов. После настройки регистров CKENx необходимо установить этот бит в 1, чтобы новые значения этих регистров вступили в силу. Бит сбрасывается автоматически после применения изменений.
[3:1]	-	-	R0	зарезервировано
[0]	UPD_CKDIV	1'b0	R/W	Применение коэффициентов деления доменов синхросигналов. После настройки регистров CKDIVMODEx необходимо установить этот бит в 1, чтобы новые значения этих регистров вступили в силу. Бит сбрасывается автоматически после смены режима делителей.

Таблица 1.842 - Формат регистра CKDIVSTAT (0x024) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:1]	-	-	R0	зарезервировано
[0]	CKDIVSTAT	0	R	Каждый бит поля отражает статус изменения соответствующего регистра CKDIVMODEx: 0 – значение регистра CKDIVMODEx не менялось 1 – значение регистра CKDIVMODEx было изменено, но не вступило в силу

Таблица 1.843 - формат регистра PLL_PRDIV (0x030) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:5]	-	-	R0	зарезервировано
[4:0]	PLL_PRDIV	5'd1	R/W	Коэффициент входного делителя PLL , корректными являются значения 1..31

Таблица 1.844 - формат регистра PLL_FBDIV (0x034) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:9]	-	-	R0	зарезервировано
[8:0]	PLL_FBDIV	9'd64	R/W	Коэффициент делителя обратной связи PLL, корректными являются значения 8..511

Таблица 1.845 - формат регистра PLL_PSDIV (0x038) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:2]	-	-	R0	зарезервировано
[1:0]	PLL_PSDIV	2’b10	R/W	Коэффициент выходного делителя: 2’b00: 1 2’b01: 2 2’b10: 4 2’b11: 8

Таблица 1.846 - формат регистра CKDIVMODE_AUD(0x100) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:8]	-	-	R0	зарезервировано
[7:0]	DIVMODE[7:0]	8'h0F	R/W	Коэффициент деления для базовой частоты аудиоинтерфейсов I2S и SPDIF Fclk=Fpll/(DIVMODE+1)

Таблица 1.847 - Формат регистра CKEN_AUD (0x104) в CRG_AUD

Биты	Название	Сброс	Доступ	Описание
[31:1]	Зарезервировано	-	-	-
[0]	CLKEN_AUD	1’b1	R/W	Разрешение тактирования аудиоинтерфейсов 1 – разрешено, 0 - запрещено

1.3.13.5 Блок датчиков температуры/напряжения

1.3.13.5.1 Назначение

Блок датчиков температуры/напряжения предназначен для контроля температуры (напряжения) ядра процессора Power PC. В случае выхода за границу порогов контролируемых значений в процессор выдаются прерывания. Контроллер датчиков содержит набор регистров для настройки усреднения показаний датчиков, обработки прерываний, выбора тестовых режимов, режимов работы, управления питанием. Более подробно регистры описаны ниже. Блок содержит 3 пары датчиков температуры и напряжения, которые запускаются синхронно. Выбор усредненных/мгновенных показаний каждого из 3-х датчиков производится программно при чтении в одном регистре 3-х усредненных/мгновенных значений. Выбор датчика при обработке прерываний задается маской в регистре контроллера.

1.3.13.5.1.1 Функции термодатчика/датчика напряжения

Усреднение показаний по нескольким последним отсчетам для устранения выбросов.

Количество точек для усреднения определяется в конфигурационном регистре. От 1 до 16 по степеням 2. Реализован вариант усреднения отсчетов не подряд, а выбираемых на более низкой частоте.

Генерация признака изменения средней температуры/напряжения, в сторону увеличения или в сторону уменьшения.

Генерация прерывания при превышении средней температурой/напряжением заданного порогового значения.

Генерация прерывания при снижении средней температуры/напряжения ниже заданного порогового значения.

Всего 4 прерывания. Два определяют границы порога по верхней границе температуры/напряжения, два определяют границы порогов по нижней границе температуры/напряжения. Из контроллера датчиков выдается в систему два прерывания, происхождение которых определяется расцветкой в регистре состояния.

Доступ к конфигурационным регистрам и возможность непосредственного опроса датчика по интерфейсу APB 3.0.

Функции усреднения данных и генерации прерываний – отключаемые

1.3.13.5.1.2 Регистры датчиков напряжения/температуры

Таблица 1.848 - Регистры датчиков напряжения/температуры

Мнемоника	Относительный адрес [11:0] APB	RW	Разряд-ность	Значение после сброса	Описание
TAVG	00 40	R	[9:0]	10’h000	Среднее значение датчика температуры (напряжения) по отсчетам
TCTRL_AVG	04 44	RW	[9:0]	10’h000	Количество отсчетов, по которым производится усреднение
TGRAD	08 48	R	[5:0]	6’b00	Флаг изменения температуры (напряжения)в сторону увеличения или уменьшения

Продолжение таблицы 1.848

Мнемоника	Относительный адрес [11:0] APB	RW	Разряд-ность	Значение после сброса	Описание
THIGH	0C 4C	RW	[19:0]	20'h00000	THIGH[19:10] – THIGH-H THIGH_H – порог срабатывания датчика превышения температуры (напряжения) при превышении температурой (напряжением) верхнего порога (задается в милливольтах) THIGH[9:0] – THIGH-L THIGH_L – порог срабатывания датчика превышения температуры (напряжения) при снижении температуры (напряжения) ниже верхнего (температурного) порога (в вольтах). Например, THIGH_L =260(125C), THIGH_H=265 (120C). Если температура повышается более 125 градусов, срабатывает прерывание (устройство вышло из рабочего режима). Если затем температура снижается ниже 120 градусов, тоже срабатывает прерывание (устройство вернулось в рабочий режим)
TLOW	10 50	RW	[19:0]	20'h00000	TLOW[19:10] – TLOW-L TLOW_L– порог срабатывания датчика снижения температуры (напряжения) при пересечении температурой (напряжением) нижнего допустимого порога TLOW_H – порог срабатывания датчика снижения температуры (напряжения) при повышении температуры (напряжения) до допустимого порога. Например, TLOW_H = 434(-40C), TLOW_L= 428(- 35C). Если температура снижается до -40 градусов, срабатывает прерывание (устройство вышло из рабочего режима). Если затем температура повышается до -35C градусов тоже срабатывает прерывание (устройство вернулось в рабочий режим)
TINT	14 54	R	[11:0]	12'b0	Регистр прерываний
TINT_MASK	18 58	RW	[11:0]	12'b0	Регистр масок прерываний
TSENS	1C 5C	R	[9:0]	10'h000	Регистр непосредственного значения напряжения на датчике.
TTMODE	20 60	RW	[2:0]	3'b0	Регистр включения тестового режима генерации прерываний
TTINT	24 64	RW	[11:0]	12'b0000	Тестовый режим генерации прерываний.
TID	28	RW	[31:0]	32'h00000001	ID блока датчиков
TSTART	2C(6C)	RW	[4:0]	7'b00000001	[0]= 1 Enable T/V ADC [1] – 1-power down, 0- enable sensor [2]= 1 TMODE датчиков [3] =1, AGND (ADC=10 в TMODE), [4] =1 AVDD(ADC=11 в TMODE), [5]=1 bypass mode [6] =1 enable инвертора clock для АЦП интерфейса

Примечания:

1) *В таблице регистров приведены примеры для температурного датчика из таблицы, взятой из документа на датчик от компании –производителя.*
2) *Регистр TSTART является общим по записи для 3-х датчиков температуры и напряжения. По чтению для нормального и bypass режимов датчика адреса -2c(6c). В тестовом режиме (TSTART[2] =1) проверяется один из режимов, указанных в TSTART[4:3], для проверки АЦП можно использовать один набор регистров.*
   С целью усреднения показаний в системе используется 3 датчика. При чтении регистров TAVG, TSENS разряды [29:20], [19:10], [9:0] относятся к 3, 2 и 1 датчикам, соответственно. Аналогично при чтении TGRAD разряды [5:4], [3:2], [1:0] относятся к 3, 2 и 1 датчикам.
   Аналогично TSTART регистр TCTRL_AVG является общим только по записи, т.к. функция усреднения выбранного датчика в основном режиме работы с разделением времени производится в режиме кругового обхода с равным по времени доступом для каждого датчика.
   TCTRL_AVG[4:0] – отсчеты по степеням двойки:
   TCTRL_AVG[4:0]= 00001 – 1 отсчет
   TCTRL_AVG[4:0]= 00010 – 2 отсчета
   TCTRL_AVG[4:0]= 00100 – 4 отсчета
   TCTRL_AVG[4:0]= 01000 – 8 отсчетов
   TCTRL_AVG[4:0]= 10000 – 16 отсчетов
   TCTRL_AVG[9:5] – период пониженной частоты, N-1.
   Отсчеты по степеням двойки используются как при усреднении подряд, так и при усреднении на пониженной частоте по последним отсчетам, когда период усреднения задается в регистре TCTRL_AVG[9:5].
3) *Если в работающих датчиках производится смена режима bypass ⇔ normal или выключенные датчики переводятся в рабочий режим, производится обнуление всех аппаратных триггеров, которые программно недоступны по записи.*
4) *Для 3-х датчиков используется 12 прерываний, по 4 прерывания для каждого датчика.*
   Тип выбора прерываний задается маской в регистре TTMODE.
   Если прерывание не тестовое, т.е. TTMODE [0] = 0, то режим выборки прерываний определяется в TTMODE [2:1]
   TTMODE [0] = 1 – источник прерываний - тестовый регистр
   TTMODE [2:1] = 0 – все прерывания по ИЛИ от всех датчиков
   TTMODE [2:1] = 1 – HH, HL, LH, LL по ИЛИ от 3-х датчиков, если возникли одновременно в своей группе (HH или HL, или LH или LL)
   TTMODE [2:1] = 2 – мажоритарно 2 из 3 в своей группе HH или HL, или LH или LL

1.3.13.5.1.3 Функциональная схема

Функциональная схема блока датчиков показана на рисунке 1.315.

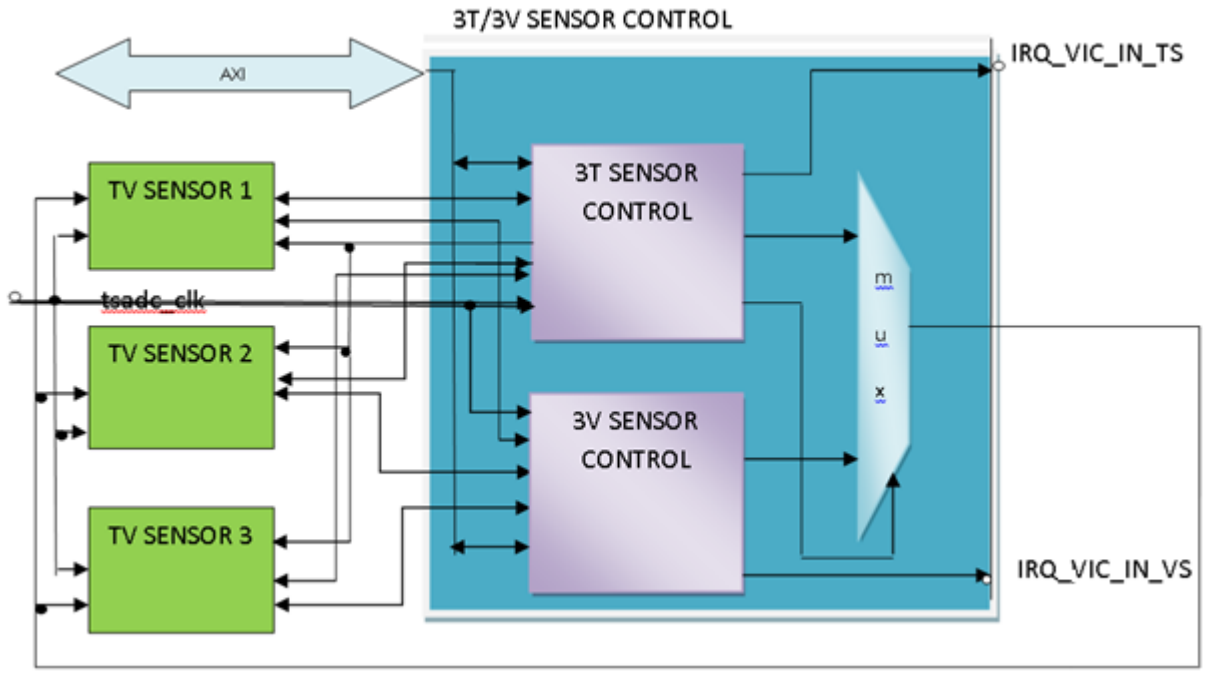


Рисунок 1.315 - Функциональная схема блока датчиков

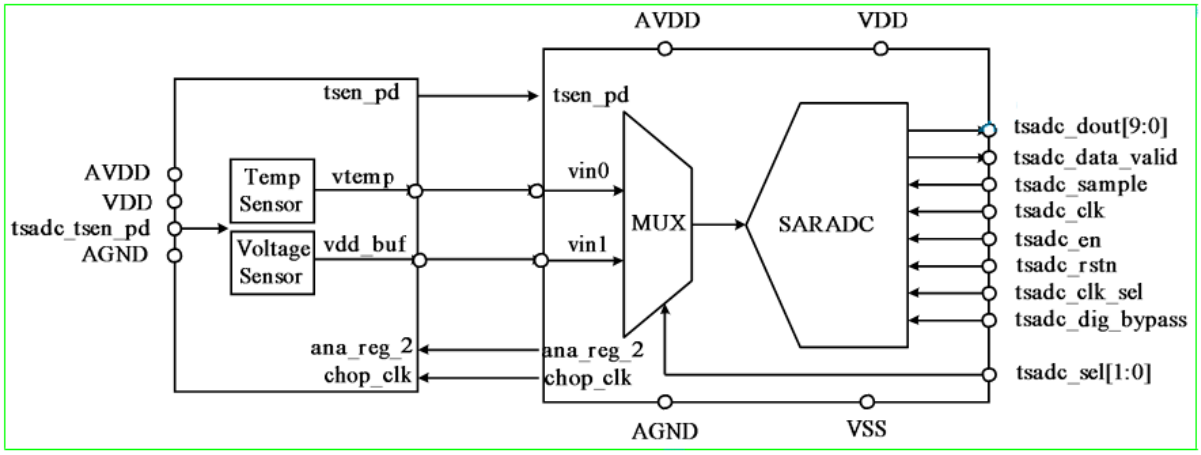


Рисунок 1.316 - Блок-схема датчика напряжения/температуры TV SENSOR

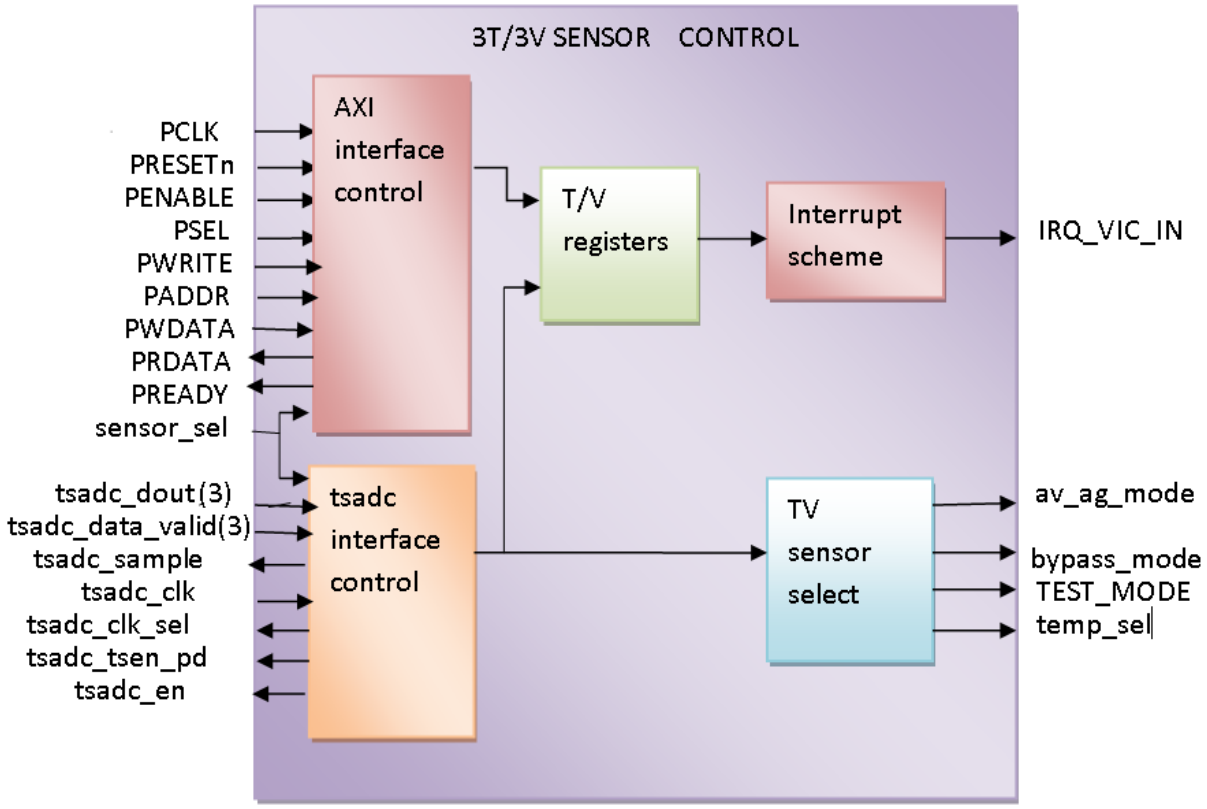


Рисунок 1.17717 - Схема управления датчиками и регистры задания режимов

Блок датчиков температуры/напряжения TV sensor содержит аналого-цифровой преобразователь напряжения SARADC, входной мультиплексор MUX и датчики температуры и напряжения. Каждый датчик имеет через SARADC 10-разрядный выход, сответственно, температуры и напряжения. MUX используется для выборки входного канала для SARADC. SARADC будет брать образец входного сигнала и преобразовывать его в цифровые данные с 10-разрядным разрешением.

Скорость преобразования может быть до 50KS/s (килосэмпл/сек). Каждое преобразование занимает 15~16 циклов синхроимпульсов. Соответствующая частота синхроимпульсов –500 кГц(макс 800кГц). Скорость преобразования может быть настроена изменением частоты синхроимпульсов или увеличением количества циклов синхроимпульсов преобразования.

Усредненные значения записываются в регистр TAVG (VAVG) на пониженной частоте, задаваемой количеством отсчетов регистра TCTRL_TAVG (TCTRL_VAVG). Если усреднение не производится, то значения в регистре TAVG (VAVG)и регистре непосредственных значений датчика TSENS(VSENS) совпадают.

В рабочем режиме запись значений 3-х датчиков температуры управляется контроллером 3T SENSOR CONTROL и, соответственно, запись значений 3-х датчиков напряжения контроллером 3V SENSOR CONTROL в режиме разделения времени.

Запуск 3-х блоков TV SENSOR в рабочем режиме производится одновременно записью в регистр TSTART(общий по записи для 3-х блоков датчиков). При чтении регистров TAVG(VAVG) и TSENS(VSENS) в одном регистре считываются показания 3 датчиков.

1.3.13.5.2 Характеристики электрических сигналов

Таблица 1.849

Параметр	Символ	Условие	Мин	Тип 5	Макс	Единица
Рабочие условия						
Напряжение	AVDD	-	1.62	1.8	1.98	V
Напряжение	VDD ENOB	-	0.9	1.0	1.1	V
Предельная температура	T j	-	-40	-	125	C
Характеристики АЦП						
Точность измерения	-	-	-	10	-	bit
Эффективное количество бит	ENOB	-	-	9	-	bit
Дифференциальная нелинейность бит ненелинейность	DNL	-	-1	-	+1	LSB
Интегральная нелинейность	INL	-	-2	-	+2	LSB
Диапазон входного напряжения	VIN	-	0	-	1	AVDD
Входная емкость	CIN	-	-	8	-	pF
Скорость сэмплирования	fs	-	-	-	50	KS/s
Соотношение сигнал/шум нелинейность	SNDR	-	-	55	-	dB
Характеристики датчика температуры						
Температурный диапазон		-	-40	-	+125	C
Точность измерения температуры		-		5	-	C
Временные характеристики						
Частота синхронизации	fclk	-	-	-	800	
Период синхросигнала	tclk	-	1.25	-	-	us
Коэффициент заполнения	-	-	45		55	%
Время преобразования	-	-	16	-	-	tclk
Время высокого уровня образцового сигнала	tsmp	-	-	1	-	tclk
Время значимого сигнала dout[7:0]	tdout	-	-	1	-	tclk
Временной интервал между передачей sel[1:0] и передним фронтом образцового сигнала	tsel1	-		1		tclk
Временной интервал между задним фронтом образцового сигнала и передачей sel[1:0]	tsel2	-	-	1		tclk
Мощность рассеивания						
Ток через аналоговый вход питания аналогового источника напряжения напруfghUнанапряжения	I AVDD	Fs=50KS/s	-	90	-	uA
		Пит.выкл выкл 3	-	1	-	uA
Ток через цифровой вход питания	I VDD	Fs=50KS/s	-	10	-	uA
		Пит.выкл		1	-	uA

Условия тестирования: AVDD = 1,8в. VDD = 1,0в, TA=25d’

Соответствие между температурой и выходными данными T/V ADC показано в Таблице 1.850.

Таблица 1.850

Температура/C	Выходные данные АЦП		
	Макс	Тип	Мин
-40	-	650	-
-35	-	642	-
-30	-	635	-
-25	-	627	-
-20	-	619	-
-15	-	611	-
-10	-	603	-

Продолжение таблицы 1.850

Температура/С	Выходные данные АЦП		
	Макс	Тип	Мин
-5	-	595	-
0	-	588	-
5	-	580	-
10	-	572	-
15	-	564	-
20	-	556	-
25	-	548	-
30	-	541	-
35	-	533	-
40	-	525	-
45	-	517	-
50	-	509	-
55	-	501	-
60	-	493	-
65	-	485	-
70	-	477	-
75	-	469	-
80	-	462	-
85	-	454	-
90	-	446	-
95	-	438	-
100	-	430	-
105	-	422	-
110	-	414	-
115	-	406	-
120	-	398	-
125	-	390	-

1.3.13.5.3 Список сигналов термодатчика/датчика напряжения

Таблица 1.851 - Список аналоговых сигналов термодатчика/датчика напряжения

Обозн.	Тип	Описание
AVDD	Питание	1.8 в напряжения
VDD	Питание	1.0 в напряжения
AGND	Земля	Земля
VSS	Земля	Земля
vtemp	Выход	Выходное напряжение температурногоьсенсора
vdd_buf	Выход	Выходное напряжение сенсора наряжения
vin0	Вход	Вход АЦП
vin1	Вход	Вход АЦП

Таблица 1.852 - Список цифровых сигналов термодатчика/датчика напряжения

Название	Тип	Состояние после обнуления	Описание
tsadc_clk	Вход	-	Сигнал синхронизации clock T/V датчика
tsadc_rstn	Вход	0	Сигнал обнуления цифровой части. Активный - низкий уровень.
tsadc_dig_bypass	Вход	0	bypass функция делает автоматически enable bandgap chooper модуля 0: enable автоматически 1: установить ana_reg_2 выход в 0
tsadc_sample	Вход	-	sample сигнал T/V датчика Активный - высокий уровень.
tsadc_en	Вход	1	enable сигнал T/V датчика Активный - высокий уровень 0: T/V датчик не активен 1: T/V датчик активен
tsadc_sel[1:0]	Вход	-	Выбор входа ADC 00: выбор vin0 01: выбор vin1 10: выбор AGND для тестового режима 11: выбор AVDD для тестового режима
tsadc_tsen_pd	Вход	0	Сигнал отключения питания датчика температуры 0: датчик активен 1: питание отлючено
tsadc_data_valid	Выход	-	Сигнал значимости 10-разрядных выходных данных. Активный- высокий
tsadc_dout[9:0]	Выход	-	10-разрядные выходные данные Выходные данные будут активны, когда tsadc_data_valid высокий и tsadc_en высокий

1.3.13.5.4 Временные диаграммы

T/V-ADC будет занимать от 15-16 циклов до завершения преобразования в зависимости от установленнх сигналов. В разделах 4.1 и 4.2 будут приведены временные диаграммы для различных установок сигналов.

1.3.13.5.4.1 Начальная временная диаграмма

Перед запуском сэмпла (tsadc_sample) мы будем ожидать перехода датчика и АЦП в стабильное состояние. Временная зависимость между сигналами tsadc_en/tsadc_tsen_pd и tsadc_sample показана на рисунке 1.318.

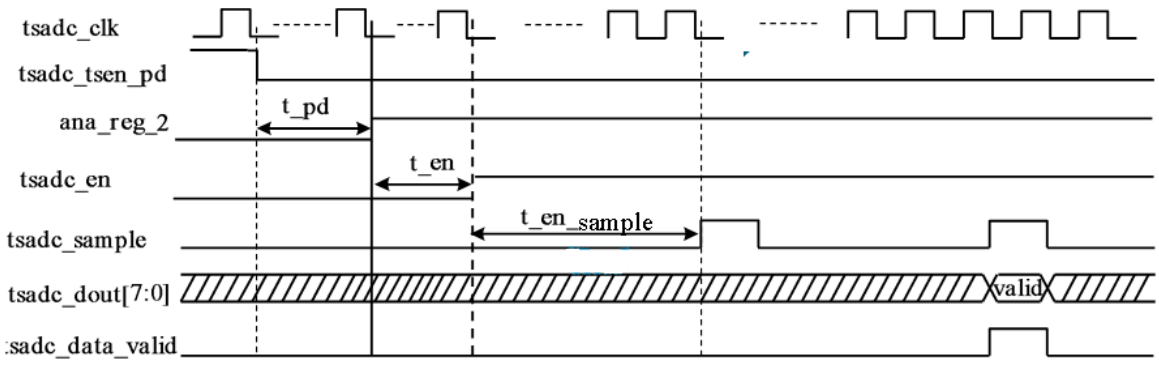


Рисунок 1.318 - Стартовая последовательность для запуска датчика и АЦП

Временная задержка t_pd между tsadc_tsen_pd (активный низкий уровень) и tsadc_ana_reg_2 (активный высокий) должна быть не менее 10us для стабильности цепей. Временная задержка t_en между tsadc_ana_reg_2 (высокий уровень) и tsadc_en (высокий уровень) должна быть больше 0us для системы, чтобы нормально функционировать. Временная задержка t_en_sample

между tsadc_en (высокий уровень) и tsadc_sample (активный высокий) должна быть не менее 90us для того, чтобы система функционировала нормально.

Таблица 1.853

Timing Parameter	MAX	TYP	MIN
t_pd			lOus
t_en			Ous
t_en_sample			9Ous

1.3.13.5.4.2 Нормальный режим

Когда T/V-ADC работает в нормальном режиме (tsadc_dig_bypass =1'b0), типичная временная диаграмма T/V-ADC показана на Рис. 4-2 и Рис.4-3. Она имеет 2 условия, когда tsadc_clk_sel имеет различные логические уровни.

Когда tsadc_clk_sel =0, T/V-ADC будет занимать 15 циклов, чтобы завершить аналого-цифровое ADC преобразование. Временная диаграмма показана на рисунке 1.319.

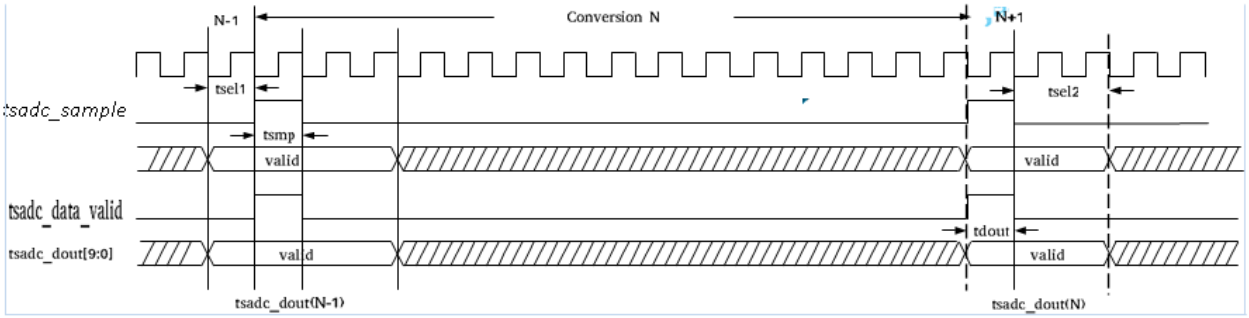


Рисунок 1.319 - T/V-ADC временная диаграмма, когда tsadc_clk_sel=0

ADC преобразование начинается, когда tsadc_sample активный высокий. Оно будет занимать 15 циклов синхроимпульсов до завершения преобразования. Когда ADC завершает преобразование, он будет выдавать сигнал валидности данных и выдаст результат преобразования. tsadc_dout валидный, когда уровень tsadc_data_valid высокий. tsadc_dout будет оставаться неизменным до 2 циклов снхроимпульсов, когда sadc_sample валидный.

Если время конверсии установлено больше 15 циклов синхроимпульсов, tsadc_dout[9:0]

будет удерживать валидные данные до 2 циклов синхроимпульсов, когда tsadc_sample валидный

Когда tsadc_clk_sel =1, T/V-ADC будет занимать 16 циклов, чтобы завершить аналого-цифровое ADC преобразование.

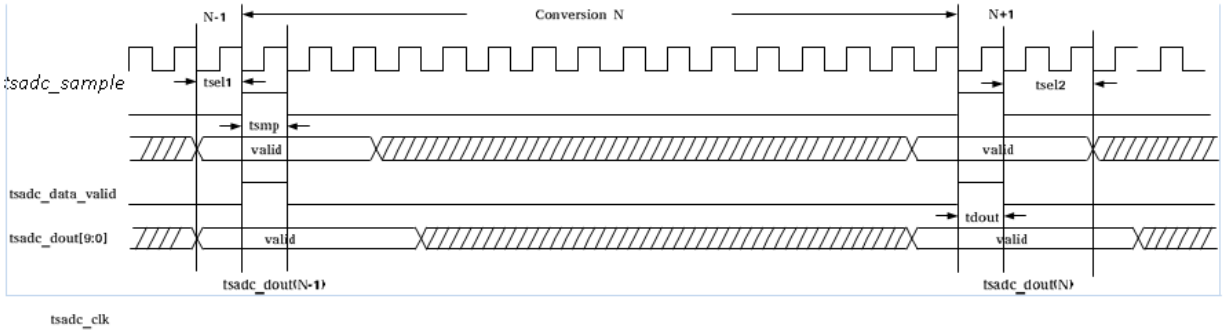


Рисунок 1.320 - T/V-ADC временная диаграмма, когда tsadc_clk_sel=1

ADC преобразование начинается, когда tsadc_sample активный высокий. Оно будет занимать 16 циклов синхроимпульсов до завершения преобразования. Когда ADC завершает преобразование, он будет выдавать сигнал валидности данных и выдаст результат преобразования. tsadc_dout будет оставаться неизменным до 3 циклов снхроимпульсов, когда sadc_sample валидный.

Если время конверсии установлено больше 16 циклов синхроимпульсов, tsadc_dout[9:0]

будет удерживать валидные данные до 3 циклов синхроимпульсов, когда tsadc_sample валидный

1.3.14Подсистема аппаратного тестирования микросхемы

1.3.14.1 Отладочный интерфейс JTAG

Микросхема поддерживает работу интерфейса JTAG в соответствии со стандартом IEEE Std 1149.1-2001. Порт JTAG в микросхеме предназначен для следующих целей:

1) Проверка выводов микросхемы и структурное тестирование печатной платы с помощью тестов граничного сканирования - BSCAN.
2) Аппаратная отладка процессорных ядер PowerPC.

1.3.14.1.1 **Поддерживаемые команды тестового порта JTAG и регистр граничного сканирования**

TAP-контроллер микросхемы поддерживает стандартные команды (инструкции) тестового порта JTAG и ряд пользовательских команд для средств отладки. Длина кода инструкций - 7 бит. Полный перечень доступных инструкций представлен в таблице 1.854.

Таблица 1.854 - Поддерживаемые инструкции JTAG

№	Название	Код	Описание
1	EXTEST	1000001	Позволяет изменять направление и состояние линий ввода/вывода, участвующих в граничном сканировании.
2	SAMPLE/PRELOAD	1000010	Позволяет считывать состояние входов/выходов микросхемы, участвующих в граничном сканировании.
3	BYPASS	1111111	Создание однобитового регистра со входом TDI и выходом TDO.
4	HIGHZ	0000011	Позволяет перевести доступные выходы микросхемы в неактивное состояние.
5	PPC_C_0	0000100	Управление встроенными TAP-контроллерами ядер PowerPC.
6	PPC_C_1	0001100	
7	PPC_C_2	0010100	
8	PPC_C_3	0011100	
9	PPC_C_4	0100100	
10	PPC_C_5	0101100	
11	PPC_C_6	0110100	
12	PPC_C_7	0111100	
13	PPC_C_8	1000100	
14	PPC_C_9	1001100	
15	PPC_C_A	1010100	
16	PPC_C_B	1011100	
17	PPC_C_C	1100100	
18	PPC_C_D	1101100	
19	PPC_C_E	1110100	
20	PPC_C_F	1111100	
21	PPC_0_0	0000101	
22	PPC_0_1	0001101	
23	PPC_0_2	0010101	
24	PPC_0_3	0011101	
25	PPC_0_4	0100101	
26	PPC_0_5	0101101	
27	PPC_0_6	0110101	
28	PPC_0_7	0111101	
29	PPC_0_8	1000101	
30	PPC_0_9	1001101	
31	PPC_0_A	1010101	
32	PPC_0_B	1011101	
33	PPC_0_C	1100101	
34	PPC_0_D	1101101	

Продолжение таблицы 1.854

№	Название	Код	Описание
35	PPC_0_E	1110101	
36	PPC_0_F	1111101	
37	PPC_1_0	0000110	
38	PPC_1_1	0001110	
39	PPC_1_2	0010110	
40	PPC_1_3	0011110	
41	PPC_1_4	0100110	
42	PPC_1_5	0101110	
43	PPC_1_6	0110110	
44	PPC_1_7	0111110	
45	PPC_1_8	1000110	
46	PPC_1_9	1001110	
47	PPC_1_A	1010110	
48	PPC_1_B	1011110	
49	PPC_1_C	1100110	
50	PPC_1_D	1101110	
51	PPC_1_E	1110110	
52	PPC_1_F	1111110	
53	GRPHLTR	0000001	Работа с внутренним блоком отладки RISCWATCH_SUPPORT.
54	GRPHLTS	0001001	
55	GRP0	0010001	
56	GRP1	0011001	
57	LDBO	0100001	

Схематическое представление регистра граничного сканирования и порта JTAG представлено на рисунке 1.321.

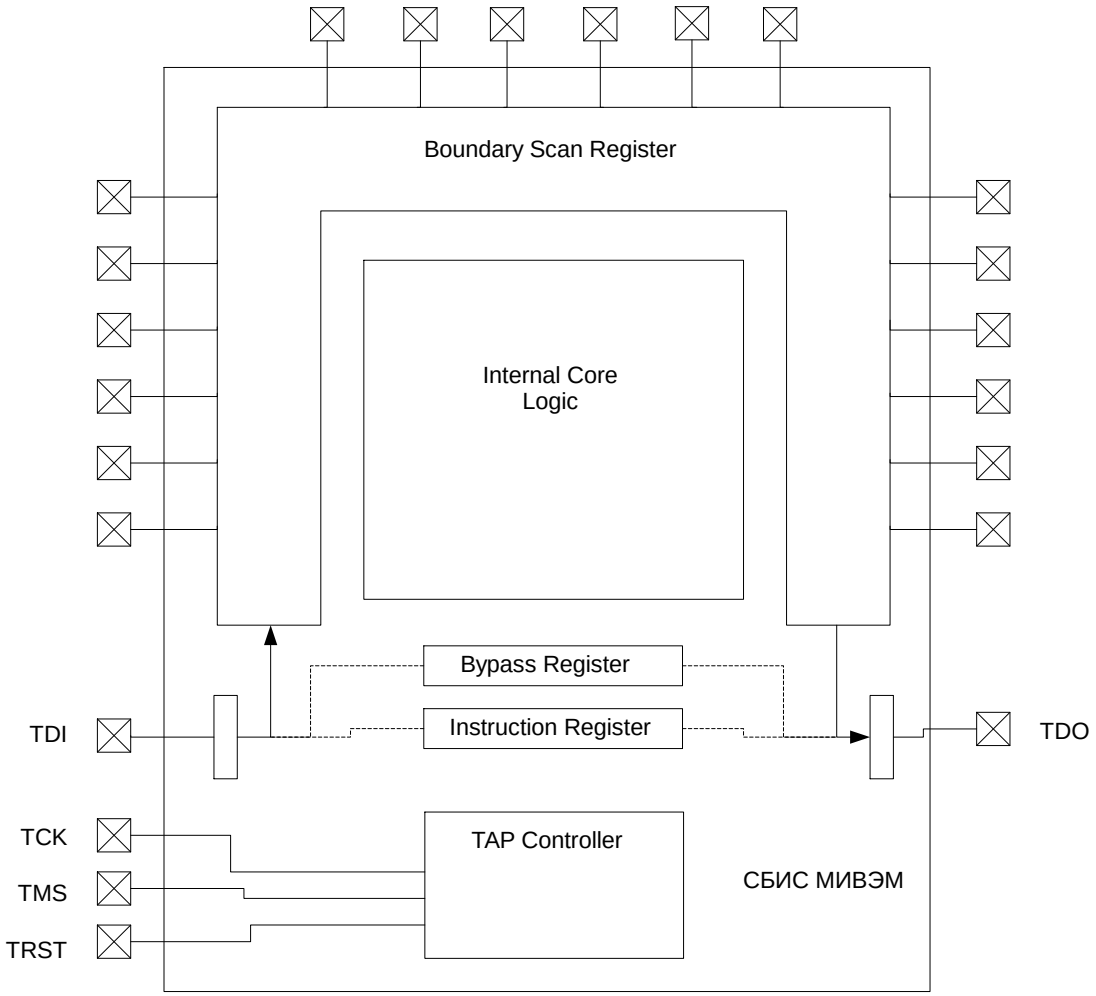


Рисунок 1.321 - Схема устройства граничного сканирования

Boundary Scan Register - регистр граничного сканирования. Полное описание регистра граничного сканирования см. в Приложении А.

Internal Core Logic - внутренняя логика микросхемы.

Bypass Register - регистр обхода JTAG.

Instruction Register - регистр команд JTAG.

TAP Controller - TAP-контроллер JTAG.

1.3.14.1.2 **Средства аппаратной отладки PowerPC**

Средства аппаратной отладки процессорных ядер PowerPC в микросхеме имеют следующую структуру:

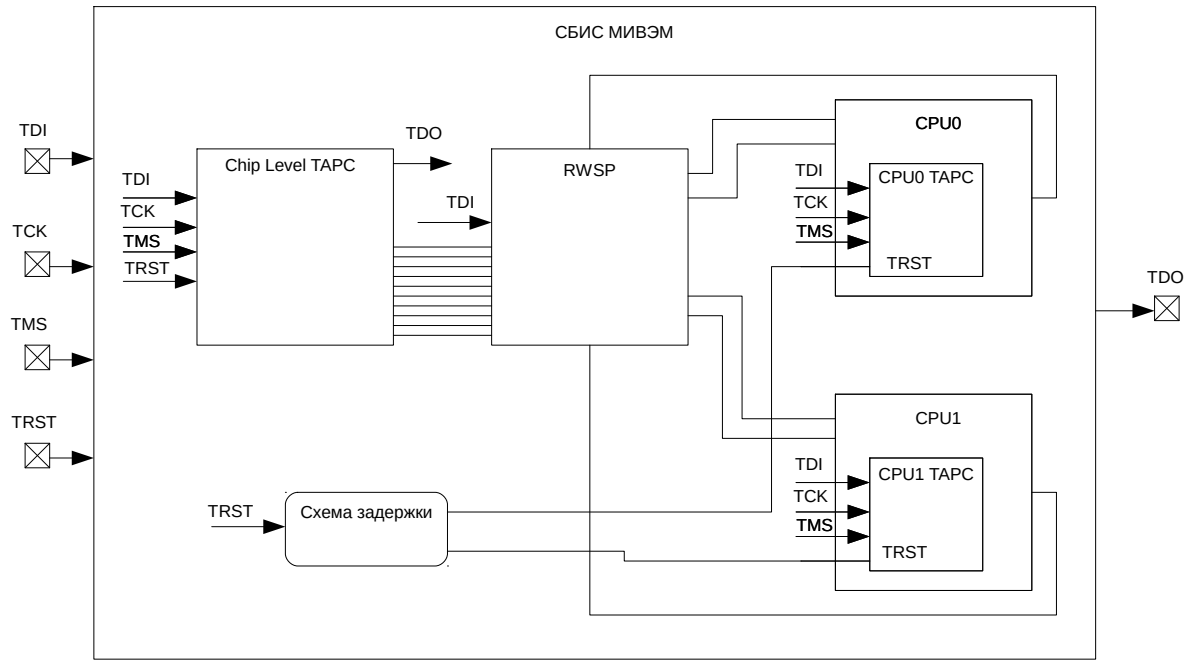


Рисунок 1.322 - Схема внутрикристальных средств аппаратной отладки PowerPC

Chip Level TAPC - TAP-контроллер верхнего уровня.

RWSP - блок RISCWATCH_SUPPORT; используется для разбиения процессорных ядер на группы и возможности управлять этими группами независимо друг от друга.

CPU0 TAPC, CPU1 TAPC - внутренние TAP-контроллеры ядер PowerPC.

Схема задержки - нужна для формирования сигналов сброса TRST встроенных TAP-контроллеров ядер PowerPC.

Для работы со средствами аппаратной отладки используется внешнее устройство "RISCWatch Processor Probe model 41K7356" (далее просто RISCWatch Processor Probe). Для работы с внешним отладчиком рекомендуется использовать следующую конфигурацию:

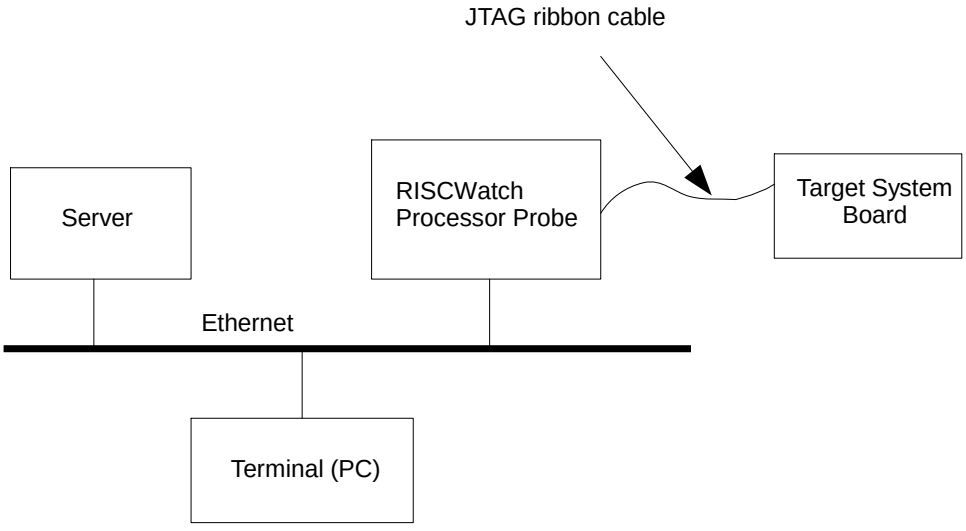


Рисунок 1.323 - Схема подключения внешнего отладчика RISCWatch Processor Probe

Инструментальная плата (**Target System Board**) с установленной микросхемы подключается с помощью ленточного кабеля JTAG (**JTAG ribbon cable**) к аппаратному отладчику **RISCWatch Processor Probe**. Отладчик через сеть TCP/IP Ethernet подключается к серверу, где установлено необходимое ПО. Для доступа пользователю необходимо удаленно

подключиться через терминал (например, ПК) к серверу и запустить программу, реализующую подачу определённых воздействий по JTAG-интерфейсу. Наглядное представление соединения между инструментальной платой (**Board under test**) для микросхемы и внешним отладчиком **RISCWatch Processor Probe** представлено на рис.1.324.

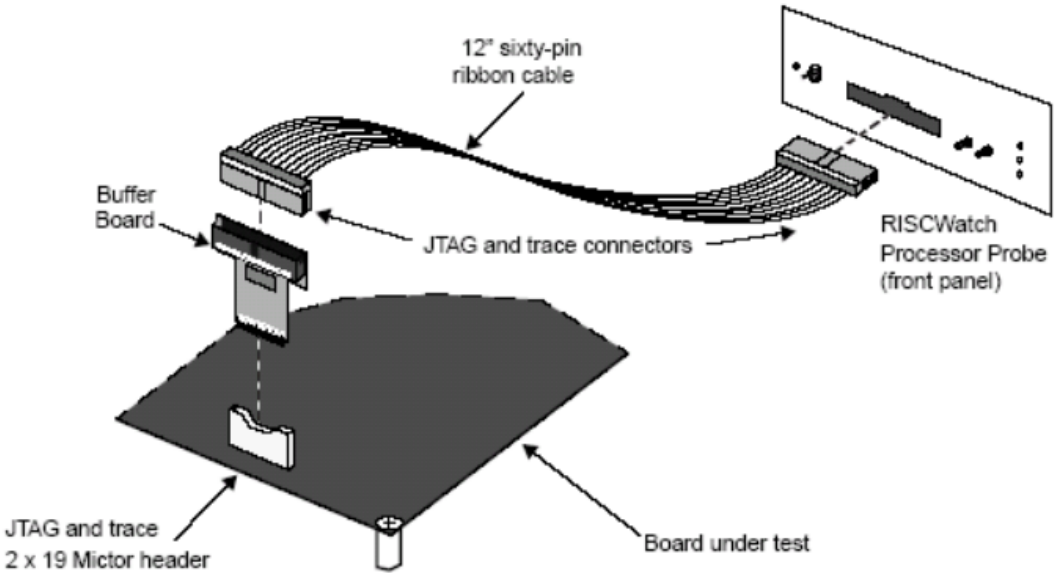
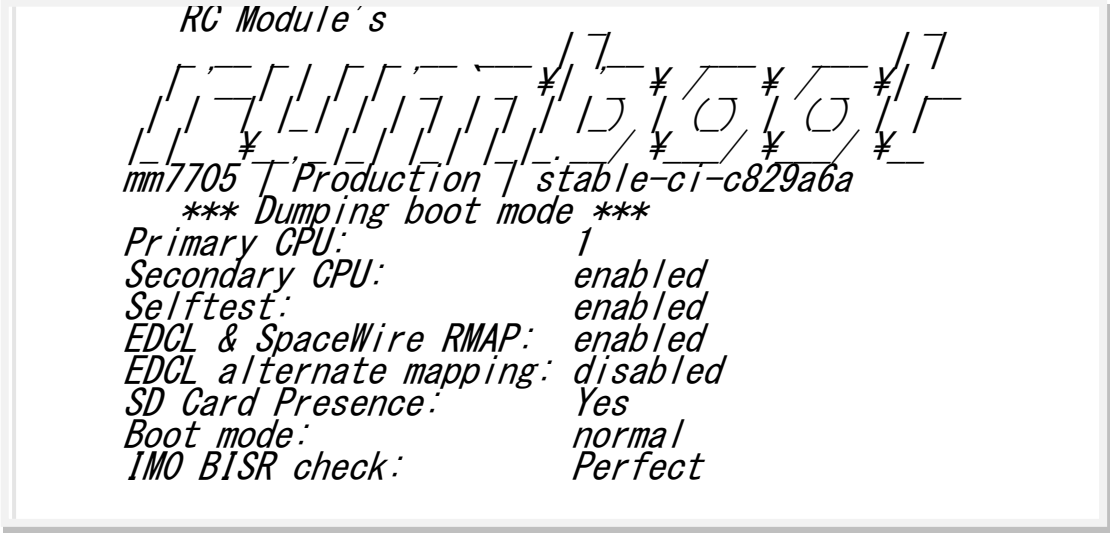


Рисунок 1.324 - Соединение инструментальной платы и отладчика RISCWatch Processor probe

1.3.14.2 Процедура самотестирования

Запуск процедуры самотестирования производится установкой соответствующего значения BOOTM (см. документ по загрузчику) при системном сбросе. При старте загрузчик выводит отладочную информацию в последовательный порт 0 со скоростью 1000000 bps 8 бит данных 1 стоповый бит, нет четности. Пример приведен ниже.



Состояние памяти IM0 отмечено пунктом IM0 Bisr check:

Значение	Расшифровка
Perfect	Исправны все ячейки, включая резервные элементы
Good	Были неисправные элементы, но они были испешно были заменены исправными
Bad	Присутствуют невосстановимые ошибки IM0, кристалл неработоспособен

Список компонентов процедуры самотестирования приведен в таблице 1.855:

Таблица 1.855 - Список компонентов процедуры самотестирования

Название	Примечание
MUSB_BIST-FullSpeed	Встроенный тест USB PHY
DSPMemoryBist-DSP-L2C0 DSPMemoryBist-DSP-L2C1	Тест кэш-памяти DSP подсистемы
DDR3_PHY_BIST-EM0 APB PHY BIST DDR3_PHY_BIST-EM1 APB PHY BIST	BIST тесты DDR контроллеров
XHSIF0-PCIe0 XHSIF0-HSVI0 XHSIF0-HSVI1 XHSIF0-HSVI2 XHSIF0-FC1 XHSIF0-XHSIF0_CTRL XHSIF0-SERDES1 XHSIF0-SERDES2 XHSIF0-SERDES3 XHSIF0-SERDES4 XHSIF1-PCIe1 XHSIF1-HSVI3 XHSIF1-HSVI4 XHSIF1-HSVI5 XHSIF1-FC2 XHSIF1-XHSIF1_CTRL XHSIF1-SERDES5 XHSIF1-SERDES6	Проверка программно-доступных регистров соответствующих блоков

Продолжение таблицы 1.855

Название	Примечание
XHSIF1-SERDES7 XHSIF1-SERDES8 HSIF-HSIF_CTRL HSIF-RMACE HSIF-HSCB0 HSIF-HSCB1 HSIF-PL301_S HSIF-PL301_M LSIF0-NOR LSIF0-SRAM LSIF0-NAND LSIF0-GIGA_GRETH0 LSIF0-GIGA_GRETH1 LSIF0-GRETH1 LSIF0-GRETH2 LSIF0-GRETH3 LSIF0-CONFIG LSIF0-MGPIO0 LSIF0-MGPIO1 LSIF0-MGPIO2 LSIF0-MGPIO3 LSIF0-MGPIO4 LSIF0-MGPIO5 LSIF0-MGPIO6 LSIF0-MGPIO7 LSIF0-MGPIO8 LSIF0-MGPIO9 LSIF0-MGPIO10 LSIF0-PL301_S LSIF0-PL301_M LSIF1-CTRLREG LSIF1-LSCB0 LSIF1-LSCB1 LSIF1-I2C0 LSIF1-I2C1 LSIF1-I2C2 LSIF1-UART0 LSIF1-UART1 LSIF1-UART2 LSIF1-SPI0 LSIF1-SPI1 LSIF1-GSPI LSIF1-SDIO LSIF1-GPIO0 LSIF1-GPIO1 LSIF1-MGPIO0 LSIF1-MGPIO1 LSIF1-MGPIO2 LSIF1-MGPIO3 LSIF1-MGPIO4 LSIF1-MUSB LSIF1-PL301_S LSIF1-PL301_M	Проверка программно-доступных регистров соответствующих блоков

Продолжение таблицы 1.855

Название	Примечание
DCR-PLB4 ARB 0 DCR-PLB4 ARB 1 DCR-PLB4 ARB 2 DCR-PLB4 ARB 3 DCR-PLB4 ARB 4 DCR-PLB4 ARB 5 DCR-PLB4 ARB 6 DCR-LTRACE 0 DCR-LTRACE 1 DCR-ITRACE 0 DCR-DMA2PLB6 0 DCR-DMA2PLB6 1 DCR-PLB6BC DCR-PLB6 DCRARB DCR-MPIC128 DCR-EM0 DDR34LMC DCR-EM0 AXIMCIF2 DCR-EM0 mcif2arb4 DCR-EM0 MCLFIR DCR-EM0 PLB6MCIF2X DCR-EM1 DDR34LMC DCR-EM1 AXIMCIF2 DCR-EM1 mcif2arb4 DCR-EM1 MCLFIR DCR-EM1 PLB6MCIF2X DCR-EM2 PLB6MCIF2X DCR-EM3 PLB6MCIF2X DCR-L2C0 DCR-L2C1 APB-SCTL APB-APB TIMER APB-APB WDT APB-APB THERMALSENS APB-DDR3PHY_EM0 APB-DDR3PHY_EM1 APB-PL301_AXI32S APB-PL301_AXI32M	Проверка программно-доступных регистров соответствующих блоков

Состояние каждого запускаемого теста выводится в ходе тестирования.

1.3 Маркировка и пломбирование

Маркировка микросхемы должна удовлетворять требованиям ТУ на микросхему ЮФКВ.431268.017ТУ. Контроль маркировки должен соответствовать пункту 7.7 ТУ.

1.4 Упаковка

Упаковка микросхемы должна удовлетворять требованиям ТУ на микросхему ЮФКВ.431268.017ТУ. Контроль упаковки должен соответствовать пункту 7.7 ТУ.

2 Использование по назначению микросхемы

2.1 Процедура начальной инициализации

При системном сбросе исполняется программа начального ROM загрузчика. Для вывода отладочной информации загрузчиком используется последовательный порт UART0 на скорости 1000000 bps 8 бит данных, нет четности.

ROM загрузчик загружает в память IM0 образ вторичного загрузчика размером до 128 Килобайт из внешней памяти в следующем порядке: NOR flash, SD карта, SPI flash. В случае невозможности загрузки из вышеперечисленных устройств загрузчик переходит в “хост-режим” и ожидает загрузки корректного образа вторичного загрузчика через один из отладочных интерфейсов: edcl, jtag.

ROM загрузчик производит начальную базовую настройку TLB для доступа к внутренним памяти IM0 и ROM и основным устройствам. Карта виртуальных адресов сформированная ROM загрузчиком в TLB представлена ниже:

Описание	Начальный адрес	Конечный адрес
IM0 память	0x40000	0x7ffff
ROM память	0xFFFC0000	0xFFFFFFFF

Начальное содержимое TLB таблиц процессорных ядер приведено ниже.

```
// MMU_TLB_ENTRY( ERPN, RPN, EPN, DSIZ, IL1I, IL1D, W, I, M, G, E, UX, UW, UR, SX, SW, SR DULXE, IULXE, TS, TID, WAY, BID, V )
.long MMU_TLB_ENTRY( 0x3ff, 0xfff00, 0xfff00, MMU_TLBE_DSIZ_1MB, 0b1, 0b1, 0b0, 0b1, 0b0, 0b1, MMU_TLBE_E_BIG_END, 0b0,0b0,0b0, 0b1,0b0,0b1, 0b0, 0b0, 0b0, MEM_WINDOW_SHARED, MMU_TLBWE_WAY_UND, MMU_TLBWE_BE_0, 0b1 )
.long MMU_TLB_ENTRY( 0x010, 0x00000, 0x00000, MMU_TLBE_DSIZ_1MB, 0b1, 0b1, 0b0, 0b1, 0b0, 0b0, MMU_TLBE_E_BIG_END, 0b0,0b0,0b0, 0b1,0b1,0b1, 0b0, 0b0, 0b0, MEM_WINDOW_SHARED, MMU_TLBWE_WAY_UND, MMU_TLBWE_BE_1, 0b1 )
.long MMU_TLB_ENTRY( 0x010, 0x10000, 0x10000, MMU_TLBE_DSIZ_256MB, 0b1, 0b1, 0b0, 0b1, 0b0, 0b1, MMU_TLBE_E_LITTLE_END, 0b0,0b0,0b0, 0b0,0b1,0b1, 0b0, 0b0, 0b0, MEM_WINDOW_SHARED, MMU_TLBWE_WAY_UND, MMU_TLBWE_BE_2, 0b1 )
.long MMU_TLB_ENTRY( 0x010, 0x20000, 0x20000, MMU_TLBE_DSIZ_256MB, 0b1, 0b1, 0b0, 0b1, 0b0, 0b1, MMU_TLBE_E_LITTLE_END, 0b0,0b0,0b0, 0b0,0b1,0b1, 0b0, 0b0, 0b0, MEM_WINDOW_SHARED, MMU_TLBWE_WAY_UND, MMU_TLBWE_BE_3, 0b1 )
.long MMU_TLB_ENTRY( 0x010, 0x30000, 0x30000, MMU_TLBE_DSIZ_256MB, 0b1, 0b1, 0b0, 0b1, 0b0, 0b1, MMU_TLBE_E_LITTLE_END, 0b0,0b0,0b0, 0b0,0b1,0b1, 0b0, 0b0, 0b0, MEM_WINDOW_SHARED, MMU_TLBWE_WAY_UND, MMU_TLBWE_BE_4, 0b1 )
```

Алгоритм работы ROM загрузчика приведен на рисунках 1.325 и 1.326.

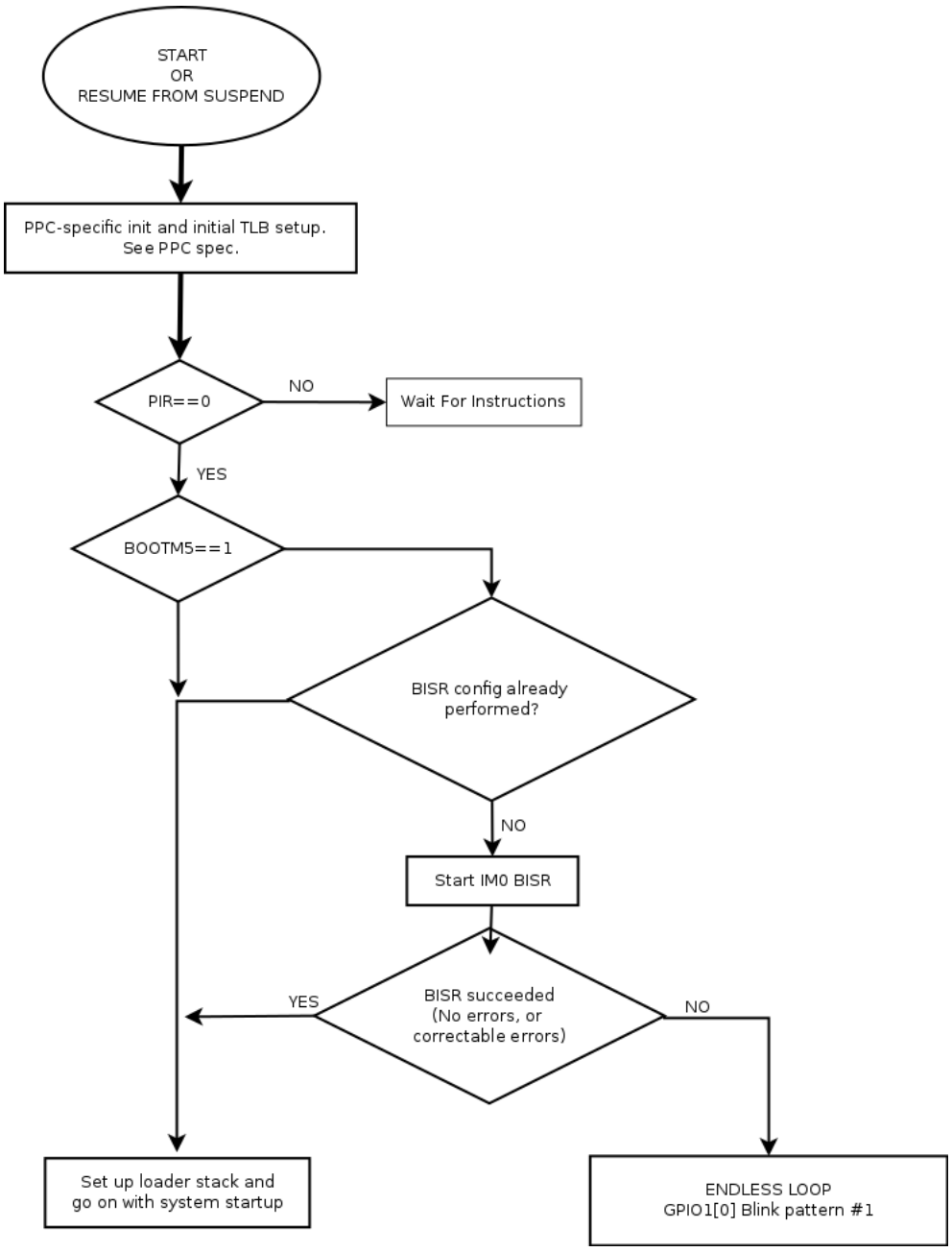


Рисунок 1.325 - Начальная процедура инициализации ROM загрузчика

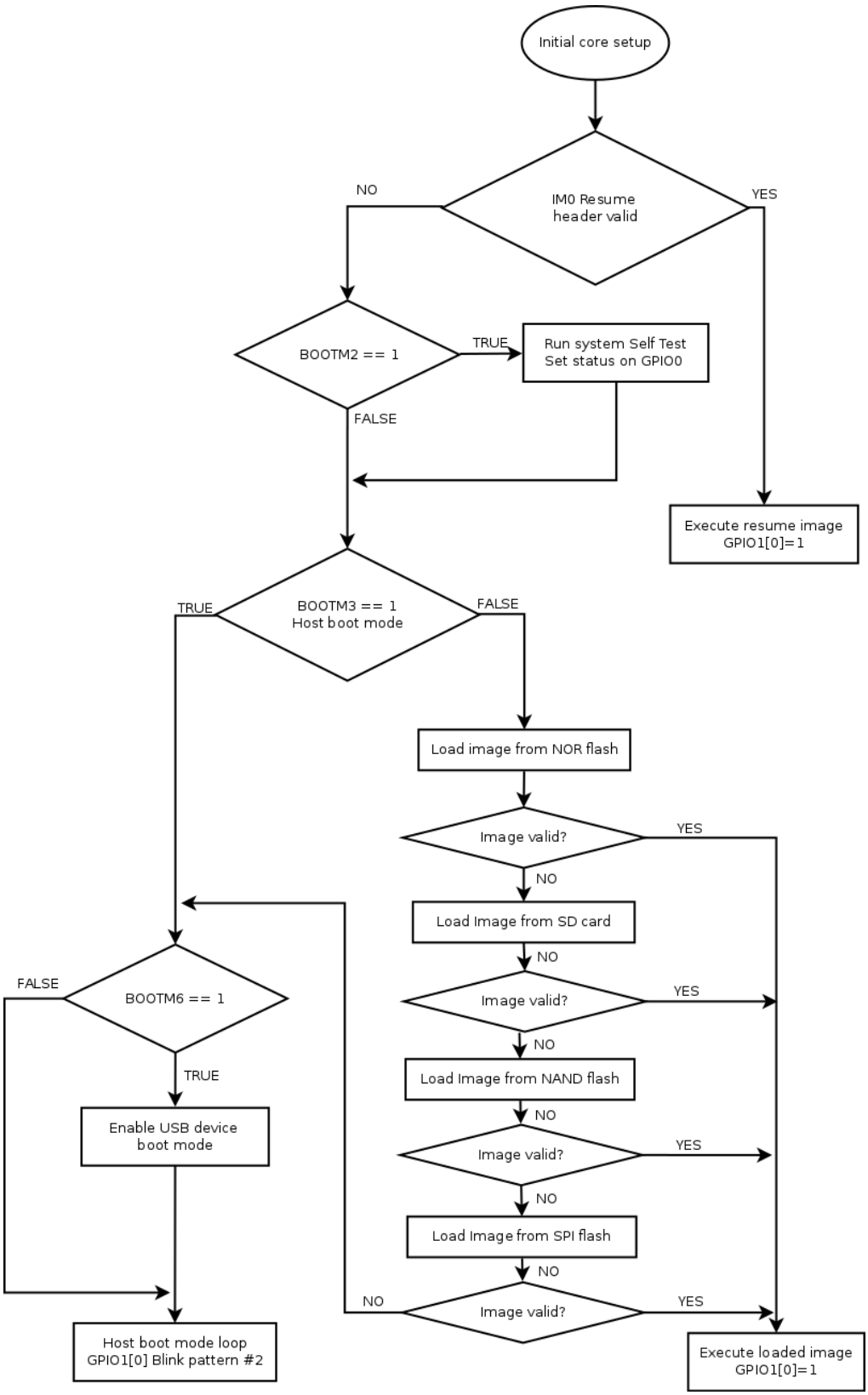


Рисунок 1.326 - Основной алгоритм работы загрузчика

Управление режимами загрузки осуществляется путем фиксации значений на внешних конфигурационных входах SYS_BOOT (см. описание ниже).

Значения выделенные синим цветом обрабатываются аппаратно. Остальные значения SYS_BOOT, выделенные зеленым цветом обрабатываются программно в коде загрузчика.

Таблица 1.856 - Выводы SYS_BOOT, мультиплексированные с портами GPIO

Функция	Варианты конфигурации	Управляющие входы	GPIO
Поменять значения PIR ядер местами	0 – не менять 1 – менять	SYS_BOOT0	GPIO0[0]
Состояние вторичного ядра (с PIR=1) в момент старта	0 – Включено (исполняет bootrom одновременно с первичным ядром) 1 – Отключено	SYS_BOOT1	GPIO0[1]
Включение режима самотестирования при старте	0 – включен 1 – выключен	SYS_BOOT2	GPIO0[2]
Принудительная загрузка в HOST режиме	0 – нормальная загрузка 1 – загрузка в HOST режиме	SYS_BOOT3	GPIO0[3]
Включение/выключение EDCL.	0 – edcl включен 1 – edcl выключен	SYS_BOOT4	GPIO0[4]
Проводить конфигурацию redundancy для памятей IM0 перед стартом загрузчика; Проверять L2C_0 и L2C_1 во время SELFTEST	0 – Не конфигурировать 1 – Конфигурировать	SYS_BOOT5	GPIO0[5]
Конфигурация сетевых интерфейсов для edcl	0 – edcl на 100-мегабитном интерфейсе 1 – edcl на гигабитных интерфейсах	SYS_BOOT6	GPIO0[6]
Зарезервировано	Зарезервировано	SYS_BOOT7	GPIO0[7]
Зарезервировано	Зарезервировано	SYS_BOOT8	GPIO1[0]
SDIO Card Detect (Подключается к разъему SD, в случае отсутствия карты попытка загрузиться с SD не производится)	0 – Карта присутствует 1 – Карта отсутствует	-	GPIO1[1]
SPI SS	Используется как сигнал chip-select при загрузки с SPI Flash	-	GPIO1[2]

Формат образа вторичного загрузчика.

Образ вторичного загрузчика должен иметь корректный заголовок, который будет валидирован ROM загрузчиком. 32-х битные поля заголовка используют порядок байт big endian. Структура заголовка с комментариями приведена ниже.

```
struct bootheader {
  uint32_t magic; /* волшебное число-идентификатор */
  uint32_t length; /* размер данных в байтах, без учета размера этого заголовка.
                      Размер данных в 0 байт является валидным случаем */
  uint32_t entry0;
                /* Адрес точки входа для ядра, исполняющего
           код загрузчика. Сюда будет передано управление после валидации
                */
  uint32_t entry1;
/*                                  Адрес точки входа для вторичного ядра.
                    NULL, если вторичное ядро не требуется.
           */
  uint32_t sum;  /* CRC32 сумма для прочитанных данных. Если длина 0 – CRC не
                    проверяется. Если поле CRC 0 – CRC для данных не проверяется. */
  uint32_t hdrsum; /* Контрольная сумма всех полей заголовка */
  char       imagedata[]; /* Данные */
}
```

Образ считается валидным если все эти условия выполнены (в порядке проверки):

Образ имеет корректный MAGIC

Образ имеет корректную контрольную сумму заголовка

Контрольная сумма для данных совпадает с контрольной суммой в заголовке

Образ загружается в начало внутренней SRAM памяти.

В случае, если в поле entryX вместо адреса находится 0, процессорное ядро переходит в бесконечный цикл.

Значения поля magic для различных образов приведены в таблице ниже

Значение поля magic	Описание
0xbeefc0de	Идентифицирует валидный образ при загрузке с ПЗУ
0xdeadc0de	Идентифицирует валидный образ при загрузке через edcl/jtag

Для добавления заголовка к файлу воспользуйтесь утилитой rumboot-packimage.

3 Хранение

Минимальный срок сохраняемости микросхем при их хранении:

- в отапливаемом хранилище или в хранилище с регулируемой влажностью и температурой по ГОСТ 15150 должен быть не менее 25 лет,

- в неотапливаемом хранилище - 16,5 лет,

Хранение микросхем на открытой площадке в упаковке фирмы-изготовителя или ЗАО НТЦ "Модуль" не допускается. Хранение микросхем, разрешенных для применения в аппаратуре единичного производства и образующих страховой запас, должно осуществляться в упаковке фирмы-изготовителя или ЗАО НТЦ "Модуль" только в отапливаемых хранилищах или хранилищах с регулируемой влажностью и температурой.

Срок сохраняемости исчисляют с даты изготовления, указанной на микросхеме.

4 Транспортирование

Транспортирование по ГОСТ 23088-80.

5 Утилизация

Микросхемы, по истечении срока хранения или не прошедшие предварительные испытания подлежат утилизации в установленном на предприятии порядке.

Приложение А

Детальная структурная схема ЦПУ подсистемы

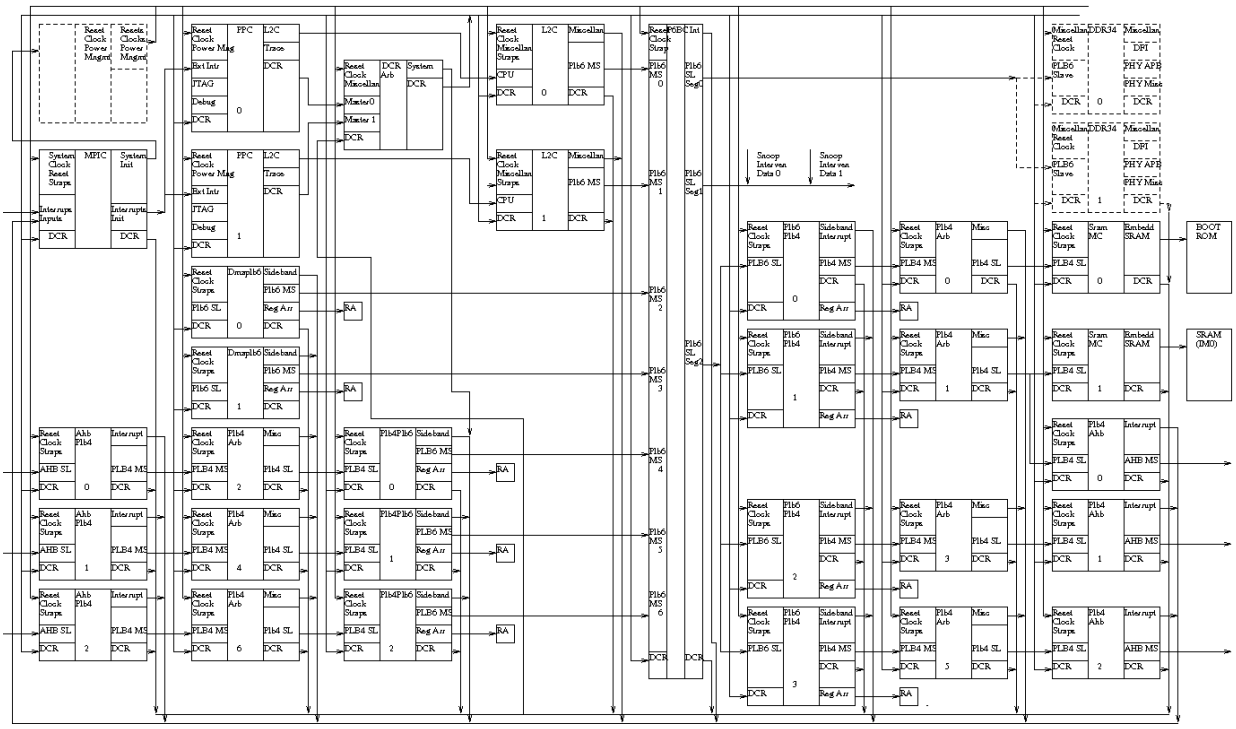


Рисунок А1 - Детальная структурная схема ЦПУ подсистемы

Приложение Б

Ядро процессора PowerPC 470S

Встроенное процессорное ядро PowerPC 476FP - это суперскалярный 32-битный процессор с сокращенным набором команд (RISC), с обработкой 4 инструкций за такт и с 6 конвейерами (конвейер L для операций с плавающей запятой [FP] и конвейер L для целочисленных операций совмещены). Ядро поддерживает архитектуру системы команд (ISA) Power версии 2.05. Архитектурная гибкость этого ядра улучшает характеристики специализированных интегральных схем (ASIC). Ядро также поддерживает когерентность памяти, что позволяет повысить эффективность мультипроцессорных решений ASIC, а также повысить масштабируемость для новых применений в таких областях, как проводная связь, хранение данных и всеобщая информатизация. На рисунке Б.1 показана общая организация процессорного ядра.

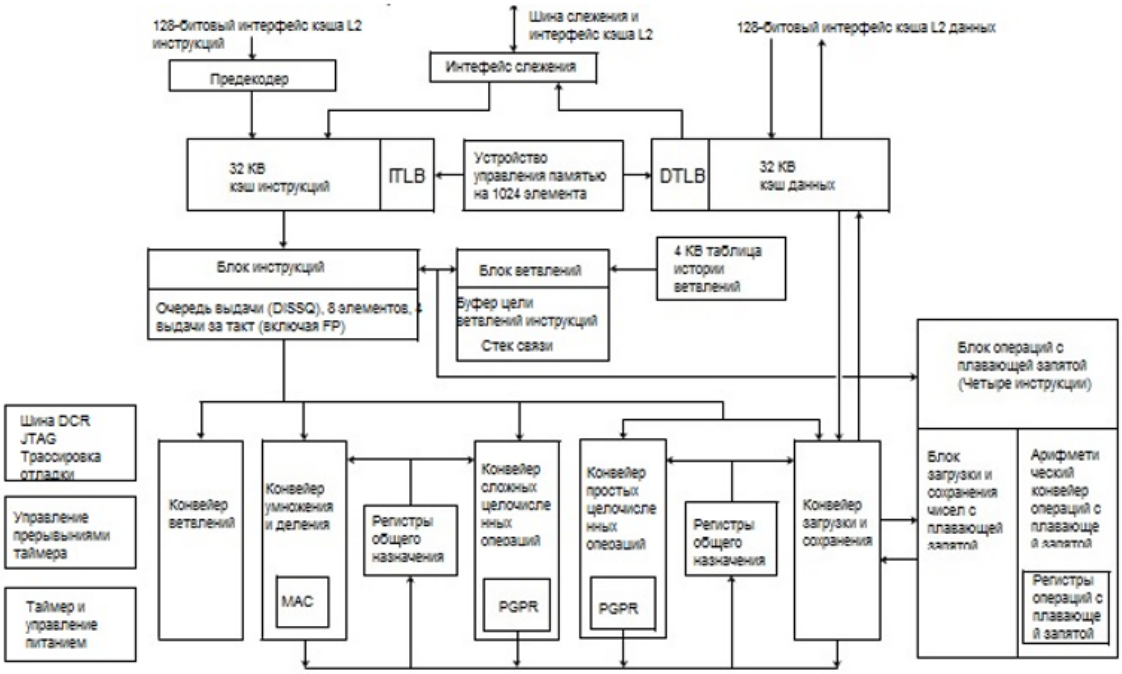


Рисунок Б.1 - Блок-схема встроенного процессорного ядра PowerPC 476FP

DCR Регистр управления устройством
DISSQ Очередь выдачи декодирования
DTLB Буфер ассоциативной трансляции данных
D-Data Данные кэша данных
FP Плавающая запятая
GPR Регистры общего назначения
ITLB Буфер ассоциативной трансляции инструкций
I-Data Данные кэша инструкций
JTAG Joint Test Action Group
KB 1024 байта
L2 Кэш уровня 2
MAC Умножение с накоплением
PGPR Буферы пре-GPR

Общие характеристики

Процессорное ядро PowerPC 476FP обладает следующими характеристиками:

Архитектура с выдачей четырех инструкций за такт (сложность декодирования и выдачи [DISS])

Пять конвейеров и отдельный конвейер для арифметики с плавающей запятой:

Конвейер ветвлений

L конвейер (для операций загрузки и сохранения)

J конвейер (для простых арифметических и логических операций)

Конвейер инструкций (конвейер простых и сложных инструкций, конвейер прочих инструкций)

Конвейер умножения и деления

Конвейер выполнения FP

Двухсцикловый конвейеризованный доступ к кэшу

Тегирование реальных адресов для кэша инструкций и кэша данных

Индексирование кэша инструкций виртуальным адресом

Индексирование кэша данных реальным адресом

Кэш инструкций и данных с возможностью слежения

Объединенный буфер ассоциативной трансляции (UTLB) на 1024 элемента

Временные буфера промежуточного регистра общего назначения (PGPR) для захвата и удержания результатов до момента завершения, когда результаты передаются в регистры общего назначения (GPRs)

Разрешение ранней доставки инструкций к модулю операций с плавающей запятой, поскольку все инструкции предварительно декодируются

Характеристики управления питанием

Следующие конструктивные особенности минимизируют потребляемую мощность процессорного ядра PowerPC 476FP:

Все защелки синхронные, так что простаивающие функции не потребляют энергию.

Все не исполняемые и простаивающие функции отключаются.

Статическая память с произвольной выборкой (SRAM) разделена таким образом, что включается или выбирается только требуемая область SRAM.

Доступны режим энергосбережения и спящий режим.

Центральная логика и модуль операций с плавающей запятой имеют отдельное разрешение подачи тактового сигнала.

Режимы управления питанием

Режимы управления питанием для ядра PowerPC 476FP: спящий режим CPU, режим ожидания CPU, режим отключения CPU.

Спящий режим CPU имеет следующие характеристики:

Тактовый генератор CPU выключен при помощи деактивизации сигнала включения тактового генератора, но таймер продолжает работать, чтобы поддерживать синхронизацию.

CPU может быть пробужден включением тактового генератора при помощи прерываний, например, внешнего прерывания, прерывания декрементора (DEC), прерывания сторожевого таймера или таймера фиксированного интервала (FIT) - если в приложении реализованы эти механизмы.

Спящий режим может управляться динамически и произвольным образом.

Поскольку процессор не поддерживает когерентность кэша или модуля управления памятью, то необходимо выполнить следующие действия:

Кэш L2 должен находиться в состоянии, когда он не устанавливается процессором или специальным DCR.

Кэш процессора должен быть инвалидирован инструкциями инвалидации кэша данных (dci) и инвалидации кэша инструкций (ici) в начале процесса пробуждения.

MMU процессора должен быть инвалидирован при помощи инструкции записи в буфер ассоциативной трансляции (tlbwe), кроме одного (или необходимого количества) элемента, содержащего исключение или прерывание.

Режим энергосбережения CPU имеет следующие характеристики:

При использовании режима энергосбережения процессор находится в состоянии ожидания или в состоянии останова.

Тактовый генератор CPU продолжает работать, но инструкции не выполняются.

Режим энергосбережения позволяет процессору обрабатывать исключения и прерывания, однако, когда выдается инструкция возврата из прерывания (rfi), процессор возвращается в режим энергосбережения.

Процессор поддерживает синхронизацию кэша и MU.

В режиме отключения CPU питание ядра PowerPC 476FP отключено.

В таблице Б.1 представлены режимы управления питанием ядра PowerPC 476FP.

Таблица Б.1 - Режимы экономии энергии PowerPC 476FP

Режим	Тактовый генератор ядра	Питание ядра	Состояние ядра	Определение	Режим работы	Эффект
Спящий	Выкл.	Вкл.	Да	Низкое потребление энергии Таймеры включены	Выкл.	Синхронизируется тактовым генератором
Энергосбережение	Вкл.	Вкл.	Да	Пониженное потребление питания Таймер включен	Ожидание	Таймер работает, прерывание обслуживается
Отключение	Выкл.	Выкл.	Нет	Питание отключено	Выкл.	Питание отключено

В таблице Б.2 представлено влияние переключения частот на режимы управления питанием PowerPC 476FP.

Таблица Б.2 - Переключение частот PowerPC 476FP

Режим	Тактовый генератор ядра	Пита-ние ядра	Состояние ядра	Определение	Режим работы	Допустимый диапазон
Сдвиг	Вкл.	Вкл.	Да	Переключение частот	Работает	Отношение тактовых частот CPU, L2, PLB6 и DCR остается постоянным □□переключение без сбоев

В таблице Б.3 приведены примеры переключения частоты.

Таблица Б.3 - Примеры переключения частоты

	Тактовая частота CPU	Тактовая частота L2	Тактовая частота PLB6
Начальная	1600 MHz	800 MHz	800 MHz
Запись супервизора (SW) 1	800 MHz	400 MHz	400 MHz
SW 2	533+ MHz	266,5 MHz	266,5 MHz
SW 3	400 MHz	200 MHz	200 MHz

Процедуры управления питанием

Процедуры управления питанием состоят из перевода ядра PowerPC 476FP в спящий режим, в режим энергосбережения, а также пробуждения процессора из этих режимов.

Спящий режим CPU

Для перевода PowerPC 476FP в спящий режим, необходимо выполнить следующие действия.

Примечание: Убедиться, что ядро не выполняет инструкции сохранения.

Выдать инструкцию системного вызова (sc), чтобы перевести процессор в привилегированный режим.

Убедиться, что ядро PowerPC 476FP не выполняет инструкции сохранения, затем выдать инструкцию синхронизации инструкций (isync), затем синхронизировать (msync).

Установить кэш L2 для спящего режима, установив L2SLEEPREQ[31] = ‘1’, затем выдать isync и перейти к инструкциям регистра специального назначения (mtmsr), чтобы установить регистр управления процессором (MSR)[WE] = ‘1’. Это переводит процессор в режим ожидания (аналогичный режиму энергосбережения).

Процессор активизирует сигнал запроса спящего режима ядра, интерфейс управления тактовым генератором и питанием (CPM) деактивирует сигнал разрешения тактового генератора CPU, и процессор переходит в спящий режим.

Режим энергосбережения CPU

Для перевода процессора в режим энергосбережения, необходимо выполнить следующие действия:

Установить MSR[WE] = ‘1’ либо вызовом управляющей программы, которая устанавливает MSR (MSR) посредством выдачи инструкции mtmsr, либо установкой регистра Save/Restore Register 1 (SRR1)[WE] = ‘1’ при обработке прерывания.

В режиме энергосбережения процессор прекращает выборку инструкций и их выполнение и переходит в состояние останова. Любое исключение или прерывание приводит к пробуждению процессора.

Для пробуждения процессора необходимо сбросить SRR1[WE] в ‘0’. В противном случае процессор остается в режиме энергосбережения после выдачи инструкции rfi.

Пробуждение процессора

Для пробуждения процессора необходимо выполнить следующие действия:

Генерировать одно из следующих прерываний: внешнее, DEC, FIT или сторожевого таймера.

Разрешить тактовый генератор CPU, активировав сигнал разрешения тактового генератора (CPMC476CLKEN). Это вызовет программу обработки прерываний.

Установить SRR1[WE] = ‘0’.

Инвалидировать элементы MMU за исключением тех, которые нужны для обработки прерываний и исключений.

Инвалидировать кэш инструкций L1 (I-кэш) выдачей инструкции ici.

Инвалидировать кэш данных L1 (D-кэш) выдачей инструкции dci (учитывать поле CT инструкций).

Подготовить кэш L2 к пробуждению, установив регистр L2 Sleep Request Register (L2SLEEPREQ)[31] = ‘0’ (См.L2 Cache Core Controller Databook), затем выдать инструкцию isync .

Обработать прерывание и выдать инструкцию rfi. Теперь процессор будет пробужден.

Дополнительные подробности о реализации и программировании спящего режима, режима энергосбережения или пробуждения процессора можно узнать в службе поддержки IBM PowerPC.

Набор реализованных инструкций

Все инструкции Power ISA Version 2.05 из Book I, Book II и Book III-E реализованы в процессорном ядре PowerPC 476FP. Поддерживаются следующие категрии инструкций:

- Базовые
- Встроенные
- Встроенная отладка кэша
- Встроенная инициализация кэша
- Встроенные с прямым порядком байтов
- Встроенная фиксация кэша
- Когерентность памяти

Средства тестирования и отладки

Подобно процессорным ядрам PowerPC 4xx, процессорное ядро PowerPC 476FP снабжено интерфейсами RISCWatch и Joint Test Action Group (JTAG), позволяющим реализовать следующие функции:

- Сброс процессорного ядра
- Выключение, останов и запуск процессорного ядра
- Выполнение операций отладки
- Отслеживание состояния и работы процессорного ядра, а также других ядер и устройств в ASIC

Дополнительные возможности отладки и отслеживания состояния обеспечивают мониторинг и отладку процессорных ядер, а также других ядер в многопроцессорной однокристальной системе. Реализованы следующие средства, определенные Power.org Common Debug Interface Technical Committee:

- Возможность мониторинга многопроцессорного регистра статуса отладки (DBSR) под управлением JTAG
- JTAG элементы управления останова и запуска

Эти функции обеспечивают средства мониторирования отдельных битов DBSR под управлением JTAG. Процессор также может быть запущен и остановлен под управлением JTAG. Дополнительную информацию об использовании средств JTAG.

Обзор модуля операций с плавающей запятой

FPU - это модуль конвейерных математических вычислений с двойной точностью, присоединенный к процессорному ядру. FPU соответствует стандарту ANSI/IEEE Standard 754-1985, IEEE Standard for Binary Floating-Point Arithmetic.

FPU PowerPC 476FP имеет следующие характеристики:

- Соответствие стандарту операций с плавающей запятой ANSI/IEEE 754-1985:
  - Стандарт операций с плавающей запятой одинарной точности
  - Стандарт операций с плавающей запятой двойной точности
- Набор инструкций PowerPC для операций с плавающей запятой
- Соответствие Book E: Enhanced PowerPC Architecture
- Суперскалярный режим работы с независимыми блоками загрузки/сохранения и выполнения операций с плавающей запятой

- Шестиступенчатый суперконвейер для выполнения арифметических операций с плавающей запятой
  - Дополнительное деление ступеней
  - Дополнительные ступени для денормализованных вычислений

Обзор кэша инструкций

Модуль кэша инструкций (ICU) разделен на несколько субмодулей: массив кэша инструкций, блок управления кэшем инструкций, буфер ассоциативной трансляции инструкций (ITLB), таблица истории ветвлений и блок выборки инструкций.

Массив кэша инструкций состоит из стандартных SRAM с данными и тегами инструкций, организованных как 4-входовый ассоциативный кэш. Шина данных инструкций ядра имеет разрядность 128 или 256 бит и загружается в буфер заполнения командной магистрали емкостью 8 слов. Метод замещения - 6-битный алгоритм "дольше всех не использовавшийся" (LRU), с возможностями фиксации или разделения кэша.

ITLB, блок управления кэшем инструкций и блок выборки инструкций - полностью синтезируемые логические блоки. Они содержат как тракт данных, так и управляющую логику.

ITLB - это 8-входовый полностью ассоциативный массив. Его главное назначение - увеличить производительность и уменьшить конфликты TLB между выборкой инструкций и загрузкой/сохранением данных. Каждый элемент ITLB содержит информацию трансляции для страницы. Процессор использует эту информацию, чтобы транслировать адреса выборки инструкций, когда MSR[IS] = ‘1’.

Модуль истории ветвлений улучшает предсказание ветвлений, отслеживая наиболее вероятный результат недавно выполненных программных ветвлений.

Обзор блока кэша данных

Блок кэша данных (DCU) состоит из следующих трех субблоков: массивы кэша данных, управление кэшем данных и TLB данных (DTLB).

Субблок массивов кэша данных содержит три массива: LRU, тегов и данных. Массивы тегов и данных - это стандартные SRAM. Массив LRU - это меньший по размеру массив двухпортовых регистров. И массив тегов, и массив данных представляют собой 4-входовый ассоциативный кэш и имеют конвейерный 2-цикловый доступ. DCU включает алгоритм замещения LRU, который использует 6-битный вектор возраста в сочетании с блокировкой канала для определения лучшего кандидата на замещение. DCU может получать с шины одновременно 256 битов данных чтения и может отправлять до 128 битов данных записи. Кэш данных неблокируемый. Когерентность кэша поддерживается при помощи уровня 2 (L2) использованием режима сквозной записи.

Субблок управления кэшем включает все элементы управления конвейера DCU, арбитраж кэша, регистры специального назначения (SPR) и конвейер слежения. Он управляет большей частью потока тракта данных и работой.

DTLB - это 8-элементный полностью ассоциативный кэш, который использует исполнительный адрес для быстрого вычисления реального адреса. Доступ к нему осуществляется параллельно с доступом к остальным тремя массивам. При возникновении промаха DTLB делается запрос к UTLB, чтобы вычислить реальный адрес.

Обзор модуля управления памятью

Модуль управления памятью (MMU) обеспечивает управление кэшем, защиту доступа и трансляцию адреса. MMU содержит UTLB, управляющую логику и регистры, поддерживающие UTLB. MMU взаимодействует с исполнительным блоком (EU), блоком инструкций (IU), ICU, DCU и следящим интерфейсом TLB. Интерфейс EU выполняет инструкции TLB, такие как чтение элемента буфера ассоциативной трансляции (tlbre), tlbwe, индекс поиска буфера

ассоциативной трансляции (tlbsx), и инвалидация транзакции шины буфера ассоциативной трансляции (tlbivax).

Интерфейс IU обеспечивает трансляцию селекторных бит области трансляции (TS) и области данных (DS) для запроса поиска от DCU, ICU или слежения TLB. ICU генерирует запрос поиска к UTLB при промахе ITLB. Аналогичным образом, DCU генерирует запрос поиска к UTLB при промахе DTLB. MMU выполняет арбитраж DCU, EU и следящего интерфейса, предоставляет данные для каждого запроса.

MMU управляется программой, но для элементов замещения предоставляется аппаратная логика поддержки. Программа записывает элементы в UTLB, так чтобы их можно было считать, используя хэш-функцию, описанную в Разделе 4.3.2 Индексная хэш-адресация UTLB, на стр. 106. Дополнительный режим MMU типа Freescale не поддерживается.

Таймеры

Процессорное ядро PowerPC 476FP содержит три таймера: базовый таймер, декрементор (DEC), таймер фиксированного интервала (FIT) и сторожевой таймер.

Базовый таймер - это 64-битный счетчик, который инкрементируется с частотой либо равной тактовой частоте процессорного ядра, либо под управлением отдельного асинхронного счетчика времени, прилагаемого к ядру. При обнулении базового таймера прерывание не генерируется.

DEC - это 32-битовый регистр, который декрементируется с той же скоростью, что и базовый таймер. Пользователь загружает в DEC определенное значение, чтобы сформировать требуемый интервал. Когда значение регистра уменьшается до нуля, устанавливается бит статуса и вырабатывается исключение, которое может извещать программу. Опционально DEC можно запрограммировать на автоматическую перезагрузку значения, содержащегося в регистре Auto-Reload Register (DECAR), после чего декрементирование DEC возобновится.

FIT может генерировать периодические прерывания на основе переключения 1 выбранного пользователем бита из 4 битов базового таймера. Когда значение выбранного бита изменяется из ‘0’ в ‘1’, то устанавливается бит статуса и вырабатывается исключение, которое может извещать программу.

Сторожевой таймер также генерирует периодическое прерывание на основе переключения выбранного пользователем бита базового таймера. Пользователь может выбирать один из четырех интервалов для сторожевого таймера. При первом переключении из ‘0’ в ‘1’ выбранного бита базового таймера сторожевой таймер генерирует исключение, которое может извещать программу. Сторожевой таймер также можно конфигурировать для инициализации аппаратного сброса, если второе переключение выбранного бита базового таймера произойдет раньше, чем будет обслужено первое исключение сторожевого таймера. Эта возможность расширяет степень восстанавливаемости при потенциальном зависании системы.

Модель программирования

Адресация памяти

В качестве реализации Power Instruction Set Architecture (ISA) Version 2.05, ядро PowerPC 476FP реализует однородное пространство 32-битных исполнительных адресов. Исполнительные адреса расширяются до виртуальных адресов, а затем транслируются в 42-битные (4 TB) действительные адреса модулем управления памятью. Организация пространства действительных адресов в пространство физических адресов описано в руководствах пользователя продуктов уровня микросхемы, в состав которых входит в ядро PowerPC 476FP.

Ядро PowerPC 476FP генерирует исполнительный адрес при выполнении инструкции обращения к памяти, ветвления, управления кэшем или буфером ассоциативной трансляции (TLB), или при выборке следующей последовательной инструкции.

Операнды, хранимые в памяти

Байты в памяти нумеруются последовательно, начиная с 0. Каждый номер - это адрес соответствующего байта.

Операнды, хранимые в памяти, обращение к которым происходит с помощью инструкций загрузки/сохранения целочисленных данных, могут представлять собой байт (8 бит), полуслово (16 бит), слово (32 бита) и двойное слово (64 бита); для многоадресных и строковых инструкций загрузки/сохранения это может быть последовательность слов или байт. Операнды, хранимые в памяти, обращение к которым происходит с помощью инструкций загрузки/сохранения данных с плавающей запятой, могут представлять собой байт, полуслово, слово, двойное слово или счетверенное слово. Адрес операнда, хранимого в памяти - это адрес его первого байта (то есть. байта с наименьшим номером). Порядок следования байтов может быть прямым или обратным, в зависимости от атрибута памяти.

Длина операнда неявно задана для каждого типа скалярных инструкций обращения к памяти (то есть, каждого типа инструкций обращения к памяти, отличных от многоадресных и строковых инструкций загрузки/сохранения). Операнд такого типа скалярной инструкции обращения к памяти имеет естественную границу выравнивания, равную длине операнда. Поэтому естественный адрес операнда - это кратное целое длины операнда. Операнд, хранимый в памяти, называется выровненным, если он выровнен по своей естественной границе; в противном случае он называется невыровненным.

В таблице Б.4 представлены инструкции обращения к памяти для операндов, хранимых в памяти.

Таблица Б.4 - Определения операндов, хранимых в памяти

Тип инструкции обращения к памяти	Длина операнда	Адрес[28:31] если выровненный
Байт (или строка)	8 бит	‘xxxx’
Полуслово	2 байта	‘xxx0’
Слово (или многоадресная)	4 байта	‘xx00’
Двойное слово (только FP)	8 байт	‘x000’
Примечание - Значок x в позиции бита адреса указывает, что бит может принимать значение ‘0’ или ‘1’ независимо от состояния других битов адреса.		

Выравнивание операнда исполнительного адреса некоторых инструкций обращения к памяти может влиять на производительность, и в некоторых случаях может привести к формированию исключения выравнивания. Для таких инструкций обращения к памяти лучшая производительность достигается, когда операнды памяти выровнены. В таблице Б.5 приводится воздействие выравнивания на типы инструкций обращения к памяти, для которых существует этот эффект. Если в таблице нет того или иного типа инструкций, это значит, что для них эффекта выравнивания отсутствует.

Таблица Б.5 - Эффекты выравнивания на инструкции обращения к памяти

Тип инструкции обращения к памяти	Эффекты выравнивания
Целочисленная загрузки/сохранения полуслова	Разбивается на два обращения по одному байту, если пересекает 8-байтовую границу; в противном случае эффекта нет.
Целочисленная загрузки/сохранения слова	Разбивается на два обращения, если пересекает 8-байтовую границу; в противном случае эффекта нет.
Многоадресная или строковая загрузки/сохранения	Разбивается на серию 4-байтовых обращений, пока не будет достигнут последний байт или 8-байтовая граница. Если после достижения 8-байтовой границы остаются еще байты, возобновляется обращение по 4 байта до достижения последнего байта или следующей 8-байтовой границы, и так далее.
Загрузка/сохранение слова FP	Исключение выравнивания, если пересекается граница слова, в противном случае нет эффекта (см. примечание).
Загрузка/сохранение двойного слова FP	Исключение выравнивания, если пересекается граница двойного слова, в противном случае нет эффекта (см. примечание).
Примечание: Блок операций с плавающей запятой может указать, что EA для конкретной инструкции FP загрузки или сохранения должен быть выровнен по границам операнда, или по границам слова. Если FPU указывает это требование, и вычисленный EA ему не удовлетворяет, то ядро PowerPC 476FP генерирует исключение выравнивания. Или FPU может указать, что EA для конкретной инструкции FP загрузки или сохранения должен принудительно выравниваться, игнорируя соответствующее количество младших бит EA и обрабатывая инструкции FP загрузки или сохранения так, как будто эти биты равны '0'. В байте, полуслове, слове, двойном слове и счетверенном слове инструкции FP загрузки или сохранения игнорируются 0, 1, 2, 3 и 4 младших бита EA.	

Инструкции управления кэшем обращаются к блоковым операндам кэша; для ядра PowerPC 476FP размер блока кэша составляет 32 байта. Однако исполнительный адрес, вычисленный инструкциями управления кэшем, не требуется выравнивать по границам блоков кэша. Вместо этого архитектура указывает соответствующие младшие биты исполнительного адреса (биты 27:31 для ядра PowerPC 476FP), которые игнорируются при выполнении этих инструкций.

Аналогичным образом, инструкции управления TLB обращаются к страничным операндам и, как определено размером страницы, соответствующие младшие биты исполнительного адреса игнорируются при выполнении этих инструкций.

Однако операнды инструкций сохранения всегда имеют длину 4 байта, и поэтому исполнительный адрес, вычисленный инструкциями ветвления, всегда выровнен по словам.

Вычисление исполнительного адреса

Если для инструкции обращения к памяти сумма исполнительного адреса и длины операнда превышает максимальную длину исполнительного адреса 232–1 (то есть, сам операнд, хранимый в памяти, пересекает границу максимального адреса), то в соответствии с архитектурой результат операции будет неопределенным. Ядро PowerPC 476FP выполняет операцию так, как будто операнд, хранимый в памяти, вернулся к началу, от максимального исполнительного адреса к исполнительному адресу 0. Тем не менее, программа не должна зависеть от этого поведения, чтобы она могла применяться с другими исполнениями, которые обрабатывают данный сценарий иначе. Соответственно, программа должна следить за тем, чтобы операнды, хранимые в памяти, не пересекали границу максимального адреса.

Примечание: Поскольку инструкции - это слова, и исполнительные адреса инструкций всегда соответствуют границам слов, операнд инструкции не может пересекать границу слова, в том числе границу максимального адреса.

Арифметика исполнительного адреса, которая вычисляет начальный адрес для операндов сохранения, возвращается от максимального адреса к адресу 0 для всех вычислений исполнительного адреса, за исключением выборки следующей последовательной инструкции.

Режимы адресации обращения к данным

Ядро PowerPC 476FP поддерживает два режима адресации обращения к данным:

Режим адресации база + смещение (D-режим:)

16-битовое поле D заполняется знаком операнда и прибавляется к содержимому регистра общего назначения (GPR), указанному RA, или к нулю, если RA = ‘0’; младшие 32 бита суммы образуют исполнительный адрес операнда, хранимого в памяти.

Режим адресации база + индекс (X-режим:)

Содержимое GPR, указанное RB (или значение 0 для непосредственной загрузки последовательности слов [lswi] и непосредственного сохранения последовательности слов ([stswi]) добавляется к содержимому GPR, указанному RA, или к 0, если

RA = ‘0’; младшие 32 бита суммы образуют исполнительный адрес операнда, хранимого в памяти.

Режимы адресации обращения к инструкциям

Ядро PowerPC 476FP поддерживает четыре режима адресации обращения к инструкциям:

Инструкции ветвления типа I (безусловные)

24-битное поле LI соединяется справа с ‘00’, заполняется знаком операнда, а затем добавляется либо к адресу инструкции ветвления, если абсолютный адрес (AA) поля инструкции равен 0), или к 0, если AA = ‘1’; младшие 32 бита суммы образуют исполнительный адрес следующей инструкции.

Выполненные инструкции ветвления формата B:

14-битное поле смещения ветвления (BD) соединяется справа с ‘00’, заполняется знаком операнда, а затем добавляется либо к адресу инструкции ветвления, если AA = ‘0’, либо к 0, если AA = ‘1’; младшие 32 бита суммы образуют исполнительный адрес следующей инструкции.

Выполненные инструкции ветвления формата XL:

Содержимое бит 0:29 регистра связи (LR) или бит 32:61 регистра счетчика (CTR) соединяется справа с ‘00’, в результате чего образуется 32-битный исполнительный адрес следующей инструкции.

Выборка следующей последовательной инструкции (включая не выполненные инструкции ветвления):

Значение 4 добавляется к адресу текущей инструкции, чтобы получить 32-битный исполнительный адрес следующей инструкции. Если адрес текущей инструкции x‘FFFF FFFC’, то ядро PowerPC 476FP возвращает следующую последовательную инструкцию к адресу 0. Такое поведение не требуется архитектурой, которая в данном случае считает адрес следующей инструкции неопределенным. Поэтому программа не должна зависеть от этого поведения, чтобы она могла применяться с другими исполнениями, которые обрабатывают данный сценарий иначе. Соответственно, если программе требуется перейти границу максимального адреса и вернуться к адресу 0, она должна поместить на границу безусловный переход со смещением 4.

В дополнение к четырем режимам адресации обращения к инструкциям, для инструкций ветвления характерно следующее поведение:

Любая инструкция ветвления с разрядом выходного переноса равным ‘1’:

Значение 4 добавляется к адресу текущей инструкции, и младшие 32 бита результата помещаются в LR. Подобно сценарию для выборки следующей последовательной инструкции, если адрес инструкции ветвления x‘FFFF FFFC’, то результат, помещенный в LR, не определен архитектурой, хотя и в этом случае ядро PowerPC 476FP возвращает измененное значение LR к адресу 0. И в этом случае программа не должна зависеть от этого поведения, чтобы она могла применяться с другими исполнениями, которые обрабатывают данный сценарий иначе.

Порядок следования байтов

Если скаляры (отдельные элементы данных и инструкции) неделимы, то к ним не применимо понятие порядок следования байтов. Бессмысленно говорить о порядке битов или групп битов внутри самой маленькой адресуемой единицы хранения, потому что такой порядок ни о чем не говорит. Только в том случае, когда скаляры, которые программист и процессор считают неделимыми элементами, могут содержать более одной единицы хранения, возникает вопрос порядка следования.

Для системы, в которой самая маленькая адресуемая единица хранения представляет собой 64-битовое двойное слово, не возникает вопроса о порядке следования байт внутри двойного слова. Все пересылки отдельных скаляров между регистрами и памятью представляют собой двойные слова, и адрес байта, содержащего старшие 8 бит скаляра, не отличается от адреса байта, содержащего любую другую часть скаляра.

Для Book III-E Enhanced PowerPC Architecture, как и для большинства архитектур современных компьютеров самая маленькая единица хранения - это 8-битовый байт. Многие скаляры представляют собой полуслова, слова или двойные слова, состоящие из групп байтов. Когда скаляр длиной в одно слово пересылается из регистра в память, то он занимает четыре последовательных однобайтовых адреса. Таким образом становится важно определить последовательность однобайтовых адресов по отношению к значению скаляра: какой байт содержит старшие 8 бит скаляра, какой байт содержит следующие 8 бит скаляра, и так далее.

С учетом того, что скаляр содержит множество байт, выбор порядка следования байтов в целом произволен. Существует 4□□□□24 способа определить порядок следования 4 байтов в слове, но смысл имеют только два из этих способов:

Порядок следования, при котором старший адрес присваивается 8 старшим битам (крайним слева) скаляра, следующий последовательный адрес - следующим 8 старшим битам, и так далее.

Этот порядок называется обратным порядком, поскольку старшие разряды (с наибольшим весом) скаляра, считающиеся двоичным номером, первыми помещаются в память. IBM eServer™ pSeries® и IBM zSeries® - это примеры архитектуры компьютера, которая использует этот порядок следования байтов.

Порядок следования, при котором старший адрес присваивается 8 младшим битам (крайним справа) скаляра, следующий последовательный адрес - следующим 8 младшим битам, и так далее.

Этот порядок называется прямым порядком, поскольку младшие разряды (с наименьшим весом) скаляра, считающиеся двоичным номером, первыми помещаются в память. he Intel® x86 - это пример архитектуры компьютера, которая использует этот порядок следования байтов.

Power ISA поддерживает и прямой, и обратный порядок следования байтов для инструкций обращения как к адресам, так и к данным. Использование того или иного порядка определяется постранично атрибутом порядка следования (E), который представляет собой поле элемента TLB для страницы. Атрибут порядка следования устанавливается в ‘0’ для страницы с обратным порядком следования байтов, и в ‘1’ для страницы с прямым порядком следования байтов.

Примеры отображения структуры

Следующая структура на языке C, s, содержит набор скаляров и строку символов. В комментариях указаны значения, присвоенные каждому элементу структуры; эти значения определяют, как байты, составляющие каждый структурный элемент, отображаются в памяти.

```
struct {
int a;     /* x‘1112_1314’ word */
long long b;        /* x‘2122_2324_2526_2728’ doubleword */ char *c;            /* x‘3132_3334’ word */
char d[7];          /* 'A','B','C','D','E','F','G' array of bytes */ short e;    /* x‘5152’ halfword */
int f;     /* x‘6162_6364’ word */
} s;
```

Правила отображения структуры языка C разрешают использование заполнения (пропущенных байт), чтобы выровнять скаляры по требуемым границам. В разделах

Распределение с обратным порядком байтов и Распределение с прямым порядком байтов показаны примеры отображения структуры, где каждый скаляр выровнен по своей естественной границе. Это выравнивание предполагает заполнение 4 байт между a и b, 1 байта между d и e, и 2 байт между e и f. В распределении с обратным порядком байт и в распределении с прямым порядком байт заполнение одинаково.

Распределение с обратным порядком байтов

Распределение с обратным порядком байтов структуры S представлено в таблице Б.6. В распределении структуры данные выделены жирным шрифтом. Шестнадцатеричные адреса показаны под данными, которые хранятся по этим адресам. Содержание каждого байта, как определено структурой S, показано как (шестнадцатеричное) число или символ (для строковых элементов). Закрашенные ячейки соответствуют заполненным байтам.

Таблица Б.6 - Распределение структуры S с обратным порядком байтов

11	12	13	14				
x‘00’	x‘01’	x‘02’	x‘03’	x‘04’	x‘05’	x‘06’	x‘07’
21	22	23	24	25	26	27	28
x‘08’	x‘09’	x‘0A’	x‘0B’	x‘0C’	x‘0D’	x‘0E’	x‘0F’
31	32	33	34	'A'	'B'	'C'	'D'
x‘10’	x‘11’	x‘12’	x‘13’	x‘14’	x‘15’	x‘16’	x‘17’
'E'	'F'	'G'		51	52		
x‘18’	x‘19’	x‘1A’	x‘1B’	x‘1C’	x‘1D’	x‘1E’	x‘1F’
61	62	63	64				
x‘20’	x‘21’	x‘22’	x‘23’	x‘24’	x‘25’	x‘26’	x‘27’

Распределение с прямым порядком байтов

В таблице Б.7 представлена структура s, распределенная в формате с прямым порядком байтов. Затененные ячейки соответствуют заполненным байтам.

Таблица Б.7 - Распределение структуры S с прямым порядком байтов

14	13	12	11				
x‘00’	x‘01’	x‘02’	x‘03’	x‘04’	x‘05’	x‘06’	x‘07’
28	27	26	25	24	23	22	21
x‘08’	x‘09’	x‘0A’	x‘0B’	x‘0C’	x‘0D’	x‘0E’	x‘0F’
34	33	32	31	'A'	'B'	'C'	'D'
x‘10’	x‘11’	x‘12’	x‘13’	x‘14’	x‘15’	x‘16’	x‘17’
'E'	'F'	'G'		52	51		
x‘18’	x‘19’	x‘1A’	x‘1B’	x‘1C’	x‘1D’	x‘1E’	x‘1F’
64	63	62	61				
x‘20’	x‘21’	x‘22’	x‘23’	x‘24’	x‘25’	x‘26’	x‘27’

Порядок байтов в инструкциях

Power ISA определяет инструкции как выровненные слова (4 байта) в памяти. В этом качестве инструкции в программном отображении обратного порядка организованы таким образом, что наиболее значимый байт (MSB) слова инструкции располагается по адресу с меньшим номером.

Рассмотрим распределение с обратным порядком байтов инструкции p по адресу x‘00’, где, например, p = add r7, r7, r4:

MSB			LSB
x‘00’	x‘01’	x‘02’	x‘03’

В распределении с прямым порядком та же инструкция организована таким образом, что наименее значимый байт (LSB) слова инструкции располагается по адресу с меньшим номером.

LSB			MSB
x‘00’	x‘01’	x‘02’	x‘03’

По определению нумерации бит Power ISA, наиболее значимый байт инструкции - это байт, содержащий биты 0-7 инструкции. Наиболее значимый байт - это байт, содержащий первичное поле кода операции (биты 0:5). Из-за этой разницы в порядке расположения байтов процессор должен при необходимости менять порядок байтов (в зависимости от того, какой порядок используется), чтобы корректно доставлять поле кода операции к декодеру инструкции. В ядре PowerPC 476FP такая смена порядка происходит между интерфейсом памяти и кэшем инструкций в соответствии со значением атрибута порядка следования для каждой страницы памяти, так чтобы байты в кэше инструкций всегда были правильно организованы для доставки непосредственно в декодер инструкций.

Если атрибут порядка следования для страницы памяти перепрограммируется с одного порядка следования на другой, то содержимое страницы памяти должно быть перезагружено структурами данных и программы, с нужным порядком следования байтов. Более того, при смене содержимого страницы памяти инструкций кэш инструкций должен быть приведен в соответствие с изменениями при помощи инвалидации кэша инструкций и повторной выборки содержимого памяти с новым порядком следования байтов.

Порядок байтов в данных

В отличие от выборки инструкций, при обращении к данным байты не могут менять порядок следования между памятью и кэшем данных. Порядок байтов данных в памяти зависит от типа данных (байт, полуслово, слово и так далее) конкретного элемента данных. Только при перемещении элемента данных определенного типа из предусмотренного архитектурой регистра или в этот регистр, как указано инструкцией обращения к памяти, становится известно, какой тип смены порядка байтов может потребоваться вследствие определенного порядка данных в странице памяти, содержащей этот элемент данных. Поэтому изменение порядка байтов во время обращения к памяти при загрузке и сохранении выполняется между кэшем данных (или, в случае промаха кэша, между памятью) и регистром загрузки приемника или регистром памяти источника, в зависимости от конкретного типа инструкции загрузки или сохранения (то есть, байт, полуслово, слово и так далее).

Сравнивая распределение структуры s с обратным и прямым порядком байтов, как показано в Разделе 2.1.3.1 Примеры распределения структуры, на стр. 37, разница в расположении байтов любого элемента данных в структуре зависит от размера конкретного элемента данных. Например, снова обращаясь к распределениям с обратным и прямым порядком структуры s:

- Порядок 4 байтов слова a изменен внутри слова, занимающего адреса от x‘00’ до x‘03’.
- Порядок 2 байтов слова e изменен внутри слова, занимающего адреса от x‘1C’ до x‘1D’.

Массив байтов d, где каждый элемент данных является байтом, выглядит одинаково при распределении с прямым и обратным порядком. Например, символ 'A' расположен по адресу x‘14’ в обоих распределениях, как с прямым, так и с обратным порядком следования байтов.

Размер элемента данных, который загружается или хранится, должен быть известен до того, как процессор сможет определить, нужно ли изменить порядок следования байтов, и если нужно, то как, во время перемещения их между регистром и кэшем данных (или памятью):

- Для загрузки и сохранения байтов, включая строки, порядок следования байтов не изменяется.
- Для загрузки и сохранения полуслов порядок следования байтов изменяется внутри полуслова, для одного порядка следования по отношению к другому.
- Для загрузки и сохранения слов (в том числе при многоадресной загрузке/сохранении) порядок следования байтов изменяется внутри слова, для одного порядка следования по отношению к другому.

Этот механизм применяется независимо от выравнивания данных. То есть, при загрузке операнда многобайтовых данных со скалярной инструкцией загрузки к байтам в кэше данных (или памяти) обращение выполняется, начиная с байта с вычисленным исполнительным адресом, а затем к байтам с большими номерами, пока не будет извлечено требуемое число байтов. Затем байты организуются таким образом, что байт либо из адреса с наибольшим номером (для регионов с обратным порядком следования байтов), либо из адреса с наименьшим номером (для регионов с прямым следованием байтов) помещается в наименее значимый байт регистра. Остальная часть регистра заполняется в соответствии с порядком остальных полученных байтов. Аналогичная процедура применяется для скалярных инструкций сохранения.

Для многоадресных инструкций загрузки/сохранения каждая группа из 4 байтов перемещается между памятью и регистром в соответствии с процедурой для скалярной инструкции загрузки слова.

Для строковых инструкций загрузки/сохранения наиболее значимый байт первого регистра перемещается в память или из памяти по начальному (с наименьшим номером) исполнительному адресу, независимо от порядка следования байтов. Следующие байты регистра (от наиболее значимого к наименее значимому, после чего выполняется переход к следующему регистру, начиная с наиболее значимого байта, и так далее) перемещаются в память или из памяти последовательно по адресам с большими номерами. Такое поведение для байтов строк дает гарантию, что если в регистры загружаются две строки, а затем сравниваются, то при сравнении первые байты строк считаются наиболее значимыми.

Инструкции с реверсом порядка следования байтов

Power ISA определяет инструкции загрузки/сохранения с реверсом порядка следования байтов, которые могут обращаться к памяти с тем или иным порядком следования байтов точно так же, как обычные (без реверса порядка следования байтов) инструкции загрузки/сохранения обращаются к памяти с противоположным порядком следования байтов. То есть, инструкция загрузки/сохранения с реверсом порядка следования байтов, обращающаяся к странице памяти с обратным порядком следования байтов, пересылает данные между кэшем данных (или памятью) и регистром точно так же как обычная инструкция загрузки/сохранения пересылает данные к странице памяти с прямым порядком следования, или из нее. Аналогичным образом, инструкция загрузки/сохранения с реверсом порядка следования байтов, обращающаяся к странице памяти с прямым порядком следования байтов, пересылает данные между кэшем данных (или памятью) и регистром точно так же как обычная инструкция загрузки/сохранения пересылает данные к странице памяти с обратным порядком следования, или из нее.

Инструкции загрузки/сохранения с реверсом порядка следования байтов полезны в тех случаях, когда определенная страница памяти содержит данные как с обратным, так и с прямым порядком следования байтов. В такой среде атрибут порядка следования байтов для страницы памяти устанавливается согласно преобладающему порядку следования байтов на странице, и для обращения к операндам, хранимых в памяти, которые используют этот преобладающий порядок, используются обычные инструкции загрузки/сохранения. Инструкции с реверсом

порядка следования байтов используются для обращения к хранимым в памяти операндам с другим порядком следования байтов (меньшинству).

Обычно программные компиляторы не способны использовать инструкции загрузки/сохранения с реверсом порядка следования байтов. Такие инструкции, как правило, используются только в специальных драйверах устройств, программируемых вручную.

Регистры

В данном разделе содержится обзор категорий и типов регистров, входящих в состав ядра PowerPC 476FP. Подробное описание каждого регистра приводится в разделах, посвященных функциям, с которыми ассоциированы эти регистры.

Все регистры ядра PowerPC 476FP 32-разрядные (биты 32:63 и биты высшего порядка 0:31 игнорируются, если не указано обратное), но определенные биты в некоторых регистрах зарезервированы и поэтому не обязательно реализуются. Для всех регистров с полями, отмеченными как зарезервированные, эти зарезервированные поля должны записываться как 0, а считываться как неопределенные. Рекомендованная практика программирования - в регистр с зарезервированными полями изначально записывать 0, а в последующем выполнять запись в регистр при помощи приема "чтение-модификация-запись": считать содержимое регистра, при помощи логических команд изменить требуемые поля, оставив зарезервированные поля неизменными, а затем выполнить запись в регистр.

Все регистры для операций с числами с плавающей запятой (FPR) 64-разрядные и указываются как биты 0:63. Все регистры сгруппированы по категориям в соответствии с функциями процессора, с которыми они ассоциированы. Кроме того, каждый регистр относится к определенному типу, в зависимости от инструкций, которые используются для чтения и записи регистров этого типа. И наконец, большинство регистров ядра PowerPC 476FP определено Power ISA, хотя некоторые регистры зависят от конкретной реализации и уникальны для ядра PowerPC 476FP.

На рисунке Б.2 представлены регистры ядра PowerPC 476FP, входящие в модель программирования пользователя, то есть, те регистры, доступ к которым является непривилегированным, и которые доступны как для программ пользователя, так и для программ супервизора.

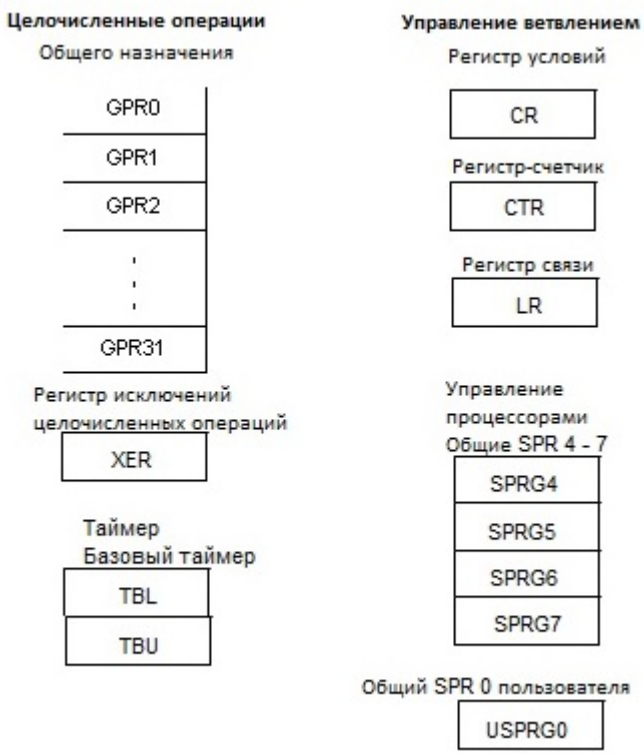


Рисунок Б.2 - Регистры модели программирования пользователя

На рисунке Б.3 представлены регистры ядра PowerPC 476FP, входящие в модель программирования супервизора, то есть, те регистры, доступ к которым является привилегированным, и которые доступны только для программ супервизора.

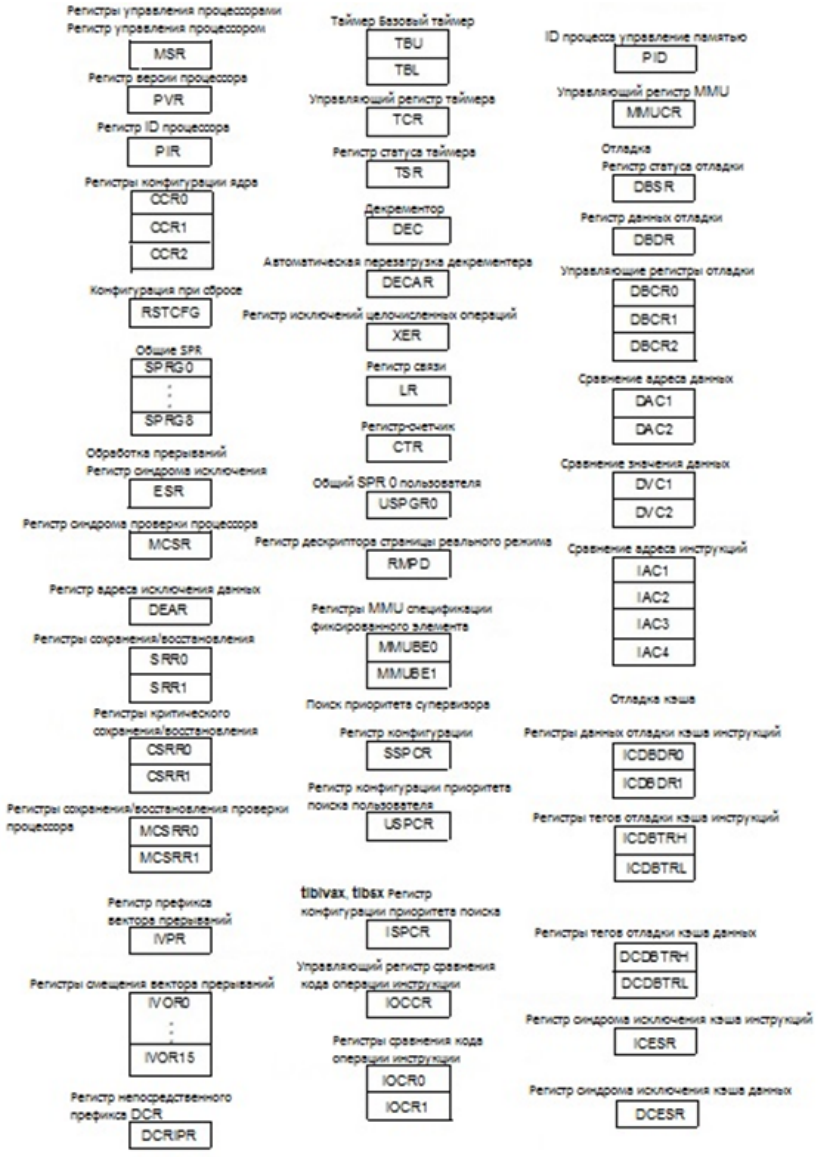


Рисунок Б.3 - Регистры модели программирования супервизора

В таблице Б.8 приведены регистры специального назначения (SPRS) PowerPC 476FP, их десятичные и двоичные номера (SPRNs), а также ссылки на разделы с их более подробным описанием. Регистры, не входящие в спецификацию Power ISA и являющиеся уникальными для ядра PowerPC 476FP, выделены курсивом. Если нет специальных указаний, то все регистры имеют доступ чтения/записи.

Таблица Б.8 - Регистры специального назначения PowerPC 476FP

SPR	Наименование SPR	Десятичный SPRN	Двоичный SPRN	Привилегированный	Доступ	Страница
CCR0	Регистр конфигурации ядра 0	947	‘11101 10011’	Да	R/W	69
CCR1	Регистр конфигурации ядра 1	888	‘11011 11000’	Да	R/W	70
CCR2	Регистр конфигурации ядра 2	889	‘11011 11001’	Да	R/W	73
CTR	Регистр-счетчик	9	‘00000 01001’	Нет	R/W	60
CSRR0	Регистр критического сохранения/восстановления 0	58	‘00001 11010’	Да	R/W	175
CSRR1	Регистр критического сохранения/восстановления 1	59	‘00001 11011’	Да	R/W	176
DAC1	Сравнение адреса данных 1	316	‘01001 11100’	Да	R/W	266
DAC2	Сравнение адреса данных 2	317	‘01001 11101’	Да	R/W	266

Продолжение таблицы Б.5

SPR	Наименование SPR	Десятичный SPRN	Двоичный SPRN	Привилегированный	Доступ	Страница
DCDBTRH	Регистр тегов отладки кэша данных, старшие разряды	925	‘11100 11101’	Да	Чтение	153
DCDBTRL	Регистр тегов отладки кэша данных, младшие разряды	924	‘11100 11100’	Да	Чтение	152
DEAR	Регистр адреса исключения данных	61	‘00001 11101’	Да	R/W	177
DVC1	Сравнение данных 1	318	‘01001 11110’	Да	R/W	266
DVC2	Сравнение данных 2	319	‘01001 11111’	Да	R/W	266
DCESR	Регистр синдрома исключения D-кэша	850	‘11010 10010’	Да	R/W	268
DCRIPR	Регистр непосредственного префикса DCR	891	‘11011 11011’	Да	R/W	74
DBCR0	Регистр управления отладкой 0	308	‘01001 10100’	Да	R/W	235
DBCR1	Регистр управления отладкой 1	309	‘01001 10101’	Да	R/W	236
DBCR2	Регистр управления отладкой 2	310	‘01001 10110’	Да	R/W	237
DBDR	Регистр данных отладки	1011	‘11111 10011’	Да	R/W	266
DBSR	Регистр статуса отладки	304	‘01001 10000’	Да	Чтение/очистка	239
		816	‘11001 10000’	Да	Запись	
DEC	Декрементор	22	‘00000 10110’	Да	R/W	159
DECAR	Автоматическая перезагрузка декрементора	54	‘00001 10110’	Да	R/W	159
ESR	Регистр синдрома исключения	62	‘00001 11110’	Да	R/W	179
ICESR	Регистр синдрома исключения I-кэша	851	‘11010 10011’	Да	R/W	140
IAC1	Сравнение адреса инструкций 1	312	‘01001 11000’	Да	R/W	240
IAC2	Сравнение адреса инструкций 2	313	‘01001 11001’	Да	R/W	240
IAC3	Сравнение адреса инструкций 3	314	‘01001 11010’	Да	R/W	240
IAC4	Сравнение адреса инструкций 4	315	‘01001 11011’	Да	R/W	240
ICDBDR0	Регистр данных отладки кэша инструкций 0	979	‘11110 10011’	Да	Чтение	137
ICDBDR1	Регистр данных отладки кэша инструкций 1	980	‘11110 10100’	Да	Чтение	137
ICDBTRH	Регистр тегов отладки кэша инструкций, старшие разряды	927	‘11100 11111’	Да	Чтение	138
ICDBTRL	Регистр тегов отладки кэша инструкций, младшие разряды	926	‘11100 11110’	Да	Чтение	137
IOCCR	Управляющий регистр сравнения кода операции инструкции	860	‘11010 11100’	Да	R/W	269
IOCR1	Регистр сравнения кода операции инструкции 1	861	‘11010 11101’	Да	R/W	269
IOCR2	Регистр сравнения кода операции инструкции 2	862	‘11010 11110’	Да	R/W	270
XER	Регистр исключений операций с фиксированной запятой	1	‘00000 00001’	Нет	R/W	64
IVOR[0-15]	Регистр смещения вектора прерываний	400 - 415	‘01100 1xxxx’	Да	R/W	178
IVPR	Регистр префикса вектора прерываний	63	‘00001 11111’	Да	R/W	179
LR	Регистр связи	8	‘00000 01000’	Нет	R/W	59
MCSRR0	Регистр сохранения/восстановления проверки процессора 0	570	‘10001 11010’	Да	R/W	176

Продолжение таблицы Б.8

SPR	Наименование SPR	Десятичный SPRN	Двоичный SPRN	Привилегированный	Доступ	Страница
MCSRR1	Регистр сохранения/восстановления проверки процессора 1	571	‘10001 11011’	Да	R/W	177
MCSR	Регистр синдрома проверки процессора	572	‘10001 11100’	Да	R/W	181
		828	‘11001 11100’	Да	Очистка	
MSR	Регистр управления процессором	-	-	Да	R/W	173
MMUBE0	Регистр MMU спецификации фиксированного элемента 0	820	‘11001 10100’	Да	R/W	121
MMUBE1	Регистр MMU спецификации фиксированного элемента 1	821	‘11001 10101’	Да	R/W	121
MMUCR	Управляющий регистр таймера	946	‘11101 10010’	Да	R/W	126
PMUCC0	Управляющий регистр ядра PMU	858	‘11010 11010’	Да	R/W	259
PMUCC0	Управляющий регистр ядра PMU, пользователя	842	‘11010 01010’	Нет	Чтение	259
PID	Регистр ID процесса	48	‘00001 10000’	Да	R/W	120
PIR	Регистр ID процессора	286	‘01000 11110’	Да	Чтение	68
PVR	Регистр версии процессора	287	‘01000 11111’	Да	Чтение	68
PWM	Регистр границ ширины импульса	886	‘11011 10110’	Да	R/W	
RMPD	Регистр дескриптора страницы реального режима	825	‘11001 11001’	Да	R/W	120
RSTCFG	Регистр конфигурации при сбросе	923	‘11100 11011’	Да	Чтение	126
SRR0	Регистр сохранения/восстановления 0	26	‘00000 11010’	Да	R/W	174
SRR1	Регистр сохранения/восстановления 1	27	‘00000 11011’	Да	R/W	175
SPRG[0-3]	Общий SPR 0 - 3	272 - 275	‘01000 100xx’	Да	R/W	67
SPRG3	Общий SPR 3	259	‘01000 00011’	Нет	Чтение	67
SPRG[4-7]	Общий SPR 4 -7	260 - 263	‘01000 001xx’	Нет	Чтение	67
SPRG[4-7]	Общий SPR 4 -7	276 - 279	‘01000 101xx’	Да	R/W	67
SPRG8	Общий SPR 8	604	‘10010 11100’	Да	R/W	67
SSPCR	Регистр конфигурации приоритета поиска супервизора	830	‘11001 11110’	Да	R/W	122
TBL	Регистр базового таймера	268	‘01000 01100’	Нет	Чтение	158
TBL	Регистр базового таймера, младшие разряды	284	‘01000 11100’	Да	Запись	158
TBU	Регистр базового таймера, старшие разряды	269	‘01000 01101’	Нет	Чтение	158
		285	‘01000 11101’	Да	Запись	
TCR	Управляющий регистр таймера	340	‘01010 10100’	Да	R/W	163
TSR	Регистр статуса таймера	336	‘01010 10000’	Да	Чтение/ Очистка	164
		848	‘11010 10000’	Да	Запись	
ISPCR	tlbivax, tlbsx Регистр конфигурации приоритета поиска	829	‘11001 11101’	Да	R/W	123
USPCR	Регистр конфигурации приоритета поиска пользователя.	831	‘11001 11111’	Да	R/W	125
USPGR0	Общий SPR 0 пользователя	256	‘01000 00000’	Нет	R/W	67

Примечание - R = чтение; W = запись.

Типы регистров

Различают пять типов регистров, содержащихся в ядре или поддерживаемых ядром PowerPC 476FP. Каждый тип регистров определяется инструкциями, которые используются для чтения и записи регистров этого типа. В следующих подразделах приводятся краткие описания каждого типа регистра, а также ассоциированные с ним функции.

Регистры общего назначения

Ядро PowerPC 476FP содержит 32 регистра общего назначения (GPR); каждый из них предназначен для хранения 32-битного целочисленного значения. Данные из кэша памяти могут быть загружены в GPR при помощи инструкций загрузки целочисленных значений; содержание GPR может быть сохранено в кэше данных или в памяти при помощи инструкций сохранения целочисленных значений. Большинство инструкций для операций с целочисленными значениями обращаются к GPR. GPR также используются как приемники и источники для большинства инструкций, которые выполняют чтение и запись регистров других типов.

Регистры специального назначения

Непосредственный доступ к регистрам специального назначения (SPR) выполняется при помощи инструкций mtspr и mfspr. Кроме того, содержимое некоторых SPR может изменяться в качестве побочного эффекта выполнения различных инструкций. Например, регистр исключений целочисленных операций (XER) представляет собой SPR, который изменяется арифметическим состоянием (таким, как перенос и переполнение) при выполнении определенных видов инструкций целочисленной арифметики.

Регистры SPR управляют использованием средств отладки, таймеров, прерываний, управления памятью, кэша и других ресурсов архитектуры процессора.

Регистр условий

Регистр условий (CR) - это уникальный тип 32-битового регистра, разделенного на восемь независимых 4-битовых полей (CR0 - CR7). CR может использоваться для записи определенных условных результатов разных арифметических и логических операций. Поэтому условные инструкции ветвления могут назначать бит CR одним из условий ветвления. Инструкции также могут выполнять логические операции с битами и изменение полей внутри CR.

Регистр управления процессором (MSR)

Регистр управления процессором (MSR) - это уникальный тип регистра, который управляет основными функциями кристалла, такими как разрешение или запрет различных типов прерываний.

Запись в MSR может производиться из GPR при помощи инструкции mtmsr. Содержимое MSR может быть считано в GPR при помощи инструкции mfmsr. Бит MSR[EE] может быть установлен или сброшен отдельно при помощи инструкций wrtee или wrteei. Содержимое MSR также автоматически сохраняется, изменяется и восстанавливается механизмом обработки прерываний.

Регистры управления устройством

Регистры управления устройством (DCR) - это регистры на кристалле, которые архитектурно и физически не принадлежат ядру PowerPC 476FP и поэтому не определены ни в спецификации Power ISA, ни в данном руководстве для ядра PowerPC 476FP. Power ISA определяет существование адресного пространства DCR и инструкций обращения к DCR, но не конкретные DCR. Инструкции обращения к DCR представляют собой пересылку в регистр управления устройством (mtdcr) и из регистра управления устройства (mfdcr), которые осуществляют обмен данными между GPR и DCR.

DCR можно использовать для управления различными встроенными функциями, такими как работа внутренних шин, периферии, а также управления поведением процессора.

Для встраивания дополнительных DCR и SPR в систему в ядре PowerPC 476FP имеются дополнительные инструкции, такие как mtdcrx (индексированная пересылка в регистр управления устройством), mtdcrux (индексированная пересылка в регистр управления устройством в режиме пользователя), mfdcrx (индексированная пересылка из регистра управления устройством), и mfdcrux (индексированная пересылка из регистра управления устройством в режиме пользователя).

Классы инструкций

Спецификация Power ISA делит все инструкции на четыре класса, определяемые первичным кодом операции (и расширенным кодом операции, при наличии):

- Определенные
- Сохраненные
- Зарезервированные (-недопустимые или фиктивные)

Класс определенных инструкций

Этот класс инструкций включает все инструкции, определенные в Power ISA Version 2.05, Book-I, Book-II и Book-III E. В общем случае определенные инструкции гарантированно поддерживаются внутри системы Power ISA, как определено архитектурой - либо внутри самой реализации процессора, либо внутри программы эмуляции, поддерживаемой операционной системой.

Единственное исключение составляют реализации (такие, как ядро PowerPC 476FP), которые обеспечивают только 32-битное подмножество Power ISA, и от системы на требуется эмуляция (и она, скорее всего не может ее реализовать) 64-битного поведения определенных инструкций.

Согласно Power ISA любая попытка выполнить определенную инструкцию приводит к одному из следующих результатов:

Программное прерывание типа исключения недопустимой инструкции, если инструкция не распознается, или

Программное прерывание типа исключения нереализованной инструкции, если инструкция распознана и не относится к операциям с плавающей запятой, но не поддерживается данной реализацией, или

Прерывание недоступности операции с плавающей запятой, если инструкция распознана как операция с плавающей запятой, но операции с плавающей запятой запрещены; или

Выполнение действий, описанных в этом документе если инструкция распознана и поддерживается данной реализацией. Поведение, определяемое архитектурой, может вызывать другие исключения.

Ядро PowerPC 476FP распознает и полностью поддерживает все инструкции класса определенных, за несколькими исключениями. Во-первых, поскольку ядро PowerPC 476FP является 32-битной реализацией, те операции, которые определены только как 64-битные, не поддерживаются совсем и всегда вызывают программное прерывание типа исключения недопустимой инструкции.

Кроме того, в ядре PowerPC 476FP не поддерживается одна определенная инструкция. Инструкция mfapidi (косвенная пересылка из ID сопроцессора) представляет собой специальную инструкцию, предназначенную для помощи в идентификации сопроцессора, который присоединен к реализации процессора. Поскольку ядро PowerPC 476FP не имеет сопроцессора, то инструкция mfapidi не поддерживается. Выполнение mfapidi вызывает прерывание типа исключения недопустимой инструкции.

Класс сохраненных инструкций

Класс сохраненных инструкций предназначен для поддержки совместимости с более ранними версиями архитектуры PowerPC или Power ISA. Этот класс инструкций содержит коды операции, определенные для этих предыдущих архитектур, но уже не включенные в Power ISA.

Любая попытка выполнить сохраненную инструкцию приводит к одному из следующих результатов:

Выполнение действий, описанных в предыдущей версии архитектуры, если инструкция распознана.

Программное прерывание типа исключения недопустимой инструкции, если инструкция не распознана.

Ядро PowerPC 476FP распознает и поддерживает одну сохраненную инструкцию - код операции mftb (пересылка из базового таймера). Эта инструкция использовалась в архитектуре PowerPC для считывания старшего (TBU) и младшего (TBL) регистров базового таймера. В Power ISA регистры TBU и TBL определены как регистры SPR, и поэтому для их чтения используется инструкция mfspr (пересылка из регистра специального назначения). Для того, чтобы на ядре PowerPC 476FP могли выполняться предшествующие программы управления базовым таймером, ядро также поддерживает сохраненный код операции инструкции mftb. Тем не менее, инструкция mftb не включена в различные разделы данного документа, описывающие реализованные инструкции, и программа должна использовать новейший механизм mfspr для чтения регистров базового таймера, чтобы гарантировать переносимость между ядром PowerPC 476FP и будущими реализациями Power ISA.

Класс зарезервированных инструкций

Этот класс инструкций включает все инструкции с первичными кодами операций (и соответствующими дополнительными кодами операций, если таковые имеются), которые не принадлежат ни к классу определенных, ни к классу сохраненных.

Зарезервированные инструкции доступны для будущих версий Power ISA. То есть, будущие версии Power ISA. могут определять любую из этих инструкций, чтобы выполнять новые функции или сделать их доступными для использования конкретными реализациями в качестве выделенных инструкций. Различают два типа зарезервированных инструкций: недопустимые и фиктивные.

Любая попытка выполнения зарезервированной недопустимой инструкции на ядре PowerPC 476FP приведет к программному прерыванию типа исключения недопустимой инструкции. Таким образом, зарезервированные недопустимые инструкции доступны для будущих расширений Power ISA, которые могут изменить архитектуру. К таким расширениям могут относиться новые формы инструкция для целочисленных операций или операций с плавающей запятой, новые формы инструкций загрузки и сохранения, которые влияют на встроенные регистры или содержание памяти.

Однако любая попытка выполнения зарезервированной фиктивной инструкции либо не окажет эффекта (то есть, будет воспринята как неисполняемая инструкция), либо приведет на ядре PowerPC 476FP к программному прерыванию типа исключения недопустимой инструкции. Поскольку реализации обычно предполагают интерпретацию зарезервированных фиктивных инструкций как обычных фиктивных инструкций, коды операции этих инструкций доступны для будущих расширений Power ISA, которые не повлияют на архитектуру. К таким расширениям могут относиться повышающие производительность подсказки, например, новые формы инструкций обращения к кэшу. Программа может использовать преимущества функций, предлагаемых новыми инструкциями, и одновременно оставаться полностью совместимой с реализациями предыдущих версий Power ISA.

В ядре PowerPC 476FP все коды операции зарезервированных фиктивных инструкций реализованы как действующие фиктивные инструкции.

Краткое описание набора реализованных инструкций

В этом разделе представлен обзор разных типов и категорий инструкций, реализованных в ядре PowerPC 476FP. В таблице Б.9 представлены инструкции ядра PowerPC 476FP, распределенные по категориям. Инструкции каждой категории описаны в следующих разделах.

Таблица Б.9 - Категории инструкций

Категория	Подкатегория	Типы инструкций
Целочисленные	Целочисленные, обращение к памяти	загрузка/сохранение
	Целочисленная арифметика	сложение, вычитание, умножение, деление, инвертирование
	Целочисленная логика	and, andc, or, orc, xor, nand, nor, xnor, расширение знака, счет нулевых старших разрядов
	Целочисленное сравнение	сравнение, логическое сравнение
	Целочисленный выбор	выбор операнда
	Целочисленная ловушка	ловушка
	Целочисленный циклический сдвиг	циклический сдвиг и вставка, циклический сдвиг и маскирование
	Целочисленный сдвиг	сдвиг влево, сдвиг вправо, алгебраический сдвиг вправо
Ветвление		ветвление, условное ветвление, ветвление по связи, ветвление по счетчику
Управление процессором	Логические регистра условий	crand, crandc, cror, crorc, crnand, crnor, crxor, crxnor
	Управление регистрами	пересылка в/из SPR, пересылка в/из DCR, пересылка в/из MSR, запись в бит разрешения внешнего прерывания, пересылка в/из CR
	Связь с системой	системный вызов, возврат из прерывания, возврат из критического прерывания, возврат из прерывания проверки процессора
	Синхронизация процессора	инструкция синхронизации
Управление памятью	Управление кэшем	распределение данных, инвалидация данных, обращение к данным, обнуление данных, очистка данных, сохранение данных, инавалидация инструкций, предвыборка инструкций
	Управление TLB	чтение, запись, поиск, синхронизация
	Синхронизация памяти	синхронизация памяти, барьер памяти
Выделенные	Выделенные арифметические	умножение с накоплением, умножение с накоплением отрицательных чисел, умножение полуслова
	Выделенные логические	обнаружение самого левого нулевого байта
	Выделенные управления кэшем	Инвалидация класса конгруэнтных данных, инвалидация класса конгруэнтных инструкций
	Выделенные отладка кэша	чтение данных, чтение инструкций

Инструкции целочисленных операций

Инструкции целочисленных операций пересылают данные между памятью и регистрами GPR, а также выполняют разнообразные операции в GPR. Эта категория инструкций, в свою очередь, разделена на восемь подкатегорий, которые описаны в следующих разделах.

Целочисленные инструкции обращения к памяти

Целочисленные инструкции обращения к памяти выполняют операции загрузки/сохранения данных между памятью и GPR. Эти инструкции оперируют байтами, полусловами и словами. Целочисленные инструкции обращения к памяти также поддерживают загрузку/сохранение нескольких регистров, строки символов и данных с реверсированным порядком байтов, а также загрузку данных с расширением знака.

В таблице Б.10 представлены целочисленные инструкции обращения к памяти ядра PowerPC 476FP. В таблице синтаксис [u] указывает, что инструкция имеет как модифицированную форму (в которой регистр адресации RA дополнен вычисленным адресом), так и немодернизированную форму. Аналогичным образом, синтаксис [x] указывает, что инструкция имеет как индексную форму (в которой адрес образуется прибавлением содержимого регистров общего назначения RA и RB), так и форму база+смещение (в которой адрес формируется прибавлением 16-битного непосредственного значения со знаком (определенного как часть инструкции) к содержимому регистра общего назначения RA).

Таблица Б.10 - Целочисленные инструкции обращение к памяти

Загрузка				Сохранение			
Байт	Полуслово	Слово	Множество/ Строка	Байт	Полуслово	Слово	Множество/ Строка
lbz[u][x]	lha[u][x] lhbrx lhz[u][x]	lwarx lwbrx lwz[u][x]	lmw lswi lswx	stb[u][x]	sth[u][x] sthbrx	stw[u][x] stwbrx stwcx.	stmw stswi stswx

Целочисленные арифметические инструкции

Арифметические операции выполняются над целочисленными или порядковыми операндами, хранящимися в регистрах. Инструкции, которые выполняют операции над двумя операндами, определены в 3-операндном формате; операция выполняется над операндами, которые хранятся в двух регистрах. Результат помещается в третий регистр. Инструкции, которые выполняют операции над одним операндом, определены в 2-операндном формате; операция выполняется над операндом, который хранится в одном регистре, а результат помещается во второй регистр. Несколько инструкций имеют непосредственный формат, в котором один из операндов источника является полем инструкции.

Большинство целочисленных арифметических инструкций имеют версии, которые в результате могут выполнения могут изменять CR[CR0] или XER[SO, OV]. Некоторые целочисленные арифметические операции также по умолчанию изменяют XER[CA].

В таблице Б.11 приведены целочисленные арифметические инструкции ядра PowerPC 476FP. В таблице синтаксис [o] указывает, что инструкция имеет как форму, которая изменяет поля XER[SO ,OV], так и форму, не изменяющую эту поля. Аналогичным образом, синтаксис [.] указывает, что инструкция имеет как форму с записью (которая изменяет CR[CR0]), так и форму без записи.

Таблица Б.11 - Целочисленные арифметические инструкции

Сложение	Вычитание	Умножение	Деление	Инвертирование
add[o][.] addc[o][.] adde[o][.] addi addic[.] addis addme[o][.] addze[o][.]	subf[o][.] subfc[o][.] subfe[o][.] subfic subfme[o][.] subfze[o][.]	mulhw[.] mulhwu[.] mulli mullw[o][.]	divw[o][.] divwu[o][.]	neg[o][.]

Целочисленные логические инструкции

В таблице Б.12 представлены целочисленные логические инструкции ядра PowerPC 476FP.

Таблица Б.12 - Целочисленные логические инструкции

И	И с дополнением	НЕ-И	ИЛИ	ИЛИ с дополнением	НЕ-ИЛИ	Исключающее ИЛИ	Эквивалентность	Расширенный знак	Подсчет нулевых старших разрядов
and[.] andi. andis.	andc[.]	nand[.]	or[.] ori oris	orc[.]	nor[.]	xor[.] xori xoris	eqv[.]	extsb[.] extsh[.]	cntlzw[.]

Целочисленные инструкции сравнения

Эти инструкции выполняют арифметическое или логическое сравнение двух операндов и помещают в CR результат сравнения.

В таблице Б.13 приведены целочисленные инструкции сравнения ядра PowerPC 476FP.

Таблица Б.13 - Целочисленные инструкции сравнения

Арифметические	Логические
cmp cmpi	cmpl cmpli

Инструкции целочисленных ловушек

В таблице Б.14 приведены инструкции целочисленных ловушек ядра PowerPC 476FP.

Таблица Б.14 - Инструкции целочисленных ловушек

Ловушка
tw twi

Целочисленные инструкции циклического сдвига

Эти инструкции выполняют циклический сдвиг операндов, хранящихся в регистрах GPR. Инструкции циклического сдвига также маскируют сдвигаемые операнды.

В таблице Б.15 приведены целочисленные инструкции циклического сдвига ядра PowerPC 476FP.

Таблица Б.15 - Целочисленные инструкции циклического сдвига

Циклический сдвиг и вставка	Циклический сдвиг и маскирование
rlwimi[.]	rlwnm[.] rlwinm[.]

Целочисленные инструкции сдвига

В таблице Б.16 приведены целочисленные инструкции сдвига ядра PowerPC 476FP. Следует учесть, что алгебраические инструкции сдвига вправо по умолчанию изменяют поле XER[CA].

Таблица Б.16 - Целочисленные инструкции сдвига

Сдвиг влево	Сдвиг вправо	Алгебраический сдвиг
slw[.]	srw[.]	sraw[.] srawi[.]

Целочисленная инструкция выбора

В таблице Б.17 приведена целочисленная инструкция сдвига ядра PowerPC 476FP. Операнд RA равен 0, если поле RA инструкции равно 0; в противном случае - это содержание GPR[RA].

Таблица Б.17 - Целочисленная инструкция выбора

Целочисленный выбор
isel

Инструкции ветвления

Эти инструкции определяют безусловный или условный переход по адресу. Инструкции условного перехода могут проверять коды условий, установленные в CR предыдущей инструкцией, и выполнять соответствующий переход. Инструкции условного перехода также могут декрементировать и проверять CTR в процессе определения перехода, и сохранять адрес возврата в регистре связи (LR). Целевой адрес ветвления может представлять собой смещение относительно адреса текущей инструкции или содержаться в LR или CTR.

В таблице Б.18 приведены инструкции ветвления ядра PowerPC 476FP. В таблице Б.18 синтаксис [l] указывает, что инструкция имеет как форму с изменением связи (помещает в LR адрес инструкции после ветвления), так и форму без изменения связи. Аналогичным образом, синтаксис [a] указывает, что инструкция имеет как форму с абсолютной адресацией (когда целевой адрес формируется непосредственно с использованием непосредственного поля, составляющего часть инструкции) и с относительной адресацией (когда целевой адрес формируется прибавлением указанного непосредственного поля к адресу инструкции ветвления).

Таблица Б.18 - Инструкции ветвления

Ветвление
b[I][a]
bc[I][a]
bcctr[I]
bclr[I]

Инструкции управления процессором

Инструкции управления процессором манипулируют системными регистрами, осуществляют связь системного ПО и синхронизируют операции процессора. В следующих разделах описаны инструкции этих трех подкатегорий инструкций управления процессором.

Логические инструкции регистра условий

Эти инструкции выполняют логические операции над указанной парой бит в CR, помещая результат в другой указанный бит. Преимущество этих инструкций состоит в том, что они могут логически объединять результаты нескольких операций сравнения, не затрачивая дополнительного времени на переходы между ними. Производительность программы может значительно повыситься, если при определении перехода проверяются сразу несколько условий.

В таблице Б.19 приведены логические инструкции регистра условий ядра PowerPC 476FP.

Таблица Б.19 - Логические инструкции регистра условий

Логические, регистра условий	
crand	crnor
crandc	cror
creqv	crorc
crnand	crxor

Инструкции управления регистрами

Эти инструкции осуществляют пересылку данных между регистрами GPR и управляющими регистрами ядра PowerPC 476FP.

В таблице Б.20 приведены инструкции управления регистрами ядра PowerPC 476FP.

Таблица Б.20 - Инструкции управления регистрами

CR	DCR	MSR	SPR
mcrf	mfdcr	mfmsr	mfspr
mcrxr	mfdcrux	mtmsr	mtspr
mfcr	mfdcrx	wrtee	
mtcrf	mtdcr	wrteei	
	mtdcrux		
	mtdcrx		

Инструкции связи с системой

Эти инструкции запускают программы уровня системного супервизора для системных функций и возврата из прерываний.

В таблице Б.21 приведены инструкции связи с системой ядра PowerPC 476FP.

Таблица Б.21 - Инструкции связи с системой

Связь с системой
rfi
rfci
rfmci
sc

Инструкция синхронизации процессора

Инструкция синхронизации процессора isync заставляет процессор завершить выполнение всех инструкций, предшествовавших isync, прежде чем разрешить любые контекстные изменения в результате инструкций, следующих за isync . Кроме того, все инструкции, следующие за isync, выполняются в контексте, установленном завершением всех инструкций, предшествовавших isync.

В таблице Б.22 приведена инструкция синхронизации процессора ядра PowerPC 476FP.

Таблица Б.22 - Инструкция синхронизации процессора

Синхронизация процесса
isync

Инструкции управления памятью

Эти инструкции управляют кэшем инструкций, кэшем данных и TLB ядра PowerPC 476FP. Инструкции также предоставляют доступ к синхронизации и упорядочиванию памяти. В следующих разделах описаны инструкции этих трех подкатегорий инструкций управления памятью.

Инструкции управления кэшем

Эти инструкции управляют операциями с кэшем данных и кэшем инструкций. Инструкции обеспечивают заполнение, очистку, инвалидацию или обнуление блоков кэша данных; каждый блок представляет собой 32-байтную строку кэша. Инструкции также обеспечивают заполнение или инвалидацию блоков кэша инструкций.

В таблице Б.23 приведены инструкции управления кэшем ядра PowerPC 476FP.

Таблица Б.23 - Инструкции управления кэшем

Кэш данных	Кэш инструкций
dcba (no-op) dcbf dcbi dcbst dcbt dcbtst dcbz dcbtls dcbtstls dcblc	icbi icbt icbtls icblc

Инструкции управления TLB

Инструкции управления TLB считывают и записывают элементы массива TLB и выполняют в массиве TLB поиск элемента, который транслируется в конкретный виртуальный адрес.

В таблице Б.24 представлены инструкции управления TLB ядра PowerPC 476FP.

Таблица Б.24 - Инструкции управления TLB

Управление TLB
tlbre tlbsx[.]

Инструкции синхронизации памяти

Инструкции синхронизации памяти позволяют программе установить порядок доступа к памяти инструкций загрузки и сохранения, которые по умолчанию слабо упорядочены процессором. Слабая упорядоченность означает, что архитектура процессора позволяет выполнять загрузку и сохранение без связи с последовательностью в потоке инструкций, за несколькими исключениями. Однако при выполнении инструкции синхронизации памяти все обращения к памяти, предшествовавшие инструкции синхронизации, должны быть выполнены, прежде чем будут разрешены обращения к памяти, вызванные инструкциями, следующими за инструкцией синхронизации.

В таблице Б.25 приведены инструкции синхронизации памяти ядра PowerPC 476FP.

Таблица Б.25 - Инструкции синхронизации памяти

Синхронизация памяти
lwsync msync mbar

Предыдущие целочисленные инструкции умножения с накоплением

Предыдущие целочисленные инструкции умножения с накоплением реализованы в ядре PowerPC 476FP разделены на четыре подкатегории и приведены в таблице Б.26.

Таблица Б.26 - Предыдущие целочисленные инструкции умножения с накоплением

Арифметические		
Умножение с накоплением	Умножение с накоплением отрицательных чисел	Умножение полуслова
macchw[o][.] macchws[o][.] macchws[o][.] macchwu[o][.] machhw[o][.] machhws[o][.] machhwsu[o][.] machhwu[o][.] maclhw[o][.] maclhws[o][.] maclhwsu[o][.] maclhwu[o][.]	nmacchw[o][.] nmacchws[o][.] nmachhw[o][.] nmachhws[o][.] nmaclhw[o][.] nmaclhws[o][.]	mulchw[.] mulchwu[.] mulhhw[.] mulhhwu[.] mullhw[.] mullhwu[.]

Обработка ветвлений

В следующих разделах приводится дополнительная информация об адресации переходов, полях инструкций, предсказании и регистрах.

Адресация переходов

Инструкция ветвления (b[l][a]) указывает смещение целевого адреса перехода как 26-битное значение (24-битное поле LI, расширенное справа на ‘00’). Это смещение рассматривается как 26-битное число со знаком, покрывающее диапазон адресов □32 MB. Аналогичным образом, инструкция ветвления (bc[l][a]) указывает смещение как 16-битное значение (14-битное поле BD, расширенные справа на ‘00’). Это смещение покрывает диапазон адресов □32 KB.

Для относительной адресации ветвления и инструкций условного ветвления b[l] и bc[l], когда поле инструкции AA = ‘0’, целевой адрес представляет собой адрес самой инструкции ветвления (адрес текущей инструкции) плюс смещение со знаком. Такое вычисление адреса по определению зацикливается - от максимального исполнительного адреса x‘FFFF FFFF’) к x‘0000 0000’ и наоборот.

Для абсолютной адресации ветвления и инструкций условного ветвления (ba[l] и bca[l], когда поле инструкции AA = ‘1’), целевой адрес представляет собой смещение с расширенным знаком. Это означает, что при абсолютной адресации инструкции ветвления целевой адрес перехода может находиться в первых или в последних 32 MB адресного пространства. При абсолютной адресации инструкций условного перехода целевой адрес перехода может находиться в первых или в последних 32 KB адресного пространства.

Две другие инструкции ветвления, bclr (условный переход по LR) и bcctr (условный переход по CTR), не используют абсолютную и относительную адресацию. Вместо них они используют косвенную адресацию, при которой целевой адрес ветвления указывается косвенным образом как содержимое LR или CTR.

Поле BI инструкции ветвления

Инструкции условного ветвления опционально могут проверять один бит CR, что указывается значением 0 поля инструкции BO (опция перехода) (см. описание поля BO в Разделе 2.5.3). Значение поля инструкции BI (индекс перехода) указывает бит CR, который будет проверяться (0 - 31). Поле BI игнорируется, если BO[0] =‘1’. Инструкция ветвления (b[l][a]) по определению безусловная, и поэтому поле BI у нее отсутствует. Место этого поля является частью поля смещения LI.

Поле BO инструкции ветвления

Поле BO указывает условие, по которому выполняется условный переход, а необходимость декрементирования CTR при переходе. Инструкция ветвления (b[l][a]) по определению безусловная, и поэтому поле BO у нее отсутствует. Место этого поля является частью поля смещения LI.

Инструкции условного перехода опционально могут проверять один бит CR. Эта опция выбирается, когда BO[0] = ‘0’; если BO[0] = ‘1’, то CR не участвует в проверке условий перехода. Если выбрана опция перехода по CR, то условие удовлетворяется (переход может быть выполнен), когда бит CR, выбранный полем BI инструкции, совпадает с BO[1].

Инструкции условного перехода также могут опционально декрементировать CTR на единицу и проверять, не равно ли декрементироемое значение 0. Эта опция выбирается, когда BO[2] = ‘0’; если BO[2] = ‘1’, то CTR не декрементируется и не участвует в проверке условий перехода. Если выбрана опция декрементирования CTR, то BO[3] указывает условие, которое должно быть удовлетворено для разрешения перехода. Если BO[3] = ‘0’, то для выполнения перехода требуется CTR □□‘0’. Если BO[3] = ‘1’, то для выполнения перехода требуется CTR ‘0’.

В таблице Б.27 описано использование битов поля BO.

Таблица Б.27 - Описание поля BO

Бит BO	Описание
BO[0]	Управление проверкой CR 0 Проверить бит CR, указанный полем BI на значение, указанное в BO[1]. 1 Не проверять CR.
BO[1]	Тестовое значение для CR. 0 Если BO[0] = ‘0’, проверить на CR[BI] = ‘0’. 1 Если BO[0] = ‘0’, проверить на CR[BI] = ‘1’.
BO[2]	Декремент CTR и управление проверкой. 0 Декрементировать CTR на единицу и проверить, удовлетворяет ли декрементированное значение CTR условию, указанному в BO[3]. 1 Не декрементировать CTR; не проверять CTR.
BO[3]	Тестовое значение для CTR. 0 Если BO[2] = ‘0’, проверить декрементированный CTR на □ ‘0’. 1 Если BO[2] = ‘0’, проверить декрементированный CTR на ‘0’.
BO[4]	Реверсирование предсказание ветвления. 0 Применить стандартное предсказание ветвления. 1 Реверсировать стандартное предсказание ветвления.

В таблице Б.28 приведены некоторые значения поля BO и соответствующие им действия; z - это обязательное нулевое значение, а y - опция предсказания ветвления.

Таблица Б.28 - Примеры значений поля BO

Значение BO	Описание
0000y	Декрементировать CTR, затем выполнить переход, если декрементированный CTR □□‘0’ и CR[BI] = ‘0’.
0001y	Декрементировать CTR, затем выполнить переход, если декрементированный CTR ‘0’ и CR[BI] = ‘0’.
001zy	Перейти, если CR[BI] = ‘0’.
0100y	Декрементировать CTR, затем выполнить переход, если декрементированный CTR □□‘0’ и CR[BI] = ‘1’.
0101y	Декрементировать CTR, затем выполнить переход, если декрементированный CTR ‘0’ и CR[BI] = ‘1’.
011zy	Перейти, если CR[BI] = ‘1’.
1z00y	Декрементировать CTR, затем выполнить переход, если декрементированный CTR □□‘0’.
1z01y	Декрементировать CTR, затем выполнить переход, если декрементированный CTR ‘0’.
1z1zz	Всегда выполнять переход.

Биты “a” и “t” поля BO могут использоваться программой для подсказки о том, будет ли скорее всего, выполнен или не выполнен переход, следующим образом:

BO = 0z1at (биты “at” = ‘11’): Переход, скорее всего, будет выполнен.

BO = 0z1at (биты “at” = ‘10’): Переход, скорее всего, не будет выполнен.

Эта подсказка перехода разрешена, если установлен CCR2[SPC5C1]. В противном случае предсказание ветвления игнорируется.

Предсказание ветвления

Условные переходы могут выполняться или не выполняться; при выполнении перехода выборка инструкции перенаправляется на целевой адрес. Если переход не выполнен, производится выборка следующей последовательной инструкции. Ядро PowerPC 476FP пытается предсказать, будет ли выполнен переход, еще до того, как доступна вся необходимая информация о направлении ветвления. Это действие называется предсказанием ветвления. В этом случае ядро может выполнить опережающую выборку инструкций в соответствии с предсказанным переходом. В случае верного предсказания производительность повышается, поскольку целевая инструкция перехода доступна сразу же, и не нужно ждать, когда будут

проверены условия перехода. В случае неверного перехода результат опережающей выборки инструкции (которая была выбрана по адресу неверного перехода) должен аннулироваться, и новая инструкция выбирается по адресу верного перехода.

В ядре PowerPC 476FP объединен механизм статического предсказания, определенный в Power ISA, и механизм динамического предсказания ветвлений, чтобы максимально увеличить частоту верных предсказаний. Динамический механизм предсказания ветвлений представляет собой оптимизацию реализации и не является частью архитектуры; он невидим для модели программирования.

Механизм статического предсказания ветвлений позволяет программе указать предпочтительное предсказание ветвления посредством бит в коде инструкции. По умолчанию статическое предсказание ветвления для условных переходов имеет следующий вид:

Предсказать, что переход будет выполнен, если ((BO[0] □□BO[2]) □□s) = 1

где s это бит 16 инструкции (знаковый бит смещения для всех условных переходов [bc], ноль для всех переходов по регистру связи [bclr] и и условных переходов по регистру-счетчику [bcctr]). То есть, выполнение условного перехода предсказывается, если их смещение перехода отрицательное (то есть, переход осуществляется назад по отношению к адресу текущей инструкции). Стандартное предсказание для этого случая базируется на bc с относительной адресацией, которая часто используется в конце циклов для подсчета количества выполненных циклов. Поскольку при выполнении цикла переход выполняется каждый раз, за исключением последнего, то лучше предсказывать выполнение перехода. Целью перехода является начало цикла, и поэтому смещение перехода отрицательно, и s = ‘1’. Поскольку такая ситуация встречается чаще всего, то переход выполняется, если s = ‘1’.

В том случае, когда смещения положительны, s = ‘0’, и предсказывается, что переход не выполняется. Кроме того, если инструкция ветвления относится к любой формebclr или bcctr, за исключением безусловного перехода, s = ‘0’, и предсказывается, что переход не выполняется.

Для bc (bca и bcla) с абсолютной адресацией этот алгоритм предсказания имеет специфические последствия. Как описано в Разделе 2.5.1, Адресация ветвления на стр. 56, если s = ‘1’, то переход выполняется по старшим адресам. Если s = ‘0’, то цель перехода находится в младшей части адресного пространства. Но при абсолютной адресации нет причин по-разному обращаться со старшей и младшей частью адресного пространства. Тем не менее, для старших адресов стандартное предсказание применяется, а для младших адресов стандартное предсказание не применяется.

Еще один бит поля BO позволяет программе управлять предсказанием ветвлений. BO[4] - это бит реверса предсказания. Если BO[4] = ‘0’, то применяется предсказание по умолчанию. Если BO[4] = ‘1’, то применяется реверсированное предсказание по умолчанию. Для тех случаев в Табл. 2-25 Примеры значений поля BO на стр. 58, когда BO[4] = y, программа может реверсировать предсказание по умолчанию, установив y в ‘1’. Это следует делать только тогда, когда предсказание по умолчанию, скорее всего, окажется неверным. Следует отметить, что для ветвления, которое всегда условно, реверс предсказания по умолчанию не разрешен, поскольку в этом случае BO[4] обозначен как z, что означает, что бит должен быть установлен в 0; в противном случае инструкция не действительна.

Регистры управления ветвлением

В ядре PowerPC 476FP предусмотрены три регистра, предназначенные для обработки ветвлений, и они описаны в следующих разделах.

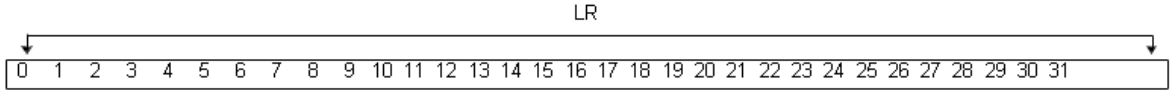
Регистр связи (LR)

Запись в LR выполняется из GPR с помощью инструкции mtspr, а его содержимое может считываться в GPR с помощью инструкции mfspr. Содержимое LR также может изменяться инструкцией ветвления с изменением связи (поле инструкции LK = ‘1’). Такая инструкция ветвления загружает в LR адрес инструкции, которая следует за инструкцией ветвления (4 + адрес инструкции ветвления). Таким образом, содержимое LR может использоваться как адрес возврата для подпрограммы, к которой был выполнен переход при помощи ветвления с

изменением связи. Инструкция bclr использует LR именно таким образом, разрешая косвенный переход по любому адресу.

При использовании в качестве адреса возврата инструкцией bclr биты 30:31 регистра LR игнорируются, поскольку все адреса инструкций находятся на границах слов.

Доступ к LR является непривилегированным.

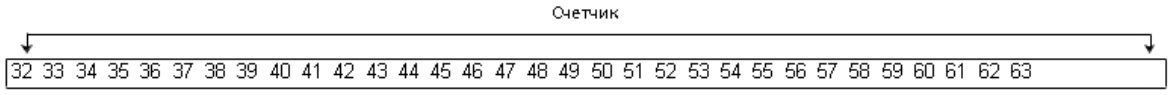


Биты	Наименование поля	Описание
0:31	LR	Содержимое регистра связи Целевой адрес инструкции bclr.

Регистр-счетчик (CTR)

Запись в CTR выполняется из GPR с помощью инструкции mtspr, а его содержимое может считываться в GPR с помощью инструкции mfspr. Содержимое CTR может быть использовано как счетчик циклов, который декрементируется и проверяется инструкциями условного перехода, в которых декремент счетчика указан как одно из условий ветвления (поле инструкции BO[2] = ‘0’). Другой вариант - содержание CTR может указывать целевой адрес для инструкции bcctr, разрешая косвенный переход по любому адресу.

Доступ к CTR является непривилегированным.



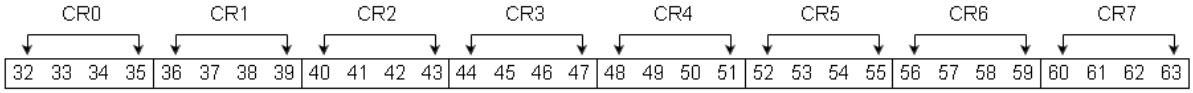
Биты	Наименование поля	Описание
32:63	Счетчик	Используется как счетчик для инструкций условного перехода с декрементом, или как целевой адрес для инструкций bcctr.

Регистр условий (CR)

CR используется для записи определенной информации (условий), связанной с результатами выполнения различных инструкций, которым разрешено изменять содержимое CR. Бит в регистре CR также может выбираться в качестве одного из проверяемых условий в инструкции условного перехода.

Регистр CR состоит из восьми 4-битовых полей (CR0 - CR7).

Доступ к CR является непривилегированным.



Биты	Наименование поля	Описание
32:35	CR0	Поле 0 регистра условий.
36:39	CR1	Поле 1 регистра условий.
40:43	CR2	Поле 2 регистра условий.
44:47	CR3	Поле 3 регистра условий.
48:51	CR4	Поле 4 регистра условий.
52:55	CR5	Поле 5 регистра условий.
56:49	CR6	Поле 6 регистра условий.
60:63	CR7	Поле 7 регистра условий.

Таблица Б.29 - Инструкции, изменяющие содержимое CR

Целочисленные						Управление процессором	Управление памятью	Сопроцессор
Обращение к памяти	Арифметические	Логические	Сравнение	Циклический сдвиг	Сдвиг	CR-логика и управление регистрами	TLB Управление	Арифметические и логические
stwcx.	add.[o] addc.[o] adde.[o] addic. addme.[o] addze.[o] subf.[o] subfc.[o] subfe.[o] subfme.[o] subfze.[o] mulhw. mulhwu. mullw.[o] divw.[o] divwu.[o] neg.[o]	and. andi. andis. andc. nand. or. orc. nor. xor. eqv. extsb. extsh. cntlzw.	cmp cmpi cmpl cmpli	rlwimi. rlwinm. rlwnm.	slw. srw. sraw. srawi.	crand crandc creqv crnand crnor cror crorc crxor mcrf mcrxr mtcrf	tlbsx.	macchw.[o] macchws.[o] macchwsu.[o] macchwu.[o] machhw.[o] machhws.[o] machhwsu.[o] machhwu.[o] maclhw.[o] maclhws.[o] maclhwsu.[o] maclhwu.[o] nmacchw.[o] nmacchws.[o] nmachhw.[o] nmachhws.[o] nmaclhw.[o] nmaclhws.[o] mulchw. mulchwu. mulhhw. mulhhwu. mullhw. mullhwu. dlmzb.

В Power ISA представлена подробная информация о том, как каждая из этих инструкций изменяет CR. В общем случае обращение к CR выполняется одним из следующих способов:

- mfcr (пересылка из регистра условий) считывает содержимое регистра CR в регистр GPR. Следует учитывать, что эта инструкция не изменяет CR, и поэтому она отсутствует в таблице.
- Инструкции условного перехода могут назначать бит CR в качестве условия ветвления. Следует учитывать, что эти инструкции не изменяют CR, и поэтому они отсутствуют в таблице.
- mtcrf (пересылка в поля регистра условий) устанавливает указанные поля CR записью в CR из GPR, под управлением поля маски, входящей в состав инструкции.
- mcrf (пересылка в регистр условий из регистра состояния и режима работы блока операций с плавающей запятой [FPSCR]) изменяет указанное поле CR, копируя в него другое указанное поле CR.
- mcrxr (пересылка в регистр условий из регистра целочисленных исключений [XER]) копирует определенные биты XER указанное поле CR и очищает соответствующие биты XER.
- Инструкции целочисленного сравнения изменяют указанное поле CR.
- CR-логические инструкции записывают в указанный бит CR результат одной из восьми логических операций над указанной парой битов CR.
- Определенные формы разных целочисленных инструкций (“.” формы) неявно изменяют CR[CR0] - как и определенные формы инструкций сопроцессора, реализованного в ядре PowerPC 476FP.

Неявноеенные формы разных целочисленных инструкций. Это различные инструкции с точкой, в мнемонике которых присутствует “.”. Большинство этих инструкций изменяют CR[CR0] согласно арифметическому сравнению 0 с 32-битовым результатом, который инструкция записывает в файл GPR. То есть, после выполнения операции, которая определена инструкцией, 32-битовый результат, записанный в файл GPR, сравнивается с 0 при помощи сравнения со знаком, независимо от того, являлась ли выполненная операция операцией со знаком. Например, логические инструкции, такие как and., or. и nor. изменяют CR[CR0] согласно сравнению со знаком с 0, даже если результат такой логической операции обычно не интерпретируется как число со знаком. Для каждой такой инструкции с точкой отдельные биты CR[CR0] изменяются следующим образом:

CR[CR0[0]] — LT	Меньше 0; устанавливается, если наиболее значимый бит 32-битового результата равен ‘1’.
CR[CR0[1]] — GT	Больше 0; устанавливается, если 32-битовый результат не нулевой, и наиболее значимый бит результата равен ‘0’.
CR[CR0[2]] — EQ	Равен 0; устанавливается, если 32-битовый результат равен ‘0’.
CR[CR0[3]] — SO	Суммарное переполнение; копируется XER[SO] при завершении операции (включая любые изменения XER[SO], выполненные самой операцией).

Примечание: При арифметическом переполнении знак результата выполнения инструкции, указанный в CR[CR0], может не отражать истинный (с бесконечной точностью) алгебраический результат инструкции, изменившей CR0. Например, если инструкция.add складывает два больших положительных числа, и результат настолько велик, что не может быть представлен как число в двоичном дополнительном коде в 32-битовом регистре, то возникает переполнение, и устанавливается CR[CR0[0]], даже несмотря на то, что результат сложения с бесконечной точностью положителен.

Аналогичным образом, сложение наибольшего 32-битового отрицательного числа в двоичном дополнительном коде (x‘8000 0000’) с самим собой приведет к арифметическому переполнению, а в целевой регистр запишется x‘0000 0000’. CR[CR0[2]] установится, указывая на нулевой результат, хотя результат с бесконечной точностью отрицателен.

CR[CR0[3]] является копией XER[SO] по завершению инструкции, независимо от того, изменяла ли XER[SO] эта инструкция, изменяющая CR[CR0].

Примечание: Если инструкция вызывает арифметическое переполнение, но не относится к тем, которые изменяют XER[SO], то помещаемое в CR[CR0[3]] значение не отражает арифметического переполнения, вызванного инструкцией; в нем просто копируется значение XER[SO], которое уже хранилось в XER до выполнения инструкции, изменяющей CR[CR0].

Существует несколько инструкций с точкой, которые не изменяют CR[CR0] так, как было описано выше. Эти инструкции: индексное условное сохранение слова (stwcx.), индексный поиск TLB (tlbsx.) и определение самого левого нулевого байта (dlmzb).

Инструкции целочисленного сравнения, изменяющие CR

Инструкции целочисленного сравнения изменяют указанное поле CR результатом сравнения двух 32-битовых числе, первое из которых берется из GPR, а второе либо является непосредственным значением, либо берется из другого GPR. Различают два типа инструкций целочисленного сравнения, арифметические и логические, и они различаются интерпретацией 32-битовых чисел, которые сравниваются. Для арифметического сравнения числа считаются имеющими знак, а для логического сравнения - не имеющими. Например, рассмотрим сравнение 0 с x‘FFFF FFFF’. При арифметическом сравнении 0 больше; при логическом сравнении x‘FFFF FFFF’ больше.

Инструкция сравнения может направить результат в любое поле регистра CR. Поле BF (биты 6:8) инструкции указывает, какое поле регистра CR изменяется. После сравнения указанное поле CR интерпретируется следующим образом:

CR[BF]0 — LT	Первый операнд меньше второго операнда.
CR[BF]1 — GT	Первый операнд больше второго операнда.
CR[BF]2 — EQ	Первый операнд равен второму операнду.
CR[BF]3 — SO	Общее переполнение; копия XER[SO].

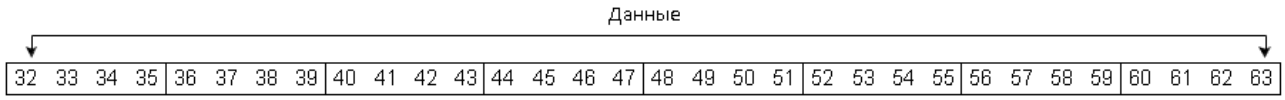
Целочисленные операции

К целочисленным операциям относится обмен данными (загрузка/сохранение) между памятью и регистрами GPR, а также различные операции над значениями в GPR и других регистрах (категории целочисленных инструкций приведены в Табл. 2-6 на стр. 49). В данном разделе описываются регистры, используемые для целочисленных операций, а также их изменение различными инструкциями.

Регистры общего назначения (GPR)

Ядро PowerPC 476FP содержит 32 регистра GPR. Содержимое этих регистров может передаваться в память и записываться из памяти при помощи инструкций целочисленного обращения к памяти. Большинство других инструкций выполняет операции над содержимым GPR.

Доступ к GPR является непривилегированным.



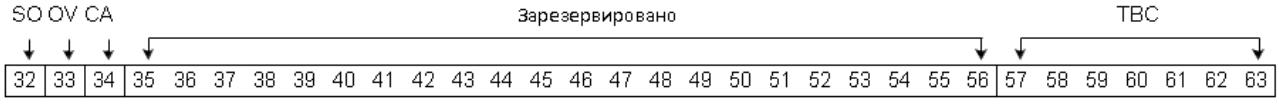
Биты	Наименование поля	Описание
32:63	Данные	Данные регистра общего назначения

Регистр исключений операций с фиксированной запятой (XER)

В XER записываются признаки переполнения и переноса от целочисленных арифметических инструкций и инструкций сдвига. Он также предоставляет счетчик байт для целочисленных индексных строковых инструкций обращения к памяти (lswx and stswx). Следует отметить, что термин исключение в названии этого регистра не связан с теми исключениями, которые имеют отношение к прерываниям, о относится к арифметическим исключениям переноса и переполнения.

Ниже изображены поля XER; в Табл. 2-27 и Табл. 2-28 приведены инструкции, которые изменяют поля XER[SO, OV] и XER[CA]. В разделах, следующих за рисунком и таблицами, поля XER описываются более подробно.

Доступ к XER является непривилегированным.



Биты	Наименование поля				Описание
32	SO	Общее переполнение. 0 Переполнения не произошло. 1 Переполнение произошло. Может быть установлен mtspr, целочисленной инструкцией процессора или сопроцессора с опцией [o]; может быть сброшен mtspr или mcrxr.			
33	OV	Переполнение. 0 Переполнения не произошло. 1 Переполнение произошло. Может быть установлен mtspr, целочисленной или назначенной инструкцией с опцией [o]; может быть сброшен mtspr, mcrxr, целочисленной или назначенной инструкцией с опцией [o]			
34	CA	Перенос. Переноса не произошло. Перенос произошел. Может быть установлен mtspr или определенными целочисленными арифметическими инструкциями и инструкциями сдвига; может быть сброшен mtspr, mcrxr или определенными целочисленными арифметическими инструкциями и инструкциями сдвига.			
35:56	Зарезервировано				
57:63	TBC	Счетчик переданных байтов. Используется как счетчик байтов lswx ustswx; записывается dlmzb[.] и mtspr.			

Таблица Б.30 Инструкции, изменяющие XER[SO,OV]

Целочисленные арифметические инструкции					Сопроцессор		Управление процессором
Сложение	Вычитание	Умножение	Деление	Инвертирование	Умножение с накоплением	Умножение с накоплением отрицательных чисел	Управление регистрами
addo[.] addco[.] addeo[.] addmeo[.] addzeo[.]	subfo[.] subfco[.] subfeo[.] subfmeo[.] subfzeo[.]	mullwo[.]	divwo[.] divwuo[.]	nego[.]	macchwo[.] macchwso[.] macchwsuo[.] macchwuo[.] machhwo[.] machhwso[.] machhwsuo[.] machhwuo[.] maclhwo[.] maclhwso[.] maclhwsuo[.] maclhwuo[.]	nmacchwo[.] nmacchwso[.] nmachhwo[.] nmachhwso[.] nmaclhwo[.] nmaclhwso[.]	mtspr mcrxr

Таблица Б.31 - Инструкции, изменяющие XER[CA]

Целочисленные арифметические инструкции		Целочисленный сдвиг	Управление процессором
Сложение	Вычитание	Алгебраический сдвиг	Управление регистрами
addc[o][.] adde[o][.] addic[.] addme[o][.] addze[o][.]	subfc[o][.] subfe[o][.] subfic subfme[o][.] subfze[o][.]	sraw[.] srawi[.]	mtspr mcrxr

Поле общего переполнения (SO)

Это поле устанавливается в ‘1’, когда выполняется инструкция, в результате которой XER[OV] устанавливается в ‘1’, за исключением инструкции mtspr(XER), которая записывает в XER[SO] значение из (RS[32]) и записывает в XER[OV] значение из (RS[33]). После установки

XER[SO] не сбрасывается до выполнения либо инструкции mtspr(XER) с данными, которые явно записывают ‘0’ в XER[SO], либо инструкции mcrxr. Инструкция mcrxr устанавливает XER[SO] (и XER[OV, CA]) в ‘0’ после копирования всех трех полей в CR[CR0[0:2]] (и установки CR[CR0[3]] в ‘0’).

Учитывая такое поведение, XER[SO] не обязательно указывает на переполнение в результате последней целочисленной арифметической операции; переполнение произошло в какой-то момент времени после последней очистки XER[SO] инструкцией mtspr(XER) или mcrxr.

XER[SO] считывается (вместе с остальными полями XER) в регистр GPR инструкцией mfspr(XER). Кроме того, различные целочисленные инструкции копируют XER[SO] в CR[CR0[3]].

Поле переполнения (OV)

Это поле изменяется определенными целочисленными арифметическими инструкциями, чтобы указать, может ли результат операции с бесконечной точностью быть представлен 32 битами. Для тех целочисленных арифметических инструкций, которые изменяют XER[OV] и результат которых имеет знак, XER[OV] = ‘1’ если результат больше, чем 231 – 1 или меньше чем –231; в противном случае, XER[OV] = ‘0’. Для тех целочисленных арифметических операций, которые изменяют XER[OV] и результат которых не имеет знака (некоторые целочисленные инструкции деления и инструкции умножения с накоплением для сопроцессора), XER[OV] = ‘1’ если результат больше 232–1; в противном случае, XER[OV] = ‘0’. Более подробное описание условий, при которых целочисленные инструкции деления устанавливают XER[OV] в 1’.

Инструкции mtspr(XER) и mcrxr также изменяют XER[OV]. В частности, инструкция mcrxr устанавливает XER[OV, CA] в ‘0’ после копирования всех трех полей в CR[CR0[0:2]] (и установки CR[CR0[3]] в ‘0’). mtspr(XER) записывает в XER[OV] значение из (RS[33]).

XER[OV] считывается (вместе с остальными полями XER) в регистр GPR инструкцией mfspr(XER).

Поле переноса

Это поле изменяется некоторыми целочисленными арифметические операции (перенос и расширенные версии сложения и вычитания) для указания того, был ли перенос наиболее значимого бита 32-битового результата. XER[CA] = ‘1’ указывает на перенос. Целочисленные инструкции алгебраического сдвига вправо изменяют XER[CA] для указания сдвигов 1 из наименее значимого бита результата, если исходный операнд отрицательный.

Инструкции mtspr(XER) и mcrxr также изменяют XER[CA]. Инструкция mcrxr устанавливает XER[CA] (и XER[SO, OV]) в ‘0’ после копирования всех трех полей в CR[CR0[0:2]] (и установки CR[CR0[3]] в ‘0’). mtspr(XER) записывает в XER[CA] значение из (RS[34]).

XER[CA] считывается (вместе с остальными полями XER) в регистр GPR инструкцией mfspr(XER). Кроме того, расширенные версии целочисленных арифметических инструкций сложения и вычитания используют XER[CA] в качестве исходного операнда для арифметических операций.

Поле счетчика переданных байтов (TBC)

Поле TBC используется целочисленными индексными строковыми инструкций обращения к памяти (lswx and stswx) в качестве счетчика байтов. Поле TBC изменяется инструкцией dlmzb[.] со значением, указывающим количество байт, включая нулевой байт, которые определяются инструкцией. Поле TBC также записывается инструкцией mtspr(XER) значением из (RS[25:31]).

XER[TBC] считывается (вместе с остальными полями XER) в регистр GPR инструкцией mfspr(XER).

Управление процессором

Ядро PowerPC 476FP предоставляет несколько регистров состояния и режима работы процессора. К ним относятся следующие регистры:

- Регистр управления процессором (MSR)
- Управляет прерываниями и другими функциями процессора.
- Общие регистры специального назначения (SPRG) SPR предназначены для использования программами общего назначения.
- Регистр версии процессора (PVR)
- Указывает конкретную реализацию процессора.
- Регистр идентификатора процессора (PIR)
- Указывает конкретный процессор в многопроцессорной системе.
- Регистр конфигурации ядра 0 (CCR0)
- Управляет определенными функциями процессора, такими как предварительная выборка инструкций.
- Регистр конфигурации ядра 1 (CCR1)
- CCR1 может вызывать все возможные исключения ошибки четности, чтобы проверить правильность работы обработчика исключений проверки процессора. Другие биты CCR1 могут вызывать полную очистку кэша данных и выбирать вход таймера процессора отличный от CPUCLOCK.
- Регистр конфигурации ядра 2 (CCR2) CCR2 определяет дополнительные параметры кэша.
- Конфигурация при сбросе (RSTCFG)
- Указывает значения определенных полей TLB, которые должны устанавливаться при сбросе.
- Регистр непосредственного префикса регистра управления устройством (DCRIPR)
- DCRIPR содержит старшие 22 бита адреса, которые используются инструкциями mtdcr and mfdcr. Этот SPR имеет шестнадцатеричный адрес x‘37B’, он допускает чтение и запись и является привилегированным.

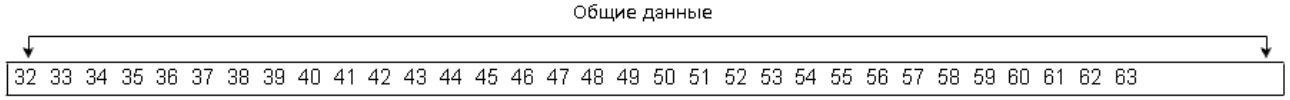
Общие регистры специального назначения (USPRG0, SPRG0 - SPRG8)

USPRG0 и SPRG0 - SPRG8 являются общими и предназначены для использования системными программами. Система часто использует эти регистры как места временного хранения. Например, подпрограмма может сохранять содержимое GPR в SPRG, а потом восстанавливать состояние GPR. Это быстрее, чем сохранение в памяти. Эти регистры записываются инструкцией mtspr, а считываются инструкцией mfspr.

Доступ к USPRG0 непривилегированный как для чтения, так и для записи.

Доступ к SPRG4 - SPRG7 непривилегированный для чтения, но привилегированный для чтения/записи и поэтому для чтения и записи используются разные номера SPR. Доступ к этим регистрам указан в Табл.2-5 на стр. 43.

Доступ к SPRG0 - SPRG3 привилегированный как для чтения, так и для записи; доступ к SPRG8 привилегированный как для чтения, так и для записи.

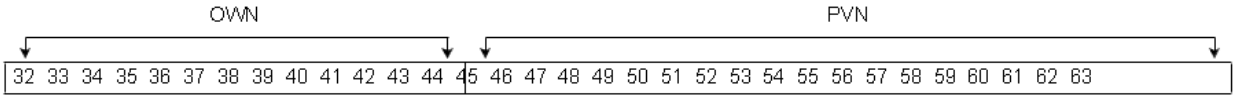


Биты	Наименование	Описание
32:63	Общие данные	Значение для программы; аппаратная часть не использует это значение.

Регистр версии процессора (PVR)

PVR предназначен только для чтения и обычно используется для идентификации конкретной реализации процессорного ядра и кристалла. Программа может считывать PVR, чтобы определить характеристики аппаратной части ядра и кристалла. PVR может считываться в GPR инструкцией mfspr.

Доступ к PVR является привилегированным.

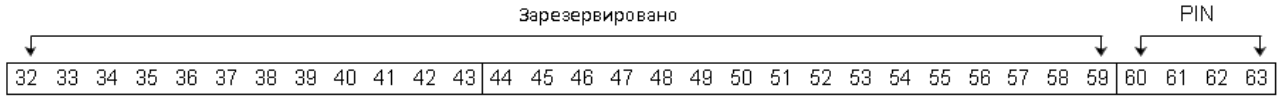


Биты	Наименование поля	Описание
32:43	OWN	Идентификатор владельца. Указывает владельца ядра. Это зависящее от реализации значение (после сброса и т.д.) определяется входными сигналами ядра.
44:63	PVN	Номер версии процессора Это зависящее от реализации значение указывает конкретную версию и применение процессорного ядра внутри кристалла. Это значение (после сброса и т.д.) определяется входными сигналами ядра.

Регистр идентификатора процессора (PIR)

Регистр PIR предназначен только для чтения и указывает конкретный процессор в многопроцессорной конфигурации, позволяя программе определить, какой именно процессор работает. Эта возможность важна для операционной системы в многопроцессорных конфигурациях. PIR может считываться в GPR инструкцией mfspr.

Доступ к PIR является привилегированным.

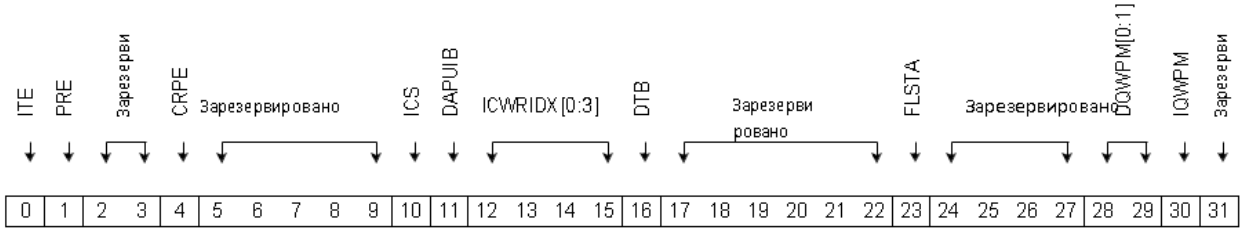


Биты	Наименование поля	Описание
32:59	Зарезервировано	
60:63	PIN	Идентификационный номер процессора (PIN).

Регистр конфигурации ядра 0 (CCR0)

CCR0 управляет некоторыми специальными функциями кристалла, в том числе работой кэша данных и сопроцессора, спекулятивной выборкой инструкций, трассировкой и работой инструкций предвыборки блока кэша. Запись в CCR0 выполняется из GPR с помощью инструкции mtspr, а его содержимое может считываться в GPR с помощью инструкции mfspr. Перекрестная ссылка после описания битового поля указывает раздел данного документа, в котором подробно описано каждое поле.

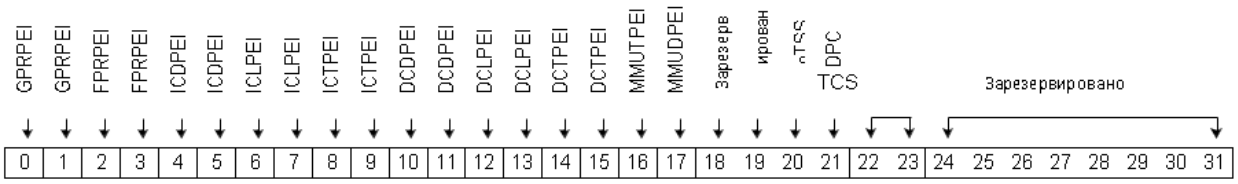
Доступ к CCR0 является привилегированным.



Биты	Наименование поля	Описание
0	ITE	Разрешение внутренней трассировки. 0 Запретить внутреннюю трассировку. 1 Разрешить внутреннюю трассировку. Инструмент или программа отладки должны установить этот бит, чтобы разрешить трассировку инструкций. Если пользовательская программа очищает ITE, любая текущая операция трассировки, выполняемая отладчиком, прекращается.
1	PRE	Разрешение восстановления четности. 0 Для кэша данных разрешен режим частичного восстановления четности. 1 Для кэша данных разрешен режим полного восстановления четности. Должен быть установлен в ‘1’, чтобы гарантировать полное исправление ошибок четности блока управления памятью (MMU) и кэша данных.
2:3	Зарезервировано	
4	CRPE	Разрешение считывания информации о четности. 0 Запретить считывание информации о четности. 1 Разрешить считывание информации о четности. Если разрешено, то при выполнении следующих инструкций информация о четности загружается в соответствующий регистр: Инструкция Регистр icread ICDBTRH, ICDBTRL, ICDBDR1 dcread DCDBTRH, DCDBTRL tlbre GPR (см. операцию tlbre)
5:9	Зарезервировано	
10	ICS	Размер запроса icbi. 0 32-байтовый запрос icbi. 1 128-байтовый запрос icbi.
11	DAPUIB	Запрещение передачи инструкций в APU. 0 Разрешено. 1 Запрещено. Инструкции не передаются в APU для декодирования.
12:15	ICWRIDX[0:3]	Индекс записи кэша инструкций (для JTAG). Указывает значение индекса для записи в кэш инструкций.
16	DTB	Запрещение передачи трассировки. 0 Разрешено. 1 Запрещено; информация о трассировке не передается. Этот механизм служит средством снижения энергопотребления, когда трассировка инструкций не нужна.
17:22	Зарезервировано	
23	FLSTA	Принудительное выравнивание загрузки/сохранения 0 Не формируется исключение выравнивания при выполнении целочисленных инструкций обращения к памяти, независимо от выравнивания. 1 Формируется исключение выравнивания при выполнении целочисленных инструкций обращения к памяти, если адрес данных не приходится на границу операнда.
24:27	Зарезервировано	
28:29	DQWPM[0:1]	Режим предсказания счетверенных слов кэша данных. Нет предсказания, вызывает задержку. Использовать EA[19]. Использовать последнее значение счетверенного слова EA[19]. Использовать NOT EA[19].
30	IQWPM	Режим предсказания счетверенных слов кэша инструкций. 0 Использовать последнее значение счетверенного слова EA[19]. 1 Использовать EA[19].
31	Зарезервировано	

Регистр конфигурации ядра 1 (CCR1)

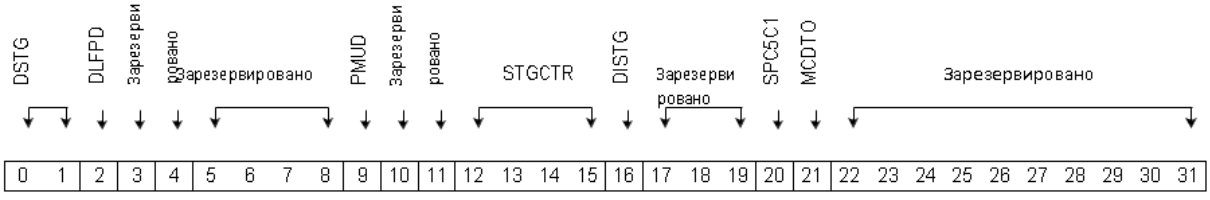
Биты 0:17 CCR1 могут вызывать все возможные исключения ошибки четности, чтобы проверить правильность работы обработчика исключений проверки процессора. Другие биты CCR1 могут вызывать полную очистку кэша данных или выбирать вход таймера процессора отличный от CPUClock. Запись в CCR1 выполняется из GPR с помощью инструкции mtspr, а его содержимое может считываться в GPR с помощью инструкции mfspr. Доступ к CCR1 является привилегированным.



Биты	Наименование поля	Описание
0:1	GPRPEI	Вставка ошибки четности GPR. GPRPEI[0]: Записывает четность в конвейер I файла регистров GPR, если установлен. GPRPEI[1]: Записывает четность в конвейер L файла регистров GPR, если установлен.
2:3	FPRPEI	Вставка ошибки четности регистра операций с плавающей запятой (FPR) FPRPEI[2]: Записывает четность в первый FPR, если установлен. FPRPEI[3]: Записывает четность во второй FPR, если установлен.
4:5	ICDPEI	Вставка ошибки четности кэша инструкций. 0 Запись нечетности (норма). 1 Запись четности (симуляция ошибок четности). Управляет инверсией битов четности, которые записываются при заполнении кэша инструкций. ICDPEI[4]: Запись четности в левый массив. ICDPEI[5]: Запись четности в правый массив.
6:7	ICLPEI	Вставка ошибки четности LRU кэша инструкций. 0 Запись нечетности (норма). 1 Запись четности (симуляция ошибки четности). ICLPEI[6]: Запись в левый массив. ICLPEI[7]: Запись в правый массив.
8:9	ICTPEI	Вставка ошибки четности тега кэша инструкций. 0 Запись нечетности (норма). 1 Запись четности (симуляция ошибки четности). Управляет инверсией битов четности, которые записываются для поля данных в кэше инструкций. ICTPEI[8]: Запись четности в левый массив. ICTPEI[9]: Запись четности в правый массив.
10:11	DCDPEI	Вставка ошибки четности кэша данных. 0 Запись нечетности (норма). 1 Запись четности (симуляция ошибки четности). Управляет инверсией битов четности, которые записываются для поля данных в кэше данных. DCDPEI[10]: Запись четности в массив четности. DCDPEI[11]: Запись четности данных в массив нечетности.
12:13	DCLPEI	Вставка ошибки четности LRU кэша данных (массив четности). 0 Запись нечетности (норма). 1 Запись четности (симуляция ошибки четности). Управляет инверсией битов четности, которые записываются для поля LRU в кэше данных. DCLPEI[12]: Запись четности LRU кэша данных в массив четности. DCLPEI[13]: Запись четности LRU кэша данных в массив нечетности.

Биты	Наименование поля	Описание
14:15	DCTPEI	Вставка ошибки четности тегов кэша данных (массив четности). 0 Запись нечетности (норма). 1 Запись четности (симуляция ошибки четности). Управляет инверсией битов четности, которые записываются для поля тегов в кэше данных. DCTPEI[14]: Запись четности в массив четности. DCTPEI[15]: Запись четности в массив нечетности.
16	MMUTPEI	Вставка ошибки четности тега модуля управления памятью. 0 Запись нечетности (норма). 1 Запись четности (симуляция ошибки четности). Управляет инверсией битов четности, которые записываются для поля тегов в MMU.
17	MMUDPEI	Вставка ошибки четности данных модуля управления памятью. 0 Запись нечетности (норма). 1 Запись четности (симуляция ошибки четности). Управляет инверсией битов четности, которые записываются для поля тегов в MMU.
18	Зарезервировано	
19	TSS	Выбор источника для таймера 0 Источник таймера CPU - тактовый генератор CPU. 1 Источник таймера CPU - альтернативный тактовый генератор.
20	Зарезервировано	
21	DPC	Запрет проверки четности (при сбросе этот бит устанавливается в ‘1’). 0 Проверка четности разрешена в кэше L1 ядра. 1 Проверка четности запрещена в кэше L1 ядра.
22:23	TCS	Выбор таймера, выбор сторожевого таймера. Значение таймера CPU увеличивается на каждый передний фронт входного сигнала синхронизации. Значение таймера CPU увеличивается на каждый четвертый передний фронт входного сигнала синхронизации. Значение таймера CPU увеличивается на каждый восьмой передний фронт входного сигнала синхронизации. Значение таймера CPU увеличивается на каждый шестнадцатый передний фронт входного сигнала синхронизации.
24:31	Зарезервирован	

Регистр конфигурации ядра 2 (CCR2)



Биты	Наименование поля	Описание
0:1	DSTG	Запрет объединения сохранения Если разрешено, сохранение всех байтов внутри полустроки L1 может быть объединено в одну пересылку. Разрешено только объединение сохранения непрерывных, перекрывающихся байтов. Запрещено независимое объединение сохранения. Зарезервировано Запрещено любое объединение сохранения.
2	DLFPD	Запрет предсказания заполнения строки кэша данных. 0 Разрешено предсказание совпадения заполнения строки. 1 Запрещено предсказание совпадения заполнения строки.
3	Зарезервировано	
4	DSTI	Запрет инвалидации теневого TLB. 0 В случае контекстной синхронизации инвалидировать теневые TLB (ITLB, DTLB). 1 Если установлен, то при контекстной синхронизации isync теневые TLB не инвалидируются.
5:8	Зарезервировано	
9	PMUD	Запрет блока монитора производительности 0 Разрешить счет PMU 1 Запретить PMU счет различных событий
10	Зарезервировано	
11	DCSTGW	Запрет объединения сохранения кэшируемой сквозной записи 0 Разрешено объединение кэшируемых сохранений с W = 1, но они должны быть непрерывными. 1 Запрещено объединение кэшируемых сохранений с W = 1.
12:15	STGCTR	Счетчик объединения сохранения Это поле описывает, как долго запрос сохранения остается в SBQ, прежде чем посылается запрос записи. Этот счетчик инициализируется в STGCTR x 2, когда SBQ0 загружается для сохранения. Его значение уменьшается на единицу каждый цикл, пока не достигнет нуля - или до новой инициализации. При достижении нулевого значения выполняется пересылка содержимого SBQ0. Объединение выполняется только сохранениями, для которых она разрешена.
16	DISTG	Запрет объединения запрещенных сохранений 0 Запрещенные сохранения могут подлежать объединению только тогда, когда они находятся в одной и той же половине строки кэша L1, и строка кэша непрерывная и не охраняемая. 1 Запрещенные сохранения не подлежат объединению.
17:19	Зарезервировано	
20	SPC5C1	Поле ‘AT’ ICU статического предсказания ветвления на Code C5 и C1 0 Не использовать поле ‘AT’ статического предсказания ветвления. 1 Использовать поле ‘AT’ статического предсказания ветвления.
21	MCDTO	Разрешение проверки процессора по тайм-ауту DCR 0 Нет тайм-аута DCR 1 Разрешена проверка процессора по тайм-ауту DCR.
22:31	Зарезервировано	

Конфигурация при сбросе (RSTCFG)

Регистр RSTCFG, предназначенный только для чтения указывает значения определенных полей TLB, которые должны устанавливаться при сбросе. Доступ к RSTCFG является привилегированным.



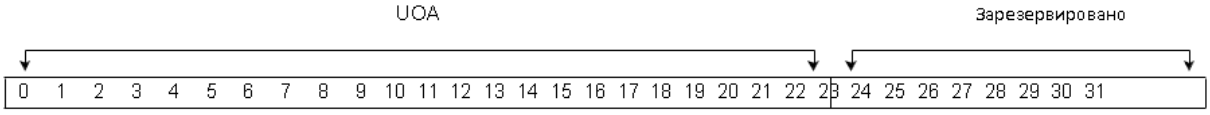
Биты	Наименование поля	Описание
0:1	Зарезервировано	
2:11	ERPN	Расширенный действительный номер страницы, только для чтения. Устанавливается значение, связанное с входными данными ядра (значениями конфигурации, определяемые реализацией кристалла).
12:16	Зарезервировано	
17	E	Порядок следования байтов, только для чтения. Устанавливается значение, связанное с входными данными ядра (значениями конфигурации, определяемые реализацией кристалла).
18:27	Зарезервировано	
28:31	U	U0 - U3, только для чтения. Устанавливается значение, связанное с входными данными ядра (значениями конфигурации, определяемые реализацией кристалла).

Регистр непосредственного префикса регистра управления устройством (DCRIPR)

Регистр непосредственного префикса регистра управления устройством DCRIPR содержит старшие 22 бита адреса, которые используются инструкциями mtdcr and mfdcr. Этот SPR имеет шестнадцатеричный адрес x‘37B’, он допускает чтение и запись и является привилегированным. Зависит от реализации; то есть, не входит в Book E-III спецификации архитектуры.

Для поддержки инструкций mtdcr[u]x и mfdcr[u]x интерфейс DCR добавляет 22 выходных вывода для старших битов адресов и один выходной вывод для указателя привилегированный/непривилегированный. Следует отметить, что привилегированный сигнал указывает, какой тип кода операции вызвал операцию DCR, присутствующую на интерфейсе DCR и прямо не связанную с битом MSR[PR]. Привилегированный (режим супервизора) код может выполнять любой из шести кодов операций DCR на интерфейсе с любым значением, указанным на привилегированном сигнале. Непривилегированный (режим пользователя) код только генерирует трафик DCR с указанием непривилегированного режима на интерфейсе. Если код пользовательского режима пытается выполнить привилегированный код операции, вырабатывается исключение, сигнализирующее о нарушении привилегии.

Доступ к DCRIPR является привилегированным.



Биты	Наименование поля	Описание
0:21	UOA	Старшие разряды адреса. Определяется реализацией. Используется для старших битов адреса DCR.
22:31	Зарезервировано	

Режимы пользователя и супервизора

В Power ISA определены два рабочих состояния, или режима: супервизора (привилегированный), и пользователя (непривилегированный). Режим работы процессора определяется MSR[PR]. Когда MSR[PR] is ‘0’, процессор работает в режиме супервизора и может выполнять все инструкции и обращаться ко всем регистрам, включая привилегированные. Когда MSR[PR] is ‘1’, процессор работает в режиме пользователя и может выполнять только непривилегированные инструкции и обращаться к непривилегированным регистрам. Попытка выполнить привилегированную инструкцию или обратиться к привилегированному регистру в режиме пользователя вызывает программное прерывание типа исключения привилегированной инструкции.

Следует отметить, что название PR поля MSR указывает на старое название режима пользователя "problem state" (задачный режим). Таким образом, значение ‘1’ в поле указывает на задачный режим, а не на привилегированный режим, как можно было бы подумать.

Привилегированные инструкции

Следующие инструкции являются привилегированными и не могут выполняться в режиме пользователя:

Таблица Б.32 - Привилегированные инструкции

Инструкция	Комментарии
dcbi	
dci	
dcread	
ici	
icread	
mfdcr	
mfmsr	
mfspr	Для любого номера SPR с SPRN[5] = ‘1’
mtdcr	
mtmsr	
mtspr	Для любого номера SPR с SPRN[5] = ‘1’
rfci	
rfi	
rfmci	
tlbre	
tlbsx	
tlbsync	
tlbwe	
wrtee	
wrteei	

Привилегированные SPR

Большинство SPR являются привилегированными. К определенным непривилегированным SPR относятся только LR, CTR, XER, USPRG0, SPRG 3 - 7 (только для чтения), TBU (только для чтения) и TBL (только для чтения). Ядро PowerPC 476FP также обращается со всеми номерами регистров SPR с ‘1’ в бите 5 поля SPRN как с привилегированными, независимо от того является ли данный номер SPR определенным, или нет. Таким образом, ядро вызывает программное прерывание типа исключения привилегированной инструкции при любой попытке обращения к такому номеру SPR в режиме пользователя. Кроме того, ядро вызывает программное прерывание исключения типа недопустимой инструкции при любой попытке обращения к undefined номеру SPR с ‘0’ в SPRN[5], в режиме пользователя. Тем не менее, результат обращения к не определенному номеру SPR в режиме супервизора является неопределенным, независимо от значения SPRN[5].

Спекулятивный доступ

Power ISA разрешает реализациям выполнять спекулятивный доступ к памяти как для выбора инструкций, так и для загрузки данных. Спекулятивный доступ определяется как любой доступ, который не требуется моделью последовательного выполнения (SEM).

Например, ядро PowerPC 476FP спекулятивно предварительно выбирает инструкции предсказанной ветви условного ветвления; если впоследствии переход не совпадает с предсказанным, то выборка инструкций из предсказанной ветви не востребуется SEM и поэтому является спекулятивной. Аналогичным образом, ядро PowerPC 476FP выполняет инструкции загрузки неупорядоченно и может считывать данные из памяти для загрузки инструкции, которая расположена после неопределенного ветвления.

Однако в некоторых случаях спекулятивный доступ неуместен. Например, попытка доступа к данным по адресам, на которые распределены устройства ввода/вывода, может привести к проблемам. Если устройство ввода-вывода является последовательным портом, то его спекулятивное чтение может вызвать потерю данных.

Архитектура предоставляет два механизма для защиты от ошибочного доступа к таким адресам памяти с необычным поведением. Первый - это охраняемый (G) атрибут памяти, который защищает от спекулятивного обращения к данным. Второй - это механизм разрешения выполнения, который защищает от спекулятивной выборки инструкций.

Синхронизация

Ядро PowerPC 476FP поддерживает операции синхронизации типа d. Различают три вида синхронизации, зависящие от архитектуры, каждая из которых описана в следующих разделах.

Контекстная синхронизация

Контекст программы - это среда, в которой выполняется программа. Например, к контексту относится режим (пользователя или супервизора), а также пространство адресов трансляции и атрибуты страниц памяти, к которым обращается программа. Контекст управляется содержимым определенных регистров и другими ресурсами, такими как MSR и TLB.

При определенных обстоятельствах аппаратная часть или программа должна выполнять принудительную синхронизацию контекста программы. К операциям синхронизации контекста относятся все прерывания, за исключением прерываний проверки процессора, а также инструкции isync, sc, rfi, rfci, mtmsr и rfmci. Операции синхронизации контекста удовлетворяют следующие требования:

Операция не инициируется до тех пор, пока все инструкции, предшествовавшие операции, не выполнены до той стадии в которой они сообщили обо всех исключениях, вызванных ими.

Все инструкции, предшествовавшие операции, должны завершиться в том контексте, в котором они начинались. То есть, на них не должно оказывать влияние любое изменение контекста, вызванное операцией синхронизации, или любыми инструкциями, выполняющимися после операции синхронизации контекста. Если операция представляет собой инструкцию sc (которая вызывает прерывание системного вызова) или сама является прерыванием, то она не инициируется до тех пор, пока есть ожидающие прерывания более высокого уровня.

Все инструкции, следующие за операцией, должны быть повторно выбраны и выполнены в контексте, которые установился после завершения операции синхронизации контекста и всех предшествовавших ей инструкций.

Если операция представляет собой инструкцию mtmsr, то она не инициируется до тех пор, пока все инструкции, предшествовавшие операции, не выполнены до той стадии в которой они сообщили обо всех исключениях, вызванных ими. После этого в регистр MSR записывается биты 32-63 регистра GPR, и выполняется операция синхронизации.

Операции синхронизации контекста не вызывают принудительного завершения обращений к памяти или их упорядочения - ни до, ни после операции синхронизации. Если требуются именно такие действия, необходимо использовать инструкции синхронизации памяти.

Кроме того, структурно прерывания проверки процессора не относятся к синхронизации контекста. Поэтому инструкция, предшествующая операции синхронизации контекста может вызвать прерывание проверки процессора после выполнения операции синхронизации контекста и нескольких инструкций. Для ядра PowerPC 476FP это может произойти только с исключениями данных проверки процессора, но не с исключениями инструкций проверки процессора.

В следующих разделах приведены примеры псевдокода, демонстрирующие эффекты синхронизации контекста. Сопроводительный текст объясняет, как программа может в дальнейшем гарантировать порядок сохранения.

Рассмотрим следующую последовательность самомодифицирующегося кода:

stw XYZ Сохранить по адресу XYZ с запрещенным кэшированием.

isync Выборка и выполнение инструкции по адресу XYZ.

В этой последовательности инструкция isync не гарантирует, что инструкция XYZ будет выбрана после того, как было выполнено сохранение в память. Невозможно гарантировать, какая будет выполнятся инструкция XYZ, либо старая версия, либо новая (сохраненная).

Теперь рассмотрим требуемую последовательность самомодифицирующегося кода.

stw Записать новую инструкцию в кэш данных.

dcbst Переслать новую инструкцию из кэша данных в память.

msync Упорядочить копию прежде чем инвалидировать старую инструкцию в кэше инструкций.

icbi Инвалидировать копию в кэше инструкций.

isync Аннулировать предварительно выбранную инструкцию и выбрать новую инструкцию, изменить контекст.

Этот пример иллюстрирует использование isync с изменениями контекста для средств отладки.

mtdbcr0 Разрешить событие отладки, сравнение адресов инструкций (IAC).

isync Ждать установки нового контекста регистра управления отладкой 0 (DBCR0).

XYZ Данная инструкция представляет собой адрес IAC; isync необходима для гарантии того, что событие IAC будет распознано при выполнении этой инструкции; без isync, инструкция XYZ может быть предварительно выбрана и отправлена для выполнения до распознавания того, что событие IAC было разрешено.

Последний пример - это использование isync для обращения к DCR с помощью инструкций mtdcr илимfdcr, базирующихся на регистре DCRIPR:

mtspr DCRIPR устанавливает значение DCRIPR для DCRn.

isync Устанавливает новый DCRn с помощью синхронизации контекста.

mtdcr Обращение к новому DCR с новым значением.

Синхронизация выполнения

Синхронизация выполнения - это разновидность синхронизации контекста. Выполнение операции синхронизации удовлетворяет первым двум требованиям к операциям синхронизации контекста, но не двум последним. То есть, операции синхронизации выполнения гарантируют, что предшествующие инструкции будут выполнены в старом контексте, но не гарантируют, что новые инструкции будут выполняться в новом контексте. Рассмотрим, например, сценарий, когда синхронизация выполнения требуется непосредственно перед выполнением инструкций, изменяющих TLB (таких как tlbwe). Необходимо выполнить инструкцию синхронизации, чтобы гарантировать, что все предыдущие обращения к памяти выполнили трансляцию адресов до выполнения tlbwe, чтобы инвалидировать элемент который может быть использован этими предшествующими инструкциями.

Существует пять инструкций синхронизации выполнения: wrtee, wrteei, msync, mbar и lwsync. Все инструкции синхронизации контекста также неявно являются инструкциями синхронизации выполнения, поскольку синхронизация контекста включает в себя и синхронизацию выполнения.

Power ISA предъявляет дополнительные требования к изменению MSR[EE] (бит разрешения внешнего прерывания). В частности, если инструкция wrtee или wrteei устанавливает MSR[EE] = ‘1’, и ожидает исполнения внешний ввод, декрементора или таймера фиксированного интервала, то прерывание должно быть принято до того, как выполнена инструкция, следующая за изменением MSR[EE]. В этом смысле инструкции, изменяющие MSR[EE], могут рассматриваться как синхронизирующие контекст в отношении бита MSR[EE], в том смысле, что они гарантируют, что следующие инструкции будут выполняться (или не выполняться в случае прерывания) в соответствии с новым контекстом MSR[EE].

Порядок обращения и синхронизация памяти

Синхронизация памяти принудительно устанавливает порядок обращения к памяти инструкций обращения к памяти, выполняемых ядром PowerPC 476FP. Существует три инструкции синхронизации памяти: msync, mbar и lwsync. Power ISA определяет разные требования упорядочения для этих трех инструкций, но ядро PowerPC 476FP реализует msync и mbar одинаково. Структурно инструкция msync сильнее и также является инструкцией выполнения, тогда как mbar - это эквивалент eieio. Таким образом, пользователю для будущей совместимости рекомендуется использовать mbar вместо eieio или операции барьера памяти.

Инструкция lwsync представляет собой облегченную версию msync. Тем не менее, msync гарантирует, что все предыдущие обращения к памяти были выполнены в отношении подсистемы памяти синхронизации выполнения, до выполнения любой инструкции после msync.

Примечание: Это требование выходит за рамки требований просто синхронизации выполнения, поскольку синхронизация выполнения не требует завершения предыдущих обращений к памяти.

Следующие два примера иллюстрируют разницу в использовании mbar versus msync.

stw Сохранить данные в устройстве ввода/вывода.

lwz Искусственная нагрузка из той же 32-битовой линии, чтобы сохранение имело место до msync.

msync Ожидание, пока завершится сохранение.

mtdcr Реконфигурация устройства ввода/вывода.

В этом примере mtdcr реконфигурирует устройство ввода/вывода так, что это вызвало бы неисполнение предыдущей инструкции сохранения, если бы mtdcr изменила устройство до завершения сохранения. Поскольку mtdcr не является инструкцией обращения к памяти, использование mbar вместо msync не гарантирует, что сохранение выполняется до разрешения mtdcr реконфигурировать устройство. Она лишь гарантирует, что последующие операции сохранения в памяти или любом устройстве не выполняются раньше предыдущих.

Еще один пример:

stb X Сохранить данные в устройстве ввода/вывода по адресу X, со сбросом бита состояния по адресу Y.

mbar Гарантирует, что предыдущая операция сохранения выполняется на устройстве до того, как разрешены последующие операции сохранения.

lbz Y Загрузка состояния из устройства ввода/вывода по адресу Y.

Здесь допустимо использовать mbar вместо msync, поскольку требуется лишь чтобы сохранение в устройство ввода/вывода произошло раньше загрузки. Другие инструкции, следующие за mbar, не выполняются раньше сохранения.

SPR, требующие синхронизации контекста

Ниже приводится перечень SPR, которые могут требовать синхронизации контекста при записи инструкцией mtspr:

- ID процесса (PID)
- Регистр 0 управления отладкой (DBCR0 - DBCR2)
- Сравнение адреса инструкций 1 - 4 (IAC1 - IAC4)
- Сравнение адреса кэша данных 1 - 2 (DAC1 - DAC2)
- Сравнение значения кэша данных 1 - 2 (DVC1 - DVC2)
- Регистр конфигурации MMU (MMUCR)
- Регистр дескрипции страницы реального режима (RMPD)
- Регистр конфигурации приоритета поиска супервизора (SSPCR)
- Регистр конфигурации приоритета поиска пользователя (USPCR)
- Регистр конфигурации приоритета поиска инвалидации (ISPCR)
- Регистр конфигурации ядра 0 - 2 (CCR0 - CCR2)
- Управляющий регистр сравнения кода операции инструкции (IOCCR)
- Регистр сравнения кода операции инструкции 1 - 2 (IOCR1 - IOCR2)

Следующие примеры демонстрируют эффекты синхронизации контекста.

Пример 1:
mtPID Изменение PID или виртуальной адресации.
isync Переключение контекста, в ожидании установки нового PID. XYZ
Инструкция XYZ базируется на новом PID.

Пример 2:
mtIAC Установка IAC1.
mtIAC2
mtIAC3
mtIAC4
mtDAC1 Установка DA1C.
mtDAC2
mtDVC1 Установка DAV.
mtDVC2
mtDBCR1 Установка управление отладкой IAC.
mtDBCR2 DAC, DVC.
mtDBCR0 Разрешение событий отладки.
isync Обеспечение, что контекст всех элементов управления отладкой установлен.
XYZ Все события отладки установлены в предыдущем коде и теперь действуют.

Пример 3:
mtCCR0 Изменение CCR0.
mtCCR1
mtCCR2
isync Обеспечение, что новый контекст конфигурации установлен.
XYZ Новая конфигурация действует.

Пример 4:
mtMMUCR Изменен регистр конфигурации MMU.
mtRMPD
mtSSPCR
mtUSPCR
mtISPCR
isync Обеспечение, что новый контекст конфигурации MMU установлен.
XYZ Новая среда MMU действует.

Пример 5:
mtIOCR1 Изменение IOCR1.
mtIOCR2 Изменение IOCR2.
mtIOCCR Изменение IOCCR.
isync Обеспечение, что новое управление ловушкой инструкции установлено. XYZ Новая ловушка инструкции действует.

Инструкции, требующие инструкции синхронизации контекста

Следующие инструкции требуют инструкции синхронизации контеста, чтобы обеспечить действие последующих инструкций:

tlbwe:

Выборка инструкции:

CSI (isync) требуется после выполнения tlbwe.

Обращение к операнду (данным):

CSI (isync) требуется перед и после после выполнения tlbwe.

Рекомендуемая последовательность для инструкции tlbwe:

isync (В случае обращения к операнду).

tlbwe Запись всех необходимых слов.

isync Обеспечение нового распределения TLB.

tlbivax:

Выборка инструкции:

- CSI (isync) требуется после выполнения tlbivax.
- Обращение к операнду (данным):
- CSI (isync) требуется перед и после после выполнения tlbivax.

Рекомендуемая последовательность для инструкции tlbivax:

- isync (В случае обращения к операнду).
- tlbivax
- tlbivax (несколько инструкций tlbivax при необходимости)
- tlbsync
- msync
- isync

Модель памяти

Ядро PowerPC 476FP и подсистема PowerPC поддерживают постоянную когерентность памяти, независимо от установки атрибута страницы M.

Кроме того, память PowerPC 476FP представляет собой слабо связанную модель и поэтому загрузка в общем случае может выполняться неупорядоченно, за исключением следующих случаев:

Обращение к операндам lwarx всегда происходит по-порядку.

Обращение к операндам с запрещенным кэшированием происходит по-порядку, включая случай, когда G = I = ‘1’.

Обращение к операндам внутри области когерентности, которая является строкой кэша L2, выполняется по-порядку.

В главе о модели памяти Power ISA Version 2.05, Book I содержится более подробная информация по следующим вопросам:

- Атомарность
- Модель кэша
- Атрибуты управление памятью
- Совместное использование памяти

Слабо связанная модель процессорного ядра PowerPC 476FP обладает следующими характеристиками:

Разрешает промахам загрузки обход или возврат в нарушение порядка, если они находятся не в той же строке кэша (области строки кэша L1).

Поддерживает порядок сохранения (хотя некоторые реализации могут разрушать неупорядоченное сохранение).

Разрешает направлять сохраняемые данные для последующей загрузки на том же процессоре.

Разрешает использование первого операнда, даже если строка ивалидирована слежением. Обычно самые новые данные отправляются L2. Данная модель памяти может приводить к следующим сценариям зависимости по данным:

```
CPU 0 CPU 1
  lwz  R3,X      st addrY
  sth  R3,Y      msync
  lwz  R31,Y     st addrX
```

В предыдущем примере кода CPU 0 мог промахнуться на X, но попасть на Y и получить новые данные X. Однако R31 загружается как новыми, так и старыми данными. Но если у CPU 0 происходит промах кэша на sth Y (сохранение полуслова или сохранение полубайта), в R31 присутствуют новые данные Y, и проблемы никакой нет.

При возникновении подобных проблем в приложениях обратитесь за дополнительной информацией в службу поддержки IBM PowerPC.

Следующие примеры демонстрируют спецификацию Power ISA и рекомендации по упорядочению операций сохранения. Эти примеры не затрагивают вопросы зависимости по данным, затронутые выше.

Примечание: msync используется для обеспечения упорядочения операций сохранения.

Пример 1: Упорядочение Тип #1 Совпадение границ операндов
CPU 0 CPU 1
lwz R3,X st addrY
sync msync
stw R3,Y st addrX
lwz R31,Y ..

Пример 2: Упорядочение Тип #1 Использование барьера памяти
CPU 0 CPU 1
lwz R3,X st addrY
sync msync
sth R3,Y st addrX
lwz R31,Y ..or
CPU 0 CPU 1
lwz R3,X st addrY
sync msync
lwz R31,Y st addrX

Пример 3: Упорядочение Тип #3, тестовый флаг для сохранения
CPU 0 CPU 1
lwz R3,X st addrY cmp X (test X) msync bne .. st addrX st R31,Z
В предыдущем примере CPU 0 st Z не является проблемой, поскольку Z изменяется только при изменении X.

Пример 4: Упорядочение Тип #3, Test Flag (См.пример 5: Практика сервера)
CPU 0 CPU 1
lwz R3,X st addrY
cmp X (test X) msync
bne loop A st addrx
sth R3,Y ..
lwz R31,Y
..
loop A

Пример 5: Практика сервера
Между следующими двумя загрузками (с инструкциями сравнения и ветвления или без них) необходима барьерная операция.

	CPU 0		CPU 1
loop1	lw X		st Y
	cmp		msync
	bne	loop1	st X

msync/mabar/isync
lw Y

Программная модель операций с плавающей запятой

Программная модель ядра PowerPC 476FP описывает функции и операции с точки зрения программиста.

В главе об операциях с плавающей запятой в Power ISA Version 2.05 определяется, что блок операций с плавающей запятой (FPU) реализует систему вычислений с плавающей запятой в соответствии со стандартом ANSI/IEEE Standard 754-1985, IEEE Standard for Binary Floating-Point Arithmetic (далее IEEE 754), однако для полного соответствия стандарту требуется программная поддержка. В IEEE 754 дано определение требуемым операциям (сложение, вычитание и так далее); термин "операция с плавающей запятой" используется для обозначения одной из этих требуемых операций или для операции, выполняемой одной из инструкций умножения-суммирования или вычисления обратной величины. В ядре PowerPC 476FP все операции с плавающей запятой соотвествуют стандарту IEEE.

Исключения операций с плавающей запятой

Каждое исключение операции с плавающей запятой и каждая категория исключения недопустимой операции связана с битом исключения в FPSCR. Следующие исключения операций с плавающей запятой обнаруживаются процессором; соответствующие поля FPSCR указаны для каждого исключения и каждой категории исключения недопустимой операции.

Исключение недопустимой операции (VX)

Таблица Б.33 - Категории исключения недопустимой операции

Категория	Поле FPSCR
SNaN	VXSNAN
Бесконечность - Бесконечность	VXISI
Бесконечность □□□ Бесконечность	VXIDI
Ноль □□□Ноль	VXZDZ
Бесконечность □□□Ноль	VXIMZ
Недопустимое сравнение	VXVC
Программный запрос	VXSOFT
Недопустимый квадратный корень	VXSQRT
Недопустимое целочисленное преобразование	VXCVI

Исключение деления на ноль (ZX)

Исключение переполнения (OX)

Исключение потери значимости (UX)

Исключение неточности

Каждое исключение операции с плавающей запятой имеет соответствующий бит разрешения в FPSCR.

Регистры операций с плавающий запятой

В этом разделе представлен обзор типов регистров, реализованных в ядре PowerPC 476FP. Подробное описание регистров операций с плавающей запятой приведено в разделах, посвященных функциям, с которыми они ассоциированы.

Определенные биты некоторых регистров являются зарезервированными и не обязательно реализованы. Для всех регистров с полями, отмеченными как зарезервированные, эти зарезервированные поля должны записываться как ‘0’, а считываться как неопределенные. Рекомендованная практика программирования - в регистр с зарезервированными полями изначально записывать ‘0’, а в последующем выполнять запись в регистр при помощи приема "чтение-модификация-запись": считать регистр, при помощи логических команд изменить требуемые поля, оставив зарезервированные поля неизменными, а затем выполнить запись в регистр.

Каждый регистр относится к определенному типу, в зависимости от инструкций, которые используются для чтения и записи регистров этого типа. Регистры процессора PowerPC 476FP определены в главе об операциях с плавающей запятой в Book-I Power ISA Version 2.05

Типы регистров

В процессоре PowerPC 476FP есть два типа регистров операций с плавающей запятой: регистры операций с плавающей запятой (FPR) и FPSCR. Каждый тип характеризуется инструкциями, которые используются для чтения и записи регистров этого типа. В следующих подразделах приводятся краткие описания каждого типа регистра, а также ассоциированные с ним инструкции.

Регистры операций с плавающей запятой (FPR0 - FPR31)

Процессор PowerPC 476FP предоставляет 32 FPR разрядностью 64 бита каждый. В каждом цикле файл FPR может считывать операнды для инструкции сохранения и арифметической инструкции, или записывать данные из инструкции загрузки и результат арифметической инструкции.



Биты	Наименование поля	Описание
0:63	Данные	Данные регистра операций с плавающей запятой.

FPRs имеют номера FPR0 - FPR31. Форматы инструкций операций с плавающей запятой имеют 5-битовое поле для указания регистров FPR, используемых в качестве операндов при выполнении соответствующих инструкций.

Каждый FPR содержит 64 бита, которые поддерживают формат двойной точности. Все инструкции, которые интерпретируют содержимое FPR как число с плавающей запятой, используют для этой интерпретации формат двойной точности. Несмотря на то, что архитектура регистров FPR 64-битовая, они содержат закодированные 66-разрядные данные плюс дополнительные 8 битов защиты на четность (всего 74 бита).

Вычислительные инструкции, а также инструкции пересылки и выбора, оперирующие данными в FPR, за исключением инструкций сравнения, помещают результат операции в FPR и опционально помещают информацию статуса в регистр условий (CR).

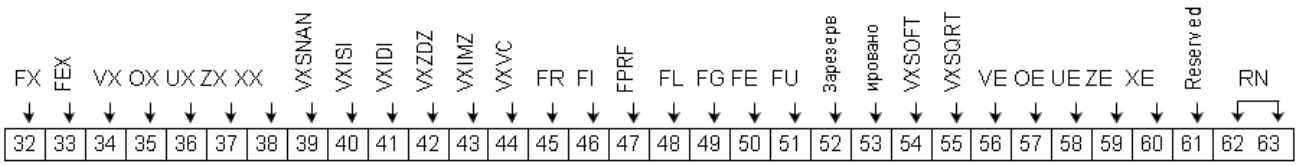
Имеются инструкции загрузки и сохранения двойной точности, которые пересылают 64 бита данных между памятью и регистрами FPR без преобразования. Инструкции загрузки одинарной точности пересылают и преобразуют числа с плавающей запятой в формате одинарной точности из памяти в то же значение двойной точности в FPR. Инструкции сохранения одинарной точности пересылают и преобразуют числа с плавающей запятой в формате двойной точности из FPR в то же значение одинарной точности в памяти.

Некоторые инструкции операций с плавающей запятой изменяют FPSCR и CR в явном виде. Некоторые из этих инструкций перемещают данные в FPR и из FPR, а также в FPSCR или из FPSCR в FPR.

Вычислительные инструкции и инструкции выбора получают значения из FPR в формате двойной точности. Для арифметических инструкций одинарной точности все входные величины должны быть представлены в формате одинарной точности; в противном случае результат помещается в FPR, а состояние бит статуса FPSCR не определено.

Регистр состояния и режима работы блока операций с плавающей запятой (FPSCR)

FPSCR управляет обработкой исключений операций с плавающей запятой и записывает статус результата операций с плавающей запятой. В FPSCR биты 0:31 зарезервированы.



Биты	Наименование поля	Описание
32	FX	Исключения операций с плавающей запятой. 0 Никакие биты исключения FPSCR не изменяются из 0 в 1. 1 Как минимум один бит исключения FPSCR изменяется из 0 в 1. Все инструкции операций с плавающей запятой, за исключением mtfsfi и mtfsf, явно устанавливают это поле в 1, если инструкция вызывает изменения любых битов исключения операций с плавающей запятой в FPSCR из 0 в 1. mcrfs, mtfsfi, mtfsf, mtfsb0 и mtfsb1 могут изменить эти биты явно.
33	FEX	Разрешение исключений операций с плавающей запятой. Логическое ИЛИ всех полей исключений операций с плавающей запятой, маскированых соответствующими полями разрешения. mcrfs, mtfsfi, mtfsf, mtfsb0 и mtfsb1 не могут изменить эти биты явно.
34	VX	Исключение недопустимой операции с плавающей запятой. Логическое ИЛИ всех полей исключений недопустимой операции. Инструкции mcrfs, mtfsfi, mtfsf, mtfsb0 и mtfsb1 не могут изменить эти биты явно.
35	OX	Исключение переполнения операций с плавающей запятой. 0 Исключение переполнения операций с плавающей запятой не произошло. 1 Исключение переполнения операций с плавающей запятой произошло.
36	UX	Исключение потери значимости операций с плавающей запятой. 0 Исключение потери значимости операции с плавающей запятой не произошло. 1 Исключение потери значимости операции с плавающей запятой произошло.
37	ZX	Исключение деления на ноль операции с плавающей запятой. 0 Исключение деления на ноль операции с плавающей запятой не произошло. 1 Исключение деления на ноль операции с плавающей запятой произошло.
38	XX	Исключение неточности операции с плавающей запятой. 0 Исключение неточности операции с плавающей запятой не произошло. 1 Исключение неточности операции с плавающей запятой произошло. Это поле является промежуточной версией FPSCR[FI]. Следующие правила описывают, как данная инструкция устанавливает это поле. Если инструкция влияет на FPSCR[FI], новое значение этого поля получается операцией логического ИЛИ старого значения этого поля с новым значением FPSCR[FI]. Если инструкция не влияет на FPSCR[FI], значение этого поля не меняется.
39	VXSNAN	Исключение недопустимой операции с плавающей запятой (SNaN). 0 Исключение недопустимой операции с плавающей запятой не произошло. 1 Исключение недопустимой операции с плавающей запятой произошло.
40	VXISI	Исключение недопустимой операции с плавающей запятой (□□– □). 0 Исключение недопустимой операции с плавающей запятой (VXISI) не произошло. 1 Исключение недопустимой операции с плавающей запятой (VXISI) произошло.
41	VXIDI	Исключение недопустимой операции с плавающей запятой (□□□□□□). 0 Исключение недопустимой операции с плавающей запятой (VXIDI) не произошло. 1 Исключение недопустимой операции с плавающей запятой (VXIDI) произошло.

Биты	Наименование поля	Описание
42	VXZDZ	Исключение недопустимой операции с плавающей запятой (0 □□0). 0 Исключение недопустимой операции с плавающей запятой (VXZDZ) не произошло. 1 Исключение недопустимой операции с плавающей запятой (VXZDZ) произошло.
43	VXIMZ	Исключение недопустимой операции с плавающей запятой (□□□□0). 0 Исключение недопустимой операции с плавающей запятой (VXIMZ) не произошло. 1 Исключение недопустимой операции с плавающей запятой (VXIMZ) произошло.
44	VXVC	Исключение недопустимой операции с плавающей запятой (недопустимое сравнение). 0 Исключение недопустимой операции с плавающей запятой (VXVC) не произошло. 1 Исключение недопустимой операции с плавающей запятой (VXVC) произошло.
45	FR	Округление дробной части числа с плавающей запятой. Последняя арифметическая инструкция или инструкция округления и преобразования либо выдает неточный результат округления, либо вызывает исключение недопустимого переполнения. См.Раздел 3.3.6 Округление на стр. 94. Этот бит не является вторым промежуточным битом округления.
46	FI	Неточность дробной части числа с плавающей запятой. Последняя арифметическая инструкция или инструкция округления и преобразования либо выдает неточный результат округления, либо вызывает исключение недопустимого переполнения. См.Раздел 3.3.6. Этот бит не является вторым промежуточным битом округления. См. определение FPSCR[XX] в части взаимоотношений между FPSCR[FI] и FPSCR[XX].
47	FPRF	Флаг результата операции с плавающей запятой (FPRF).
48	FL	Число с плавающей запятой меньше или отрицательное.
49	FG	Число с плавающей запятой больше или положительное.
50	FE	Число с плавающей запитой равно нулю.
51	FU	Число с плавающей запятой неупорядоченное или не является числом (NaN).
52	Зарезервировано.	Зарезервировано.
53	VXSOFT	Исключение недопустимой операции с плавающей запятой (запрос программы). 0 Исключение недопустимой операции с плавающей запятой (запрос программы) не произошло. 1 Исключение недопустимой операции с плавающей запятой (запрос программы) произошло.
54	VXSQRT	Исключение недопустимой операции с плавающей запятой (недопустимый квадратный корень). 0 Исключение недопустимой операции с плавающей запятой (недопустимый квадратный корень) не произошло. 1 Исключение недопустимой операции с плавающей запятой (недопустимый квадратный корень) произошло.
55	VXCVI	Исключение недопустимой операции с плавающей запятой (недопустимое целочисленное преобразование). 0 Исключение недопустимой операции с плавающей запятой (недопустимое целочисленное преобразование) не произошло. 1 Исключение недопустимой операции с плавающей запятой (недопустимое целочисленное преобразование) произошло.
56	VE	Разрешение исключения недопустимой операции с плавающей запятой разрешено. 0 Исключения недопустимой операции с плавающей запятой запрещены. 1 Исключения недопустимой операции с плавающей запятой разрешены.

Биты	Наименование поля	Описание
57	OE	Разрешение исключения переполнения операций с плавающей запятой. 0 Исключения переполнения операции с плавающей запятой запрещены. 1 Исключения переполнения операции с плавающей запятой разрешены.
58	UE	Разрешение исключения потери значимости операций с плавающей запятой. 0 Исключения потери значимости операции с плавающей запятой запрещены. 1 Исключения потери значимости операции с плавающей запятой разрешены.
59	ZE	Разрешение исключения деления на ноль операции с плавающей запятой. 0 Исключения деления на ноль операции с плавающей запятой запрещены. 1 Исключения деления на ноль операции с плавающей запятой разрешены.
60	XE	Разрешение исключения неточности операций с плавающей запятой. 0 Исключения исключения операции с плавающей запятой запрещены. 1 Исключения исключения операции с плавающей запятой разрешены.
61	Зарезервировано.	
62:63	RN	Управление округлением чисел с плавающей запятой. Округлить до ближайшего. Округлить до нуля. Округлить до +□□. Округлить до –□□ См. Округление на стр. 94.

Примечание - Установка FPSCR[NI] = ‘1’ предназначена для того, чтобы разрешить результатам быть приблизительными, а действия более предсказуемыми и менее зависимыми по данным, чем при FPSCR[NI] = ‘0’. Например, в режиме, отличном от IEEE, вместо денормализованного числа возвращается 0, и режим, отличный от IEEE, может возвращать большое число вместо бесконечности.

Форматы данных с плавающей запятой

Числа с плавающей запятой представляются в двух двоичных форматах фиксированной длины. Числа с одинарной точностью представлены 32-битным одинарным форматом. Числа с двойной точностью представлены 64-битным двойным форматом. Одинарный формат может использоваться для данных в памяти, но не может храниться в FPR. Двойной формат может использоваться для данных в памяти и для данных в PR. Когда число с плавающей запятой загружается из памяти при помощи одинарной инструкции загрузки, оно преобразуется в двойной формат и помещается в указанный FPR. И наоборот, когда число с плавающей запятой сохраняется в память из FPR при помощи одинарной инструкции сохранения, оно преобразуется в одинарный формат, прежде чем помещается в память. См. инструкции загрузки FP и инструкции сохранения FP главе об операциях с плавающей запятой в Book-I Power ISA Version 2.05.

Числа формата с плавающей запятой, состоят из трех полей.

Таблица Б.34 - Поля формата

Поле	Описание
S	Бит знака
EXP	Показатель степени + смещение
FRACTION	Дробная часть

Длина полей показателя степени и дробной части различается для одинарного и двойного формата.

Представление числа

Представление чисел в формате с плавающей запятой состоит из бита знака (S), смещенного показателя степени (EXP) и дробной части (FRACTION) мантиссы. Мантисса включает неявный ведущий бит, справа от которого располагается дробная часть. Этот неявный ведущий бит равен ‘1’ для нормализованных чисел и ‘0’ для денормализованных чисел и расположен в позиции единичного бита (то есть, это первый бит слева после двоичной запятой).

Значения величин в двух форматах чисел в плавающей запятой соответствуют параметрам, указанным в таблице Б.35.

Таблица Б.35 - Поля чисел с плавающей запятой согласно IEEE 754

Параметр	Одинарный	Двойной
Смещение показателя степени	+127	+1023
Максимальный показатель степени	+127	+1023
Минимальный показатель степени	-126	-1022
Ширана полей (биты)		
Знак	1	1
Показатель степени	8	11
Дробная часть	23	52
Мантисса	24	53

FPR поддерживают только двойной формат чисел с плавающей запятой.

Числовые и нечисловые значения, представляемые каждым из двух поддерживаемых форматов, являются приближениями действительных чисел и включают нормализованные числа, денормализованные числа и нулевые значения. Нечисловые значения - это бесконечности и неопределенности ((NaN). Бесконечности граничат с действительными числами, но сами не являются числами, и при операциях с ними не выполняются стандартные правила арифметики. С действительными числами их связывает только порядком. Тем не менее, возможно определить ограниченные операции с числами и бесконечностями. Относительное расположение всех определенных категорий на числовой оси показано на рисунке Б.4.

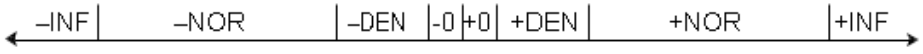


Рисунок Б.4 - Приближение к действительным числам

Неопределенности не связаны с числовыми значениями порядком или величиной, но являются кодами, которые используются для передачи диагностической информации, например, представления неинициализированных переменных.

Различные значения чисел с плавающей запятой, определенный в архитектуре, описываются в следующих разделах.

Двоичные числа с плавающей запятой

Машинное представление используется как приближенное значение действительных чисел. Поддерживаются три категории чисел: нормализованные числа, денормализованные числа и нулевые значения.

Нормализованные числа

Нормализованные числа (□NOR) имеют несмещенный показатель степени в диапазоне:

–126 до 127 в одинарном формате

-1022 до 127 в двойном формате

Это числа, у которых неявный единичный бит равен 1. Нормализованные числа интерпретируются следующим образом: NOR = (–1)s □□2E □□(1.дробная часть),

где s - знак, E - несмещенный показатель степени, а 1. дробная часть - мантисса, состоящая из ведущего бита (неявного бита) и дробной части.

Диапазон модуля (M) нормализованного числа с плавающей запятой приблизительно равен:

Одинарный формат:

1.2 □□10–38 □□M □□3.4 □□1038

Двойной формат:

2.2 □□10–308 □□M □□1.8 □□10308

Денормализованные числа

Денормализованные числа (□DEN) имеют смещенный показатель степени, равный нулю, и ненулевое значение дробной части. Это ненулевые числа с модулем, меньшим, чем у представляемых нормализованных чисел. Это числа, у которых неявный единичный бит равен 0. Денормализованные числа интерпретируются следующим образом:

DEN = (–1)s □□2Emin □□□(0.дробная часть),

где Emin минимальное представимое значение показателя степени (–126 для чисел одинарной точности, –1022 для чисел двойной точности).

Нулевые значения

Нулевые значения (□0) имеют смещенный показатель степени, равные нулю, и нулевое значение дробной части. Нулевые значения могут быт положительными или отрицательными. Знак нулевых значений игнорируется в операциях сравнения; при сравнении +0 считается равным –0).

Бесконечности

Бесконечности (□□) - это числа с максимальным значением смещенного показателя степени:

255 в одинарном формате

2047 в двойном формате

И нулевую дробную часть. Они используются для приближенного представления чисел, больших по модулю, чем максимальное нормализованное число.

Арифметика бесконечностей определяется как ограниченный случай арифметики действительных чисел, с ограниченным числом операций между числами и бесконечностями. Бесконечности и действительные числа могут быть связаны порядком в аффином смысле:

–□□< любое конечное число < +□

Арифметические действия с бесконечностями всегда точны и не вызывают исключений, кроме исключений, вызываемых недопустимыми операциями.

Неопределенности

Неопределенности (NaNs) - это значения с максимальным смещенным показателем степени и ненулевым значением дробной части. Бит знака игнорируется; то есть неопределенности не считаются ни положительными, ни отрицательными. Если старший бит дробной части равен ‘0’, то такая неопределенность называется сигнальной (SNaN).

Сигнальные NaN используются для сигнализации исключений - при появлении в качестве операндов вычислительных инструкций.

Несигнальные неопределенности используются для отображения результатов некоторых недопустимых операций, например, недопустимых арифметических операций с бесконечностями или неопределенностями, когда исключение недопустимой операции запрещено (FPSCR[VE] = ‘0’). Несигнальные неопределенности проходят через все инструкции операций с плавающей запятой, кроме fcmpo, frsp и fctiw. Несигнальные неопределенности не сигнализируют об исключениях, кроме операций упорядоченного сравнения и преобразования в целое число. Таким образом, конкретный код QNaN может сохраняться в результате последовательности операций с плавающей запятой и передавать диагностическую информацию, которая помогает обнаружить результаты недопустимых операций.

Когда QNaN является результатом операции с плавающей запятой вследствие того, что один из операндов является NaN, или QNaN была сгенерирована в результате исключения недопустимой операции, применяется следующее правило, чтобы обнаружить NaN со старшим битом дробной части, установленным в 1, и сохранить его как результат.

если FPR(FRA) - NaN то FPR(FRT) □□FPR(FRA)

иначе если FPR(FRB) - NaN

то если инструкция - frsp
то FPR(FRT) □□FPR(FRB)[0:34] || 290 иначе FPR(FRT) □□FPR(FRB)
иначе если FPR(FRC) - NaN то FPR(FRT) □□FPR(FRC)
иначе если сгенерирована QNaN
то FPR(FRT) □□сгенерированная QNaN

Если указанный FRA операнд - NaN, то в качестве результата сохраняется NaN. В противном случае, если указанный FRB операнд - NaN (если в инструкции указан операнд FRB), то в качестве результата сохраняется NaN с младшими 29 битами результата, установленными в ‘0’, если выполнялась инструкция frsp. В противном случае, если указанный FRC операнд - NaN, то в качестве результат сохраняется NaN. В противном случае, если была сгенерирована QNaN в результате запрещенного исключения недопустимой операции, то в качестве результата сохраняется QNaN. Если QNaN генерируется как результат операции, то сгенерированная QNaN имеет бит знака ‘0’, в поле показателя степени и в старшем бите дробной части - ‘1’, а в остальных битах дробной части - 0. Любая инструкция, которая генерирует QNaN как результат запрещенной недопустимой операции, должна генерировать такую QNaN (то есть, x‘7FF8 0000 0000 0000’). NaN двойной точности представимы в одинарном формате только в том случае, если младшие 29 бит дробной части NaN двойной точности равны нулю.

Знак результата

Следующие правила определяют знак результата арифметической операции, округления или преобразования, когда операция не вызывает исключение. Они применимы даже в тех случаях, когда операнды или результаты являются нулями или бесконечностями.

Знак результата операции сложения - это знак операнда с наибольшей абсолютной величиной. Знак результат операции вычитания x – y такой же, как знак результата операции сложения x + (–y).

Когда сумма двух операндов с противоположными знаками или разность двух операндов с одинаковыми знаками в точности равна нулю, знак результата положителен для всех режимов округления, за исключением округления до -бесконечности, когда результат считается отрицательным.

Знак результата операции умножения или деления - исключающее ИЛИ знаков операндов.

Знак результата инструкции frsqrte всегда положителен, за одним исключением - обратная величина квадратного корня из

-0 равна -бесконечности.

Знак результата операции frsp[.] or fctiw совпадает со знаком преобразуемого операнда.

Для инструкций умножения-сложения изложенные выше правила применяются сначала к операции умножения, а затем к операции сложения или вычитания (один из операндов операции сложения или вычитания является результатом операции умножения).

Обработка данных и точность

Инструкции предназначены для пересылки данных с плавающей запятой между регистрами FPR и памятью. Для данных двойного формата данные не изменяются при пересылке. Для данных одинарного формата преобразование из одинарного в двойной формат выполняется при загрузке из памяти в FPR. Преобразование формата из двойного в одинарный выполняется при сохранении данных из регистра FPR в память. Инструкции загрузки/сохранения не вызывают исключения операций с плавающей запятой.

Все вычислительные инструкции, инструкции пересылки и fsel используют двойной формат чисел с плавающей запятой. Числа с плавающей запятой одинарной точности получаются в результате выполнения следующих типов инструкций.

Загрузка одинарного числа с плавающей запятой.

Эта форма инструкции берет из памяти операнд с одинарной точностью, преобразует в формат с двойной точностью и загружает в FPR. Эти инструкции не вызывают исключений операций с плавающей запятой.

Округление до однократной точности.

Инструкция frsp округляет операнд двойной точности до одинарной точности, проверяя показатель степени на соответствие диапазону для одинарной точности и обрабатывает исключения согласно соответствующим битам разрешения, а затем помещает этот операнд в FPR как операнд с двойной точностью. Для результатов арифметических инструкций с одинарной точностью, операций загрузки с одинарной точностью и других случаев инструкции frsp эта операция не изменяет значение.

Примечание: Инструкция frsp разрешает преобразование числа двойной точности в число одинарной точности с соответствующей проверкой исключения и округлением. Эту инструкцию следует использовать для преобразования чисел с плавающей запятой двойной точности (получившихся в результате операций загрузки и арифметических операций с двойной точностью) в числа одинарной точности, прежде чем сохранить их в элементы памяти одинарного формата или использовать их в качестве операндов для арифметических инструкций одинарной точности. Числа, получающиеся в результате выполнения инструкций загрузки или арифметических инструкций одинарной точности уже имеют одинарный формат и могут напрямую помещаться в элементы памяти одинарного формата или использоваться напрямую в качестве операндов для арифметических инструкций одинарной точности без предварительного выполнения инструкции frsp перед операцией сохранения или арифметической операцией.

Арифметические инструкции одинарной точности.

Эта форма инструкции берет операнды из регистров FPR в двойном формате, выполняет операцию с промежуточным результатом так, словно этот промежуточный результат имеет бесконечную точность и неограниченный диапазон показателя степени, а затем приводит этот промежуточный результат к одинарному формату. Биты статуса в FPSCR устанавливаются так, чтобы отражать результат одинарной точности. Затем результат преобразуется в двойной формат и помещается в FPR. Результат находится в диапазоне, поддерживаемом одинарным форматом.

Входные величины должны быть представимыми в одинарном формате. Если это условие не соблюдено, результат помещается в FPR, а установка бит статуса в FPSCR не определена.

Сохранение одинарного числа с плавающей запятой.

Эта форма инструкции преобразует операнд двойной точности в одинарный формат и сохраняет этот операнд в памяти. Эти инструкции не вызывают исключений операций с плавающей запятой. (Предполагается, что сохраняемое значение является результатом выполнения инструкции одного из трех предыдущих типов.)

Когда результат инструкции загрузки числа с одинарной точностью, frsp или арифметической инструкции одинарной точности сохраняется в FPR, младшие 29 бит дробной части равны нулю.

Примечание: Число одинарной точности может быть использовано в арифметических операциях двойной точности. Обратное возможно только в том случае, если число можно представить в одинарном формате.

Округление

Округление применяется к операциям с числовыми операндами (операнды не являются бесконечностями или неопределенностями). Округление промежуточного результата таких операций может вызвать исключение переполнения, исключение потери значимости или исключение неточности. Следующее описание предполагает, что операции не вызывают исключений, и что их результат является числовым.

Арифметические инструкции, инструкции округления и преобразования имеют промежуточные результаты, которые могут рассматриваться как числа бесконечной точности с неограниченным диапазоном показателя степени. Такие промежуточные результаты при необходимости нормализуются или денормализуются, затем округляются до нужного формата.

Окончательный результат затем помещается в нужный FPR в двойном формате или целочисленном формате, в зависимости от инструкции.

Арифметические инструкции, инструкции округления и преобразования, округляющие промежуточные результаты, устанавливают FPSCR[FR, FI]. Если в процессе округления дробная часть инкрементировалась, то FPSCR[FR] = ‘1’; в противном случае FPSCR[FR] = ‘0’. Если округленный результат является неточным, то FPSCR[FI] = ‘1’; в противном случае FPSCR[FI] = ‘0’.

Инструкции оценки устанавливают FPSCR[FR, FI] в неопределенное значение. Остальные инструкции операций с плавающей запятой не изменяют FPSCR[FR, FI].

FPSCR[RN] определяет один из четырех программируемых режимов округления.

Пусть z - это промежуточный арифметический результат или операнд в операции преобразования. Если z может быть точно представлен в нужном формате, то результат во всех режимах округления равняется z , представленным в нужном формате. Если z невозможно точно представить в нужном формате, пусть z1 и z2 располагаются по обе стороны от z как следующее большее и следующее меньшее число, представимое в нужном формате. Тогда z1 или z2 можно использовать в качестве приближенного результата в нужном формате.

На рисунке Б.5 изображено соотношение z, z1 и z2 для данного случая. Следующие правила определяют четыре режима округления. LSb - это наименее значимый бит.

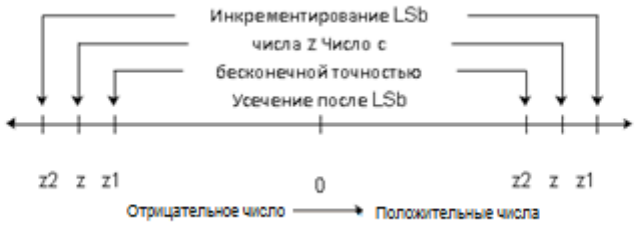


Рисунок Б.5 - Выбор z1 и z2

Таблица Б.36 - Режимы округления

FPSCR[RN]	Режим округления	Описание
00	Округлить до ближайшего.	Выбрать число, ближайшее к z, либо z1 либо z2. В случае равенства выбрать четное число (LSb равен 0).
01	Округлить до нуля.	Выбрать меньшее по модулю (z1 или z2).
10	Округлить до +бесконечности.	Выбрать z1.
11	Округлить до - бесконечности.	Выбрать z2.

Инструкции операций с плавающей запятой

Первичный код операции 63 используется для арифметических инструкций двойной точности и других инструкций, таких как инструкции манипуляции регистром состояния и режима работы блока операций с плавающей запятой. Первичный код операции 59 используется для арифметических инструкций одинарной точности.

Инструкции одинарной точности, для которых есть соответствующие инструкции двойной точности, имеют тот же формат и расширенный код операции, что соответствующие инструкции двойной точности.

Инструкции предназначены для выполнения арифметических операций, округления, преобразования, сравнения и других операций в регистрах операций с плавающей запятой, а также для пересылки данных между памятью и этими регистрами, и для явной манипуляции регистром FPSCR.

Эти инструкции подразделяются на две категории.

Вычислительные инструкции

К вычислительным относятся инструкции, которые выполняют сложение, вычитание, умножение, деление, извлечение квадратного корня, округление, преобразование, сравнение, а также сочетание этих операций. Эти инструкции выполняют операции с плавающей запятой. Они помещают информацию о состоянии в FPSCR. Эти инструкции описаны в Разделе 3.4.5, Арифметические инструкции с плавающей запятой на стр. 100, Разделе 3.4.6, Инструкции округления и преобразования чисел с плавающей запятой на стр. 101 и Разделе 3.4.7, Инструкции сравнения чисел с плавающей запятой на стр. 101.

Невычислительные инструкции

Невычислительные инструкции выполняют операции загрузки и сохранения, перемещают содержимое одного регистра операций с плавающей запятой в другой регистр, возможно, с переменой знака, явно манипулируют регистром FPSCR и выбирают число из одного из двух регистров операций с плавающей запятой на основе значения третьего регистра операций с плавающей запятой. Эти операции не считаются операциями с плавающей запятой. За исключением инструкций, которые явно манипулируют FPSCR, они не изменяют содержимое FPSCR. Эти инструкции описаны в Разделе 3.4.8, Инструкции регистра состояния и режима работы на стр. 102.

Число с плавающей запятой состоит из показателя степени со знаком и мантиссы со знаком. Значение, представляемое этим числом, является результатом вычисления 2exponent. Кодировка выполнена в формате данных, чтобы отражать конечные числовые значения, бесконечность и неопределенность (NaN). Операции, в которых участвуют бесконечности, дают результаты в соответствии со стандартными математическими правилами. Неопределенности не пмеют математической интерпретации, но их кодировка поддерживает изменяемое поле диагностической информации. Неопределенности могут использоваться для указания, например, неинициализированных переменных, они также могут возникать в результате некоторых недопустимых операций.

Один класс исключений, возникающих при выполнении инструкций операций с плавающей запятой, является уникальным: это исключения операций с плавающей запятой. Биты, установленные в FPSCR, указывают на исключения операций с плавающей запятой. Они могут вызывать программное прерывание разрешенного исключения приниматься, точно или неточно, если установлены соответствующие управляющие биты.

Категории инструкций

Инструкции операций с плавающей запятой могут быть разделены на две категории, вычислительные и не вычислительные. К вычислительным инструкциям относятся те, которые выполняют арифметические операции или преобразование операндов. Не вычислительные инструкции выполняют операции загрузки/сохранения, пересылки (с возможным изменением знака) или выбора данных. Кроме того, некоторые не вычислительные инструкции могут выполнять запись непосредственно в FPSCR. Все инструкции, выполняющиеся в конвейере загрузки/сохранения, относятся к не вычислительным, тогда как большинство инструкций, выполняющихся в арифметическом конвейере, являются вычислительными.

Все операнды с плавающей запятой хранятся внутри в формате двойной точности. Арифметические операции, обозначенные как одинарные, требуют, чтобы внутренние данные были представимы в одинарном формате (то есть, имели показатель степени между

-126 и 127 и мантиссу, точно представимую в 24 битах). Если данные невозможно представить таким образом, то результат сохраняется в FPR, а биты статуса, устанавливаемые в FPSCR и CR (соответствующие) не определены.

В целях совместимости, для уменьшения вероятности серьезных проблем в результате ошибки пользователя, а также для обеспечения возможности тестирования с применением случайных кодов, операции с одинарной точностью выполняются с операндами двойной точности. Для всех случаев, за исключением fdivs, операция выполняется так, как будто она

имеет двойную точность, а затем результат округляется до одинарной точности. Для fdivs выполняется необходимое (потенциально небольшое) количество итераций, чтобы получить результат с одинарной точностью, затем частное должным образом округляется.

Во всех случаях исключения результата (переполнение, потеря значимости и неточность) определяются и фиксируются в зависимости от результата, а не исходных операндов. По умолчанию (маскированное исключение) результаты такие же, как для инструкций одинарной точности. В случае исключений маскированного переполнения или потери значимости возвращаются 11 наименее значимых бит скорректированного истинного порядка.

Результаты всех операций одинарной точности округляются до одинарной точности. Эти результаты сохраняются в формате двойной точности, но ограничены диапазоном одинарной точности (показателя степени и дробной части). Все биты статуса устанавливаются в зависимости от результата одинарной точности.

Инструкции загрузки и сохранения

Набор инструкций процессора PowerPC 476FP включает инструкции загрузки из памяти в FPR и сохранения из FPR в память.

Данные, полученные от ядра PowerPC 476FP, могут иметь одинарную или двойную точность, в прямом или обратном порядке следования байтов. Кроме того, получаемые данные выровнены по словам. Данные в FPR должны иметь формат двойной точности и обратного порядка следования байтов.

Различают два основных формата инструкций загрузки: с одинарной точностью и с двойной точностью. Поскольку регистры FPR поддерживают только двойной формат чисел с плавающей запятой, то инструкции загрузки чисел с плавающей запятой одинарного формата преобразуют данные одинарного формата в двойной, прежде чем загрузить операнд в указанный FPR. Стадии преобразования и загрузки выполняются следующим образом.

Пусть WORD[0:31] - это операнд с плавающей запятой одинарной точности, полученный из памяти.

Нормализованный операнд

if WORD[1:8] > 0 and WORD[1:8] < 255 then FPR(FRT)[0:1] □□WORD[0:1]

FPR(FRT)[2] □□□WORD[1] FPR(FRT)[3] □□□WORD[1] FPR(FRT)[4] □□□WORD[1] FPR(FRT)[5:63] □□WORD[2:31] || 290

Денормализованный операнд

if WORD[1:8] = 0 and WORD[9:31] □□□0 then sign □□□WORD[0]

exp □□-126

frac[0:52] □□0b0 || WORD[9:31] || 290 normalize the operand

do while frac[0] = 0

frac □□frac[1:52] || 0b0 exp □□exp - 1

FPR(FRT)[0] □□sign FPR(FRT)[1:11] □□exp + 1023 FPR(FRT)[12:63] □□frac[1:52]

Zero / Infinity / NaN

if WORD[1:8] = 255 or WORD[1:31] = 0 then FPR(FRT)[0:1] □□WORD[0:1]

FPR(FRT)[2] □□WORD[1]

FPR(FRT)[3] □□WORD[1]

FPR(FRT)[4] □□WORD[1] FPR(FRT)[5:63] □□WORD[2:31] || 290

Для инструкций загрузки чисел с плавающей запятой двойной точности преобразование не требуется, поскольку данные из памяти копируются непосредственно в FPR.

Некоторые инструкции загрузки чисел с плавающей запятой изменяют GPR(RA) с исполнительным адресом. Для этих форм если RA □□0, то исполнительный адрес помещается в GPR(RA), и элемент хранения (байт, полуслово, слово или двойное слово), адресуемое EA, загружается в FPR(RT). Если RA = 0, эта форма инструкции недопустима.

Обращения к памяти инструкций загрузки чисел с плавающей запятой вызывают исключения сохранения, если программе не разрешено считывать ячейку памяти. Обращения к

памяти инструкций загрузки чисел с плавающей запятой вызывают исключения ошибки TLB при попытках программы обратиться к недоступным ячейкам памяти.

Примечание - RA и RB обозначают регистры GPR, а FRT обозначает FPR. Поддерживается, как прямой, так и обратный порядок следования байтов.

Таблица Б.37 - Инструкции загрузки чисел с плавающей запятой

Мнемоника	Операнды	Инструкция
lfd	FRT, D(RA)	Загрузка числа с плавающей запятой двойной точности.
lfdu	FRT, D(RA)	Загрузка числа с плавающей запятой двойной точности с корректировкой.
lfdux	FRT, RA, RB	Загрузка числа с плавающей запятой двойной точности с корректировкой, индексированная.
lfdx	FRT, RA, RB	Загрузка числа с плавающей запятой двойной точности индексированная.
lfs	FRT, D(RA)	Загрузка одинарного числа с плавающей запятой.
lfsu	FRT, D(RA)	Загрузка числа с плавающей запятой одинарной точности с корректировкой.
lfsux	FRT, RA, RB	Загрузка числа с плавающей запятой одинарной точности с корректировкой, индексированная.
lfsx	FRT, RA, RB	Загрузка числа с плавающей запятой одинарной точности индексированная.
lfiwax	FRT, RA, RB	Загрузка числа с плавающей запятой как целочисленного слова, алгебраически индексированная.

Инструкции сохранения чисел с плавающей запятой

Различают три основных формата инструкций сохранения: с одинарной точностью, с двойной точностью и целочисленные. Целочисленная форма предоставляется инструкцией stfiwx. Поскольку регистры FPR поддерживают только двойной формат чисел с плавающей запятой, то инструкции сохранения чисел с плавающей запятой одинарного формата преобразуют данные одинарного формата в двойной, прежде сохранить операнд в памяти. Стадии преобразования выполняются следующим образом.

Пусть WORD[0:31] - это слово, которое требуется записать в память.

Денормализация не требуется (включая ноль/бесконечность/неопределенность)

if FPR(FRS)[1:11] > 896 or FPR(FRS)[1:63] = 0 then WORD[0:1] □□FPR(FRS)[0:1]

WORD[2:31] □□FPR(FRS)[5:34]

Денормализация требуется

if 874 □□□FRS[1:11] □□□896 then sign □□□FPR(FRS)[0]

exp □□□□FPR(FRS)[1:11] – 1023

frac □□0b1 || FPR(FRS)[12:63] denormalize operand

do while exp < –126

frac □□0b0 || frac[0:62] exp □□exp + 1

WORD[0] □□sign WORD[1:8] □□0x00 WORD[9:31] □□frac[1:23]

else WORD □□undefined

Следует учитывать, что если величина, сохраняемая инструкцией сохранения чисел с плавающей запятой одинарной точности, по модулю больше, чем максимальное число, представимое в одинарном формате, то применим первый случай (денормализация не требуется). Результат, сохраненный в WORD, представляет собой структурированную величину, но не равную значению в регистре источнике. Результат загрузки числа с плавающей запятой одинарной точности из WORD при сравнении не будет равен содержимому первоначального регистра источника.

Для инструкций сохранения чисел с плавающей запятой двойной точности и для инструкций сохранения чисел двойной точности как целочисленных значений преобразование не требуется, поскольку данные из FPR копируются непосредственно в память.

Некоторые инструкции сохранения чисел с плавающей запятой изменяют GPR(RA) с исполнительным адресом. Для этих форм, если RA □□0, исполнительный адрес помещается в GPR(RA).

Обращения к памяти инструкций сохранения чисел с плавающей запятой вызывают прерывания сохранения данных, если программе не разрешено записывать в ячейку памяти. Обращения к памяти инструкций целочисленного сохранения чисел с плавающей запятой вызывают прерывания ошибки TLB при попытках программы обратиться к недоступным ячейкам памяти.

Примечание: RA и RB обозначают регистры GPR, а FRS обозначает FPR. Поддерживается, как прямой, так и обратный порядок следования байтов.

Таблица Б.38 - Инструкции сохранения чисел с плавающей запятой

Мнемоника	Операнды	Инструкция
stfd	FRS, D(RA)	Сохранение числа с плавающей запятой двойной точности.
stfdu	FRS, D(RA)	Сохранение числа с плавающей запятой двойной точности с корректировкой.
stfdux	FRS, RA, RB	Сохранение числа с плавающей запятой двойной точности с корректировкой, индексированная.
stfdx	FRS, RA, RB	Сохранение числа с плавающей запятой двойной точности индексированная.
stfiwx	FRS, RA, RB	Сохранение числа с плавающей запятой как целочисленного слова, алгебраически индексированная.
stfs	FRS, D(RA)	Сохранение числа с плавающей запятой одинарной точности.
stfsu	FRS, D(RA)	Сохранение числа с плавающей запятой одинарной точности с корректировкой.
stfsux	FRS, RA, RB	Сохранение числа с плавающей запятой одинарной точности с корректировкой, индексированная.
stfsx	FRS, RA, RB	Сохранение числа с плавающей запятой одинарной точности индексированная.

Инструкции пересылки чисел с плавающей запятой

Эти инструкции копируют данные из одного регистра операций с плавающей запятой в другой, изменяя бит знака (бит 0), как описано в спецификации Power Instruction Set Architecture (ISA) Version 2.05 для fneg, fabs и fnabs. Эти инструкции обращаются с неопределенностями как с любыми другими числами (например, знаковый бит NaN может изменяться fneg, fabs и fnabs). Эти инструкции не изменяют FSPCR.

Таблица Б.39 - Инструкции пересылки чисел с плавающей запятой

Мнемоника	Операнды	Инструкция
fabs[.]	FRT, FRB	Модуль числа с плавающей запятой
fmr[.]	FRT, FRB	Пересылка числа с плавающей запятой в регистр
fnabs[.]	FRT, FRB	Модуль отрицательного числа с плавающей запятой
fneg[.]	FRT, FRB	Инвертирование числа с плавающей запятой

Арифметические инструкции для чисел с плавающей запятой

Эти инструкции выполняют элементарные арифметические операции.

Таблица Б.40 - Элементарные арифметические операции над числами с плавающей запятой

Мнемоника	Операнды	Инструкция
fadd[.]	FRT, FRA, FRB	Сложение чисел с плавающей запятой.
fadds[.]	FRT, FRA, FRB	Сложение чисел с плавающей запятой одинарного формата.
fcfid[.]	FRT, FRB	Преобразование целочисленного двойного слова в число с плавающей запятой.
fcpsgn[.]	FRT, FRB	Копирование знака числа с плавающей запятой.
fctid[.]	FRT, FRB	Преобразование числа с плавающей запятой.в целочисленное двойное слово.
fctiw[.]	FRT, FRB	Преобразование числа с плавающей запятой.в целочисленное слово.

Продолжение таблицы Б.40

Мнемоника	Операнды	Инструкция
fctiwz[.]	FRT, FRB	Преобразование числа с плавающей запятой в целое с округлением до нуля.
fdiv[.]	FRT, FRA, FRB	Деление чисел с плавающей запятой.
fdivs[.]	FRT, FRA, FRB	Деление чисел с плавающей запятой одинарного формата.
fmul[.]	FRT, FRA, FRB	Умножение чисел с плавающей запятой.
fmuls[.]	FRT, FRA, FRB	Умножение чисел с плавающей запятой одинарного формата.
fre[.]	FRT, FRB	Вычисление обратной величины числа с плавающей запятой.
fres[.]	FRT, FRB	Вычисление обратной величины числа с плавающей запятой одинарного формата.
frsqrte[.]	FRT, FRB	Вычисление обратной величины квадратного корня числа с плавающей запятой.
frsqrtes[.]	FRT, FRB	Вычисление обратной величины квадратного корня числа с плавающей запятой одинарного формата.
fsqrt[.]	FRT, FRB	Квадратный корень числа с плавающей запятой.
fsqrts[.]	FRT, FRB	Квадратный корень числа с плавающей запятой одинарного формата.
fsub[.]	FRT, FRA, FRB	Вычитание чисел с плавающей запятой.
fsubs[.]	FRT, FRA, FRB	Вычитание чисел с плавающей запятой одинарного формата.

Инструкции умножения-сложения чисел с плавающей запятой

Эти операции сочетают операции умножения и сложения без промежуточной операции округления. Дробная часть промежуточного результата имеет разрядность 106 бит (бит L, FRACTION), и все 106 бит участвуют в сложении или вычитании, которое входит в состав инструкции.

Биты FPSCR устанавливаются следующим образом:

Биты исключения переполнения, потери значимости и неточности, биты FR и FI, а также поле FPRF устанавливаются в зависимости от конечного результата операции, а не на основе результата умножения.

Биты исключения недопустимой операции устанавливаются так, как будто умножение и сложение выполнялись двумя отдельными инструкциями (fmul[s], а затем fadd[s] или fsub[s]). То есть, умножение бесконечности на 0 или любого числа на SNaN, и сложение с SNaN приведет к установке соответствующих битов исключений.

Таблица Б.41 - Инструкции умножения-сложения чисел с плавающей запятой

Мнемоника	Операнды	Инструкция
fmadd[.]	FRT, FRA, FRB, FRC	Умножение-сложение чисел с плавающей запятой
fmadds[.]	FRT, FRA, FRB, FRC	Умножение-сложение чисел с плавающей запятой одинарного формата.
fmsub[.]	FRT, FRA, FRB, FRC	Умножение-вычитание чисел с плавающей запятой.
fmsubs[.]	FRT, FRA, FRB, FRC	Умножение-вычитание чисел с плавающей запятой одинарного формата.
fnmadd[.]	FRT, FRA, FRB, FRC	Умножение-сложение отрицательных чисел с плавающей запятой
fnmadds[.]	FRT, FRA, FRB, FRC	Умножение-сложение отрицательных чисел с плавающей запятой одинарного формата.
fnmsub[.]	FRT, FRA, FRB, FRC	Умножение-вычитание отрицательных чисел с плавающей запятой
fnmsubs[.]	FRT, FRA, FRB, FRC	Умножение-вычитание отрицательных чисел с плавающей запятой одинарного формата.

Инструкции округления и преобразования чисел с плавающей запятой

Инструкции округления чисел с плавающей запятой приведены в таблице Б.42.

Таблица Б.42 - LINK \Инструкции округления и преобразования чисел с плавающей запятой

Мнемоника	Операнд	Инструкция
frim[.]	FRT, FRB	Округления числа с плавающей запятой до меньшего целого числа.
frin[.]	FRT, FRB	Округления числа с плавающей запятой до ближайшего целого числа.
frip[.]	FRT, FRB	Округления числа с плавающей запятой до большего целого числа.
friz[.]	FRT, FRB	Округления числа с плавающей запятой до нуля.
frsp[.]	FRT, FRB	Округления числа с плавающей запятой до одинарного формата.

Инструкции сравнения чисел с плавающей запятой

Инструкции сравнения чисел с плавающей запятой сравнивают содержимое двух регистров операций с плавающей запятой. При сравнении игнорируется знак нуля (+0 считается равным –0). Результат сравнения может быть упорядоченным и неупорядоченным.

Операция сравнения устанавливает один бит в указанном поле CR в ‘1’ а остальные три бита в ‘0’. FPSCR[FPCC] устанавливается аналогичным образом.

Установка поля CR, а также FPSCR[FPCC] приведена в Табл. 3-11 на стр. 102.

Таблица Б.43 - Установки сравнения

Бит	Наименование	Описание
0	FL	(FRA) < (FRB)
1	FG	(FRA) > (FRB)
2	FE	(FRA) = (FRB)
3	FU	(FRA) ? (FRB) (неупорядоченный)

Таблица Б.44 - Инструкции сравнения и выбора чисел с плавающей запятой

Мнемоника	Операнды	Инструкция
fcmpo	BF, FRA, FRB	Сравнение чисел с плавающей запятой, упорядоченное.
fcmpu	BF, FRA, FRB	Сравнение чисел с плавающей запятой, неупорядоченное.
fsel[.]	FRT, FRA, FRB, FRC	Выбор числа с плавающей запятой.

Инструкции регистра состояния и режима работы блока операций с плавающей запятой

Каждая инструкция регистра состояния и режима работы блока операций с плавающей запятой синхронизирует эффекты всех инструкций с плавающей запятой, выполненных данным процессором. Инструкция регистра состояния и режима работы блока операций с плавающей запятой обеспечивает завершение всех инструкций с плавающей запятой, инициированных данным процессором, прежде чем инициируется инструкция регистра состояния и режима работы блока операций с плавающей запятой, а также то, что ни одна последующая инструкция с плавающей запятой не будет инициализирована данным процессором до завершения инструкций регистра состояния и режима работы блока операций с плавающей запятой. В частности:

Все исключения, вызванные ранее инициированными инструкциями, записываются в FPSCR до инициации инструкции регистра состояния и режима работы блока операций с плавающей запятой.

Все обращения программных прерываний разрешенного типа исключения, вызванные ранее инициированными инструкциями, происходят до инициации инструкции регистра состояния и режима работы блока операций с плавающей запятой.

Ни одна инструкция с плавающей запятой, которая использует установку любых бит FPSCR или изменяет их, не инициируется до завершения инструкции регистра состояния и режима работы блока операций с плавающей запятой.

Это не относится к инструкциям загрузки и сохранения чисел с плавающей запятой.

В таблице Б.45 приведены инструкции регистра состояния и режима работы блока операций с плавающей запятой.

Таблица Б.45 - Инструкции регистра состояния и режима работы блока операций с плавающей запятой

Мнемоника	Операнды	Инструкция
mcrfs		Пересылка в регистр условий из FPSCR
mffs[.]	FRT	Пересылка из FPSCR
mtfsb0[.]	BT	Пересылка в FPSCR бита 0.
mtfsb1[.]	BT	Пересылка в FPSCR бита 1.
mtfsf[.]	FLM, FRB	Пересылка в поля FPSCR
mtfsfi[.]	BF, U	Непосредственная пересылка в поля FPSCR

Модуль управления памятью

Модуль управления памятью (MMU) PowerPC 476FP обеспечивает управление кэшем, защиту доступа и трансляцию адреса. MMU содержит общий буфер ассоциативной трансляции (UTLB), управляющую логику и регистры, поддерживающие UTLB. MMU взаимодействует с исполнительным блоком (EU), блоком кэша инструкций (ICU), блоком кэша данных (DCU) и следящим интерфейсом TLB. Интерфейс EU дает возможность выполнять инструкции операций буфера ассоциативной трансляции (TLB): tlbre, tlbwe, tlbsx, tlbivax (дополнительную информацию об этих инструкциях см.Раздел 4.8, Особенности программирования на стр. 127). Интерфейс блока инструкций (IU) обеспечивает трансляцию селекторных бит области трансляции (TS) и DSIZ для запроса поиска от DCU, ICU или отслеживания TLB. Интерфейс ICU генерирует запрос поиска к UTLB при промахе буфера ассоциативной трансляции инструкций (ITLB). Аналогичным образом, интерфейс DCU генерирует запрос поиска к UTLB при промахе буфера ассоциативной трансляции данных (DTLB).

MMU - это программно управляемый модуль с аппаратной поддержкой, доступной для замещения элементов. Программа записывает элементы в UTLB, так чтобы их можно было считsdatь, используя хэш-функцию, описанную в Разделе 4.3.2, Индексная хэш-адресация UTLB, на стр. 106. Режим работы MMU типа Freescale не поддерживается.

В таблице Б.46 приведены характеристики MMU процессора PowerPC 476FP.

Таблица Б.46 – Характеристики MMU процессора PowerPC 476FP

Функция	PowerPC 476FP
Размер PowerPC 476FP	1024 записи
Архитектура массива памяти	SRAM1P.
Ассоциативность UTLB	4-входовой ассоциативный кэш, чтение четырех записей одновременно
Время доступа к кэшу данных MMU	Переменное, от 6 до 30 циклов, в зависимости от числа одновременных запросов, используемых хэшей.
Поддержка размеров страницы	4 KB, 16 KB, 64 KB, 1 MB, 16 MB, 256 MB, 1 GB.
Дескрипроты страниц	WIMGE, U[0:3], IL1I, IL1D. Определения этих полей см. в Табл.4-5 на стр. 110.
Поле ID трансляции (TID)	16 бит
Расширенный действительный номер страницы (ERPN)	10 бит
Механизм поиска UTLB	Поиск до семи хэшей, оптимизированный по размену страницы, в порядке, установленном в регистре SPR супервизора или пользователя.

Трансляция адресов

Описание трансляции адресов MMU приведено на рисунке Б.6. 49-битовый виртуальный адрес (VA) формируется вставкой 32-битового исполнительного адреса (EA) с 1-битовым адресным пространством (AS) и 16-битового ID процесса (PID). При помощи AS, PID, и EA из UTLB получаются 10-битовый расширенный действительный номер страницы (ERPN) и 20-

битовый действительный номер страницы (RPN). Они соединяются друг с другом при помощи 12-битового смещения, образуя 42-битовый реальный адрес.

Страница размером 4 K требует 20 бит EA (и RPN), тогда как страница размером 1G требует только 2 бит EA (и RPN), как показано на рисунке Б.6.

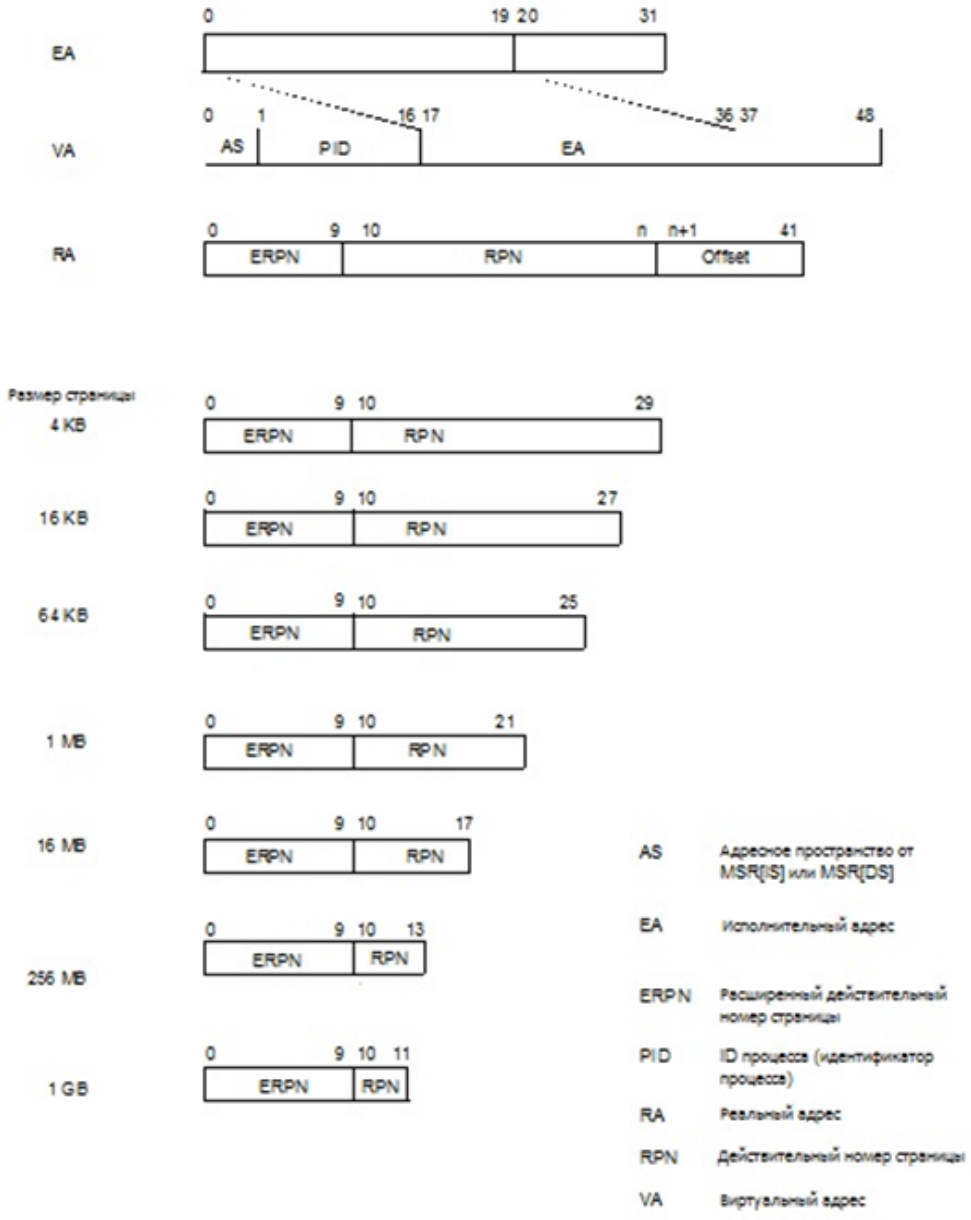


Рисунок Б.6 - Распределение адресов для каждого размера страницы

Реализация MMU

На рисунке Б.7 изображена структура MMU. Это 4-канальный ассоциативный кэш памяти с хэш-адресацией MMU выполняет запрос арбитража и состоит из массива тегов UTLB, логики сравнения и массива данных UTLB. Реализация PowerPC 476FP поддерживает UTLB на 1024 записи.

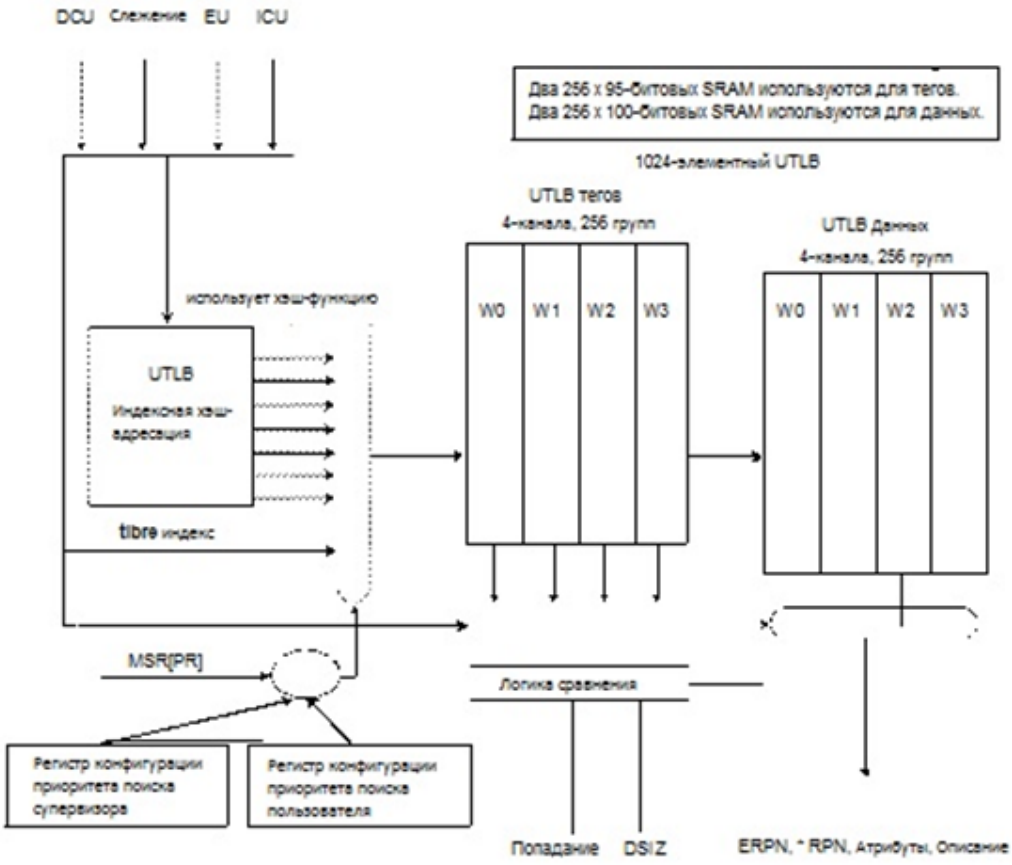


Рисунок Б.7 - Блок-схема MMU

*Максимальный размер страницы ITLB и DTLB составляет 256 MB. Страница размером 1 GB должна быть разбита на области по 256 MB.

DCU Блок кэша данных
DSIZ Декодированный размер страницы
DTLB Теневой буфер ассоциативной трансляции данных
ERPN Расширенный действительный номер страницы
EU Исполнительный блок
ICU Блок кэша инструкций
ITLB Теневой буфер ассоциативной трансляции инструкций
MSR[PR] Регистр управления процессором, задачный режим
RPN Действительный номер страницы
SRAMСтатическая память с произвольной выборкой
UTLB Общий буфер ассоциативной трансляции

Буфер ассоциативной трансляции

Общий буфер ассоциативной трансляции (UTLB) - это аппаратный ресурс, который управляет атрибутами трансляции, защиты и сохранения. Один общий 4-канальный ассоциативный кэш TLB на 1024 записи используется для обращения и к инструкциям, и к данным. Кроме того, ядро PowerPC 476FP реализует два отдельных, меньших по размеру теневых массива TLB, один для выборки инструкций, другой для обращения к данным. Эти теневые TLB повышают производительность, уменьшая задержку трансляции адреса, а также уменьшая конфликт главного общего TLB между выборкой инструкции и сохранением данных.

Управление записями TLB осуществляется программой. Системное ПО определяет стратегию замещения записи TLB или аппаратное замещение, а также использование табличной информации страницы. Запись TLB содержит всю информацию, необходимую для

идентификации страницы, указания трансляции адреса, управления разрешениями доступа и присваивания атрибутов хранения.

Запись в TLB производится копированием информации из GPR и поля MMUCR[STID] при помощи последовательности из трех инструкций tlbwe. Чтение из TLB производится копированием информации в GPR и поля MMUCR[STID] при помощи последовательности из трех инструкций tlbre. Программа также может выполнять поиск конкретных TLB с помощью инструкции tlbsx[.].. Ядро PowerPC 476FP также позволяет программе инвалидировать каждую запись TLB при помощи инструкции tlbivax или tlbwe.

Метод доступа к UTLB, ассоциативная трансляция, информация об атрибутах и управлении доступом для ERPN, а также RPN описаны в следующих разделах.

Индексная хэш-адресация UTLB

Для повышения эффективности использования UTLB и для улучшения распределения реализована хэш-функция, которая индексирует массивы UTLB.

Хэш-функция исключающего ИЛИ (XOR) используется, когда запись ищется при помощи инструкций индексированного поиска TLB (tlbsx), или когда она инвалидирована локальными или удаленными операциями tlbivax. Хэш обходится при операциях чтения элемента TLB (tlbre), поскольку инструкция tlbre указывает и канал, и индексированный адрес.

Например, страница 4 KB с 16-битовым PID равным x‘00D5’ и 20-битовым исполнительным адресом (EA) равным x‘E9B6C’ должна быть размещена в UTLB по 8-битовому индексированному адресу x‘E0’. Это вычисляется следующим образом:

Бит 7 индексного адреса UTLB = PID[15] XOR EA[19] XOR EA[7]. Бит 6 индексного адреса UTLB = PID[14] XOR EA[18] XOR EA[6]. Бит 5 индексного адреса UTLB = PID[13] XOR EA[17] XOR EA[5]. Бит 4 индексного адреса UTLB = PID[12] XOR EA[16] XOR EA[4].

Бит 3 индексного адреса UTLB = PID[11] XOR EA[15] XOR EA[11] XOR EA[3]. Бит 2 индексного адреса UTLB = PID[10] XOR EA[14] XOR EA[10] XOR EA[2]. Бит 1 индексного адреса UTLB = PID[9] XOR EA[13] XOR EA[9] XOR EA[1].

Бит 0 индексного адреса UTLB = PID[8] XOR EA[12] XOR EA[8] XOR EA[0].

Каждый из индексов UTLB поддерживает до четырех записей, обозначенных как каналы 0 - 3.

Хэширование страниц разного размера с разными, не перекрывающимися исполнительными адресами, может поступать на один и тот же индексный адрес UTLB. Например, хэш 4 KB используемый для EA равного x‘000F0’, транслируется в индексный адрес x‘F0’. Хэш 1 MB используемый для не перекрывающегося EA равного x‘F0000’, транслируется в индексный адрес x‘F0’. Это не проблема, потому что канал 0 может быть использован для страницы 4 KB, а канал 1 для страницы 1 MB. Тем не менее, программа должна принимать это во внимание при установке записей в UTLB, чтобы избежать случайного перекрытия записей.

В таблице Б.47 приведены используемые биты PID и адреса EA.

Таблица Б.47 - Установка функции хэширования генерации адреса UTLB

PID	UTLB Индекс-ный бит	Биты PID	Биты исполнительного адреса для каждого размера страницы						
			4 KB	16 KB	64 KB	1 MB	16 MB	256 MB	1 GB
PID □□‘0’	7	31	19, —, 7	17, —, 7	15, 7	11, —	7	—	—
	6	30	18, —, 6	16, —, 6	14, 6	10, —	6	—	—
	5	29	17, —, 5	15, —, 5	13, 5	9, —	5	—	—
	4	28	16, —, 4	14, —, 4	12, 4	8, —	4	—	—
	3	27	15, 11, 3	13, —, 3	11, 3	7, 3	3	3	—
	2	26	14, 10, 2	12, —, 2	10, 2	6, 2	2	2	—
	1	25	13, 9, 1	11, 9, 1	9, 1	5, 1	1	1	1
	0	24	12, 8, 0	10, 8, 0	8, 0	4, 0	0	0	0

Продолжение таблицы Б.47

PID	UTLB Индек-сный бит	Биты PID	Биты исполнительного адреса для каждого размера страницы						
			4 KB	16 KB	64 KB	1 MB	16 MB	256 MB	1 GB
PID □□‘0’	7	—	19, —, 7	17, —, 7	15, 7	11, —	7	—	—
	6	—	18, —, 6	16, —, 6	14, 6	10, —	6	—	—
	5	—	17, —, 5	15, —, 5	13, 5	9, —	5	—	—
	4	—	16, —, 4	14, —, 4	12, 4	8, —	4	—	—
	3	—	15, 11, 3	13, —, 3	11, 3	7, 3	3	3	—
	2	—	14, 10, 2	12, —, 2	10, 2	6, 2	2	2	—
	1	—	13, 9, 1	11, 9, 1	9, 1	5, 1	1	1	1
	0	—	12, 8, 0	10, 8, 0	8, 0	4, 0	0	0	0
Примечание - Прочерк (-) указывает, что EA не используется для вычисления хэша для данного размера страницы									

Инициализация одиночной записи UTLB

Аппаратная часть при сбросе инициализирует одну запись UTLB. Эта запись устанавливается для доступа к запрещенной и защищенной области привилегированного кэша размером 4 GB (старшие адреса пространства 4 GB). Эта запись соответствует вектору сброса процессора по адресу x‘FFFF FFFC’ и имеет следующие характеристики:

Индексный адрес	x‘F0’ (4 KB хэш эквивалентен PID[0:15] = x‘0000’ and EA[0:19] = x‘FFFFF’)
Канал 3	
EPN	x‘FFFFF’
TS	‘0’
DSIZ	‘000000’ (страница 4 KB)
RPN	x‘FFFFF’

WIMG ‘0101’
IL1I, IL1D ‘11’
UX, UW, UR ‘000’
SX, SW, SR ‘101’
ERPN Из значений конфигурации кристалла, зависящих от реализации.
U0-U3 Из значений конфигурации кристалла, зависящих от реализации.
E Из значений конфигурации кристалла, зависящих от реализации.

Инициализация этой одной записи UTLB дублирует функцию пересылки этих же значений в CU и DCU во время спроса, инициализируя вектор сброса в ITLB и DTLB. Однако запись в ITLB и DTLB может быть инвалидирована отслеживанием msync, isync, rfi или CSI (инструкцией переключения контекста) прежде чем программа произведет запись в UTLB. Запись вектора сброса в UTLB обеспечивает, что промах ITLB и DTLB в данном случае находит соответствующую запись.

Массив тегов

Хэшированный индексный адрес (не хэшируется только индексный адрес tlbre) передается в массив тегов. Массив тегов содержит информацию, необходимую для определения, совпадает ли с допустимым элементом запрос UTLB от EU, ICU, DCU или интерфейса слежения. Массив тегов состоит из двух SRAM1P размером 256 □□94. Каждая запись тега имеет размер 44 бита, плюс три бита четности. Таким образом, в одном SRAM1P хранится канал 0 и канал 1 тегов, а во втором SRAM1P - канал 2 и канал 3. Индексный адрес передается в оба SRAM1P в одном такте, так что сравнение всех четырех каналов может быть выполнено одновременно в следующем такте. Информация, хранимая в массиве тегов, приведена в Табл. 4-3. Нечетность сохраняется, когда CCR1[MMUTPEI] = ‘0’. Четность сохраняется, когда CCR1[MMUTPEI] = ‘1’.

Таблица Б.48 - Описание поля тегов UTLB

WS	Биты	Наименование	Размер	Описание
0	0:19	EPN	20	Исполнительный номер страницы. EPN и DSIZ определяют страницу (размер и начальный адрес), которую представляет эта запись UTLB. Неиспользованные - из-за большего, чем минимальный, DSIZ - биты этого поля игнорируются.
0	20	EPNPar	1	Бит четности, покрывающий EPN.
0	21	Valid	1	Если установлен, остальные поля описывают допустимую запись UTLB.
0	22	TS	1	Пространство трансляции Эта запись совпадает только если бит пространства трансляции совпадает с битом запроса.
0	23:28	DSIZ	6	Это поле описывает размер страницы для данной записи. Поддерживаются страницы с размером 4 KB (минимальный размер), 16 KB, 64 KB, 1 MB, 16 MB, 256 MB и 1 GB. Запись Размер страницы ‘000000’ 4 KB ‘000001’ 16 KB ‘000011’ 64 KB ‘000111’ 1 MB ‘001111’ 16 MB ‘011111’ 256 MB ‘111111’ 1 GB Примечание: Это декодирование используется таким образом, чтобы логика могла использовать закрепленные биты для компараторов условного разрешения на соответствующих битах адреса.
0	29	DSIZPar	1	Бит паритета, которые покрывает поля Valid, TS и DSIZ.
0	30:45	TID	16	ID Трансляции (MMUCR[0:15]). Это поле описывает ID процесса, для которого эта запись допустима. Если TID = ‘0’, он считается совпадающим с PID. Если TID != ‘0’, то поле TID должно совпадать с PID, чтобы поиск вернулся с положительным результатом.
0	46	TIDPar	1	Бит паритета, покрывающий TID.

Сравнение

Совпадение виртуального адреса с записью TLB найдено, когда теги Valid, TS, EPN и TID имеют следующие значения:

Valid == 1.

TS == запрашиваемое адресное пространство (AS).

EPN == запрашиваемый EA, где количество сравниваемых бит зависет от тега DSIZ.

• TID != 0.

TID == запрашиваемый идентификатор процесса (PID).

В таблице Б.49 приведены количество сравниваемых битов EPN и EA в зависимости от размера страницы.

Таблица Б.49 - Сравнение EPN и EA

Размер страницы	DSIZ	Сравнение
4 KB	‘000000’	EPN[0:19] == EA[0:19]
16 KB	‘000001’	EPN[0:19] == EA[0:17]
64 KB	‘000011’	EPN[0:19] == EA[0:15]
1 MB	‘000111’	EPN[0:19] == EA[0:11]
16 MB	‘001111’	EPN[0:19] == EA[0:7]
256 MB	‘011111’	EPN[0:19] == EA[0:3]
1 GB	‘111111’	EPN[0:19] == EA[0:1]

Максимальный размер страницы ITLB и DTLB составляет 256 MB. Поэтому страница размером 1 GB должна быть преобразована в области по 256 MB. Это делается автоматически аппаратной частью.

Массив данных

Массив данных содержит информацию о трансляции адресов, управлении памятью и биты разрешения для записи. Массив данных также состоит из двух SRAM1P размером 256 □□100. Каждая запись данных имеет размер 47 бита, плюс 3 бита четности. Таким образом, в одном SRAM1P хранится канал 0 и канал 1 данных, а во втором SRAM1P - канал 2 и канал 3. Индексный адрес находится в регистре-защелке и передается в оба SRAM1P данных в одном такте, так чтобы эти данные можно было извлечь из регистра защелки в запрашивающий блок, если результаты сравнения тегов совпадают.

Информация, хранимая в массиве данных, приведена в таблице Б.50. Нечетность сохраняется, когда CCR1[MMUDPEI] = ‘0’. Четность сохраняется, когда CCR1[MMUDPEI] = ‘1’.

Таблица Б.50 - Описание поля данных UTLB

WS	Биты	Наименование	Размер	Описание
1	0:19	RPN	20	Действительный номер страницы. Это поле содержит биты действительного адреса, которые замещают EPN адреса поиска. Для страниц 4 KB используется весь RPN. Для страниц большего размера наименее значимые биты RPN не используются. Тем не менее, неиспользованные биты должны быть установлены в 0 - в противном случае возникнуть неопределенные результаты.
1	20	RPNPar	1	Бит паритета, покрывающий RPN.
1	21:30	ERPN	10	Расширенный действительный номер страницы Это поле содержит 10 наиболее значимых битов исполнительного адреса. Они всегда используются и вставляются в RPN.
1	31	ERPNPar	1	Бит паритета, покрывающий ERPN.
2	32:36	WIMGE	5	Это поле описывает тип страницы для данной записи в кэше уровня L2 W Сквозная запись (L1 всегда выполняет сквозную запись). I Кэш запрещен. M Когерентная страница (все страницы L1 когерентны). G Защищенный доступ. E Порядок следования битов (это описывает страницу L1).
2	37	IL1I	1	Запрет инструкции L1. Если установлен, то эта страница считается запрещенной к кэшированию для кэша L1, независимо от бита, используемого для L2.
2	38	IL1D	1	Запрет данных L1. Если установлен, то эта страница считается запрещенной к кэшированию для кэша L1, независимо от бита, используемого для L2.
2	39:42	U	4	Биты, определенные пользователем. Это поле может использоваться на системном уровне с любой целью.
2	43	UX	1	Пользовательский бит разрешения выполнения.
2	44	UW	1	Пользовательский бит разрешения записи.
2	45	UR	1	Пользовательский бит разрешения чтения.
2	46	SX	1	Бит разрешения выполнения супервизора.
2	47	SW	1	Бит разрешения записи супервизора.
2	48	SR	1	Бит разрешения чтения супервизора.
2	49	StorPermPar	1	Бит четности, который покрывает атрибуты памяти и разрешения.

Записи с принудительно установленными аппаратной частью I = 1 = IL1I = IL1D

Если программа устанавливает I = 1 инструкцией tlbwe (WS = 2), то аппаратная часть MMU принудительно устанавливает IL1I = IL1D = 1.

Запись в элементы UTLB

Программа помещает записи в UTLB при помощи инструкции tlbwe. Операционная система должна понимать подробности хэширования размеров страниц, чтобы выбрать ID процессов и исполнительные адреса, добиваясь наилучшего использования UTLB. Каждая инструкция tlbwe предоставляет размер страницы, PID и EA. В tlbwe также может указываться один из четырех каналов, чтобы позволить аппаратной части выбирать записываемый канал.

Поскольку процессор имеет 32-битовую архитектуру, а записи тегов и данных UTLB требуют 91 бита, то для записи нового элемента UTLB необходима последовательность из трех инструкций lbwe (WS = 0, WS = 1 и WS = 2). При выполнении первой инструкции tlbwe с WS = 0, в регистр-защелку MMUCR помещаются значения V, индекса tlb и канала tlb. Поля индекса и канала tlb используются в двух последующих инструкциях tlbwe с WS = 1 и WS = 2 для выбора элемента tlb. Кроме того, бит V для запоминания значения (LVALID), когда WS = 2.

Так MMU поддерживает атоманрые записи UTLB.

Примечание: WS - это поле, обозначающее, какое слово элемента пересылается (то есть, WS = 0 указывает на слово 0 TLB, и так далее).

Фиксированные элементы UTLB

MMU позволяет программе определять до шести фиксированных элементов UTLB. Эти элементы обычно используются для страниц, содержащих ядро операционной системы или программу обработки прерываний. Фиксированные элементы автоматически пропускаются при аппаратном выборе. Фиксированные элементы также защищены от локальных и удаленных инструкций tlbivax. Фиксированные элементы могут перезаписываться программно при помощи инструкции lbwe. Местоположение фиксированного элемента в регистрах фиксированных элементов MMU (MMUBE0 or MMUBE1) указывается инструкцией tlbwe в RA[5:7]. После записи индексный адрес фиксированной инструкции может быть получен из MMUBE0 или MMUBE1 инструкцией mfspr. Фиксированный элемент поддерживается только каналом 0, и поэтому программа должна следить, чтобы фиксированные элементы не помещались по одному и тому же индексному адресу UTLB.

Аппаратная поддержка выбора канала

Для уменьшения нагрузки на программу при определении местоположения записей и для того, чтобы автоматически избегать перезаписи фиксированных элементов, аппаратная часть осуществляет поддержку в выборе канала записи для следующей tlbwe. Каждый индексный адрес UTLB имеет соответствующий 2-битовый счетчик. Инструкция lbwe с WS = 0 указывает, что для выбора канала используется значение счетчика, когда RA[0] = ‘0’. Содержимое счетчика инкрементируеся каждый раз при tlbwe с WS = 2, когда у соответствующая tlbwe с WS = 0 имеет V = 1. Счетчик сбрасывается в значение канала, когда этот канал получает lbwe с WS = 0, V = 0. Таким образом, следующий элемент записывается по этому индексу в канал, который только что "освободился" инструкцией tlbwe, WS = 0, V = 0. Аналогичным образом, счетчик сбрасывается в значение канала, когда этот канал инвалидируется локальной или удаленной инструкцией tlbivax. Счетчик автоматически пропускает фиксированный элемент. Например, если счетчик указывает на канал 3, а канал 0 содержит фиксированный элемент, то после следующей инструкции tlbwe, WS = 2 счетчик принимает значение 1.

Поиск элементов UTLB

Поиск элементов UTLB осуществляется ICU в ответ на промах TLB со стороны инструкций, DCU ответ на промах TLB со стороны данных, и EU в ответ на инструкцию tlbsx.

Яядро PowerPC 476FP реализует два теневых массива TLB, один для выборки инструкций, другой для обращения к данным. Эти массивы дублируют значение подмножества элементов в главном массиве, UTLB (UTLB в контексте этого обсуждения). Назначение этих теневых массивов TLB состоит в том, чтобы уменьшить задержку операции трансляции адреса и избежать конкуренции за UTLB между выборкой инструкций и обращением к данным.

Оба теневых массива TLB (ITLB и DTLB) содержат восемь элементов. Обращение к теневым массивам TLB не вызывает задержки, и выполнение инструкции продолжается конвейерным методом, при условии, что запрашиваемый адрес найден в теневом TLB. Если запрашиваемый адрес не найден в теневом TLB, выборка инструкции или обращение к данным автоматически останавливаются, и адрес ищется в UTLB. Если адрес найден в UTLB, задержка, связанная с промахом в теневом массиве, составляет пять циклов - при отсутствии конкуренции. Если промах имеет место также и в UTLB, то сообщается об исключении промаха TLB инструкции или данных.

Замещение элементов в теневых массивах TLB управляется аппаратно и происходит циклически. При промахе теневого TLB, который приводит к попаданию UTLB, аппаратная часть выталкивает старейший элемент теневого массива TLB замещает его новой трансляцией.

Аппаратная часть также инвалидирует все элементы в обоих теневых массивах TLB при любой синхронизации контекста. Операции синхронизации контекста:

- Любое прерывание (включая проверку процессора)
- Выполнение isync
- Выполнение rfi, rfci или rfmci
- Выполнение sc

Следует учитывать, что существуют другие операции изменения контекста, которые не вызывают автоматической синхронизации контекста в аппаратной части. Например, выполнение инструкции tlbwe изменяет содержимое UTLB, но не вызывает синхронизации контекста, и поэтому не инвалидирует и не изменяет элементы теневых TLB. Для того, чтобы изменения элементов UTLB (или других связанных с адресом ресурсов, таких как PID) отражались в теневых TLB, программа должна проследить, чтобы операция изменения контекста происходила до того, как предпринимается любая попытка использовать адрес, ассоциированный с измененными элементами UTLB (как со старым, так и с новым содержимым этих элементов). Инвалидируя теневые массивы TLB, операция синхронизации контекста принуждает программу обновить элементы теневой TLB измененной информацией из UTLB при обращении к каждой странице памяти.

Поиск при промахах TLB инструкций и промахах TLB данных

Когда происходит промах TLB инструкций или TLB данных, ICU или DCU выставляет EA с запросом. Бит AS устанавливается ICU или IU (для DCU). EA с зафиксированным содержимым регистра PID в ICU или DCU, хэшируется для получения индексного адреса UTLB на основе определяемого размером страниц порядка хэширования в регистре конфигурации приоритета поиска супервизора (SSPCR) или в регистре конфигурации приоритета поиска пользователя (USPCR). Поскольку для определения совпадающего элемента в массиве тегов требуется четыре цикла, последующее хэширование размера страниц, указанное в SSPCR или USPCR, используется для направления индексных адресов в массив тегов.

Элементы, помещенные в UTLB с TID[0:15] = x‘0000’, считаются глобальными страницами. Они совпадают, независимо от значения PID запроса, PID[0:15] = x‘0000’ или нет. Тем не менее, поскольку часть PID используется при хэшировании, глобальная страница может существовать в UTLB, даже несмотря на то, что индекс TLB не совпадает с хэшем EA-PID. Эта ситуация разрешается двойным прохождением порядка поиска, когда запрашиваемый PID[0:15] не равен x‘0000’: один раз с использованием значения x‘0000’ вместо действительного PID, а второй раз с использованием действительного значения PID[0:15].

Поиск записей UTLB выполняется EU в ответ на инструкцию tlbsx. EU выставляет исполнительный адрес (EA) с запросом. EA вместе с полем установки ID трансляции (STID), установленном в MMUCR, хэшируется для получения индексного адреса UTLB на основе определяемого размером страниц порядка хэширования в SSPCR. При обнаружении совпадающего элемента канал и индексный адрес возвращаются EU. Программа может сравнить индексный адрес, который вернулся в регистры SPR MMUBE0 и MMUBE1, чтобы определить, не является ли элемент найденный элемент TLB фиксированным. Затем программа может использовать инструкцию tlbre для чтения содержимого элемента UTLB.

Чтение элементов UTLB

Элемент UTLB может считываться EU при помощи инструкции tlbre. Канал и индексный адрес, указанные в инструкции tlbre, должны быть сначала получены инструкцией tlbsx. Для чтения информации элемента, хранящейся в массивах тегов (WS = 0) и данных (WS = 1 и WS = 2), требуются три инструкции tlbre.

Инвалидация элементов UTLB

Элемент UTLB может быть инвалидирован инструкцией tlbwe с WS = 0 и V = 0.

Элемент UTLB также может быть инвалидирован локальной или удаленной (слежение) инструкцией tlbivax. Локальная инструкция tlbivax получается как запрос EU с соответствующим исполнительным адресом (EA). EA вместе со STID, установленном в регистре SPR, MMUCR, хэшируется для получения индексного адреса UTLB на основе определяемого размером страниц порядка хэширования в ISPCR. Если совпадающий элемент найден в UTLB, в соответствующий бит V тега записывается в ‘0’, если элемент не фиксированный. Если элемент tlb фиксированный, он не инвалидируется.

Удаленная инструкция tlbivax получается как запрос слежения. MMU удерживает сигнал SnpAvail активным до того, как обнаруживается второй активный запрос слежения. Таким образом, одновременно могут обслуживаться два запроса слежения. Каждые запрос слежения выставляется с соответствующими полями AS, PID и EA. EA вместе с полями PID хэшируется для получения индексного адреса UTLB на основе определяемого размером страниц порядка хэширования в ISPCR. Если совпадающий элемент найден в UTLB, в соответствующий бит V тега записывается ‘0’, при условии, что элемент не фиксированный. Если элемент tlb фиксированный, он не инвалидируется. Если индексный адрес совпадающего элемента эквивалентен значению запомненного в регистре-защелка индексного адреса (LINDEX), то бит LVALID очищается - если элемент не является фиксированным. Это позволяет частично записанному элементу инвалидироваться при слежении.

Управление доступом

Когда совпадающий элемент TLB идентифицирован, а адрес транслирован, механизм управления доступом определяет, имеет ли программа доступ (выполнения, чтения, записи, чтения и записи) к странице, на которую указывает адрес.

Доступ выполнения

Бит UX или SX TLB управляет доступом выполнения страницы памяти, в зависимости от режима работы процессора (пользователя или супервизора).

Режим пользователя (MSR[PR] = ‘1’)

Инструкции могут выбираться из страницы памяти и выполняться, если в режиме пользователя бит UX управления доступом для этой страницы равен ‘1’. Если бит UX управления доступом равен ‘0’, инструкции с этой страницы не будут выбираться и не будут помещаться в любой кэш в результате запроса выборки этой страницы в режиме пользователя.

Более того, если модель последовательного выполнения в режиме пользователя вызывает инструкцию со страницы, которая не разрешена для выполнения в режиме пользователя (то есть, UX = ‘0’, когда MSR[PR] = ‘1’), воспринимается прерывание памяти инструкций типа

исключения управления доступом выполнения (дополнительную информацию см. в Разделе 7, Прерывания процессора и исключения на стр. 167).

Режим супервизора (MSR[PR] = ‘0’)

Инструкции могут выбираться из страницы памяти и выполняться, если в режиме супервизора бит SX управления доступом для этой страницы равен ‘1’. Если бит SX управления доступом равен ‘0’, инструкции с этой страницы не будут выбираться и не будут помещаться в любой кэш в результате запроса выборки этой страницы в режиме супервизора.

Более того, если модель последовательного выполнения в режиме супервизора вызывает инструкцию со страницы, которая не разрешена для выполнения в режиме супервизора (то есть, SX = ‘0’, когда MSR[PR] = ‘0’), то воспринимается прерывание памяти инструкций типа исключения управления доступом выполнения (дополнительную информацию см. в Разделе 7, Прерывания процессора и исключения на стр. 167).

Доступ записи

Бит UW или SW TLB управляет доступом записи к странице, в зависимости от режима работы процессора (пользователя или супервизора).

Режим пользователя (MSR[PR] = ‘1’)

Операции сохранения (включая инструкции управления кэшем dcbz idcbtst) разрешены для страницы памяти в режиме пользователя, если бит UW управления доступом для этой страницы равен ‘1’. Попытка в режиме пользователя выполнить операцию сохранения на страницу, для которой бит UW управления доступом равен ‘0’, вызывает исключение управления доступом записи. В случае инструкции stswx с длиной строки 0 прерывание не принимается, и операция не выполняется. Для всех остальных операций сохранения выполнение инструкции подавляется, и принимается прерывание сохранения данных.

Несмотря на то, что инструкция управления кэшем dcbi относится к классу инструкций сохранения, ее выполнение является привилегированным и поэтому не вызывает прерывания сохранения данных при попытке выполнить ее в режиме пользователя (вместо этого произойдет программное прерывание исключения типа привилегированной ниструкции).

Режим супервизора (MSR[PR] = ‘0’)

Операции сохранения (включая инструкции управления кэшем dcbz, dcbtst и dcbtstls) разрешены для страницы памяти в режиме супервизора, если бит SW управления доступом для этой страницы равен ‘1’. Попытка в режиме супервизора выполнить операцию сохранения на страницу, для которой бит SW управления доступом равен ‘0’, вызывает исключение управления доступом записи. В случае инструкции stswx с длиной строки 0 прерывание не принимается, и операция не выполняется. Для всех остальных операций сохранения выполнение инструкции подавляется, и принимается прерывание сохранения данных.

Управление чтением

Бит UR или SR TLB управляет доступом чтения к странице, в зависимости от режима работы процессора (пользователя или супервизора).

Режим пользователя (MSR[PR] = ‘1’)

Операции загрузки (включая инструкции управления кэшем dcbst, dcbf, dcbt, icbi и icbt) разрешены для чтения страницы памяти в режиме пользователя, если бит UR управления доступом для этой страницы равен ‘1’. Попытка в режиме пользователя выполнить операцию загрузки на страницу, для которой бит UR управления доступом равен ‘0’, вызывает исключение управления доступом чтения. Для инструкций загрузки (за исключением lswx с длиной строки 0) или инструкций dcbst, dcbf или icbi, выполнение инструкции подавляется, и принимается прерывание сохранения данных. В случае инструкции lswx с длиной строки 0, а также dcbt или icbt, прерывание не принимается, и операция не выполняется.

Режим супервизора (MSR[PR] = ‘0’)

Операции загрузки (включая инструкции управления кэшем dcbst, dcbf, dcbt, dcbi, dcbtls, dcblc, icbi, icbt, и icblc) разрешены для чтения страницы памяти в режиме супервизора, если бит

SR управления доступом для этой страницы равен ‘1’. Попытка в режиме супервизора выполнить операцию загрузки на страницу, для которой бит SR управления доступом равен ‘0’, вызывает исключение управления доступом чтения. Для инструкций загрузки (за исключением lswx с длиной строки 0) или инструкций dcbst, dcbf, dcbi или icbi, выполнение инструкции подавляется, и принимается прерывание сохранения данных. В случае инструкции lswx с длиной строки 0, а также dcbt или icbt, прерывание не принимается, и операция не выполняется.

Управление доступом в инструкциях управления кэшем

В данном разделе описывается, как механизм управления влияет на каждую из инструкций управления кэшем.

dcbz	Эта инструкция считается инструкцией сохранения с точки зрения контроля доступа, поскольку она изменяет данные в блоке кэша. Поэтому она может использовать прерывания сохранения данных типа исключения управления доступа записи.
dcbi	Эта инструкция считается инструкцией загрузки с точки зрения контроля доступа, поскольку она может изменять значение местоположения памяти инвалидацией текущей копии местоположения кэша данных, таким образом восстанавливая значение местоположения до предыдущего, которое хранится в памяти. Поэтому она может использовать прерывания сохранения данных типа исключения управления доступа записи.
dcba	Эта инструкция считается фиктивной при любых обстоятельствах и поэтому не может использовать любую форму прерывания сохранения данных.
icbi	Эта инструкция считается инструкцией загрузки с точки зрения контроля доступа. Поэтому она может использовать прерывания сохранения данных типа исключения управления доступа чтения. Эта инструкция может вызывать прерывание сохранения данных (но не прерывание сохранения инструкции) даже несмотря на то, что выполнится операция на кэше инструкций. Прерывания сохранения инструкции ассоциированы с исключениями, которые возникают при выборке инструкций, тогда как прерывания сохранения данных ассоциируются с исключениями, которые возникают при выполнении инструкции обращения к памяти или управления кэшем.
dcbt и icbt	Эти инструкции считается инструкциями загрузки с точки зрения управления доступом. В этом качестве они могут вызывать исключения управления доступом чтения. Однако поскольку эти инструкции действуют просто как подсказки, которые указывают блок кэша, к которому вероятнее всего обратится процессор в ближайшем будущем, такие исключения не приводят к прерыванию сохранения данных. Вместо этого при возникновении исключения управления доступом чтения инструкция считается фиктивной.
dcbtst	Эта инструкция считается инструкцией сохранение с точки зрения контроля доступа. В этом качестве она может вызывать исключения управления доступом сохранения. Однако поскольку эта инструкция действуют просто как подсказка, которая указывают блок кэша, к которому вероятнее всего обратится процессор в ближайшем будущем, такие исключения не приводят к прерыванию сохранения данных. Вместо этого при возникновении исключения управления доступом чтения инструкция считается фиктивной.
dcbf и dcbst	Эти инструкции считается инструкциями загрузки с точки зрения управления доступом. Поэтому они могут использовать прерывания сохранения данных типа исключения управления доступа чтения. Очистка или сохранение грязной строки из кэша не считается сохранением, поскольку предшествующая операция сохранения уже изменила строку кэша, и инструкция dcbf или dcbst просто перемещает в память результаты предыдущей операции сохранения.

dci и ici Эти инструкции не генерируют адрес. Кроме того, механизм управления доступом не влияет на эти инструкции. Это привилегированные инструкции, и при выполнении в режиме супервизора они инвалидируют весь соответствующий кэш.

Атрибуты памяти

Каждый элемент TLB определяет несколько атрибутов памяти для каждой страница памяти, с которой он ассоциирован. Атрибуты памяти влияют на то, каким образом будет выполняться доступ к данной странице памяти. Атрибуты памяти (и соответствующие поля элементов TLB):

Сквозная запись (W)

Запрет кэширования (I)

Требование когерентности памяти (M)

Защита (G)

Порядок следования байтов (E)

Определяемые пользователем (U0, U1, U2, U3)

Поддерживаются все сочетания этих атрибутов, за исключением тех, которые одновременно определяют в области сквозную запись и запрет кэширования.

Сквозная запись (W)

Кэш данных процессора PowerPC 476FP игнорирует атрибут сквозной записи. Данные для всех операций сохранения записываются в память, а не только в кэш данных. Если в кэше данных также существует указанная строка (то есть, имеет место попадание операции сохранения), то данные также записываются в кэш данных. Исключение выравнивания происходит в том случае, если инструкция dcbz обращается к странице памяти с атрибутом сквозной записи или запрета кэширования. Исключение сохранения данных происходит в том случае, если инструкции lwarx, stwcx обращаются к странице памяти с атрибутом сквозной записи или запрета кэширования.

Запрет кэширования (I)

Если страница памяти обозначена как запрещенная для кэширования (I = 1), то операции загрузки, сохранения и выборки инструкция обращаются к памяти, а не к соответствующему кэшу. Если I = 0 то страница считается кэшируемой, и операции могут выполняться в кэше.

Исключение выравнивания происходит в том случае, если инструкция dcbz обращается к странице памяти с атрибутом сквозной записи или запрета кэширования. Исключение сохранения данных происходит в том случае, если инструкции lwarx, stwcx обращаются к странице памяти с атрибутом сквозной записи или запрета кэширования. Если адрес обращения операции загрузки, сохранения dcbz или выборки access to caching inhibited storage to be in the respective cache, это ошибка программирования; результаты такого обращения являются неопределенными. Если местоположение других инструкций управления кэшем находятся в кэше, когда устанавливается атрибут запрета кэширования, это не является ошибкой программирования. Поведение этих инструкций определено как для страниц с I = 0, так и для страниц с I = 1. Дополнительную информацию об обращении к памяти с запретом кэширования см. в Разделе 5.

Принудительная аппаратная установка IL1I и IL1D

Поля запрета L1 инструкции (IL1I) и поле запрета L1 данных (IL1D) указывают, что страница будет считаться запрещенной для кэширования в кэше L1, независимо от бита I, используемого для атрибута кэша L2.

Таблица Б.51 - Управление доступом в инструкциях управления кэшем

Инструкция	Примечания
dcbf, dcbst, icbi	Генерирует прерывание нарушения разрешения защиты чтения, если для страницы UR = ‘0’.
dcbi	Генерирует прерывание нарушения разрешения защиты чтения, если для страницы UR = ‘0’. Если MSR[PR] = ‘1’, возникает исключения ошибки привилегированной или недопустимой инструкции.
dcbt, icbt	Эти инструкции являются фиктивными, если UR = ‘0’.
dcbtst, dcbz	Генерирует прерывание нарушения разрешения защиты записи, если для страницы UW = ‘0’.
dcba	Эта инструкция является фиктивной.

Требование когерентности памяти (M)

Атрибут требования когерентности памяти определен архитектурой, чтобы поддержать когерентность кэша и памяти в многопроцессорных системах с общей памятью. Если элемент TLB создан с M = 1, то любое обращение к странице памяти, ассоциированной с этим элементом TLB, указывается - при помощи соответствующего интерфейсного сигнала атрибута пересылки -

как с требованием когерентности памяти, однако установка этого атрибута не оказывает влияния на операции внутри процессора PowerPC 476FP.

Защита (G)

Атрибут защиты памяти предназначен для управления спекулятивным доступом для некорректных областей памяти. Память считается корректной, если соответствующая действительная память существует и работоспособна, и если влияние одиночного обращения к ней неотличимо от влияния нескольких идентичных обращений. В этом случае данные и инструкции могут в произвольном порядке выбираться из корректной памяти, не вызывая нежелательных побочных эффектов.

В общем случае некорректную память следует маркировать как защищенную. Поскольку такая память может представлять собой управляющий регистр или устройство ввода/вывода, или включать несуществующие местоположения, то неупорядоченное обращение к этой памяти может привести к тому, что устройства ввода вывода будут выполнять непредусмотренные операции, или вызывать исключение проверки процессора. Например, когда входной буфер последовательного устройства ввода/вывода отображен в памяти, то неупорядоченный или спекулятивный доступ к этому местоположению может привести к потере данных из входного буфера, если выполнение инструкции прервано, а потом возобновлено.

Обращение к защищенной области памяти выполняется только в том случае, если обращение вызвано инструкцией, которая заведомо требуется моделью последовательного выполнения, или если обращение является загрузкой, а область памяти уже находится в кэше данных. Когда обращение к защищенной памяти инициировано, то если кэширование памяти также запрещено, то доступны только конкретные указанные байты в соответствии с размером операнда данного типа инструкции. Обращение к памяти, помеченной как кэшируемая, допускает обращение к целому блоку кэша, либо в самом кэше, либо в памяти. Во избежание непредусмотренных результатов для поддержки коррекной модели памяти, память должна быть защищенной, а ее кэширование запрещено.

Защита памяти не влияет на выборку инструкций. Архитекрура не запрещает выборку инструкций из защищенной памяти, но системное ПО обычно должно предотвращать такую выборку, маркируя все защищенные страницы памяти как неисполняемые (UX/SX = 0). Тогда при попытке выборки инструкции с такой страницы обращения к памяти не произойдет, и результатом будет прерывание выборки инструкции из памяти с типом исключения управления доступом выполнения - при попытке выполнения инструкции из любого адреса внутри страницы.

Порядок следования байтов (E)

Атрибут порядка следования байтов в памяти (E) управляет порядком выполнения операций загрузки, сохранения и выборки. Порядок следования байтов указывает на порядок, в котором отдельные байты многобайтового скалярного операнда организованы в памяти. Операнды на странице памяти с E = 0 имеют обратный порядок следования байтов, когда наиболее значимый байт размещается по самому младшему адресу памяти. Операнды на странице памяти с E = 1 имеют прямой порядок следования байтов, когда наименее значимый байт размещается по самому младшему адресу памяти.

Определяемые пользователем (U0 - U3)

Ядро PowerPC 476FP предоставляет четыре определяемых пользователем атрибута памяти (U0 - U3), которые могут использоваться для управления поведением памяти, которое будет зависеть от системы. По умолчанию эти атрибуты памяти не оказывают влияния на работу ядра PowerPC 476FP, хотя все обращения к памяти указывают подсистеме памяти значения U0 - U3 при помощи соответствующих интерфейсных сигналов атрибутов пересылки. Затем конкретная система может воспользоваться этими атрибутами для управления поведением на системном уровне.

Поддерживаемые сочетания атрибутов памяти

Режимы работы памяти, когда W = 1 и I = 1 (то есть, память с разрешенной сквозной записью и запрещенным кэшированием), не поддерживаются. Для всех поддерживаемых сочетаний атрибутов памяти W и I, атрибуты памяти G, E и U0 - U3 могут использоваться в любых сочетаниях.

Перекрытие

Для нескольких страниц, распределенных на один и тот же исполнительный адрес, применяются следующие правила:

Если несколько страниц существуют на одном процессоре, то:

Биты I (I, IL1I, IL1D) должны совпадать с соответствующими битам на всех страницах (см. примечание ниже). Биты W не должны совпадать на всех страницах.

Биты M не должны совпадать на всех страницах. В таком случае поддерживать когерентность данных должна программа.

Если несколько страниц существуют на нескольких процессорах, то: Биты I (I, IL1I, IL1D) не должны совпадать на всех страницах. Бит W должен совпадать на всех страницах. (Требование Book E).

Биты M не должны совпадать на всех страницах. В таком случае поддерживать когерентность данных должна программа.

Примечание: Для нескольких страниц, которые существуют на одном процессоре и распределены на один и тот же действительный адрес, биты I (I, IL1I, IL1D) не должны совпадать при следующих условиях, которые должна обеспечить программа:

Для тех страниц с нулевым битом I страница должна быть маркирована как защищенная и неисполняемая, чтобы предотвратить спекулятивный доступ.

Для адресов с разными атрибутами кэшируемости программа должна обеспечить, чтобы только страницы с одинаковыми атрибутами обращались к перекрывающимся действительным адресам. Или программа может соответственно распределять кэш между разными кэшируемыми обращениями, чтобы гарантировать, что обращение к любому I = 1 не находится в соответствующем кэше. Когда бит I равен единице, данные не должны быть в кэше любого уровня.

Например, рассмотрим кэшируемую страницу 64 KB и некэшируемую страницу 4 KB (наименьший размер страницы для ядра PowerPC 476FP) которые обе распределены на один и тот же действительный адрес (например, страница 4 KB распределена на последние 4 KB действительных адресов, на которые распределена страница 64 KB). В этом случае страница размером 64 KB маркируется как защищенная и кэшируемая. Кроме того, программа должна обеспечить, что при работе в 64 KB странице обращения к последним 4 KB адресов не выполняются.

Регистры MMU

В таблице Б.52 приведены регистры специального назначения (SPR) MMU, доступные для программы. В остальной части раздела эти регистры рассматриваются подробно.

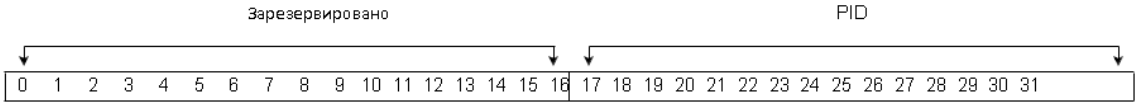
Таблица Б.52 - Регистры MMU SPR

Наименование	Значение при сбросе	SPRN	Описание
PID	x‘XXXX XXXX’	x‘030’	Регистр ID процессора.
RMPD	x‘XXXX XXXX’	x‘339’	Регистр дескрипции страницы реального режима.
MMUBE0	x‘XXXX XXXX’, ‘000’	x‘334’	Регистр 0 фиксированных элементов MMU.

Продолжение таблицы Б.52

Наименование	Значение при сбросе	SPRN	Описание
MMUBE1	x‘XXXX XXXX’, ‘000’	x‘335’	Регистр 1 фиксированных элементов MMU.
SSPCR	x‘0000 000X’	x‘33E’	Регистр конфигурации приоритета поиска супервизора.
USPCR	x‘0000 000X’	x‘33F’	Регистр конфигурации приоритета поиска пользователя.
ISPCR	x‘0000 000X’	x‘33D’	Регистр конфигурации приоритета поиска/инвалидации.
RSTCFG	связанный	x‘39B	Конфигурационный регистр сброса.
MMUCR	‘00000’, ‘X’, x‘000 0000’	x‘3B2’	Конфигурационный регистр MMU.

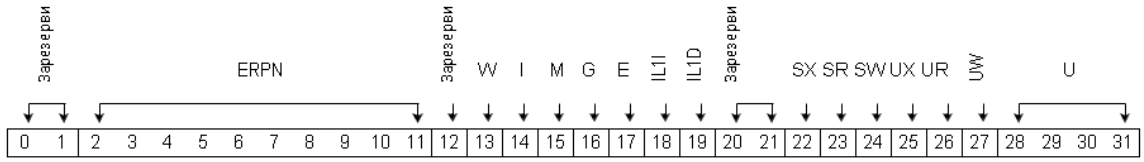
Регистр ID процесса (PID)



Биты	Наименование поля	Описание
0:15	Reserved	
16:31	PID	ID процесса. Это поле используется для отслеживания рассылки MMU с адресом EA[0:19], чтобы хэшировать индексный адрес UTLB.

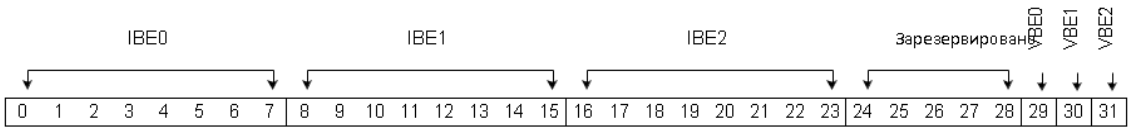
Регистр дескрипции страницы реального режима (RMPD)

Страничный доступ реального режима применяется только для отладки и должен использоваться осторожно. Таким образом, этот регистр зарезервирован разработчиками для отладки.



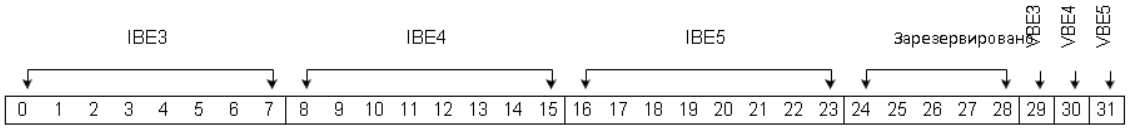
Биты	Наименование поля	Описание
0:1	Зарезервировано	
2:11	ERPN	Расширенный действительный номер страницы
12	Зарезервировано	
13	W	Сквозная запись.
14	I	Кэш запрещен.
15	M	Требование когерентности памяти.
16	G	Защищено.
17	E	Порядок следования байтов.
18	IL1I	Запрет кэша инструкций L1.
19	IL1D	Запрет кэша данных L1.
20:21	Зарезервировано	
22	SX	Разрешение выполнения супревизором.
23	SR	Разрешение чтения супревизором.
24	SW	Разрешение записи супревизором.
25	UX	Разрешение выполнения пользователем.
26	UR	Разрешение чтения пользователем.
27	UW	Разрешение записи пользователем.
28:31	U	U0 - U3. Определяемые пользователем биты.

Регистр 0 фиксированных элементов MMU (MMUBE0)



Биты	Наименование поля	Описание
0:7	IBE0	Только для чтения. Индексный адрес UTLB для фиксированного элемента 0.
8:15	IBE1	Только для чтения. Индексный адрес UTLB для фиксированного элемента 1.
16:23	IBE2	Только для чтения. Индексный адрес UTLB для фиксированного элемента 2.
24:28	Зарезервировано	
29	VBE0	Бит V для фиксированного элемента 0. 0 Нет фиксированного элемента по индексному адресу в IBE0. 1 Есть фиксированный элемент в канале 0 по индексному адресу в IBE0.
30	VBE1	Бит V для фиксированного элемента 1. 0 Нет фиксированного элемента по индексному адресу в IBE1. 1 Есть фиксированный элемент в канале 0 по индексному адресу в IBE1.
31	VBE2	Бит V для фиксированного элемента 2. 0 Нет фиксированного элемента по индексному адресу в IBE2. 1 Есть фиксированный элемент в канале 0 по индексному адресу в IBE2.

Регистр MMU фиксированных элементов 1 (MMUBE1)



Биты	Наименование поля	Описание
0:7	IBE3	Только для чтения. Индексный адрес UTLB для фиксированного элемента 3.
8:15	IBE4	Только для чтения. Индексный адрес UTLB для фиксированного элемента 4.
16:23	IBE5	Только для чтения. Индексный адрес UTLB для фиксированного элемента 5.
24:28	Зарезервировано	
29	VBE3	Бит V для фиксированного элемента 3. 0 Нет фиксированного элемента по индексному адресу в IBE3. 1 Есть фиксированный элемент в канале 0 по индексному адресу в IBE3.
30	VBE4	Бит V для фиксированного элемента 4. 0 Нет фиксированного элемента по индексному адресу в IBE4. 1 Есть фиксированный элемент в канале 0 по индексному адресу в IBE4.
31	VBE5	Бит V для фиксированного элемента 5. 0 Нет фиксированного элемента по индексному адресу в IBE5. 1 Есть фиксированный элемент в канале 0 по индексному адресу в IBE5.

Регистры конфигурации приоритета поиска

Приоритетом поиска UTLB управляют три набора регистров. Два регистра, SSPCR и USPCR, используются для промахов TLB со стороны инструкций и со стороны данных. SSPCR предназначен для режима супервизора/привилегированного (MSR[PR] = ‘0’), а USPCR используется для режима пользователя/задачного (MSR[PR] = ‘1’).

Еще один регистр, конфигурации поиска приоритета/инвалидации (ISPCR), используется для локальных операций tlbsx и tlbivax, а также для входящих snoops, которые являются результатом внешних операций tlbivax. Разделение регистров уменьшает количество страниц в поиске до минимума, увеличивая производительность путем снижения задержки поиска.

Все три набора регистров записываются программой при начальной установке UTLB. Например, в режиме пользователя при наличии множества страниц 4 KB, нескольких страниц 64

KB и небольшого количества страниц 256 MB первый поиск USPCR выполняет при помощи хэша 4 KB, затем хэша 64 KB, и наконец, хэша 256 MB

Регистр конфигурации приоритета поиска супервизора (SSPCR)

Регистр используется при MSR[PR] = ‘0’.

Биты	Наименование поля	Описание
0:3	ORD1	Порядок 1.
4:7	ORD2	Порядок 2.
8:11	ORD3	Порядок 3.
12:15	ORD4	Порядок 4.
16:19	ORD5	Порядок 5.
20:23	ORD6	Порядок 6.
24:27	ORD7	Порядок 7.
28:31	Зарезервировано	

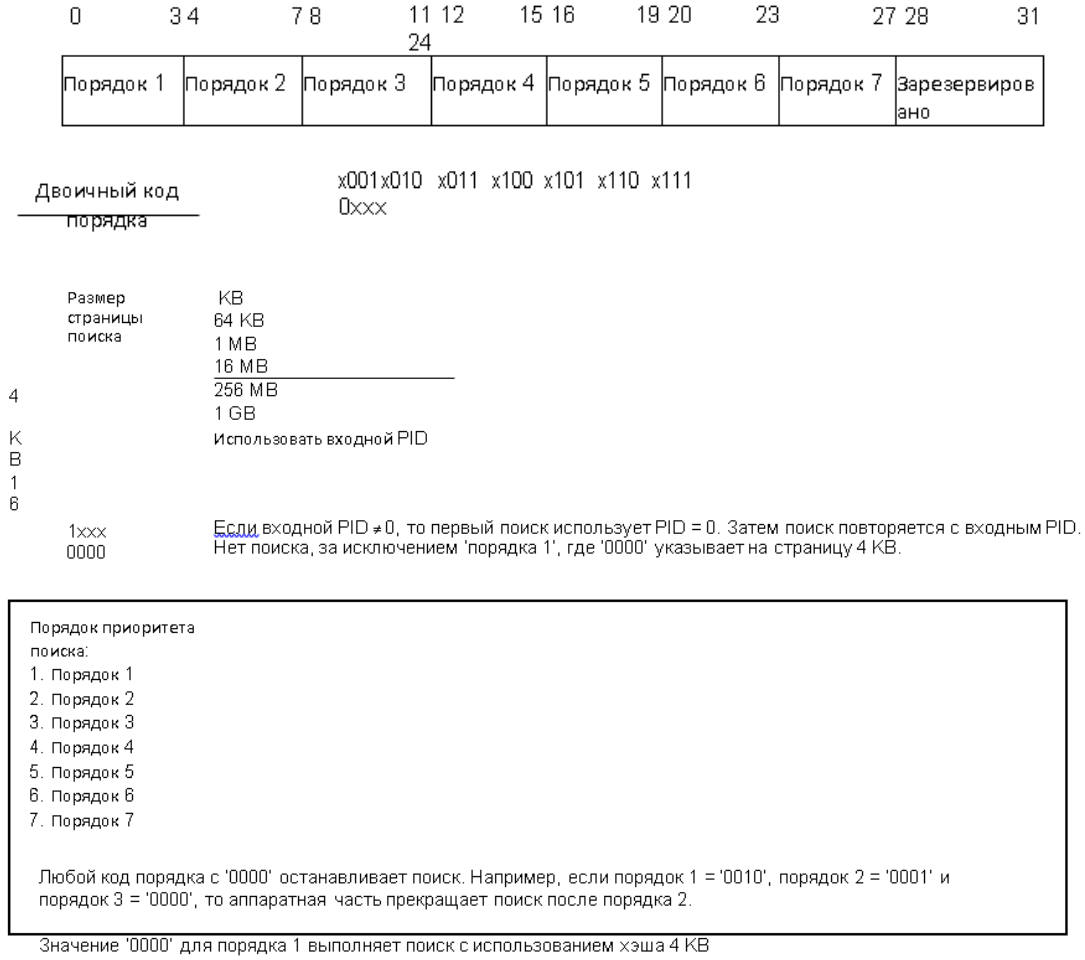
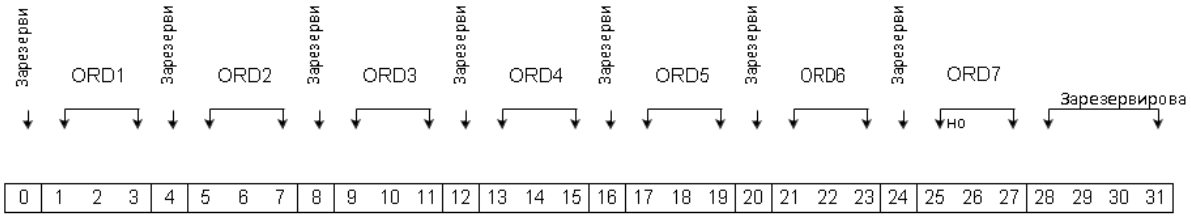


Рисунок Б.7 - Регистры конфигурации приоритета поиска супервизора

Регистр конфигурации приоритета поиска/инвалидации (ISPCR)

ISPCR используется для локальных инструкций tlbsx и tlbivax, а также для входящего слежения, которое является результатом внешних инструкций tlbivax.



Биты	Наименование поля	Описание
0	Зарезервировано	
1:3	ORD1	Порядок 1.
4	Зарезервировано	
5:7	ORD2	Порядок 2.
8	Зарезервировано	
9:11	ORD3	Порядок 3.
12	Зарезервировано	
13:15	ORD4	Порядок 4.
16	Зарезервировано	
17:19	ORD5	Порядок 5.
20	Зарезервировано	
21:23	ORD6	Порядок 6.
24	Зарезервировано	
25:27	ORD7	Порядок 7.
28:31	Зарезервировано	

Двоичный код порядка	Размер страницы поиска
001	4 KB
010	16 KB
011	64 KB
100	1 MB
101	16 MB
110	256 MB
111	1 GB
000	Нет поиска, за исключением 'порядка 1', где '0000' указывает на страницу 4 KB.

Порядок приоритета поиска:
1. Порядок 1
2. Порядок 2
3. Порядок 3
4. Порядок 4
5. Порядок 5
6. Порядок 6
7. Порядок 7

Любой код порядка с '0000' останавливает поиск. Например, если порядок 1 = '010', порядок 2 = '001' и порядок 3 = '000', то аппаратная часть прекращает поиск после порядка 2.

Значение '000' для порядка 1 выполняет поиск с использованием хэша 4 KB

Рисунок Б.8 -. Регистр конфигурации приоритета поиска/инвалидации

Регистр конфигурации приоритета поиска пользователя (USPCR)
Регистр используется при MSR[PR] = ‘1’.

Биты	Наименование поля	Описание
0:3	ORD1	Порядок 1. Значения кодов и размеры страниц для всех этих полей см. на Рис. 4-5 на стр. 126.
4:7	ORD2	Порядок 2.
8:11	ORD3	Порядок 3.
12:15	ORD4	Порядок 4.
16:19	ORD5	Порядок 5.
20:23	ORD6	Порядок 6.
24:27	ORD7	Порядок 7.
28:31	Зарезервировано	

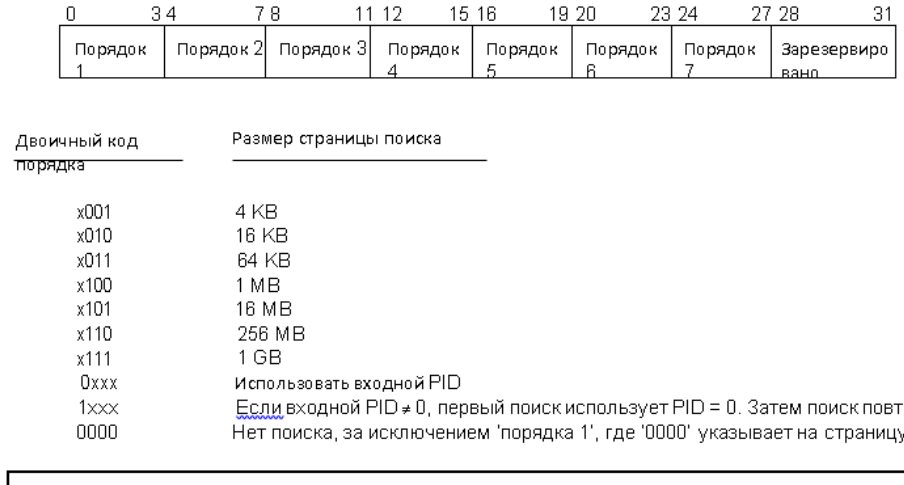


Рисунок Б.10 -. Регистры конфигурации приоритета поиска пользователя (USPCR)

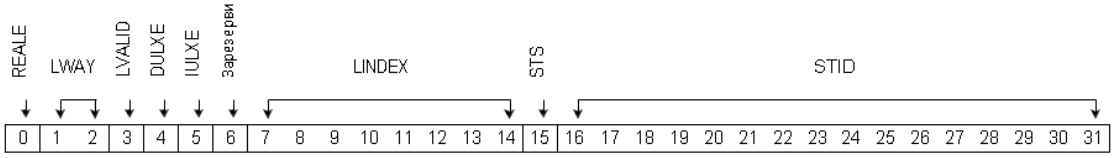
Любой код порядка с ‘0000’ останавливает поиск. Например, если порядок 1 = ‘0000’, порядок 2 = ‘0001’ и порядок 3 = ‘0000’, то аппаратная часть прекращает поиск после порядка 2.

Значение ‘0000’ для порядка 1 выполняет поиск с использованием хэша 4 KB

Регистр конфигурации при сбросе (RSTCFG)

См. Конфигурация при сбросе (RSTCFG).

Регистр конфигурации MMU (MMUCR)



Биты	Наименование поля	Описание
0	REALE	Разрешение реального режима. 0 Биты трансляции адреса, сохранения и разрешения получаются от UTLB. 1 Биты трансляции адреса, сохранения и разрешения получаются от RMPD.
1:2	LWAY	Зафиксированный канал Только для чтения. Устанавливается tlbwe с выбором слова (WS) = 0. Устанавливается в RA[1:2], когда RA[0] = ‘1’ и RA[4] = ‘0’. Устанавливается в ‘00’, когда RA[4] = ‘1’, чтобы фиксированные элементы помещались в канал 0. Устанавливается значение аппаратной поддержки, когда RA[0] = ‘0’. Используется tlbwe с WS = 1 и WS = 2.
3	LVALID	Зафиксированная допустимость. Только для чтения. Устанавливается tlbwe с WS = 0 в значение бита V. Очищается при tlbwe WS = 2 или когда tlbivax дает совпадение LINDEX до tlbwe WS = 2.
4	DULXE	Разрешение исключения типа зафиксированной пользователем строки данных. 0 Попытка выполнить dcbf в режиме пользователя не генерирует исключение. 1 Попытка выполнить dcbf в режиме пользователя генерирует исключение.
5	IULXE	Разрешение исключения типа зафиксированной пользователем строки инструкций. 0 Попытка выполнить icbi в режиме пользователя не генерирует исключение. 1 Попытка выполнить icbi в режиме пользователя генерирует исключение.
6	Зарезервировано	
7:14	LINDEX	Зафиксированный индексный адрес. Только для чтения. Устанавливается tlbwe с WS = 0. Используется для индексного адреса tlbwe с WS = 1 и WS = 2.
15	STS	Устанавливает пространство трансляции. Устанавливается программой до tlbwe.
16:31	STID	Устанавливает ID трансляции. Устанавливается программой до tlbwe. Записывается в поле UTLB TID во время tlbwe, WS = 0 Используется в хэше индексного адреса при выполнении инструкций tlbsx и tlbivax (локальной).

Описание блока UTLB

UTLB состоит из следующих частей:

Массив тегов

Массив данных

Когерентность TLB

Массивы организованы таким образом, что одновременно могут считываться четыре элемента.

Массив тегов

Массив тегов содержит информацию, необходимую для определения, совпадает ли с допустимым элементом запрос UTLB от EU, ICU, DCU или интерфейса слежения.

Особенности программирования

UTLB ядра PowerPC 476FP управляется программно. Инициализация элементов и инвалидация элементов (за исключением внешне инструкции tlbivax) требуют явных программных инструкций. Типичный поиск UTLB, являющийся результатом промахов в ITLB или DTLB, выполняется аппаратной частью. Программа должна обрабатывать четыре инструкции: tlbsx, tlbre, tlbwe, tlbivax.

Индексный поиск TLB (tlbsx)

Инструкция lbsx предоставляет пространство трансляции (посредством MMUCR[STS]), ID трансляции (посредством MMUCR[STID]) и исполнительный адрес (EA), которые используются для поиска UTLB. Поиск в UTLB ведется в соответствии с порядком, указанным в ISPCR. Если совпадение найдено, то возвращаются канал и индексный адрес.

Формат инструкции tlbsx:

tlbsx RT, RA, RB Rc

Исполнительный адрес вычисляется следующим образом:
EA = (RA) + (RB)
если RA = 0, EA = 0 + (RB)
если Rc = 1
CR[CR0[0]] □□0
CR[CR0[1]] □□0 CR[CR0[3]] □□XER[SO]

Виртуальный адрес равен MMUCR[STS] | MMUCR[STID] | EA[0:n], где n = 19 для страницы 4 KB.

Если в UTLB найден допустимый элемент со следующими совпадениями: MMUCR[STS] совпадает с tlbentry[TS], и если tlbentry[TID] □□□‘0’, MMUCR[STID] совпадает с tlbentry[TID], и EA[0:19] совпадает tlbentry EPN[0:19]

тогда
RT[33:34] □□Попадание канала, передается на MMU_iwbRdData[1:2]
RT[40:47] □□индексный адрес совпадающего элемента TLB, передается на MMU_iwbRdData[8:15] если Rc = 1
CR[CR0[2]] □□1
иначе
(RT) □□не определен
если Rc = 1
CR[CR0[2]] □□0

Чтение элемента TLB (tlbre)

Инструкция tlbre используется для чтения содержимого элемента. Перед инструкцией tlbre необходимо выполнить

инструкцию tlbsx, чтобы получить канал UTLB и индексный адрес. Формат инструкции tlbre:

tlbre RT, RA, WS

TLB[(RA)33:34] указывает канал tlbentry, котоый будет считываться. TLB[(RA)40:47] указывает индексный адрес tlbentry. Программа может определить, является ли элемент фиксированным, сравнивая его с регистрами MMUBE0 и MMUBE1.

если WS = 0
RT[32:60] □□tlbentry[EPN(0:19), V, TS, DSIZ(0:5), BLTD] если CCR0[CRPE] = 0 RT[61:63] □□000
иначе
RT[61:63] □□tlbentry[EPNPar, DSIZPar, TIDPar] MMUCR[STID] □□tlbentry[TID]
иначе если WS = 1
RT[32:51] □□tlbentry[RPN] RT[54:63] □□tlbentry[ERPN] если CCR0[CRPE] = 0
RT[52] □□0
RT[53] □□0
иначе
RT[52] □□tlbentry[RPNPar] RT[53] □□tlbentry[ERPNPar]
иначе if WS = 2
RT[46:56] □□tlbentry[IL1I, IL1D, U(0:3),W,I,M,G,E], передается на MMU_iwbRdData[14:24]
RT[58:63] □□tlbentry[UX,UW,UR,SX,SW,SR]
если CCR0[CRPE] = 0 RT[32] □□0’b0
иначе
RT[32] □□Четность сохранения/разрешения иначе (RT),

MMUCR[STID] □□*не определен*

Запись элемента TLB (tlbwe)

Инструкция tlbwe используется программой для помещения элементов в UTLB. Программа может указать канал для записи или использовать аппаратную поддержку счетчиков каналов. Если канал указан программой, запись может быть зафиксирована, что защитит ее от последующих записей с аппаратной поддержкой выбора канала или tlbivax (локальной или удаленной). Программа может инвалидировать элементы, отчищая бит достоверности при помощи tlbwe. Фиксированные записи, инвалидированные таким образом, должны либо указывать, что запись фиксированная (с RA[36] = ‘1’), либо указывать канал 0 в инструкции tlbwe (с RA[32:34] = ‘100’).

Формат инструкции:

tlbwe RS, RA, WS

Если TLB[(RA)36] = 1, канал записываемого tlbentry равен '0',

иначе если TLB[(RA)36] = 0 и TLB[(RA)32] = 1, канал tlbentry указывается TLB[(RA)33:34], иначе определяется счетчиком каналов, соответствующим индексному адресу

Если TLB[(RA)36] = 1, tlbentry будет записан как фиксированный. TLB[(RA)37:39] указывает фиксированный элемент, 000 - 101

если WS = 0

Индексный адрес UTLB получается хэшированием входного EPN[0:19] и DSIZ с MMUCR STID[8:15] Канал UTLB, индексный адрес и бит достоверности фиксируются в MMUCR[LWAY], MMUCR[LINDEX] и MMUCR[LVALID] для использования с WS = 1 и WS = 2

если бит V = ‘1’,

tlbentry[EPN, V, TS, DSIZ] □□□RS[32:59]

EPN, TS, и DSIZ записаны в UTLB. Бит V не записывается до WS = 2.

tlbentry[TID] □ MMUCR[STID]

if V-bit = ‘0’,

tlbentry[V] □□‘0’

иначе если WS = 1

tlbentry[RPN] □□□□RS[32:51]

tlbentry[ERPN] □□□□RS[54:63]

запись в канал UTLB по индексному адресу, указанному MMUCR[LWAY] и MMUCR[LINDEX],

иначе если WS = 2

tlbentry[IL1I,IL1D,U(0:3),W,I,M,G,E] □□□RS[46:56]

tlbentry[UX,UW,UR,SX,SW,SR] □□□RS[58:63]

запись в канал UTLB по индексному адресу, указанному MMUCR[LWAY] и MMUCR[LINDEX]

tlbentry[V] □□MMUCR[LVALID]

иначе

tlbentry □□□не определен

Биты четности автоматически вычисляются аппаратной частью и вставляются в массивы тегов и данных UTLB.

Инвалидация TLB, индексированный адрес (tlbivax)

Инструкция tlbivax выполняет поиск в UTLB и при нахождении совпадения инвалидирует первую совпадающую запись. Формат инструкции tlbivax:

tlbivax RA, RB

Исполнительный адрес вычисляется следующим образом: EA = (RA) + (RB)

если RA = 0, EA = 0 + (RB)

Если в UTLB найден достоверный элемент со следующими совпадениями, и этот элемент не указан как фиксированный в регистрах MMUBE0 или MMUBE1, то этот элемент инвалидируется записью 0 в его бит V:

MMUCR[STS] в tlbentry[TS], и

если tlbentry[TID] □□0, MMUCR[STID] вtlbentry[TID], и

EA[0:19] в tlbentry EPN[0:19]

Используется порядок хэширования, определенный ISPCR. Поскольку программа устанавливает MMUCR[STID], то нет необходимости принудительно устанавливать ненулевой PID в 0 для первого набора поисков.

MMUCR[STS], MMUCR[STID] и EA[0:19] также передаются из ядра через L2 на шину PLB6, так что другие когерентные процессоры могут инвалидировать совпадающий элемент.

Инструкция tlbivax не инвалидирует теневые TLB (ITLB и DTLB). Для этого нужна дополнительная инструкция tlbsync.

Когерентность UTLB

Поддержка когерентности UTLB включает четыре операции:

локальная tlbivax

локальная tlbsync (обрабатывается исполнительным блоком, генерирует сигнал ожидания)

удаленная tlbivax

удаленная tlbsync (обрабатывается L2, генерирует сигнал ожидания)

MMU отвечает на локальную инструкцию tlbivax инвалидацией совпадающего не фиксированного элемента UTLB. Блок SYNC отвечает на локальную инструкцию tlbivax передачей соответствующих MMUCR[STS], MMUCR[STID] и EA[0:19] через L2 на шину PLB6.

Программа - точнее, ее ядро - управляет UTLB процессора PowerPC 476FP. Обязательное требование - инструкции tlbivax и tlbsync должны выполняться и действовать в системе последовательно. Несколько процессоров никогда не обрабатывают не не выполняют эти инструкции одновременно.

Специальные операции tlbsync

Главный процессор выполняет следующую последовательность инструкций, чтобы обеспечить инавалидацию элементов TLB и синхронизировать эффекты tlbivax:

mtmmucr RS Устанавливает поля STS и STID для операции tlbivax.

isync Обеспечивает завершение MMUCR.

tlbivax Инвалидирует элемент TLB.

... Последовательность нескольких инструкций mtmmucr и tlbivax.

isync Очищает локальные ITLB и DTLB для принудительной синхронизации с UTLB.

mbar Обеспечивает, что последующая инструкция tlbsync не обойдет инструкции tlbivax.

tlbsync Синхронизирует удаленные процессоры и обеспечивает, что все процессоры завершат выполнение всех ожидающих инструкций tlbivax.

msync Обеспечивает, что все инструкции tlbivax и tlbsync завершаются до начала следующей инструкции.

Удаленная операция tlbsync

Удаленная операция tlbsync является строгой в том смысле, что кэш L2 обеспечивает завершение процессором инструкции tlbsync до того, как принимать следующую инструкцию msync. Попытки выполнения msync повторяются до завершения tlbsync. После завершения tlbsync система может обнаружить все операции сохранения, включая инструкцию и выборки.

Во время удаленной операции tlbsync процессор обеспечивает завершение всех предшествующих операций, включая выборку инструкций, а также следит за тем, что операция синхронизации контекста sync завершена.

Удаленная операция tlbsync CPU

Последовательность удаленной операции tlbsync:

Обнаружить удаленную операцию tlbsync посредством IU через блок SYNC.

Очистить все незавершенные инструкции и прекратить выборку инструкций. IU не даст ICU выполнять выборку. Теневые TLB, ITLB и DTLB, инвалидируются (операцией переключения контекста).

Для LRACC выдается псевдоудаленная операция tlbsync, чтобы маркировать псевдооперацию в EU и DCU. Поскольку это псевдооперация, она не подтверждается и не исполняется.

Убедиться, что завершены все операции чтения и записи.

Выдать блоку SYNC команду записи x‘F’.

Блок SYNC отправляет команду в кэш L2, указывая, что удаленная операция tlbsync завершена. Когда блок SYNC получает подтверждение от кэша L2 (или становится доступным), блок SYNC указывает IU, что удаленная операция tlbsync завершена.

IU отправляет запрос выборки ICU, чтобы возобновить выполнение инструкций.

Удаленные операции tlbsync кэша L2

Кэш L2 должен обеспечить, что все предшествующие удаленные операции tlbivax и tlbsync завершены удаленным сопроцессором, прежде чем принимаются последующие инструкции msync. Этот удаленный кэш L2 не гарантирует выполнение операций загрузки и сохранения, которые могут использовать предыдущие трансляции.

Последовательность выполнения удаленной операции tlbsync:

Обнаружить удаленную инструкцию tlbsync на PLB6.

Установить флаг, указывающий на выполнение инструкции tlbsync. Любая последующая удаленная инструкция msync повторяет попытки выполнения до тех пор, пока tlbsync не будет завершена сопроцессором.

Кэш L2 завершает все ожидающие запросы от сопроцессора, включая выборку инструкций. Кэш L2 продолжает все обычные операции, за исключением инструкции msync от PLB6.

Кэш L2 завершает операцию tlbsync сбросом флага tlbsync, когда сопроцессор возвращает команду записи lwsync, x‘F’.

Кэш L2 принимает удаленную msync и подтверждение.

Кэш инструкций и данных

Ядро PowerPC 476FP имеет два отдельных контроллера и массива кэша уровня 1 (L1): кэш инструкций (I-кэш) и кэш данных (D- кэш.) Эти контроллеры и массивы позволяют выполнять параллельные запросы и минимизируют остановки конвейера. Каждый массив кэша имеет размер 32 KB. Оба контроллера кэша имеют 32-байтовые каналы. Оба контроллера кэша являются 4-канальными и ассоциативными. Реализация ядра PowerPC 476FP также предоставляет специальные инструкции отладки, которые могут непосредственно считывать массивы данных и тег. Контроллер кэша инструкций и контроллер кэша данных связаны с кэшем уровня 2 (L2). Интерфейс кэша L2 состоит из 256-битовой общей шины чтения (чтение из кэша L2) и 128-битовой шины записи (запись в кэш L2).

Оба кэша поддерживают симметричную многопроцессорную (SMP) когерентность через внутреннее соединение, локальную процессорную шину 6 (PLB6), и поддерживают до восьми согласованных управляющих устройств и процессоров. Для облегчения программирования оба кэша совместимы с Power Instruction Set Architecture (ISA) Version 2.05.

Оба кэша имеют защиту четности от ошибок программы. При обнаружении таких ошибок процессор обращается к обработчику прерываний проверки процессора, где программа может предпринять необходимые действия.

Далее в этом разделе представлена более подробная информация о работе контроллеров и массивов кэша инструкций и кэша данных.

Организация и работа массива кэша

Кэш инструкций и кэш данных организованы одинаково. Тем не менее, поля для тегов и данных в массивах немного различаются, поскольку функции у массивов разные. Кэш инструкций и кэш данных тегированы действительным адресом (RA).

Каждый кэш представляет собой 4-канальный ассоциативный кэш. Каждый кэш имеет 256 групп, а размер строки каждого кэша составляет 32 байта.

Таблица Б.53 иллюстрирует общую организацию каналов и групп в массивах кэша, а в таблице Б.54 приведены конкретные значения параметров, представленных в таблице Б.53.

Таблица Б.53 - Организация массивов кэша инструкций и кэша данных

Группа 0	Строка 0	Строка n	Строка 2n	Строка (w – 1)n
	Канал 0	Канал 1	Канал 2	Канал 3
Группа 1	Строка 1	Строка n + 1	Строка (w – 2)n + 1	Строка (w – 1)n + 1
□	□	□	□	□
□	□	□	□	□
□	□	□	□	□
Группа 254	Строка n - 2	Строка 2n – 2	Строка (w – 1)n – 2	Строка wn – 2
Группа 255	Строка n - 1	Строка 2n – 1	Строка (w – 1)n – 1	Строка wn – 1

Как показано в таблице Б.54, поле тега для каждой строки в каждом канале сдержит старшие биты адреса, ассоциирующегося со строкой, которая в данный момент присутствует в этом канале. Средние биты адреса формируют индекс для выбора конкретной группы кэша, а пять младших битов адреса формируют байтовое смещение для выбора конкретного байта (или нескольких байт, в зависимости от размера операции) из 32-байтовой строки кэша.

Таблица Б.54 - Размер и параметры кэша

Размер кэша	Каналы(w)	Группы(n)	Биты тега адреса	Биты установки адреса	Биты байтового смещения адреса.
32 KB	4	256	RA[0:18]	RA[29:36]	RA[37:41] или EA[27:31]

Биты тега адреса, показанные в таблице Б.54, являются битами RA и приведены только для иллюстрации. Поскольку кэш инструкций тэгируется виртуальным адресом, а кэш данных тегируется действительным адресом, то биты тега адреса в каждом массиве разные.

Контроллер кэша инструкций

Блок кэша инструкций (ICU) за один цикл выдает четыре инструкции блоку инструкций (IU) ядра PowerPC 476FP.

ICU также обеспечивает выполнение инструкций управления кэшем инструкций PowerPC для предвыборки или инвалидации строк кэша, или для групповой инвалидации всего кэша. Предоставляются также ресурсы для управления и отладки операций кэша инструкций.

Операции I-кэша

Кэш инструкций может принимать четыре типа операций кэша инструкций:

icbt (инструкция предвыборки блока кэша), включая icbtls (инструкция предвыборки блока кэша и установки фиксации) и icblc (инструкция очистки фиксации блока кэша)

icbi (инструкция инвалидации блока кэша)

ici (инструкция инвалидации кэша)

icread (инструкция чтения кэша)

Кроме того, кэш инструкций поддерживает операции четности. Кэш инструкций содержит биты четности и аппаратную часть обнаружения множества попаданий, чтобы защитить от программных ошибок данных.

Операции четности кэша инструкций

Кэш инструкций содержит биты четности и аппаратную часть обнаружения множества попаданий, чтобы защитить от программных ошибок данных. Логика четности кэша инструкций может обнаруживать два типа ошибок. В первом типе биты четности, хранящиеся в массиве RAM, проверяются для соответствующих данных в строке кэша инструкций, когда строка RAM считывается для выборки инструкций. Следует учитывать, что ошибка четности не сигнализируется как результат инструкции icread.

Второй тип ошибок, которые могут обнаруживаться - это множество попаданий. Этот тип ошибки возникает тогда, когда бит тега адреса искажен, в результате чего два тега в массиве кэша инструкций совпадают с одним и тем же входным адресом. Ошибки множества совпадений могут быть обнаружены при любой выборке инструкции. Ошибки четности любого рода не обнаруживаются при поиске спекулятивной выборки или поиске icb. Такой поиск рассматривается как попадание кэша и не вызывает дальнейших действий до того, как поиск выборки инструкций по проблемному адресу не вызовет ошибку, которая будет обнаружена.

При обнаружении ошибки четности и установленном MSR[ME] (то есть, разрешены прерывания проверки процессора), процессор передает управление блоку обработки прерываний проверки процессора. Как и в случае любого прерывания проверки процессора, после передачи управление блоку обработки прерываний проверки процессора MCSRR0 содержит значение самой старой незавершенной инструкции в конвейере в момент появления исключения, а MCSRR1 содержит старый контекст MSR. Блок обработки прерываний может при помощи инструкции icccі опрашивать регистр состояния проверки процессора (MCSR), чтобы определить, обусловлен ли вызов ошибкой четности кэша инструкций.

Блок обработки возвращается к прерванному процессу при помощи инструкции rfmci.

Если прерывания проверки четности и проверки процессора разрешены, то ошибки четности кэша инструкций всегда исправляемые. Следует также отметить, что прерывание проверки процессора является асинхронным; то есть, адрес возврата в MCSRR0 не указывает на адрес инструкции с ошибкой четности. Прерывание проверки процессора принимается сразу же при обнаружении ошибки. Некоторые выполняемые инструкции сбрасываются и повторно выполняются после прерывания, как будто процессор реагирует на внешнее прерывание.

Инструкция очистки фиксации блока кэша (icblc)

Инструкция icblc очищает бит фиксации для данной строки кэша инструкций. Если поле CT установлено в ‘0’ (для кэша L1), icblc очищает бит фиксации для последнего использованного

(LRU), достоверного или фиксированного массива в кэше инструкций. Если поле CT установлено в 2, то инструкция icbic отправляет запрос на очистку фиксации в L2. Целевая строка остается достоверной в кэше.

Инструкция инвалидации блока кэша (icbi)

Если блок, содержащий байт, адресуемый EA, находится в кэше инструкций любого процессора, то блок (строка кэша) инвалидируется в кэше инструкций. Эта инструкция передается всем процессорам на PLB.

Инструкция инвалидации кэша (icbi)

Эта инструкция инвалидирует весь кэш инструкций L1. Инструкция ici не направляется в кэш L2. Инструкция ici генерирует исключение, если не выполняется в режиме супревизора. Если поле инструкции CT равно 2, то инструкция считается фиктивной.

В Power Instruction Set Architecture (ISA) указывается, что программа должна поместить инструкцию isync после инструкции ici, чтобы инвалидировать любые инструкции, которые могли быть уже выбраны из прежнего содержимого кэша инструкций после isync.

icbt

Инструкция icbt - это подсказка для установки конкретной строки кэша, но операция не гарантирует установку строки кэша. Таким образом, строка кэша может быть отслежена, или строка кэша может быть очищена инструкцией icbi.

После получения icbt ICU проверяет буферы заполнения строки, очередь выборки и тег, чтобы определить, находится ли запрашиваемая строка уже в кэше. Если запрашиваемой строки нет в кэше, запрос попадает в очередь выборки. Эта операция использует поле CT инструкции. Если CT установлено в ‘0’, и запрос промахивается на кэше инструкций, то запрашивается буфер заполнения, и отправляется запрос на кэш L2. В случае попадания кэша при CT = ‘0’ никаких действий не требуется.

Когда CT = 2, для кэша инструкций никаких действий не предпринимается, но отправляется запрос на кэш L2. Поскольку icbt не гарантирует записи данных в кэш, то контроллер кэша инструкций не требует ответа от контроллера кэша L2.

icbtls

Инструкция icbtls аналогична инструкции icbt, за исключением того, что она при записи фиксирует данные в кэше. Это значит, что при выборке вместе с запросом чтения на кэш L2 посылается сигнал фиксации. Для строк, которые уже находятся в кэше, бит фиксации устанавливается при заполнении строки.

Эта инструкция также использует поле CT. Когда CT = 2, фиксируется только строка в кэше инструкций. Если строка уже находится в кэше инструкций, строка фиксируется, и в кэш L2 запрос не отправляется. Когда CT = 2, контроллеру кэша L2 отправляется управляющий сигнал, указывающий, что строка также должна фиксироваться в L2.

И в этом случае кэшу инструкций не требуется ответ от кэша L2.

icread

Эта инструкция считывает содержание указанного физического местоположения в кэше инструкций и сохраняет данные в регистрах отладки. Контроллер кэша не выполняет трансляцию адрес или обработку исключений для этой инструкции. Поскольку доступ осуществляется к содержанию конкретного местоположения в кэше, запрос icread использует модифицированный формат для EA. В таблице Б.55 описывается формат EA для инструкции icread.

Таблица Б.55 - Формат EA для icread

Биты адреса	Описание
0:16	Не используются.
17:18	Канал кэша инструкций
19:26	Индекс кэша инструкций
27:29	Адрес слова внутри строки кэша инструкций L1.
30:31	Не используются.
Примечание: Инструкция icread не направляется в кэш L2.	

Примечание - Ядро PowerPC 476FP автоматически не синхронизирует контекст между инструкцией icread и последующими инструкциями mfspr, которые считывают результаты инструкции icread в регистры общего назначения (GPRs). Чтобы гарантировать, что инструкции mfspr получат результаты инструкции icread, необходимо использовать последовательность инструкций, как в приведенном ниже примере:

```
icread   regA,regB      # Читать информацию кэша. Добавляется содержимое GPR A и GPR B,
# и результат используется для указания индекса строки кэша, которая будет считываться.
isync                   # Убедиться, что icread завершена перед попыткой чтения результатов.
mficdbdr0    regC       # Переслать информацию инструкции в GPR C.
mficdbdr1    regD       # Переслать информацию инструкции в GPR D.
mficdbtrh    regE       # Переслать старшие биты тега в GPR E.
mficdbtrl    regF       # Переслать младшие биты тега в GPR F.
```

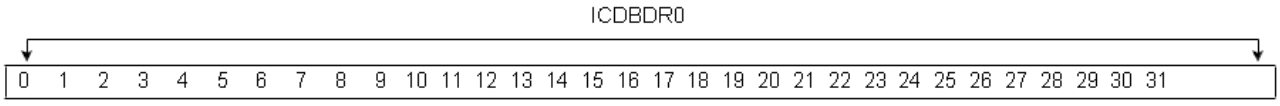
В результате операции icread выполняется запись в следующие регистры специального назначения (SPR):

ICDBDR0

ICDBDR1

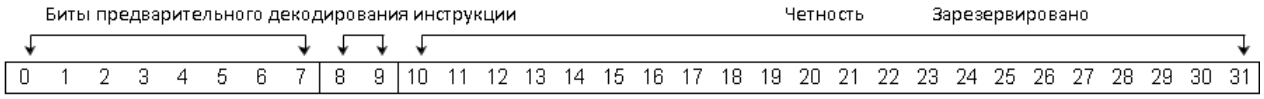
ICDBTRL

ICDBTRH Регистр данных отладки кэша инструкций 0 (ICDBDR0)



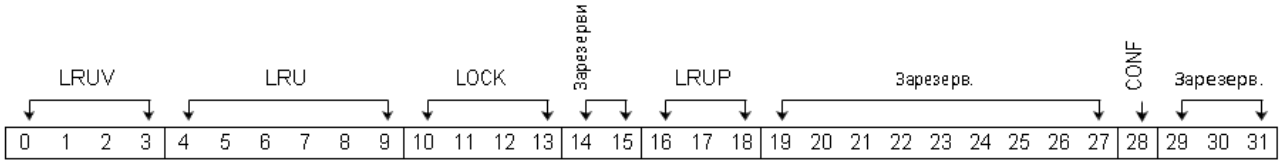
Биты	Описание
0:31	Слово инструкции

Регистр данных отладки кэша инструкций 1 (ICDBDR1)



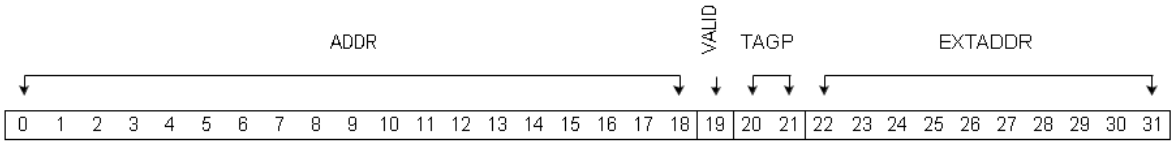
Биты	Описание
0:7	Биты предварительного декодирования инструкции
8:9	Четность.
10:31	Зарезервировано.

Регистр тегов отладки кэша инструкций, младшие разряды (ICDBTRL)



Бит	Наименование поля	Описание
0:3	LRUV	Биты достоверности LRU. Один бит для каждого канала в группе.
4:9	LRU	Значение LRU для группы.
10:13	LOCK	Биты фиксации. Один бит для каждого канала в группе.
14:15	Зарезервировано	
16:18	LRUP	Четность LRU. Четность для массива, биты [0:5]. Четность для массива, биты [6:9]. Четность для массива, биты [10:14].
19:27	Зарезервировано	
28	CONF	Бит конфликта каналов.
29:31	Зарезервировано	

Регистр тегов отладки кэша инструкций, старшие разряды (ICDBTRH)



Бит	Наименование поля	Описание
0:18	ADDR	Адрес тега.
19	VALID	Бит достоверности для этого элемента. Этот бит является битом тега и получается из массива LRU.
20:21	TAGP	Биты четности тега.
22:31	EXTADDR	Расширенный адрес тега.

Операции четности кэша инструкций

Кэш инструкций содержит биты четности и аппаратную часть обнаружения множества попаданий, чтобы защитить от программных ошибок данных. Логика четности кэша инструкций может обнаруживать два типа ошибок. В первом типе биты четности, хранящиеся в массиве RAM, проверяются для соответствующих данных в строке кэша инструкций, когда строка RAM считывается для выборки инструкций. Следует учитывать, что ошибка четности не сигнализируется как результат инструкции icread

Второй тип ошибок, которые могут обнаруживаться - это множество попаданий. Этот тип ошибки возникает тогда, когда бит тега адреса искажен, в результате чего два тега в массиве кэша инструкций совпадают с одним и тем же входным адресом. Ошибки множества совпадений могут быть обнаружены при любой выборке инструкции. Ошибки четности любого рода не обнаруживаются при поиске спекулятивной выборки или поиске icb. Такой поиск рассматривается как попадание кэша и не вызывает дальнейших действий до того, как поиск выборки инструкций по проблемному адресу не вызовет ошибку, которая будет обнаружена.

При обнаружении ошибки четности и установленном MSR[ME] (то есть, разрешены прерывания проверки процессора), процессор передает управление блоку обработки прерываний проверки процессора. Как и в случае любого прерывания проверки процессора, после передачи управления блоку обработки прерываний проверки процессора MCSRR0 содержит значение самой старой незавершенной инструкции в конвейере в момент появления исключения, а MCSRR1 содержит старый контекст MSR. Блок обработки прерываний может при помощи инструкции icccci опрашивать регистр состояния проверки процессора (MCSR), чтобы определить, обусловлен ли вызов ошибкой четности кэша инструкций.

Блок обработки возвращается к прерванному процессу при помощи инструкции rfmci.

Если прерывания проверки четности и проверки процессора разрешены, то ошибки четности кэша инструкций всегда исправляемые. Следует также отметить, что прерывание

проверки процессора является асинхронным; то есть, адрес возврата в MCSRR0 не указывает на адрес инструкции с ошибкой четности. Прерывание проверки процессора принимается сразу же при обнаружении ошибки. Некоторые выполняемые инструкции сбрасываются и повторно выполняются после прерывания, как будто процессор реагирует на внешнее прерывание.

Спекулятивная предвыборка

В общем случае все инструкции выбираются спекулятивно. ICU выбирает (или предварительно выбирает) два кодовых потока инструкций: один для последовательного потока, а другой для потока предсказанного ветвления.

Поскольку ICU выдает четыре инструкции одновременно, он обращается к следующей строке кэша или к строке кэша предсказанной инструкции даже если он не определит, требует ли программный код этих инструкций. Таким образом, выполняется спекулятивная предвыборка инструкций.

Процессор ограничивает спекулятивную выборку, поскольку излишняя спекулятивная выборка снижает эффективность работы кэша, замещая строки кэша ненужными строками предвыборки.

Вся выборка из кэша L2 выполняется при помощи действительных (физических) адресов.

Исключения

Кэш инструкций генерирует три разных типа исключения:

- Прерывание памяти инструкции
- Промах инструкции объединенного буфера ассоциативной трансляции (UTLB)
- Проверка процессора со стороны инструкций

Эти исключения передаются IU, где инструкция тегируется как специальное исключение со стороны инструкций. Затем оно передается на стадию записи в регистр конвейера инструкций как ошибочное поручение, выполненное IU. Эти три исключения являются взаимоисключающими. Ошибочное поручение состоит из четырех фиктивных инструкций, переданных на декодирование и выдачу (DISS) и маркированных как ошибка. Эти инструкции остаются допустимыми до очистки. Последующая выборка в кэш L2 блокируется. Любые данные, которые кэш L2 возвращает со статусом проверки процессора, не записываются в кэш. Тем не менее, Слежение продолжает выполняться.

Прерывание памяти инструкции

Кэш инструкций генерирует прерывание памяти инструкции (ISI) только во время инструкции нарушения защиты выполнения. Это происходит, когда запрашиваемая модулем управления памятью (MMU) страница возвращается без разрешения выполнения.

Промах инструкции UTLB

Кэш инструкций генерирует промах инструкции UTLB, когда запрос MMU на доступ к элементу TLB инструкций (ITLB) приводит к промаху UTLB.

Проверка процессора со стороны инструкций

Кэш инструкций генерирует проверку процессора со стороны инструкций, когда обнаруживается аппаратная ошибка в ICU. Это может быть ошибка чтения от L2 или ошибка четности из нескольких источников. Кэш инструкций содержит регистр синдрома ошибки кэша инструкций (ICESR), SPR, который предназначен для различения произошедших ошибок. ICESR доступен только в режиме супервизора и очищается при чтении.

Регистры специального назначения ICU

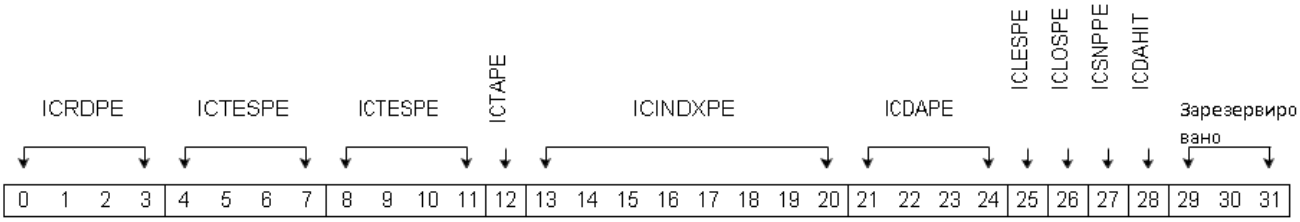
В таблице Б.56 приведены SPR, используемые в ICU.

Таблица Б.56 - Регистры специального назначения ICU

Регистр	Наименование	Адрес	Чтение/запись	Привилеги-рованный
ICESR	Регистр синдрома ошибки кэша инструкций	x‘851’	R/W	Да
ICDBDR0	Регистр данных отладки кэша инструкций, инструкция	x‘979’	R	Да
ICDBDR1	Регистр данных отладки кэша инструкций, предварительное декодирование	x‘980’	R	Да
ICDBTRL	Регистр тегов отладки кэша инструкций, младшие разряды	x‘926’	R	Да
ICDBTRH	Регистр тегов отладки кэша инструкций, старшие разряды	x‘927’	R	Да

Регистр синдрома ошибки кэша инструкций (ICESR)

ICESR предоставляет синдром для различения исключений, которые могут генерировать один и тот же тип прерывания. При генерировании одного из этих типов прерываний устанавливаются бит или биты, соответствующие конкретному исключению, которое вызвало это прерывание, а все остальные биты ICESR очищаются.



Бит	Наименование поля	Описание
0:3	ICRDPE	Ошибка четности интерфейса чтения кэша инструкций. Номер бита указывает, какое слово содержит ошибку на шине данных. Могут быть установлены несколько битов.
4:7	ICTESPE	Ошибка четности четной группы тега.
8:11	ICTESPE	Ошибка четности нечетной группы тега.
12	ICTAPE	Ошибка четности в теге SRAM.
13:20	ICINDXPE	Индекс ошибки четности в кэше. Отражает биты 19:26 действительного адреса.
21:24	ICDAPE	Ошибка четности в ISD.
25	ICLESPE	Ошибка четности в четной группе LRU/достоверный SRAM.
26	ICLOSPE	Ошибка четности в нечетной группе LRU/достоверный SRAM.
27	ICSNPPE	Ошибка четности отслеживания кэша инструкций. Ошибка четности на запросе отслеживания, полученном от кэша L2.
28	ICDAHIT	Попадание в массив данных кэша инструкций. При установке этот бит модифицирует ICDAPE. Если установлены и ICDAPE, и ICDAHIT, это указывает на ошибку четности данных на запросе загрузки, который попадает в кэш инструкций. Если установлен только ICDAPE, ошибка четности относится к запросу, который обслуживается из буферов заполнения строки. Если ICDAPE не установлен, этот бит следует игнорировать.
29:31	Зарезервировано	

Регистры конфигурации ядра CCR0, CCR1 и CCR2 помогают выполнять отладку и управление функциями.

Самомодифицирующийся код

Этот пример самомодифицирующегося кода иллюстрирует использование инструкций управления кэшем для улучшения когерентрости кэша. В данном примере программа, выполняющаяся на ядре PowerPC 476FP, сохраняет новые данные в памяти с целью последующего ветвления и выполнения этих данных, которые являются инструкциями.

Приведенный пример кода иллюстрирует последовательность, которую должна использовать программа при записи самомодифицирующегося кода. В примере предполагается, что addr1 указывает кэшируемую страницу памяти.

```
stw     regN, addr1    # Сохранить данные (инструкцию) в regN по addr1 кэше данных. dcbst
addr1  # Записать новую инструкцию из кэша данных в память.
msync # Ждать записи данных в память.
icbi     addr1   # Инвалидировать addr1, если он существует, в кэше инстукций.
isync   # Очистить все предварительно выбранные инструкции в ICU и блоке
# инструкций и выполнить их повторную выборку. Старая копия
# инструкций по addr1
# могла быть уже выбрана.
```

На этом этапе программа может начать выполнение инструкции из addr1, и при этом гарантируется, что новая инструкция распознана.

Контроллер кэша данных

Кэш данных представляет собой модель со сквозной записью и поэтому все сохраняемые данные одновременно записываются в кэш данных и кэш-память L2. Кроме того, кэш данных представляет собой слабо связанную модель памяти, и поэтому загрузка и сохранение данных в процессоре могут выполняться неупорядоченно. В общем случае сохранение выполняется упорядоченно.

Кэш PowerPC имеет два уровня: структуру кэша L1 и структуру кэша L2. Они дополняют системные кэши, такие как кэш L3 и системная память в некоторых системах.

В PowerPC размер строк в кэше L1 и L2 отличаются. Размер строки кэша L1 составляет 32 байта, кэша L2 - 128 байт. Поэтому все операции с кэшем выполняются таким образом, что кэш L1 оперирует со своими четырьмя строками независимо от кэша L2. Эта независимая работа кэша незаметна для пользователей и программистов.

Блок кэша данных (DCU) состоит из следующих трех субблоков: массивы кэша данных, управление кэшем данных (DCC) и буфер ассоциативной памяти данных (DTLB).

Субблок массивов содержит три массива: LRU, тегов и данных. Массивы тегов и данных - это стандартные SRAM. Массив LRU - это меньший по размеру массив двухпортовых регистров. И массив тегов, и массив данных представляют собой 4-канальный ассоциативный кэш и имеют конвейерный 2-цикловый доступ. DCU использует алгоритм замещения LRU, который использует 6-битовый вектор возраста в сочетании с фиксацией каналов для определения лучшего кандидата на замещение. DCU может получать с шины одновременно 256 битов данных чтения и может отправлять до 128 битов данных записи. Кэш данных неблокируемый, и когерентность кэша поддерживается при помощи интерфейса L2 в режиме сквозной записи.

Субблок DCC включает все элементы управления конвейера DCU, арбитраж кэша, SPR и конвейер слежения. Он управляет большей частью потока тракта данных и операциями.

DTLB - это 8-элементный полностью ассоциативный кэш, который использует EA для быстрого вычисления реального адреса. Доступ к нему осуществляется параллельно с остальными тремя массивами. При возникновении промаха DTLB делается запрос к UTLB, чтобы вычислить реальный адрес. DCC также обеспечивает выполнение инструкций управления кэшем данных PowerPC для предвыборки, очистки, инвалидации или обнуления строк кэша, или для групповой инвалидации всего кэша. Предоставляются также ресурсы для управления и отладки операций кэша данных.

Операции DCU

Кэш данных является неблокируемым кэшем и работает упорядоченно. Тем не менее, промахи загрузки могут обрабатываться вне очереди, поскольку задержки кэша L2 и памяти различаются. Завершение загрузки поддерживается упорядоченным, поскольку подтверждение (разрешение завершения) всех инструкций упорядочено.

Все операции загрузки и сохранения базируются на выравнивании операндов или на двухсловных границах. Поэтому операнды, пересекающие границу, будут дублировать операцию. Например, операнд создаст дополнительный цикл конвейера.

Операции загрузки

Инструкции загрузки, которые обращаются к страницам кэша L2 кэшируемой памяти и промахиваются в кэше данных, приводят к запросам чтения строки кэша, которые направляются к интерфейсу чтения данных PLB. Однако операции загрузки в страницы памяти с запрещенным кэшированием обращаются только к запрошенным байтам, в соответствии с типом инструкции. Этот доступ только к запрашиваемым байтам является требованием архитектуры. Тем не менее, DCU принудительно распространяет это требование на любую загрузку в страницу памяти с запрещенным кэшированием. Последующие операции загрузки кэша L2 в те же запрещенные для кэширования области вызовут новые запросы, направленные к интерфейсу чтения данных PLB. Данные из запрещенных для кэширования областей не используются повторно из буфера заполнения строки кэша данных (DCLFD).

DCU содержит четыре буфера DCLFD, и поэтому одновременно могут выполняться четыре независимых запроса заполнения строки кэша данных. DCU может продолжать обработку следующих обращений загрузки и сохранения во время заполнения этих строк.

DCU также содержит 8-элементную очередь промаха загрузки (LMQ), которая хранит до восьми ожидающих инструкций загрузки, которые либо промахнулись в кэше данных, либо обращались к страницам памяти с запрещенным кэшированием. Инструкция загрузки остается в LMQ до тех пор, пока запрашиваемые данные не попадут в буфер DCLFD. Данные доставляются в файл регистров, и инструкция удаляется из LMQ.

Операции сохранения

Обработка инструкций в DCU зависит от нескольких факторов, в том числе атрибутов запрещения кэширования (I), сквозной записи (W) и защиты (G), а также от того, разрешено ли кэширование промахов сохранение для строк кэша данных. Следует рассматривать три возможные ситуации:

Распределена ли уже строка кэша данных (если строка уже не находится в кэше данных).

Записаны ли данные непосредственно в память или только в кэш данных

Могут ли сохраняемые данные быть объединены с сохраняемыми данными предыдущих и последующих инструкций перед записью в память; сохранение данных является упорядоченным.

Объединение при сохранении

В общем случае операции записи в память вызываются разными инструкциями сохранения, которые указывают области со сквозной записью или с запретом кэширования, которые могут быть объединены в одно одновременное обращение к 16-байтным блокам памяти, хотя объединение при сохранении может быть запрещено установкой регистра CCR2.

Данная последовательность двух операций сохранения может быть объединена, если целевые байты находятся внутри одного выровненного счетверенного слова памяти независимо от того, являются ли они смежными. Следующие операции сохранения могут объединяться с уже объединенной последовательностью, если подчиняются тем же двум правилам (одно выровненное счетверенное слово с набором уже объединенных байтов). Например, последовательность трех операций сохранения слова по адресам 4, 8 и 0 может быть объединена, поскольку первые две являются смежными, а третья (сохранить слово по адресу 0) является смежной в отношении комбинации первых двух.

Дополнительное требование для объединения при сохранении применяется к страницам памяти с запрещенным кэшированием. В частности, данная операция сохранения на страницу памяти с запрещенным кэшированием может быть объединена с предыдущими операциями, если целевые байты данной операции сохранения не перекрываются с уже объединенными байтами. Другими словами, операция сохранения на странице с запрещенным кэшированием должно

одновременно удовлетворять условиям смежности и неперекрываемости с предыдущими операциями хранения, с которыми эта операция объединяется. Благодаря этому несколько операций записи, ассоциированные с последовательностью инструкций сохранения, каждая из которых обращается к общей области с запрещенным кэшированием, будут выполняться независимо в этой области.

И наконец, данная операция сохранения не объединяется с предыдущей операцией сохранения, если она отделена от предыдущей операции сохранения инструкциейmsync или mbar и если обе операции сохранения обращаются к странице памяти, которая одновременно защищена и запрещена к кэшированию.

Операции очистки строки

Поскольку в кэше данных выполняется сквозная запись, кэш L1 и кэш L2 всегда изменяются одновременно. Таким образом, в кэше данных нет грязных битов, и операции очистки не требуется.

Операции dcbi, dcbf и dcbz отправляются в кэш L2 с операндом RA, и кэш L2 действует соответствующим образом внутри кэша данных.

Ниже описываются операции очистки кэша данных:

dcbf	Кэш L2 очищает в память грязную строку, указанную RA, и инвалидирует строку кэша. Кэш L1 инвалидирует до четырех попадающих строк кэша той же самой строки кэша L2.
dcbi	Кэш L2 инвалидирует строку кэша, указанную RA. Кэш L1 инвалидирует до четырех попадающих строк кэша той же самой строки кэша L2.
dcbz	Кэш L2 записывает нули во всю строку кэша, указанную RA. Кэш L1 записывает нули до четырех попадающих строк кэша той же самой строки кэша L2.

Порядок доступа к памяти

В общем случае DCU может выполнять операции загрузки и сохранения не согласованно с потоком инструкций. То есть, обращения к памяти, ассоциированные с последовательностью инструкций загрузки и сохранения, могут быть выполнены в памяти в порядке, который отличается от порядка следования инструкций. Например, операции загрузки могут выполняться до более ранних операций сохранения, или операции сохранения могут выполняться до более ранних операций загрузки. Кроме того, более поздние операции загрузки и сохранения, которые попадают в кэш данных, могут выполняться до более ранних операций загрузки и сохранения, которые промахиваются в кэше данных.

DCU принудительно устанавливает требования модели последовательного исполнения ISA, в частности, что общий результат последовательности операций загрузки и сохранения должен быть таким же, какой предполагает порядок инструкций. Это означает, например, что если последующая операция загрузки считывает тот же адрес, который записан предыдущей операцией сохранения, DCU гарантирует, что при загрузке будут использованы данные, записанные операцией сохранения, а не более старые, предварительно сохраненные данные. Но подсистема памяти все еще может обнаружить доступ чтения, ассоциированный с еще более поздней операцией загрузки, прежде чем обнаружит доступ записи, ассоциированный с более ранней операцией сохранения.

Если DCU только что отправила запрос на интерфейс чтения данных кэша L2, и этот запрос конфликтует (то есть, обращается одному или нескольким одинаковым байтам) с более ранним запросом записи, отправленным на интерфейс записи данных кэша L2, то DCU останавливает запрос чтения на интерфейсе чтения данных кэша L2 до тех пор, пока запрос записи не будет подтвержден на интерфейсе записи кэша L2. Когда более ранний запрос записи подтвержден, запрос чтения выставляется, и подсистема кэша L2 должна обеспечить, чтобы данные, возвращаемые для запроса чтения, отражали значение данных, записанных операцией записи.

И наоборот, если запрос записи конфликтует с более ранним запросом чтения, DCU останавливает запрос записи до тех пор, пока запрос чтения не будет подтвержден.

Система PowerPC обеспечивает синхронизацию памяти, чтобы программа могла управлять порядком, в котором обращения к памяти ассоциируются с последовательностью выполняемых инструкций. Кроме того, в задействованных строках кэша L1 и кэша L2 должно быть инвалидировано слежение, если устройство ввода\вывода записывает данные через PLB6.

Когерентность кэша данных

Поскольку кэш данных PowerPC 476FP характеризуется сквозной записью, а кэш L2 write-back and inclusive кэша данных, все пересылки данных по PLB6 отслеживаются кэшем L2 и фильтруются для инвалидации слежения кэша данных L1. Это уменьшает траффик слежения для кэша данных L1 и повышает производительность кэша. Поскольку кэш L2 хранит состояние кэша, то кэш данных не поддерживает и не манипулирует многими протоколами когерентности системы.

Поэтому кэш данных скорее повышает производительность процессора, а не улучшает обслуживания системного уровня.

Управление и отладка кэша данных

Ядро PowerPC 476FP предоставляет различные регистры и инструкции для управления операциями с кэшем данных, а также для отладки кэша данных.

Инструкции управления и отладки кэша данных

Подробное описание инструкций, приведенных в данном разделе, см. в спецификации Power ISA Version 2.05.

В описаниях инструкций термин блок обозначает единицу хранения, с которой оперируют блоковые инструкции кэша. Для ядра PowerPC 476FP блок равнозначен строке кэша.

Программа использует следующие инструкции для управления кэшем данных.

Обнуление блока кэша данных (dcbz)

Инструкция обнуления блока кэша данных (dcbz) записывает нули в указанную строку кэша в обоих кэшах, L1 и L2. Поскольку размер строки кэша L2 больше размера строки кэша L1, то DCU повторяет инструкцию, так что запрос dcbz выполняется для каждой строки кэша L1, которая существует внутри кэша L2. Каждая повторяемая операция воспринимается как независимая инструкция внутри L-конвейера, и на L2 передается только первая инструкция dcbz повторяемой группы. Все повторяемые операции выполняют поиск в кэше данных L1, и каждая повторяемая операция, попавшая в кэш, записывает нули в строке кэша L1. Если операция промахивается, то ресурсы в L1 не изменяются, но операция точно так же отправляется на L2, как будто это первая повторяемая операция.

dcbz - это передача на L2, которая использует интерфейс памяти DCU. Она назначает элемент в очереди буфера памяти (SBQ) и генерирует запрос dcbz на L2, когда оказывается в начале очереди.

dcbz генерирует исключение типа управления доступом записи, если разрешение записи отсутствует.

Очистка фиксации блока кэша данных (dcblc)

Эта инструкция отменяет фиксацию строки кэша L1 или L2. DCU проверяет существование допустимой трансляции адреса и выполняет поиск в массиве тегов, чтобы определить, существует ли данный адрес в кэше. Если существует, бит фиксации очищается для совпадающего канала кэша, но только при подтвержденной инструкции. Операция очистки фиксации не инвалидирует строку кэша. Несмотря на то, что строка может существовать в обоих кэшах, L1 иL2, dcblc разблокирует только запрашиваемую строку в кэше, указанном в поле CT.

Если CT указывает только кэш L2, todcblc передается на кэш L2 при помощи интерфейса чтения DCU. Это требует назначения буфера заполнения строки и генерирует запрос без данных кэшу L2. После принятия запроса назначение буфера заполнения строки отменяется.

dcblc генерирует исключение типа управления доступом чтения, если разрешение чтения отсутствует.

Сохранение блока кэша данных (dcbst)

Инструкция dcbst очищает грязные данные из кэша L2. В сущности, для кэша L1 это фиктивная инструкция, поскольку кэш L1 допускает только сквозную запись и не может содержать грязные данные.

dcbst - это передача на кэш L2, которая использует интерфейс памяти DCU.

dcbst генерирует исключение типа управления доступом чтения, если разрешение чтения отсутствует.

Очистка блока кэша данных (dcbst)

В локальном режиме, описанном в спецификации Power ISA Version 2.05, инструкция dcbf инвалидирует строку в процессорах системы. Следует учитывать, что поддерживается только режим L = 0 (локальный).

Инструкция dcbf выталкивает грязные данные в указанную строку кэша L2 для записи их в память, а затем инвалидирует строку. Поскольку кэш L1 допускает только сквозную запись, в L1 нет грязных данных. Таким образом, dcbf в кэш L1 только инвалидирует строку в кэше L1. Очистка данных не требуется. Следует учесть, что несмотря на то, что указанная строка может быть запрещена для кэширования, она все равно должна быть найдена и очищена из кэша, если она существует.

Поскольку эта инструкция оперирует строкой кэша L2, и размер строки кэша L2 больше размера строки кэша L1, то DCU повторяет инструкцию, так что запрос dcbf выполняется для каждой строки кэша L1, которая существует внутри кэша L2. Каждая повторяемая операция воспринимается как независимая инструкция внутри L-конвейера, и на L2 передается только первая операция повторяемой группы. Все повторяемые операции выполняют поиск в кэше данных L1, и каждая повторяемая операция, попавшая в кэше, инвалидирует попавшую строку кэша. Если операция промахивается, то ресурсы в L1 не изменяются, но операция точно так же отправляется на кэш L2, как будто это первая повторяемая операция. dcbf не может инвалидировать строку в кэше L1, пока она не подтверждена.

dcbf в отношении фиксированной строки снимает фиксацию в кэше L1 и в кэше L2.

dcbf - это передача на кэш L2, которая использует интерфейс памяти DCU. Она назначает элемент в очереди SBQ и генерирует запрос без данных на кэш L2.

Инструкция dcbf генерирует исключение при одном из двух следующих условий:

Инструкция dcbf выполняется в режиме пользователя, и бит DULXE (MMUCR[4]) установлен.

Целевой блок dcbf не имеет разрешения чтения.

Инвалидация блока кэша данных (dcbst)

Инструкции dcbi и dcbf выполняют одну и ту же функцию для кэша L1 и кэша L2. То есть, dcbi выталкивает в память грязные данные в кэше L2. Несмотря на то, что у инструкции dcbi нет поля L, она работает в локальном режиме, как инструкция dcbf.

Инструкция dcbi инвалидирует строку в кэше L1 и кэше L2. L2 воспринимает dcbi как dcbf (то есть, инвалидирует строку и очищает данные). Несмотря на то, что указанная строка может быть запрещена для кэширования, она все равно должна быть найдена и инвалидирована в кэше, если существует.

Поскольку эта инструкция оперирует строкой кэша L2, и размер строки кэша L2 больше размера строки кэша L1, то DCU повторяет инструкцию, так что запрос dcbi выполняется для каждой строки кэша L1, которая существует внутри кэша L2. Каждая повторяемая операция воспринимается как независимая инструкция внутри L-конвейера, и на кэш L2 передается только первая операция повторяемой группы. Все повторяемые операции выполняют поиск в кэше данных L1, и каждая повторяемая операция, попавшая в кэше, инвалидирует попавшую строку

кэша. При промахе операции ресурсы в L1 не изменяются, но операция точно так же отправляется на кэш L2, как будто это первая повторяемая операция. dcbi не может инвалидировать строку в кэше L1, пока она не подтверждена.

dcbi в отношении фиксированной строки снимает фиксацию в кэше L1 и в кэше L2.

dcbi - это передача на кэш L2, которая использует интерфейс памяти DCU. Она назначает элемент в очереди SBQ и генерирует запрос без данных на кэш L2.

Инструкция dcbi генерирует исключение при одном из двух следующих условий:

Инструкция dcbi выполняется в режиме пользователя.

Целевой блок dcbi не имеет разрешения чтения.

Инвалидация кэша данных (dci)

Эта инструкция инвалидирует весь кэш данных L1 или L2, в зависимости от поля CT. Если она предназначена для кэша L2, DCU передает одну инструкцию dci на L2. Если она предназначена для кэша L1, DCU генерирует повторные запросы dci, пока не будет инвалидирован весь кэш данных. Каждая инструкция dci может инвалидировать две группы в кэше данных. Не последние повторные запросы требуют виртуального подтверждения, прежде чем строки кэша инвалидируются. Последний запрос требует обчыного подтверждения.

Инструкция dci может направляться только в кэш L1 или кэш L2, и поэтому она дает возможность нарушать инклюзивный характер взаимоотношений кэша L1 и кэша L2 посредством выполнения инструкции только для кэша L2. Другими словами, разрешается выполнять инструкцию dci для L1 и L2, именно в таком порядке, вслед за чем должна следовать инструкция sync.

В Power ISA определено, что программа должна помещать инструкцию msync после инструкции dci, чтобы гарантировать завершение dci до того, как будут выполняться следующие обращения к памяти данных. Однако для кэша L1 достаточно инструкции isync вместо msync.

Если кэш L2 является общим, то инвалидация кэша L2 требует, чтобы программа инвалидировала кэш инструкций L1 и кэш данных L1.

dci - это передача на кэш L2, которая использует интерфейс памяти DCU. Она назначает элемент в очереди SBQ и генерирует запрос без данных на кэш L2.

Инструкция dci генерирует исключение, если она не выполняется в режиме супревизора.

Предвыборка блока кэша данных (dcbt)

Эта инструкция дает подсказку, что к указанной строке возможно обращение в будущем. Она считается спекулятивной операцией и влияет либо на кэш L1, либо на кэш L2, в зависимости от поля CT. Оба типа инструкции dcbt посылают запрос на кэш L2, с той разницей, что запрос dcbt только на L2 требует, чтобы данные вернулись в L1. Запрос может быть отправлен на кэш L2 до подтверждения, потому что инструкция является всего лишь подсказкой.

Если поле CT указывает на L1, то DCU ищет в кэше L1 соответствующую строку. При попадании dcbt не выполняет никаких действий и не передается на кэш L2. При промахе инструкция назначает буфер заполнения строки (CT = L1) и отправляет запрос на данные с кэша L2.

Если поле CT указывает на L2, то поиск в кэше L1 не выполняется, и запрос на L2 посылается как dcbt без возврата данных.

dcbt - это передача на кэш L2, которая использует интерфейс чтения DCU. Она назначает буфер заполнения строки и генерирует запрос либо с данными, либо без данных, в зависимости от поля CT и попадания в кэше L1.

dcbt не генерирует исключение, если разрешение чтения отсутствует. В этом случае dcbt становится фиктивной и не оказывает воздействия на L1. Она также является фиктивной при обращении к защищенной странице или странице с запретом кэширования.

Предвыборка блока кэша данных с фиксацией (dcbtls)

Эта инструкция может либо загрузить и зафиксировать строку в кэше, либо зафиксировать строку, которая уже есть в кэше. Это не спекулятивная операция, и она влияет либо на кэш L1, либо на кэш L2, в зависимости от поля CT. Инструкция dcbtls назначает строку кэша при промахе, а затем фиксирует строку в кэше L1 или L2. Возможны несколько вариантов действий dcbtls:

Если CT = L1 и dcbtls промахивается в кэше L1, то назначается буфер заполнения строки, и строка в конечном счете пересылается в кэш L1 как фиксированная строка.

Если CT = L1, dcbtls попадает в кэше, и строка нефиксированная, то она фиксируется.

Если CT = L1, то dcbtls попадает в кэше, и строка фиксированная, то она считается фиктивной.

Если CT = L2, то назначается буфер заполнения строки, и на кэш L2 отправляется запрос dcbtls без данных. Это ничего не меняет в кэше L1.

dcbtls - это передача на кэш L2, которая использует интерфейс чтения DCU. Она назначает буфер заполнения строки и генерирует запрос либо с данными, либо без данных, в зависимости от поля CT и попадания в кэше L1.

Инструкция dcbtls генерирует исключение при одном из двух следующих условий:

Инструкция dcbtls выполняется в режиме пользователя.

Целевой блок dcbtls не имеет разрешения чтения.

При overlocked состоянии в кэше исключение не генерируется.

Примечание:

Уникальный сценарий может иметь место в кэше L1 при замещении фиксированной строки, даже если фиксация была установлена недавно. При назначении буфера заполнения строки канал в группе выбирается на основе текущего LRU и статуса фиксации. Следующая dcbtls может совпасть со строкой кэша, которую буфер заполнения строки в конечном счете заместит. При таком сценарии dcbtls может установить бит фиксации до того, как произойдет заполнение строки. Поэтом при заполнении строки элемент будет переписан, и фиксированная строка удалена. Для предотвращения этого сценария инструкции dcbtls должна предшествовать lwsync. Требования к этому сценарию следующие:

Кэшируемый буфер заполнения строки, который заместит канал X, существует, группа Y в кэше L1 существует.

Инструкция dcbtls попадает в L1, нефиксированная область канала X группы Y.

dcbtls устанавливает бит фиксации для канала X группы Y до заполнения линии.

Когда кэшируемый буфер заполнения строки готов выполнить заполнение линии, он замещает фиксированную строку в канале X группы Y.

Предвыборка блока кэша данных для операции сохранения (dcbtst)

Эта инструкция дает подсказку, что возможна запись указанной строки в будущем. Она считается спекулятивной операцией и влияет либо на кэш L1, либо на кэш L2, в зависимости от поля CT. Оба типа инструкции dcbtst посылают запрос на кэш L2, с той разницей, что запрос dcbtst только на L2 требует, чтобы данные вернулись в кэш L1. Запрос может быть отправлен на кэш L2 до подтверждения, потому что инструкция является всего лишь подсказкой.

Если поле CT указывает на L1, то DCU ищет в кэше L1 соответствующую строку. При попадании инструкция dcbtst не выполняет никаких действий и не передается на кэш L2. При промахе инструкция назначает буфер заполнения строки (CT = L1) и отправляет запрос на данные в кэш L2.

Если поле CT указывает на L2, то поиск в кэше L1 не выполняется, и запрос на L2 посылается как dcbtst без возврата данных.

dcbtst - это передача на кэш L2, которая использует интерфейс чтения DCU. Она назначает буфер заполнения строки и генерирует запрос либо с данными, либо без данных, в зависимости от поля CT и попадания в кэше L1.

dcbtst не генерирует исключение, если разрешение записи отсутствует. В этом случае инструкция dcbtst становится фиктивной и не оказывает воздействия на L1. Она также является фиктивной при обращении к защищенной странице или странице с запретом кэширования.

Предвыборка блока кэша данных для операции сохранения с фиксацией (dcbtstls)

Эта инструкция может либо загрузить и зафиксировать строку в кэше (в предположении, что в ближайшем будущем будет выполнено сохранение в указанную строку), либо зафиксировать строку, которая уже есть в кэше. Это не спекулятивная операция, и она влияет либо на кэш L1, либо на кэш L2, в зависимости от поля CT. Инструкция dcbtstls назначает строку кэша при промахе, а затем фиксирует строку в кэше L1 или L2. Возможны несколько вариантов действий инструкции dcbtstls:

Если CT = L1 и dcbtstls промахивается в кэше L1, то назначается буфер заполнения строки, и строка в конечном счете пересылается в кэш L1 как фиксированная строка.

Если CT = L1, dcbtstls попадает в кэше, и строка нефиксированная, то она фиксируется.

Если CT = L1, то dcbtstls попадает в кэше, и строка фиксированная, то строка считается фиктивной.

Если CT = L2, то назначается буфер заполнения строки, и на кэш L2 отправляется запрос dcbtstls без данных. Это ничего не меняет в кэше L1.

Инструкция dcbtstls генерирует исключение при одном из двух следующих условий:

Инструкция dcbtstls выполняется в режиме пользователя.

Целевой блок dcbtstls не имеет разрешения чтения.

При блокированном состоянии кэша исключение не генерируется.

dcbtstls - это передача на кэш L2, которая использует интерфейс чтения DCU. Она назначает буфер заполнения строки и генерирует запрос либо с данными, либо без данных, в зависимости от поля CT и попадания в кэше L1.

Примечание:

Уникальный сценарий может иметь место в кэше L1 при замещении фиксированной строки, даже если фиксация была установлена недавно. При назначении буфера заполнения строки канал в группе выбирается на основе текущего LRU и статуса фиксации. Следующая dcbtls может совпасть со строкой кэша, которую буфер заполнения строки в конечном счете заместит. При таком сценарии dcbtstls может установить бит фиксации до того, как произойдет заполнение строки. Поэтому при заполнении строки элемент будет переписан, и фиксированная строка удалена. Для предотвращения этого сценария инструкции dcbtstls должна предшествовать lwsync. Требования к этому сценарию следующие:

Кэшируемый буфер заполнения строки, который заместит канал X, существует, группа Y в кэше L1 существует.

Инструкция dcbtstls попадает в L1, нефиксированная область канала X группы Y.

dcbtstls устанавливает бит фиксации для канала X группы Y до заполнения линии.

Когда кэшируемый буфер заполнения строки готов выполнить заполнение линии, он замещает фиксированную строку в канале X группы Y.

Чтение кэша данных (dcread)

Инструкция dcread считывает содержание указанного физического местоположения в кэше данных и сохраняет данные в регистрах отладки. DCU не выполняет ни трансляцию адреса, ни обработку исключений для этой операции. Поскольку доступ осуществляется к содержанию конкретного местоположения в кэше, запрос dcread использует модифицированный формат для EA.

Биты адреса	Описание
0:16	Не используются.
17:18	Канал кэша данных.
19:26	Индекс кэша данных.
27:29	Адрес слова в строке кэша данных L1.
30:31	Не используются.

Инструкция dcread не направляется в кэш L2.

Инструкция dcread генерирует исключение, если не выполняется в режиме супревизора.

Инструкции барьера памяти

Процессор PowerPC 476FP предоставляет три типа инструкций барьера памяти: msync, mbar и lwsync.

Инструкция msync эквивалентна sync L = 0, lwsync эквивалентна sync L = 1, а mbar должна быть аналогична eiei, но в реализации PowerPC 476FP msync аналогична mbar. Однако для будущей совместимости рекомендуется при программировании использовать mbar для функциональности mbar. Другими словами, не следует заменять mbar инструкцией msync.

Синхронизация памяти (msync)

Инструкция msync аналогична инструкции sync L = 0 в спецификации Power ISA Version 2.05.

Инструкция msync блокирует выдачу всех последующих инструкций, пока IU не получит сигнал от L2, указывающий, что операция завершена. Инструкция должна выполняться на конвейере L, чтобы быть переданной в L2.

DCU отвечает только за подтверждение и передачу инструкции в L2 при помощи интерфейса памяти. Инструкция msync убедиться, что все операции загрузки и сохранения завершены, прежде чем посылать запрос на L2. Таким образом, она не может назначить строку в SBQ, пока LMQ, буфера заполнения строки и очередь попаданий памяти пусты. Она не должна ждать, пока опустеет SBQ. Поскольку SBQ используется для отправки msync в L2 и представляет собой очередь FIFO, все операции сохранения гарантированно покидают DCU до msync. msync должна быть подтверждена до того, как покинет SBQ, но может назначать элемент в SBQ до получения подтверждения.

Инструкция msync также выполняет функцию синхронизации системы и гарантирует завершение всех предшествующих операций.

Барьер памяти (mbar)

Инструкция mbar аналогична инструкции sync L = 0 в спецификации Power ISA Version 2.05.

Инструкция mbar блокирует выдачу всех последующих инструкций, пока IU не получит сигнал от L2, указывающий, что операция завершена. Инструкция должна выполняться на конвейере L, чтобы быть переданной в L2.

DCU отвечает только за подтверждение и передачу инструкции в L2 при помощи интерфейса памяти. Инструкция mbar должна убедиться, что все операции загрузки и сохранения завершены, прежде чем посылать запрос на L2. Таким образом, она не может назначить строку в SBQ, пока LMQ, буфера заполнения строки и очередь попаданий памяти пусты. Она не должна ждать, пока опустеет SBQ. Поскольку SBQ используется для отправки mbar в L2 и представляет собой очередь FIFO, все операции сохранения гарантированно покидают DCU до mbar. mbar должна быть подтверждена до того, как покинет SBQ, но может назначать элемент в SBQ до получения подтверждения.

Облегченная синхронизация (lwsync)

Инструкция lwsync аналогична инструкции sync L = 1 в спецификации Power ISA Version 2.05.

Инструкция lwsync блокирует выдачу всех последующих инструкций, пока операция не будет завершена. Операция считается завершенной, когда запрос подтверждается кэшем L2.

Регистры конфигурации ядра (CCR0, CCR1 и CCR2)

Регистры конфигурации ядра CCR0, CCR1 и CCR2 помогают выполнять отладку и управление функциями.

Операции dcbt и dcbtst

Инструкция dcbt обычно используется как подсказка процессору, что в ближайшем будущем исполняющаяся программа, скорее всего, обратится к определенному блоку данных. Процессор может начать заполнение этим блоком кэша данных, так чтобы, когда исполняющаяся

программа в конце концов выполнит загрузку из этого блока, он уже будет находиться в кэше, что повысит производительность.

Инструкция dcbtst обычно используется с той же целью, но конкретно для случаев, когда в ближайшем будущем исполняющаяся программа, скорее всего, выполнит сохранение блока. Разница между инструкциями dcbtst и dcbt важна только в системах с общей памятью и аппаратной поддержкой когерентности кэша. В таких системах инструкция dcbtst пытается определить блок в кэше данных таким образом, чтобы процессору впоследствии было удобнее записывать в этот блок.

По умолчанию инструкции dcbt и dcbtst игнорируются, если невозможно начать немедленное заполнение запрашиваемого блока кэша, а ожидание может привести к останову конвейера выполнения DCU. Например, инструкция cbt игнорируется, если все три буфера DCLFD уже используются, и выполнение следующих инструкций обращения к памяти задерживается.

Тем не менее, инструкции dcbt и dcbtst могут также использоваться как удобный механизм установки фиксированной, известной среды в кэше данных. Это полезно при определении содержимого для фиксации строки кэша, при детерминированной работе на определенной последовательности кода или для отладки аппаратной части низкого уровня и программ.

Поскольку ядро PowerPC 476FP поддерживает аппаратную когерентность, эти инструкции предвыборки не гарантируют операции, и поэтому целевая строка кэша может не ускориться при определенных условиях, например, при слежении, а конвейер DCU может останавливаться, как было описано выше.

Операция dcread

Инструкция dcread может быть использована для непосредственного чтения как информации тегов, так и указанного слова данных в указанном элементе кэша данных. Слово данных считывается в регистр GPR, указанный в коде инструкции. Информация тегов считывается в пару регистров SPR, DCDBTRH и DCDBTRL. Информация тегов впоследствии может быть перемещена в регистры GPR mfspr.

Выполнение инструкции dcread генерирует эквивалент EA, который затем используется для выбора конкретного слова данных из конкретной строки кэша.

Таблица Б.57 - Формат исполнительного адреса для icread и dcread

Биты адреса	Описание
0:16	Не используются.
17:18	Канал кэша.
19:26	Индекс кэша данных.
27:29	Адрес слова в строке кэша L1.
30:31	Не используются.

EA, генерируемый инструкцией dcread, должен быть выровнены по словам (то есть, EA[30:31] должен быть 0); в противном случае произойдет программная ошибка, и результат будет неопределенным.

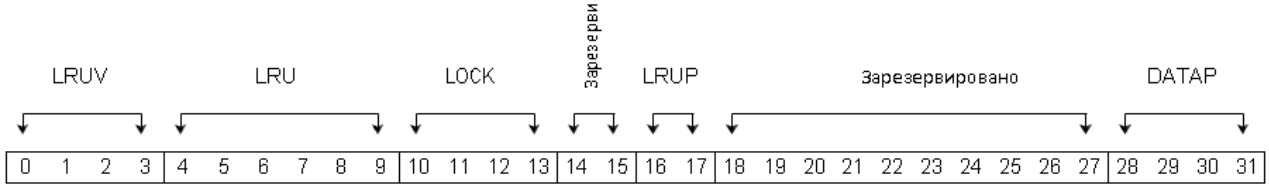
Если бит CCR0[CRPE] установлен, выполнение инструкции dcread также загружает информацию о четности в регистр DCDBTRL. Следует отметить, что поле DCDBTRL[DATAP], в отличие от всех остальных полей четности, загружает проверочные значения четности вместо непосредственных значений четности. То есть, поле DATAP всегда будет загружаться нулями, за исключением тех случаев, когда произошла ошибка четности, или его специально загрузили соответствующими битами из CCR1.

Выполнение инструкции dcread привилегированное и предназначено только для отладки.

Примечание: Использование инструкции dcread может не обеспечить корректную информацию, когда DCU находится в процессе выполнения операций с кэшем, связанных с ранее выполненными инструкциями, такими как заполнение строк и очистка строк. Кроме того, ядро PowerPC 476FP автоматически не синхронизирует контекст между инструкцией dcread и последующими инструкциями mfspr, которые считывают результаты инструкции dcread в регистры общего назначения (GPR). Чтобы гарантировать, что инструкция dcread выполняется корректно, и что инструкции mfspr получат результаты инструкции dcread, необходимо использовать последовательность инструкций, как в приведенном ниже примере:

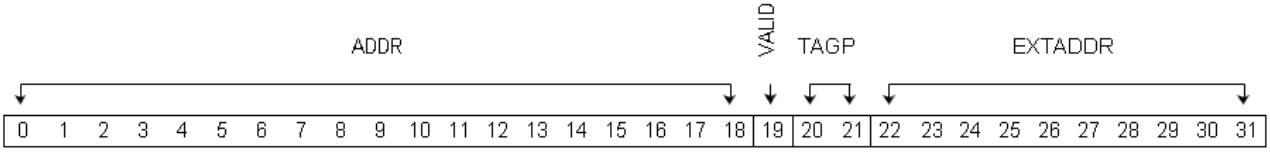
```
msync                   # Обеспечить завершение всех предыдущих операций с кэшем завершены.
dcread    regT,regA,regB # Чтение информации кэша; содержимое GPR A и GPR B
                         # складывается, и результат используется для указания индекса строки
                         кэша, который будет
                         # считываться. Слово данных пересылается в GPR T, и информация тега
                         # считывается в DCDBTRH и DCDBTRL.
isync                   # Обеспечить завершение dcread до попытки чтения. результатов.
mfdcdbtrh regD          # Переслать старшие разряды тега в GPR D.
mfdcdbtrl regE          # Переслать младшие разряды тега в GPR E.
```

Регистр тегов отладки кэша данных, младшие разряды (DCDBTRL)



Бит	Наименование	Описание
0:3	LRUV	Биты достоверности LRU. Один бит для каждого канала в группе.
4:9	LRU	Значение LRU для группы.
10:13	LOCK	Бит фиксации. Один бит для каждого канала в группе.
14:15	Зарезервировано	
16:17	LRUP	Четность LRU
18:27	Зарезервировано	
28:31	DATAP	Четность данных для слова, к которому обращаются

Регистр тегов отладки кэша данных, старшие разряды (DCDBTRH)



Биты	Наименование поля	Описание
0:18	ADDR	Адрес тега. RA[10:28].
19	VALID	Бит достоверности для этого элемента.
20:21	TAGP	Биты четности тега. Четность RA[0:28].
22:31	EXTADDR	Расширенный адрес тега. RA[0:9].

Операции четности кэша данных

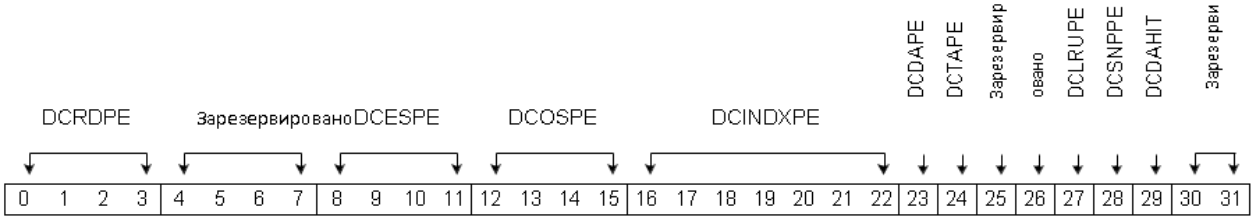
Кэш данных содержит биты четности и аппаратную часть обнаружения множества попаданий, чтобы защитить от программных ошибок данных. Теги кэша данных и данные защищены. Четность данных определяется побайтно; строка кэша содержит 258 битов данных и 32 бита четности. Четность тегов определяется на основе сохраненного действительного адреса, поскольку тегируется именно действительный адрес; биты RA от 0 до 28 имеют один бит четности. Кроме того, каждая строка кэша имеет один бит четности на шесть бит LRU, один бит четности на 4 бита достоверности и 4 бита фиксации.

При обнаружении ошибки четности и установленном MSR[ME] (то есть, разрешены прерывания проверки процессора), процессор передает управление блоку обработки прерываний проверки процессора. Как и в случае любого прерывания проверки процессора, после передачи управление блоку обработки прерываний проверки процессора MCSRR0 содержит значение самой старой незавершенной инструкции в конвейере в момент появления исключения, а

MCSRR1 содержит старый контекст MSR. Обработчик прерывания может запрашивать, установлен ли MCSR, MCSR[DC], чтобы определить произошла ли ошибка четности кэша данных, а затем либо инвалидировать кэш данных (с помощью dci), либо обратиться к системе, чтобы она завершила процесс или сбросила процессор. Блок обработки возвращается к прерванному процессу при помощи инструкции rfmci.

Регистр статуса исключения кэша данных (DCESR)

Регистр статуса исключения кэша данных (DCESR) предоставляет дополнительную информацию об ошибках четности кэша данных. Этот регистр сообщает, какая операция вызвала ошибку, какой интерфейс обнаружил ошибку, и какой индекс строки кэша поврежден. Он также указывает на обнаружение нескольких ошибок.



Бит	Наименование	Описание
0:3	DCRDPE	Ошибка четности интерфейса чтения кэша данных. Номер бита указывает, какое слово содержит ошибку на шине данных. Могут быть установлены несколько битов.
4:7	Зарезервировано	
8:11	DCESPE	Ошибка четности четной группы кэша данных. Это поле используется для ошибок четности массива тегов, и в нем могут быть установлены несколько битов. Ошибки могут сообщаться даже при запросе нечетной группы. 0:3 каналы в группе с (addr[19] XOR addr[26]) равным ‘0’
12:15	DCOSPE	Ошибка четности нечетной группы кэша данных. Могут быть установлены несколько бит. 0:3 каналы в группе с (addr[19] XOR addr[26]) равным ‘1’. Это поле используется для ошибок четности массива тегов, и в нем могут быть установлены несколько битов. Ошибки могут сообщаться даже при запросе четной группы.
16:22	DCINDXPE	Индекс ошибки четности в кэше. Отражает биты 20:26 действительного адреса. Бит19 может вычислен из полей DCESPE и DCOSPE.
23	DCDAPE	Ошибка четности массива данных кэша данных. Если установлен, данные имеют ошибку четности. Если причина запроса - промах, ошибка не сообщается.
24	DCTAPE	Ошибка четности массива тегов кэша данных. Если установлен, то как минимум один тег, ассоциированный либо с четной, либо с нечетной группой, имеет ошибку четности, местонахождение которой указывают поля DCESPE и DCOSPE.
25	Зарезервировано	
26	DCLRUPE	Ошибка четности LRU/достоверности/фиксации кэша данных. Ошибка четности произошла в поле четности или нечетности LRU/достоверности/фиксации запрашиваемой группы.
27	DCSNPPE	Ошибка четности слежения кэша данных. Ошибка четности на запросе отслеживания, полученном от кэша L2.
28	DCDAHIT	Попадание в массив данных кэша данных. При установке этот бит модифицирует DCDAPE. Если установлены и DCDAPE, и DCDAHIT, это указывает на ошибку четности данных на запросе загрузки, который попадает в кэш данных. Если установлен только DCDAPE, ошибка четности относится к запросу, который обслуживается из буферов заполнения строки. Если DCDAPE не установлен, этот бит следует игнорировать.

Бит	Наименование	Описание
29	DCDAAPU	APU массива данных кэша данных. При установке этот бит модифицирует DCDAPE. Если установлены и DCDAPE, и DCDAAPU, это указывает на ошибку четности данных на запросе загрузки для APU. Если установлен только DCDAPE, ошибка четности является следствием запроса CPU. Если DCDAPE не установлен, этот бит следует игнорировать.
30:31	Зарезервировано	

Если обработчик прерываний приведен в действие до того, как ошибка четности исказила состояние процессора, то процесс является обратимым, и обработчик прерывания может инвалидироать кэш данных и возобновить процесс. Чтобы гарантировать, что ошибки четности подлежат восстановлению, программа пользователя должна удовлетворять двум условиям. Во-первых, она должна маркировать кэшируемые страницы данных как страницы сквозной записи, а не с обратной записью. Во-вторых, устанавливаемый программно бит (CCR0[PRE]) должен быть установлен. Этот бит на один цикл останавливает все инструкции загрузки на последней стадии конвейера загрузки/сохранения, но только если требуется, чтобы ошибки четности подлежали восстановлению. Остановка конвейера гарантирует обнаружение любой ошибки четности. Таким образом результирующее прерывание проверки процессора принимается до завершения инструкции загрузки и искажения содержимого целевого GPR. Установка CCR0[PRE] снижает общую производительность приложения. Однако если состояние конвейера загрузки/сохранения таково, что инструкция загрузки останавливается на последней стадии по какой-то причине, не связанной с восстановлением четности, CCR0[PRE] не вызывает дополнительной остановки на один цикл.

Следует отметить, что прерывание проверки процессора типа исключения четности является асинхронным; то есть, адрес возврата в MCSRR0 не обязательно указывает на адрес инструкции, которая обнаружила ошибку четности в кэше данных. Прерывание проверки процессора принимается сразу же после обнаружения ошибки четности, и некоторые выполняемые инструкции могут быть очищены, а затем снова выполнены после прерывания, если процессор отрабатывает внешнее прерывание.

Симуляция четности кэша данных для отладки программы

Ошибки четности возникают в кэше редко и непредсказуемо. Поэтому поля CCR1[DCDPEI], CCR1[DCTPEI], CCR1[DCUPEI], CCR1[DCMPEI] и CCR1[FCOM] могут использоваться для симуляции влияния ошибки четности кэша данных, чтобы можно было проверить программу обработки прерываний.

39 бит четности кэша в каждой строке кэша содержат один бит четности на байт (то есть, 32 бита четности на строку длиной 32 байта) плюс следующие биты четности:

Два бита четности для тега адреса (бит достоверности (V) не включается в подсчет битов четности для тега)

Один бит четности для 4-битового поля U строки

Бит четности для каждого из четырех измененных (грязных) бит строки

Для тега данных предусмотрены два бита четности, поскольку четность вычисляется для двух перемежающихся битов поля тега, чтобы защититься от события, которое искажает два соседних бита. Другие биты данных физически чередуются таким образом, чтобы можно было использовать один бит четности для байта данных или другого поля. Все биты четности вычисляются и сохраняются, когда строка впервые помещается в кэш. Кроме того, данные и модифицированные (грязные) биты четности (но не тега и не биты четности пользователя) изменяются при изменении строки, в результате выполнения инструкции сохранения или dcbz.

Обычно для каждой группы битов, требующей защиты, вычисляется четность, которую проверяет аппаратная часть. Однако если установлен любой из бит CCR1[DCTPEI], вычисленная

четность для соответствующих бит тега инвертируется, и сохраняется как нечетность. Аналогичным образом если установлен любой из бит CCR1[DCUPEI], вычисленная четность для бит пользователя инвертируется, и сохраняется как нечетность. Точно так же, если установлен бит CCR1[DCDPEI], то четность для любых байтов данных, которые записаны либо в процессе заполнения строки, либо при выполнении инструкции сохранения, устанавливается как нечетность. Тогда при последующей загрузке данных, сохраненных с нечетностью, это вызывает прерывание проверки процессора типа исключения четности, и выполняется программа обработки прерываний. Приведенный ниже псевдокод представляет собой пример, в котором поле CCR1[DCDPEI используется для симуляции ошибки четности в байте 0 целевой строки кэша.

dcbt <адрес целевой строки> # Переместить целевую строку в кэш. msync # Ожидание dcbt.

mtspr CCR1, Rx # Установка CCR1[DCDPEI].

isync # Ожидание изменения контекста CCR1.

stb <адрес целевого байта># Сохранить данные в байте 0 целевой строки. msync # Ожидание, пока завершиться операция сохранения.

mtspr CCR1, Rz # Сброс CCR1[ICDPEI0].

isync # Ожидание изменения контекста CCR1. lb <byte 0 of target line> # Загруженный байт вызвает прерывание.

Если установлен бит CCR1[DCMPEI], то четность для любых модифицированных (грязных) битов, которые записаны либо в процессе заполнения строки, либо при выполнении инструкции сохранения dcbz, устанавливается как нечетность. Если кроме CCR1[DCMPEI] установлен также бит CCR1[FFF], то четность для всех четырех модифицированных (грязных) битов устанавливается как нечетность. Обращение сохранения к строке кэша, которая уже находится в кэше и в странице памяти, для которой установлен атрибут сквозной записи, не изменяет модифицированные (грязные) биты и модифицированные (грязные) биты четности. Таким образом, на эти обращения установка CCR1[DCMPEI] не влияет.

Бит CCR1[FCOM] позволяет симулировать ошибку четности множества попаданий. При его установке симулируется промах dcbt, что инициирует заполнение строки, даже если строка уже находится в кэше. Таким образом, этот бит позволяет одной и той же строке быть заполненной в кэш несколько раз, что генерирует ошибку четности множества попаданий, когда делается попытка чтения данных из этих строк кэша. Приведенный ниже псевдокод представляет собой пример, в котором поле CCR1[FCOM] используется для симуляции ошибки четности множества попаданий в кэше данных.

mtspr CCR0, Rx # Установить CCR0[GDCBT].

dcbt <целевой адрес строки> # Эта инструкция dcbt при необходимости заполняет первую копию целевой строки. msync # Ожидание, пока завершится заполнение.

mtspr CCR1, Ry # Установка CCR1[FCOM].

isync # Ожидание изменения контекста CCR1. dcbt <адрес целевой строки> # Заполнить вторую копию указанной строки. msync # Ожидание, пока завершится заполнение.

mtspr CCR1, Rz # Сбрс CCR1[FCOM].

isync # Ожидание изменения контекста CR1. br <байт 0 целевой строки> # Загруженный байт вызывает прерывание.

Приложение В

Описание регистра граничного сканирования

Таблица В.1 - Описание регистра граничного сканирования

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
0	TMODE	входная без мультиплексирования функционального пути
1	MCHKIN	входная
2	MCHKOUT	выходная
3	CLK_TMR_MPIC	входная без мультиплексирования функционального пути
4	DEBUG_HALT_N	входная
5	GPIO0_0	двунаправленная
6	GPIO0_0 OEN	управление направлением
7	GPIO0_1	двунаправленная
8	GPIO0_1 OEN	управление направлением
9	GPIO0_2	двунаправленная
10	GPIO0_2 OEN	управление направлением
11	GPIO0_3	двунаправленная
12	GPIO0_3 OEN	управление направлением
13	GPIO0_4	двунаправленная
14	GPIO0_4 OEN	управление направлением
15	GPIO0_5	двунаправленная
16	GPIO0_5 OEN	управление направлением
17	GPIO0_6	двунаправленная
18	GPIO0_6 OEN	управление направлением
19	GPIO0_7	двунаправленная
20	GPIO0_7 OEN	управление направлением
21	SPI0_SCLK	двунаправленная без мультиплексирования функционального пути
22	SPI0_SCLK OEN	управление направлением
23	SPI0_MOSI	двунаправленная
24	SPI0_MOSI OEN	управление направлением
25	SPI0_MISO	двунаправленная
26	SPI0_MISO OEN	управление направлением
27	SPI1_SCLK	двунаправленная без мультиплексирования функционального пути
28	SPI1_SCLK OEN	управление направлением
29	SPI1_MOSI	двунаправленная
30	SPI1_MOSI OEN	управление направлением
31	SPI1_MISO	двунаправленная
32	SPI1_MISO OEN	управление направлением
33	VIN0_FIELD	двунаправленная

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
34	VIN0_FIELD_OEN	управление направлением
35	VIN0_CLKI	входная без мультиплексирования функционального пути
36	VIN0_DE	двунаправленная
37	VIN0_DE_OEN	управление направлением
38	VIN0_VSYNC	двунаправленная
39	VIN0_VSYNC_OEN	управление направлением
40	VIN0_HSYNC	двунаправленная
41	VIN0_HSYNC_OEN	управление направлением
42	VIN0_D23	двунаправленная
43	VIN0_D23_OEN	управление направлением
44	VIN0_D22	двунаправленная
45	VIN0_D22_OEN	управление направлением
46	VIN0_D21	двунаправленная
47	VIN0_D21 OEN	управление направлением
48	VIN0_D20	двунаправленная
49	VIN0_D20 OEN	управление направлением
50	VIN0_D19	двунаправленная
51	VIN0_D19 OEN	управление направлением
52	VIN0_D18	двунаправленная
53	VIN0_D18_OEN	управление направлением
54	VIN0_D17	двунаправленная
55	VIN0_D17 OEN	управление направлением
56	VIN0_D16	двунаправленная
57	VIN0_D16_OEN	управление направлением
58	VIN0_D15	двунаправленная
59	VIN0_D15_OEN	управление направлением
60	VIN0_D14	двунаправленная
61	VIN0_D14_OEN	управление направлением
62	VIN0_D13	двунаправленная
63	VIN0_D13_OEN	управление направлением
64	VIN0_D12	двунаправленная
65	VIN0_D12_OEN	управление направлением
66	VIN0_D11	двунаправленная
67	VIN0_D11_OEN	управление направлением
68	VIN0_D10	двунаправленная
69	VIN0_D10_OEN	управление направлением
70	VIN0_D9	двунаправленная
71	VIN0_D9_OEN	управление направлением
72	VIN0_D8	двунаправленная

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
73	VIN0_D8_OEN	управление направлением
74	VIN0_D7	двунаправленная
75	VIN0_D7_OEN	управление направлением
76	VIN0_D6	двунаправленная
77	VIN0_D6 OEN	управление направлением
78	VIN0_D5	двунаправленная
79	VIN0_D5_OEN	управление направлением
80	VIN0_D4	двунаправленная
81	VIN0_D4_OEN	управление направлением
82	VIN0_D3	двунаправленная
83	VIN0_D3_OEN	управление направлением
84	VIN0_D2	двунаправленная
85	VIN0_D2_OEN	управление направлением
86	VIN0_D1	двунаправленная
87	VIN0_D1_OEN	управление направлением
88	VIN0_D0	двунаправленная
89	VIN0_D0_OEN	управление направлением
90	VIN1_FIELD	двунаправленная
91	VIN1_FIELD_OEN	управление направлением
92	VIN1_CLKI	входная без мультиплексирования функционального пути
93	VIN1_DE	двунаправленная
94	VIN1_DE_OEN	управление направлением
95	VIN1_VSYNC	двунаправленная
96	VIN1_VSYNC OEN	управление направлением
97	VIN1_HSYNC	двунаправленная
98	VIN1_HSYNC_OEN	управление направлением
99	VIN1_D23	двунаправленная
100	VIN1_D23_OEN	управление направлением
101	VIN1_D22	двунаправленная
102	VIN1_D22_OEN	управление направлением
103	VIN1_D21	двунаправленная
104	VIN1_D21_OEN	управление направлением
105	VIN1_D20	двунаправленная
106	VIN1_D20_OEN	управление направлением
107	VIN1_D19	двунаправленная
108	VIN1_D19_OEN	управление направлением
109	VIN1_D18	двунаправленная
110	VIN1_D18_OEN	управление направлением
111	VIN1_D17	двунаправленная
112	VIN1_D17_OEN	управление направлением

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
113	VIN1_D16	двунаправленная
114	VIN1_D16_OEN	управление направлением
115	VIN1_D15	двунаправленная
116	VIN1_D15_OEN	управление направлением
117	VIN1_D14	двунаправленная
118	VIN1_D14_OEN	управление направлением
119	VIN1_D13	двунаправленная
120	VIN1_D13_OEN	управление направлением
121	VIN1_D12	двунаправленная
122	VIN1_D12_OEN	управление направлением
123	VIN1_D11	двунаправленная
124	VIN1_D11_OEN	управление направлением
125	VIN1_D10	двунаправленная
126	VIN1_D10_OEN	управление направлением
127	VIN1_D9	двунаправленная
128	VIN1_D9_OEN	управление направлением
129	VIN1_D8	двунаправленная
130	VIN1_D8_OEN	управление направлением
131	VIN1_D7	двунаправленная
132	VIN1_D7_OEN	управление направлением
133	VIN1_D6	двунаправленная
134	VIN1_D6_OEN	управление направлением
135	VIN1_D5	двунаправленная
136	VIN1_D5_OEN	управление направлением
137	VIN1_D4	двунаправленная
138	VIN1_D4_OEN	управление направлением
139	VIN1_D3	двунаправленная
140	VIN1_D3_OEN	управление направлением
141	VIN1_D2	двунаправленная
142	VIN1_D2_OEN	управление направлением
143	VIN1_D1	двунаправленная
144	VIN1_D1_OEN	управление направлением
145	VIN1_D0	двунаправленная
146	VIN1_D0_OEN	управление направлением
147	CLK_AUD	входная без мультиплексирования функционального пути
148	CLK_CPU	входная без мультиплексирования функционального пути
149	CLK_VDU	входная без мультиплексирования функционального пути
150	CLK_DDR	входная без мультиплексирования функционального пути

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
151	MII_TX_CLK	двунаправленная без мультиплексирования функционального пути
152	MII_TX_CLK_OEN	управление направлением
153	MII_RX_CLK	двунаправленная без мультиплексирования функционального пути
154	MII_RX_CLK_OEN	управление направлением
155	MII_RXD0	двунаправленная
156	MII_RXD0_OEN	управление направлением
157	MII_RXD1	двунаправленная
158	MII_RXD1_OEN	управление направлением
159	MII_RXD2	двунаправленная
160	MII_RXD2_OEN	управление направлением
161	MII_RXD3	двунаправленная
162	MII_RXD3_OEN	управление направлением
163	MII_RX_DV	двунаправленная
164	MII_RX_DV_OEN	управление направлением
165	MII_RX_ER	двунаправленная
166	MII_RX_ER_OEN	управление направлением
167	MII_RX_COL	двунаправленная
168	MII_RX_COL_OEN	управление направлением
169	MII_RX_CRS	двунаправленная
170	MII_RX_CRS_OEN	управление направлением
171	MII_TXD0	двунаправленная
172	MII_TXD0_OEN	управление направлением
173	MII_TXD1	двунаправленная
174	MII_TXD1_OEN	управление направлением
175	MII_TXD2	двунаправленная
176	MII_TXD2_OEN	управление направлением
177	MII_TXD3	двунаправленная
178	MII_TXD3_OEN	управление направлением
179	MII_TX_EN	двунаправленная
180	MII_TX_EN_OEN	управление направлением
181	MII_TX_ER	двунаправленная
182	MII_TX_ER_OEN	управление направлением
183	MII_MDC	двунаправленная
184	MII_MDC_OEN	управление направлением
185	MII_MDIO	двунаправленная
186	MII_MDIO_OEN	управление направлением
187	VOUT1_D0	выходная
188	VOUT1_D1	выходная

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
189	VOUT1_D2	выходная
190	VOUT1_D3	выходная
191	VOUT1_D4	выходная
192	VOUT1_D5	выходная
193	VOUT1_D6	выходная
194	VOUT1_D7	выходная
195	VOUT1_D8	выходная
196	VOUT1_D9	выходная
197	VOUT1_D10	выходная
198	VOUT1_D11	выходная
199	VOUT1_D12	выходная
200	VOUT1_D13	выходная
201	VOUT1_D14	выходная
202	VOUT1_D15	выходная
203	VOUT1_D16	выходная
204	VOUT1_D17	выходная
205	VOUT1_D18	выходная
206	VOUT1_D19	выходная
207	VOUT1_D20	выходная
208	VOUT1_D21	выходная
209	VOUT1_D22	выходная
210	VOUT1_D23	выходная
211	VOUT1_HSYNC	выходная
212	VOUT1_VSYNC	выходная
213	VOUT1_DE	выходная
214	VOUT1_CLKO	выходная
215	VOUT1_FIELD	выходная
216	VOUT0_D0	выходная
217	VOUT0_D1	выходная
218	VOUT0_D2	выходная
219	VOUT0_D3	выходная
220	VOUT0_D4	выходная
221	VOUT0_D5	выходная
222	VOUT0_D6	выходная
223	VOUT0_D7	выходная
224	VOUT0_D8	выходная
225	VOUT0_D9	выходная
226	VOUT0_D10	выходная
227	VOUT0_D11	выходная
228	VOUT0_D12	выходная
229	VOUT0_D13	выходная

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
230	VOUT0_D14	выходная
231	VOUT0_D15	выходная
232	VOUT0_D16	выходная
233	VOUT0_D17	выходная
234	VOUT0_D18	выходная
235	VOUT0_D19	выходная
236	VOUT0_D20	выходная
237	VOUT0_D21	выходная
238	VOUT0_D22	выходная
239	VOUT0_D23	двунаправленная
240	VOUT0_D23_OEN	управление направлением
241	VOUT0_HSYNC	двунаправленная
242	VOUT0_HSYNC_OEN	управление направлением
243	VOUT0_VSYNC	двунаправленная
244	VOUT0_VSYNC_OEN	управление направлением
245	VOUT0_DE	двунаправленная
246	VOUT0_DE_OEN	управление направлением
247	VOUT0_CLKO	выходная
248	VOUT0_FIELD	выходная
249	SDIO_CLK	выходная
250	SDIO_CMD	двунаправленная
251	SDIO_CMD_OEN	управление направлением
252	SDIO_D0	двунаправленная
253	SDIO_D0_OEN	управление направлением
254	SDIO_D1	двунаправленная
255	SDIO_D1_OEN	управление направлением
256	SDIO_D2	двунаправленная
257	SDIO_D2_OEN	управление направлением
258	SDIO_D3	двунаправленная
259	SDIO_D3_OEN	управление направлением
260	UART0_RXD	двунаправленная
261	UART0_RXD_OEN	управление направлением
262	UART0_TXD	двунаправленная
263	UART0_TXD_OEN	управление направлением
264	UART1_RXD	двунаправленная
265	UART1_RXD_OEN	управление направлением
266	UART1_TXD	двунаправленная
267	UART1_TXD_OEN	управление направлением
268	UART2_RXD	двунаправленная
269	UART2_RXD_OEN	управление направлением
270	UART2_TXD	двунаправленная

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
271	UART2_TXD_OEN	управление направлением
272	LSCB0_TX_INH_B0	двунаправленная
273	LSCB0_TX_INH_B0 OEN	управление направлением
274	LSCB0_nRXB0	двунаправленная
275	LSCB0_nRXB0_OEN	управление направлением
276	LSCB0_RXB0	двунаправленная
277	LSCB0_RXB0_OEN	управление направлением
278	LSCB0_nTXB0	двунаправленная
279	LSCB0_nTXB0_OEN	управление направлением
280	LSCB0_TXB0	двунаправленная
281	LSCB0_TXB0_OEN	управление направлением
282	LSCB0_TX_INH_A0	двунаправленная
283	LSCB0_TX_INH_A0 OEN	управление направлением
284	LSCB0_nRXA0	двунаправленная
285	LSCB0_nRXA0_OEN	управление направлением
286	LSCB0_RXA0	двунаправленная
287	LSCB0_RXA0_OEN	управление направлением
288	LSCB0_nTXA0	двунаправленная
289	LSCB0_nTXA0_OEN	управление направлением
290	LSCB0_TXA0	двунаправленная
291	LSCB0_TXA0_OEN	управление направлением
292	LSCB0_CLK_IN_0	двунаправленная без мультиплексирования функционального пути
293	LSCB0_CLK_IN_0_OEN	управление направлением
294	LSCB1_TX_INH_B1	двунаправленная
295	LSCB1_TX_INH_B1 OEN	управление направлением
296	LSCB1_nRXB1	двунаправленная
297	LSCB1_nRXB1_OEN	управление направлением
298	LSCB1_RXB1	двунаправленная
299	LSCB1_RXB1_OEN	управление направлением
300	LSCB1_nTXB1	двунаправленная
301	LSCB1_nTXB1_OEN	управление направлением
302	LSCB1_TXB1	двунаправленная
303	LSCB1_TXB1_OEN	управление направлением
304	LSCB1_TX_INH_A1	двунаправленная
305	LSCB1_TX_INH_A1 OEN	управление направлением
306	LSCB1_nRXA1	двунаправленная
307	LSCB1_nRXA1_OEN	управление направлением
308	LSCB1_RXA1	двунаправленная
309	LSCB1_RXA1_OEN	управление направлением

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
310	LSCB1_nTXA1	двунаправленная
311	LSCB1_nTXA1_OEN	управление направлением
312	LSCB1_TXA1	двунаправленная
313	LSCB1_TXA1_OEN	управление направлением
314	LSCB1_CLK_IN_1	двунаправленная без мультиплексирования функционального пути
315	LSCB1_CLK_IN_1_OEN	управление направлением
316	SD_REFCLK1	входная без мультиплексирования функционального пути
317	SD_REFCLK0	входная без мультиплексирования функционального пути
318	USB_CLK	входная без мультиплексирования функционального пути
319	EXT_MEM_CE0	двунаправленная без мультиплексирования функционального пути
320	EXT_MEM_CE0_OEN	управление направлением
321	EXT_MEM_CE1	двунаправленная
322	EXT_MEM_CE1_OEN	управление направлением
323	EXT_MEM_CE2	двунаправленная без мультиплексирования функционального пути
324	EXT_MEM_CE2_OEN	управление направлением
325	EXT_MEM_CE3	двунаправленная
326	EXT_MEM_CE3_OEN	управление направлением
327	EXT_MEM_CE4	двунаправленная
328	EXT_MEM_CE4_OEN	управление направлением
329	EXT_MEM_CE5	двунаправленная
330	EXT_MEM_CE5_OEN	управление направлением
331	EXT_MEM_A0	двунаправленная
332	EXT_MEM_A0_OEN	управление направлением
333	EXT_MEM_A1	двунаправленная
334	EXT_MEM_A1_OEN	управление направлением
335	EXT_MEM_A2	двунаправленная без мультиплексирования функционального пути
336	EXT_MEM_A2_OEN	управление направлением
337	EXT_MEM_A3	двунаправленная без мультиплексирования функционального пути
338	EXT_MEM_A3_OEN	управление направлением
339	EXT_MEM_A4	двунаправленная
340	EXT_MEM_A4_OEN	управление направлением

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
341	EXT_MEM_A5	двунаправленная
342	EXT_MEM_A5_OEN	управление направлением
343	EXT_MEM_A6	двунаправленная
344	EXT_MEM_A6_OEN	управление направлением
345	EXT_MEM_A7	двунаправленная
346	EXT_MEM_A7_OEN	управление направлением
347	EXT_MEM_A8	двунаправленная без мультиплексирования функционального пути
348	EXT_MEM_A8_OEN	управление направлением
349	EXT_MEM_A9	двунаправленная
350	EXT_MEM_A9_OEN	управление направлением
351	EXT_MEM_A10	двунаправленная без мультиплексирования функционального пути
352	EXT_MEM_A10_OEN	управление направлением
353	EXT_MEM_A11	двунаправленная
354	EXT_MEM_A11_OEN	управление направлением
355	EXT_MEM_A12	двунаправленная
356	EXT_MEM_A12_OEN	управление направлением
357	EXT_MEM_A13	двунаправленная
358	EXT_MEM_A13_OEN	управление направлением
359	EXT_MEM_A14	двунаправленная
360	EXT_MEM_A14_OEN	управление направлением
361	EXT_MEM_A15	двунаправленная
362	EXT_MEM_A15_OEN	управление направлением
363	EXT_MEM_A16	двунаправленная
364	EXT_MEM_A16_OEN	управление направлением
365	EXT_MEM_A17	двунаправленная
366	EXT_MEM_A17_OEN	управление направлением
367	EXT_MEM_A18	двунаправленная
368	EXT_MEM_A18_OEN	управление направлением
369	EXT_MEM_A19	двунаправленная
370	EXT_MEM_A19_OEN	управление направлением
371	EXT_MEM_A20	двунаправленная
372	EXT_MEM_A20_OEN	управление направлением
373	EXT_MEM_A21	двунаправленная
374	EXT_MEM_A21_OEN	управление направлением
375	EXT_MEM_A22	двунаправленная
376	EXT_MEM_A22_OEN	управление направлением
377	EXT_MEM_A23	двунаправленная без мультиплексирования функционального пути

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
378	EXT_MEM_A23_OEN	управление направлением
379	EXT_MEM_A24	двунаправленная
380	EXT_MEM_A24_OEN	управление направлением
381	EXT_MEM_A25	двунаправленная
382	EXT_MEM_A25_OEN	управление направлением
383	EXT_MEM_WE	двунаправленная
384	EXT_MEM_WE_OEN	управление направлением
385	EXT_MEM_OE	двунаправленная
386	EXT_MEM_OE_OEN	управление направлением
387	EXT_MEM_BLE_L	двунаправленная
388	EXT_MEM_BLE_L_OEN	управление направлением
389	EXT_MEM_BLE_H	двунаправленная
390	EXT_MEM_BLE_H_OEN	управление направлением
391	EXT_MEM_BHE_L	двунаправленная
392	EXT_MEM_BHE_L_OEN	управление направлением
393	EXT_MEM_BHE_H	двунаправленная
394	EXT_MEM_BHE_H_OEN	управление направлением
395	EXT_MEM_D0	двунаправленная
396	EXT_MEM_D0_OEN	управление направлением
397	EXT_MEM_D1	двунаправленная
398	EXT_MEM_D1_OEN	управление направлением
399	EXT_MEM_D2	двунаправленная
400	EXT_MEM_D2_OEN	управление направлением
401	EXT_MEM_D3	двунаправленная
402	EXT_MEM_D3_OEN	управление направлением
403	EXT_MEM_D4	двунаправленная
404	EXT_MEM_D4_OEN	управление направлением
405	EXT_MEM_D5	двунаправленная
406	EXT_MEM_D5_OEN	управление направлением
407	EXT_MEM_D6	двунаправленная без мультиплексирования функционального пути
408	EXT_MEM_D6_OEN	управление направлением
409	EXT_MEM_D7	двунаправленная
410	EXT_MEM_D7_OEN	управление направлением
411	EXT_MEM_D8	двунаправленная
412	EXT_MEM_D8_OEN	управление направлением
413	EXT_MEM_D9	двунаправленная
414	EXT_MEM_D9_OEN	управление направлением
415	EXT_MEM_D10	двунаправленная
416	EXT_MEM_D10_OEN	управление направлением

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
417	EXT_MEM_D11	двунаправленная
418	EXT_MEM_D11_OEN	управление направлением
419	EXT_MEM_D12	двунаправленная без мультиплексирования функционального пути
420	EXT_MEM_D12_OEN	управление направлением
421	EXT_MEM_D13	двунаправленная без мультиплексирования функционального пути
422	EXT_MEM_D13_OEN	управление направлением
423	EXT_MEM_D14	двунаправленная
424	EXT_MEM_D14_OEN	управление направлением
425	EXT_MEM_D15	двунаправленная
426	EXT_MEM_D15_OEN	управление направлением
427	EXT_MEM_D16	двунаправленная
428	EXT_MEM_D16_OEN	управление направлением
429	EXT_MEM_D17	двунаправленная
430	EXT_MEM_D17_OEN	управление направлением
431	EXT_MEM_D18	двунаправленная
432	EXT_MEM_D18_OEN	управление направлением
433	EXT_MEM_D19	двунаправленная
434	EXT_MEM_D19_OEN	управление направлением
435	EXT_MEM_D20	двунаправленная
436	EXT_MEM_D20_OEN	управление направлением
437	EXT_MEM_D21	двунаправленная
438	EXT_MEM_D21_OEN	управление направлением
439	EXT_MEM_D22	двунаправленная
440	EXT_MEM_D22_OEN	управление направлением
441	EXT_MEM_D23	двунаправленная
442	EXT_MEM_D23_OEN	управление направлением
443	EXT_MEM_D24	двунаправленная
444	EXT_MEM_D24_OEN	управление направлением
445	EXT_MEM_D25	двунаправленная без мультиплексирования функционального пути
446	EXT_MEM_D25_OEN	управление направлением
447	EXT_MEM_D26	двунаправленная без мультиплексирования функционального пути
448	EXT_MEM_D26_OEN	управление направлением
449	EXT_MEM_D27	двунаправленная без мультиплексирования функционального пути
450	EXT_MEM_D27_OEN	управление направлением

Продолжение таблицы В.1

Номер бита	Соответствующая ячейка граничного сканирования	Тип ячейки
451	EXT_MEM_D28	двунаправленная
452	EXT_MEM_D28_OEN	управление направлением
453	EXT_MEM_D29	двунаправленная
454	EXT_MEM_D29_OEN	управление направлением
455	EXT_MEM_D30	двунаправленная
456	EXT_MEM_D30_OEN	управление направлением
457	EXT_MEM_D31	двунаправленная
458	EXT_MEM_D31_OEN	управление направлением
459	EXT_MEM_D32	двунаправленная
460	EXT_MEM_D32_OEN	управление направлением
461	EXT_MEM_D33	двунаправленная
462	EXT_MEM_D33_OEN	управление направлением
463	EXT_MEM_D34	двунаправленная
464	EXT_MEM_D34_OEN	управление направлением
465	EXT_MEM_D35	двунаправленная
466	EXT_MEM_D35_OEN	управление направлением
467	EXT_MEM_D36	двунаправленная
468	EXT_MEM_D36_OEN	управление направлением
469	EXT_MEM_D37	двунаправленная
470	EXT_MEM_D37_OEN	управление направлением
471	EXT_MEM_D38	двунаправленная
472	EXT_MEM_D38_OEN	управление направлением
473	EXT_MEM_D39	двунаправленная
474	EXT_MEM_D39_OEN	управление направлением
475	AUD_SCLK	выходная
476	AUD_WS	выходная
477	SPDIF	выходная
478	AUD_D0	выходная
479	AUD_D1	выходная
480	AUD_D2	выходная
481	AUD_D3	выходная
482	GPIO1_0	двунаправленная
483	GPIO1_0_OEN	управление направлением
484	GPIO1_1	двунаправленная
485	GPIO1_1_OEN	управление направлением
486	GPIO1_2	двунаправленная
487	GPIO1_2_OEN	управление направлением
488	GPIO1_3	двунаправленная
489	GPIO1_3_OEN	управление направлением
490	GPIO1_4	двунаправленная
491	GPIO1_4_OEN	управление направлением

Продолжение таблицы В.1

Номер бита	**Соответствующая ячейка граничного сканирования**	**Тип ячейки**
492	GPIO1_5	двунаправленная
493	GPIO1_5_OEN	управление направлением
494	GPIO1_6	двунаправленная
495	GPIO1_6_OEN	управление направлением
496	GPIO1_7	двунаправленная
497	GPIO1_7_OEN	управление направлением

Лист регистрации изменений

Изм.	Номера страниц				Всего страниц в докумете	№ документа	Входящий № сопроводительного докум. и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					
2		все					ЮФКВ.210-2019	Харламова	**24.04.19**

					ЮФКВ.431268.017РЭ	Лист 1663
Изм	Лист	№ докум.	Подп.	Дата		

Инв.№подл.	Подп. и дата	Взам.инв.№	Инв.№дубл.	Подп. и дата
34992-2	13.05.2019	34992-1		