УТВЕРЖДЕН

ЮФКВ.431268.008РЭ-ЛУ

Микросхема интегральная К1888ВС018
Руководство по эксплуатации
ЮФКВ.431268.008РЭ

Инв.№подл.	Подп. и дата	Взам.инв.№	Инв.№дубл.	Подп. и дата	Справ.№
26424-3	23.12.2015	26424-2			

Содержание

1 Назначение и область применения 7
2 Технические характеристики 8
2.1 Основные функциональные характеристики СБИС СНП-ВП 8
3 Общая структура СБИС СНП-ВП 10
3.1 Схема синхронизации 12
3.2 Карта памяти СБИС СНП-ВП 14
4 Блок цифровой обработки сигналов на основе процессорных ядер с архитектурой NeuroMatrix (NMB) 16
4.1 Процессорная система NMU на базе DSP ядра NMC3 16
4.2 Интервальные таймеры TIMER0 и TIMER1 19
4.3 Системный контроллер процессорной системы NMU (NMCSC) 21
4.4 Старт и остановка выборки команд ядром NMC3 23
4.5 Контроллер прерываний (INTC) 25
4.6 PU (Permutation Unit) – блок упаковки/распаковки векторов данных 28
4.6.1 Аппаратно поддерживаемые операции 28
4.6.2 Режимы и порядок работы PU 32
4.6.3 Конфигурационные регистры 33
4.7 Система общей памяти (SMU) 40
4.8 Список периферийных регистров для ядра NMC3 40
5 Процессорная система на базе ядра ARM11 (ARMU). 42
5.1 Микропроцессорная сборка на базе ARM1176JZF 46
5.1.1 Контроллер кэш-памяти второго уровня L220 48
5.1.2 Интерфейс ПДП и «ближняя» память ядра (TCM) 48
5.1.3 Блоки поддержки отладки — ETM11, ETB11 и JTAG интерфейс 50
5.2 Системный коммутатор шин HPMS 55
5.2.1 Параметры портов коммутатора HPMS 56
5.2.2 Регистровая модель коммутатора HPMS 59
5.3 Периферийный коммутатор шин HPMP 63
5.3.1 Параметры портов коммутатора HPMP 65
5.3.2 Регистровая модель коммутатора HPMP 65
5.4 Блок управления сигналами сброса и синхросигналами CRGS 69
5.4.1 Устройство блока CRGS 69
5.4.2 Генератор сигналов сброса 71
5.4.3 Генератор синхросигналов 73
5.4.4 Регистровая модель блока CRGS 80
5.5 Блок управления сигналами сброса и синхросигналами CRGM 88
5.5.1 Устройство блока CRGM 88
5.6 Системные векторные контроллеры прерываний VIC0 и VIC1 89
5.6.1 Устройство векторных контроллеров прерываний 89
5.6.2 Подробности работы с контроллером VIC 92
5.6.3 Последовательность шагов при обработке прерываний 94
5.6.4 Регистровая модель системного контроллера прерываний VIC 95

5.7 Контроллер внешних прерываний EIRC 104
5.7.1 Устройство контроллера внешних прерываний 104
5.7.2 Регистровая модель контроллера внешних прерываний EIRC 105
5.7.3 Прерывания 106
5.8 Блок сдвоенных таймеров DIT 107
5.8.1 Устройство блока сдвоенных таймеров 107
5.8.2 Регистровая модель блока сдвоенных таймеров 108
5.8.3 Прерывания 113
5.9 Сторожевой таймер WDOG 114
5.9.1 Устройство блока сторожевого таймера 114
5.9.2 Регистровая модель блока сторожевого таймера 115
5.9.3 Прерывания 119
5.10 Таймер реального времени RTC 120
5.10.1 Устройство блока таймера реального времени 120
5.10.2 Спецификация регистров. 120
5.10.3 Описание регистров 121
5.10.4 Прерывания 126
5.11 Системный контроллер ARMSC 127
5.11.1 Устройство системного контроллера ARMSC 127
5.11.2 Регистровая модель контроллера ARMSC 128
5.11.3 Прерывания 133
5.12 Блок управления питанием PWRC 134
5.12.1 Программный механизм включения и выключения питания 135
5.12.2 Автоматический механизм включения и выключения питания 135
5.12.3 Программно доступные регистры блока PWRC 137
5.13 Контроллер ПДП XDMAC 142
5.13.1 Устройство контроллера ПДП XDMAC 142
5.13.2 Программно доступные регистры контроллера ПДП 143
5.13.3 Включение и выключение контроллера ПДП 146
5.13.4 Процедура настройки канала 146
5.14 Контроллер асинхронного последовательного интерфейса UART 148
5.14.1 Устройство UART портов СБИС СНП-ВП 148
5.14.2 Описание функционирования UART порта 150
5.14.3 Регистровая модель контроллера порта UART 156
5.14.4 Прерывания 169
5.15 Контроллер последовательного синхронного интерфейса SPI 171
5.15.1 Устройство SPI порта СБИС СНП-ВП 171
5.15.2 Протокол передачи данных в различных режимах SPI порта 172
5.15.3 Регистровая модель контроллера порта SPI 175
5.15.4 Прерывания 182
5.16 Контроллер шины USB2.0 183
5.16.1 Устройство контроллера USB 183
5.16.2 Регистровая модель контроллера USB 184
5.16.3 Рекомендации по подключению интерфейса USB 214
5.17 Блок портов общего назначения GPIO 216
5.17.1 Устройство портов общего назначения 216
5.17.2 Регистровая модель контроллера портов GPIO 216
5.17.3 Прерывания 217
5.18 Контроллер интерфейса CAN 218
5.18.1 Устройство контроллера шины CAN 218

5.18.2 Протокол шины CAN 219
5.18.3 Принципы работы CAN контроллера 220
5.18.4 Программная модель контроллера CAN 226
5.19 Контроллер интерфейса I2C 245
5.19.1 Устройство контроллера интерфейса I2C 245
5.19.2 Программная модель контроллера интерфейса I2C 247
5.19.3 Функционирование контроллера интерфейса I2C 256
5.20 Контроллер интерфейса ETHERNET 264
5.20.1 Принципы работы контроллера GRETH 264
5.20.2 Программные драйверы 272
5.20.3 Программная модель контроллера GRETH 272
5.21 Контроллер внешней динамической памяти LPDDR2 278
5.21.1 Отображение адресов 278
5.21.2 Регистровая модель контроллера EMIC 279
5.21.3 Регистровая модель контроллера физического уровня PHYC 303
5.21.4 Стандартный алгоритм инициализации контроллеров 310
5.21.5 Режимы пониженного потребления 311
5.21.6 Управление режимами пониженного потребления 313
5.21.7 Проверка правильности адреса 314
5.21.8 Взаимодействие с регистрами внешней памяти 315
5.22 Загрузочное ПЗУ BROM 316
5.22.1 Алгоритм работы начального загрузчика 317
5.22.2 Состояние процессора перед началом исполнения загруженной программы 318
5.22.3 Начальная загрузка через порт SPI 320
5.22.4 Начальная загрузка через порт UART 321
5.22.5 Начальная загрузка через порт Ethernet 323
5.22.6 Использование GPIO для индикации состояния начального загрузчика 323
6 Блок аналого-цифрового преобразования (ADCU) 324
7 Блок аппаратных каналов преобразования частоты DDC 325
7.1 Аппаратный канал, осуществляющий обработку оцифрованных данных (канал 0... канал 7) 326
7.1.1 Входной коммутатор канала 327
7.1.2 Схема формирования синуса и косинуса (гетеродин) 328
7.1.3 Генератор Т-кодов 329
7.1.4 Комплексный смеситель 331
7.1.5 Дециматор HC1 и HC2 332
7.1.6 Блоки нормализации H1 и H2 332
7.1.7 Блок квадратора 333
7.1.8 Округление результатов вычислений в каналах 333
7.1.9 Схема накопления и упаковки СНУ 333
7.2 Аппаратный канал с КИХ-фильтром, осуществляющий обработку данных из памяти (канал 12) 335
7.2.1 Распаковщик 335
7.2.2 Схема формирования синуса и косинуса (гетеродин) 336
7.2.3 Генератор Т-кодов 337
7.2.4 Комплексный смеситель 337
7.2.5 Блок дециматора HC1 338
7.2.6 Блоки нормализации H1 и H2 338
7.2.7 Блок КИХ-фильтра 338
7.2.8 Схема накопления и упаковки СНУ 339

7.3 Прерывания блока DDC 340
7.4 Конфигурационные регистры блока DDC 341
7.4.1 Регистры общего назначения 343
7.4.2 Регистры управления каналами, осуществляющими обработку оцифрованных данных (канал 0, ..., канал 7) 349
7.4.3 Регистры управления аппаратного канала с КИХ-фильтром (канал 12) 360
8 Блок аппаратных корреляторов CORB 364
8.1 Аппаратный канал CH 365
8.1.1 Гетеродин (GTRDN) 366
8.1.2 Генератор псевдослучайных последовательностей (PRNG) 367
8.2 Схема приоритетов и анализа зависимых каналов (SPDP) 370
8.3 Программная модель CORB 371
8.3.1 Регистры общего назначения 373
8.3.2 Регистры канала для обращения за данными 375
8.3.3 Регистры канала для управления гетеродином 376
8.3.4 Регистры канала для управления генератором времени универсального генератора псевдослучайных последовательностей 378
8.3.5 Регистры канала для табличного формирования ПСП 380
8.3.6 Регистры канала для формирования ПСП на сдвиговых регистрах 381
8.3.7 Регистры канала, содержащие результат корреляции P_RE, P_IM, E_RE, E_IM, EE_RE, EE_IM, L_RE, L_IM, LL_RE, LL_IM 383
9 Интервальный таймер для формирования временных интервалов и синхронизации с внешним событием (ITU) 384
9.1 Принцип работы блока ITU 384
9.2 Конфигурационные регистры блока ITU 387
10 Генератор дальномерного кода для калибровки (CALB) 390
10.1 Конфигурационные регистры блока CALB 390
11 Контроллер ПДП (DMAC) 392
11.1 Канал DDC-память 392
11.2 Канал память-DDC 393
11.3 Канал память-COR 393
11.4 Канал память-память 393
11.5 Каналы CP0-память, память-CP0, CP1-память и память-CP1 393
11.6 Пакетные запросы 394
11.7 Программная модель контроллера ПДП 394
11.7.1 Регистр управления каналом (CSR) 395
11.7.2 Регистр маски запроса на прерывание (MASK) 397
11.7.3 Регистр-счётчик данных (CNT) 397
11.7.4 Регистр начального адреса чтения (RDADDR) 397
11.7.5 Регистр начального адреса записи (WRADDR) 397
11.7.6 Регистр состояния блока ПДП (CORE_STATUS) 397
11.8 Прерывания 398
11.9 Байтовый коммуникационный порт 400
11.9.1 Структурная схема интерфейсной части контроллера ПДП и коммуникационных портов 400
11.9.2 Внешние выводы коммуникационного порта 401
11.9.3 Организация обмена данными по коммуникационному порту 402
11.9.4 Арбитраж шины коммуникационного порта 403
12 Электрические и конструктивные характеристики СБИС СНП-ВП 406
12.1 Состав и расположение внешних выводов СБИС СНП-ВП 406

12.2 Конструктивные характеристики....413
12.3 Электрические характеристики....414
12.4 Порядок подачи питания при включении....417
13 Рекомендации по включению микросхемы....418
13.1 Рекомендации по включению АЦП....418
13.2 Рекомендации по проектированию печатной платы для USB интерфейса....420
13.3 Рекомендации по проектированию печатной платы для LPDDR2 интерфейса....422
13.4 Рекомендации по реализации схемы начального сброса....425
13.5 Рекомендации по включению коммуникационных портов....426

1 Назначение и область применения

Специализированная большая интегральная микросхема К1888ВС018 сигнального навигационного процессора для высокопроизводительных профессиональных применений (далее по тексту СБИС СНП-ВП) предназначена для создания высокопроизводительных профессиональных приемоизмерительных модулей, обеспечивающих прием и обработку существующих и перспективных сигналов ГНСС ГЛОНАСС/GPS/GALILEO и функциональных дополнений SBAS/СДКМ, применяемых в составе высокопроизводительной профессиональной навигационной аппаратуры. СБИС СНП ВП обеспечивает обработку сигналов ГЛОНАСС (L1OF, L1OC, L2OF, L2OC, L3OC); GPS (C/A (L1), L2C (L2), I5 + Q5 (L5); GALILEO (E1, E5a, E5b); SBAS (C/A (L1, L5); СДКМ (L1OC, L3OC), а также прием и обработку аналоговых сигналов в полосе частот до 100МГц.

СБИС содержит четыре 10-ти разрядных АЦП, обеспечивающих до 90Мвыб/с; ядро RISC процессора ARM1176JZF-S; два процессорных ядра с архитектурой NeuroMatrix для выполнения цифровой обработки сигналов; блок цифрового КИХ фильтра; встроенную память объемом 16Мбит; ПЗУ объемом 512 64-х разрядных слов; контроллер интерфейса с внешней памятью LPDDR2; блок интерфейсов, позволяющий осуществлять взаимодействие с внешними устройствами, включающий в себя Ethernet, UART, SPI, I2C, USB, CAN, GPIO; контроллер JTAG интерфейса для обеспечения возможности отладки программного обеспечения.

СБИС СНП-ВП работает на частоте 320МГц.

Документ описывает структуру СБИС, систему команд, карту памяти всех процессорных ядер, и описание работы всех подсистем.

2 Технические характеристики

2.1 Основные функциональные характеристики СБИС СНП-ВП

В данном разделе приводятся основные характеристики, дающие представление об особенностях архитектуры и производительности СБИС СНП-ВП.

В состав СБИС СНП-ВП входят:

- вычислительное ядро на базе RISC процессора ARM1176JZF-S;
- блок цифровой обработки сигналов на основе процессорных ядер с архитектурой NeuroMatrix для выполнения функций корреляции и декодера Витерби;
- встроенную память объемом 16 Мбит для выполнения программ и хранения данных;
- контроллер интерфейса с внешней памятью LPDDR2;
- блок интерфейсов, позволяющий осуществлять взаимодействие с внешними устройствами, включающий в себя: UART, SPI, I2C, USB, CAN, GPIO;
- блок Ethernet MAC с поддержкой удаленной отладки (EDCL-интерфейс);
- АЦП для приема навигационного сигнала;
- блок цифрового подавления узкополосных помех;
- часы реального времени с независимым электропитанием;
- блок формирования секундной метки и синхронизации с внешним событием;
- блок управления потреблением;
- блок управления прерываниями;
- интерфейс к флеш-памяти с последовательным SPI интерфейсом;
- блок ПЗУ для хранения кода программы начального загрузчика;
- блок внутрикристальной шины для обмена данными;
- интерфейс для обеспечения возможности отладки программного обеспечения;

В таблице ниже приведены основные параметры микросхемы.

Таблица 2.1 – Основные параметры микросхемы.

Наименование параметра	Значение параметра
Напряжение питания ядра, В	1,2 ±5%
Напряжения питания буферов ввода-вывода, В	3,3 ±10%
Частота входного синхросигнала, МГц	25-40
Частота работы встроенного RISC процессора, МГц	320
Объем встроенной памяти, Кбайт	2048
Инструментальная погрешность измерения (дискрет), не хуже:	
- псевдодальности, нс	0,1
- фазы несущей частоты, град	0,5
- частоты несущей, Гц	0,05
- тактовой частоты кода, Гц	0,01
Количество аналоговых входов от приемных трактов	4
Количество цифровых двухразрядных входов	12
Разрядность встроенных АЦП	10
Количество подавляемых узкополосных помех	6
Пониженная температура среды, ^{0}C	
- рабочая	-40
- предельная	-60
Повышенная температура среды, ^{0}C	
- рабочая	+85
- предельная	+100

3 Общая структура СБИС СНП-ВП

Общая структурная схема СБИС СНП-ВП приведена на рисунке ниже (см.Рисунок 3.1).

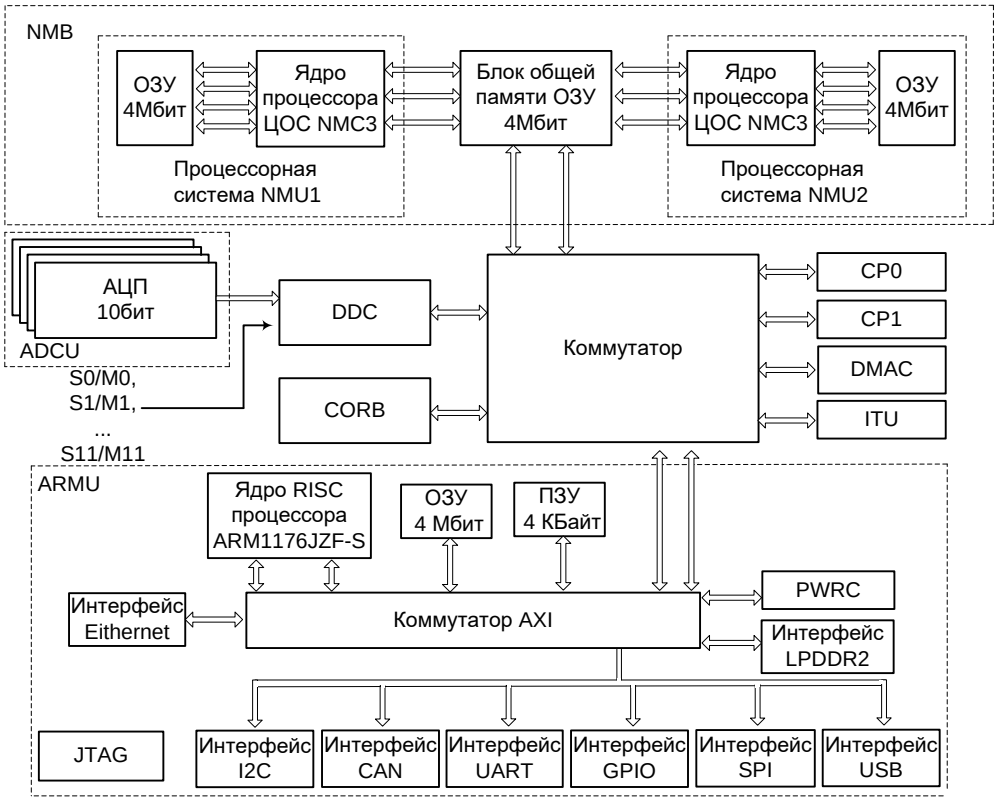


Рисунок 3.1 – Общая структурная схема СБИС СНП-ВП

СБИС содержит следующие функциональные узлы:

ARMU (ARM Unit) – процессорная система, состоящая из микропроцессорного ядра ARM11, памяти объёмом 4 Мбит, развитой шинной структуры с набором коммутаторов, а также периферийных устройств. Более подробно процессорная система ARMU будет описана ниже.

NMB (NeuroMatrix Block) блок цифровой обработки сигналов на основе процессорных ядер с архитектурой NeuroMatrix. Более подробно процессорная система NMU будет описана ниже.

ADCU - блок аналогово-цифрового преобразования, состоящий из четырёх 10-разрядных АЦП.

DDC (Digital Down Converter) – блок каналов преобразования частоты, который состоит из 8 аппаратных каналов, осуществляющих следующие операции:

- выбор источника сигнала,
- цифровое гетеродинирование с последующим накоплением,
- фильтрацию помех и возведение комплексного сигнала в квадрат,
- упаковку сигнала в 64-х разрядные слова и генерацию адреса для записи в память упакованных данных.

Управление данным блоком может осуществляться любой процессорной системой.

CORB (Correlation Block) – блок аппаратных корреляторов, осуществляющий корреляцию навигационных сигналов.

DMAC (DMA Controller) – контроллер ПДП, обеспечивающий обмен между памятью и блоком DDC, обмен память – память, обмен память – блок CORB. Управление осуществляется всеми процессорными системами.

ITU (Interval Timer Unit) – блок формирования секундной метки и синхронизации с внешним событием. Управление может осуществлятся любой процессорной системой.

PWRC (Power Control) – блок управления питанием. Данный блок позволяет отключать питание блоков СБИС. Данный блок служит для экономии потребляемой мощности.

CPx (Communication Port) – байтовый коммуникационный порт. С помощью этого порта можно реализовывать обмен между внешними устройствами (процессорами) типа «точка-точка» без использования дополнительной аппаратуры. Передаваемые и принимаемые по портам данные читаются из памяти или пишутся в память процессора в режиме ПДП, освобождая при этом процессорное ядро для другой работы.

3.1 Схема синхронизации

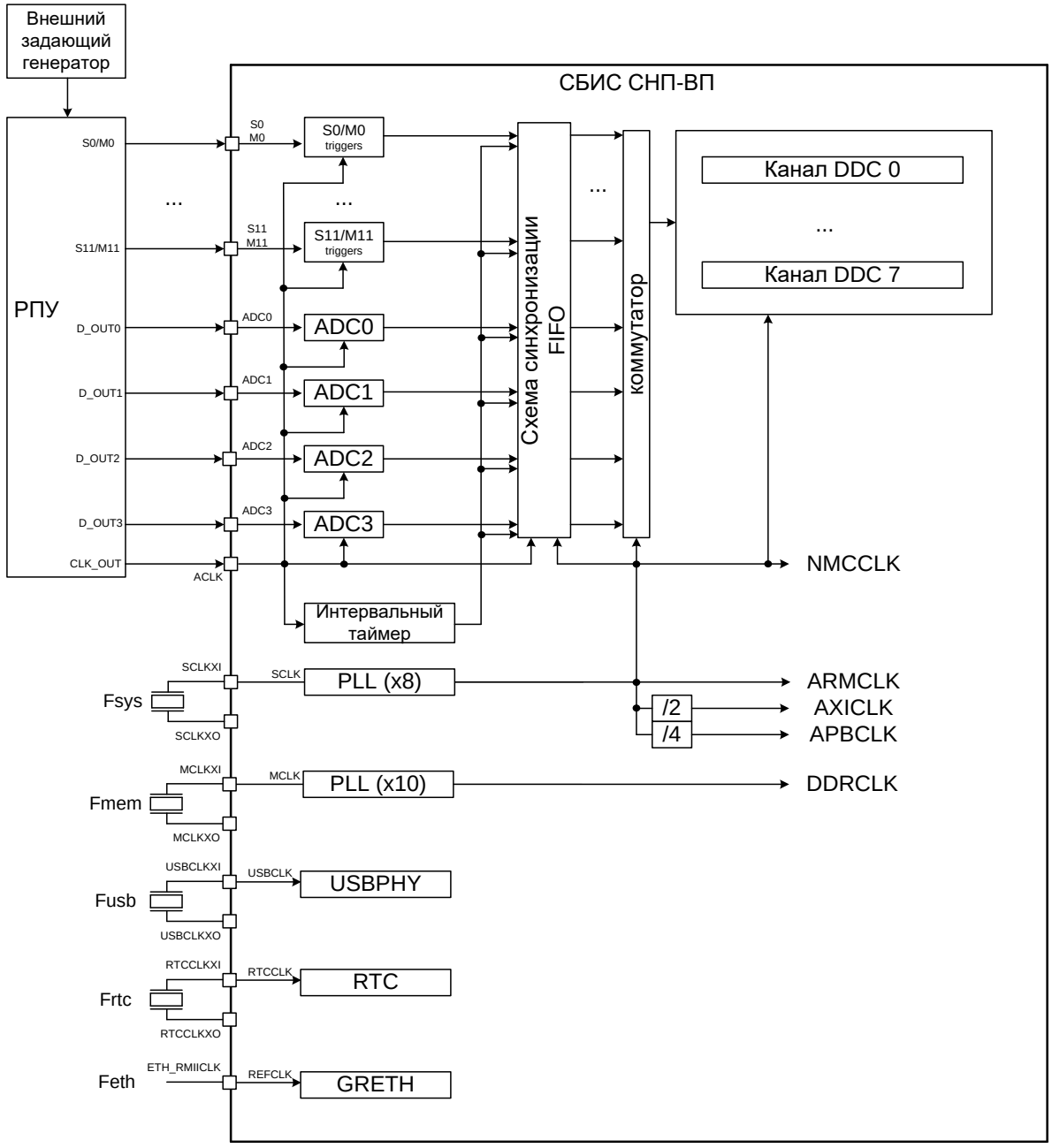


Рисунок 3.2 - Схема тактирования СНП-ВП

Для тактирования АЦП (ADC0 - ADC3) и входов SIGNx/MAGNx предназначен отдельный тактовый сигнал ACLK (см. Рисунок 3.2) , работающий с частотой $F_{ADC,}$ которая не должна превышать 90МГц. При этом входной тактовый сигнал, используемый для формирования основных тактовых сигналов микросхемы (SCLKXI), формируется независимо и не обязательно от того же генератора, который формирует тактовый сигнал для РПУ (радиоприемное устройство). При этом интервальный таймер в СБИС СНП-ВП тактируется от сигнала ACLK и формирует метки времени, жестко привязанные к входным данным. Все каналы могут запускаться,

останавливаться, перестраиваться и менять области для записи данных в память синхронно по этим меткам.

Для тактирования цифровой части микросхемы используются входы SCLKXx, MCLKXx, USBCLKXx, RTCCLKx и ETH_RMIICLK. Для генерации тактовых сигналов SCLK, MCLK, USBCLK и RTCCLK могут использоваться как кварцевые осцилляторы, так и генераторы. Для генерации тактового сигнала ETH_MIICLK интерфейса Ethernet (GRETH) используется микросхема физического интерфейса Ethernet, подключаемая к СБИС СНП-ВП по интерфейсу RMII. Ниже (Таблица 3.1) приведены разрешенные частоты для перечисленных выше опорных тактовых сигналов.

Таблица 3.1 — Разрешенный диапазон частот для опорных сигналов

Имя	Обозначение	Разрешенная частота, МГц
ACLK	F_{ADC}	до 90 МГц
SCLK	F_{SYS}	до 40 МГц
MCLK	F_{MEM}	до 40 МГц
USBCLK	F_{USB}	48 МГц
RTCCLK	F_{RTC}	32 кГц
ETH_RMIICLK	F_{ETH}	50 МГц

3.2 Карта памяти СБИС СНП-ВП

Карта памяти СБИС СНП-ВП приведена на рисунке ниже (см. Рисунок 3.3):

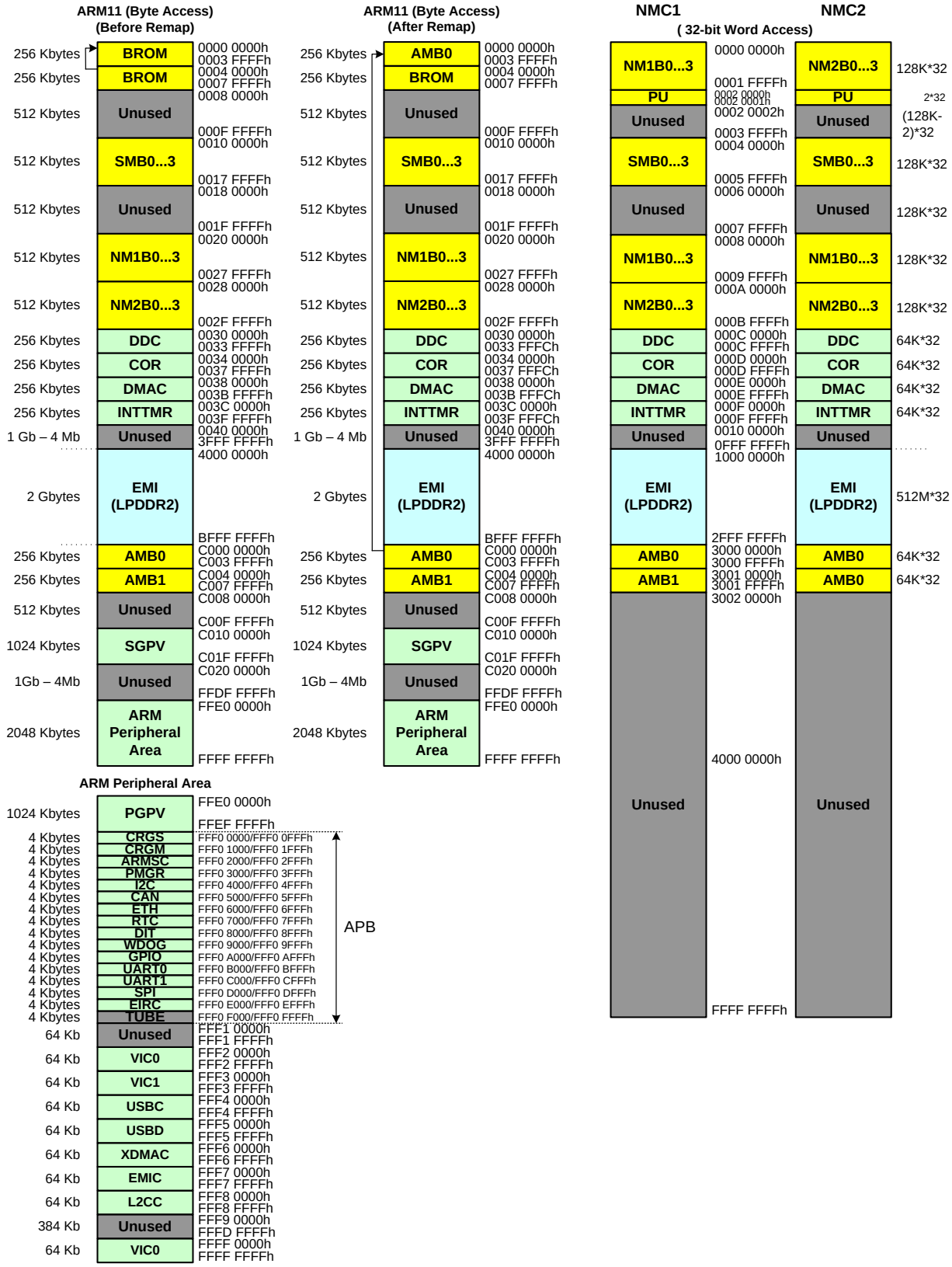


Рисунок 3.3 - Карта памяти СБИС СНП-ВП

BROM - ПЗУ загрузчика;
SMB3, SMB2, SMB1, SMB0 - четыре банка памяти блока общей памяти;
NM1B0, NM1B1, NM1B2, NM1B3 - четыре банка памяти процессорной системы NMU1;
NM2B0, NM2B1, NM2B2, NM2B3 - четыре банка памяти процессорной системы NMU2;
DDC – конфигурационные регистры блока DDC;
COR – конфигурационные регистры блока корреляторов;
DMAC – конфигурационные регистры блока ПДП;
INTTMR – конфигурационные регистры блока интервальных таймеров;
EMI – область внешней динамической памяти LPDDR2 (до 2х Гбайт);
AMB0, AMB1 - два банка памяти процессорной ситемы ARMU;
SGPV –регистры конфигурации коммутатора HPMS;
ARM Peripheral Area – регистры устройств, подключенных к коммутатору HPMP.

4 Блок цифровой обработки сигналов на основе процессорных ядер с архитектурой NeuroMatrix (NMB)

Структурная схема блока цифровой обработки сигналов на основе процессорных ядер с архитектурой NeuroMatrix приведена ниже (см. Рисунок 4.1).

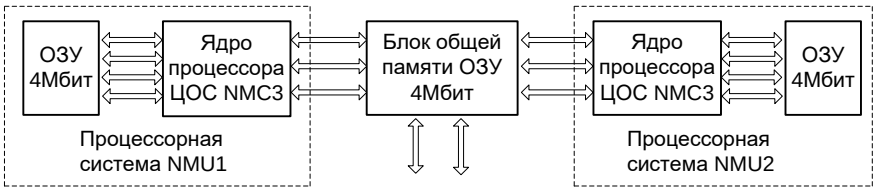


Рисунок 4.1 – Структурная схема блока цифровой обработки сигналов на основе процессорных ядер с архитектурой NeuroMatrix

Блок NMB состоит из блока общей паяти (SMU) и двух идентичных процессорных систем NMU1 и NMU2, содержащих процессорное ядро DSP процессора NMC3.

4.1 Процессорная система NMU на базе DSP ядра NMC3

Состав процессорной системы NMU приведен на рисунке ниже (см. Рисунок 4.2).

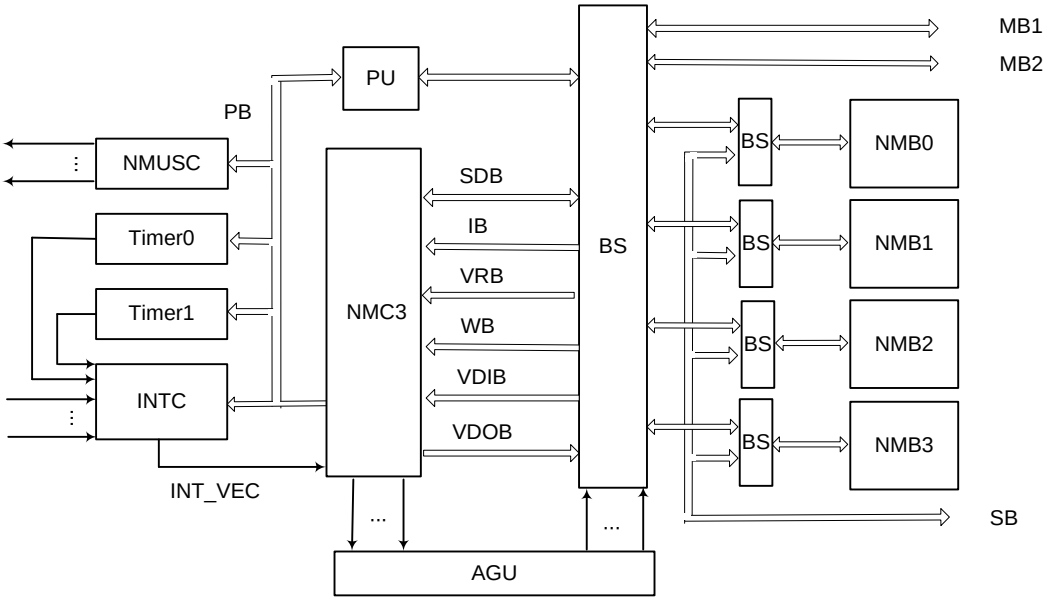


Рисунок 4.2 – Структурная схема процессорной системы на базе DSP

ядра NeuroMatrix Core 3

Процессорная система на базе DSP ядра NeuroMatrix Core 3 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-доступные со стороны NMC3 периферийные регистры и выполняет функции управления прерываниями другим процессорным системам NMU и ARMU и осуществляет вывод на внешние выводы тестовых сигналов.

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

- высокоприоритетное и низкоприоритетное от процессорной системы NMU;
- высокоприоритетное и низкоприоритетное от процессорной системы ARMU;
- одно прерывания от блока CORB;
- два прерывания от блока DDC;
- три прерывания от блока ITU;
- 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 ниже.

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.

MB (Memory Bank) - банк памяти объёмом 16K×64, который может адресоваться до 32- или 64-разрядного слова.

4.2 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 4.3).

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).

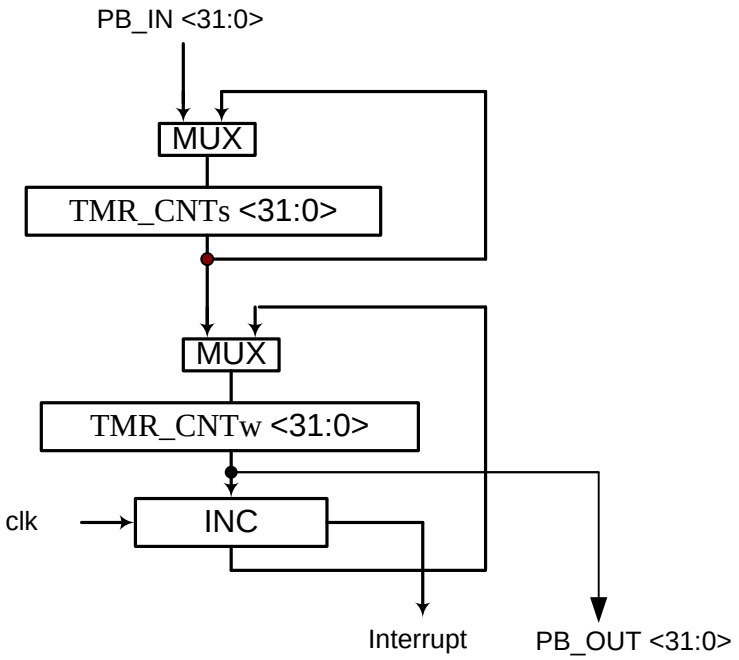


Рисунок 4.3- Структурная схема таймера

Теневой и рабочий регистр счетчика имеют одинаковый формат (см. Рисунок 4.4).

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 4.4 – 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 4.5 – 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 так же 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 2^{32} процессорных такта, для этого надо записать значение 0000_0000h в теневой регистр счетчика.

4.3 Системный контроллер процессорной системы NMU (NMCSC)

Контроллер управляет запросами на прерывание к процессорной системе **ARMU** и другой процессорной системе **NMU**, контролирует выводы тестового порта и содержит номер процессорной системы **NMU**, а так же формирует признаки ошибки при обращении ядром **NMC3** в память по несуществующим адресам. 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.

31–9	8	7	6	5	4	3	2	1	0
Зарезервировано	ERPR	ERMB2	ERMB1	TD	NMBR	ARMUL	ARMUH	NMUL	NMUH

Рисунок 4.6 – Формат регистра NMCSCR.

0-й разряд **(NMUH) -** запрос на высокоприоритетное прерывание другой процессорной системе **NMU**. Запись единицы в данный разряд вызовет высокоприоритетное прерывание в контроллере прерываний (**INTC**) другой системы **NMU**. Сброс данного разряда происходит автоматически.

1-й разряд **(NMUL) -** запрос на низкоприоритетное прерывание другой процессорной системе **NMU**. Запись единицы в данный разряд вызовет низкоприоритетное прерывание в контроллере прерываний (**INTC**) другой системы **NMU**. Сброс данного разряда происходит автоматически.

2-й разряд **(ARMUH) -** запрос на высокоприоритетное прерывание процессорной системе **ARMU**. Запись единицы в данный разряд вызовет высокоприоритетное прерывание в системном контроллере (**ARMSC**) системы **ARMU**. Сброс данного разряда происходит автоматически.

3-й разряд **(ARMUL) -** запрос на низкоприоритетное прерывание процессорной системе **ARMU**. Запись единицы в данный разряд вызовет низкоприоритетное прерывание в системном контроллере (**ARMSC**) системы **ARMU**. Сброс данного разряда происходит автоматически.

4-й разряд **(NMBR)** определяет номер процессорной системы NMU. Данный разряд доступен только для чтения:

0 – для процессорной системы NMU1;

1 – для процессорной системы NMU2;

Попытка записи в данный разряд 0 или 1 не оказывает никакого влияния на его значение.

Поле **TD** (8-5 разряды) является тестовым портом. Значение, записанное в данное поле, будет через два такта после записи выдано на внешние выводы микросхемы:

на внешние выводы микросхемы **TD3** – **TD0** для процессорной системы NMU1;

на внешние выводы микросхемы **TD7** – **TD4** для процессорной системы NMU2.

9-й разряд (**ERMB1**) - устанавливается, если по шине MB1 было обращение по несуществующим адресам 2_0002h - 3_FFFFh, 6_0002h - 7_FFFFh. При этом реальное обращение будет происходит в память SMU. ДАнный разряд доступен по чтению и записи.

10-й разряд (**ERMB2**) - устанавливается, если по шине MB2 было обращение по несуществующим адресам 2_0002h - 3_FFFFh, 6_0002h - 7_FFFFh. При этом реальное обращение будет происходит в память SMU.

11-й разряд (**ERPR**) устанавливается, если по шине MB1 или MB2 было обращение по несуществующим адресам 10_0000h - 0FFF_FFFFh. При этом запись будет проигнорирована, при чтении получаемы данные не определены.

4.4 Старт и остановка выборки команд ядром NMC3

Перед запуском процессорного ядра NMC3 процессорной системы NMU необходимо поместить код исполняемой программы во внутренней памяти соответствующей процессорной системы NMU с адреса 0h. После запуска программного счетчика будет произведена выборка команд с адреса 0h. Это возможно сделать двумя способами: процессорной системой ARMU (возможно только после загрузки ядра ARM1176) или использовать коммуникационные порты (возможно без начальной загрузки ядра ARM1176).

Порядок действий при использовании процессорной системы ARMU:

- поместить программу во внутреннюю память NMU средствами процессорной системы ARMU;
- запустить программный счетчик ядра NMC3 путем записи единицы в регистр STR[0] (см. описание блока PWRC 5.12).

Порядок действий при использовании коммуникационных портов:

- Установить вывод BOOT_NMx=1 (или прописать регистр LDR[0]=1 блока PWRC). Это означает загрузку по коммуникационному порту. Вывод BOOT_NM инициализирует 0-й бит регистра LDR после системного сброса. В процессе работы LDR[0] может быть изменен программно процессорной системой ARMU;
- обеспечить передачу 256-ти слов размерностью 64 бита по соответствующему коммуникационному порту (для процессорной системы NMU0 используется CP0, для процессорной системы NMU1 используется CP1);
- после приема 256-ти 64-х разрядных слов данные будут помещены во внутреннюю память NMU, начиная с адреса 0000h. После этого запуск ядра NMC3 происходит автоматически.

Программный счетчик ядра NMC3 процессорной системы NMU может быть остановлен. Это может быть использовано:

- для уменьшения потребления;
- перед отключением питания или тактового сигнала;
- для использования внутренней памяти NMU другой процессорной системой.

Для остановки используется программно доступный регистр ***WAIT_REG.*** Регистр находится в нулевом окне периферийном окне и имеет адрес pr8. Формат ***WAIT_REG*** приведен на рисунке ниже.

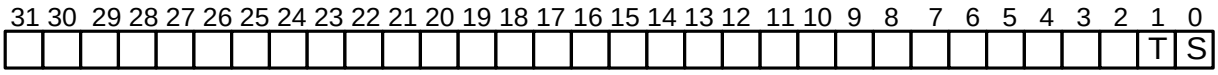


Рисунок 4.7 – Формат регистра WAIT_REG.

S=1 - остановить программный счетчик;

S=0 - не останавливать программный счетчик;

S=1, T=0 - остановка без блокировки SB шины (можно использовать память останавливаемого NMU другим процессорным системам);

S=1, T=1 - остановка с блокировкой SB шины (нельзя использовать память останавливаемого NMU другим процессорным системам, при этом команды записи игнорируются, а чтение возвращает данные, которые являются не определены). Данный останов для отключения питания или тактового сигнала.

Для остановки программного счетчика NMC необходимо:

- очистить конвейер (подать исполняемый NOP со сброшенным битом параллельной работы);
- записать единицу в нулевой бит регистра WAIT_REG (pr8 в нулевом регистровом окне);
- дождаться ее выполнение;
- подать команду перехода (move_pc, read_pc) на метку.

После повторного запуска программного счетчика (если питание не отключалось) выполнение продолжится с метки, на которую был выполнен переход.

Ниже приведен пример программы для остановки программного счетчика:

```
        nul; nul; nul; nul;
        nul; nul; nul; nul;
        pr8 = 1;
        nul; nul; nul; nul;
        nul; nul; nul; nul;
        pc = IAG_Stopped;
        nul;
    <IAG_Stopped>
```

Для процессорной системы NMU предусмотрено внешнее входное прерывание STOP_IAG, которое извещает, что выборку команд необходимо остановить. Программа обработки прерывания должна содержать программный код, описанный выше, должен быть обязательно выполнен перед отключением питания. Это гарантирует, что все отправленные запросы на чтение данных получили свои данные, и не осталось никаких данных от останавливаемого ядра.

4.5 Контроллер прерываний (INTC)

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

- высокоприоритетное и низкоприоритетное от процессорной системы NMU;
- высокоприоритетное и низкоприоритетное от процессорной системы ARMU;
- два прерывания от блока DDC;
- одно прерывание от блока CORB;
- три прерывания от блока ITU;
- два прерывание от блока DMAC;
- одно прерывание от блока TIMER0;
- одно прерывание от блока TIMER1;
- одно прерывание от блока PU;
- одно прерывание для установки программного счетчика.

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

Запросы на прерывания приходят на входы блока INTC IRQ[15:0] (см. Рисунок 4.8). 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.

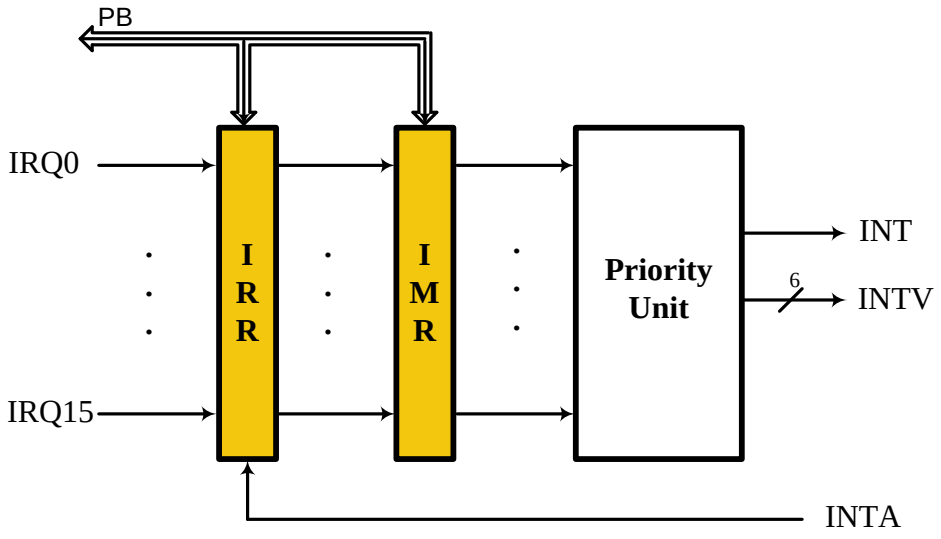


Рисунок 4.8 – Структура контроллера прерываний.

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 **IRQ**x 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 ниже (Таблица ***4***.1).

Таблица 4.1 – 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 4.9). Единица в соответствующем разряде означает запрос на прерывание.

31–16	15–0
Reserved	IRQ<15:0>

Рисунок 4.9 – Формат регистра IRR

Формат регистра **IMR** приведен ниже (см. Рисунок 4.10). 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 4.10 – 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;
        intr clear 03c0h;
```

или

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

Список прерываний, которые подаются на входы запросов IRQ, и соответствующий им адрес-вектор, приведен ниже (Таблица 4.2).

Таблица 4.2 – Список внешних прерываний для ядра NMC3

№	источник	Адрес вектор		Описание
		bin	hex	
1	INT_ARMH_IN	1000_0000b	80h	Самое приоритетное. Высокоприоритетное прерывание от ARM-а
2	INT_NMUH_IN	1000_1000b	88h	Высокоприоритетное прерывание от соседнего ядра NMC
3	INT_TIMER0	1001_0000b	90h	Прерывание от таймера 0
4	INT_MS_ITU	1001_1000b	98h	Прерывание от интервального таймера по ms
5	INT_CH_DDC	1010_0000b	A0h	Прерывание от каналов блока DDC
6	INT_AUX_ITU	1010_1000b	A8h	Прерывание от интервального таймера по дополнительному временному интервалу
7	INT_DMAC	1011_0000b	B0h	Прерывания от блока DMAC - INT_DMAC_DR
8	INT_DMAC	1011_1000b	B8h	Прерывания от блока DMAC - INT_DMAC_MC
9	INT_TIMER1	1100_0000b	C0h	Прерывание от таймера 1
10	INT_COR	1100_1000b	C8h	Прерывание от блока корреляторов
11	INT_ARML_IN	1101_0000b	D0h	Низкоприоритетное прерывание от ARM-а
12	INT_NMUL_IN	1101_1000b	D8h	Низкоприоритетное прерывание от соседнего ядра NMC
13	INT_PU	1110_0000b	E0h	Прерывание по окончанию работы упаковщика
14	INT_STRB_ITU	1110_1000b	E8h	Прерывание от интервального таймера по началу выдачи программного строба
15	INT_PRG_DDC	1111_0000b	F0h	Прерывание от каналов блока DDC
16	STOP_IAG	1111_0000b	F8h	Самое не приоритетное

4.6 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 втором – 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, 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.

4.6.1 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 4.11)

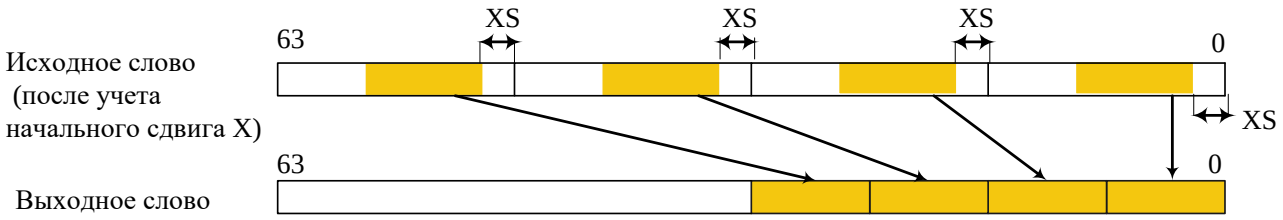


Рисунок 4.11 – Упаковка без насыщения

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 4.12).

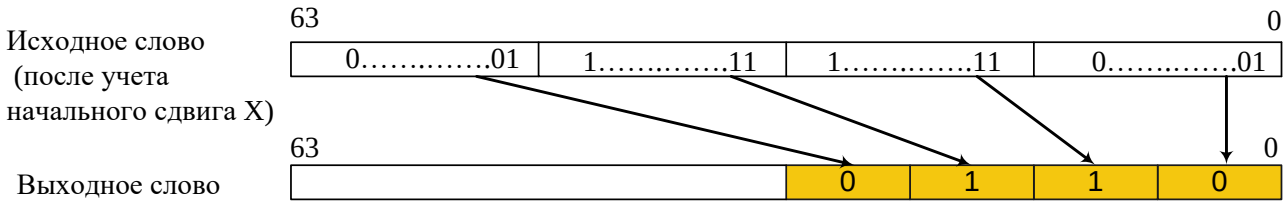


Рисунок 4.12 – Упаковка в вектора с одноразрядными данными

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 4.13).
- Распространять знаковый разряд или распространять нуль.

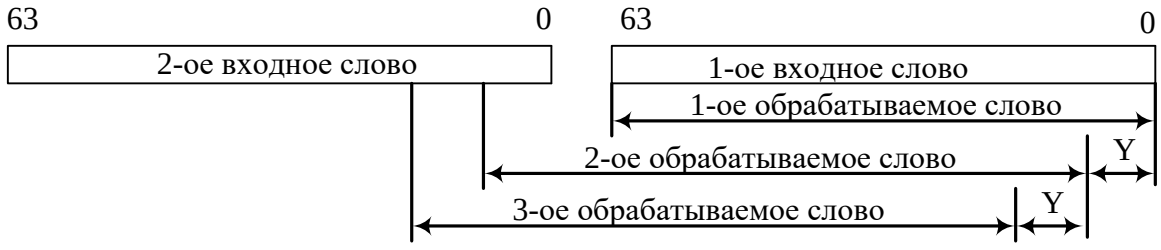


Рисунок 4.13 - Получение слов для обработки из входного потока данных

Пример распаковки для N = 8, M = 16, распространение нулем представлен ниже (см. Рисунок 4.14).

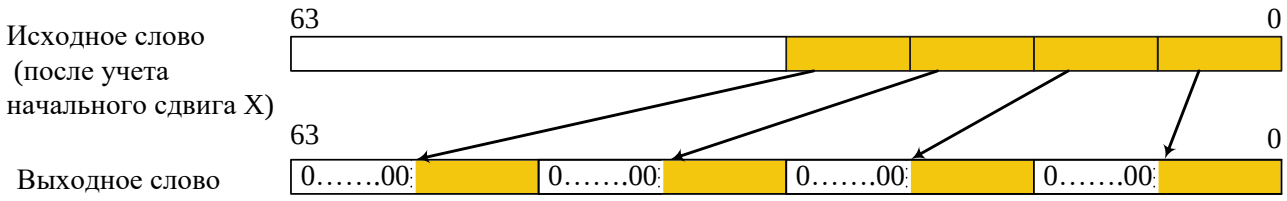


Рисунок 4.14 – Распаковка N=8, M=16, распространение нулем

Пример распаковки для N = 8, M = 16, распространение знаком представлен ниже (см. Рисунок 4.15)

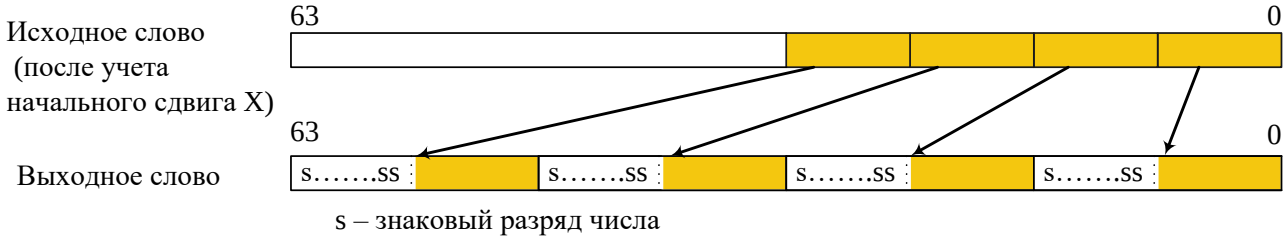


Рисунок 4.15 – Распаковка N=8, M=16, распространение знаком

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 4.16).

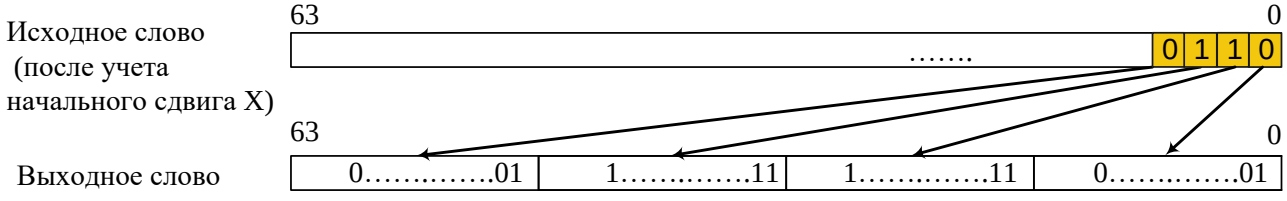


Рисунок 4.16 – Распаковка входного слова одноразрядных данных

Порядок выдачи данных в выходном слове. Результат может быть выдан, начиная с младших элементов или начиная со старших элементов в выходном слове (см. Рисунок 4.17 и Рисунок 4.18).

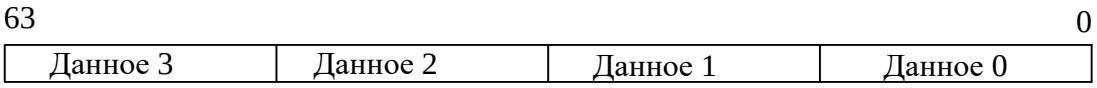


Рисунок 4.17 – Выдача слова, начиная с младших элементов

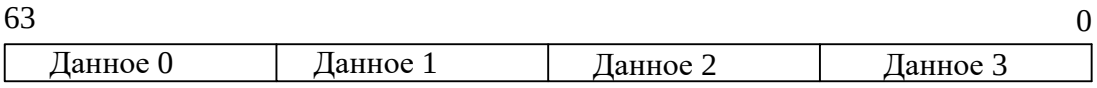


Рисунок 4.18 – Выдача вектора, начиная со старших элементов

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 4.19). Данное действие выполняется вместе с режимом распаковки.

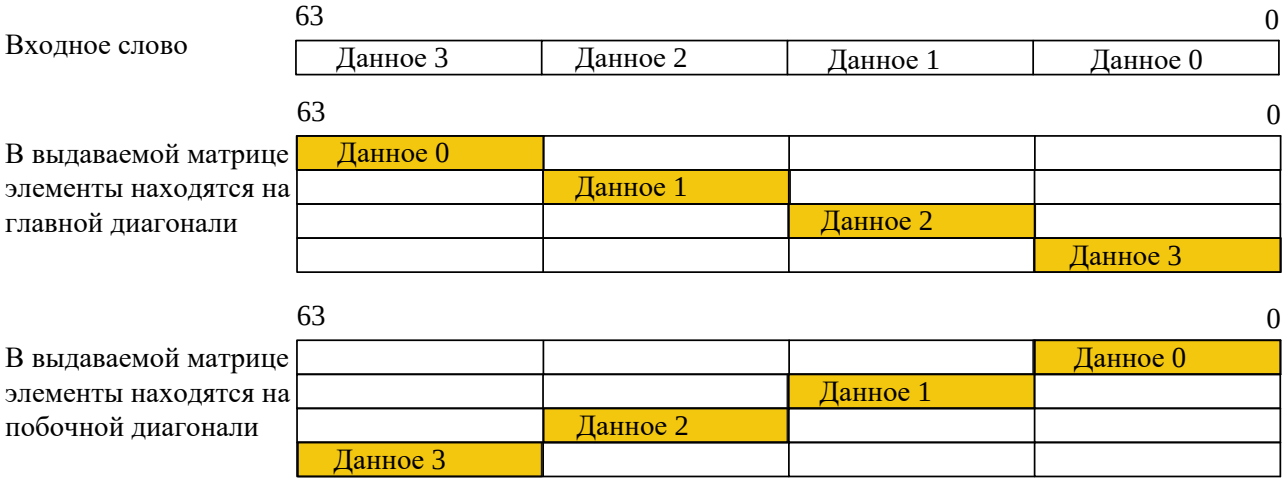


Рисунок 4.19 – Получение диагональной матрицы

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 4.20). Данное действие выполняется вместе с режимом распаковки.

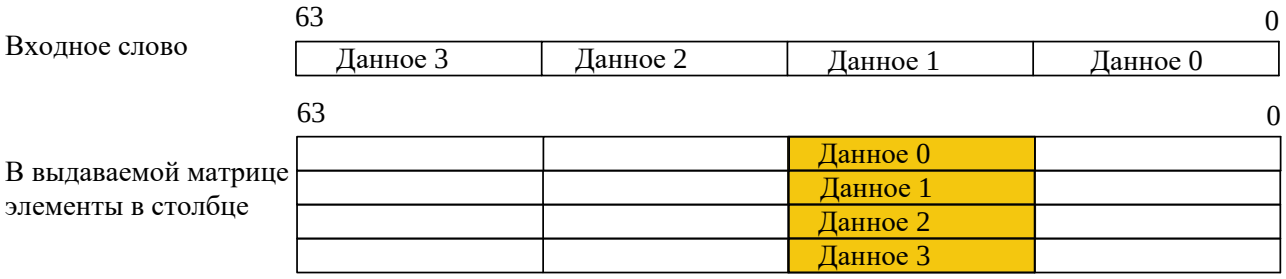


Рисунок 4.20– Размещение входного слова в первом столбце матрицы

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.

4.6.2 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.

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.

4.6.3 Конфигурационные регистры

Таблица 4.3 – Коды регистров блока PU (периферийные регистры для NMC3)

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

Регистры блока **PU** доступны *только для записи* как периферийные регистры ядра **NMC3**. Коды регистров блока **PU** приведены выше (Таблица 4.3). 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).

31–10	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	0	0	0	0	0	0	0
зарезервировано	Xs	SHIFT	COLNUM	COL	DIAG	MATOUT	MATIN	MSB	BIT	SAT	SE	PACK A A	MODE

Рисунок 4.21 – 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 ниже (Таблица 4.4). Все другие сочетания значений являются запрещенными.

Таблица 4.4 – 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)

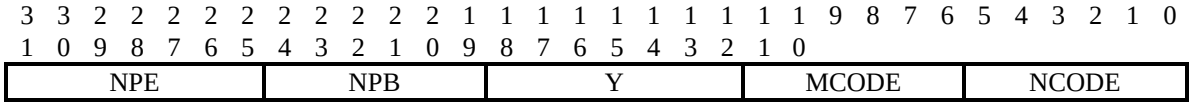


Рисунок 4.22 – Формат регистра PUR8

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 4.23 – 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 4.24 – 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 4.25 – 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 4.26 – 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)

31–0
CNT_OUT

Рисунок 4.27 – 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)

31–0
MASKS

Рисунок 4.28 – 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 4.29)

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

Рисунок 4.29. 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 4.30).

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 4.30 - 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 антупны 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 4.31 – 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 4.32 – 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 4.33).

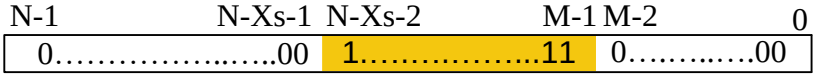


Рисунок 4.33. Формат общей части для получения маски MASKSAT при упаковке

Оставшиеся разряды маски получаются путем дублирования этой общей части. Ниже приведен пример полученной маски с общей частью 000…111…000 (см. Рисунок 4.34).

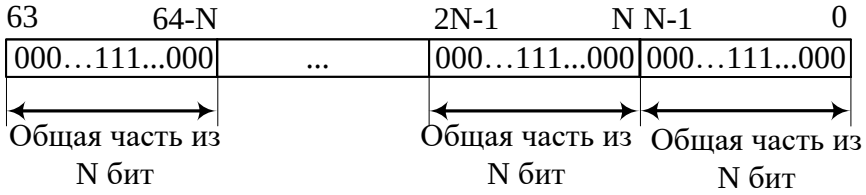


Рисунок 4.34 - Пример полученной маски MASKSAT при упаковке

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 4.35).

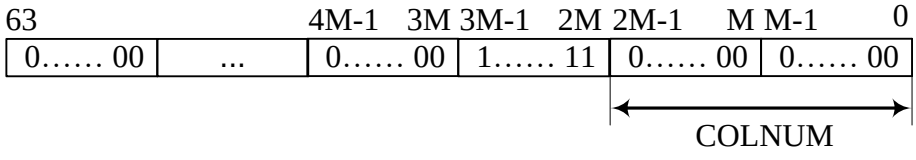


Рисунок 4.35. Пример маски MASKSAT при получении матрицы с элементами в столбце

При получении матрицы с элементами в главной или побочной диагонали поле **MASKSAT<63:0>** содержит единицы в **M** младших разрядах (см. Рисунок 4.36).

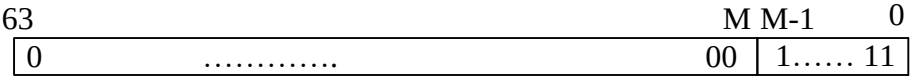


Рисунок 4.36 - Пример маски MASKSAT при получении матрицы с элементами в главной или побочной диагонали

4.7 Система общей памяти (SMU)

Структурная схема системы общей памяти приведена ниже (см. Рисунок 4.37).

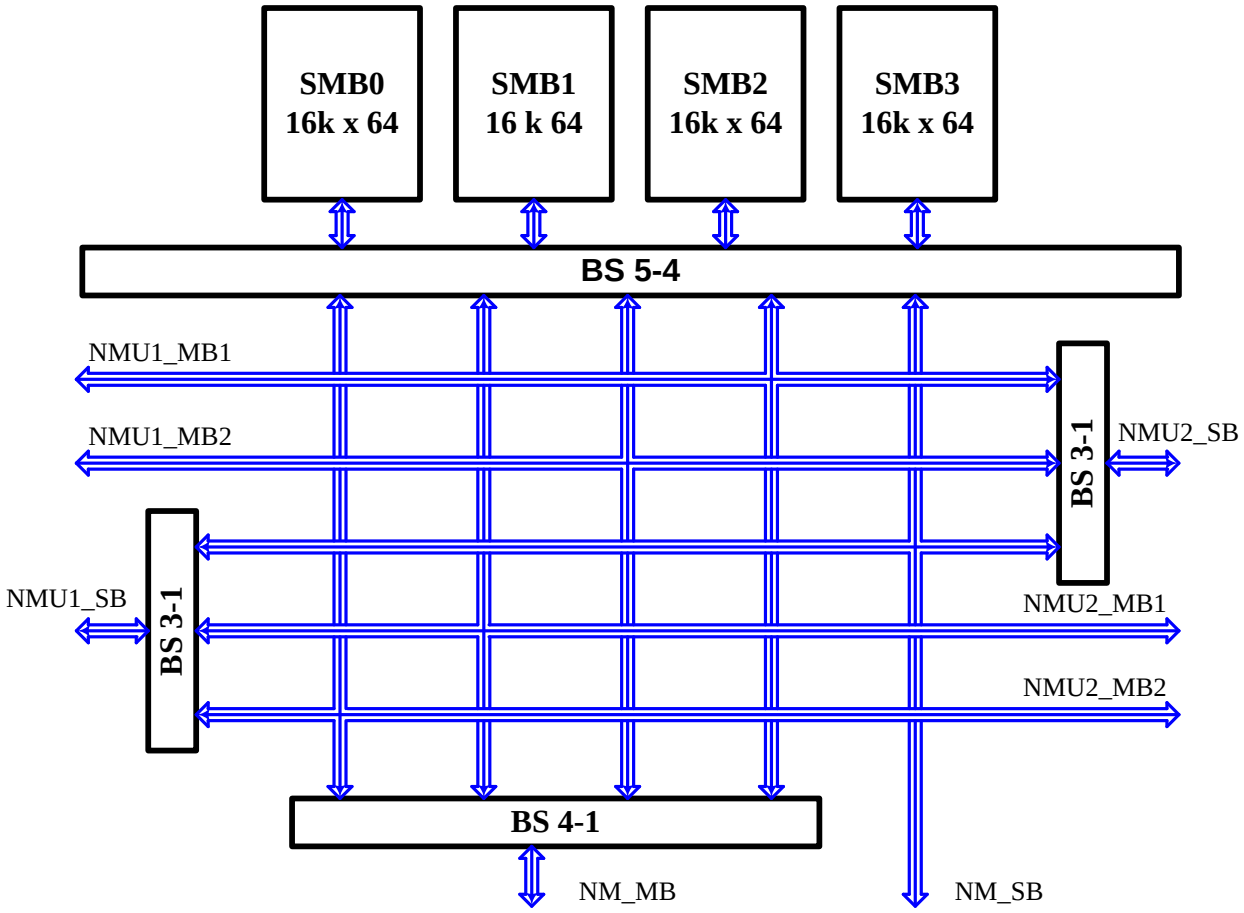


Рисунок 4.37- Структурная схема системы общей памяти (SMU)

Система общей памяти может работать на частоте до 320 МГц и содержит следующие функциональные узлы:

BS (Bus Switch) – шинный коммутатор.

SMB (Shared Memory Bank) - банк памяти объёмом 16K×64, который может адресоваться до 32- или 64-разрядного слова.

4.8 Список периферийных регистров для ядра NMC3

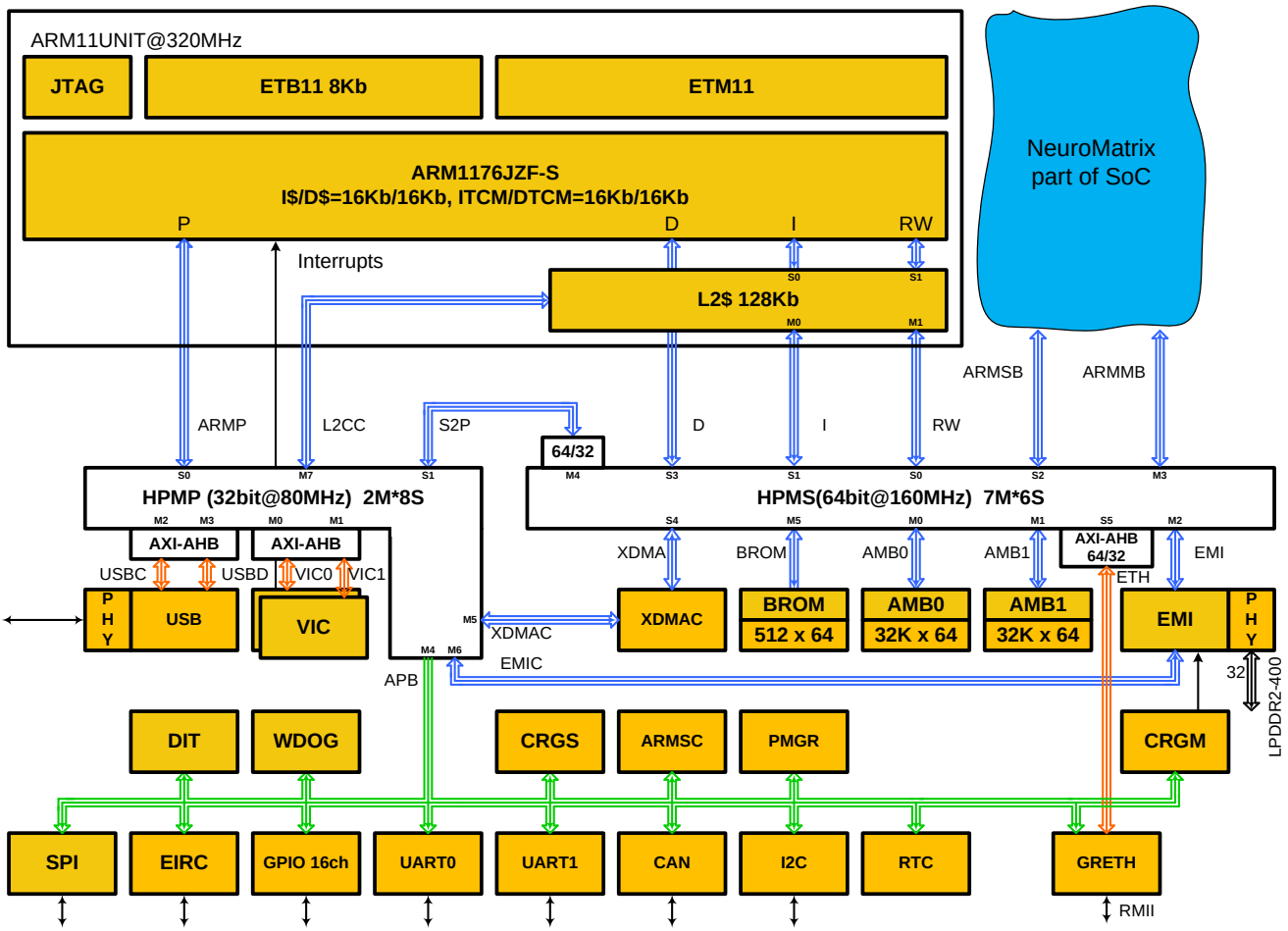
Список всех периферийных регистров для ядра NMC3 приведен в таблице ниже (Таблица 4.5). Все другие сочетания значений являются запрещенными.

Таблица 4.5 – 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		WAIT_REG		PUR2
111001				PUR3
111010				PUR4
111011				PUR5
111100				PUR6
111101				PUR7
111110				PUR8
111111				PUR9

5 Процессорная система на базе ядра ARM11 (ARMU).

Структурная схема ARMU части системы приведена ниже (см.Рисунок 5.1).



ARM11UNIT – ARM1176 Unit
ARM1176JZF-S – ARM Core
ETM11 – ARM11 Embedded Trace Macrocell
ETB11 – ARM11 Embedded Trace Buffer
L2$ - L2 Cache Memory
XDMAC – Direct Memory Controller
USB – Universal Serial Bus 2.0 High Speed Device
VIC0, VIC1 – Vectored Interrupt Controller
BROM – Boot ROM
AMB0,1 – AMBA Memory Bank 0,1 RAM
DIT – Dual Timers
WDOG – Watchdog Timer
CRGS – System Clock and Reset Generator
ARMSC – ARM System Controller
PMGR – Power Manager
CRGM – Memory Clock and Reset Generator
RTC – Real Time Clock
SPI – Serial Peripheral Interface Controller
EIRC – External Interrupt Controller
GPIO – Generap Purpose Input/Output
UART0,1 - Universal Asynchronous Receiver/Transmitter
EMI – External Memory Interface (LPDDR2)
CAN – Controller Area Network Controller
I2C - Inter-Integrated Circuit Controller
GRETH – Ethernet MAC with RMII Interface

HPMS – System AXI 64b High Performance Interconnect Matrix
HPMP – Peripheral AXI 32b High Performance Interconnect Matrix
I – ARM11 AXI 64b Instruction Bus
RW – ARM11 AXI 64b Data Bus
D – ARM11 DMA AXI 64b TCM Interface
ARMP – ARM11 AXI 32b Peripheral Bus
ARMMB –ARM-NMC AXI 64b Master Bus
ARMSB – ARM-NMC AXI 64b Slave Bus
S2P – AXI 64b HPMS->HPMP Interface
L2CC – L2$ Configuration AXI 32b Interface
XDMAC – XDMA Controller AXI 32b Config. Interface
EMIC – External Memory Interface Config. AXI 32b Bus
USBC – USB Controller CPU AXI 32b Interface
USBD - USB Controller DMA AXI 32b Interface
VIC0,VIC1 – AHB 32b VIC Bus
XDMA – XDMAC AXI 64b Master Interface
BROM – Boot ROM AXI 64b Bus
AMB0 – AMBA Memory Bank 0 AXI 64b Bus
AMB1 – AMBA Memory Bank 1 AXI 64b Bus
ETH – GRETH AHB 32b Master Bus
EMI – External Memory Interface AXI 64b Bus
APB – AMBA Peripheral Bus

Рисунок 5.1 – Структурная схема ARMU части системы

Процессорная система на базе ядра ARM11 содержит следующие функциональные узлы:

ARM11UNIT – СФ блок, включающий в себя микропроцессорное ядро ARM, кэш-память второго уровня размером 128Кбайт, а также дополнительные блоки отвечающие за отладку программ (ETM11, ETB11 и JTAG).

ARM1176JZF-S – микропроцессорное ядро фирмы ARM Ltd. В его состав входят кэш-память команд и данных объёмом 16 Кбайт каждая, памяти TCM команд и данных (16Кбайт каждая), а также блок операций с плавающей точкой. Обмен ядра с внешним миром осуществляется с помощью двух 64-разрядных шин: команд (I) и данных (RW), 64-разрядной шины ПДП(D), а также 32-разрядной периферийной шины (P).

ETM11 (ARM11 Embedded Trace Macrocell) – специальный блок, облегчающий процесс отладки программ. Блок позволяет записывать во внутренний буфер (ETB11) трассы выполненных ядром команд на рабочей частоте. Программирование блока осуществляется по JTAG интерфейсу.

ETB11 (ARM11 Embedded Trace Buffer) – внутренний буфер объемом 8 Кб, хранящий записываемые блоком ETM11 трассы. Позволяет записывать данные на «быстрой» частоте процессора, а выгружать для анализа на «медленной» частоте JTAG интерфейса.

L2$ – совмещенная кэш-память команд и данных второго уровня размером 128Кб. Имеет набор программно доступных через 32-х разрядную AXI шину регистров.

JTAG (JTAG Interface) – интерфейс с 5-выводным тестовым портом, реализованным согласно стандарту IEEE Std. 1149.1-1990. JTAG интерфейс позволяет тестировать СБИС СНП-ВП как законченное изделие или в составе электронной аппаратуры.

HPMx (AXI High Performance Matrix) – коммутаторы шин, работающих согласно стандарту AMBA AXI фирмы ARM Ltd. ARMU содержит два таких коммутатора: системный (HPMS) и периферийный (HPMP). Системный коммутатор работает с 64-разрядными шинами и осуществляет коммутацию 7-ми мастеров и 6-ти управляемых устройств и является многоуровневым, т.е. обеспечивает одновременную передачу данных всеми мастерами при условии работы с разными управляемыми устройствами. Также системный коммутатор позволяет программно настраивать приоритеты мастеров при обращении к одному управляемому устройству. Периферийный коммутатор работает с 32-разрядными шинами. Он осуществляет коммутацию 2-х мастеров, работающих по стандарту AXI, и 8-ми управляемых устройств (четыре AHB устройства, одно APB устройство и 3 AXI устройства) и является двухуровневым. Периферийный коммутатор также позволяет программно настраивать приоритеты доступа мастеров.

DMAC (Direct Memory Access Controller) – контроллер ПДП, обеспечивающий обмен между памятью и периферийными устройствами (UART0, UART1, SPI, USB, I2C) и управляемый процессорным ядром ARM11 через периферийную шину APB.

VIC (Vectored Interrupt Controller) – два контроллера прерываний, каждый из которых поддерживает до 32-х прерываний с программно настраиваемыми адрес-векторами и приоритетами. Контроллеры соединены в цепочку, поддерживая все 64 прерывания СБИС СНП-ВП.

CRGS (System Clock and Reset Generator) – блок, управляющий генерацией синхросигналов и сигналов сброса. Позволяет программным образом менять рабочую частоту некоторых блоков, а также производить программный сброс микросхемы.

ARMSC (ARM System Controller) – системный контроллер процессорной системы ARMU.

Данный контроллер содержит набор программно доступных со стороны ARM периферийных регистров, которые выполняют следующие функции:

- Регистры состояния входов – показывают состояние входов, управляющих источником начальной загрузки, а также состоянием тестовых входов
- Регистры управления АЦП – позволяют управлять режимом работы встроенного блока АЦП

- Регистры генерации прерываний к NMU части – управляют генерацией прерываний к NMU части

PMGR (Power Manager) – блок управления питанием, позволяет выборочно снимать питание с частей микросхемы.

DIT (Dual Input Timer) – два интервальных 32-разрядных таймера.

WDOG (Watchdog Unit) – сторожевой таймер.

RTC (Real Time Clock) – счётчик реального времени. В блоке имеются:

- регистр текущего времени, в котором хранятся секунды, минуты, часы и дни недели;
- регистр текущей даты, содержащий число, месяц, год и столетие;
- регистр будильника, содержащий секунду, минуту, час и день недели срабатывания будильника;
- управляющий регистр и регистр состояния, с помощью которых производится управление запросом на прерывание, включение тестового режима и сброс счётчиков, а также контроль над записью некорректных данных в регистры блока.

BROM (Boot ROM) – банк памяти ПЗУ объёмом 4 Кбайт, который содержит программу начального загрузчика СБИС СНП-ВП.

AMBx (ARM Memory Bank) – 2 банка памяти ОЗУ объёмом 256 Кбайт, которые могут адресоваться до 8-, 16-, 32- или 64-разрядного слова.

XDMAC – универсальный 4-х канальный контроллер ПДП.

EMI (External Memory Interface) – 32-х разрядный интерфейс к внешней памяти LPDDR2-400.

CRGM (Memory Clock and Reset Generator) – блок, управляющий генерацией синхросигналов и сигналов сброса. Позволяет программным образом менять рабочую частоту интерфейса с внешней памятью (независимо от остальных блоков микросхемы).

USB2.0 (Universal Serial Bus 2.0 High Speed Device) – контроллер интерфейса шины USB, соответствующей спецификации 2.0 (High-speed).

SPI (Serial Peripheral Interface) – контроллер синхронного последовательного интерфейса, обеспечивающий обмен по интерфейсу Motorola SPI.

UARTx (Universal Asynch. Receiver-Transmitter) – два контроллера асинхронного последовательного интерфейса.

GPIO (General Purpose Input/Output) – блок программируемых вводов/выводов общего назначения. Блок GPIO содержит 16 каналов.

CAN (Controler Area Network) – контроллер интерфейса CAN.

I2C – контроллер интерфейса I2C. Поддерживает Standart(100kbps) и Fast(400kbps) режимы.

GRETH - Gaisler Research’s Ethernet MAC контроллер обеспечивает интерфейс между AMBA- AHB шиной и сетью Ethernet. Он поддерживает скорость передачи 10/100 Мбит/сек в полудуплексном и дуплексном режимах. Интерфейс Ethernet поддерживает RMII интерфейс, который должен быть соединен 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 GRETH имеет доступ ко всем областям внутренней памяти.

EIRC (External Interrupt Controller) – контроллер внешних прерываний.

Все функциональные узлы ARMU соединяются шинами четырех типов:

- 32 и 64-разрядные AXI (AMBA eXtendable Interconnect) шины (изображены синим цветом);
- 32-разрядные AHB (AMBA High Speed Bus) шины (изображены оранжевым цветом);
- 32-разрядные APB (AMBA Peripheral Bus) шины (изображены зеленым цветом);

Здесь и далее шины изображены таким образом, чтобы показать направление, как потоков данных, так и адресов. Широкая стрелка отображает направление обмена данными (шину данных), узкая стрелка внутри неё – направление выдачи адреса (шину адреса). Ниже даётся краткое описание шин СБИС СНП-ВП:

I – 64-разрядная AXI шина загрузки команд микропроцессорного ядра из памяти. Инициатор обмена – ARM1176UNIT.

RW – 64-разрядная AXI шина обмена данными между микропроцессорным ядром и памятью. Инициатор обмена – ARM1176UNIT.

D – 64-разрядная AXI шина встроенных в ядро каналов ПДП, через эту шину осуществляется загрузка/выгрузка памятей TCM. Инициатор обмена – ARM1176UNIT.

L2CC – 32-разрядная AXI шина для настройки и управления кэш-памятью второго уровня. Инициатор обмена – ARM1176UNIT.

ARMP – 32-разрядная AXI шина обмена данными между микропроцессорным ядром и периферийными устройствами. Инициатор обмена – ARM1176UNIT.

ARMMB – 64-разрядная AXI шина обмена данными между ARMU и SMU. Инициатор обмена – ARMU.

ARMSB – 64-разрядная AXI шина обмена данными между SMU и ARMU. Инициатор обмена – SMU.

XDMA – 64-разрядная AXI шина контроллера ПДП (XDMAC). Инициатор обмена – XDMAC.

XDMAC – 32-разрядная AXI шина контроллера для управления контроллером XDMAC.

S2P – 32-разрядная AXI шина, позволяющая блокам XDMAC и GRETH получить доступ к периферийными устройствам, подключенным к периферийному коммутатору.

BROM, AMB0, AMB1 – 64-разрядные AXI шины обмена данными между соответствующими банками внутреннего ОЗУ и ПЗУ и мастерами подключенными к системному коммутатору HPMS.

VICx – 32-разрядные шины обмена данными между ARM1176UNIT и контроллерами прерываний (VIC). Инициатор обмена – ARM1176UNIT.

APB – 32-разрядная шина обмена данными между ARMU и периферийными устройствами. Инициатор обмена – ARMU.

USBC – 32-разрядная шина обмена данными между ARM1176UNIT и контроллером USB. Инициатор обмена – ARM1176UNIT.

USBD – 32-разрядная шина обмена данными между контроллером ПДП XDMAC и контроллером USB. Инициатор обмена - XDMAC.

EMI – 64-разрядная шина для доступа во внешнюю динамическую память.

EMIC – 32-разрядная шина для управления контроллером внешней памяти.

5.1 Микропроцессорная сборка на базе ARM1176JZF

Процессорное ядро ARM1176JZF-S включает в себя целочисленное ядро, базирующееся на архитектуре ARM ARM11 v6. Оно поддерживает системы команд ARM и Thumb™, технологию Jazelle для обеспечения возможности прямого выполнения байт-кодов Java, а также включает набор SIMD команд для использования в программах цифровой обработки сигналов, позволяющих обрабатывать 16- или 8-разрядные данные в 32-разрядных регистрах. Использованное в СБИС СНП-ВП процессорное ядро ARM1176JZF-S основано на его версии r0p7 и имеет конфигурацию, характеризующуюся наличием следующих свойств:

- высокоскоростной интерфейс Advanced Microprocessor Bus Architecture (AMBA) Advanced Extensible Interface (AXI),
- целочисленное ядро с интегрированной логикой EmbeddedICE-RT,
- восьмиступенчатый конвейер,
- предсказание переходов со стеком возврата,
- режим с малой задержкой обработки прерываний,
- интегрированные сопроцессор отладки CP14 и системный сопроцессор CP15,
- устройства управления памятью команд и данных Memory Management Units (MMU), работающие с буферами MicroTLB и единым главным буфером Main TLB
- кэш память команд первого уровня объемом 16Кбайт,
- кэш память данных первого уровня объемом 16Кбайт, включая неблокирующий кэш данных с системой Hit-Under-Miss (HUM),
- виртуально индексируемые и физически адресуемые кэши,
- 64-разрядный интерфейс к обоим блокам кэш памяти,
- векторный сопроцессор для вычислений с плавающей запятой(VFP),
- TCM (Tightly Coupled Memory) память команд 16К,
- TCM (Tightly Coupled Memory) память данных 16К,
- интерфейс ПДП к памятям TCM,
- поддержка трассировки,
- система отладки на базе JTAG.

Структурная схема ядра представлена на рисунке (см. Рисунок 5.2).

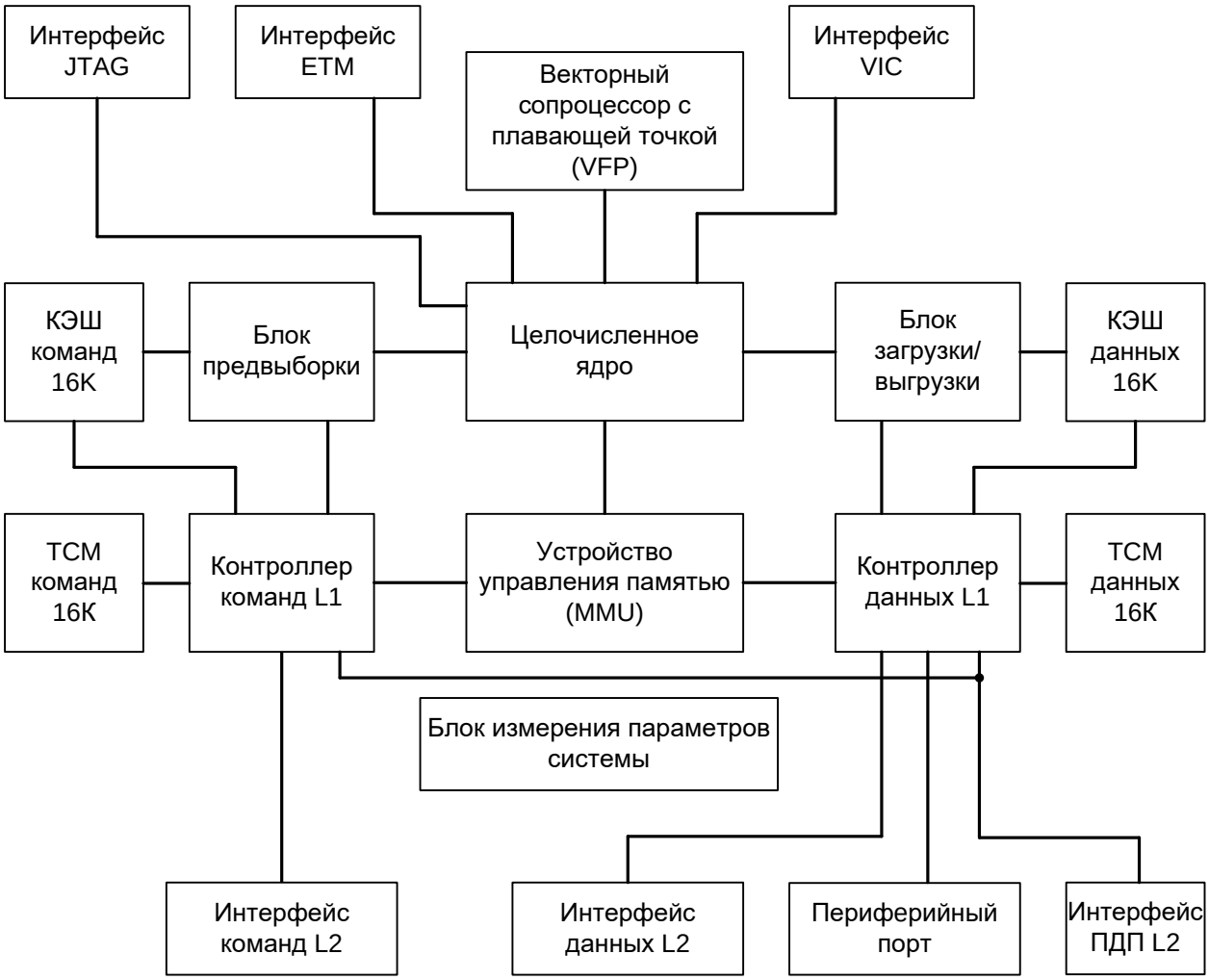


Рисунок 5.2 – Структурная схема процессорного ядра ARM1176JZF-S

Процессорное ядро ARM1176JZF-S включает все нововведения архитектуры ARMv6, имеющей следующие особенности:

- Команды Store/Load Exclusive для байтов, полуслов и двойных слов, а также новую команду Clear Exclusive.
- Реальная команда no-operation («нет операции») и команда yield.
- Дополнительные физические регистры для динамического переименования архитектурных регистров.
- Ограничение на размер кэша при помощи CP15 c1. Размер кэша может быть ограничен значением 16 КБайт для *операционных систем* (ОС), которые не поддерживают раскраску страниц.
- Изменения переразметки битов в TEX. Дескрипторы таблицы страниц MMU ARMv6 используют большое количество битов для описания параметров внутреннего и внешнего кэшей. Едва ли в реальности какому-нибудь приложению понадобятся все эти возможности одновременно. Таким образом, оказывается возможным сконфигурировать процессор ARM1176JZF-S для поддержки лишь ограниченного набора опций при помощи механизма переразметки TEX. Это подразумевает, что записи в таблице страниц имеют косвенный характер.

- Таблица кодирования TEX CB содержит два бита, связанные с таблицей страниц, состояние которых управляется ОС. Для совместимости на уровне кода с существующими версиями ОС для ARMv6 такая структура обеспечивает работу с различными режимами функционирования MMU. Данный подход называется TEX Remap и управляется битом [28] в регистре 1 CP15.
- Изменен порядок использования бита AP. В процессоре ARM1176JZF-S код b111 в APX и AP[1:0] означает доступ «только для чтения» в привилегированном и пользовательском режимах. AP[0] указывает тип отмены доступа, ошибку бита доступа, когда CP15 c1[29] равно 1.

Более подробное описание процессорного ядра ARM1176JZF-S можно найти в документе ARM DDI 0301F «ARM1176JZF-S. Revision r0p7. Technical Reference Manual». Описание архитектуры ARMv6 можно найти в документе ARM DDI 0100I «ARM Architecture Reference Manual»

5.1.1 Контроллер кэш-памяти второго уровня L220

Для увеличения производительности процессорного ядра применяется кэш память второго уровня объёмом 128Кб. В СБИС СНП-ВП применен контроллер кэш-памяти L220 фирмы ARM (используется версия r1p7 контроллера). Ниже приведены его технические характеристики:

- размер кэш памяти 128Кб
- кэш-память физически адресуемая
- программно настраиваемая ассоциативность кэш-памяти (от direct-mapped до 8-way)
- длина кэш строки фиксирована и составляет 32 байта
- поддержка атрибутов областей памяти ARMv6
- контроллер имеет набор программно доступных регистров, позволяющих выполнять настройку параметров контроллера

Более подробное описание контроллера кэш-памяти L220 можно найти в документе ARM DDI 0329I «L220 Cache Controller. Revision r1p7.

5.1.2 Интерфейс ПДП и «ближняя» память ядра (TCM)

Ближняя память ядра TCM (Tightly Coupled Memory) является внутренней памятью ядра ARM с временем доступа равной времени доступа в кэш-память первого уровня. Эта память позволяет ядру получить быстрый доступ к критичным блокам памяти без неопределенности, вносимой обычной кэш-памятью. Ядро ARM содержит раздельные TCM памяти команд и данных объемом 16Кб каждая.

TCM память может использоваться для хранения критически важных подпрограмм, например процедур обработки прерываний, подпрограмм реального времени, и иных случаем когда кэш-память может внести непредсказуемую задержку исполнения. TCM память данных может быть использована для хранения критических данных, таких как локальные параметры, стеки прерываний, и иных плохо подходящих для кэширования данных.

TCM память может быть отражена в любом месте адресного пространства ядра ARM, при этом в TCM не дублируются данные основной памяти, расположенной по тем же физическим адресам. Включение/выключение, настройка отражения, а также настройка других параметров TCM выполняется с помощью регистров системного сопроцессора CP15.

Для эффективной подкачки данных в TCM память используются 2 встроенных в ядро ARM канала ПДП. Каналы используют отдельный 64-х разрядный AXI интерфейс и управляются с помощью регистров системного сопроцессора CP15. Каналы ПДП имеют следующие свойства:

- каналы ПДП работают в фоновом к ядру режиме
- одновременно может быть активен только один канал ПДП

- состояние передачи можно контролировать программно
- каналы программируются используя виртуальные адреса
- каждый канал ПДП может работать как ТСМ данных, так и с ТСМ команд.

5.1.3 Блоки поддержки отладки — ETM11, ETB11 и JTAG интерфейс

СБИС СНП-ВП имеет ряд блоков для аппаратной поддержки отладки программ на своём процессорном ядре ARM. Помимо средств отладки самого ядра используются также блоки Embedded Trace Macrocell (ETM11) и Embedded Trace Buffer (ETB11), которые взаимодействуют через интерфейс JTAG с внешним устройством — аппаратным отладчиком RealView ICE (далее RVI).

5.1.3.1 Устройство блоков аппаратной поддержки отладки

Аппаратные средства поддержки отладки имеют структуру, показанную ниже (см. Рисунок 5.3).

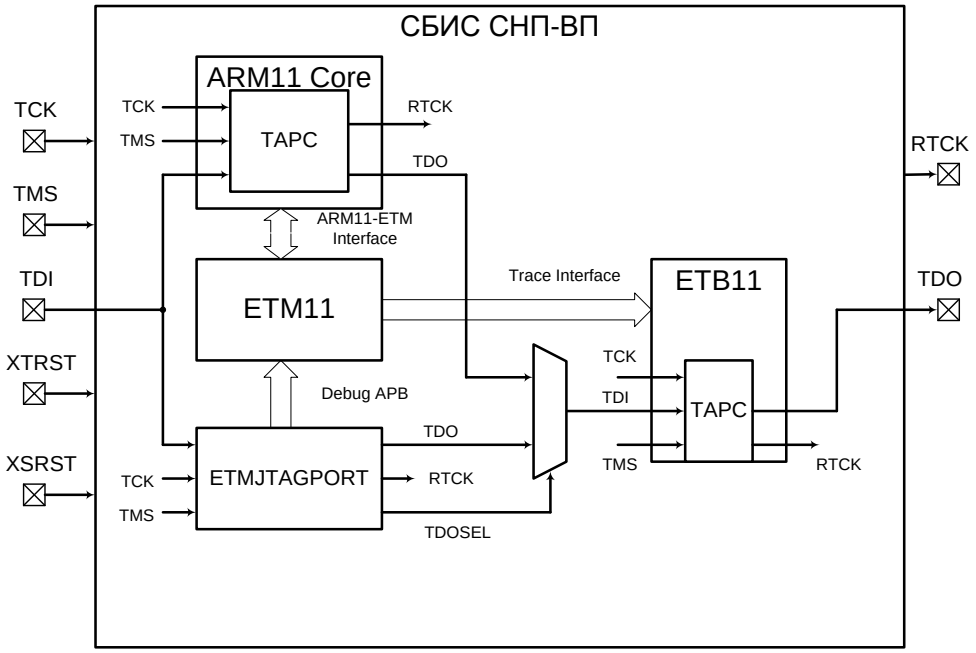


Рисунок 5.3 — Схема внутрикристальных аппаратных средств отладки программ на ядре ARM

Блоки соединены в цепь от входа данных порта JTAG (TDI) до выхода данных (TDO).

ARM11 Core — ядро ARM1176JZF-S, содержащее собственный TAPC (Test Access Port Controller). Описание принципов работы с TAPC ядра ARM11 дано в документе ARM DDI 0301G «ARM1176JZF-S Technical Reference Manual» (Chapter 14).

ETMJTAGPORT — блок, преобразующий команды JTAG в команды записи и чтения регистров ETM11.

Debug APB — отладочная шина между ETMJTAGPORT и ETM11 — она не присоединена к общей шинной системе СБИС, регистры ETM11 не доступны программно для ядра ARM.

ETM11 (Embedded Trace Macrocell) — блок, управляющий трассировкой программ ядра ARM11 через специализированный интерфейс. Спецификация блока дана в документе ARM DDI 0318C «CoreSight ETM11 Technical Reference Manual».

ARM11-ETM Interface — интерфейс между ядром ARM11 и ETM11, по которому передаётся отладочная информация.

ETB11 (Embedded Trace Buffer) — буфер отладочной информации. Поскольку интерфейс JTAG относительно медленный, то в ходе выполнения программ отладочная информация

должна скапливаться в буфере. Блок ETB11 служит для временного хранения данных, читаемых через JTAG. Используемая в СБИС СНП-ВП конфигурация ETB11 не поддерживает функцию трассировки памяти. Спецификация блока дана в документе ARM DDI 0275D «ETB11 Technical Reference Manual».

Trace Interface — интерфейс, по которому блок ETM11 передаёт данные блоку ETB11.

Блоки аппаратной поддержки отладки имеют следующие версии реализации:

- Embedded Trace Macrocell — блок ETM11CS версии r0p2,
- Embedded Trace Buffer — блок ETB11 версии r0p1 и размером буфера 8Кб.

Описанная выше схема является типовой для микропроцессорных систем на основе ядра ARM11. Протоколы обмена между СБИС и внешними средствами отладки специфицированы компанией ARM Limited. Отладку можно производить с хост компьютера с помощью поставляемого компанией ARM Limited пакета RealView ICE, включающего в себя

- аппаратный отладчик — устройство, подключаемое к инструментальной плате и ПК,
- блок питания аппаратного отладчика,
- все необходимые соединительные кабели,
- компакт-диск с ПО для поддержки работы с аппаратным отладчиком,
- комплект документации.
-

5.1.3.2 Устройство порта JTAG и подключение к внешнему аппаратному отладчику

Порт JTAG в микросхеме используется для двух целей:

- для контроля на этапе производства микросхем в режиме тестирования (вход TSTMODE=1),
- для отладки программ ARM в рабочем режиме (вход TSTMODE=0).

В СБИС СНП-ВП используются расширенный вариант порта JTAG, включающий в себя следующие выводы микросхемы:

Таблица 5.1 — Выводы микросхемы, используемые для отладки

Вывод	Тип буфера	Примечание
TDI	in	вход данных тестового порта
TDO	out	выход данных тестового порта
TMS	in	выбор режима тестирования
TCK	in	тактовый сигнал тестового порта
RTCK	out	выход синхросигнала TCK тестового порта
XTRST	in (pullup 33K)	сброс тестового порта
XSRST	inout (pullup 33K)	сброс программный от отладчика

Аппаратный отладчик подключается к плате через 20-пиновый IDC разъем с шагом выводов 2,54 мм. Рекомендуется размещать этот разъем на плате как можно ближе к микросхеме, так как это минимизирует ухудшение параметров сигналов вследствие влияния длинных дорожек на плате.

На рисунке (см. Рисунок 5.4) приведена типовая схема подключения JTAG разъема к микросхеме СНП-ВП. Разъяснения касательно схемы сброса и подключения входа сброса XTRST даны в соответствующей главе (глава 13.4).

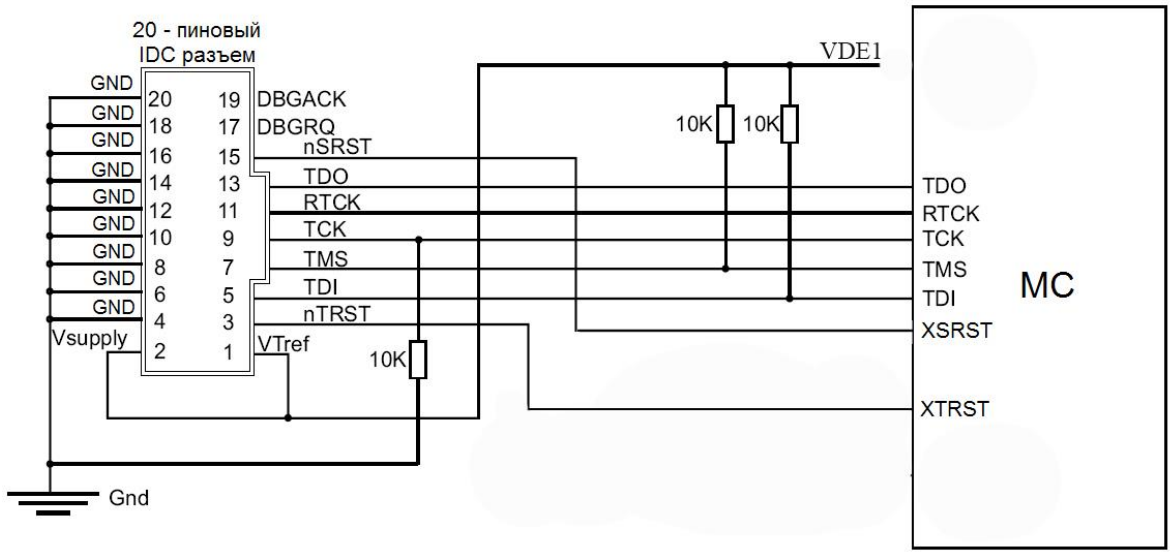


Рисунок 5.4 — Схема типового подключения микросхемы по JTAG интерфейсу

К описанному разъему подключают кабель аппаратного отладчика. Ниже в таблице приведено описание контактов JTAG разъема кабеля отладчика RVI:

Таблица 5.2 — Описание контактов JTAG разъема блока RealView ICE для использования с микросхемой СНП-ВП

Контакт	I/O	Описание
DBGACK	-	Этот сигнал не поддерживается микросхемой. Можно не использовать, либо подключить к Земле.
DBGRQ	-	Этот сигнал не поддерживается микросхемой. Можно не использовать, либо подключить к Земле.
GND	-	Земля.
nSRST	inout	Внешний программный сброс микросхемы от отладчика. Сигнал с активным низким уровнем. В RVI используются аналоговый ключ для управления этим выходом. В активном состоянии выход коммутируется с Землей, в неактивном — через pullup резистор номинала 4,7 КОм с питанием VTref. Pullup резистор нужен, чтобы RVI мог отслеживать сброс микросхемы, инициированный не от него. Данный вывод содержит встроенный pullup резистор для исключения случайного сброса при отсутствии подключения к RVI.
nTRST	out	Сброс тестового порта в микросхеме от отладчика. Сигнал с активным низким уровнем. В RVI используются аналоговый ключ для управления этим выходом. В активном состоянии выход коммутируется с Землей, в неактивном — с питанием VTref. Данный вывод содержит встроенный pullup резистор для исключения случайного сброса при отсутствии подключения к RVI.
TCK	out	Тактовый сигнал тестового порта от отладчика. В RVI используются аналоговый ключ для управления этим выходом. Для генерации логического нуля выход коммутируется с Землей, для генерации логической 1 — с питанием VTref. Частота определяется автоматически в зависимости от длины кабеля и задержек – см. описание RTCK. Этот сигнал должен иметь pulldown резистор.
RTCK	in	Возвращенный сигнал TCK. В микросхеме используется ядро RISC процессора ARM1176JZF-S, которое предполагает синхронизацию тактового сигнала TCK со своим внутренним системным синхросигналом. Для поддержки отладки таких ядер RVI имеет режим автоподстройки частоты TCK (Adaptive Clocking Mode). В этом режиме изменение выходов TCK, TDI и TMS не происходит до получения возвращенного фронта RTCK. Защелкивание данных на TDO происходит относительно RTCK.
TDI	out	Вход данных тестового порта от отладчика. В RVI используются аналоговый ключ для управления этим выходом. Для генерации логического нуля выход коммутируется с землей, для генерации логической 1 — с питанием VTref. Этот сигнал должен иметь pullup резистор.
TMS	out	Выбор режима тестирования от отладчика. В RVI используются аналоговый ключ для управления этим выходом. Для генерации логического нуля выход коммутируется с землей, для генерации логической 1 — с питанием VTref. Этот сигнал должен иметь pullup резистор.
TDO	in	Выход данных тестового порта к отладчику. Этот сигнал должен иметь pullup резистор.
Vsupply	in	Контакт не подключается к RVI. Контакт зарезервирован для поддержки другого отладочного оборудования, которое может питаться от отлаживаемой системы.
VTref	in	Напряжение смещения. RVI определяет наличие питания в отлаживаемой системе, питание должно быть как минимум 0,628 В. Обычно Vtref подключается к питанию Vdd платы. Внутри RVI имеется 10 КОм pulldown резистор на этом сигнале. Используется для сопряжения RVI с системами с различными уровнями логических 0 и 1.

Для корректной отладки по JTAG интерфейсу необходимо иметь надежное соединение между RVI отладчиком и платой.

Частота обмена по JTAG интерфейсу зависит от длины кабеля.

- приведенная схема разводки применима для коротких кабелей (около 30 см). В этом случае максимальная частота обмена не может быть выше единиц МГц.
- для кабелей средней длины мощности выходных буферов микросхемы сигналов RTCK, TDO и XSRST может быть недостаточно из-за высокой емкостной нагрузки от кабеля. В этом случае необходимо рассматривать кабель как длинную линию и добавлять на плату дополнительные буфера с высокой нагрузочной способностью для этих сигналов и согласующие резисторы, чтобы избежать отражений сигнала в кабеле.
- для очень длинных кабелей в RVI используется преобразование сигналов JTAG порта в LVDS сигналы и специальный переходник, подключаемый в JTAG разъем платы для выполнения обратного преобразования (LVDS probe).

Более подробное описание RVI отладчика и способы его подключения к микросхеме можно найти в документе ARM DUI 0155H «RealView ICE and RealView Trace. Version 3.2. User Guide» (Chapter 9, Appendix A).

5.2 Системный коммутатор шин HPMS

Системный коммутатор шин HPMS (High Performance Matrix System) работает согласно стандарту AMBA AXI фирмы ARM. Коммутатор HPMS работает с 64-разрядными шинами, он осуществляет коммутацию 6-ю мастеров и 6-ти управляемых устройств и является многоуровневым, т.е. обеспечивает одновременную передачу данных всеми мастерами при условии работы с разными управляемыми устройствами. Также системный коммутатор позволяет программно (запись внутренних конфигурационных регистров SGPV) настраивать приоритеты мастеров при обращении к одному управляемому устройству.

ЗАМЕЧАНИЕ.

После окончания инициализации СБИС СНП-ВП приоритеты портов коммутатора настроены по умолчанию, т.е. для работы коммутатора нет необходимости проводить его инициализацию и настройку. Дополнительная конфигурация коммутатора нужна только в случае желания изменить настройки схемы арбитража.

На рисунке ниже приведена схема включения коммутатора HPMS, а также доступные каждому мастеру управляемые устройства (см. Рисунок 5.5).

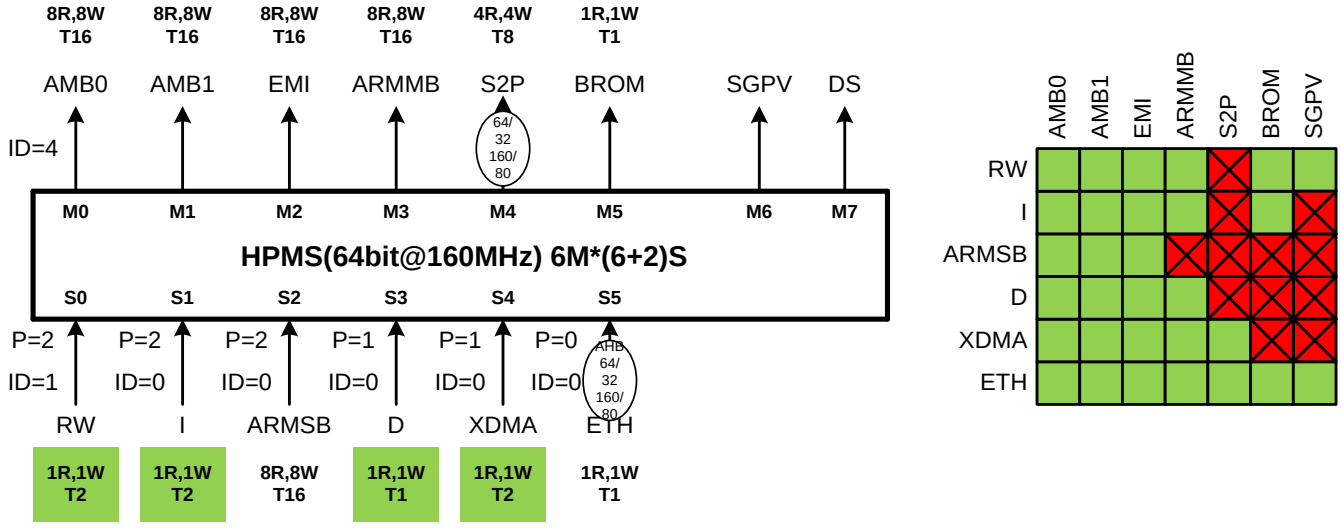


Рисунок 5.5 - Схема включения коммутатора HPMS

Шина AXI представляет собой интерфейс типа точка-точка – каждое master устройство должно быть подключено к slave-порту (коммутатора или slave устройства), а каждое slave-устройство должно быть подключено к master-порту (коммутатора или master устройства). Таким образом, для соединения нескольких master и slave устройств используется коммутатор: master устройства подключаются к slave-портам коммутатора, а slave устройства подключаются к master-портам коммутатора.

Помимо обычных slave устройств, есть еще два специальных – DS (Default Slave) и SGPV (Global Programmer View).

DS отвечает на запросы по несуществующим адресам, т.к. любая AXI транзакция должна заканчиваться.

SGPV содержит программно доступные регистры для настройки параметров коммутатора.

Каждый порт коммутатора может иметь переходники, называемые мостами. Например, мост перехода с шины AXI на шину AHB, изменение разрядности шины с 64 бит до 32 бит, и

т.д. Также для каждого slave-порта коммутатора имеется набор параметров, настраиваемых на этапе разработки СБИС. В таблице ниже приведены значения параметров этих портов.

Таблица 5.3- Параметры slave-портов коммутатора HPMS

	Название порта	**Конвейерный регистр**	**Активные запросы чтения**	**Активные запросы записи**	**Общее кол-во запросов**	**Приоритет по умолчанию**	**Схема защиты от взаимоблокировки (deadlock)**	**Дополнительно**
S0	RW	-	1	1	2	2	SingleSlave	
S1	I	-	1	1	2	2	SingleSlave	
S2	ARMSB	+	8	8	16	2	SingleSlave	
S3	D	-	1	1	2	1	SingleSlave	
S4	XDMA	-	1	1	2	1	SingleSlave	
S5	ETH	-	1	1	1	0	SingleSlave	Мост 64/32, AHB-AXI

Для каждого master-порта коммутатора тоже имеется набор параметров, настраиваемых на этапе разработки СБИС. В таблице ниже приведены значения параметров этих портов.

Таблица 5.4- Параметры master-портов коммутатора HPMS

	Название порта	**Конвейерный регистр**	**Активные запросы чтения**	**Активные запросы записи**	**Общее кол-во запросов**	**Схема арбитража**	**Дополнительно**
M0	AMB0	+	8	8	16	ProgLRG	
M1	AMB1	+	8	8	16	ProgLRG	
M2	EMI	-	8	8	16	ProgLRG	
M3	ARMMB	+	8	8	16	ProgLRG	
M4	S2P	+	4	4	8	ProgLRG	Мост 64/32
M5	BROM	-	1	1	1	ProgLRG	

5.2.1 Параметры портов коммутатора HPMS

5.2.1.1 Конвейерный регистр

Общий параметр для slave и master портов коммутатора. В общем случае коммутатор не буферизует запрос, но если на порте коммутатора стоит конвейерный регистр, то запрос буферизуется, т.е. добавляется 1 такт задержки для первого запроса (latency).

5.2.1.2 Активные запросы чтения

Общий параметр для slave и master портов коммутатора. Активным запросом чтения называется запрос, по которому master устройство выдало адрес чтения, но не получило всех считываемых данных.

Для slave-портов этот параметр показывает количество активных запросов чтения от подключенного master устройства, которое может принять коммутатор.

Для master-портов этот параметр показывает количество активных запросов чтения, которое может выдать коммутатор к подключенному slave-устройству.

5.2.1.3 Активные запросы записи

Общий параметр для slave и master портов коммутатора. Активным запросом записи называется запрос, по которому master устройство выдало адрес записи, но не закончило передачу записываемых данных и не получило подтверждение записи данных от slave устройства.

Для slave-портов этот параметр показывает количество активных запросов записи от подключенного master устройства, которое может принять коммутатор.

Для master-портов этот параметр показывает количество активных запросов записи, которое может выдать коммутатор к подключенному slave-устройству.

5.2.1.4 Общее количество запросов

Суммарное количество запросов, которое может выдать или принять коммутатор по порту, независимо от их типа.

5.2.1.5 Схема защиты от взаимоблокировки

Параметр slave-портов коммутатора. Все slave-порты коммутатора имеют схему защиты от взаимоблокировок (deadlock'ов). Идея заключается в том, что master-устройство может иметь активные запросы (записи и/или чтения) только к одному slave устройству (схема SingleSlave). Запрос, адресованный другому slave устройству, принимается только после завершения всех текущих активных запросов. На рисунке ниже приведен алгоритм работы схемы защиты от взаимоблокировок внутри slave порта коммутатора (см. Рисунок 5.6).

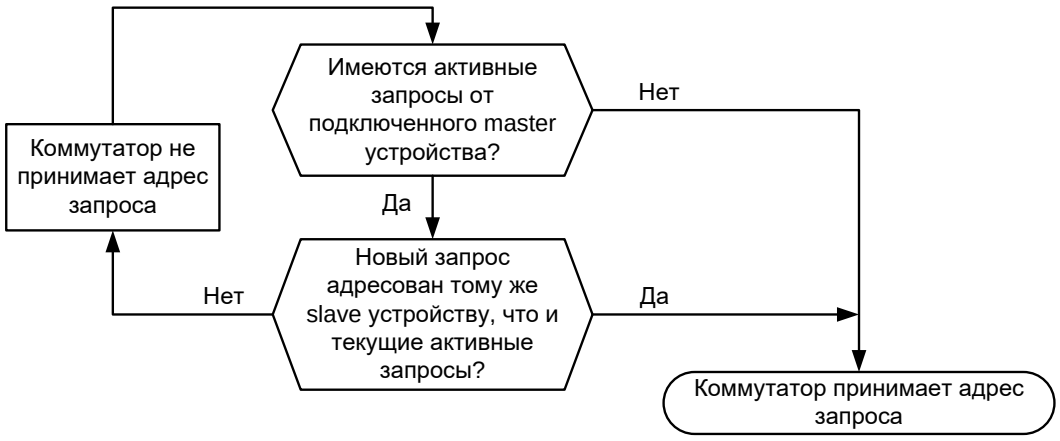


Рисунок 5.6 - Алгоритм работы схемы защиты от взаимоблокировок

5.2.1.6 Доступные управляемые устройства

Параметр slave-портов коммутатора. Slave устройства, подключенные к перечисленным master-портам коммутатора, доступны для master устройства, подключенного к данному slave-порту коммутатора (см. Рисунок 5.5). Например, процессорное ядро ARM может получить доступ ко всем slave устройствам, подключенным к коммутатору, кроме S2P. Другое master устройство ARMSB (NMU часть СБИС) может получить доступ только к банкам внутренней памяти AMBx и внешней памяти EMI.

5.2.1.7 Схема арбитража коммутатора HPMS

Параметр master-портов коммутатора. Т.к. одновременно несколько master-устройств могут обратиться к одному slave-устройству, каждый master-порт коммутатора имеет схему арбитража, а каждый slave-порт имеет определенный приоритет. Начальные значения приоритетов указаны в таблице выше (Таблица 5.3). Коммутатор HPMS на всех master-портах имеет

одинаковую схему арбитража, работающую по алгоритму Least Recently Granted (LRG). В общем, этот алгоритм предполагает, что арбитраж выигрывает master-устройство, которое было обслужено наиболее давно.

Запросы на чтение (AR) и запросы на запись (AW) используют раздельные схемы арбитража. Приоритеты записи и чтения могут быть настроены по-разному.

Независимо от особенностей алгоритма работы арбитража, схема в текущем такте принимает решение по арбитражу на следующий такт. Если в текущем такте нет активных запросов от master устройств, то применяется арбитраж по умолчанию (арбитраж выигрывает master-устройство, подключенное к slave-порту с максимальным приоритетом).

Алгоритм арбитража LRG. Каждый master-порт имеет регистры, в которых задаются приоритеты каждого slave-порта, иными словами master-устройства, подключенного к этому порту. По-умолчанию master-устройства имеют приоритет, указанный в таблице выше. Приоритеты могут быть изменены путем записи соответствующих конфигурационных регистров SGPV. На рисунке ниже приведен пример работы LRG алгоритма арбитража (см. Рисунок 5.7).

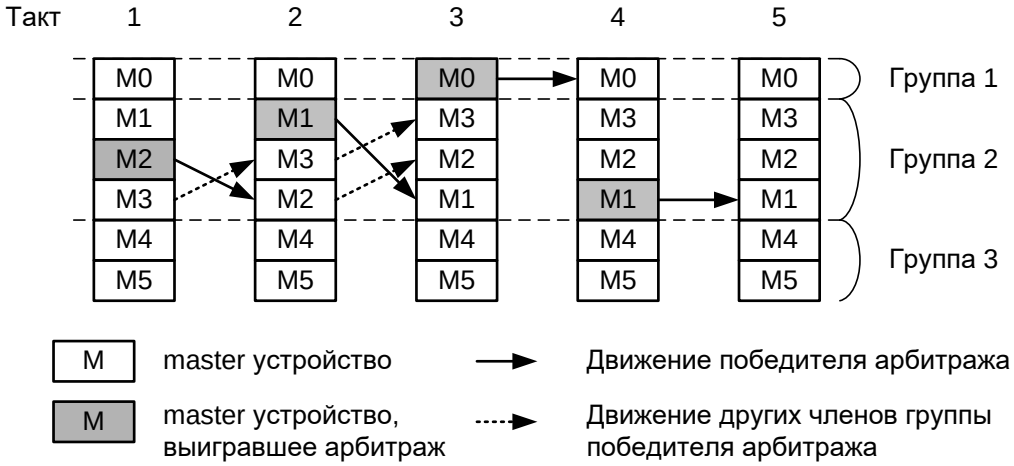


Рисунок 5.7 - Пример работы LRG алгоритма арбитража

Все master-устройства с одинаковым приоритетом формируют группу. В результате арбитража master-устройство может менять положение в группе, но не может покинуть её. Также новое master-устройство не может присоединиться к группе. Арбитраж выигрывает группа, в которой находятся master-устройства с наивысшим приоритетом, и которые пытаются отослать запрос, а внутри группы выигрывает master-устройство, которое дольше всех не обслуживалось (см. Рисунок 5.7). После того как master-устройство выиграло арбитраж, внутри группы оно начинает иметь наименьший приоритет (на рисунке опускается вниз) – это гарантирует, что другие master-устройства этой группы тоже получат доступ к slave-устройству, т.к. их приоритет внутри группы повышается (на рисунке поднимаются вверх). Если присвоить всем master-устройствам разные приоритеты, то схема арбитража вырождается в схему с фиксированными приоритетами, т.к. группы имеют только по одному члену и master-устройства не конкурируют внутри группы. Другой крайний случай, когда всем master-устройствам присваивается одинаковый приоритет – схема арбитража вырождается в чисто LRG схему арбитража, т.к. имеется всего одна группа.

5.2.1.8 Преобразователь протокола

Параметр master-портов коммутатора. Master порты коммутатора могут иметь блоки преобразования протоколов, которые преобразуют запросы AXI в запросы AHB или APB. Блоки преобразования протокола выполняют буферизацию запросов внутри себя.

5.2.2 Регистровая модель коммутатора HPMS

Программно доступные регистры SGPV коммутатора HPMS расположены в области памяти 0xC0100000 - 0xC01FFFFF общим размером 1 Мб.

Каждое master-устройство подключается к определенному slave-порту HPMS-коммутатора, каждое slave-устройство подключается к определенному master-порту (Таблица 5.5). Соответствие порта (Таблица 5.5) и четырехкилобайтного регистрового блока взаимно однозначно, AMIBs (AXI Master Interface Blocks) – регистровые блоки для master-портов, ASIBs (AXI Slave Interface Blocks) – регистровые блоки для slave-портов HPMS-коммутатора.

Таблица 5.5 - Карта адресов регистровой модели HPMS-коммутатора

Address offset from base	**Name**
ASIBs	
0x00004700	ETH
0x00004600	XDMA
0x00004500	D
0x00004400	SB
0x00004300	I
0x00004200	RW
AMIBs	
0x00008000	SGPV
0x00007000	BROM
0x00006000	S2P
0x00005000	MB
0x00004000	EMI
0x00003000	AMB1
0x00002000	AMB0
0x00001000	ID registers
0x00000000	Address control register

Существуют следующие ограничения при доступе к регистрам SGPV:

- Поддерживаются только AXI-транзакции с AxSIZE = 32;
- Поддерживаются только secure-транзакции;
- Не поддерживается режим интерливинга;
- Поддерживаются только выровненные транзакции.

В таблице ниже приведены регистры из блоков ASIB,

Адрес конкретного регистра вычисляется по формуле **0xC0100000** + **Address offset from base** + ***Address offset***

Таблица 5.6 - Описание регистров для ASIB HPMS-коммутатора

Address offset	Type	Width	Reset Value	Name	Description
0x00C	-	-	-	-	Не используется.
0x020	RW	3	4	sync_mode	Не используется в функциональном режиме, запись запрещена.
0x024	RW	1	1	fn_mod2	Не используется.
0x028	RW	3	0	fn_mod_ahb	Не используется.
0x02C – 0x03C	-	-	-	-	Не используется.
0x040	RW	4	0	wr_tidemark	Насколько должно быть заполенено FIFO для транзакций записи прежде, чем начнется чтение из него (передача транзакции записи дальше по интерфейсу), актуально только для ETH-порта. 0 – опция отключена; 1 – в FIFO должно содержаться одна или больше одной транзакции; 2 - в FIFO должно содержаться две или больше двух транзакций; 3 - в FIFO должно содержаться три или больше трех транзакций; 4 - в FIFO должно содержаться четыре или больше четырех транзакций. Запись значений больше 4 запрещена, т.к. размер FIFO – 4.
0x044 – 0x0FC	-	-	-	-	Не используется.
0x100	RW	4	см. Таблица 5.3	read_qos	Приоритет для канала чтения.
0x104	RW	4	см Таблица 5.3	write_qos	Приоритет для канал записи.
0x108	RW	2	0	fn_mod	Запись 1 в 0-й бит разрешает только один активный запрос чтения. Запись 1 в 1-й бит разрешает только один активный запрос записи.
0x10C – 0xFFC	-	-	-	-	Не используется.

В следующей таблице приведены регистры из блоков AMIB. Адрес конкретного регистра вычисляется по формуле **0xC0100000** + **Address offset from base** + ***Address offset***

Таблица 5.7 - Описание регистров для AMIB HPMS-коммутатора

Address offset	Type	Width	Reset Value	Name	Description
0x000	-	-	-	-	Не используется.
0x004	-	-	-	-	Не используется.
0x008	RW	2	0	fn_mod_bm_iss	Устанавливает приоритет выбранного порта в 1. Запись 1 в 0-й бит устанавливает приоритет канала чтения в 1. Запись 1 в 1-й бит устанавливает приоритет канала записи в 1.
0x00C	-	-	-	-	Не используется.
0x020	RW	3	4	sync_mode	Не используется в функциональном режиме, запись запрещена.
0x024	RW	1	1	fn_mod2	Не используется..
0x028	-	-	-	-	Не используется.
0x02C – 0x03C	-	-	-	-	Не используется.
0x040	RW	4	2	wr_tidemark	Насколько должно быть заполенено FIFO для транзакций записи прежде, чем начнется чтение из него (передача транзакции записи дальше по интерфейсу), актуально только для S2P-порта. 0 – опция отключена; 1 – в FIFO должно содержаться одна или больше одной транзакции; 2 - в FIFO должно содержаться две или больше двух транзакций; 3 - в FIFO должно содержаться три или больше трех транзакций; 4 - в FIFO должно содержаться четыре или больше четырех транзакций. Запись значений больше 4 запрещена, т.к. размер FIFO – 4.
0x044	RW	2	-	ahb_cntl	Актуально только для S2P порта - для функции преобразующей AXI-транзакции в AHB-транзакции. Запись 1 в 0-й бит (decerr_en) устанавливает ответ DECERR на невыровненную AXI-транзакцию или в ответ на AXI-транзакцию в которой установлены не все WSTRB. Запись 1 в 1-й бит (force_incr) устанавливает тип AHB-транзакции в тип INCR, вне зависимости от типа AXI-транзакции.
0x100	-	-	-	-	Не используется.
0x104	-	-	-	-	Не используется.
0x108	RW	2	0	fn_mod	Актуально только для S2P порта. Запись 1 в 0-й (read_iss_override) бит разрешает только один активный запрос чтения. Запись 1 в 1-й (write_iss_override) бит разрешает только один активный запрос записи.
0x10C – 0xFFC	-	-	-	-	Не используется.

Далее приведено описание регистров из регистрового блока *Address control register*.

Таблица 5.8 – Описание регистрового блока Address control register HPMS-коммутатора

Address offset	Type	Width	Reset Value	Name	Description
0x000	WO	8	0x00	Remap	После выполнения процедуры начальной загрузки, загрузчик записывает в это поле значение 0x1, происходит процедура переназначения карты памяти: область AMB0 (C000_000h – C003_FFFFh) переносится в область памяти 0000_0000h – 0003_FFFFh. Запись других значений не влияет на работу.
0x004	WO	-	-	-	Не используется.
0x008	WO	1	0	security0	Определяет должна ли транзакция для master-порта AMB0 быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x00C	WO	1	0	security1	Определяет должна ли транзакция для master-порта AMB1 быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x010	WO	1	0	security2	Определяет должна ли транзакция для master-порта EMI быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x014	WO	1	0	security3	Определяет должна ли транзакция для master-порта MB быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x018	WO	1	0	security4	Определяет должна ли транзакция для master-порта S2P быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x01C	WO	1	0	security5	Определяет должна ли транзакция для master-порта BROM быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x110 - 0xFFF	RO	-	-	-	Не используется.

При доступе к регистрам из регистровых блоков ASIB или AMIB создается регистровый блок *ID Registers*, содержащий информацию о HPMS-коммутаторе.

Таблица 5.9 - Описание регистрового блока ID Registers HPMS-коммутатора

Address offset	Type	Width	Reset Value	Name	Description
0x0 – 0xFCC	RO	-	-	-	Не используется.
0xFD0	RO	8	0x04	Peripheral ID4	4KB count (0x0), JEP106 continuation code (0x4).
0xFD4	RO	8	0x00	Peripheral ID5	Не используется.
0xFD8	RO	8	0x00	Peripheral ID6	Не используется.
0xFDC	RO	8	0x00	Peripheral ID7	Не используется.
0xFE0	RO	8	0x01	Peripheral ID0	Part number[7:0] устройства.
0xFE4	RO	8	0xB3	Peripheral ID1	JEP106[3:0] (0xB), part number[11:8] устройства (0x3).
0xFE8	RO	8	0x7B	Peripheral ID2	Версия устройства (0x7), JEP code flag (0x1), JEP106[6:4] (0x3).
0xFEC	RO	8	0x00	Peripheral ID3	Номер последней модификации. Не используется.
0xFF0	RO	8	0x0D	Component ID0	Содержит 0-й байт CoreSight Component ID.
0xFF4	RO	8	0xF0	Component ID1	Содержит 1-й байт CoreSight Component ID.
0xFF8	RO	8	0x05	Component ID2	Содержит 2-й байт CoreSight Component ID.
0xFFC	RO	8	0xB1	Component ID3	Содержит 3-й байт CoreSight Component ID.

5.3 Периферийный коммутатор шин HPMP

Периферийный коммутатор шин HPMP (High Performance Matrix Peripheral) работает согласно стандарту AMBA AXI фирмы ARM. Коммутатор HPMP работает с 32-разрядными шинами. Он осуществляет коммутацию 2-х мастеров, работающих по стандарту AXI, и 8-ми управляемых устройств (четыре AHB устройства, 3 AXI устройства и одно APB устройство) и является многоуровневым, т.е. обеспечивает одновременную передачу данных всеми мастерами при условии работы с разными управляемыми устройствами. Также системный коммутатор позволяет программно (запись внутренних конфигурационных регистров PGPV) настраивать приоритеты мастеров при обращении к одному управляемому устройству.

ЗАМЕЧАНИЕ.

После окончания инициализации СБИС СНП-ВП приоритеты портов коммутатора настроены по умолчанию, т.е. для работы коммутатора нет необходимости проводить его инициализацию и настройку. Дополнительная конфигурация коммутатора нужна только в случае желания изменить настройки схемы арбитража.

На рисунке ниже приведена схема включения коммутатора HPMP, а также доступные каждому мастеру управляемые устройства (см. Рисунок 5.8).

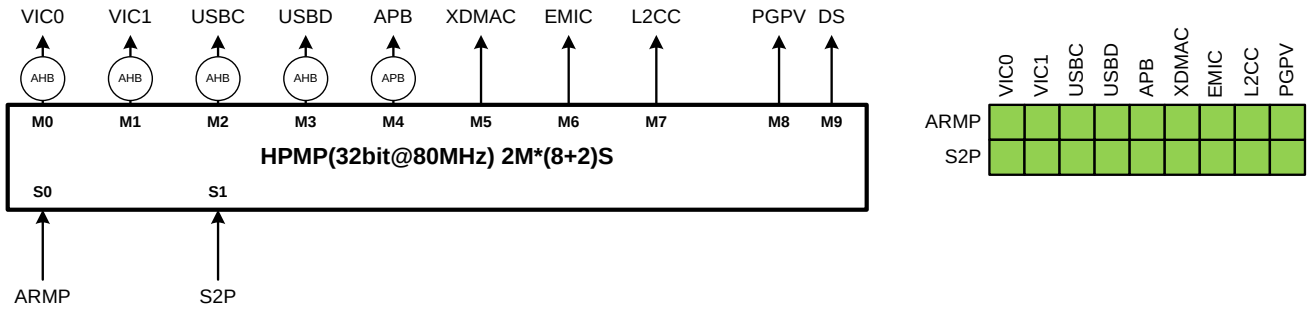


Рисунок 5.8 - Схема включения коммутатора HPMP

Шина AXI представляет собой интерфейс типа точка-точка – каждое master устройство должно быть подключено к slave-порту (коммутатора или slave устройства), а каждое slave-устройство должно быть подключено к master-порту (коммутатора или master устройства). Таким образом, для соединения нескольких master и slave устройств используется коммутатор: master устройства подключаются к slave-портам коммутатора, а slave устройства подключаются к master-портам коммутатора.

Помимо обычных slave устройств, есть еще два специальных – DS (Default Slave) и PGPV (Global Programmer View).

DS отвечает на запросы по несуществующим адресам, т.к. любая AXI транзакция должна быть завершена.

PGPV содержит программно доступные регистры для настройки параметров коммутатора.

Каждый порт коммутатора может иметь переходники, называемые мостами. Например мост перехода с шины AXI на шину AHB, изменение разрядности шины с 64 бит до 32 бит, и т.д. Также для каждого slave-порта коммутатора имеется набор параметров, настраиваемых на этапе разработки СБИС. В таблице ниже приведены значения параметров этих портов.

Таблица 5.10- Параметры slave-портов коммутатора HPMP

	Название порта	**Конвейерный регистр**	**Активные запросы чтения**	**Активные запросы записи**	**Общее кол-во запросов**	**Приоритет по умолчанию**	**Схема защиты от взаимоблокировки (deadlock)**	**Дополнительно**
S0	ARMP	-	1	1	1	1	SingleSlave	
S1	S2P	-	4	4	8	0	SingleSlave	

Для каждого master-порта коммутатора тоже имеется набор параметров, настраиваемых на этапе разработки СБИС. В таблице ниже приведены значения параметров этих портов.

Таблица 5.11- Параметры master-портов коммутатора HPMP

	Название порта	Конвей-ерный регистр	Актив-ные за-просы чтения	Актив-ные за-просы записи	Общее кол-во запросов	Схема арбитража	Дополнительно
M0	VIC0	-	1	1	1	ProgLRG	Мост AXI-AHB
M1	VIC1	-	1	1	1	ProgLRG	Мост AXI-AHB
M2	USBC	-	1	1	1	ProgLRG	Мост AXI-AHB
M3	USBD	-	1	1	1	ProgLRG	Мост AXI-AHB
M4	APB	-	1	1	1	ProgLRG	Мост AXI-APB
M5	XDMAC	-	1	1	1	ProgLRG	
M5	EMIC	-	1	1	2	ProgLRG	
M5	L2CC	-	1	1	2	ProgLRG	

5.3.1 Параметры портов коммутатора HPMP

Параметры коммутатора HPMP идентичны параметрам коммутатора HPMS (см. п. 5.2.1).

5.3.2 Регистровая модель коммутатора HPMP

Программно доступные регистры PGPV коммутатора HPMP расположены в области памяти 0xFFE00000 - 0xFFEFFFFF общим размером 1 Мб.

Каждому порту коммутатора соответствует четырехкилобайтный регистровый блок. Master-устройства подключаются к блокам ASIB (AXI Slave Interface Block), slave-устройства подключаются к блокам AMIB (AXI Master Interface Block). Смещения блоков ASIB и AMIB относительно базового адреса (0xFFE00000) коммутатора HPMP указаны в таблице ниже (Таблица 5.12).

Таблица 5.12 - Карта адресов регистровой модели коммутатора HPMP

Смещение блока	Имя порта
ASIBs:	
0x00004300	S2P
0x00004200	ARMP
AMIBs:	
0x00009000	L2CC
0x00008000	EMIC
0x00007000	XDMAC
0x00006000	APB
0x00005000	USBD
0x00004000	USBC
0x00003000	VIC1
0x00002000	VIC0
0x00001000	ID registers
0x00000000	Address control register

Существуют следующие ограничения при доступе к регистрам PGPV:

- Поддерживаются только AXI-транзакции с AxSIZE = 32;

- Поддерживаются только secure-транзакции;
- Не поддерживается режим интерливинга;
- Поддерживаются только выровненные транзакции.

В таблице ниже приведены регистры из блоков ASIB (Таблица 5.13).

Адрес конкретного регистра вычисляется по формуле **0xFFE00000** + С**мещение блока +** ***Address offset***

Таблица 5.13 - Регистры из ASIB HPMP-коммутатора

Address offset	**Type**	**Width**	**Reset Value**	**Name**	**Description**
0x00C	-	-	-	-	Не используется.
0x020	RW	3	4	sync_mode	Не используется.
0x024	RW	1	1	fn_mod2	Не используется.
0x028	RW	3	0	fn_mod_ahb	Не используется.
0x02C – 0x03C	-	-	-	-	Не используется.
0x040	RW	4	0	wr_tidemark	Не используется.
0x044 – 0x0FC	-	-	-	-	Не используется.
0x100	RW	4	см. Таблица 5.10	read_qos	Приоритет для канала чтения.
0x104	RW	4	см. Таблица 5.10	write_qos	Приоритет для канал записи.
0x108	RW	2	0	fn_mod	Запись 1 в 0-й бит разрешает только одну активную транзакцию чтения. Запись 1 в 1-й бит разрешает только одну активную транзакцию записи.
0x10C – 0xFFC	-	-	-	-	Не используется.

В таблице ниже приведены регистры из блоков AMIB (Таблица 5.14).

Адрес конкретного регистра вычисляется по формуле **0xFFE00000** + С**мещение блока +** ***Address offset***

Таблица 5.14 - Регистры из AMIB HPMP-коммутатора

Address offset	Type	Width	Reset Value	Name	Description
0x000	-	-	-	-	Не используется.
0x004	-	-	-	-	Не используется.
0x008	RW	2	0	fn_mod_bm_iss	Устанавливает приоритет выбранного порта в 1. Запись 1 в 0-й бит устанавливает приоритет канала чтения в 1. Запись 1 в 1-й бит устанавливает приоритет канала записи в 1.
0x00C		-	-	-	Не используется.
0x020	RW	3	4	sync_mode	Не используется.
0x024	RW	1	1	fn_mod2	Не используется.
0x028	-	-	-	-	Не используется.
0x02C – 0x03C	-	-	-	-	Не используется.
0x040	RW	4	2	wr_tidemark	Не используется.
0x044	RW	2	-	ahb_cntl	Актуально только для портов: VIC0, VIC1, USBC, USBD - для функции преобразующей AXI-транзакции в AHB-транзакции. Запись 1 в 0-й бит (decerr_en) устанавливает ответ DECERR на невыровненную AXI-транзакцию или в ответ на AXI-транзакцию в которой установлены не все WSTRB. Запись 1 в 1-й бит (force_incr) устанавливает тип AHB-транзакции в тип INCR, вне зависимости от типа AXI-транзакции.
0x100	-	-	-	-	Не используется.
0x104	-	-	-	-	Не используется.
0x108	RW	2	0	fn_mod	Не используется.
0x10C – 0xFFC	-	-	-	-	Не используется.

Далее приведено описание регистров из регистрового блока *Address control register*.

Таблица 5.15 - Описание регистрового блока Address Control Register HPMP-коммутатора

Address offset	Type	Width	Reset Value	Name	Description
0x000	WO	8	0x00	Remap	Не используется.
0x004	WO	-	-	-	Не используется.
0x008	WO	1	0	security0	Определяет должна ли транзакция для master-порта VIC0 быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x00C	WO	1	0	security1	Определяет должна ли транзакция для master-порта VIC1 быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x010	WO	1	0	security2	Определяет должна ли транзакция для master-порта USBC быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x014	WO	1	0	security3	Определяет должна ли транзакция для master-порта USBD быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x018	WO	15	0	security4	Определяет должна ли транзакция для master-порта APB и для конкретного устройства быть типа secure или non-secure. 0-й бит регистра – для CRGS; 1-й бит регистра – для CRGM; 2-й бит регистра – для ARMSC; 3-й бит регистра – для PRGM; 4-й бит регистра – для I2C; 5-й бит регистра – для CAN; 6-й бит регистра – для ETH; 7-й бит регистра – для RTC; 8-й бит регистра – для DIT; 9-й бит регистра – для WDOG; 10-й бит регистра – для GPIO; 11-й бит регистра – для UART0; 12-й бит регистра – для UART1; 13-й бит регистра – для SPI; 14-й бит регистра – для EIRC. 0 – Non-secure; 1 – secure.
0x01C	WO	1	0	security5	Определяет должна ли транзакция для master-порта XDMAC быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x01C	WO	1	0	security6	Определяет должна ли транзакция для master-порта EMIC быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x01C	WO	1	0	security7	Определяет должна ли транзакция для master-порта L2CC быть типа secure или non-secure. 0 – Non-secure; 1 – secure.
0x110 - 0xFFF	RO	-	-	-	Не используется.

Регистровый блок *ID Registers*, содержащий информацию о HPMP-коммутаторе аналогичен этому же регистровому блоку в HPMS-коммутаторе.

5.4 Блок управления сигналами сброса и синхросигналами CRGS

Блок управления сигналами сброса и синхросигналами CRGS предназначен для управления генерацией синхросигналов и сигналов сброса внутри СБИС СНП-ВП. Блок имеет следующие характеристики:

- управление схемой ФАПЧ
- управление 8 доменами синхросигналов внутри СБИС
- управление внутренними сигналами сброса СБИС на основе 4 источников сброса
- возможность программной настройки параметров через шину APB
- подключение к системной шине СБИС через AMBA APB интерфейс, используются полный 32-битный интерфейс шины данных

5.4.1 Устройство блока CRGS

В таблице ниже приведен список внешних выводов, относящихся к блоку управления сигналами сброса и синхросигналами CRGS.

Таблица 5.16 – Выводы микросхемы, относящиеся к CRGS

Вывод	Тип буфера	Примечание
SCLKXI	in	Входы внешнего синхросигнала от кварца или генератора.
SCLKXO	inout	Входы внешнего синхросигнала от кварца
XSRST	inout	Внешний программный сброс микросхемы.
XRESET	in	Вход основного сброса СБИС СНП-ВП
XTRST	in	Вход сброса тестовой логики
PLLBYPASS	in	Вход выбора режима работы ФАПЧ
SPLLSTOP	out	Выход индикации входа СБИС в STOP режим
SPLL_L	out	Тестовый выход схемы ФАПЧ (не используется)

На рисунке ниже (см. Рисунок 5.9) приведена схема блока CRGS.

Входы SCLKXI и SCLKXO служат для подключения сигнала опорной частоты для ФАПЧ. Для этого можно использовать кварцевый резонатор, или можно использовать внешний генератор, подключив его к входу SCLKXI. Максимальная частота опорного синхросигнала SCLK F_{SYS} = 40МГц. Опорный сигнал SCLK подключается к входу XCLK блока CRGS.

Блок CRGS содержит схему управления ФАПЧ и генерацией внутренних синхросигналов (Clock Generation Block). Опорный сигнал SCLK попадает на этот блок, умножается на схеме ФАПЧ на 16 до частоты $16*F_{SYS}$ и затем делится на 2 для достижения хорошей скважности сигнала. Большая часть внутренней логики блока CRGS, работает на именно от этого синхросигнала CCLK с частотой $8*F_{SYS}$ (максимально 320МГц). Синхросигнал CCLK также распределяется на 8 внутренних доменов синхросигнала. Каждый домен имеет собственный делитель частоты и схему отключения синхросигнала для уменьшения потребления (Clock Gating).

СБИС СНП-ВП имеет вход PLLBYPASS, подача 1 на этот вход выключает схему ФАПЧ, и в этом случае синхросигнал CCLK идентичен синхросигналу SCLK.

Выход SPLLSTOP предназначен для индикации входа микросхемы в режим пониженного энергопотребления (STOP-режим). Также имеются тестовый выход схемы ФАПЧ – SPLL_L. Высокий уровень этого сигнала показывает, что схема ФАПЧ зафиксировалась (PLL locked) и генерируемый синхросигнал на её выходе стабилен.

СБИС СНП-ВП имеет три внешних входа сброса – XRESET, XSRST и XTRST.

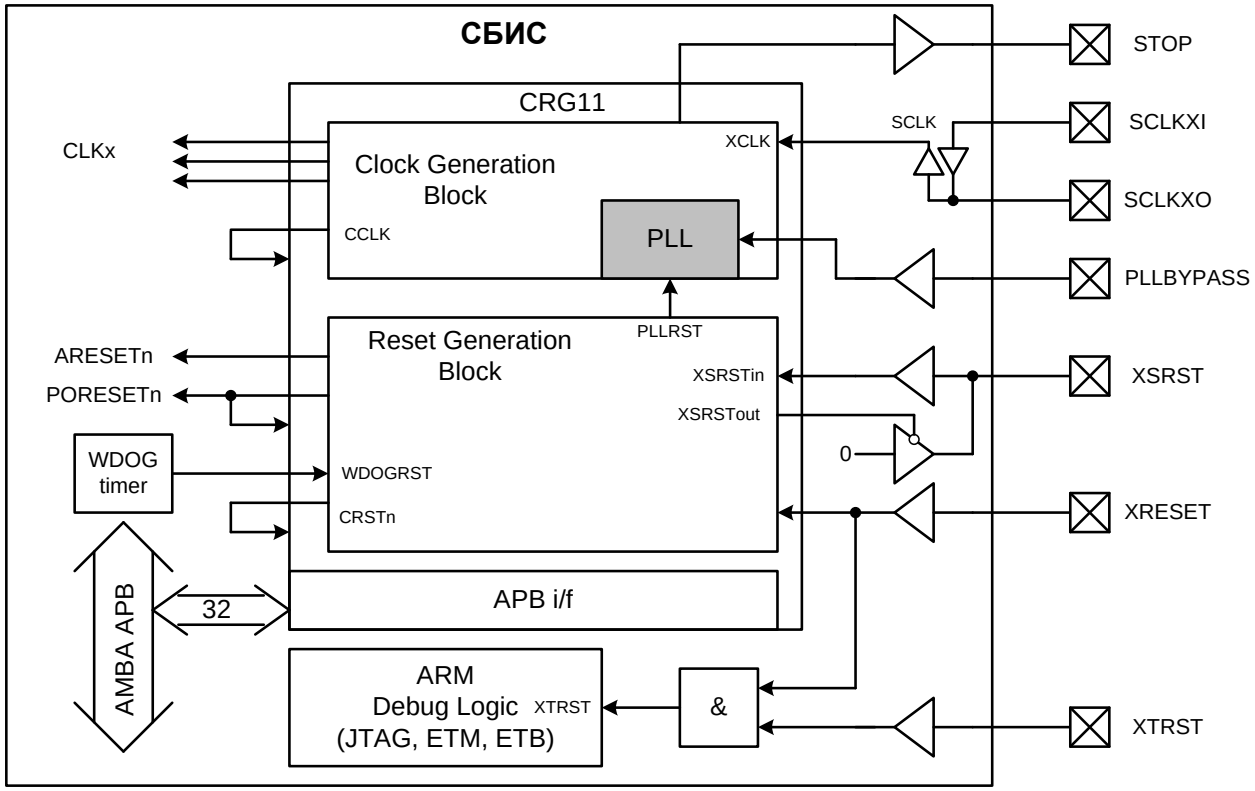


Рисунок 5.9 – Схема блока CRGS

XRESET. Основной сброс микросхемы. Сигнал с активным низким уровнем. Сбрасывается вся микросхема СНП-ВП, в том числе внутренняя логика, обеспечивающая работу механизма отладки. Предполагается, что сигнал XRESET подается на микросхему при возникновении следующих событий:

- включение питания
- нажатие кнопки на плате

XSRST. Внешний программный сброс. Сигнал с активным низким уровнем. Двунаправленный сигнал. Служит для программного сброса микросхемы отладчиком через JTAG интерфейс, а также для отслеживания отладчиком состояния микросхемы. Т.е., если произойдет инициированный изнутри микросхемы сброс (внутренний программный сброс или сброс от сторожевого таймера) или системный сброс (XRESET), то отладчик отследит это событие.

XTRST. Дополнительный сброс, отвечающий за сброс внутренней логики, обеспечивающей работу механизма отладки (ETM11, ETB11, JTAG). Сигнал с активным низким уровнем. Сигнал не подключается к блоку CRGS, изображен для полноты внутренней схемы генерации сигналов сброса. Возникает в двух случаях:

- выставляется аппаратным отладчиком через JTAG интерфейс
- также генерируется при возникновении XRESET, т.к. при полном сбросе системы тестовая логика тоже должна сбрасываться (обеспечивается соединением сигналов XRESET и XTRST по логическому «И» внутри микросхемы)

Кроме внешних сигналов сброса СБИС СНП-ВП имеет два внутренних источника сброса – сторожевой таймер (сигнал WDOGRST) и внутренний программный сброс, инициируемый записью в регистр CRSWR (сигнал SOFTRST). Таким образом, всего имеется 4 источника сброса: 2 внешних (XRESET и XSRST) и два внутренних (WDOGRST и SOFTRST).

При активации одного из источников сброса блок CRGS генерирует внутренние сигналы сброса СБИС. Таких сигналов 5 штук: ARESETn, PORESETn, XSRSTout, CRSTn и PLLRST.

Блок CRGS имеет подключение к периферийной шине APB, это позволяет изменять параметры блока путем записи программно доступных регистров. Можно настраивать следующие параметры:

- менять частоту генерируемых синхросигналов, а также выключать генерацию синхросигнала для каждого домена
- управлять генерацией внутренних сигналов сброса
- менять настройки схемы управления ФАПЧ
- производить программный сброс СБИС
- управлять режимом пониженного энергопотребления (STOP-режим)

Инициализация СБИС происходит с параметрами, заданными на этапе проектирования. После окончания инициализации блок CRGS находится в полностью настроенном состоянии, т.е. нет необходимости в дополнительной настройке без острой необходимости.

Блок CRGS предоставляет огромные возможности по управлению внутренними сигналами сброса и синхросигналами, ошибка в программировании конфигурационных регистров приведет к неработоспособности СБИС, вплоть до полного “зависания”! Выйти из такой ситуации возможно только сбросив СБИС внешним сигналом XRESET.

5.4.2 Генератор сигналов сброса

Как уже говорилось выше, при активизации одного из источников сброса, блок CRGS генерирует внутренние сигналы сброса СБИС. Таких сигналов 5 штук: ARESETn, PORESETn, XSRSTout, CRSTn и PLLRST. В таблице ниже приведено соответствие источников сброса и генерируемых внутренних сигналов сброса.

Таблица 5.17 – Соответствие источников сброса и внутренних сигналов сброса

Внутренний сигнал сброса	Источник сброса					
	Внешний сброс XRESET	Программный сброс XSRST	SOFTRST		WDOGRST	
			бит SWRSTM=1	бит SWRSTM=0	бит WDRSTM=1	бит WDRSTM=0
ARESETn	+	+	+		+	
XSRSTout	+	-	+		+	
PORESETn	+	-	-		-	
CRSTn	+	+	+	-	+	-
PLLRST	+	+	+	-	+	-

Символ “+” означает, что при активации источника сброса, находящегося в заголовке столбца, генерируется соответствующий внутренний сигнал сброса, символ “-” – сигнал сброса не генерируется.

5.4.2.1 Внешний системный сброс XRESET

Активация внешнего сброса XRESET приводит к полному сбросу СБИС, генерируются все внутренние сигналы сброса. Внешний сигнал сброса XRESET должен быть активным на входе СБИС в течении как минимум 3 тактов синхросигнала SCLK.

5.4.2.2 Внешний программный сброс XSRST

Программный сброс XSRST генерируется внешним отладчиком СБИС, например RealView Ice. Сигнал является двунаправленным, т.к. внешний отладчик должен отслеживать состояние микросхемы. Т.е. если сброс будет инициирован изнутри (источники сброса WDOGRST или SOFTRST) или по причине системного сброса (XRESET) – отладчик отследит это событие по генерации внутреннего сигнала сброса XSRSTout. Внешний сигнал сброса XSRST должен быть активным на входе СБИС в течении как минимум 7 тактов синхросигнала CCLK.

Внешний сброс XSRST игнорируется, если активен внутренний сигнал сброса XSRSTout. Также XSRST игнорируется после деактивации XSRSTout в течение времени, зависящего от длительности активного состояния самого сигнала XSRSTout, которое задается полем SRSTMODE[3:0] регистра CRRSC (Таблица 5.18).

Таблица 5.18 – Время игнорирования сброса XSRST и значения SRSTMODE

Поле SRSTMODE[3:0] регистра CRRSC	Количество тактов синхросигнала CCLK, в течение которого активный уровень внешнего сигнала сброса XSRST игнорируется
0000	16 тактов CCLK
0001	16 тактов CCLK
0010	16 тактов CCLK
0011	16 тактов CCLK
0100	16 тактов CCLK
0101	16 тактов CCLK
0110	16 тактов CCLK
0111	24 тактов CCLK
1000	32 тактов CCLK
1001	48 тактов CCLK
1010	64 тактов CCLK
1011	96 тактов CCLK
1100	128 тактов CCLK
1101	192 тактов CCLK
1110	256 тактов CCLK
1111	384 тактов CCLK

5.4.2.3 Внутренний программный сброс SOFTRST

Внутренний программный сброс возникает сразу после установки бита SWRSTREQ регистра CRSWR. Генерация внутренних сигналов сброса зависит от значения бита SWRSTM регистра CRRSC. Если бит установлен, то генерируются все внутренние сигналы сброса, кроме PORESETn. Если не установлен, то не происходит сброса управляющих регистров блока CRGS и схемы ФАПЧ (Таблица 5.17).

5.4.2.4 Сброс от сторожевого таймера WDOGRST

Внутренний сброс, инициируемый блоком сторожевого таймера WDOGRST. Этот сброс генерируется периферийным блоком WDOG (см. п. 5.9). Генерация внутренних сигналов сброса зависит от значения бита WDRSTM регистра CRRSC. Если бит установлен, то генерируются все внутренние сигналы сброса кроме PORESETn. Если не установлен, то не происходит сброса управляющих регистров блока CRGS и схемы ФАПЧ (Таблица 5.17).

5.4.2.5 Внутренний сигнала сброса ARESETn

Сигнал сброса ARESETn используется в СБИС СНП-ВП для сброса процессорного ядра ARM, блоков NMU, SMU, а также всех периферийных устройств NMC и ARM части. Сигнал генерируется при активации любого из четырех источников сброса. Минимальная длительность активного состояния этого сигнала сброса задается полем ARSTMODE[3:0] регистра CRRSC. Реально длительность активного состояния сигнала ARESETn больше, т.к. он деактивируется в момент времени, когда фронты всех синхросигналов, генерируемых CRGS, совпадают и переключаются из нуля в единицу.

Значение поля ARSTMODE[3:0] = b’0111 (96 тактов CCLK) задано на этапе проектирования, уменьшение этого значения может привести к некорректной инициализации СБИС.

5.4.2.6 Внутренний сигнал сброса XSRSTout

Этот сигнал используется для того, чтобы внешний отладчик мог отследить сброс СБИС, инициированный от любого источника сброса, кроме самого внешнего отладчика. Сигнал генерируется при активации любого из трех источников сброса – XRESET, WDOGRST, SOFTRST. Длительность активного состояния этого сигнала сброса задается полем SRSTMODE[3:0] регистра CRRSC. Значение поля SRSTMODE[3:0] = b'0111 (96 тактов CCLK) задано на этапе проектирования. Значение определяется параметрами внешнего отладчика, неправильное задание этого параметра приведет к невозможности отслеживать состояние СБИС внешним отладчиком.

5.4.2.7 Внутренний сигнал сброса PORESETn

Внутренний сигнал сброса, генерируемый только при подаче внешнего сигнала сброса XRESET. Используется для сброса блоков СБИС, которые не сбрасываются при активации других источников сброса. Часть регистров блока CRGS сбрасывается этим сигналом.

5.4.2.8 Внутренний сигнал сброса CRSTn

Этот сигнал сброса используется для сброса внутренней логики блока CRGS. Всегда генерируется при активации внешних источников сброса XRESET и XSRST. Генерирование при активации источников сброса WDOGRST и SOFTRST определяется значением конфигурационных бит SWRSTM и WDRSTM регистра CRRSC.

5.4.2.9 Внутренний сигнал сброса PLLRST

Этот сигнал сброса используется для сброса схемы ФАПЧ. Генерируется в тех же случаях, что и сигнал CRSTn, а также при смене делителей схемы ФАПЧ (т.е. при смене частоты синхросигнала CCLK) и входе СБИС в STOP-режим. Более подробное описание сигнала дано в разделе описания настроек схемы ФАПЧ (см. п. 5.4.3.1).

5.4.3 Генератор синхросигналов

На рисунке ниже приведена схема генератора синхросигналов (Clock Generation Block), входящего в состав блока CRGS (см. Рисунок 5.10).

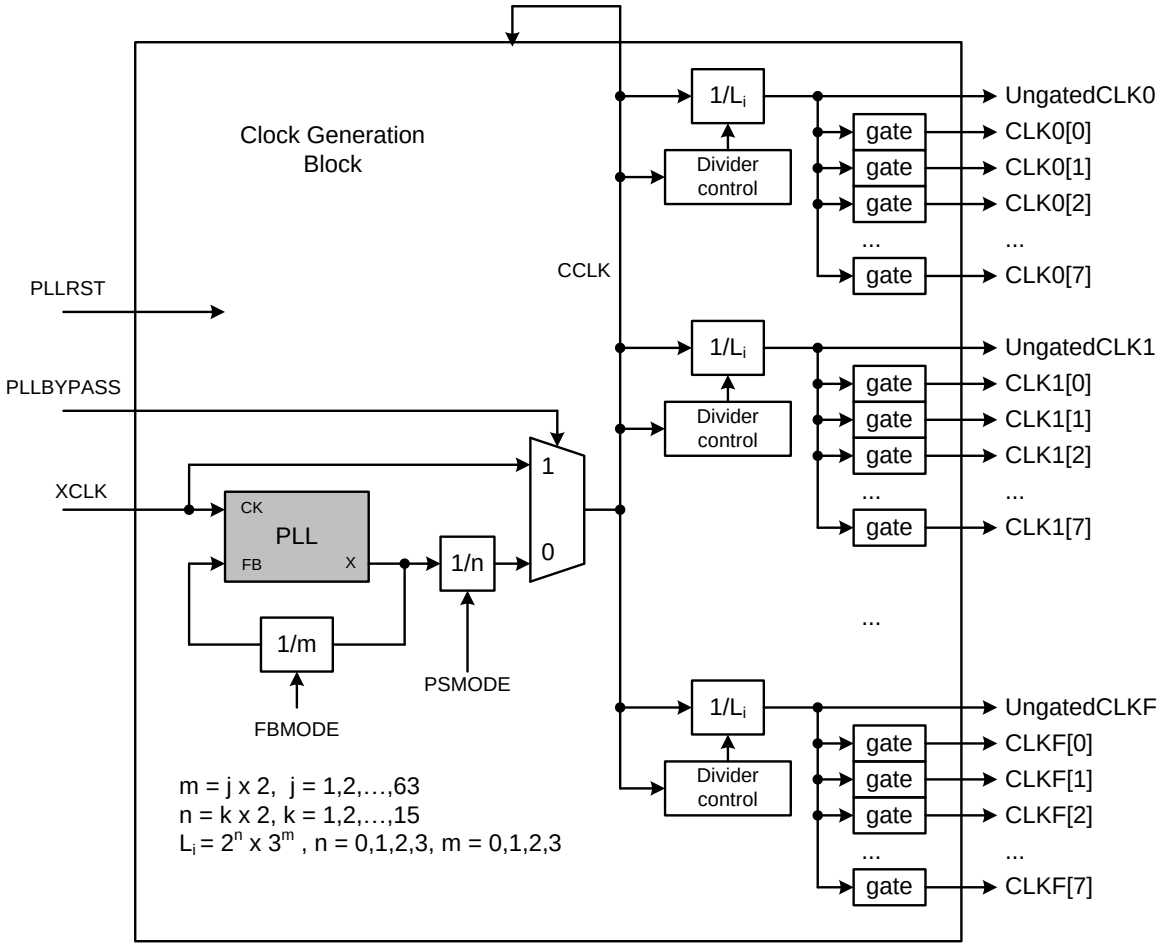


Рисунок 5.10 – Схема генератора синхросигналов

Генератор синхросигналов содержит схему управления ФАПЧ и блоки делителей получаемого со схемы ФАПЧ синхросигнала. Опорный сигнал XCLK попадает на вход схемы ФАПЧ, частота сигнал на выходе схемы определяется по следующей формуле:

$$F_{CCLK} = F_{XCLK} * m/n$$

где значение m задается полем FBMODE регистра CRPLC, а значение n полем PSMODE того же регистра. При проектировании СБИС заданы следующие значения m=16, n=2, таким образом, на выходе схемы ФАПЧ получается сигнал CCLK частотой $8*F_{SYS}$ (макс. 320 МГц) с хорошей скважностью.

Умножение опорного сигнала на схеме ФАПЧ может быть отключено путем подачи 1 на вход СБИС PLLBYPASS. В этом случае частота синхросигнала CCLK будет идентична частоте входного синхросигнала XCLK.

В СБИС СНП-ВП используется 8 доменов синхросигналов – домены с номерами 0-7.

Частота синхросигнала каждого домена может быть изменена и получена делением частоты синхросигнала CCLK на делителе. Значение делителя (L_i) может быть задано путем записи поля DIVMODEx регистра CRDMx соответствующего домена.

Каждый домен синхросигнала может иметь до 9 синхросигналов:

- одного синхросигнала UngatedCLKx, который может быть выключен только при полном отключении синхросигналов в микросхеме (STOP режиме)
- восьми синхросигналов (CLKx[y]), каждый из которых может быть выключен независимо от других, путем записи полей регистра CRLPx соответствующего домена

Реально в каждом домене используется только часть синхросигналов. В процессе проектирования СБИС были заданы частоты всех используемых доменов синхросигналов. Список используемых синхросигналов (выходов CRGS), а также их частоты по-умолчанию, приведен в таблице ниже (Таблица 5.19).

Таблица 5.19 – Используемые в СБИС СНП-ВП синхросигналы

Домен	Частота домена/ соотношение с опорной частотой по умолч.	Макс. частота, МГц	Вывод CRGS	Подключен-ные блоки СБИС	Комментарий
CLK0	F_{NMC}/ 8*F_{SYS}	320	UngatedCLK0	NMCfree	невыключаемый синхросигнал NMU
			CLK0[0]	NMC0	блок NMC0
			CLK0[1]	NMC1	блок NMC1
			CLK0[2]	COR+SMU+ DDC	блоки COR, SMU и DDC
			CLK0[3]	FILTR	блок фильтров
CLK1	F_{ARM}/ 4*$_{FSYS}$	160/ 320	UngatedCLK1	ARM11 macro	ARM11+ETM11+ETB11
CLK2	F_{L2C}/ 4*F_{SYS}	160	UngatedCLK2	L2C	Кэш-память второго уровня L2C
CLK3	F_{HPMS}/ 4*F_{SYS}	160	UngatedCLK3	HPMS+AMBx	Коммутатор HPMS, банки памяти AMB0 и AMB1
			CLK3[0]	BROM	Банк памяти ПЗУ
			CLK3[1]	-	Не используется
			CLK3[2]	-	Не используется
			CLK3[3]	XDMAC	Контроллер ПДП (XDMAC)
			CLK3[4]	HPMS_GPV	Блок SGPV коммутатора HPMS
CLK4	F_{HPMP}/ 2*F_{SYS}	80	UngatedCLK4	HPMP, ARMSC, PMGR, GRETH, CAN, CRGx, EMIC	Коммутатор HPMP и невыключаемые периферийные блоки на APB шине
			CLK4[0]	HPMP_GPV	Блок PGPV коммутатора HPMP
			CLK4[1]	VIC0, VIC1	Контроллеры прерываний VIC0 и VIC1
			CLK4[2]	USB	Контроллер шины USB2.0
			CLK4[3]	RTC, SPI, UARTx, GPIO, EIRC, I2C, DIT, WDOG	Выключаемые периферийные блоки на APB шине
CLK5	F_{ETB}/ 4*F_{SYS}	160	UngatedCLK5	ATCLK	Подблок отладки в ARM
CLK6	F_{DIT}/ 2*F_{SYS}	80	CLK6[0]	DITcntr	Частота счета счетчиков таймеров DIT
CLK7	F_{WDOG}/ 2*F_{SYS}	80	CLK7[0]	WDOGcntr	Частота счета счетчика сторожевого таймера WDOG

5.4.3.1 Управление схемой ФАПЧ

Сигнал сброса PLLRST.

Схема ФАПЧ сбрасывается в следующих случаях:

- активен сигнал внешнего сброса XRESET
- программный сброс (SOFTRST) и бит маски SWRSTM равен 1
- сброс сторожевого таймера (WDOGRST) и бит маски WDRSTM равен 1

- меняется частота генерации схемы ФАПЧ (значение FBMODE)
- микросхема переходит в STOP режим

Длительность активного состояния сигнала сброса PLLRST задана на этапе проектирования СБИС и не может быть изменена, т.к. жестко связана с параметрами аналоговой схемы ФАПЧ (2.4 мкс = 96 тактов синхросигнала XCLK).

Время стабилизации схемы ФАПЧ (PLL Lockup Time).

Сразу после деактивации сигнала PLLRST схема ФАПЧ входит в режим ожидания стабилизации выходного синхросигнала. Время стабилизации тоже является параметром схемы ФАПЧ и составляет 100 мкс, этот параметр задается полем LUWMODE регистра CRPLC (по-умолчанию 6144 такта синхросигнала XCLK). Установка этого параметра меньше чем 100 мкс, приведет к неправильному функционированию СБИС.

По истечении заданного времени взводится бит PLLRDY регистра CRRDY, а также генерируется прерывание PLLRDYINT, если оно разрешено (бит RDYINTM регистра CRIMA равен 0).

Установка частоты генерации схемы ФАПЧ.

Частота синхросигнала CCLK определяется значением полей FBMODE, PSMODE и PLLBYPASS регистра CRPLC. Значения полей FBMODE и PSMODE заданы на этапе проектирования СБИС, а также могут быть переопределены программно. Значение бита PLLBYPASS устанавливается во время активной фазы сигнала сброса CRSTn, в соответствии со значением на внешнем одноименном входе СБИС. Значение бита PLLBYPASS также может быть изменено программно. Ниже в таблице приведены режимы работы схемы ФАПЧ в зависимости от этих параметров.

Таблица 5.20 – Режимы работы схемы ФАПЧ

PLLBYPASS	FBMODE	Режим работы схемы ФАПЧ
1	любое	CRG входит в режим обхода схемы ФАПЧ (PLL bypass)
0	6'b000000	Такое сочетание значений ЗАПРЕЩЕНО!
0	не 6'b000000	Режим генерации. По истечении времени стабилизации схемы ФАПЧ, генерируется стабильный синхросигнал CCLK, в соответствии со значением полей FBMODE и PSMODE

Программное изменение бита PLLBYPASS приведет к немедленному изменению режима работы схемы ФАПЧ.

Смена частоты генерации схемы ФАПЧ происходит сразу после изменения значения поля FBMODE. Во время смены частоты генерации все внутренние синхросигналы CLKx[y] останавливаются, и генерируется сигнал PLLRST. После деактивации сигнала PLLRST, CRG переходит в режим ожидания стабилизации схемы ФАПЧ, пока не истечет время, заданное полем LUWMODE регистра CRPLC. По истечении этого времени устанавливается бит PLLRDY и восстанавливаются внутренние синхросигналы CLKx[y].

ВНИМАНИЕ! Запрещено выставлять параметры схемы ФАПЧ таким образом, чтобы частота генерируемого синхросигнала CCLK превысила 320 МГц – это приведет к неправильному функционированию СБИС.

5.4.3.2 Изменение частоты синхросигналов доменов

Изменить частоту синхросигнала любого домена можно изменяя параметры делителя этого домена. На рисунке ниже приведена последовательность действий для смены частоты синхросигнала домена x (см. Рисунок 5.11).

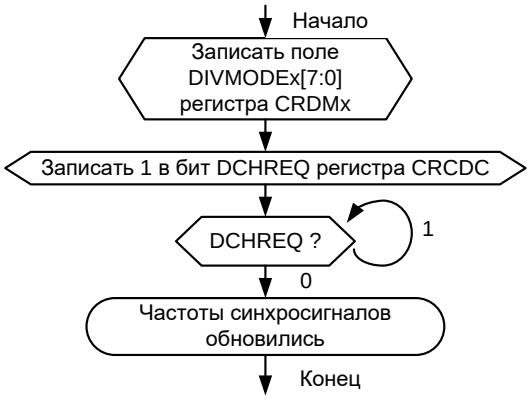


Рисунок 5.11 – Действия для изменения частоты синхросигнала домена

Частота синхросигналов домена определяется по следующей формуле:

$$Fclkx = Fcclk / (2^n \times 3^m)$$

где x – номер домена синхросигнала, значения m и n задаются полем DIVMODEx регистра CRDMx. Можно одновременно изменить настройки делителей нескольких доменов.

После записи поля DIVMODEx необходимо установить бит DCHREQ регистра CRDC, чтобы CRG начал процесс смены частоты синхросигналов. Частота изменяется не мгновенно, а в момент, когда совпадут положительные фронты синхросигналов всех доменов. По окончанию смены частот бит DCHREQ автоматически сбрасывается.

ВНИМАНИЕ! Запрещено выставлять параметры делителей таким образом, чтобы частота синхросигнала на выходе делителя превысила максимальное значение частоты для этого домена (Таблица 5.19).

ВНИМАНИЕ! Частоты синхросигналов разных доменов связаны между собой! Для правильного функционирования СБИС необходимо задавать частоты доменов руководствуясь рисунком ниже (см. Рисунок 5.12).

На рисунке (см Рисунок 5.12) также указаны домены синхросигнала относящиеся к интерфейсу с внешней памятью LPDDR2. Эти синхросигналы генерируются другим блоком генерации синхросигналов – CRGM. Для CRGM используются входы опорного сигнала MCLKXI и MCLKXO, интерфейс AXI между системным коммутатором HPMS и контроллером LPDDR2 внешней памяти EMI имеет асинхронную развязку. Более подробно CRGM описан в главе 5.5.

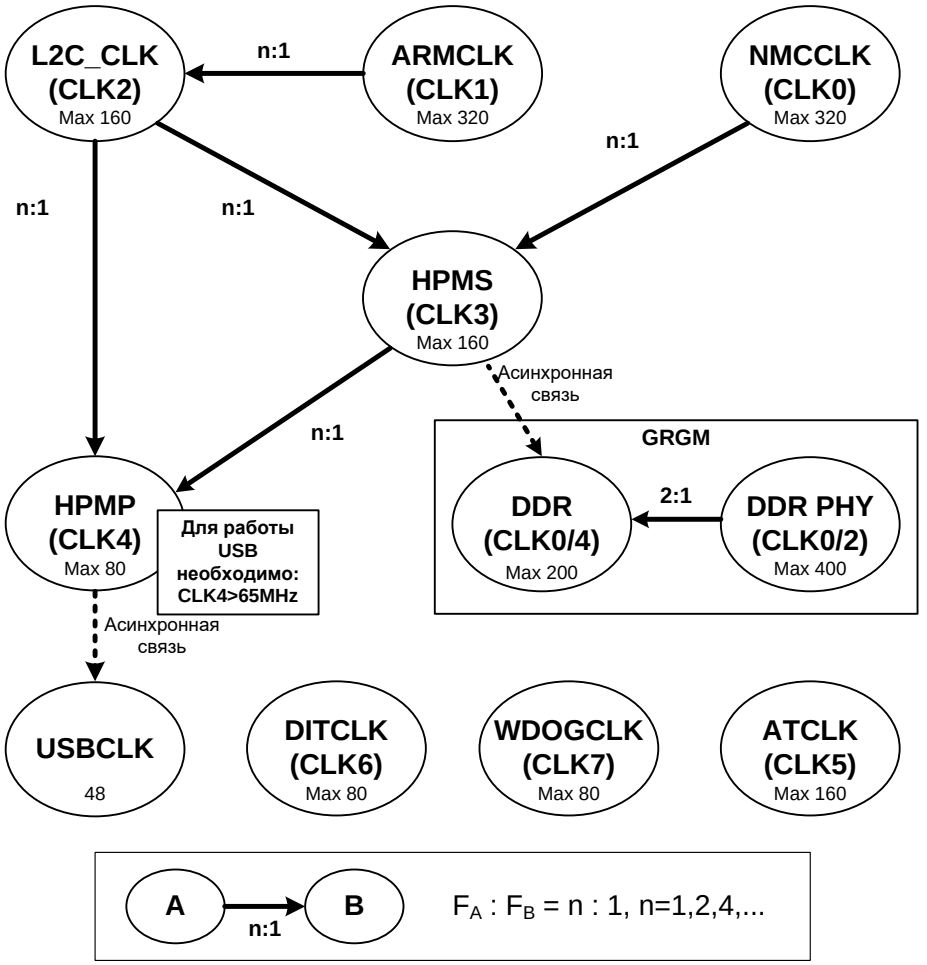


Рисунок 5.12 - Соотношение частот синхросигналов доменов

5.4.3.3 Частичное отключение синхросигналов

СБИС СНП-ВП позволяет отключать выдачу синхросигнала независимо для разных выходных портов CRGS (кроме портов UngatedCLKx – синхросигнал на этих выходах можно выключить только в режиме полного выключения). CRGS производит отключение/включение синхросигналов в соответствии со спецификацией интерфейса управления пониженным энергопотреблением (Low Power Interface) шины AMBA AXI. Основным свойством этого интерфейса является возможность предупредить периферийное устройство об отключении синхросигнала. Периферийное устройство, в свою очередь может разрешить эту операцию, а может и отклонить. Для периферийных устройств, не поддерживающих интерфейс управления пониженным энергопотреблением, считается, что они всегда готовы к отключению синхросигнала.

Для начала процесса выключения синхросигнала CLKx[y], необходимо записать 0 в бит CSYSREQ_Rx[y]. Если периферийное устройство разрешает выключение синхросигнала, то CRG выключает синхросигнал CLKx[y], если периферийное устройство не разрешает выключение синхросигнала, то CRG отсылает запрос на выключение еще раз. Блок CRG шлет запросы до тех пор, пока периферийной устройство не примет запрос, или пока программист не изменит бит CSYSREQ_Sx[y] на 1. Периферийное устройство может не давать разрешение на выключение синхросигнала по причине своей занятости, т.е. для гарантированного выключения синхросигнала периферийного устройства необходимо убедиться, что устройство не занято работой.

Для запуска ранее остановленного синхросигнала CLKx[y] необходимо установить бит CSYSREQ_Rx[y] в 1.

ВНИМАНИЕ! При выключении синхросигналов периферийных блоков необходимо учитывать их взаимодействие с другими блоками, которые не будут выключены. Попытка обращения к блоку с выключенным синхросигналом приведет к зависанию СБИС. Особенно аккуратно надо отключать синхросигналы промежуточных блоков. Например, если выключить синхросигнал коммутатора HPMS, то любой запрос от ядра ARM по шинам I и RW приведет к зависанию СБИС.

5.4.3.4 STOP режим

В этом режиме отключаются все внутренние синхросигналы микросхемы. На рисунке ниже (см. Рисунок 5.13) показаны структурная схема подключения блоков, участвующих в процессе входа/выхода из STOP режима.

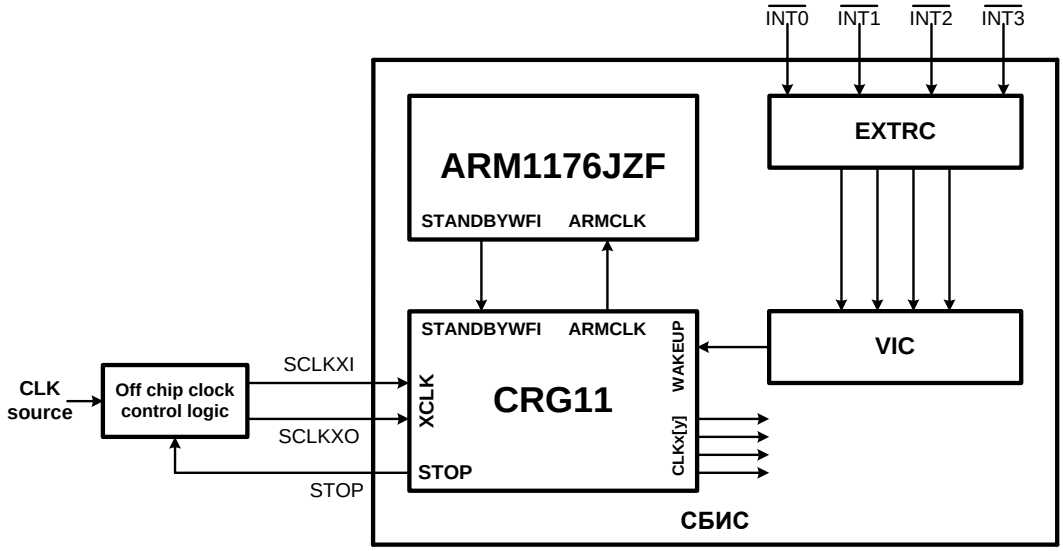


Рисунок 5.13 – Схема соединения блоков, участвующих в переходе в STOP режим

Вход в STOP режим осуществляется следующим образом:

- Программист выполняет операцию Wait For Interrupt в сопроцессоре CP15 процессорного ядра ARM. Как только все текущие операции будут закончены, ядро переходит в Standby режим и сигнализирует об этом выставлением сигнала STANDBYWFI.
- Если разрешен вход в STOP режим (бит STOPEN регистра CRSTP равен 1), CRG11 пытается остановить все CLKx[y] сигналы (используется взаимодействие по интерфейсу управления пониженным энергопотреблением).
- Когда все CLKx[y] остановлены CRGS останавливает все UngatedCLKx синхросигналы.
- Далее выставляется сигнал PLLRST и прекращается генерация синхросигнала на схеме ФАПЧ.
- Микросхема выдает 1 на внешний вывод SPLLSTOP. По этому сигналу можно отключить подачу сигналов SCLKXI и SCLKXO на СБИС СНП-ВП, хотя это и необязательно.
- Возврат к рабочему режиму происходит после генерации внешнего сигнала прерывания - подключенному через VIC к порту WAKEUP блока CRGS. Это прерывание не должно быть маскировано регистрами VIC блока. В случае внешнего прерывания в контроллере внешних прерываний EIRC должен быть выставлен высокий выходной уровень сигнала прерывания. Далее микросхема снимает внешний

сигнал SPLLSTOP, после чего должна быть восстановлена подача сигналов SCLKXI и SCLKXO на СБИС, если они были выключены.

5.4.4 Регистровая модель блока CRGS

Программно доступные регистры блока управления сигналами сброса и синхросигналами CRGS расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение CRGS Base = 0xFFF00000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.21 – Спецификация регистровблока CRGS

Адрес	Разряд-ность	Имя	Сигнал сброса	Описание
Base + 000h	32	CRPLC	CRSTn	Регистр управления PLL
Base + 004h	5	CRRDY	CRSTn	Регистр состояния PLL
Base + 008h	2	CRSTP	CRSTn	Регистр управления STOP режимом
Base + 00Ch	-	-		Зарезервировано
Base + 010h	1	CRIMA	CRSTn	Регистр управления маской прерывания
Base + 014h	1	CRPIC	CRSTn	Регистр снятия прерываний
Base + 018h	-	-		Зарезервировано
Base + 01Ch	-	-		Зарезервировано
Base + 020h	32	CRRSC	CRSTn	Регистр управления сигналами сброса
Base + 024h	1	CRSWR	CRSTn	Регистр управления программным сбросом
Base + 028h	-	-		Зарезервировано
Base + 02Ch	4	CRRSM	PORESETn	Регистр мониторинга сигналов сброса
Base + 030h	1	CRCDC	CRSTn	Регистр управления делением синхросигнала
Base + 034h to 0FCh	-	-		Зарезервировано
Base + 100h	8	CRDM0	CRSTn	Регистр режима деления синхросигнала CLK0
Base + 104h	32	CRLP0	CRSTn	Регистр состояния режима LP домена CLK0
Base + 108h, 10Ch	-	-		Зарезервировано
Base + 110h	8	CRDM1	CRSTn	Регистр режима деления синхросигнала CLK1
Base + 114h	32	CRLP1	CRSTn	Регистр состояния режима LP домена CLK1
Base + 118h, 11Ch	-	-		Зарезервировано
Base + 120h	8	CRDM2	CRSTn	Регистр режима деления синхросигнала CLK2
Base + 124h	32	CRLP2	CRSTn	Регистр состояния режима LP домена CLK2
Base + 128h, 13Ch	-	-		Зарезервировано
Base + 140h	8	CRDM4	CRSTn	Регистр режима деления синхросигнала CLK4
Base + 144h	32	CRLP4	CRSTn	Регистр состояния режима LP домена CLK4
Base + 148h, 15Ch	-	-		Зарезервировано
Base + 160h	8	CRDM6	CRSTn	Регистр режима деления синхросигнала CLK6
Base + 164h	32	CRLP6	CRSTn	Регистр состояния режима LP домена CLK6
Base + 168h, 16Ch	-	-		Зарезервировано
Base + 170h	8	CRDM7	CRSTn	Регистр режима деления синхросигнала CLK7
Base + 174h	32	CRLP7	CRSTn	Регистр состояния режима LP домена CLK7
Base + 178h, 17Ch	-	-		Зарезервировано
Base + 180h	8	CRDM8	CRSTn	Регистр режима деления синхросигнала CLK8
Base + 184h	32	CRLP8	CRSTn	Регистр состояния режима LP домена CLK8
Base + 188h, 18Ch	-	-		Зарезервировано
Base + 190h	8	CRDM9	CRSTn	Регистр режима деления синхросигнала CLK9
Base + 194h	32	CRLP9	CRSTn	Регистр состояния режима LP домена CLK9
Base + 198h to 1FCh	-	-		Зарезервировано

Далее дано описание полей всех регистров.

5.4.4.1 Регистр управления PLL. CRPLC

Таблица 5.22 – Формат регистра CRPLC

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 25]	Зарезервировано	ЧТ	7'h00	-
[24]	PLLBYPASS	ЧТ/ЗП	Значение на внешнем входе СБИС (PLLBYPASS)	Используется, чтобы устанавливать режим работы схемы ФАПЧ 0 – режим генерации схемы ФАПЧ 1 – синхросигнал обходит схему ФАПЧ (PLL bypass)
[23:20]	Зарезервировано	ЧТ	4'h0	-
[19:16]	LUWMODE[3:0]	ЧТ/ЗП	4'b1011	Используются для установления времени ожидания стабилизации схемы ФАПЧ (PLL Lockup Time). Ниже приведены различные значения поля и соответствующие им периоды времени ожидания установа. LUWMODE[3:0] TLUW 4'b0000 64X TXCLK 4'b0001 128 X TXCLK 4'b0010 512 X TXCLK 4'b0011 768 X TXCLK 4'b0100 1024 X TXCLK 4'b0101 1536 X TXCLK 4'b0110 2048 X TXCLK 4'b0111 3072 X TXCLK 4'b1000 4096 X TXCLK 4'b1001 6144 X TXCLK 4'b1010 8192 X TXCLK 4'b1011 12288 X TXCLK 4'b1100 16384 X TXCLK 4'b1101 24576 X TXCLK 4'b1110 32768 X TXCLK 4'b1111 49152 X TXCLK где TLUW – количество тактов синхросигнала XCLK ожидания стабилизации ФАПЧ
[15:12]	Зарезервировано	ЧТ	4'h0	-
[11:8]	PSMODE[3:0]	ЧТ/ЗП	4'b0001	Поля PSMODE[3:0] и FBMODE[5:0] управляют частотой синхросигнала, генерируемого схемой ФАПЧ. См. Описание поля FBMODE[5:0]
[7:6]	Зарезервировано	ЧТ	2'b00	-
[5:0]	FBMODE[5:0]	ЧТ/ЗП	6'b000000, если вход СБИС PLLBYPASS=1 6'b000100, если вход СБИС PLLBYPASS=0	Поля PSMODE[3:0] и FBMODE[5:0] управляют частотой синхросигнала, генерируемого схемой ФАПЧ. FCCLK = FXCLK * m * n FBMODE [5:0] m 6'b000000 PLL не работает 6'b 000001 m = 1 X 2 6'b 000010 m = 2 X 2 6'b 000011 m = 3 X 2 6'b 000100 m = 4 X 2 6'b 000101 m = 5 X 2 6'b 000110 m = 6 X 2 6'b 000111 m = 7 X 2 ..... 6'b 111001 m = 57 X 2

				6’b 111010 m = 58 X 2 6’b 111011 m = 59 X 2 6’b 111100 m = 60 X 2 6’b 111101 m = 61 X 2 6’b 111110 m = 62 X 2 6’b 111111 m = 63 X 2 PSMODE[3:0] n 4’b0000 n = 1 4’b 0001 n = 1 / (1 X 2) 4’b 0010 n = 1 / (2 X 2) 4’b 0011 n = 1 / (3 X 2) 4’b 0100 n = 1 / (4 X 2) 4’b 0101 n = 1 / (5 X 2) 4’b 0110 n = 1 / (6 X 2) 4’b 0111 n = 1 / (7 X 2) 4’b 1000 n = 1 / (8 X 2) 4’b 1001 n = 1 / (9 X 2) 4’b 1010 n = 1 / (10 X 2) 4’b 1011 n = 1 / (11 X 2) 4’b 1100 n = 1 / (12 X 2) 4’b 1101 n = 1 / (13 X 2) 4’b 1110 n = 1 / (14 X 2) 4’b 1111 n = 1 / (15 X 2) Отметим: 1) Скважность синхросигнала CCLK не 50/50, когда PSMODE = 4’b0000 2) Бит PLLBYPASS должен быть установлен прежде, чем биты PSMODE станут равными 4’b0000, или прежде чем сменят значение с 4’b0000 3) С изменением значений битов FBMODE, происходит сброс схемы ФАПЧ. Блок CRGS переходит в режим ожидания стабилизации схемы ФАПЧ. 4) Бит PLLBYPASS не должен меняться пока биты FBMODE[5:0] установлены в 6’b000000

5.4.4.2 Регистр состояния PLL. CRRDY

Таблица 5.23 – Формат регистра CRRDY

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 5]	Зарезервировано	ЧТ	27’h0000000	-
[4]	PSRMNT	ЧТ	1’b0	Бит PSRMNT указывает, что установленные биты PSMODE (регистра CRPLC) отражены в частоте генерируемого схемой ФАПЧ сигнала. 0 – поле PSMODE не отражено в частоте генерируемого сигнала. 1 – поле PSMODE отражено в частоте генерируемого PLL сигнала.
[3:1]	Зарезервировано	ЧТ	3’b000	-
[0]	PLLRDY	ЧТ	1’b0	Бит PLLRDY устанавливается по истечении времени стабилизации ФАПЧ, которое задается полем LUWMODE[3:0] (регистра CRPLC). 0 – PLL не готова 1 – PLL готова Отметим:

				Этот бит не показывает, что PLL защелкнулась, а лишь только говорит об истечении времени, запрограммированного в регистре CRPLC.

5.4.4.3 Регистр управления STOP режимом. CRSTP

Таблица 5.24 –Формат регистра CRSTP

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 2]	Зарезервировано	ЧТ	30'h00000000	-
[1]	STOPMNT	ЧТ/ЗП	1'b0	Бит STOPMNT используется для определения режима работы CRGS. Когда CRG находится в режиме останова (STOP режиме), то бит установлен. 0 – CRG не находится в режиме останова 1 – CRG находится в режиме останова
[0]	STOPEN	ЧТ/ЗП	1'b0	Бит разрешения входа в STOP режим. 0 – вход в STOP режим запрещен 1 – разрешен вход в режим останова при установлении сигнала STANDBYWFI от ядра ARM

5.4.4.4 Регистр управления маской прерывания. CRIMA

Таблица 5.25 – Формат регистра CRIMA

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 1]	Зарезервировано	ЧТ	31'h00000000	-
[0]	RDYINTM	ЧТ/ЗП	1'b1	Бит маскирует прерывание PLLRDYINT, которое устанавливается после истечения времени установа PLL. 0 – нет маскирования прерывания PLLRDYINT 1 – прерывание PLLRDYINT маскируется

5.4.4.5 Регистр снятия прерываний. CRPIC

Таблица 5.26 – Формат регистра CRPIC

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 1]	Зарезервировано	ЧТ	31'h00000000	-
[0]	PLLRDYINT	ЧТ	1'b0	Бит показывает состояние прерывания PLLRDYINT, которое устанавливается по истечении времени стабилизации схемы ФАПЧ. 0 – PLLRDYINT не выставлено 1 – PLLRDYINT выставлено

5.4.4.6 Регистр управления сигналами сброса. CRRSC

Таблица 5.27 – Формат регистра CRRSC

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 20]	Зарезервировано	ЧТ	12'h000	-
[19:16]	SRSTMODE[3:0]	ЧТ/ЗП	4'b0111	Бит используется для установления длительности активного состояния XSRSTout в тактах синхросигнала CCLK. Определяется параметрами внешнего отладчика.

				4’b0000 8 X TCCLK 4’b0001 12 X TCCLK 4’b0010 16 X TCCLK 4’b0011 24 X TCCLK 4’b0100 32 X TCCLK 4’b0101 48 X TCCLK 4’b0110 64 X TCCLK 4’b0111 96 X TCCLK ← начальное 4’b1000 128 X TCCLK значение 4’b1001 192 X TCCLK 4’b1010 256 X TCCLK 4’b1011 384 X TCCLK 4’b1100 512 X TCCLK 4’b1101 768 X TCCLK 4’b1110 1024 X TCCLK 4’b1111 1536 X TCCLK
[15:10]	Зарезервировано	ЧТ	6’h00	-
[9]	WDRSTM	ЧТ/ЗП	1’b0	Управляет реакцией CRG на сброс от сторожевого таймера (WDOGRST). 0 – CRSTn и PLLRST НЕ генерируются 1 – CRSTn и PLLRST генерируются
[8]	SWRSTM	ЧТ/ЗП	1’b0	Управляет реакцией CRG на внутренний программный сброс (SOFTRST). 0 – CRSTn и PLLRST НЕ генерируются 1 – CRSTn и PLLRST генерируются
[7:4]	Зарезервировано	ЧТ	4’h0	-
[3:0]	ARSTMODE[3:0]	ЧТ/ЗП	4’b0111	Бит используется для установления длительностью активного состояния ARESETn в тактах синхросигнала CCLK. 4’b0000 8 x TCCLK (Min.) 4’b0001 12 x TCCLK (Min.) 4’b0010 16 x TCCLK (Min.) 4’b0011 24 x TCCLK (Min.) 4’b0100 32 x TCCLK (Min.) 4’b0101 48 x TCCLK (Min.) 4’b0110 64 x TCCLK (Min.) 4’b0111 96 x TCCLK (Min.) ← н.знач. 4’b1000 128 x TCCLK (Min.) 4’b1001 192 x TCCLK (Min.) 4’b1010 256 x TCCLK (Min.) 4’b1011 384 x TCCLK (Min.) 4’b1100 512 x TCCLK (Min.) 4’b1101 768 x TCCLK (Min.) 4’b1110 1024 x TCCLK (Min.) 4’b1111 1536 x TCCLK (Min.) В таблице выше представлена минимально возможная длительность активного состояния ARESETn.

5.4.4.7 Регистр управления программным сбросом. CRSWR

Таблица 5.28 – Формат регистра CRSWR

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 1]	Зарезервировано	ЧТ	31’h00000000	-
[0]	SWRSTREQ	ЧТ/ ЗП1	1’b0	Бит используется для генерации внутреннего программного сброса. Программный сброс имеет место сразу же после установления бита. Генерируемые внутренние сигналы сброса определяются битом SWRSTM (регистра CRRSM)

5.4.4.8 Регистр мониторинга сигналов сброса. CRRSM

Таблица 5.29 – Формат регистра CRRSM

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 4]	Зарезервировано	ЧТ	28'h0000000	-
[3]	PORESET	ЧТ/ ЗП0	1'b1	Бит показывает состояние сигнала сброса PORESETn. Устанавливается при генерации сигнала сброса PORESETn. Необходимо записывать нуль для снятия этого бита. 0 – PORESETn не устанавливался 1 – PORESETn устанавливался Бит инициализируется при установлении сигнала PORESETn, запись 1 игнорируется
[2]	SRST	ЧТ/ ЗП0	1'b0	Бит показывает состояние программного сброса (XSRST) от внешнего отладчика. Устанавливается при установке сигнала XSRSTin. Необходимо записывать нуль для снятия этого бита. 0 – XSRSTin не устанавливался 1 – XSRSTin устанавливался Бит инициализируется при установлении сигнала PORESETn. Запись 1 игнорируется.
[1]	SWRST	ЧТ/ ЗП0	1'b0	Бит отслеживает программный сброс. Устанавливается только тогда, когда имел место программный сброс. Необходимо записывать нуль для снятия этого бита. 0 –программного сброса не было 1 – был программный сброс Бит инициализируется при установлении сигнала PORESETn. Запись 1 игнорируется
[0]	WDRST	ЧТ/ ЗП0	1'b0	Бит отслеживает сброс от сторожевого таймера. Устанавливается при генерации сброса от сторожевого таймера. Необходимо записывать нуль для снятия этого бита. 0 – сброса от сторожевого таймера не было 1 – был сброс от сторожевого таймера Бит инициализируется при установлении сигнала PORESETn. Запись 1 игнорируется.

5.4.4.9 Регистр управления делением синхросигнала. CRCDC

В таблице ниже приведен формат регистра CRCDC.

Таблица 5.30 – Формат регистра CRCDC

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 1]	Зарезервировано	ЧТ	31'h00000000	-
[0]	DCHREQ	ЧТ/ЗП	1'b0	Бит обновляет режим деления синхросигнала. Установка этого бита после установки регистров CRDMx приводит к изменению частоты генерируемых синхросигналов CLK0-CLK9. Бит автоматически сбрасывается после изменения частот генерируемых синхросигналов. 0 – частоты генерируемых синхросигналов не обновлены (Начальное значение) 1 – обновление частот в процессе.

5.4.4.10 Регистр режима деления синхросигнала.CRDMx

В таблице ниже приведен формат регистра ("x" принимает значения от 0,1,2,4,6,7,8 и 9).

Таблица 5.31 – Формат регистра CRDMx

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 8]	Зарезервировано	ЧТ	24'h0000000	-
[7: 0]	DIVMODEx [7:0]	ЧТ /ЗП	в соответствии с таблицей (Таблица 5.19)	Биты используются для установления режима деления синхросигнала для CLKx домена. Частота генерируемого синхросигнала может быть определена по следующим формулам: $Fclkx = Fcclk / (2^n \times 3^m)$ (где n=0~3 и m=0~3) Fclkx – частота синхросигнала CLKx домена Fcclk – частота синхросигнала CCLK Параметры M и N могут быть определены согласно битам DIVMODEx: DIVMODEx m n Частота CLKx 8'h00 0 0 fCLKx = fCCLK x (1/1) 8'h01 0 1 fCLKx = fCCLK x (1/2) 8'h03 0 2 fCLKx = fCCLK x (1/4) 8'h07 0 3 fCLKx = fCCLK x (1/8) 8'h02 1 0 fCLKx = fCCLK x (1/3) 8'h05 1 1 fCLKx = fCCLK x (1/6) 8'h0B 1 2 fCLKx = fCCLK x (1/12) 8'h17 1 3 fCLKx = fCCLK x (1/24) 8'h08 2 0 fCLKx = fCCLK x (1/9) 8'h11 2 1 fCLKx = fCCLK x (1/18) 8'h23 2 2 fCLKx = fCCLK x (1/36) 8'h47 2 3 fCLKx = fCCLK x (1/72) 8'h1A 3 0 fCLKx = fCCLK x (1/27) 8'h35 3 1 fCLKx = fCCLK x (1/54) 8'h6B 3 2 fCLKx = fCCLK x (1/108) 8'hD7 3 3 fCLKx = fCCLK x (1/216)

5.4.4.11 Регистр состояния режима LP домена CLKx. CRLPx

В таблице ниже приведен формат регистра ("x" принимает значения от 0,1,2,4,6,7,8 и 9).

Таблица 5.32 – Формат регистра CRLPx

<table>
<tr><th>Биты</th><th>Название</th><th>Тип</th><th>Значение после инициа-лизации</th><th>Выполняемая функция</th></tr>
<tr><td>[31: 24]</td><td>CENx[7:0]</td><td>ЧТ</td><td>8’hff</td><td>Разрешает генерацию CLKx[y].<table><tr><td>CENx[y]</td><td></td></tr><tr><td>0</td><td>CLKx[y] остановлен</td></tr><tr><td>1</td><td>CLKx[y] не остановлен</td></tr></table></td></tr>
<tr><td>[23: 16]</td><td>CACTIVE_Cx[7:0]</td><td>ЧТ</td><td>8’h00</td><td>Биты отслеживают внутренние сигналы CACTIVE_Cx[7:0]. Биты устанавливаются, когда внешний порт CACTIVEx[y] имеет высокий уровень. Этот сигнал информирует CRG, что периферийное устройство не готово к отключению синхросигнала.</td></tr>
<tr><td>[15: 8]</td><td>LPOWERHSx[7:0]</td><td>ЧТ</td><td>8’h00</td><td>Биты отслеживают внутренние сигналы LPOWERHSx[7:0]. Биты устанавливаются, когда в интерфейсе управления пониженным энергопотреблением между CRGS и периферийным блоком в домене CLKx[y] происходит обмен. Т.е. CRG пытается отключить или включить синхросигнал CLKx[y]. Во время такого обмена невозможна запись бита CSYSREQ_Rx[y].<table><tr><td>LPOW-ERHSx[y]</td><td></td></tr><tr><td>0</td><td>Запись в CSYSREQ_Rx[y] разрешена</td></tr><tr><td>1</td><td>Запись в CSYSREQ_Rx[y] игнорируется</td></tr></table></td></tr>
<tr><td>[7: 0]</td><td>CSYSREQ_Rx[7:0]</td><td>ЧТ/ ЗП</td><td>8’hff</td><td>Биты, управляющие выдачей запроса на включение/выключение синхросигнала CLKx[y] по интерфейсу управления пониженным энергопотреблением.<table><tr><td>CSYSREQ_ Rx[y]</td><td></td></tr><tr><td>0</td><td>Запрос на остановку CLKx[y]</td></tr><tr><td>1</td><td>Запрос на активацию CLKx[y]</td></tr></table></td></tr>
</table>

5.5 Блок управления сигналами сброса и синхросигналами CRGM

Блок управления сигналами сброса и синхросигналами CRGM предназначен для генерации синхросигналов контроллера внешней памяти EMI и его физического интерфейса.

Блок CRGM полностью идентичен блоку CRGS, за исключением особенностей перечисленных ниже.

5.5.1 Устройство блока CRGM

В таблице ниже приведен список внешних выводов, относящихся к блоку управления сигналами сброса и синхросигналами CRGM.

Таблица 5.33 – Выводы микросхемы, относящиеся к CRGM

Вывод	Тип буфера	Примечание
MCLKXI	in	Входы внешнего синхросигнала от кварца или генератора.
MCLKXO	inout	Входы внешнего синхросигнала от кварца
XSRST	inout	Внешний программный сброс микросхемы.
XRESET	in	Вход основного сброса СБИС СНП-ВП
XTRST	in	Вход сброса тестовой логики
PLLBYPASS	in	Вход выбора режима работы ФАПЧ
MPLLSTOP	out	Выход индикации входа СБИС в STOP режим
MPLL_L	out	Тестовый выход схемы ФАПЧ (не используется)

Входы MCLKXI и MCLKXO служат для подключения сигнала опорной частоты для ФАПЧ. Для этого можно использовать кварцевый резонатор, или можно использовать внешний генератор, подключив его к входу MCLKXI. Максимальная частота опорного синхросигнала MCLK F_{MEM} = 40МГц. Опорный сигнал MCLK подключается к входу XCLK блока CRGM.

Блок CRGM содержит схему управления ФАПЧ и генерацией внутренних синхросигналов (Clock Generation Block). Опорный сигнал MCLK попадает на этот блок, умножается на схеме ФАПЧ на 20 до частоты 20*F_{MEM}. Частота синхросигнала, получаемого с ФАПЧ, в два раза выше чем реально нужно для функционирования интерфейса с внешней памятью (F_{PHY} = 400МГц) – дальнейшее деление синхросигнала выполняется на на ненастраиваемых аппаратных делителях, находящихся рядом с интерфейсом (для получения хорошей скважности) (Таблица 5.34).

Входы сброса XRESET, XSRST, XTRST, а также вход PLLBYPASS являются общими для CRGS и CRGM.

Таблица 5.34 – Используемые в CRGM синхросигналы

Домен	Частота домена/ соотношение с опорной частотой по умолч.	Макс. частота, МГц	Вывод CRGM	Подключенные блоки СБИС	Комментарий
CLK0	2*F_{PHY}/ 20*F_{MEM}	800	CLK0[0]	LPDDR2 PHY	Двойная частота памяти LPDDR2

5.6 Системные векторные контроллеры прерываний VIC0 и VIC1

СБИС СНП-ВП содержит 2 контроллера прерываний PL192 фирмы ARM. Каждый контроллер прерываний VIC в СБИС СНП-ВП имеет следующие характеристики:

- предоставляет программный интерфейс для взаимодействия процессорного ядра ARM и системой прерываний
- поддержка 32-х векторных прерываний (взаимодействие с ядром через VIC порт) с настраиваемыми программно приоритетами и маскированием
- поддержка генерации программных прерываний
- подключение к системной шине СБИС через AMBA AHB интерфейс, используются полный 32-битный интерфейс шины данных

5.6.1 Устройство векторных контроллеров прерываний

Блок не имеет сигналов, соединенных с внешними выводами микросхемы.

Т.к. каждый контроллер поддерживает только 32 прерывания от периферийных устройств, то для увеличения общего количества прерываний до 64-х два контроллера соединены в цепочку. На рисунке ниже приведена структурная схема подключения контроллеров к ядру ARM.

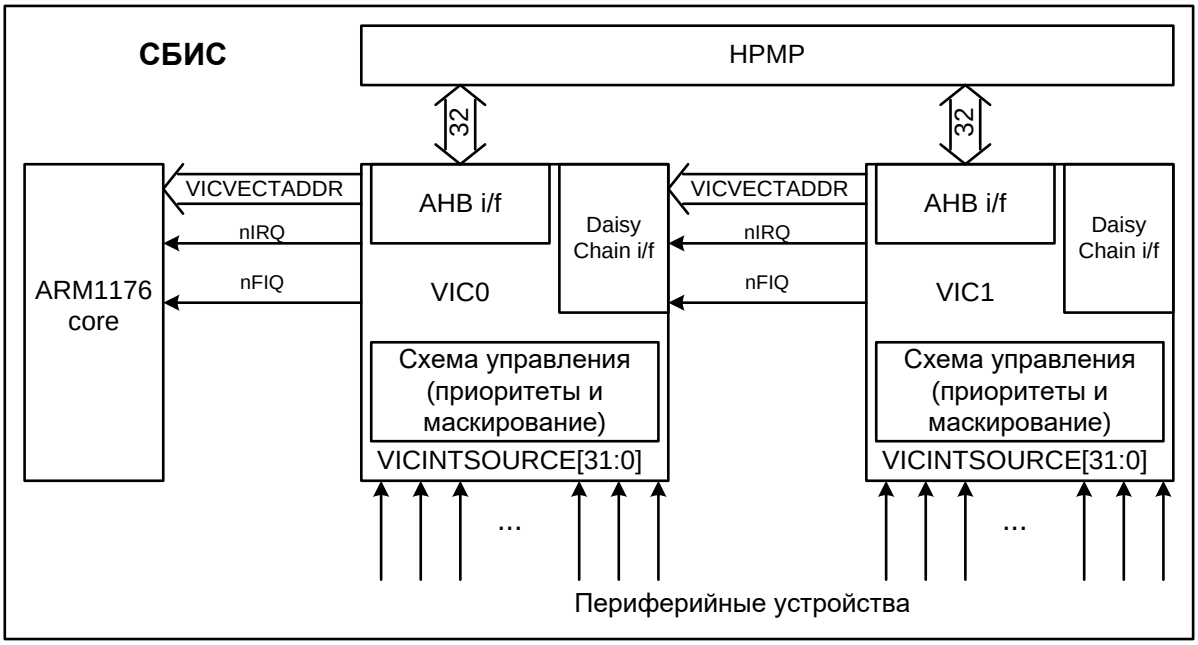


Рисунок 5.14 – Схема подключения контроллеров прерываний к ядру

Каждый векторный контроллер прерываний (VIC, Vectored Interrupt Controller) предоставляет программный интерфейс для взаимодействия процессора с системой прерываний. При возникновении прерывания в системе, обычный порядок действий такой:

- нужно определить источник прерывания
- определить адрес процедуры обработки для этого прерывания (адрес ISR)
- если одновременно возникают несколько прерываний – то выбрать, какое из них обрабатывать первым

Все эти задачи VIC решает на уровне аппаратуры. Вместе с сигналом прерывания процессору передается адрес процедуры обработки (VICVECTADDR) для прерывания с наивысшим

приоритетом. Контроллер VIC может контролировать до 32-х прерываний от периферийных устройств, подключенных к входам VIVINTSOURCE[31:0].

В процессорном ядре ARM различаются два типа прерываний.

- Быстрое прерывание FIQ (Fast Interrupt Request – сигнал nFIQ) для обработки запросов, требующих минимального времени задержки
- Обычное прерывание IRQ (Interrupt Request – сигнал nIRQ), предоставляющее дополнительные возможности при обработке запросов, но с большим, чем в случае FIQ, временем задержки

Обычно используется только один источник быстрых прерываний, что позволяет сократить время задержки до предела. Экономия достигается за счет следующих возможностей.

- Можно не тратить время на определение источника прерывания – он один
- Не надо обеспечивать рекурсивную обработку прерываний. В этом случае, обработчик прерываний может обойтись дополнительными регистрами режима FIQ (CPSR[4:0]=0x11) и исключить операцию сохранения контекста.

ЗАМЕЧАНИЕ.

Более высокий приоритет быстрого прерывания, по сравнению с обычным, обеспечивается процессором ARM. VIC может передавать процессору сигналы быстрого и обычного прерываний одновременно.

Каждому из 32-х возможных на входе VIC прерываний можно присвоить программный приоритет, записав нужное значение в соответствующий регистр программного приоритета. Программные приоритеты используются только для обычных прерываний(IRQ) и позволяют менять порядок их обработки динамически, в процессе работы. Если несколько обычных прерываний имеют одинаковый программный приоритет, то для определения порядка их обработки используется аппаратный приоритет. Аппаратный приоритет используется и в том случае, когда программные приоритеты не установлены. Аппаратный приоритет равен номеру прерывания. Таким образом, прерывание FIQ имеет максимальный приоритет, далее идут прерывания 0-31. Прерывание 0 имеет высший аппаратный приоритет, а прерывание 31 – низший.

Для каждого из 32-х возможных на входе VIC прерываний в VIC есть регистр адреса процедуры обработки прерывания (VICVECTADDRx). Эти регистры заполняются программно и используются только для обычных прерываний. При возникновении одного или нескольких обычных прерываний VIC выбирает, на основании программных и аппаратных приоритетов, одно из них и передает процессору сигнал прерывания IRQ и содержимое соответствующего регистра адреса процедуры обработки прерывания (по шине VICVECTADDR). Одновременно адрес процедуры обработки прерывания записывается в регистр VICADDRESS.

Чтение процессором адреса процедуры обработки прерывания из регистра VICADDRESS имеет важный побочный эффект: VIC получает сигнал о том, что процессор начал обработку обычного прерывания. При получении этого сигнала VIC блокирует обработку обычных прерываний с приоритетом равным или меньшим, чем приоритет текущего обрабатываемого прерывания. Блокировка снимается при записи любого значения в регистр VICADDRESS.

ЗАМЕЧАНИЕ.

Блокировка обработки прерываний равного или меньшего чем текущее обрабатываемое прерывание приоритета не влияет на регистры VICIRQSTATUS и VICRAWSTATUS.

Программа обработки прерываний может не использовать адреса процедур обработки прерываний из регистров VICVECTADDRx. В этом случае, в эти регистры можно записать другую информацию, например, номер соответствующего прерывания, что ускорит определение источника прерывания при обработке прерываний.

В таблице ниже приведено соответствие прерываний от периферийных устройств и входов контроллера VIC:

Таблица 5.35 – Прерывания в СБИС СНП-ВП

№	VIC0		№	VIC1	
	Источник	**Описание**		**Источник**	**Описание**
0	SWINT0	Программное прерывание 0	32	XDMACH2	Прерывание от XDMAC канала 2
1	SWINT1	Программное прерывание 1	33	XDMACH3	Прерывание от XDMAC канала 3
2	COMMRX	Канал отладки (DCC) Rx от ARM1176	34	EXTINT2	Внешнее прерывание 2
3	COMMTX	Канал отладки (DCC) Tx от ARM1176	35	EXTINT3	Внешнее прерывание 3
4	TIMINT0	Прерывание таймера 0	36	-	зарезервировано
5	TIMINT1	Прерывание таймера 1	37	-	зарезервировано
6	WDOGINT	Прерывание сторожевого таймера	38	-	зарезервировано
7	UART0INT	Прерывание канала UART0	39	-	зарезервировано
8	UART1INT	Прерывание канала UART1	40	-	зарезервировано
9	NMC0HP	NMC0 высокоприоритетное прерывание	41	-	зарезервировано
10	NMC1HP	NMC1 высокоприоритетное прерывание	42	-	зарезервировано
11	INT_CH_DDC	Прерывание от блока DDC	43	-	зарезервировано
12	INT_MS_ITU	Прерывание от интервального таймера по основному временному интервалу	44	-	зарезервировано
13	SSPINT	Общее прерывание от SSP блока	45	-	зарезервировано
14	I2CINT	Прерывание от I2C контроллера	46	-	зарезервировано
15	CANINT	Прерывание от CAN контроллера	47	-	зарезервировано
16	RTCINT	Прерывание от блока RTC	48	-	зарезервировано
17	EMIINT	Общее прерывание контроллера EMI	49	-	зарезервировано
18	CORINT	Прерывание от блока корреляторов	50	-	зарезервировано
19	XDMACH0	Прерывание от XDMAC канала 0	51	-	зарезервировано
20	XDMACH1	Прерывание от XDMAC канала 1	52	-	зарезервировано
21	NDMADR	DR прерывание от NMU DMA контроллера	53	-	зарезервировано
22	NDMAMC0	MC0 прерывание от NMU DMA контроллера	54	INT_PRG_DDC	прерывание от блока DDC
23	NDMAMC1	MC1 прерывание от NMU DMA контроллера	55	L2CCINT	Общее прерывание от кэш-памяти L2CC
24	INT_AUX_ITU	Прерывание от интервального таймера по временному интервалу	56	DMAIRQ	Прерывание (Non-Secure) каналов ПДП ARM1176
25	INT_STRB_ITU	Прерывание от интервального таймера по началу выдачи программного строба	57	DMASIRQ	Прерывание (Secure) каналов ПДП ARM1176
26	ETHINT	Прерывание от Ethernet контроллера	58	DMAERRIRQ	Ошибка доступа каналов ПДП ARM1176
27	USBINT	Прерывание от USB контроллера	59	nPMUIRQ	От System Metric блока ARM1176
28	NMC0LP	NMC0 низкоприоритетное прерывание	60	SPLLRDYINT	Готовность SPLL от CRGS
29	NMC1LP	NMC1 низкоприоритетное прерывание	61	MPLLRDYINT	Готовность MPLL от CRGM
30	EXTINT0	Внешнее прерывание 0	62	nVALFIQ	Тестовое FIQ прерывание от ARM1176
31	EXTINT1	Внешнее прерывание 1	63	nVALIRQ	Тестовое IRQ прерывание от ARM1176

Прерывания SWINTx – это программные прерывания. Хотя любое прерывание VIC контроллера может быть активировано путем записи соответствующего бита регистра VICSOFTINT, два выделенных программных прерывания позволяют использовать программные прерывания, не маскируя никакие из аппаратных прерываний.

Прерывания COMMRX и COMMTX – это прерывания от канала отладки (Debug Communication Channel) используются при отладке программ через JTAG порт. В нормальном режиме необходимо маскировать эти прерывания.

Прерывания 36-53 – это тестовые прерывания, в нормальном режиме не используются.

Остальные прерывания генерируются периферийными устройствами, описание причин генерации и способ обработки каждого конкретного прерывания приведены в разделе, описывающем соответствующий периферийный блок.

Общая схема обработки любого прерывания следующая:

- после генерации прерывания процессорное ядро переходит по адресу процедуры обработки прерывания, соответствующей активному прерыванию
- в процедуре обработки прерывания необходимо сбросить источник прерывания, путем записи соответствующего регистра в периферийном устройстве. VIC не буферизует сигналы прерываний, поэтому сброс прерывания в периферийном блоке приведет к снятию прерывания nIRQ или nFIQ к ядру. По этой же причине, если прерывание от периферийного блока будет активировано на короткое время, и за это время ядро не успеет перейти по адресу процедуры обработки прерываний – то такое прерывание не будет обработано.

5.6.2 Подробности работы с контроллером VIC

5.6.2.1 Регистр VICRAWINTR

Состояние всех входов VICINTSOURCE[31:0] отображается в регистре VICRAWINTR. Каждый разряд регистра соответствует входу с номером, равным номеру разряда. Значение разряда 0 означает отсутствие прерывания на входе, а 1 – его наличие.

Значение в этом регистре отражает значение прерывания до маскирования.

5.6.2.2 Определение типа прерывания

Тип прерывания определяется значением разряда регистра VICINTSELECT, номер которого равен номеру прерывания. Для FIQ значение разряда 1, для IRQ – 0.

5.6.2.3 Определение приоритета прерывания

В VIC приоритеты связаны только с прерываниями IRQ. Приоритет прерывания определяется на основе программного и аппаратного приоритетов.

Программный приоритет – это значение соответствующего 4-х разрядного регистра приоритета VICVECTPRIORITY*n*, где *n* – номер прерывания. Значение программного приоритета – целое число от 0 до 15. 0 – высший приоритет, 15 – низший. По умолчанию все прерывания имеют одинаковый низший приоритет 15.

Аппаратный приоритет равен номеру прерывания на входе. 0 – высший приоритет, 31 – низший.

При сравнении приоритетов сначала используется программный приоритет, а при равенстве программных приоритетов – аппаратный.

5.6.2.4 Маскирование

Прерывание может быть закрыто программным или аппаратным способом. Сигнал закрытого прерывания, поступивший на вход VIC, не передается на выход.

Прерывание с номером *n* закрыто программным способом, если в *n*-м разряде регистра VICINTENABLE записан ноль и/или если это IRQ прерывание с программным приоритетом *p* и в *p*-м разряде регистра VICPRIORITYMASK записан ноль.

IRQ прерывание с номером n закрыто аппаратным способом, если его приоритет равен или ниже приоритета прерывания, обрабатываемого процессором в данный момент. Для снятия аппаратного маскирования после окончания обработки прерывания необходимо произвести запись любого значения в регистр VICADDRESS.

5.6.2.5 Регистры состояния VICIRQSTATUS и VICFIQSTATUS

Единица в разряде регистра VICIRQSTATUS или VICFIQSTATUS означает, что на входе VIC есть сигнал соответствующего прерывания, и это прерывание не закрыто программным способом.

5.6.2.6 Адрес процедуры обработки прерывания

Адрес процедуры обработки прерывания находится в регистре VICVECTADDR*n*, где *n* – номер прерывания. Для IRQ прерываний значение этого регистра передается процессору вместе с сигналом прерывания и одновременно записывается в регистр VICVECTADDR. Для FIQ прерываний регистры VICVECTADDR*n* не используются.

5.6.2.7 Регистр программных прерываний VICSOFTINT

Регистр программных прерываний VICSOFTINT позволяет процессору эмулировать внешние прерывания программным способом. Запись единицы в *n*-ый разряд регистра VICSOFTINT приводит к тому же результату, что и появление сигнала прерывания на *n*-м входе VIC.

5.6.2.8 Взаимодействие процессора и VIC при обработке обычных прерываний

Обычное прерывание (IRQ) может обрабатываться двумя способами. Выбор способа определяется значением разряда VE (VIC Enable) контрольного регистра (Control Register) системного сопроцессора CP15 процессорного ядра ARM.

Если разряд VE контрольного регистра сопроцессора CP15 равен единице, то при обычном прерывании (IRQ) процессор прерывается, переходит в режим IRQ с заблокированными прерываниями IRQ и выполняет переход по адресу, содержащемуся в регистре VICADDRESS (адрес передается по шине VICVECTADDR). Одновременно VIC получает сигнал, означающий, что процессор начал обработку прерывания.

Если разряд VE контрольного регистра сопроцессора CP15 равен нулю, то при обычном прерывании (IRQ) процессор прерывается, переходит в режим IRQ с заблокированными прерываниями IRQ и выполняет переход по адресу 0x18 относительно начала вектора прерываний. Для того чтобы послать VIC сигнал, означающий, что процессор начал обработку прерывания, нужно выполнить считывание из регистра VICADDRESS.

При получении от процессора сигнала начала обработки прерывания VIC обнуляет передаваемый процессору сигнал IRQ прерывания, блокирует IRQ прерывания, приоритет которых равен или ниже, чем приоритет обрабатываемого процессором прерывания, и продолжает обрабатывать прерывания с более высоким приоритетом.

Закончив обработку прерывания, нужно выполнить запись любого значения в регистр VICADDRESS. В результате этой записи VIC получает сигнал, означающий, что процессор закончил обработку прерывания. Получив этот сигнал, VIC снимает блокировку IRQ прерываний, приоритет которых равен или ниже, чем приоритет обрабатываемого процессором прерывания.

5.6.3 Последовательность шагов при обработке прерываний

5.6.3.1 IRQ прерывание. Бит VE=1

При возникновении IRQ прерывания процессор прерывается и переходит по адресу из регистра VICADDRESS*n, n* – номер прерывания. Далее надо выполнить следующие действия.

Сохранить контекст.

Разрешить IRQ прерывания процессора, чтобы прерывания более высокого приоритета могли быть обслужены.

Выполнить содержательную часть обработки прерывания.

Обнулить устройство, вызвавшее прерывание, или обнулить разряд регистра VICSOFTINT, записав единицу в соответствующий разряд регистра VICSOFTINTCLEAR, если прерывание вызвано программным способом.

Запретить IRQ прерывания процессора.

Записать любое значение в регистр VICADDRESS, чтобы сообщить VIC об окончании обработки прерывания. Эта процедура обязательна для снятия аппаратного маскирования прерываний.

Восстановить контекст.

Выполнить возврат из процедуры обработки прерывания. При этом будет восстановлено состояние процессора, таким, как оно было в момент прерывания. В частности, будут разрешены IRQ прерывания.

5.6.3.2 IRQ прерывание. Бит VE=0. Прерывания векторные

Если бит VE контрольного регистра сопроцессора CP15 равен нулю, то при возникновении прерывания адрес перехода не передается процессорному ядру автоматически. При возникновении IRQ прерывания процессор прерывается и переходит по адресу 0x18 относительно начала таблицы векторов прерываний (т.е. 0x00000018 или 0xFFFF0018 (в High Vectors Mode)). Далее надо выполнить следующие действия.

Считать регистр VICADDRESS и перейти по адресу, содержащемуся в этом регистре. Это действие можно выполнить одной командой загрузки значения регистра VICADDRESS в регистр PC. При этом нет необходимости в дополнительном переходе. Команда загрузки из регистра VICADDRESS может находиться прямо в векторе прерываний.

Далее как в п.5.6.3.1, начиная с пункта 1.

5.6.3.3 IRQ прерывание. Бит VE=0. Прерывания не векторные

В этом способе, в отличие от предыдущего, источник прерывания определяется программно (чтение VICIRQSTATUS), а не аппаратно (по значению в VICADDRESS). При возникновении IRQ прерывания процессор прерывается и переходит по адресу 0x18 относительно начала таблицы векторов прерываний. Далее надо выполнить следующие действия.

Перейти на общую процедуру обработки IRQ прерываний.

Сохранить контекст.

Разрешить IRQ прерывания процессора, чтобы прерывания более высокого приоритета могли быть обслужены.

Выполнить холостое считывание регистра VICADDRESS для передачи VIC сигнала о начале обработки прерывания. Это чтение включает аппаратное маскирование прерываний.

Считать регистр VICIRQSTATUS и определить номер обрабатываемого прерывания.

Определить адрес процедуры обработки выбранного прерывания и перейти по этому адресу.

Далее как в разделе 5.6.3.1, начиная с пункта 3.

5.6.3.4 FIQ прерывание

Как уже говорилось выше, обычно в системе только одно быстрое FIQ прерывание, поэтому для его обработки используется упрощенная процедура обработки. При возникновении FIQ прерывания, процессор прерывается и переходит по адресу 0x1C относительно начала таблицы векторов прерываний. Далее надо выполнить следующие действия.

Перейти на процедуру обработки FIQ прерываний. Команда перехода может находиться прямо в адрес-векторе прерывания FIQ.

Выполнить содержательную часть обработки прерывания.

Обнулить устройство, вызвавшее прерывание, или обнулить разряд регистра VICSOFTINT, записав единицу в соответствующий разряд регистра VICSOFTINTCLEAR, если прерывание вызвано программным способом.

Выполнить возврат из процедуры обработки прерывания. При этом будет восстановлено состояние процессора, таким, как оно было в момент прерывания. В частности, будут разрешены FIQ прерывания.

5.6.4 Регистровая модель системного контроллера прерываний VIC

Программно доступные регистры системного контроллеров прерываний VIC0 и VIC1 расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение VIC0 Base = 0xFFF20000, VIC1 Base = 0xFFF30000 и общий размер 4 Кб для каждого. В отличие от других периферийных блоков VIC имеет дополнительное отражении на карту памяти по адресу 0xFFFF0000. Это сделано для ускорения входа в процедуру обработчика прерываний.

VIC0 и VIC1 имеют одинаковый набор программно доступных регистров, спецификация регистров представлена в таблице ниже:

Таблица 5.36 - Спецификация регистров контроллера VIC

Адрес	Тип	Разр.	Начальное значение	Имя	Описание
VIC Base + 0x000	ЧТ	32	0x00000000	VICIRQSTATUS	Состояние обычных прерываний (IRQ)
VIC Base + 0x004	ЧТ	32	0x00000000	VICFIQSTATUS	Состояние быстрых прерываний (FIQ)
VIC Base + 0x008	ЧТ	32	-	VICRAWINTR	Состояние входов
VIC Base + 0x00C	ЧТ/ ЗП	32	0x00000000	VICINTSELECT	Выбор типа прерывания (IRQ/FIQ, 1 – FIQ)
VIC Base + 0x010	ЧТ/ ЗП	32	0x00000000	VICINTENABLE	Маска прерываний
VIC Base + 0x014	ЗП	32	-	VICINTENCLEAR	Обнуление регистра маски прерываний
VIC Base + 0x018	ЧТ/ ЗП	32	0x00000000	VICSOFTINT	Программируемые прерывания
VIC Base + 0x01C	ЗП	32	-	VICSOFTINTCLEAR	Обнуление регистра программируемых прерывания
VIC Base + 0x020	ЧТ/ ЗП	1	0x0	VICPROTECTION	Флаг защищенного режима
VIC Base + 0x024	ЧТ/ ЗП	16	0xFFFF	VICSWPRIORITY MASK	Маска программных приоритетов
VIC Base + 0x028	ЧТ/ ЗП	4	0xF	VICPRIORITYDAISY	Приоритет подчиненного VIC (Не используется)
VIC Base + 0x100	ЧТ/ ЗП	32	0x00000000	VICVECTADDR0	Адрес процедуры обработки прерывания 0
VIC Base + 0x104	ЧТ/ ЗП	32	0x00000000	VICVECTADDR1	Адрес процедуры обработки прерывания 1
VIC Base + 0x108	ЧТ/ ЗП	32	0x00000000	VICVECTADDR2	Адрес процедуры обработки прерывания 2
VIC Base + 0x10C	ЧТ/ ЗП	32	0x00000000	VICVECTADDR3	Адрес процедуры обработки прерывания 3
VIC Base + 0x110	ЧТ/ ЗП	32	0x00000000	VICVECTADDR4	Адрес процедуры обработки прерывания 4
VIC Base + 0x114	ЧТ/ ЗП	32	0x00000000	VICVECTADDR5	Адрес процедуры обработки прерывания 5
VIC Base + 0x118	ЧТ/ ЗП	32	0x00000000	VICVECTADDR6	Адрес процедуры обработки прерывания 6
VIC Base + 0x11C	ЧТ/ ЗП	32	0x00000000	VICVECTADDR7	Адрес процедуры обработки прерывания 7
VIC Base + 0x120	ЧТ/ ЗП	32	0x00000000	VICVECTADDR8	Адрес процедуры обработки прерывания 8
VIC Base + 0x124	ЧТ/ ЗП	32	0x00000000	VICVECTADDR9	Адрес процедуры обработки прерывания 9
VIC Base + 0x128	ЧТ/ ЗП	32	0x00000000	VICVECTADDR10	Адрес процедуры обработки прерывания 10
VIC Base + 0x12C	ЧТ/ ЗП	32	0x00000000	VICVECTADDR11	Адрес процедуры обработки прерывания 11
VIC Base + 0x130	ЧТ/ ЗП	32	0x00000000	VICVECTADDR12	Адрес процедуры обработки прерывания 12
VIC Base + 0x134	ЧТ/ ЗП	32	0x00000000	VICVECTADDR13	Адрес процедуры обработки прерывания 13
VIC Base + 0x138	ЧТ/ ЗП	32	0x00000000	VICVECTADDR14	Адрес процедуры обработки прерывания 14
VIC Base + 0x13C	ЧТ/ ЗП	32	0x00000000	VICVECTADDR15	Адрес процедуры обработки прерывания 15
VIC Base + 0x140	ЧТ/ ЗП	32	0x00000000	VICVECTADDR16	Адрес процедуры обработки прерывания 16

VIC Base + 0x144	ЧТ/ ЗП	32	0x00000000	VICVECTADDR17	Адрес процедуры обработки прерывания 17
VIC Base + 0x148	ЧТ/ ЗП	32	0x00000000	VICVECTADDR18	Адрес процедуры обработки прерывания 18
VIC Base + 0x14C	ЧТ/ ЗП	32	0x00000000	VICVECTADDR19	Адрес процедуры обработки прерывания 19
VIC Base + 0x150	ЧТ/ ЗП	32	0x00000000	VICVECTADDR20	Адрес процедуры обработки прерывания 20
VIC Base + 0x154	ЧТ/ ЗП	32	0x00000000	VICVECTADDR21	Адрес процедуры обработки прерывания 21
VIC Base + 0x158	ЧТ/ ЗП	32	0x00000000	VICVECTADDR22	Адрес процедуры обработки прерывания 22
VIC Base + 0x15C	ЧТ/ ЗП	32	0x00000000	VICVECTADDR23	Адрес процедуры обработки прерывания 23
VIC Base + 0x160	ЧТ/ ЗП	32	0x00000000	VICVECTADDR24	Адрес процедуры обработки прерывания 24
VIC Base + 0x164	ЧТ/ ЗП	32	0x00000000	VICVECTADDR25	Адрес процедуры обработки прерывания 25
VIC Base + 0x168	ЧТ/ ЗП	32	0x00000000	VICVECTADDR26	Адрес процедуры обработки прерывания 26
VIC Base + 0x16C	ЧТ/ ЗП	32	0x00000000	VICVECTADDR27	Адрес процедуры обработки прерывания 27
VIC Base + 0x170	ЧТ/ ЗП	32	0x00000000	VICVECTADDR28	Адрес процедуры обработки прерывания 28
VIC Base + 0x174	ЧТ/ ЗП	32	0x00000000	VICVECTADDR29	Адрес процедуры обработки прерывания 29
VIC Base + 0x178	ЧТ/ ЗП	32	0x00000000	VICVECTADDR30	Адрес процедуры обработки прерывания 30
VIC Base + 0x17C	ЧТ/ ЗП	32	0x00000000	VICVECTADDR31	Адрес процедуры обработки прерывания 31
VIC Base + 0x200	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY0	Программный приоритет 0
VIC Base + 0x204	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY1	Программный приоритет 1
VIC Base + 0x208	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY2	Программный приоритет 2
VIC Base + 0x20C	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY3	Программный приоритет 3
VIC Base + 0x210	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY4	Программный приоритет 4
VIC Base + 0x214	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY5	Программный приоритет 5
VIC Base + 0x218	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY6	Программный приоритет 6
VIC Base + 0x21C	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY7	Программный приоритет 7
VIC Base + 0x220	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY8	Программный приоритет 8
VIC Base + 0x224	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY9	Программный приоритет 9
VIC Base + 0x228	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY10	Программный приоритет 10
VIC Base + 0x22C	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY11	Программный приоритет 11
VIC Base + 0x230	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY12	Программный приоритет 12
VIC Base + 0x234	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY13	Программный приоритет 13
VIC Base	ЧТ/	4	0xF	VICVECTPRIORITY14	Программный приоритет 14

+ 0x238	ЗП				
VIC Base + 0x23C	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY15	Программный приоритет 15
VIC Base + 0x240	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY16	Программный приоритет 16
VIC Base + 0x244	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY17	Программный приоритет 17
VIC Base + 0x248	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY18	Программный приоритет 18
VIC Base + 0x24C	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY19	Программный приоритет 19
VIC Base + 0x250	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY20	Программный приоритет 20
VIC Base + 0x254	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY21	Программный приоритет 21
VIC Base + 0x258	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY22	Программный приоритет 22
VIC Base + 0x25C	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY23	Программный приоритет 23
VIC Base + 0x260	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY24	Программный приоритет 24
VIC Base + 0x264	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY25	Программный приоритет 25
VIC Base + 0x268	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY26	Программный приоритет 26
VIC Base + 0x26C	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY27	Программный приоритет 27
VIC Base + 0x270	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY28	Программный приоритет 28
VIC Base + 0x274	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY29	Программный приоритет 29
VIC Base + 0x278	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY30	Программный приоритет 30
VIC Base + 0x27C	ЧТ/ ЗП	4	0xF	VICVECTPRIORITY31	Программный приоритет 31
VIC Base + 0xF00	ЧТ/ ЗП	32	0x00000000	VICADDRESS	Адрес процедуры обработки для текущего IRQ прерывания
VIC Base + 0xFE0	ЧТ	8	0x92	VICPERIPHID0	Идентификатор периферийного устройства, разряды [7:0]
VIC Base + 0xFE4	ЧТ	8	0x11	VICPERIPHID1	Идентификатор периферийного устройства, разряды [15:8]
VIC Base + 0xFE8	ЧТ	8	0x04	VICPERIPHID2	Идентификатор периферийного устройства, разряды [23:16]
VIC Base + 0xFEC	ЧТ	8	0x00	VICPERIPHID3	Идентификатор периферийного устройства, разряды [31:24]
VIC Base + 0xFF0	ЧТ	8	0x0D	VICPCELLID0	Идентификатор PrimeCell, разряды [7:0]
VIC Base + 0xFF4	ЧТ	8	0xF0	VICPCELLID1	Идентификатор PrimeCell, разряды [15:8]
VIC Base + 0xFF8	ЧТ	8	0x05	VICPCELLID2	Идентификатор PrimeCell, разряды [23:16]
VIC Base + 0xFFC	ЧТ	8	0xB1	VICPCELLID3	Идентификатор PrimeCell, разряды [31:24]

Далее приведено подробное описание каждого регистра.

5.6.4.1 Регистр состояния обычных прерываний (IRQ). VICIRQSTATUS

Таблица 5.37 – Формат регистра VICIRQSTATUS

Биты	Название	Тип	Выполняемая функция
[31:0]	IRQStatus	ЧТ/ЗП	VICIRQSTATUS[i] = 0 – нет запроса на обычное прерывание VICIRQSTATUS[i] = 1 – есть запрос на обычное прерывание

Каждый разряд регистра отражает состояние источника прерываний с учетом масок VICINTENABLE, VICPRIORITYMASK и VICINTSELECT.

5.6.4.2 Регистр состояния быстрых прерываний (FIQ). VICFIQSTATUS

Таблица 5.38 – Формат регистра VICFIQSTATUS

Биты	Название	Тип	Выполняемая функция
[31:0]	FIQStatus	ЧТ	VICFIQSTATUS[i] = 0 – нет запроса на быстрое прерывание VICFIQSTATUS[i] = 1 – есть запрос на быстрое прерывание

Каждый разряд регистра отражает состояние источника прерываний с учетом масок VICINTENABLE и VICINTSELECT.

5.6.4.3 Регистр состояния входов. VICRAWINTR

Таблица 5.39 –Формат регистра VICRAWINTR

Биты	Название	Тип	Выполняемая функция
[31:0]	RawInterrupt	ЧТ	VICRAWINTR[i] = 0 – нет запроса на прерывание VICRAWINTR[i] = 1 – есть запрос на прерывание

Каждый разряд регистра отражает состояние источника прерываний на входе VIC, т.е. до наложения масок. Состояние всех входов VIC отображается в регистре VICRAWINTR. Каждый разряд регистра соответствует входу с номером, равным номеру разряда. Значение разряда 0 означает отсутствие прерывания на входе, а 1 – его наличие.

5.6.4.4 Регистр выбора типа прерывания. VICINTSELECT

Таблица 5.40 – Формат регистра VICINTSELECT

Биты	Название	Тип	Выполняемая функция
[31:0]	IntSelect	ЧТ/ЗП	VICINTSELECT[i] = 0 – обычное прерывание (IRQ) VICINTSELECT[i] = 1 – быстрое прерывание (FIQ)

Тип прерывания определяется значением разряда регистра VICSELECT, номер которого равен номеру прерывания. Для FIQ значение разряда 1, для IRQ – 0.

5.6.4.5 Регистр маски прерываний. VICINTENABLE

Таблица 5.41 – Формат регистра VICINTENABLE

Биты	Название	Тип	Выполняемая функция
[31:0]	IntEnable	ЧТ/ЗП	VICINTENABLE[i] = 0 – прерывание заблокировано VICINTENABLE[i] = 1 – прерывание разрешено

Прерывание может быть закрыто программным или аппаратным способом. Сигнал закрытого прерывания, поступивший на вход VIC, не передается на выход.

Прерывание с номером n закрыто программным способом, если в n-м разряде регистра VICINTENABLE записан ноль и/или если это IRQ прерывание с программным приоритетом p и в p-м разряде регистра VICPRIORITYMASK записан ноль.

IRQ прерывание с номером n закрыто аппаратным способом, если его приоритет равен или ниже приоритета прерывания, обрабатываемого процессором в данный момент.

Заблокированные прерывания не передаются процессору и не отражаются в регистрах VICFIQSTATUS и VICIRQSTATUS. Запись единицы в соответствующий разряд регистра VICINTENABLE разрешает прерывание. Запись нуля игнорируется. Для обнуления разряда регистра VICINTENABLE надо записать единицу в соответствующий разряд регистра VICINTENCLEAR.

5.6.4.6 Регистр обнуления маски прерываний. VICINTENCLEAR

Таблица 5.42 – Формат регистра VICINTENCLEAR

Биты	Название	Тип	Выполняемая функция
[31:0]	IntEnable Clear	ЗП	Запись нуля игнорируется. Запись единицы в разряд регистра VICINTENCLEAR вызывает обнуление соответствующего разряда регистра VICINTENABLE.

5.6.4.7 Регистр программных прерываний. VICSOFTINT

Таблица 5.43 – Формат регистра VICSOFTINT

Биты	Название	Тип	Выполняемая функция
[31:0]	SoftInt	ЧТ/ЗП	VICSOFTINT [i] = 0 – программное прерывание не установлено VICSOFTINT [i] = 1 – программное прерывание установлено

Регистр программных прерываний VICSOFTINT позволяет процессору эмулировать внешние прерывания программным способом. Запись единицы в *n*-ый разряд регистра VICSOFTINT приводит к тому же результату, что и появление сигнала прерывания на *n*-м входе VIC.

Запись единицы в разряд регистра VICSOFTINT эквивалентна появлению сигнала прерывания на соответствующем входе VIC. Запись нуля игнорируется. Для обнуления разряда регистра VICSOFTINT надо записать единицу в соответствующий разряд регистра VICSOFTINTCLEAR.

5.6.4.8 Регистр обнуления программных прерываний. VICSOFTINTCLEAR

Таблица 5.44 – Формат регистра VICSOFTINTCLEAR

Биты	Название	Тип	Выполняемая функция
[31:0]	SoftIntClear	ЗП	Запись нуля игнорируется. Запись единицы вызывает обнуление соответствующего разряда регистра VICSOFTINT

5.6.4.9 Флаг защищенного режима. VICPROTECTION

Таблица 5.45 – Формат регистра VICPROTECTION

Биты	Название	Тип	Функции
[31:1]	-	-	Резервные биты, читаются как нуль, не изменяются
[0]	Protection	ЧТ/ЗП	VICPROTECTION [0] = 0 – незащищенный режим VIC VICPROTECTION [0] = 1 – защищенный режим VIC

В защищенном режиме доступ к регистрам VIC как по чтению, так и по записи, возможен только в привилегированном режиме. В незащищенном режиме доступ к регистрам VIC, кроме VICPROTECTION, свободный. Доступ к регистру VICPROTECTION возможен только в привилегированном режиме.

5.6.4.10 Регистр маски программных приоритетов. VICSWPRIORITYMASK

Таблица 5.46 – Формат регистра VICSWPRIORITYMASK

Биты	Название	Тип	Функции
[31:16]	-	-	Резервные биты, читаются как нуль, не изменяются
[15:0]	SWPriorityMask	ЧТ/ЗП	VICINTSELECT [i] = 0, i < 16 – обычные прерывания (IRQ) с программным приоритетом i заблокированы VICINTSELECT [i] = 1, i < 16 – обычные прерывания (IRQ) с программным приоритетом i разрешены

Заблокированные с помощью маски VICSWPRIORITYMASK прерывания не передаются процессору и не отражаются в регистре VICIRQSTATUS.

5.6.4.11 Регистры адресов процедур обработки прерывания. VICVECTADDR[0:31]

Таблица 5.47 – Формат регистра VICVECTADDR[0:31]

Биты	Название	Тип	Выполняемая функция
[31:0]	VectorAddr[31:0]	ЧТ/ЗП	Содержит векторные адреса ISR(процедур обработчиков прерываний). Запись в регистр должна выполняться, когда соответствующее прерывание блокировано. В противном случае, результат не определен.

5.6.4.12 Регистры приоритета обычных прерываний. VICVECTPRIORITY[0:31]

Таблица 5.48 – Формат регистра VICVECTPRIORITY[0:31]

Биты	Название	Тип	Функции
[31:4]	-	-	Резервные биты, читаются как нуль, не изменяются
[3:0]	VectPriority	ЧТ/ЗП	VICINTSELECT*n*[3:0] – приоритет прерывания *n*

Запись в регистр должна выполняться, когда соответствующее прерывание блокировано. В противном случае результат не определен.

В VIC приоритеты связаны только с прерываниями IRQ. Приоритет прерывания определяется на основе программного и аппаратного приоритетов.

Программный приоритет – это значение соответствующего 4-х разрядного регистра приоритета VICVECTPRIORITY*n*, где *n* – номер прерывания. Значение программного приоритета – целое число от 0 до 15. 0 – высший приоритет, 15 – низший.

Аппаратный приоритет равен номеру прерывания на входе. 0 – высший приоритет, 31 – низший.

При сравнении приоритетов сначала используется программный приоритет, а при равенстве программных приоритетов – аппаратный.

5.6.4.13 Регистр приоритета подчиненного VIC. VICSWPRIORITYDAISY

Регистр приоритета подчиненного VIC не используется, т.к. в СБИС СНП-ВП нет каскадирования контроллеров прерываний. Используется только один системный контроллер прерываний VIC.

5.6.4.14 Регистр адреса процедуры обработки текущего IRQ прерывания. VICADDRESS

Таблица 5.49– Формат регистра VICADDRESS

Биты	Название	Тип	Выполняемая функция
[31:0]	VectAddr	ЧТ/ЗП	Регистр содержит значение регистра VICVECTADDR*n*, где *n* – номер IRQ прерывания, переданного для обработки процессору, а если такого нет, то значение регистра VICVECTADDR*n* для последнего обработанного IRQ прерывания.

Адрес процедуры обработки прерывания находится в регистре VICVECTADDR*n*, где *n* – номер прерывания. Для IRQ прерываний значение этого регистра передается процессору вместе с сигналом прерывания и одновременно записывается в регистр VICVECTADDR. Для FIQ прерываний регистры VICVECTADDR*n* не используются.

Считывание из регистра VICVECTADDR должно выполняться в начале процедуры обработки прерывания, и воспринимается VIC как сигнал о начале обработки IRQ прерывания. По этому сигналу VIC аппаратно блокирует IRQ прерывания с приоритетом меньшим или равным приоритету обрабатываемого прерывания и возобновляет обработку IRQ прерываний.

Запись в регистр VICVECTADDR должна выполняться после окончания обработки прерывания, непосредственно перед выходом из процедуры обработки прерывания. Такая запись, служащая сигналом об окончании обработки прерывания, не меняет значения регистра. По этому сигналу VIC снимает аппаратную блокировку IRQ прерываний с приоритетом меньшим или равным приоритету обрабатываемого прерывания.

Результат считывания регистра VICVECTADDR или записи в этот регистр во всех случаях, кроме описанных выше, не определен.

5.6.4.15 Идентификатор периферийного устройства. VICPERIPHID0-3

Таблица 5.50 – Поля идентификатора периферийного устройства VICPERIPHID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для VIC – 0x192
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 VIC192 0x0

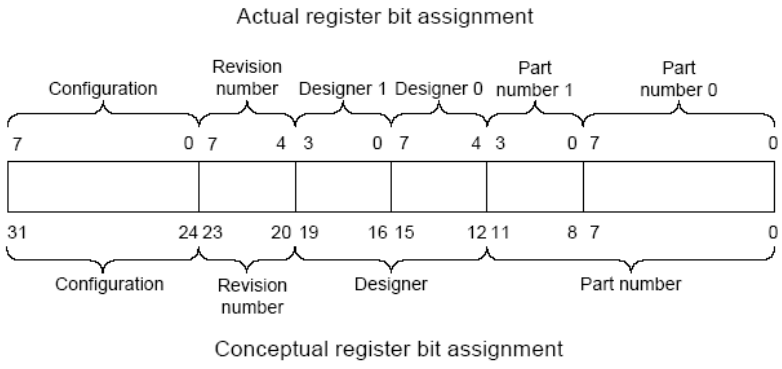


Рисунок 5.15 – Формат регистра VICPERIPHID0-3

5.6.4.16 PrimeCell идентификатор. VICPCELLID[0:3]

Регистры VICPCELLID[0:3] – 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. На рисунке ниже (см. Рисунок 5.16) приведено распределение полей этого регистра.

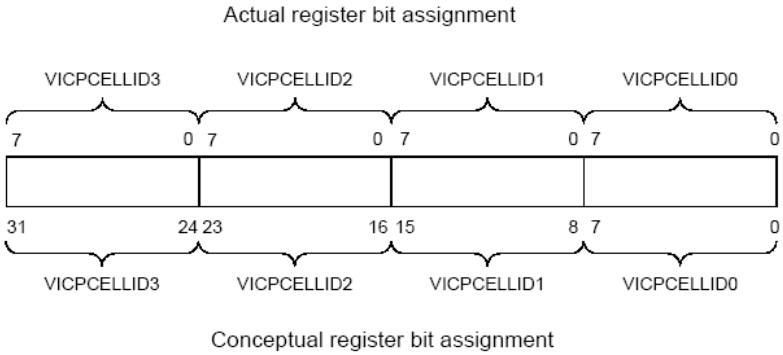


Рисунок 5.16 – Формат регистра VICPCELLID[0:3]

5.7 Контроллер внешних прерываний EIRC

Контроллер внешних прерываний EIRC в СБИС СНП-ВП имеет следующие характеристики:

- 4 внешних прерывания
- для каждого прерывания можно настроить его тип (по низкому уровню, по высокому уровню, по фронту, по срезу), а также маскировать это прерывание
- возможность использования внешнего прерывания для выхода из режима пониженного энергопотребления (STOP-режим)
- подключение к системной шине СБИС через AMBA APB интерфейс, используются 8 младших бит шины данных

5.7.1 Устройство контроллера внешних прерываний

В таблице ниже приведен список внешних выводов, относящихся к блоку EIRC.

Таблица 5.51- Выводы микросхемы, входящие в состав блока EIRC

Вывод	Тип буфера	Примечание
INT0 … INT3	in	4 входа внешних прерываний

На рисунке ниже приведена схема контроллера внешних прерываний:

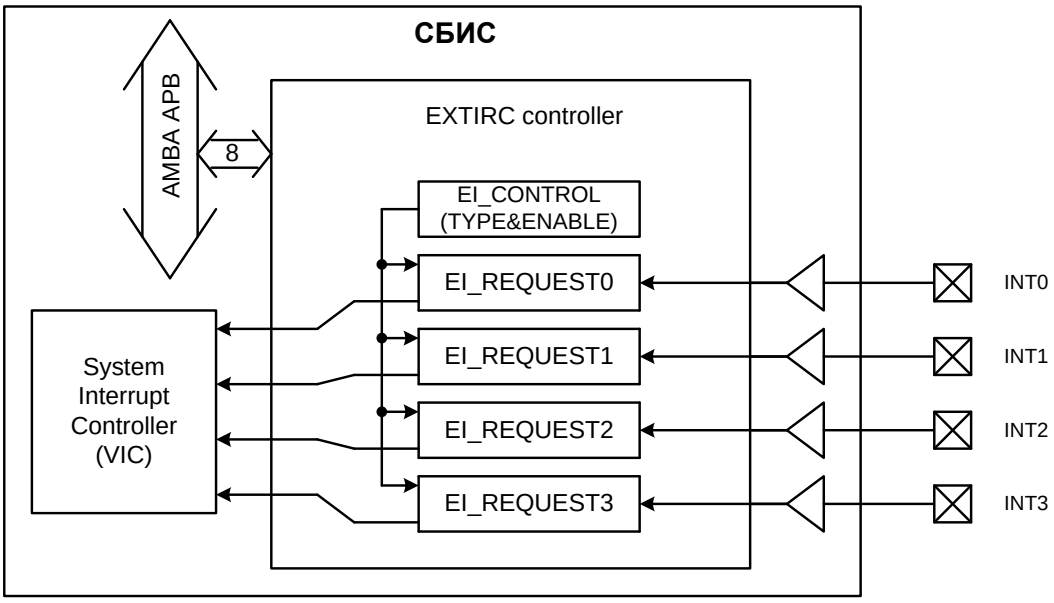


Рисунок 5.17 - Схема контроллера внешних прерываний

СБИС СНП-ВП имеет 4 входа внешних прерываний, каждое из которых раздельно обрабатывается системным контроллером прерываний VIC.

Для каждого из четырех внешних прерываний можно настроить тип прерывания, по которому контроллер будет считать внешнее прерывание активным:

- по низкому уровню
- по высокому уровню

- по фронту
- по срезу

В случае совпадения типа поданного внешнего прерывания и настроенного в контроллере выдается соответствующий сигнал к системному контроллеру прерываний (VIC).

5.7.2 Регистровая модель контроллера внешних прерываний EIRC

Программно доступные регистры контроллера внешних прерываний EIRC расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение EIRC Base = 0xFFF0E000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.52- Спецификация регистров контроллера EIRC

Адрес	Тип	Разрядность	Начальное значение	Имя	Описание
EIRC Base +0x00	ЧТ/ЗП	4	0x0	EIENB	Регистр разрешения внешних прерываний
EIRC Base +0x04	ЧТ/ЗП	4	0x0	EIREQ	Регистр запроса внешних прерываний
EIRC Base +0x08	ЧТ/ЗП	8	0x55	EILVL	Регистр уровня внешних прерываний
EIRC Base + 0x0C to 0xFFC	-	-	-	-	Зарезервировано

Далее приведено подробное описание каждого регистра.

5.7.2.1 Регистр разрешения внешних прерываний. EIENB

В таблице ниже приведен формат регистров EIENB.

Таблица 5.53 - Формат регистров EIENB

Биты	Название бит	Выполняемая функция
[31:4]	-	Зарезервировано
[3]	ENB3	Если бит ENB нуль, то по соответствующему входу внешний запрос на прерывание любого типа игнорируется. 0 – внешнее прерывание заблокировано (по умолчанию) 1 – внешнее прерывание разрешено
[2]	ENB2	
[1]	ENB1	
[0]	ENB0	

5.7.2.2 Регистр запроса внешних прерываний. EIREQ

В таблице ниже приведен формат регистра EIREQ.

Таблица 5.54 - Формат регистра EIREQ

Биты	Название бит	Выполняемая функция
[31:4]	-	Зарезервировано
[3]	REQ3	Если есть запрос на прерывание, бит REQ устанавливается в 1. 0 – Нет запроса на прерывание (по умолчанию) 1 – Есть запрос на прерывание
[2]	REQ2	
[1]	REQ1	
[0]	REQ0	

5.7.2.3 Регистр уровня внешних прерываний. EILVL

В таблице ниже приведен формат регистра EILVL.

Таблица 5.55 – Формат регистра EILVL

Биты	Название бит	Выполняемая функция
[31:8]	-	Зарезервировано
[7:6]	LVL3[1:0]	Ниже приведены возможные значения бит LVLx[1:0] для каждого из четырех каналов. Значение этого регистра определяет, при каком типе внешнего прерывания будет генерироваться внутреннее прерывание. 00 – внутреннее прерывание генерируется при низком уровне внешнего сигнала 01 – внутреннее прерывание генерируется при высоком уровне внешнего сигнала (по умолчанию) 10 – внутреннее прерывание генерируется по переднему фронту внешнего сигнала 11 - внутреннее прерывание генерируется по заднему фронту внешнего сигнала
[5:4]	LVL2[1:0]	
[3:2]	LVL1[1:0]	
[1:0]	LVL0[1:0]	

5.7.3 Прерывания

Как уже было отмечено выше, СБИС СНП-ВП имеет 4 входа внешних прерываний, каждое из которых раздельно обрабатывается системным контроллером прерываний VIC. Процедура инициализации контроллера для отслеживания внешнего прерывания следующая:

- сбросить соответствующий внешнему прерыванию бит регистра EIENB
- установить соответствующий внешнему прерыванию бит регистра EILVL
- сбросить соответствующий внешнему прерыванию бит регистра EIREQ
- установить соответствующий внешнему прерыванию бит регистра EIENB

Соответствующий внешнему прерыванию бит регистра EIENB должен быть сброшен во время установки бит других регистров блока. Также необходимо сбросить бит регистра EIREQ перед установкой бита регистра EIENB, т.к. иначе возможна генерация прерывания в момент инициализации контроллера EIRC.

При использовании внешнего прерывания для вывода СБИС из режима пониженного энергопотребления с отключенными синхросигналами (STOP-режим), необходимо выставить поле LVLx[1:0] регистра EILVL в значение 01 (“по высокому уровню”) до выключения синхросигналов. В противном случае будет невозможно вывести микросхему из STOP-режима с использованием внешнего прерывания.

5.8 Блок сдвоенных таймеров DIT

Блок сдвоенных таймеров в СБИС СНП-ВП имеет следующие характеристики:

- два 32/16 – разрядных таймера
- для каждого таймера можно настроить режим его работы (свободный счет, периодичный, одиночный)
- счетчики таймеров работают на частоте FDIT = FSYS
- каждый таймер имеет собственный выход прерывания
- подключение к системной шине СБИС через AMBA APB интерфейс, используются полный, 32 разрядный интерфейс шины данных

5.8.1 Устройство блока сдвоенных таймеров

Блок не имеет сигналов, соединенных с внешними выводами микросхемы.

На рисунке ниже приведена структурная схема блока:

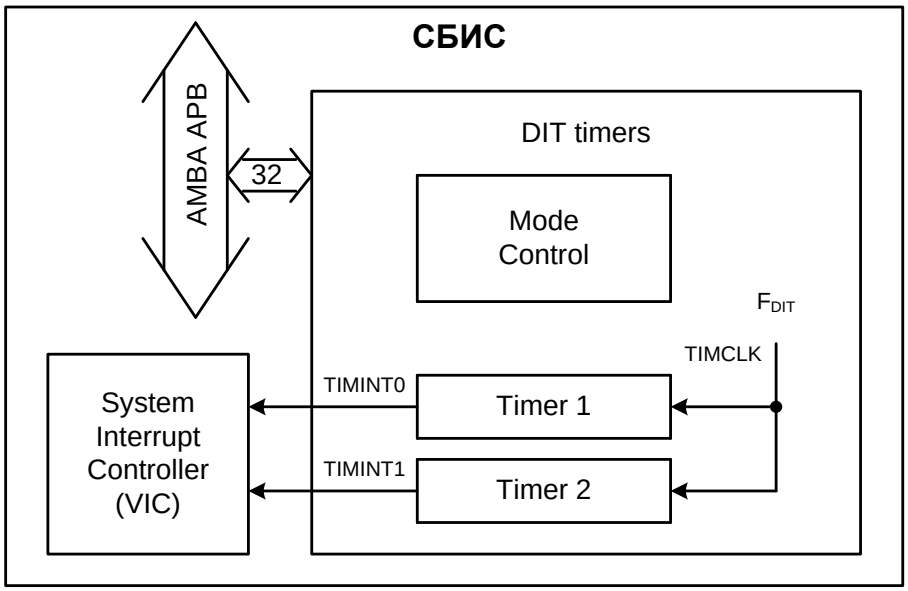


Рисунок 5.18 – Схема блока сдвоенных таймеров

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 счетчика переключаются на частоте F_{DIT}, причем возможно программным образом уменьшить частоту счета в 16 или 256 раз независимо для каждого счетчика.

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
- периодический (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.

5.8.2 Регистровая модель блока сдвоенных таймеров

Программно доступные регистры блока сдвоенных таймеров DIT расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение DIT Base = 0xFFF08000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.56- Спецификация регистров блока DIT

Адрес	Тип	Разряд-ность	Начальное значение	Имя	Описание
DIT Base + 0x00	ЧТ/ЗП	32	0x00000000	Timer1Load	Регистр загрузки. TimerXLoad
DIT Base + 0x04	ЧТ	32	0xFFFFFFFF	Timer1Value	Регистр текущего значения. TimerXValue
DIT Base + 0x08	ЧТ/ЗП	8	0x20	Timer1Control	Регистр управления. TimerXControl
DIT Base + 0x0C	ЗП	-	-	Timer1IntClr	Регистр снятия прерываний. TimerXIntClr
DIT Base + 0x10	ЧТ	1	0x0	Timer1RIS	Регистр состояния прерывания до наложения маски. TimerXRIS
DIT Base + 0x14	ЧТ	1	0x0	Timer1MIS	Регистр состояния прерывания после наложения маски. TimerXMIS
DIT Base + 0x18	ЧТ/ЗП	32	0x00000000	Timer1BGLoad	Фоновый регистр загрузки. TimerXBGLoad
DIT Base + 0x20	ЧТ/ЗП	32	0x00000000	Timer2Load	Регистр загрузки. TimerXLoad
DIT Base + 0x24	ЧТ	32	0xFFFFFFFF	Timer2Value	Регистр текущего значения. TimerXValue
DIT Base + 0x28	ЧТ/ЗП	8	0x20	Timer2Control	Регистр управления. TimerXControl
DIT Base + 0x2C	ЗП	-	-	Timer2IntClr	Регистр снятия прерываний. TimerXIntClr
DIT Base + 0x30	ЧТ	1	0x0	Timer2RIS	Регистр состояния прерывания до наложения маски. TimerXRIS
DIT Base + 0x34	ЧТ	1	0x0	Timer2MIS	Регистр состояния прерывания после наложения маски. TimerXMIS
DIT Base + 0x38	ЧТ/ЗП	32	0x00000000	Timer2BGLoad	Фоновый регистр загрузки. TimerXBGLoad
DIT Base + 0x40 to 0xEFC	-	-	-	-	Зарезервировано
DIT Base + 0xF00	ЧТ/ЗП	1	0x0	TimerITCR	Регистр управления тестом инте-грации. TimerITCR
DIT Base + 0xF04	ЗП	2	0x0	TimerITOP	Регистр управления тестом инте-грации. TimerITOP
DIT Base + 0xF08 to 0xFDC	-	-	-	-	Зарезервировано
DIT Base + 0xFE0	ЧТ	8	0x04	TimerPeriphID0	Идентификатор периферийного устройства. TimerPeriphID0 биты [7:0]
DIT Base + 0xFE4	ЧТ	8	0x18	TimerPeriphID1	Идентификатор периферийного устройства. TimerPeriphID1 биты [15:8]
DIT Base + 0xFE8	ЧТ	8	0x04	TimerPeriphID2	Идентификатор периферийного устройства. TimerPeriphID2 биты [23:16]
DIT Base + 0xFEC	ЧТ	8	0x00	TimerPeriphID3	Идентификатор периферийного устройства. TimerPeriphID3 биты [31:24]
DIT Base + 0xFF0	ЧТ	8	0x0D	TimerPCellID0	PrimeCell идентификатор. TimerPCellID0 биты [7:0]
DIT Base + 0xFF4	ЧТ	8	0xF0	TimerPCellID1	PrimeCell идентификатор. TimerPCellID1 биты [15:8]

DIT Base + 0xFF8	ЧТ	8	0x05	TimerPCellID2	PrimeCell идентификатор. TimerPCellID2 биты [23:16]
DIT Base + 0xFFC	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.

5.8.2.1 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 всегда то значение, которое будет прописано в счетчике, как только он станет равным нулю в периодическом режиме.

5.8.2.2 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.

5.8.2.3 Регистр управления. TimerXControl

Формат регистра управления приведен в таблице ниже.

Таблица 5.57 - 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.

5.8.2.4 Регистр снятия прерываний. TimerXIntClr

Любая запись в данный регистр снимает прерывание с выхода счетчика.

5.8.2.5 Регистр состояния прерывания до наложения маски. TimerXRIS

Регистр TimerXRIS указывает на состояние прерываний до наложения маски от счетчика. Формат регистра представлен в таблице ниже.

Таблица 5.58 - Формат регистра TimerXRIS

Биты	Имя	Тип	Функция
[31:1]	-	-	Резервные биты, не изменяются, игнорируются при чтении
[0]	TimerXRIS	ЧТ	Состояние необработанных прерываний счетчика

5.8.2.6 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 ниже.

Таблица 5.59 - 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

5.8.2.7 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».

5.8.2.8 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 ниже приведен формат полей этого регистра.

Таблица 5.60- 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

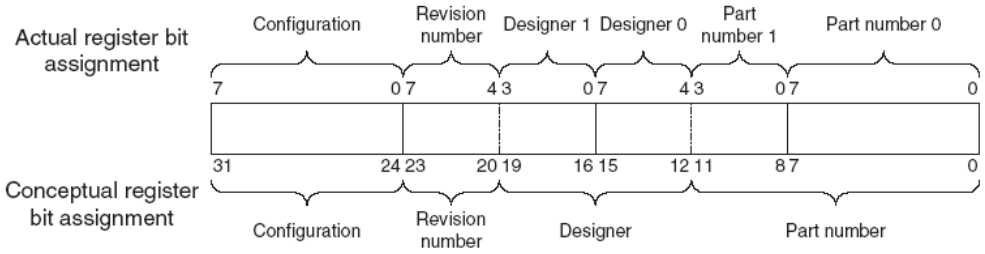


Рисунок 5.19 – Формат регистра TimerPeriphID0-3

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.

5.8.2.9 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 ниже приведен формат этого регистра.

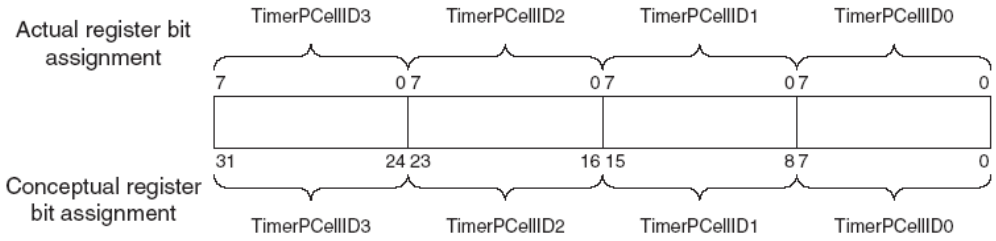


Рисунок 5.20 – Формат регистра TimerPCellID0-3

5.8.2.10 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 ниже.

Таблица 5.61 - 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.

5.8.2.11 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 ниже.

Таблица 3-15 – 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

5.8.3 Прерывания

Блок имеет 2 прерывания, каждое из которых подключено к системному контроллеру прерываний (VIC0).

5.9 Сторожевой таймер WDOG

Блок сторожевого таймера в СБИС СНП-ВП имеет следующие характеристики:

- 32– разрядный счетчик, с программируемым временем срабатывания
- счетчик таймера работает на частоте FWDOG = FSYS
- выход прерывания
- подключение к системной шине СБИС через AMBA APB интерфейс, используются полный, 32 разрядный интерфейс шины данных

5.9.1 Устройство блока сторожевого таймера

На рисунке ниже приведена структурная схема блока:

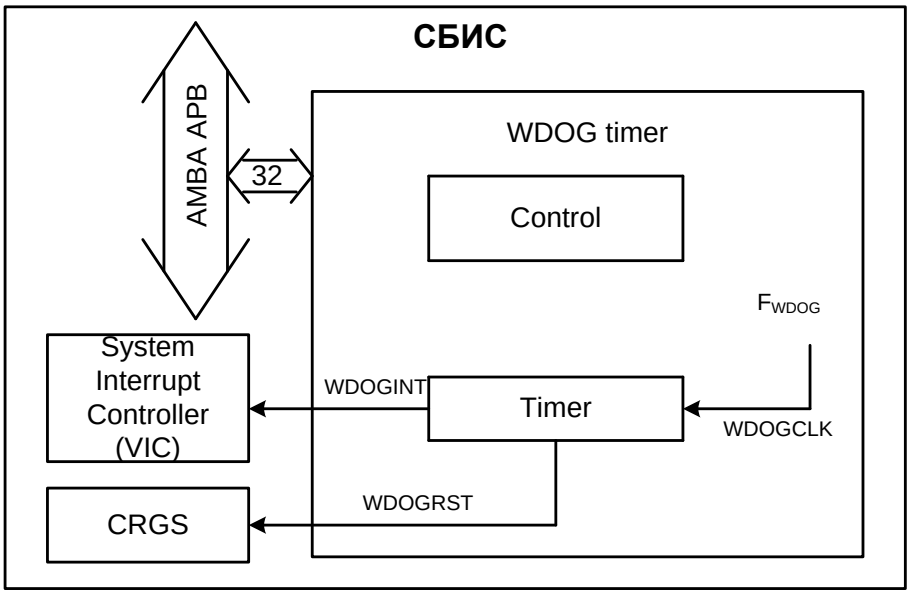


Рисунок 5.21 – Схема блока сторожевого таймера

Блок состоит из программируемого 32-разрядного таймера, работающего на частоте F_{WDOG}, а также схемы управления.

После снятия сигнала системного сброса блок находится в выключенном состоянии. Принцип работы с блоком следующий:

Программным образом задается интервал счета таймера, после чего счетчик начинает декрементироваться.

Когда счетчик достигает нулевого значения, генерируется сигнал прерывания WDOGINT, а счетчик перезаписывается начальным значением. Если до достижения счетчиком нулевого значения, прерывание не будет обработано (сброшено), то будет сформирован сигнал WDOGRST, который, если это разрешено в настройках CRGS, приведет к сбросу СБИС. Также сигнал WDOGRST поступает на внешний вывод микросхемы WDT.

Для исключения случайного сброса СБИС вследствие неправильной записи в регистры блока сторожевого таймера, предусмотрена возможность программно заблокировать эти регистры для записи.

5.9.2 Регистровая модель блока сторожевого таймера

Программно доступные регистры блока сторожевого таймера WDOG расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение WDOG Base = 0xFFF09000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.62 - Спецификация регистров блока WDOG

Адрес	Тип	Разрядность	Начальное значение	Имя	Описание
WDOG Base + 0x00	ЧТ/ЗП	32	0xFFFFFFFF	WdogLoad	Регистр загрузки. WdogLoad
WDOG Base + 0x04	ЧТ	32	0xFFFFFFFF	WdogValue	Регистр значения. WdogValue
WDOG Base + 0x08	ЧТ/ЗП	2	0x0	WdogControl	Регистр управления. WdogControl
WDOG Base + 0x0C	ЗП	-	-	WdogIntClr	Регистр снятия прерываний. WdogIntClr
WDOG Base + 0x10	ЧТ	1	0x0	WdogRIS	Регистр состояния прерывания до наложения маски. WdogRIS
WDOG Base + 0x14	ЧТ	1	0x0	WdogMIS	Регистр состояния прерывания после наложения маски. WdogMIS
WDOG Base + 0x18 to 0xBFC	-	-	-	-	Зарезервировано
WDOG Base + 0xC00	ЧТ/ЗП	32	0x0	WdogLock	Регистр блокировки. WdogLock
WDOG Base + 0xC04 to 0xEFC	-	-	-	-	Зарезервировано
WDOG Base + 0xF00	ЧТ/ЗП	1	0x0	WdogITCR	Регистр управления тестом аннграции. WdogITCR
WDOG Base + 0xF04	ЗП	2	0x0	WdogITOP	See Integration Test Output Set Register, WdogITOP
WDOG Base + 0xF08 to 0xFDC	-	-	-	-	Зарезервировано
WDOG Base + 0xFE0	ЧТ	8	0x05	WdogPeriphID0	Идентификатор периферийного устройства. WdogPeriphID0
WDOG Base + 0xFE4	ЧТ	8	0x18	WdogPeriphID1	Идентификатор периферийного устройства. WdogPeriphID1
WDOG Base + 0xFE8	ЧТ	8	0x14	WdogPeriphID2	Идентификатор периферийного устройства. WdogPeriphID2
WDOG Base + 0xFEC	ЧТ	8	0x00	WdogPeriphID3	Идентификатор периферийного устройства. WdogPeriphID3
WDOG Base + 0xFF0	ЧТ	8	0x0D	WdogPCellID0	Идентификатор PrimeCell. WdogPCellID0
WDOG Base + 0xFF4	ЧТ	8	0xF0	WdogPCellID1	Идентификатор PrimeCell. WdogPCellID1
WDOG Base + 0xFF8	ЧТ	8	0x05	WdogPCellID2	Идентификатор PrimeCell. WdogPCellID2
WDOG Base + 0xFFC	ЧТ	8	0xB1	WdogPCellID3	Идентификатор PrimeCell. WdogPCellID3

Далее приведено подробное описание каждого регистра.

5.9.2.1 Регистр загрузки. WdogLoad

Представляет 32-битный регистр, доступный на запись и чтение, который хранит значение, с которого счетчик начинает декрементироваться. Если в регистре записано некоторое значение, то счетчик незамедлительно перезагружается новым значением. Минимально возможным значением для WdogLoad является 1. Если значение регистра WdogLoad установлено в нуль, то прерывания вырабатываются незамедлительно.

5.9.2.2 Регистр значения. WdogValue

Представляет 32-битный регистр, доступный только на чтение, выдает текущее значение декрементирующего счетчика.

5.9.2.3 Регистр управления. WdogControl

Представляет регистр, доступный на запись и чтение, который разрешает программному обеспечению (ПО) управлять сторожевым таймером. В таблице ниже приведен формат регистра WdogControl.

Таблица 5.63 - 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.

5.9.2.4 Регистр снятия прерываний. WdogIntClr

Любая запись в данный регистр, снимает прерывание WDOG, и перезагружает счетчик значением регистра WdogLoad.

5.9.2.5 Регистр состояния прерывания до наложения маски. WdogRIS

Данный регистр определяет состояние прерываний до наложения маски, формируемых счетчиком. Регистр WdogRIS возводится в единицу в момент, когда счетчик сторожевого таймера достигает нуля. В таблице ниже приведен формат регистра WdogRIS.

Таблица 5.64 - Формат регистра WdogRIS

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервировано
[0]	WDOGRIS	ЧТ	Состояние прерывания от счетчика до наложения маски.

5.9.2.6 Регистр состояния прерывания после наложения маски. WdogMIS

Данный регистр определяет состояние прерывания счетчика после наложения маски. Это значение представляется логическим элементом «И» между битом WDOGRIS и битом INTEN регистра управления, и представляет аналогичное значение выходного сигнала WDOGINT. В таблице ниже приведен формат регистра WdogMIS.

Таблица 5.65 - Формат регистра WdogMIS

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервировано
[0]	WDOGMIS	ЧТ	Состояние прерывания от счетчика после наложения маски.

5.9.2.7 Регистр блокировки. WdogLock

Данный регистр может блокировать все другие регистры, доступные по записи. Таким образом, можно защитить регистры WDOG от некорректно работающего программного обеспечения. Запись значения 0x1ACCE551 разблокирует все регистры блока, т.е. разрешает программный доступ по записи ко всем регистрам. Запись любого другого значения блокирует доступ по записи. Чтение из этого регистра возвращает состояние блокировки:

- 0 – доступ по записи разрешен (нет блокировки)
- 1 – доступ по записи запрещен (заблокирован)

В таблице ниже приведен формат регистра WdogLock.

Таблица 5.66 - 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

5.9.2.8 Идентификатор периферийного устройства. WdogPeriphID0-3

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 ниже приведен формат этого регистра.

Таблица 5.67 - 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 конфигурацией периферии. Для DIT 0x0

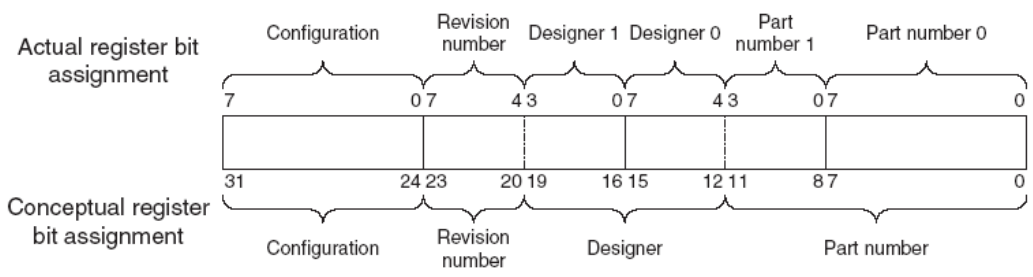


Рисунок 5.22 – Формат регистра WdogPeriphID0-3

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.

5.9.2.9 PrimeCell идентификатор. WdogPCellID0-3

Регистры 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 ниже приведен формат этого регистра.

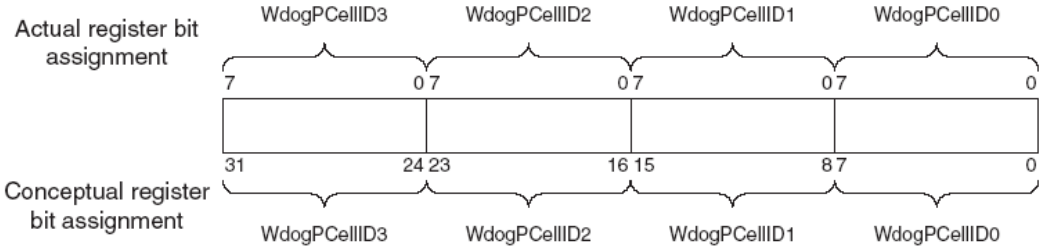


Рисунок 5.23 – Формат регистра WdogPCellID0-3

5.9.2.10 Регистр управления тестом интеграции. WdogITCR

Этот 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 ниже.

Таблица 5.68 - 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.

5.9.2.11 Регистр управления выходными сигналами. WdogITOP

Когда используется режим теста интеграции, выходные сигналы прерывания непосредственно управляются битами этого регистра. Формат регистра приведен в таблице ниже.

Таблица 5.69 - 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

5.9.3 Прерывания

Блок имеет выход прерывания, которое подключено к системному контроллеру прерываний (VIC0).

5.10 Таймер реального времени RTC

Блок RTC предназначен для счёта реального времени. В блоке имеются регистр текущего времени, в котором хранятся секунды, минуты, часы и дни недели, регистр текущей даты, содержащий число, месяц, год и столетие, регистр будильника, содержащий секунду, минуту, час и день недели срабатывания будильника.

RTC имеет функцию будильника – предоставляется возможность программной установки часа, минуты, секунды и (необязательно) дня недели срабатывания будильника. При срабатывании будильника генерируется прерывание.

5.10.1 Устройство блока таймера реального времени

Блок RTC состоит из двух основных частей: блока интерфейса и блока счётчиков (см. Рисунок 5.24). Блок счётчиков имеет отдельное питание, что обеспечивает возможность автономной работы счётчиков реального времени от внешней батареи, пока основная часть микросхемы отключена от питания.

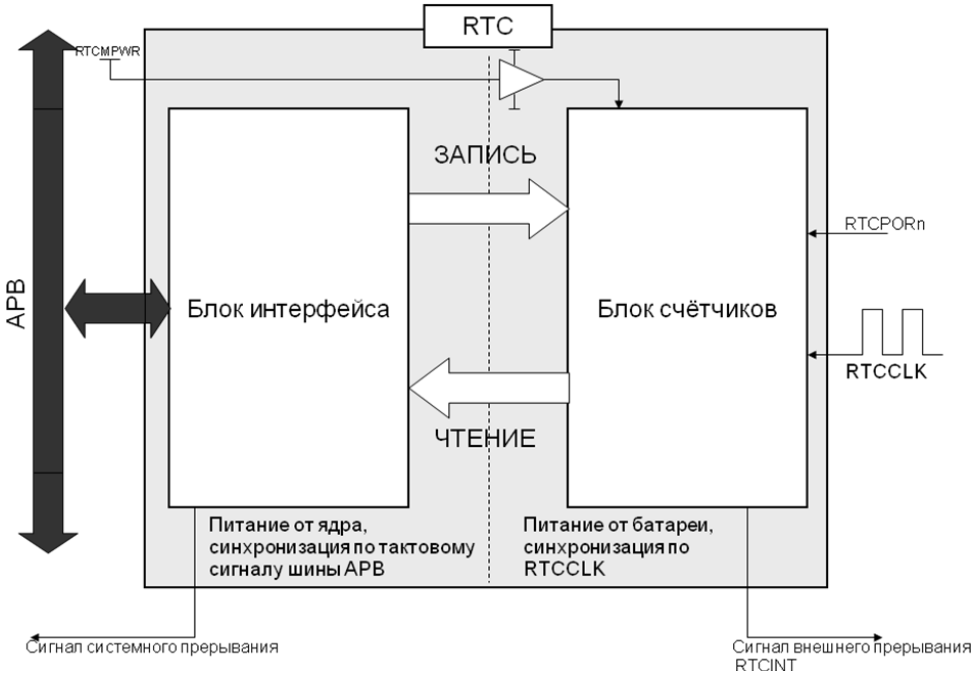


Рисунок 5.24 – Структурная схема таймера реального времени

Блок RTC имеет временной параметр – интервал записи. Запись в регистры автономной части может длиться от 1 до 2 интервалов записи. Запись производится, если всё это время на внешний вход RTCMPWR подан высокий уровень сигнала. Такой механизм позволяет избежать неожиданной записи при выключении основного питания микросхемы.

Интервал записи составляет 128 мс.

5.10.2 Спецификация регистров.

Программно доступные регистры блока таймера реального времени RTC расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение RTC Base = 0xFFF07000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже (Таблица 5.70).

Таблица 5.70 – Спецификация регистров RTC

Адрес	Название регистра	Значение после сброса	Доступ	Описание
Base + 0x00h	TIMEREG	06000000h	Чтение (всегда) и запись (только когда бит TR регистра RTCFR равен 1).	Регистр реального времени
Base + 0x04h	DATEREG	14000101h	Чтение (всегда) и запись (только когда бит DR регистра RTCFR равен 1).	Регистр даты
Base + 0x08h	ALARMREG	01000000h	Чтение (всегда) и запись (только когда бит AR регистра RTCFR равен 1).	Регистр будильника
Base + 0x0Ch	RTCCR	00000000h	Чтение (всегда) и запись (только когда бит CR регистра RTCFR равен 1).	Управляющий регистр
Base + 0x10h	RTCFR	0000000Fh	Чтение (всегда).	Регистр флагов

5.10.3 Описание регистров

5.10.3.1 Сброс регистров RTC.

Сброс регистров RTC может быть произведен двумя способами:

- аппаратный сброс с помощью вывода RTCPORn,
- программный сброс.

Регистры блока счётчиков не сбрасываются при аппаратном сбросе основной части микросхемы.

Для того чтобы программно сбросить устройство, нужно по адресу регистра RTCCR записать слово, в котором 6-й бит (BPRS) равен 0. После любого сброса часы не работают (бит CEN=0).

Сброс блока счётчиков может производиться в течение 1-2 интервалов записи.

Если устройство находится в состоянии сброса, бит RA регистра RTCFR имеет значение 1, если устройство не находится в состоянии сброса, бит RA регистра RTCFR имеет значение 0. Прежде чем записать новые значения в регистры устройства следует дождаться окончания сброса (признак RA в регистре RTCFR).

5.10.3.2 Общие правила записи в регистры устройства

Общее правило для всех регистров: запись в регистр игнорируется, если соответствующий этому регистру флаг готовности для записи в регистре RTCFR в состоянии 0. Однако, поскольку снятие флагов готовности регистров устройства происходит автоматически (аппаратно), то устройство может стать готовым для записи в любой момент времени.

Регистр недоступен для записи в период времени между обращением по адресу этого регистра по шине APB и подтверждением записи. В течение этого периода времени соответствующий этому регистру флаг готовности в регистре RTCFR имеет значение 0. Если непосредственно перед записью регистра не производился сброс устройства, то период неготовности регистра для записи может составлять от 1 до 2 интервалов записи.

В регистры устройства не могут быть записаны некорректные данные. Некорректными считаются данные, не соответствующие реальной дате или времени, например, 31 сентября 2007 года или 13:61:25, также некорректной считается запись значения 0 в поле ДЕНЬ НЕДЕЛИ регистров TIMEREG или ALARMREG. На корректность проверяются данные, записываемые в регистры TIMEREG, DATEREG и ALARMREG. При попытке записи некорректных данных в какой-либо из регистров запись в него не осуществляется, а также бит CEN регистра RTCCR сбрасывается в 0 (подробнее о бите CEN в разделе «Управляющий регистр»).

К зарезервированным битам можно обращаться по записи с любым значением, и это не будет расценено как попытка записи некорректных данных.

Корректные значения секунд: от 0 до 59.

Корректные значения минут: от 0 до 59.

Корректные значения часов: от 0 до 23.

Корректные значения кодов дней недели: от 1 до 7.

Корректные значения чисел месяца: от 1 до 28, 29, 30 или 31 в зависимости от текущего месяца и года.

Корректные значения кодов месяцев: от 1 до 12.

Корректные значения лет: от 0 до 99.

Попытка записи по адресу регистра RTCFR никак не повлияет на устройство.

Поля в регистрах устройства имеют разные режимы доступа, обозначаемые условно, как на рисунке ниже (см. Рисунок 5.25).

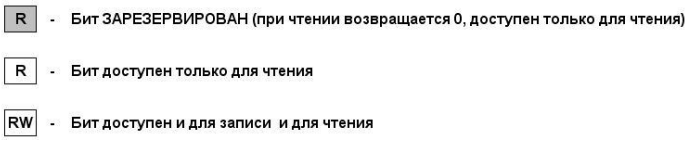


Рисунок 5.25 — Условные обозначения режимов доступа

5.10.3.3 Регистр реального времени TIMEREG

Регистр TIMEREG содержит поля текущих секунд, минут, часов реального времени и дня недели.

Регистр	TIMEREG																													
№№ бит	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
Поля	0	0	0	0	0	ДЕНЬ НЕДЕЛИ			0	0	0	ЧАСЫ					0	0	МИНУТЫ						0	0	СЕКУНДЫ					
Доступ	R					RW			R			RW					R		RW						R		RW				

Рисунок 5.26 — Регистр реального времени

Таблица 5.71 — Структура регистра TIMEREG

Название поля	№№ бит	Описание
СЕКУНДЫ	[5:0]	В этом поле хранится текущее значение секунд реального времени. Поле может принимать значения от 0 до 59 (00h – 3Bh).
МИНУТЫ	[13:8]	В этом поле хранится текущее значение минут реального времени. Поле может принимать значения от 0 до 59 (00h – 3Bh).
ЧАСЫ	[20:16]	В этом поле хранится текущее значение часов реального времени. Поле может принимать значения от 0 до 23 (00h – 17h).
ДЕНЬ НЕДЕЛИ	[26:24]	В этом поле закодирован текущий день недели. При записи текущей даты в регистр DATEREG следует также записывать правильное значение текущего дня недели в поле ДЕНЬ НЕДЕЛИ регистра TIMEREG. Дни недели кодируются следующим образом: 001 – Понедельник 010 – Вторник 011 – Среда 100 – Четверг 101 – Пятница 110 – Суббота 111 – Воскресенье

Значение регистра TIMEREG после сброса – 06000000h (соответствует времени 00:00:00, и дню недели – субботе).

5.10.3.4 Регистр даты DATEREG

Регистр даты содержит поля текущего числа, месяца, года и столетия.

Регистр	DATEREG																													
№№ бит	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
Поля	0	0	0	СТОЛЕТИЕ					0	ГОД							0	0	0	0	МЕСЯЦ				0	0	0	ЧИСЛО				
Доступ	R			RW					R	RW							R				RW				R			RW				

Рисунок 5.27 — Регистр даты

Таблица 5.72 — Структура регистра DATEREG

Название поля	№№ бит	Описание
ЧИСЛО	[4:0]	В этом поле хранится текущее число. Поле может принимать значения от 1 до 28, 29, 30 или 31, в зависимости от значений полей МЕСЯЦ и ГОД (01h – 1Fh).
МЕСЯЦ	[11:8]	В этом поле закодирован текущий месяц. Месяцы кодируются следующим образом: 0001 – Январь 0010 – Февраль 0011 – Март 0100 – Апрель 0101 – Май 0110 – Июнь 0111 – Июль 1000 – Август 1001 – Сентябрь 1010 – Октябрь 1011 – Ноябрь 1100 – Декабрь
ГОД	[22:16]	В этом поле хранится текущий год в интервале от 0 до 99 (00h – 63h).
СТОЛЕТИЕ	[31:24]	В этом поле хранится текущее столетие. Поле может принимать значения от 0 до 255.

Значение регистра DATEREG после сброса – 14000101h (соответствует дате 1 января 2000 года).

На этом примере показано, как закодировать дату 30 марта 2014 года (слово 140E031Eh):

Регистр	DATEREG																													
№№ бит	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
Значение	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0

Рисунок 5.28 — Пример кодирования даты

5.10.3.5 Регистр будильника ALARMREG

Регистр ALARMREG содержит поля секунд, минут, часов и дня недели срабатывания будильника.

Регистр	ALARMREG																													
№№ бит	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
Поля	0	0	0	0	0	ДЕНЬ НЕДЕЛИ			0	0	0	ЧАСЫ					0	0	МИНУТЫ						0	0	СЕКУНДЫ					
Доступ	R					RW			R			RW					R		RW						R			RW				

Рисунок 5.29 — Регистр будильника

Таблица 5.73 — Структура регистра ALARMREG

Название поля	№№ бит	Описание
СЕКУНДЫ	[5:0]	В этом поле хранится текущее значение секунды срабатывания будильника. Поле может принимать значения от 0 до 59 (00h — 3Bh).
МИНУТЫ	[13:8]	В этом поле хранится текущее значение минуты срабатывания будильника. Поле может принимать значения от 0 до 59 (00h — 3Bh).
ЧАСЫ	[20:16]	В этом поле хранится текущее значение часа срабатывания будильника. Поле может принимать значения от 0 до 23 (00h – 17h).
ДЕНЬ НЕДЕЛИ	[26:24]	В этом поле закодирован день недели срабатывания будильника. Дни недели кодируются таким же образом, как и в регистре TIMEREG: 001 – Понедельник 010 – Вторник 011 – Среда 100 – Четверг 101 – Пятница 110 – Суббота 111 – Воскресенье

Значение регистра ALARMREG после сброса – 01000000h (соответствует времени 00:00:00 и дню недели – понедельнику).

5.10.3.6 Управляющий регистр

Регистр	RTCCR																													
№№ бит	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
Поля	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CEN	RS	0	EIE	T	0	IAW	IAE
Доступ	R																								RW	RW	R	RW	RW	R	RW	RW

Рисунок 5.30 — Управляющий регистр

Таблица 5.74 — Структура управляющего регистра

Название поля	№№ бит	Описание
IAE (Interrupt on Alarm Enable)	<0>	IAE=1 – прерывание от RTC разрешено. IAE=0 – прерывание от RTC выключено.
IAW (Interrupt on Alarm by Weekday)	<1>	IAW=1 – при сравнении регистров TIMEREG и ALARMREG учитываются все поля (в том числе и поля ДЕНЬ НЕДЕЛИ). IAW=0 – при сравнении регистров TIMEREG и ALARMREG учитываются только поля ЧАСЫ, МИНУТЫ и СЕКУНДЫ (поля ДЕНЬ НЕДЕЛИ не сравниваются)
T (Test mode)	<3>	T=1 – устройство находится в тестовом режиме, в котором регистры часов переключаются с частотой 32768 Гц. T=0 – устройство находится в обычном режиме, в котором регистры часов переключаются с частотой 1 Гц (в соответствии с течением реального времени).
EIE (External Interrupt Enable)	<4>	EIE=1 – внешний вывод RTCINT при срабатывании будильника переключается в активное состояние (активное состояние – 1). EIE=0 – вывод RTCINT всегда остается в неактивном состоянии.
RS (Reset)	<6>	При чтении: RS=0 – устройство находится в состоянии сброса. RS=1 – устройство не находится в состоянии сброса. При записи: RS=1 – никак не влияет на устройство. RS=0 – дается команда на программный сброс устройства.
CEN (Count Enable)	<7>	CEN=0 – регистры устройства не переключаются. CEN=1 – регистры устройства переключаются в соответствии с течением реального времени (или с частотой 32768 Гц если T=1).

После сброса в битах IAE, IAW, T, EIE, CEN устанавливается значение 0.

При записи в регистр RTCCR слова, в котором 6-й бит (RS) имеет значение 0, происходит сброс устройства, затем бит RS принимает значение 1.

5.10.3.7 Регистр флагов

Регистр	RTCFR																													
№№ бит	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
Поля	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	AW	DW	TW	RA	CR	AR	DR	TR
Доступ	R																								R	R	R	R	R	R	R	R

Рисунок 5.31 — Регистр флагов

Таблица 5.75 — Структура регистра флагов

Название поля	№№ бит	Описание
TR (TIMEREG Ready)	<0>	TR=1 – регистр TIMEREG готов для записи. TR=0 – регистр TIMEREG не готов для записи.
DR (DATEREG Ready)	<1>	DR=1 – регистр DATEREG готов для записи. DR=0 – регистр DATEREG не готов для записи.
AR (ALARMREG Ready)	<2>	AR=1 – регистр ALARMREG готов для записи. AR=0 – регистр ALARMREG не готов для записи.
CR (Control register Ready)	<3>	CR=1 – регистр RTCCR готов для записи. CR=0 – регистр RTCCR не готов для записи.
RA (Reset Active)	<4>	RA=1 – устройство находится в состоянии сброса. RA=0 – устройство не находится в состоянии сброса
TW (Time Wrong)	<5>	TW=1 – последняя попытка записи в регистр TIMEREG не прошла из-за того, что данные были некорректными. TW=0 – последняя попытка записи в TIMEREG содержала корректные данные.
DW (Date Wrong)	<6>	DW=1 – последняя попытка записи в регистр DATEREG не прошла из-за того, что данные были некорректными. DW=0 – последняя попытка записи в DATEREG содержала корректные данные.
AW (Alarm time Wrong)	<7>	AW=1 – последняя попытка записи в регистр ALARMREG не прошла из-за того, что данные были некорректными. AW=0 – последняя попытка записи в ALARMREG содержала корректные данные.

После аппаратного сброса устройства регистр RTCFR имеет значение 0000000Fh.

Программный сброс устанавливает в регистр RTCFR значение 0000000Fh.

5.10.4 Прерывания

Причиной прерывания от устройства RTC может стать только срабатывание будильника. Устройство имеет два сигнала прерывания: внутрисистемный и внешний (RTCINT).

Внутрисистемное прерывание возникает при соблюдении условий (1), (2) и (3):

1) Бит IAE=1.

Если бит IAW=1, то

2) Регистры TIMEREG и ALARMREG полностью равны.

Если бит IAW=0, то

2) Поля СЕКУНДЫ, МИНУТЫ и ЧАСЫ регистров TIMEREG и ALARMREG равны.

3) Регистр TIMEREG переключается в состояние, соответствующее следующей секунде реального времени.

Сигнал внутрисистемного прерывания снимается при любом обращении по записи к регистру RTCCR или к регистру ALARMREG.

Сигнал внешнего прерывания RTCINT устанавливается в активное состояние при соблюдении условий (1), (2) и (3):

1) Бит EIE=1.

Если бит IAW=1, то

2) Регистры TIMEREG и ALARMREG полностью равны.

Если бит IAW=0, то

2) Поля СЕКУНДЫ, МИНУТЫ и ЧАСЫ регистров TIMEREG и ALARMREG равны.

3) Регистр TIMEREG переключается в состояние, соответствующее следующей секунде реального времени.

Сигнал внешнего прерывания RTCINT держится в активном состоянии в течение 31 мкс и не может быть снят программно.

5.11 Системный контроллер ARMSC

Системный контроллер ARMSC выполняет вспомогательные функции в СБИС СНП-ВП. Данный контроллер содержит набор программно доступных со стороны ядра ARM периферийных регистров, которые выполняют следующие функции:

- Регистры состояния входов – показывают состояние входов МС, управляющих источником начальной загрузки, а также состоянием тестовых входов
- Управление сигналами SPI_CSx
- Регистры конфигурации блока USB
- Регистры управления АЦП – позволяют управлять режимом работы встроенного АЦП
- Регистры управления прерываниями NMC части
- Регистр приоритета ядер NMU

5.11.1 Устройство системного контроллера ARMSC

В таблице ниже приведен список внешних выводов, относящихся к системному контроллеру ARMSC.

Таблица 5.76– Выводы микросхемы, относящиеся контроллеру ARMSC

Вывод	Тип буфера	Примечание
BOOTM[2:0]	in	входы управления источником начальной загрузки
TSTMODE	in	вход управления тестовым режимом
TST_I[2:0]	in	3 бита тестовых входов
SPI_CS7 … SPI_CS0	out	сигналы выбора ведомого SPI устройства

На рисунке (см. Рисунок 5.32) приведена схема включения системного контроллера ARMSC.

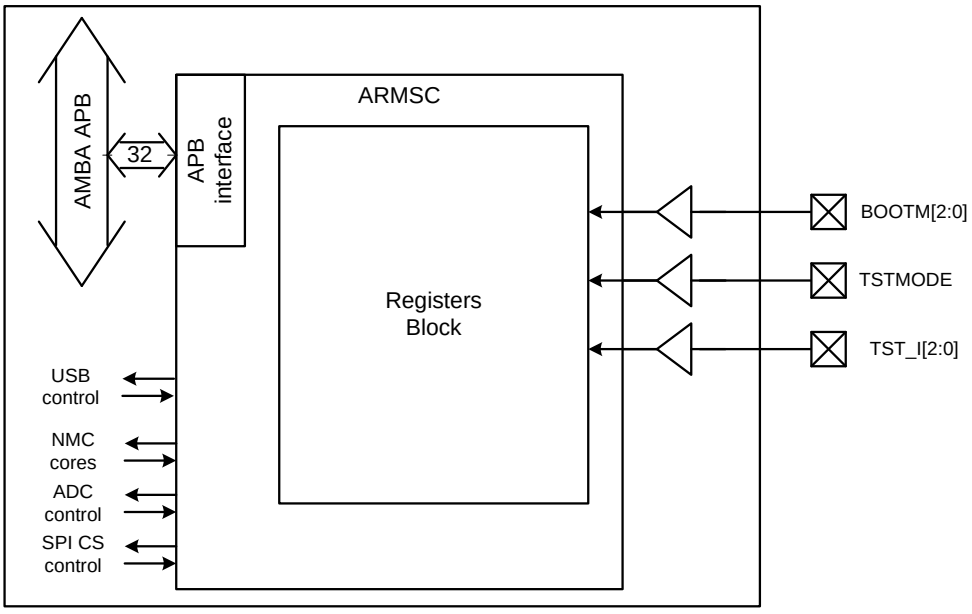


Рисунок 5.32 – Схема системного контроллера ARMSC

Подробное описание работы с регистрами смотри в описании соответствующих блоков СБИС СНП-ВП.

5.11.2 Регистровая модель контроллера ARMSC

Программно доступные регистры контроллера ARMSC расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение ARMSC Base = 0xFFF02000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.77 – Спецификация регистров контроллера ARMSC

Адрес	Тип доступа	Разрядность	Начальное значение	Имя регистра	Описание
ARMSC Base + 0x00	ЧТ	7	значения на входах	BOOTM	Регистр состояния внешних входов
ARMSC Base + 0x04	ЧТ/ЗП	3	3'b0	SPICSCTL	Регистр выбора ведомого SPI устройства
ARMSC Base + 0x08	ЧТ/ЗП	32	32'b0	EDCLJMP	Регистр адреса загрузки EDCL
ARMSC Base + 0x0C	ЧТ	4	4'b0	USBSTA	Регистр состояния USB
ARMSC Base + 0x10	ЧТ/ЗП	1	0	USBCTL	Регистр управления USB
ARMSC Base + 0x14	ЧТ/ЗП	8	8'h11	ADCCTL	Регистр управления АЦП
ARMSC Base + 0x18	ЧТ/ЗП	2	2'b11	PERSLP	Регистр управления потреблением памяти периферийных устройств
ARMSC Base + 0x1C	-	-	-	-	зарезервировано
ARMSC Base + 0x20	ЧТ	13	13'b0	NMUINTSTA	Регистр состояния прерываний от NMU части СБИС
ARMSC Base + 0x24	ЗП1	13	13'b0	NMUINTCLR	Регистр сброса прерываний от NMU части СБИС
ARMSC Base + 0x28	ЗП1	6	6'b0	NMUINTREQ	Регистр генерации прерывания к NMU части
ARMSC Base + 0x2C	ЧТ/ЗП	2	2'b0	NMUCTRL	Регистр приоритета ядер NMC
ARMSC Base + 0x30 to 0xFFC	-	-	-	-	зарезервировано

Далее приведено подробное описание каждого регистра.

5.11.2.1 Регистры состояния внешних входов BOOTM

Регистр BOOTM используется для контроля значений на следующих внешних входах СБИС: TST_I[2:0], BOOTM[2:0] и TSTMODE. Программа начального загрузчика использует часть этих регистров для определения источника начальной загрузки и режима работы. После окончания работы начального загрузчика входы TST_I и BOOTM могут использоваться как входы общего назначения. Вход TSTMODE должен быть равен нулю всегда, т.к. изменение этого сигнала на единицу приведет к сбою в работе СБИС СНП-ВП.

Таблица 5.78 – Формат регистра BOOTM

	Название	Тип	Выполняемая функция
[6:4]	TST_I[2:0]	ЧТ	Значение на входах TST_I[2:0]
[3]	TSTMODE	ЧТ	Значение на входе TSTMODE
[2:0]	BOOTM[2:1]	ЧТ	Значение на входах BOOTM[2:0

5.11.2.2 Регистр выбора ведомого SPI устройства SPICSCTL

Регистр SPICSCTL используется для выбора активного ведомого SPI устройства (управление выводами SPI_CS7…SPI_CS0 микросхемы).

Таблица 5.79 – Формат регистра SPICSCTLR

	Название	Тип	Выполняемая функция
[2:0]	SPICSCTL	ЧТ/ЗП	Выбор активного ведомого устройства SPI: 000 – SPI_CS0 001 – SPI_CS1 … 111 – SPI_CS7

5.11.2.3 Регистр адреса загрузки EDCL EDCLJMP

Регистр EDCLJMP используется в варианте загрузке по EDCL. Начальный загрузчик отслеживает содержимое этого регистра. В момент изменения адреса в регистре на адрес отличный от нулевого, происходит переход по этому адресу и окончание работы начального загрузчика.

Таблица 5.80 - Формат регистра EDCLJMP

	Название	Тип	Выполняемая функция
[31:0]	EDCLJMPADD	ЧТ/ЗП	Адрес перехода при загрузке по EDCL

5.11.2.4 Регистр состояния USB USBSTA

Регистр USBSTA используется для отслеживания состояния блока USB. Формат регистра приведен в таблице ниже.

Таблица 5.81 - Формат регистра USBSTA

	Название	Тип	Выполняемая функция
[3]	SPEED	ЧТ	0 – USB FullSpeed mode 1 – USB HighSpeed mode
[2]	SETUP	ЧТ	Сигнал состояния блока USB (1 – setup state)
[1]	BUSRESET	ЧТ	Сигнал состояния блока USB (1 – bus reset state)
[0]	SUSPEND	ЧТ	Сигнал состояния блока USB (1 – suspend state)

5.11.2.5 Регистр управления USB USBCTL

Регистр USBCTL используется для управления потребления блока USB. Формат регистра приведен в таблице ниже.

Таблица 5.82 - Формат регистра USBCTL

	Название	Тип	Выполняемая функция
[0]	PD	ЧТ/ЗП	USB PHY Analog Power Down

5.11.2.6 Регистр управления потреблением памяти периферийных устройств PERSLP

Регистр PERSLP используется для управления режимом питания макросов памяти периферийных устройств GRETH и CAN(в случае если блоки не используются - можно включить

режим пониженного потреления макросами памяти). Формат регистра приведен в таблице ниже.

Таблица 5.83 - Формат регистра PERSLP

	Название	Тип	Выполняемая функция
[1]	ETH_EN	ЧТ/ ЗП	0 – блок выключен (режим пониженного потребления) 1 – блок включен После записи 0 (включения режима пониженного потребления), запись 1 возможна не ранее чем через 16 тактов
[0]	CAN_EN	ЧТ/ ЗП	0 – блок выключен (режим пониженного потребления) 1 – блок включен После записи 0 (включения режима пониженного потребления), запись 1 возможна не ранее чем через 16 тактов

5.11.2.7 Регистр управления АЦП ADCCTRL

Биты регистра ADCCTRL используются для управления потреблением и работой встроенных АЦП. В таблице ниже приведен формат регистра.

Таблица 5.84 – Формат регистра ADCCTRL

Биты	Название	Тип	Выполняемая функция
[7]	ADC23_GAIN	ЧТ/ ЗП	Управление усилением входного аналогового тракта для АЦП3 и АЦП2 0 - усиления не происходит (амплитуда входного сигнала 0-1,2В) 1 - усиление в два раза (амплитуда входного сигнала 0-0,6В)
[6]	CH3_SB	ЧТ/ ЗП	Управление режимом потребления АЦП3 0 – АЦП включен 1 – АЦП в режиме ожидания (standby) Бит не оказывает влияния если ADC23_EN=0.
[5]	CH2_SB	ЧТ/ ЗП	Управление режимом потребления АЦП2 0 – АЦП включен 1 – АЦП в режиме ожидания (standby) Бит не оказывает влияния если ADC23_EN=0.
[4]	ADC23_EN	ЧТ/ ЗП	Управление питанием АЦП3 и АЦП2 1 – АЦП включены 0 – АЦП выключены (powerdown)
[3]	ADC01_GAIN	ЧТ/ ЗП	Управление усилением входного аналогового тракта для АЦП1 и АЦП0 0 - усиления не происходит (амплитуда входного сигнала 0-1,2В) 1 - усиление в два раза (амплитуда входного сигнала 0-0,6В)
[2]	CH1_SB	ЧТ/ ЗП	Управление режимом потребления АЦП1 0 – АЦП включен 1 – АЦП в режиме ожидания (standby) Бит не оказывает влияния если ADC01_EN=0.
[1]	CH0_SB	ЧТ/ ЗП	Управление режимом потребления АЦП0 0 – АЦП включен 1 – АЦП в режиме ожидания (standby) Бит не оказывает влияния если ADC01_EN=0.
[0]	ADC01_EN	ЧТ/ ЗП	Управление питанием АЦП1 и АЦП0 1 – АЦП включены 0 – АЦП выключены (powerdown)

5.11.2.8 Регистр состояния прерываний от NMU части СБИС. NMUINTSTA

Регистр NMUINTSTA предназначен для определения состояния прерываний от NMU части СБИС. В таблице ниже приведен формат регистра.

Таблица 5.85 – Формат регистра NMUINTSTA

Биты	Название	Тип	Выполняемая функция
[12]	CORINT	ЧТ	Состояние прерывания CORINT 0 – прерывание не активно 1 – прерывание активно
[11]	DDCCHINT	ЧТ	Состояние прерывания DDCCHINT 0 – прерывание не активно 1 – прерывание активно
[10]	DDCPRG	ЧТ	Состояние прерывания DDCPRG 0 – прерывание не активно 1 – прерывание активно
[9]	NDMADR	ЧТ	Состояние прерывания NDMADR 0 – прерывание не активно 1 – прерывание активно
[8]	NDMAMC1	ЧТ	Состояние прерывания NDMAMC1 0 – прерывание не активно 1 – прерывание активно
[7]	NDMAMC0	ЧТ	Состояние прерывания NDMAMC0 0 – прерывание не активно 1 – прерывание активно
[6]	INT_STRB_ITU	ЧТ	Состояние прерывания INT_STRB_ITU 0 – прерывание не активно 1 – прерывание активно
[5]	INT_AUX_ITU	ЧТ	Состояние прерывания от интервального таймера INT_AUX_ITU 0 – прерывание не активно 1 – прерывание активно
[4]	INT_MS_ITU	ЧТ	Состояние прерывания от интервального таймера INT_MS_ITU 0 – прерывание не активно 1 – прерывание активно
[3]	NMC1HP	ЧТ	Состояние прерывания NMC1HP 0 – прерывание не активно 1 – прерывание активно
[2]	NMC0HP	ЧТ	Состояние прерывания NMC0HP 0 – прерывание не активно 1 – прерывание активно
[1]	NMC1LP	ЧТ	Состояние прерывания NMC1LP 0 – прерывание не активно 1 – прерывание активно
[0]	NMC0LP	ЧТ	Состояние прерывания NMC0LP 0 – прерывание не активно 1 – прерывание активно

5.11.2.9 Регистр сброса прерываний от NMU части СБИС. NMUINTCLR

Регистр NMUINTCLR предназначен для сброса от NMU части. В таблице ниже приведен формат регистра.

Таблица 5.86 – Формат регистра NMUINTCLR

Биты	Название	Тип	Выполняемая функция
[12]	CORINTCLR	ЗП1	Сброс прерывания CORINT. Запись 1 сбрасывает прерывание.
[11]	DDCCHINTCLR	ЗП1	Сброс прерывания DDCCHINT. Запись 1 сбрасывает прерывание.
[10]	DDCPRGCLR	ЗП1	Сброс прерывания DDCPRG. Запись 1 сбрасывает прерывание.
[9]	NDMADRCLR	ЗП1	Сброс прерывания NDMADR. Запись 1 сбрасывает прерывание.
[8]	NDMAMC1CLR	ЗП1	Сброс прерывания NDMAMC1. Запись 1 сбрасывает прерывание.
[7]	NDMAMC0CLR	ЗП1	Сброс прерывания NDMAMC0. Запись 1 сбрасывает прерывание.
[6]	INT_STRB_ITUCLR	ЗП1	Сброс прерывания INT_STRB_ITU. Запись 1 сбрасывает прерывание.
[5]	INT_AUX_ITUCLR	ЗП1	Сброс прерывания INT_AUX_ITU. Запись 1 сбрасывает прерывание.
[4]	INT_MS_ITUCLR	ЗП1	Сброс прерыванияINT_MS_ITU. Запись 1 сбрасывает прерывание.
[3]	NMC1HPCLR	ЗП1	Сброс прерывания NMC1HP. Запись 1 сбрасывает прерывание.
[2]	NMC0HPCLR	ЗП1	Сброс прерывания NMC0HP. Запись 1 сбрасывает прерывание.
[1]	NMC1LPCLR	ЗП1	Сброс прерывания NMC1LP. Запись 1 сбрасывает прерывание.
[0]	NMC0LPCLR	ЗП1	Сброс прерывания NMC0LP. Запись 1 сбрасывает прерывание.

5.11.2.10 Регистр генерации прерывания к NMU части. NMUINTREQ

Регистр NMUINTREQ предназначен для генерации прерывания от ARMU части к NMU части. Запись 1 в соответствующее поле приводит к генерации прерывания. Прерывание активируется на 1 такт синхросигнала периферийной шины APB, после чего автоматически деактивируется. В таблице ниже приведен формат регистра.

Таблица 5.87 – Формат регистра NMUINTREQ

Биты	Название	Тип	Выполняемая функция
[5]	NMI_NMC1	ЗП1	Немаскируемое прерывание к NMC1. Запись 1 генерирует прерывание.
[4]	NMI_NMC0	ЗП1	Немаскируемое прерывание к NMC0. Запись 1 генерирует прерывание.
[3]	HPINT_NMC1	ЗП1	Высокоприоритетное прерывание к NMC1. Запись 1 генерирует прерывание.
[2]	HPINT_NMC0	ЗП1	Высокоприоритетное прерывание к NMC0. Запись 1 генерирует прерывание.
[1]	LPINT_NMC1	ЗП1	Низкоприоритетное прерывание к NMC1. Запись 1 генерирует прерывание.
[0]	LPINT_NMC0	ЗП1	Низкоприоритетное прерывание к NMC0. Запись 1 генерирует прерывание.

5.11.2.11 Регистр приоритета ядер NMC NMUCTRL

Младший бит регистра NMUCTRL задает приоритет запросов ядер NMC0 и NMC1 к системе общей памяти SMU (см. п. 4.7). В таблице ниже приведен формат регистра.

Таблица 5.88 – Формат регистра NMUCTRL

Биты	Название	Тип	Выполняемая функция
[0]	NMC0SMU	ЧТ/ЗП	Состояние приоритетов доступа NMU1 и NMU2 к SMU памяти. 0 – запросы NMU2 приоритетней 1 – запросы NMU1 приоритетней

5.11.3 Прерывания

Контроллер ARMSC выполняет вспомогательную функцию для обработки прерываний от NMU части СБИС. Прерывания от NMU части СБИС не могут быть подключены напрямую к системному контроллеру прерываний VIC, поэтому контроллер ARMSC выполняет функцию буферизации прерываний от NMU части, а также управляет сбросом этих прерываний.

Всего к системному контроллеру прерываний VIC подключено 13 буферизованных в ARMSC прерываний от NMC части – NMC0HP, NMC1HP, DDCCHINT, INT_MS_ITU, CORINT, NDMADR, NDMAMC0, NDMAMC1, INT_AUX_ITU, INT_STRB_ITU, NMC0LP, NMC1LP и DDCPRG.

5.12 Блок управления питанием PWRC

Блок управления питанием позволяет отключать питание для некоторых блоков микросхемы, которые называются домены питания, позволяя этим экономить потребляемую мощность. СБИС СНП-ВП содержит следующие домены питания:

- PD_NMU0 - домен питания содержащий процессорную систему NMU0;
- PD_NMU1 - домен питания содержащий процессорную систему NMU1;
- PD_ARMU0 - домен питания содержащий процессорную систему ARMU;
- PD_SMU - домен питания содержащий систему общей памяти (SMU), коммуникационные порты (CP0, CP1), контроллер ПДП (DMAC), блок преобразования частоты (DDC) без КИХ фильтра, блок аппаратных корреляторов (CORB);
- PD_FILTR - домен питания содержащий КИХ-фильтр (часть блока DDC);
- PD_RTC - домен питания содержащий таймер реального времени (RTC);

Так же блок PWRC формирует сигналы запуска процессорных систем NMU0 и NMU1. Работа возможна в двух режимах: автоматическом и программном. В автоматическом режиме последовательность для включения и выключения происходит автоматически. В программном режиме программист путем записи в конфигурационные регистры формирует управляющие сигналы (сигнал включения питания, сигнал управления изолирующими ячейками и т.д.). В этом случае программист сам формирует необходимую последовательность управляющих сигналов для включения/выключения питания. Программирование блока осуществляется процессорной системой ARMU.

В таблице ниже приведены все допустимые режимы работы

Таблица 5.89 - Режимы работы СБИС СНП-ВП

№	домены питания						примечание
	PD_ARMU	PD_RTC	PD_NMU1	PD_NMU0	PD_SMU	PD_FILTR	
1	+	+	+	+	+	+	данный режим устанавливается после общего системного сброса и включения всех источников питания
2	+	+	+	+	+		отключение домена осуществляется силовыми ключами расположенными на микросхеме
3	+	+	+		+	+	отключение домена осуществляется силовыми ключами расположенными на микросхеме
4	+	+	+		+		отключение домена осуществляется силовыми ключами расположенными на микросхеме
5	+	+		+	+	+	отключение домена осуществляется силовыми ключами расположенными на микросхеме
6	+	+		+	+		отключение домена осуществляется силовыми ключами расположенными на микросхеме
7	+	+			+	+	отключение домена осуществляется силовыми ключами расположенными на микросхеме
8	+	+			+		отключение домена осуществляется силовыми ключами расположенными на микросхеме
9	+	+					отключение домена осуществляется силовыми ключами расположенными на микросхеме
10		+					в данном режиме выключение источника питания VDDI осуществляется внешним (по отношению к микросхеме) устройством

5.12.1 Программный механизм включения и выключения питания

Программное управление устанавливается после системного сброса **PWRC**. Если в процессе работы менялся бит PM регистра **CFGR**, то необходимо в состоянии автомата 0000 сбросить бит PM регистра CFGR.

Путем программной записи в регистры **PWR**, **STR**, **CRGR** производится управление изоляционными ячейками, включением/отключением питания, запуском и остановкой тактового сигнала, подачей сигнала сброса для доменов. При программном управлении следует учитывать следующие особенности, нарушение которых может привести к сбою в работе:

- перед выключением питания должны быть включены изоляционные ячейки;
- если выключалось питание, то изоляционные ячейки должны быть отключены только после подачи сброса на этот домен;
- во время подачи сброса обязательно должен работать тактовый сигнал для этого домена;
- для того чтобы не было зависания при обращении к NMU при выключенном тактовом сигнале NMU, необходимо включить изоляционные ячейки NMU. При этом при чтении будут читаться фиктивные данные, а команды записи будут уходить "в никуда";
- при остановке тактового сигнала домена PD_SMU необходимо до этого остановить тактовые сигналы для доменов PD_NMU0, PD_NMU1, PD_FILTR;
- при выключении питания домена PD_SMU необходимо до этого выключить питание для доменов PD_NMU0, PD_NMU1, PD_FILTR;

5.12.2 Автоматический механизм включения и выключения питания

В автоматическом режиме установка и сброс сигналов, управляющих изоляционными ячейками и отключением питания, осуществляется автоматически в необходимой последовательности в соответствии с настройками регистров. Событие, которое запускает последовательность действий на включение или выключение определяется полями ON_CNTRL и OFF_CNTRL регистра ***CFGR***.

При автоматическом выключении питания/тактового сигнала происходит следующая последовательность действий:

- формирование запроса на остановку тактового сигнала/выключение питания (установка бита STOP_IAG регистра STR);
- получение подтверждения от блока о готовности остановки (ожидание STOP_RDY=1 регистра STR);
- включение изоляционных ячеек путем установки бита ISO_ON регистра PWR;
- выключение питания (если установлен бит T регистра CFGR) путем установки бита PWR_OFF регистра PWR;

При включении питания происходит автоматически следующая последовательность действий:

- включение питания путем сброса бита PWR_OFF регистра **PWR**;
- ожидание интервала времени, заданного регистром **TPWRDR**;
- включение тактового сигнала;
- формирование сигнала сброса на включаемый домен;
- ожидание интервала времени, заданного регистром **TRSTR**;
- снятие сигнала сброса и отключение изоляционных ячеек;
- запрос на загрузку программы через коммуникационный порт и запуск адресного генератора (только для NMU при установленном бите LOAD_TYPE регистра **LDR**);

При включении тактового сигнала происходит автоматически следующая последовательность действий:

- включение тактового сигнал путем выдачи сигнала CACTIVE в CRG и ожидание подтверждения от CRG сигналом CSYSREQ=1;
- отключение изоляционных ячеек;
- запуск адресного генератора (только для NMU);

Последовательностью действий управляет автомат, приведенный на рисунке ниже (см. Рисунок 5.33). Сигналы над линией перехода означают, по какому событию происходит переход. Если событие отсутствует, то переход происходит в следующем такте. Под стрелкой обозначены сигналы формируемые автоматом (в сером прямоугольнике).

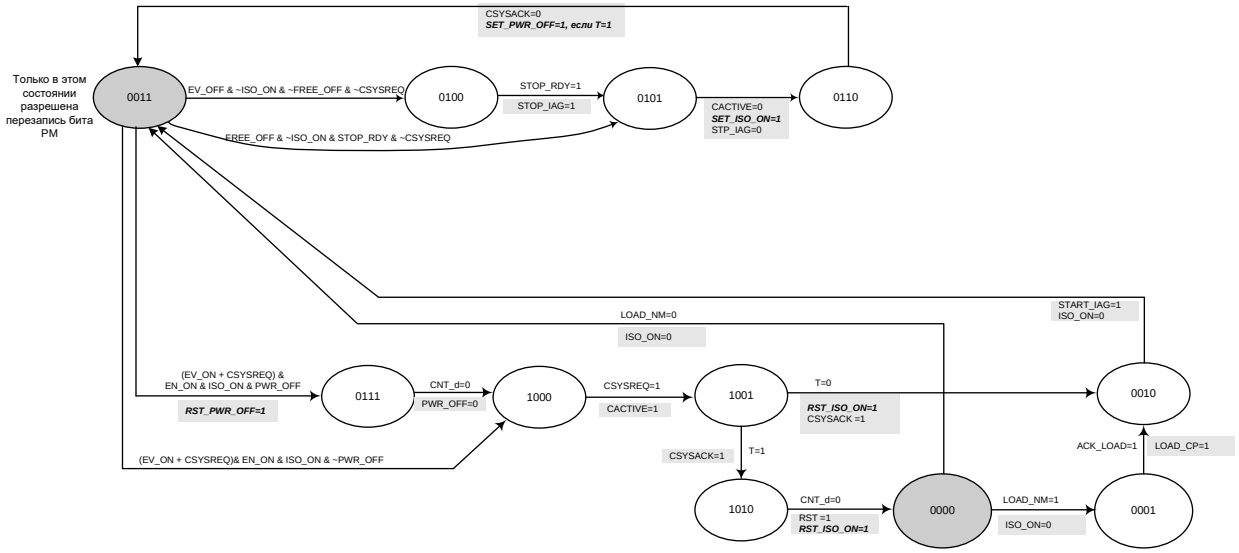


Рисунок 5.33 – Граф перехода автомата включения/выключения

Ниже в таблице описаны состояния автомата (Таблица 5.90).

Таблица 5.90 – Описание состояний автомата при автоматическом включении/выключении

остояние автомата	выдаваемые сигналы в данном состоянии	ожидание события для перехода
0000	состояние после системного сброса блока управления питанием.	
0001	формирование запроса для загрузки процессорной системы через коммуникационный порт (LOPAD_CP=1)	ожидание подтверждения конца загрузки ACK_LOAD=1
0010	Выдача сигнала на запуск адресного генератора (START_IAG=1)	
0011	В данном состоянии разрешено менять управление с автоматического на программное и обратно.	ожидание события для выключения при включенном блоке, ожидание события для включения при выключенном блоке
0100	Выдача запроса на останов блока STP_IAG=1	ожидание подтверждения остановки блока STOP_RDY=1
0101	Выдача сигнала CACTIVE =0	
0110	Выдача сигнала CSYSACK =0, если необходимо снять питание, то формирование сигнала выключения питания (SET_PWR_OFF=1), который будет активен в следующем такте	
0111	включения питания блока, запуск счетчика времени включения питания	Ожидание конца интервала включения питания (CNT_d=0)
1000	Выдача сигнала CACTIVE =1 в CRG	Ожидание CSYSREQ=1
1001	Выдача сигнала CSYSACK =1 в CRG, если не отключалось питание, то формирование сигнала отключения изоляционных ячеек (RST_ISO_ON =1), который будет активен в следующем такте	
1010	запуск счетчика времени системного сброса, выдача сброса для включаемого (блока RST=1), формирование сигнала отключения изоляционных ячеек (RST_ISO_ON =1), который будет активен при переходе в состояние 0000	Ожидание конца интервала системного сброса (CNT_d=0)

5.12.3 Программно доступные регистры блока PWRC

Ниже в таблице приведены регистры блока управления питанием (Таблица 5.91). После таблицы дано описание каждого регистра. Поля регистров обозначенные коричневым цветом устанавливаются только аппаратно и доступны только на чтение. Пометка ***R*** обозначает, что поле (регистр) доступен для чтения. Пометка ***W*** обозначает, что поле (регистр) доступен для записи.

Таблица 5.91 - Регистры блока PWRC

	название регистра	адрес регистра в адресном пространстве ARMU	
	для домена NMU0		
1	LDR	FFF0_3000	R/W
2	PWR	FFF0_3004	R/W
3	PWR_set	FFF0_3008	W
4	PWR_reset	FFF0_300C	W
5	CFGR	FFF0_3010	R/W
6	CRGR	FFF0_3014	R/W
7	CRGRset	FFF0_3018	W
8	CRGR_reset	FFF0_301C	W
9	STR	FFF0_3020	R/W
10	STMR	FFF0_3024	R/W
11	TRSTR	FFF0_3028	R/W
12	TPWRDR	FFF0_302C	R/W
	для домена NMU1		
1	LDR	FFF0_3040	R/W
2	PWR	FFF0_3044	R/W
3	PWR_set	FFF0_3048	W
4	PWR_reset	FFF0_304C	W
5	CFGR	FFF0_3050	R/W
6	CRGR	FFF0_3054	R/W
7	CRGRset	FFF0_3058	W
8	CRGR_reset	FFF0_305C	W
9	STR	FFF0_3060	R/W
10	STMR	FFF0_3064	R/W
11	TRSTR	FFF0_3068	R/W
12	TPWRDR	FFF0_306C	R/W
	для домена SMU		
1	LDR	FFF0_3080	R/W
2	PWR	FFF0_3084	R/W
3	PWR_set	FFF0_3088	W
4	PWR_reset	FFF0_308C	W
5	CFGR	FFF0_3090	R/W
6	CRGR	FFF0_3094	R/W
7	CRGRset	FFF0_3098	W
8	CRGR_reset	FFF0_309C	W
9	STR	FFF0_30A0	R/W
10	STMR	FFF0_30A4	R/W
11	TRSTR	FFF0_30A8	R/W
12	TPWRDR	FFF0_30AC	R/W
	для домена FILTR		
1	LDR	FFF0_30C0	R/W
2	PWR	FFF0_30C4	R/W
3	PWR_set	FFF0_30C8	W
4	PWR_reset	FFF0_30CC	W
5	CFGR	FFF0_30D0	R/W
6	CRGR	FFF0_30D4	R/W
7	CRGRset	FFF0_30D8	W
8	CRGR_reset	FFF0_30DC	W
9	STR	FFF0_30E0	R/W
10	STMR	FFF0_30E4	R/W
11	TRSTR	FFF0_30E8	R/W
12	TPWRDR	FFF0_30EC	R/W

5.12.3.1 LDR (Load Register)

Формат регистра приведен на рисунке ниже.

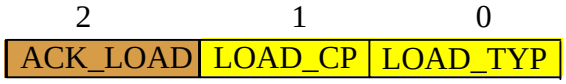


Рисунок 5.34 – Формат регистра LDR

LOAD_TYP - загрузка процессорной системы NMU0, NMU1 ***(R/W).*** Определяет как будет проходить загрузка процессорной системы NMU:

0 - загрузка процессорной системы NMU осуществляется процессорной системой ARMU;

1 - загрузка процессорной системы NMU через коммуникационный порт;

Данный бит инициализируется внешним выводом BOOTM_NM0 и BOOTM_NM1 при системном сбросе. При последующей работе данный бит может быть изменен программно.

LOAD_CP - запрос на загрузку процессора NMU через коммуникационный порт ***(R/W).*** Сброс данного бита происходит автоматически после окончания загрузки.

0 - нет запроса;

1 - запрос на загрузку через коммуникационный порт;

ACK_LOAD - подтверждение об окончании загрузки по коммуникационному порту ***(R)***:

0 - загрузка не выполнена.

1 - загрузка выполнена.

Программирование данного регистра для отключаемых доменов SMU и FILTR не оказывает никакого действия.

Для доменов PD_SMU и PD_FILTR данные регистры не работу не влияют

5.12.3.2 PWR (Power Control Register)

Формат регистра приведен на рисунке ниже.

Рисунок 5.35 – формат регистра PWRCR

Регистр доступен по чтению и записи. Данный регистр управляет непосредственно изолирующими ячейками доменов и отключением питания доменов.

ISO_ON - сигнал управления изолирующими ячейками ***(R/W)***

0- изолирующая ячейка отключена;

1- изолирующая ячейка включена.

PWR_OFF - сигнал управления питанием ***(R/W)***.

0- питание включено;

1- питание отключено.

После системного сброса изолирующие ячейки отключены, а питание для всех доменов включено.

5.12.3.3 CFGR (Config Register)

Формат регистра приведен на рисунке ниже.

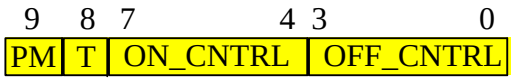


Рисунок 5.36 - Формат регистра STNMU0, STNMU1, STSMU

Регистр управления автоматическим режимом включения/выключения доменов.

PM - управляет разрешением автоматического включения/выключения доменов ***(R/W):***

1- возможно автоматическое включение и выключение доменов;

0- автоматическое включение и выключение доменов не возможно (только программное управление). Является значение по умолчанию.

OFF_CNTRL[3:0] - событие для автоматического выключения питания/тактового сигнала ***(R/W)***:

00xx - не выключать.

01xx - выключить.

10xx - выключить по готовности (событие FREE_OFF). Означает, что произойдет выключение у PD_NMU как только произойдет остановка адресного генератора, для PD_SMU как только закончит все передачи DMAC;

1100 - выключить по прерыванию от RTC;

1101 - выключить по сигналу основной метки (мс) от интервального таймера;

1110 - выключить по сигналу дополнительной метки от интервального таймера;

1111 - выключить по сигналу программируемого генератора от интервального таймера.

ON_CNTRL [3:0] - событие для включения питания/тактового сигнала***(R/W)***:

00xx - не включать

01xx - включить

1100 - включить по прерыванию от RTC

1101 - включить по сигналу основной метки (мс) от интервального таймера

1110 - включить по сигналу дополнительной метки от интервального таймера

1111 - включить по сигналу программируемого генератора от интервального таймера

T - показывает тип отключения ***(R/W)***:

0 - производится отключение тактового сигнала;

1 - производится отключение тактового сигнала и питания.

5.12.3.4 STMR

Рисунок 5.37 - Формат регистра STMR

STATE - состояние конечного автомата R/W. Позволяет отслеживать состояние автомата. Запись возможна только для тестирования.

5.12.3.5 CRGR (CRG Control Register)

Формат регистра приведен на рисунке ниже.

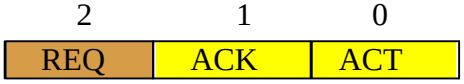


Рисунок 5.38 - Формат регистра CRGCR

Регистр отвечает за обмен управляющими сигналами с блоком CRG11. Блок PWRC соединен с блоком CRG11 по интерфейсу AMBA AXI Low power Interface для отключения тактового сигнала.

ACT - CACTIVE выход в блок CRG11 ***(R/W)***

ACK - CSYSACK выход в блок CRG11 ***(R/W)***

REQ - CSYSREQ вход от CRG11 ***(R)***

5.12.3.6 STR (Start Register)

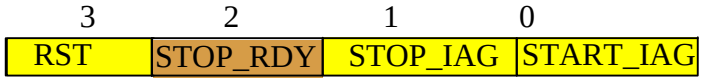


Рисунок 5.39 - формат регистра STR

START_IAG - формирует сигнал запуска адресного генератора команд для NMU0, NMU1 ***(R/W)***. Для доменов SMU и FILTR данное поле значения не имеет. Запись единицы формирует сигнал запуска адресного генератора. Сброс происходит автоматически после запуска.

STOP_IAG - сигнал запроса на остановку тактового сигнала ***(R/W):***

0 - не готовиться к остановке тактового сигнала;

1 - подготовиться к остановке тактового сигнала;

По данному сигналу в процессорной системе NMU формируется прерывание. Подпрограмма обработки прерывания должна закончить все обмены и остановить адресный генератор. Домен SMU должен закончить обмен с использованием каналов DMA и коммуникационных портов. Для домена FILTR значение данного поля не имеет.

STOP_RDY - показывает готовность домена к остановке тактового сигнала ***(R):***

0 - домен готов к остановке тактового сигнала;

1 - домен не готов к остановке тактового сигнала;

RST - формирует сигнал сброса для домена. (***R/W***)

0 - сигнал сброса не формируется;

1 - сигнал сброса формируется.

При автоматической установке данный бит автоматически сбрасывается через интервал определяемый регистром ***TRSTR.***

5.12.3.7 TRSTR[31:0]

Задает значение интервала времени в тактах, в течении которого будет формироваться сброс. Используется только для автоматического включения доменов.

5.12.3.8 TPWRDR[31:0]

Задает значение интервала времени в тактах, в течении которого питание домена станет стабильным после включения. Используется только для автоматического включения доменов.

5.13 Контроллер ПДП XDMAC

Контроллер обеспечивает периферийным устройствам прямой доступ в память, а также может производить обмен данными между разными блоками памяти. Особенности контроллера ПДП:

- 4 универсальных канала, каждый из которых может запускаться как программно, так и по запросу периферийного устройства,
- встроенный буфер на 16 64-разрядных слов (128 байт),
- программная настройка типов транзакций на шине AMBA AXI, настраивается длина пакета, разрядность передаваемых данных и тип адресации,
- программная настройка приоритетов между одновременно работающими каналами - можно выбрать либо фиксированный приоритет, либо круговой (round-robin).

Передача данных реализуется контроллером с помощью транзакций чтения и записи на шине AMBA AXI: контроллер производит чтение данных из источника, помещает данные во встроенный буфер, затем записывает данные в устройство-приёмник. Источником и приёмником данных могут быть блоки памяти или периферийные устройства системы, причём контроллер поддерживает как фиксированную адресацию (для устройств с программно доступным регистром-буфером), так и адресацию с пост-инкрементом (например, для блоков памяти).

5.13.1 Устройство контроллера ПДП XDMAC

Блок не имеет сигналов, соединенных с внешними выводами микросхемы.

На рисунке ниже приведена структурная схема блока:

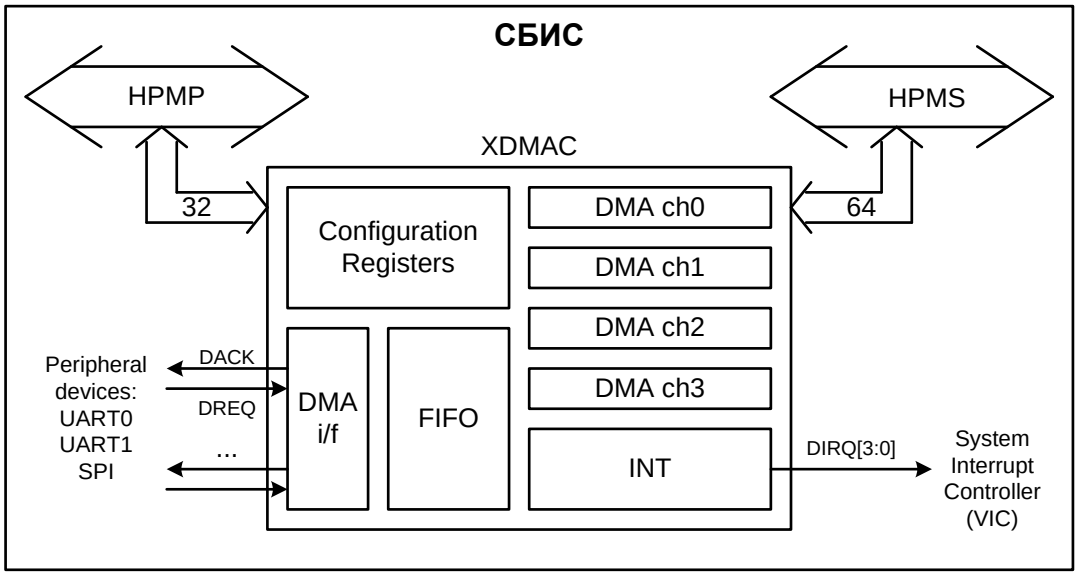


Рисунок 5.40 – Схема блока контроллера ПДП

Контроллер ПДП имеет два AXI интерфейса: 64х разрядный master-интерфейс и 32х разрядный slave интерфейс. Master интерфейс используется для передачи данных, а slave интерфейс используется для настройки каналов контроллера ПДП. Всего XDMAC имеет 4 канала передачи данных, каждый из которых может быть настроен независимо. Каждый канал может осуществлять пересылки память-память, а также память-периферия. Доступ к периферийным устройствам осуществляется через AXI интерфейс S2P, соединяющий системный коммутатор HPMS и периферийный коммутатор HPMP.

К контроллеру через интерфейс ПДП подключены 3 периферийных устройства: SPI, UART0 и UART1. Остальные периферийные устройства либо не поддерживают работу с контроллером ПДП (CAN, I2C), либо имеют встроенные контроллеры ПДП (USB, Ethernet).

Каждый канал контроллера ПДП имеет выход прерывания, подключенных к системному контроллеру прерываний VIC0.

5.13.2 Программно доступные регистры контроллера ПДП

Каждый канал имеет собственный набор программно доступных регистров. Имеется также общий конфигурационный регистр контроллера. Программно доступные регистры контроллера XDMAC расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение XDMAC Base = 0xFFF60000 и общий размер 64 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.92 - Список регистров контроллера ПДП

Канал	Название	Адрес регистра	Тип	Описание
Общий	XDACS	Base+00h	ЧТ/ЗП	Общий конфигурационный регистр контроллера
Ch0	XDTBC0	Base+10h	ЧТ/ЗП	Регистр размера передаваемых данных
	XDSSA0	Base+14h	ЧТ/ЗП	Регистр начального адреса в источнике данных
	XDDSA0	Base+18h	ЧТ/ЗП	Регистр начального адреса в приёмнике данных
	XDSAC0	Base+1Ch	ЧТ/ЗП	Регистр настройки транзакций к источнику данных
	XDDAC0	Base+20h	ЧТ/ЗП	Регистр настройки транзакций к приёмнику данных
	XDDES0	Base+28h	ЧТ/ЗП	Регистр запуска
Ch1	XDTBC1	Base+40h	ЧТ/ЗП	Регистр размера передаваемых данных
	XDSSA1	Base+44h	ЧТ/ЗП	Регистр начального адреса в источнике данных
	XDDSA1	Base+48h	ЧТ/ЗП	Регистр начального адреса в приёмнике данных
	XDSAC1	Base+4Ch	ЧТ/ЗП	Регистр настройки транзакций к источнику данных
	XDDAC1	Base+50h	ЧТ/ЗП	Регистр настройки транзакций к приёмнику данных
	XDDES1	Base+58h	ЧТ/ЗП	Регистр запуска
Ch2	XDTBC2	Base+70h	ЧТ/ЗП	Регистр размера передаваемых данных
	XDSSA2	Base+74h	ЧТ/ЗП	Регистр начального адреса в источнике данных
	XDDSA2	Base+78h	ЧТ/ЗП	Регистр начального адреса в приёмнике данных
	XDSAC2	Base+7Ch	ЧТ/ЗП	Регистр настройки транзакций к источнику данных
	XDDAC2	Base+80h	ЧТ/ЗП	Регистр настройки транзакций к приёмнику данных
	XDDES2	Base+88h	ЧТ/ЗП	Регистр запуска
Ch3	XDTBC3	Base+A0h	ЧТ/ЗП	Регистр размера передаваемых данных
	XDSSA3	Base+A4h	ЧТ/ЗП	Регистр начального адреса в источнике данных
	XDDSA3	Base+A8h	ЧТ/ЗП	Регистр начального адреса в приёмнике данных
	XDSAC3	Base+ACh	ЧТ/ЗП	Регистр настройки транзакций к источнику данных
	XDDAC3	Base+B0h	ЧТ/ЗП	Регистр настройки транзакций к приёмнику данных
	XDDES3	Base+B8h	ЧТ/ЗП	Регистр запуска

XDACS - общий конфигурационный регистр контроллера. Формат регистра XDACS приведён ниже (Таблица 5.93).

Таблица 5.93 - Формат регистра XDACS

Биты	Название	Тип	Описание
[31:29]			Зарезервировано
[28]	XE	ЧТ/ЗП	Разрешение работы контроллера ПДП. 0 - работа всех каналов запрещена, 1 - работа каналов разрешена. Запись значения 0 в данный бит останавливает все работающие каналы.
[27:25]			Зарезервировано
[24]	CP	ЧТ/ЗП	Приоритеты каналов при выдаче транзакций: 0 - фиксированный приоритет (канал 0 - наиболее приоритетный, канал 3 - наименее приоритетный), 1 - круговой приоритет (наиболее приоритетный - следующий по номеру после текущего).
[23:17]			Зарезервировано
[16]	XS	ЧТ	Статус контроллера: 0 - все каналы контроллера не работают, 1 - хотя бы один канал контроллера работает
[15:0]			Зарезервировано

XDTBC - регистр размера блока передаваемых данных в байтах. Формат регистра XDTBC приведён ниже (Таблица 5.94).

Таблица 5.94 - Формат регистра XDTBC

Биты	Название	Тип	Описание
[31:0]	TBC	ЧТ/ЗП	Счётчик передаваемых данных (в байтах).

XDSSA - регистр начального адреса в источнике данных. Формат регистра XDSSA приведён ниже (Таблица 5.95).

Таблица 5.95 - Формат регистра XDSSA

Биты	Название	Тип	Описание
[31:0]	SSA	ЧТ/ЗП	Начальный адрес чтения из источника.

XDDSA - регистр начального адреса в приёмнике данных. Формат регистра XDDSA приведён ниже (Таблица 5.95).

Таблица 5.96 - Формат регистра XDDSA

Биты	Название	Тип	Описание
[31:0]	DSA	ЧТ/ЗП	Начальный адрес записи в приёмник.

XDSAC - регистр настройки транзакций к источнику данных. Формат регистра XDSAC приведён ниже (Таблица 5.97).

Таблица 5.97 – Формат регистра XDSAC

Биты	Название	Тип	Описание
[31:18]			Зарезервировано
[17:16]	SBS	ЧТ/ЗП	Размер данных в транзакциях чтения. Данное поле устанавливает разрядность данных в транзакциях чтения на шине AMBA AXI. 00b - 1 байт, 01b - 2 байта, 10b - 4 байта, 11b - 8 байт.
[15:12]			Зарезервировано
[11:8]	SBL	ЧТ/ЗП	Длина пакета в транзакциях чтения. Данное поле устанавливает длину пакета в транзакциях чтения на шине AMBA AXI. Контроллер может использовать пакеты меньшей длины при невыровненных адресе или размере передачи, то есть данное поле устанавливает максимальную длину пакета. 0000b - 1 элемент данных в пакете, 0001b - 2 элемента данных в пакете, ..., 1111b - 16 элементов данных в пакете.
[7:3]			Зарезервировано
[2]	SAF	ЧТ/ЗП	Тип адресации при обращении к источнику: 0 - с пост-инкрементом (тип INCR на шине AMBA AXI), 1 - фиксированный адрес (тип FIXED на шине AMBA AXI).
[1]			Зарезервировано
[0]	SRL	ЧТ/ЗП	Восстановление начального адреса чтения. При адресации с пост-инкрементом текущий адрес обращения к источнику меняется. Если после работы канала необходимо восстановить в регистре XDSSA начальный адрес, то следует записать в этот бит значение 1. Если необходимо оставить получившееся значение регистра XDSSA, то следует записать 0.

XDDAC - регистр настройки транзакций к приёмнику данных. Формат регистра XDDAC приведён ниже (Таблица 5.98).

Таблица 5.98 – Формат регистра XDDAC

Биты	Название	Тип	Описание
[31:18]			Зарезервировано
[17:16]	DBS	ЧТ/ЗП	Размер данных в транзакциях записи. Данное поле устанавливает разрядность данных в транзакциях записи на шине AMBA AXI. 00b - 1 байт, 01b - 2 байта, 10b - 4 байта, 11b - 8 байт.
[15:12]			Зарезервировано
[11:8]	DBL	ЧТ/ЗП	Длина пакета в транзакциях записи. Данное поле устанавливает длину пакета в транзакциях записи на шине AMBA AXI. Контроллер может использовать пакеты меньшей длины при невыровненных адресе или размере передачи, то есть данное поле устанавливает максимальную длину пакета. 0000b - 1 элемент данных в пакете, 0001b - 2 элемента данных в пакете, ..., 1111b - 16 элементов данных в пакете.
[7:3]			Зарезервировано
[2]	DAF	ЧТ/ЗП	Тип адресации при обращении к приёмнику: 0 - с пост-инкрементом (тип INCR на шине AMBA AXI), 1 - фиксированный адрес (тип FIXED на шине AMBA AXI).
[1]			Зарезервировано
[0]	DRL	ЧТ/ЗП	Восстановление начального адреса записи. При адресации с пост-инкрементом текущий адрес обращения к приёмнику меняется. Если после работы канала необходимо восстановить в регистре XDDSA начальный адрес, то следует записать в этот бит значение 1. Если необходимо оставить получившееся значение регистра XDDSA, то следует записать 0.

XDDES - регистр запуска канала. Формат регистра XDDES приведён ниже (Таблица 5.99).

Таблица 5.99 -Формат регистра XDDES

Биты	Название	Тип	Описание
[31:29]			Зарезервировано
[28]	CE	ЧТ/ЗП	Бит запуска канала. 0 - канал остановлен, 1 - канал запущен. Запись значения 1 запускает канал, запись значения 0 останавливает канал. Канал также может остановиться по одной из следующих причин: завершение передачи (нормальное завершение), ошибка транзакции (обращение в недоступную область адресного пространства), запрещение работы контроллера (сброс бита XE в регистре XDACS).
[27:25]			Зарезервировано
[24]	SE	ЧТ/ЗП	Программный триггер запуска канала. Запись значения 1 в данный бит инициирует программный триггер запуска канала.
[23:20]	TF	ЧТ/ЗП	Выбор триггера запуска канала. Данное поле устанавливает, какое событие является триггером запуска для данного канала: 0000b - нет триггера запуска, 0001b - программный триггер запуска (бит SE), 0010b - триггером запуска является запрос от устройства 0, 0011b - триггером запуска является запрос от устройства 1, ..., 1111b - триггером запуска является запрос от устройства 13.
[19:13]			Зарезервировано
[12]	BT	ЧТ/ЗП	Режим запуска: 0 - нормальный режим, 1 - режим поблочного запуска. В нормальном режиме канал запускается сразу после триггера запуска и работает до окончания передачи заданного в регистре XDTBC количества данных. В режиме поблочного запуска на один триггер запуска канал производит чтение и запись одного блока данных. Размер блока определяется полями SBS и SBL регистра XDSAC (то есть канал производит одну транзакцию чтения). В режиме поблочного запуска канал также останавливается, когда заканчивает передачу заданного в регистре XDTBC количества данных.
[11:9]			Зарезервировано
[8]	BR	ЧТ/ЗП	Восстановление счётчика XDTBC после окончания работы канала. В ходе работы счётчик XDTBC декрементируется. Если после работы канала необходимо восстановить в регистре XDTBC начальное значение счётчика, то следует записать в этот бит значение 1. Если необходимо оставить получившееся значение регистра XDTBC, то следует записать 0.
[7:5]			Зарезервировано
[4]	AT	ЧТ/ЗП	Способ выдачи подтверждения к периферийному устройству. 0 - подтверждение выдаётся в момент чтения, 1 - подтверждение выдаётся в момент записи.
[3:2]			Зарезервировано
[1]	EI	ЧТ/ЗП	Включение запроса на прерывание по ошибке транзакции. 0 - прерывание маскируется, 1 - прерывание разрешено.
[0]	TI	ЧТ/ЗП	Включение запроса на прерывание по завершению работы. 0 - прерывание маскируется, 1 - прерывание разрешено.

5.13.3 Включение и выключение контроллера ПДП

В начальном состоянии все каналы контроллера выключены. Перед запуском канала необходимо разрешить использование контроллера в общем конфигурационном регистре контроллера (XDACS). Контроллер также может быть выключен с помощью регистра XDACS.

5.13.4 Процедура настройки канала

Запуск канала производится по следующему алгоритму:

- прочитать регистр запуска канала XDDES - убедиться, что канал закончил работу после предыдущего запуска,
- записать в регистр XDTBC размер передаваемых за один запуск данных в байтах,
- записать в регистр XDSSA начальный адрес в источнике данных,
- записать в регистр XDDSA начальный адрес в приёмнике данных,

- записать в регистр XDSAC параметры транзакций к источнику,
- записать в регистр XDDAC параметры транзакций к приёмнику,
- записать в регистр XDDES параметры запуска канала.

5.14 Контроллер асинхронного последовательного интерфейса UART

Контроллер универсального асинхронного приемопередатчика (UART – Universal Asynchronous Receiver Transmitter) в СБИС СНП ВП имеет следующие характеристики:

- Содержит независимые буферы приема (32x12) и передачи (32x8) типа FIFO, что позволяет снизить интенсивность прерываний центрального процессора. Есть возможность программно ограничить глубину этих буферов до одного слова, в таком случае они будут работать в режиме буферных регистров.
- Программное управление скоростью обмена. Обеспечивается возможность деления тактовой частоты опорного тактового сигнала в диапазоне от 16 до 65535*16. Допускается использование нецелых коэффициентов деления частоты.
- Поддержка стандартных элементов асинхронного протокола связи – стартового, стопового бит и бита контроля четности, которые добавляются перед передачей и удаляются после приема.
- Независимое маскирование прерываний от буфера FIFO передатчика, буфера FIFO приемника, по таймауту приемника, а также в случае обнаружения ошибки.
- Поддержка прямого доступа к памяти.
- Обнаружение ложных стартовых бит.
- Формирование и обнаружение сигнала разрыва линии.
- Функция управления модемом не поддерживается.
- Возможность организации аппаратного управления потоком данных.
- Полностью программируемый асинхронный последовательный интерфейс с характеристиками:
  - Символ данных 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 бит.
  - Скорость передачи данных – до UARTCLK/16 Бод (максимум 921600 Бод).
- Наличие идентификационного регистра, однозначно идентифицирующего контроллер, что позволяет операционной системе выполнять автоматическую конфигурацию.

5.14.1 Устройство UART портов СБИС СНП-ВП

В таблице ниже приведен список внешних выводов, относящихся к портам UART0 и UART1:

Таблица 5.100 – Выводы микросхемы, входящие в состав UART портов

Вывод	Тип буфера	Примечание
XCTS0	in	UART0. Сигнал готовности к передаче (активный низкий уровень)
UART0RXD	in	UART0. Вход данных
UART0TXD	out	UART0. Выход данных
XRTS0	out	UART0. Сигнал запроса на передачу (активный низкий уровень)
XCTS1	in	UART1. Сигнал готовности к передаче (активный низкий уровень)
UART1RXD	in	UART1. Вход данных
UART1TXD	out	UART1. Выход данных
XRTS1	out	UART1. Сигнал запроса на передачу (активный низкий уровень)

Примечание. Выводы XCTSx и XRTSx микросхемы соответствует выводам nUARTCTS и nUARTRTS контроллера соответственно. Далее по тексту будут использоваться названия выводов контроллера.

На рисунке (см. Рисунок 5.41) приведена блок-схема контроллера порта UART.

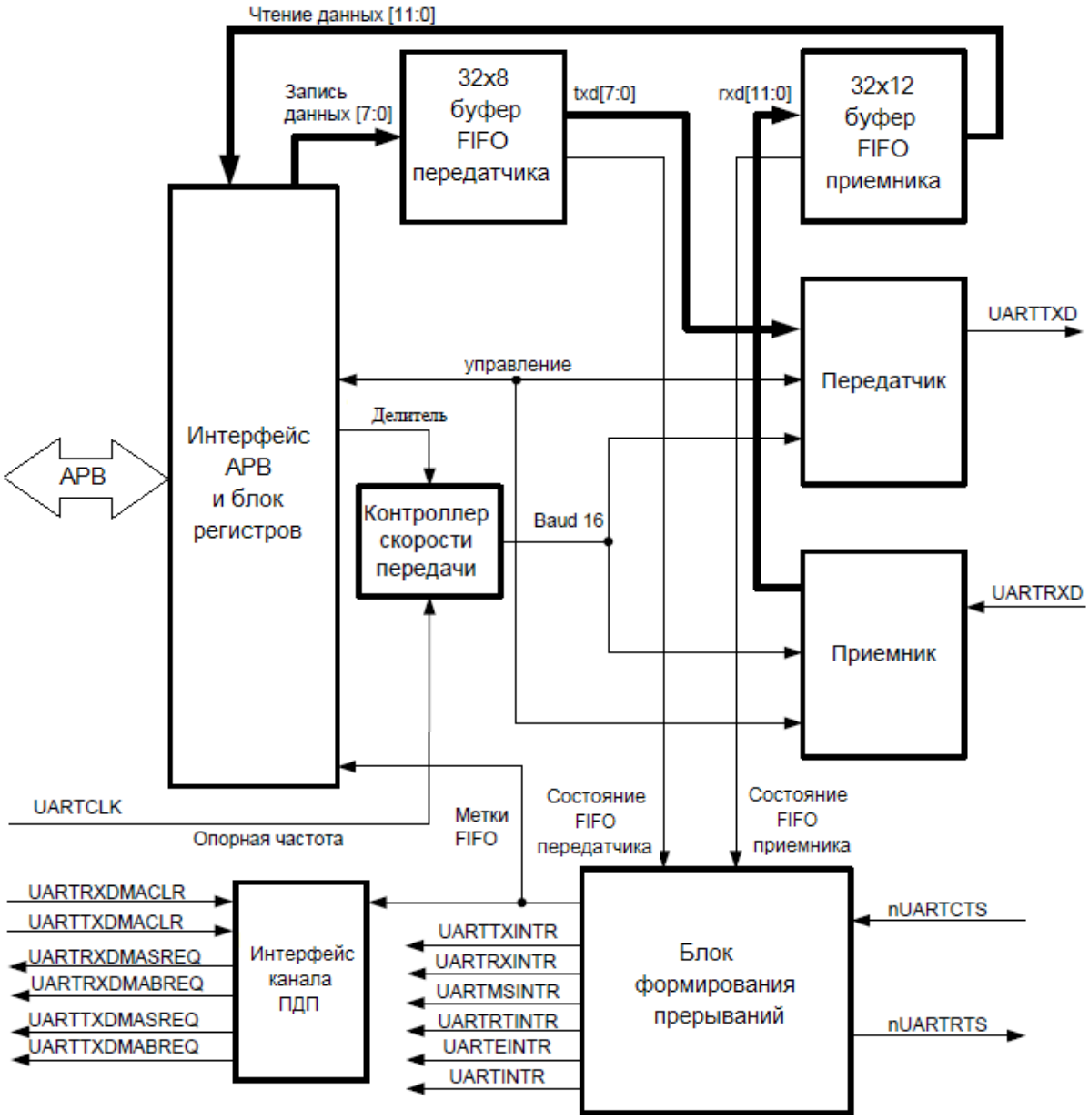


Рисунок 5.41 – Блок - схема контроллера интерфейса UART

Интерфейс APB и блок регистров

Блок содержит набор программно-доступных регистров и интерфейс с шиной APB.

Контроллер скорости передачи

Блок содержит счетчики, формирующие внутренний тактовый сигнал Baud16. Этот сигнал имеет частоту в 16 раз выше скорости обмена и используется для тактирования блоков приемника и передатчика.

Буфер FIFO передатчика

Буфер имеет ширину 8 бит и глубину 32 слова. Данные от центрального процессора, записанные через шину APB, сохраняются в буфере до тех пор, пока не будут считаны блоком передатчика. Существует возможность программно сократить глубину FIFO передатчика до одного слова, в этом случае он будет функционировать как буферный регистр.

Буфер FIFO приемника

Буфер имеет ширину 12 бит и глубину 32 слова. Принятые приемником данные и соответствующие им коды ошибок сохраняются в буфере до тех пор, пока не будут считаны центральным процессором через шину APB. Существует возможность программно сократить глубину FIFO приемника до одного слова, в этом случае он будет функционировать как буферный регистр.

Блок передатчика

Блок передатчика осуществляет параллельно-последовательное преобразование данных, считанных из буфера передатчика. Передатчик выдает последовательный поток бит в порядке: стартовый бит, биты данных, начиная с младшего значащего разряда, бит четности, и, наконец, стоповые биты, в соответствии с конфигурацией, записанной в регистре управления линией.

Блок приемника

Приемник выполняет последовательно-параллельное преобразование данных, полученных от периферийного устройства после обнаружения корректного стартового импульса. Кроме того, производятся проверки переполнения буфера приемника, выявляются ошибки контроля четности, ошибки в структуре сигнала, а также разрыва линии. Признаки обнаружения этих ошибок также сохраняются в буфере FIFO приемника.

Блок формирования прерываний

Блок формирует множество прерываний внутри, но к системному контроллеру прерываний подключено только одно комбинированное прерывание UARTINTR Точную причину прерывания можно определить программно.

Интерфейс канала ПДП

Этот интерфейс позволяет подключить контроллер UART к контроллеру ПДП и организовать передачу данных без участия центрального процессора.

5.14.2 Описание функционирования UART порта

Настройки контроллера хранятся в регистре UARTLCR (UART Line Control Register). Внутренне это 30-битный регистр, но доступ к нему осуществляется через следующие программно-доступные регистры по шине APB:

- UARTLCR_H – регистр управления линией (длина слова, количество стоповых-бит, контроль четности, генерация состояния разрыва линии)
- UARTIBRD – целая часть делителя, используемого в контроллере скорости передачи
- UARTFBRD – дробная часть делителя, используемого в контроллере скорости передачи

5.14.2.1 Дробный коэффициент деления

Коэффициент деления для формирования скорости передачи данных состоит из 22 бит, при этом для представления его целой части выделено 16 бит, а для дробной части 6 бит. Возможность задания дробных коэффициентов деления позволяет осуществлять обмен данными как со стандартными информационными скоростями, так и с произвольными.

Целая часть коэффициента деления записывается в 16-битный регистр UARTIBRD (UART Integer Baud Rate Divider). Дробная часть коэффициента деления записывается 6-битный в регистр UARTFBRD (UART Fractional Baud Rate Divider):

Коэффициент деления вычисляется по формуле:

$$BAUDDIV = F_{UARTCLK} / (16 * Baud\ Rate) = BRD_I + BRD_F$$

где BRD_I – целая часть, а BRD_F – дробная часть коэффициента деления,

$F_{UARTCLK}$ – тактовая частота контроллера UART;

Baud Rate – требуемая скорость передачи данных.

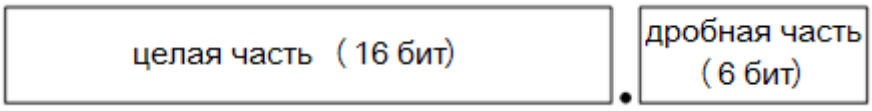


Рисунок 5.42 – Дробный коэффициент деления

Шестибитное значение, записываемое в регистр UARTFBRD, вычисляется путем выделения дробной части требуемого коэффициента деления, умножения ее на 64 (то есть на 2^n, где n – ширина регистра UARTFBRD) и округления до ближайшего целого числа:

$$m = integer(BRD_F * 64 + 0.5)$$

где integer – операция отсечения дробной части числа.

В таблице ниже приведены некоторые значения делителей и соответствующие им скорости передачи для $F_{UARTCLK}$ = 80МГц:

Таблица 5.101 – Возможные скорости порта UART

Делитель BAUDDIV	BRD_I	BRD_F	Получаемая скорость UART, Бод	Стандартный ряд скоростей UART, Бод	Ошибка, %
4166,666667	4166	0,666	1200,00	1200	0,00
2083,333333	2083	0,333	2400,01	2400	0,00
1041,666667	1041	0,666	4799,98	4800	0,00
520,8333333	520	0,833	9600,10	9600	0,00
347,2222222	347	0,222	14400,18	14400	0,00
260,4166667	260	0,416	19199,68	19200	0,00
173,6111111	173	0,611	28800,35	28800	0,00
130,2083333	130	0,208	38401,57	38400	0,00
86,80555556	86	0,805	57595,72	57600	0,01
65,10416667	65	0,104	76794,30	76800	0,01
43,40277778	43	0,402	115191,45	115200	0,01
21,70138889	21	0,701	230382,90	230400	0,01
10,85069444	10	0,85	461126,99	460800	-0,07
5,425347222	5	0,425	922339,05	921600	-0,08

Дробный коэффициент деления частоты обеспечивает требуемую точность стандартной скорости обмена UART в диапазоне от 1200 до 921600 бод включительно.

5.14.2.2 Передача и прием данных

Для хранения передаваемых и принимаемых данных используются буфера FIFO глубиной 32 слова каждый. Буфер приемника имеет ширину 12 бит, т.к. дополнительно к байту данных хранит еще 4 бита состояния.

Для начала передачи данные заносятся в буфер FIFO передатчика. Если работа приемопередатчика UART разрешена, то начинается передача информационного кадра с параметрами, указанными в регистре управления линией UARTLCR_H. Передача данных продолжается до опустошения буфера FIFO передатчика. Сразу же после попадания первых данных в буфер FIFO передатчика сигнал BUSY переходит в высокое состояние. Это состояние 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 выдан передатчиком в линию. Сигнал BUSY может находиться в высоком состоянии даже после перевода контроллера в выключенное состояние.

В случае, если приемник находился в неактивном состоянии (на линии входного сигнала UARTRXD постоянно присутствовала единица) и произошел переход входного сигнала с высокого на низкий логический уровень (обнаружен стартовый бит), включается счетчик, такти-

руемый сигналом Baud16, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов.

Стартовый бит считается достоверным в случае, если сигнал на линии UARTRXD сохраняет низкий логический уровень в течение восьми тактов сигнала Baud16 с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

В случае если обнаружен достоверный стартовый бит, производится регистрация последовательности данных на входе приемника. Очередной бит данных фиксируются каждые 16 отсчетов тактового сигнала Baud16 с момента определения стартового бита (что приблизительно соответствует середине каждого символа). Производится регистрация всех бит данных (согласно запрограммированным параметрам) и 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 принято полностью, данные сохраняются в буфер FIFO приемника, наряду с четырьмя битами признаков ошибки, связанных с этим принятым словом.

На рисунке ниже приведен кадр данных UART.

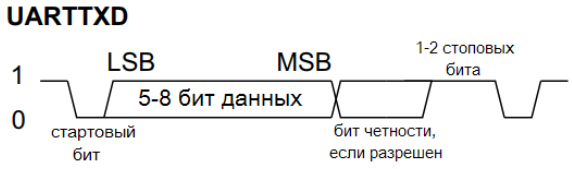


Рисунок 5.43 – Кадр данных UART

5.14.2.3 Признаки ошибок

Вместе с каждым принятым словом (символом) в буфере FIFO приемника сохраняются соответствующие этому слову признаки ошибок. Три признака ошибок сохраняются в [10:8] разрядах слов буфера FIFO приемника. Еще существует признак переполнения FIFO, который хранится в 11-м разряде слова. В таблице ниже приведены функции разрядов слов буфера FIFO приемника:

Таблица 5.102 – Функции разрядов слов FIFO приемника

Разряд FIFO	Функция
11	Признак переполнения
10	Ошибка разрыва линии
9	Ошибка контроля четности
8	Ошибка в структуре кадра
7:0	Принятый символ данных

Признак переполнения не соответствует символу в приемном FIFO вместе с которым он хранится. Ошибка переполнения возникает в случае, когда приемный FIFO буфер полный, следующий символ полностью принят во входном сдвиговом регистре и данные продолжают поступать. В такой ситуации какое-то количество принимаемых символов будет перезаписано в сдвиговом регистре. Когда в FIFO освободится место и очередной принятый символ будет переписан из сдвигового регистра в FIFO к нему добавится признак переполнения. Фактически наличие признака переполнения в старших разрядах слова FIFO означает, что какое-то количество символов перед ним было потеряно.

5.14.2.4 Режим выключенных буферов FIFO

Контроллер позволяет программно выключить буфера FIFO приемника и передатчика (далее по тексту этот режим будет называться режимом «выключенного FIFO»). В этом режиме размер буферов FIFO приемника и передатчика сокращается до одного слова каждый. В действительности аппаратно FIFO не выключаются, а происходит манипуляция с указателями, которая позволяет рассматривать FIFO просто как байтовый буферный регистр. В режиме выключенного FIFO и если выходной сдвиговый регистр свободен, то передаваемые данные записываются сразу в него минуя буферный регистр. Ошибка переполнения при приеме в этом режиме генерируется в ситуации, когда принято два символа (один в буферном регистре, а другой в сдвиговом регистре) и нет чтения по шине APB.

5.14.2.5 Тестирование в режиме обратной петли (loopback)

Контроллер может быть переведен в тестовый режим обратной петли (loopback) путем установки бита LBE в регистре управления UARTCR. В этом режиме данные с выхода UARTTXD передаются на вход UARTRXD внутри микросхемы.

5.14.2.6 Режим аппаратного управления потоком

Модемный интерфейс в СБИС СНП-ВП не используется, используются только выводы аппаратного управления потоком nUARTCTS и nUARTRTS. Режим аппаратного управления потоком опционален и имеет программные настройки.

На рисунке ниже приведена схема соединения двух микросхем с использованием сигналов аппаратного управления потоком.

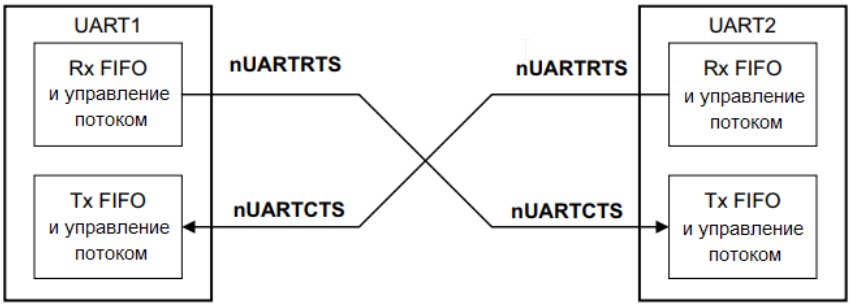


Рисунок 5.44 – Соединение с использованием аппаратного управления потоком

Если включено RTS управление потоком, то выходной сигнал nUARTRTS будет иметь активное состояние до тех пор, пока данные в буфере FIFO приемника не достигнут некоторого, заранее запрограммированного количества. Таким образом приемник сообщает передатчику, что необходимо приостановить передачу данных до освобождения места в приемном FIFO. Если RTS управление выключено, то прием ведется до момента полного заполнения приемного FIFO.

Если включено CTS управление потоком, то передатчик может передавать только в случае наличия активного состояния на входном сигнале nUARTCTS. Перед передачей каждого символа передатчик проверяет уровень этого сигнала, и если он неактивный, то передача не происходит. Если в процессе передачи символа сигнал nUARTCTS переходит в неактивное состояние, то передача текущего символа завершится. Если CTS управление выключено, то передача ведется до полного опустошения, передающего FIFO.

Управление режимами аппаратного управления потоком осуществляется с помощью бит RTSEn и CTSEn регистра управления UARTCR. Режимы RTS управления и CTS управления потоком могут быть включены как по отдельности, так и вместе. В таблице ниже (Таблица 5.103) приведены их возможные комбинации в зависимости от установки управляющих бит регистра UARTCR.

Таблица 5.103 – Возможные комбинации режимов аппаратного управления потоком

CTSEn	RTSEn	Описание
1	1	Включены оба режима аппаратного управления потоком (RTS и CTS)
1	0	Включен режим CTS управления потоком
0	1	Включен режим RTS управления потоком
0	0	Оба режима управления потоком выключены

Примечание. Если включен режим RTS управления потоком, то ПО не может использовать бит RTSEn регистра UARTCR для управления выводом nUARTRTS.

5.14.2.7 Интерфейс канала ПДП

Контроллер имеет интерфейс с внешним контроллером ПДП (контроллер XDMAC в СБИС СНП ВП). Управление этим интерфейсом осуществляется с помощью регистра управления каналом ПДП UARTDMACR. Интерфейс канала ПДП содержит следующие сигналы:

Для приемника:

- UARTRXDMASREQ – запрос на считывание каналом ПДП одного символа из приемного FIFO. Запрос выдается в состоянии, когда в приемном FIFO находится как минимум один принятый символ.
- UARTRXDMABREQ - запрос на считывание каналом ПДП пакета данных из приемного FIFO. Запрос выдается в состоянии, когда в приемном FIFO находится символов больше чем заранее запрограммированное количество (пороговый уровень, watermark level) в регистре UARTIFLS.
- UARTRXDMACLR – сигнал подтверждения от контроллера ПДП, сигнализирующий об окончании обработки ранее выданного контроллером UART запроса на чтение. Если подтверждение соответствует запросу на пакет данных, то оно выдается во время передачи последнего символа пакета.

Для передатчика:

- UARTTXDMASREQ – запрос на передачу (запись) каналом ПДП одного символа в передающее FIFO. Запрос выдается в состоянии, когда в передающем FIFO есть место хотя-бы для одного символа.
- UARTTXDMABREQ - запрос на передачу (запись) каналом ПДП пакета данных в передающее FIFO. Запрос выдается в состоянии, когда в передающем FIFO находится символов меньше чем заранее запрограммированное количество (пороговый уровень, watermark level) в регистре UARTIFLS.
- UARTTXDMACLR - сигнал подтверждения от контроллера ПДП, сигнализирующий об окончании обработки ранее выданного контроллером UART запроса на запись. Если подтверждение соответствует запросу на пакет данных, то оно выдается во время передачи последнего символа пакета.

ПДП запросы на отдельный символ и на пакет не взаимоисключающие, оба запроса могут быть активны одновременно. Например, если в приемном FIFO находится символов больше, чем пороговый уровень, то оба запроса будут активны. Далее, когда символов станет меньше порогового уровня, то пакетный запрос будет снят, а запрос на один символ будет активен до полного опустошения FIFO. Такое поведение полезно если в FIFO находится количество символов, которое не может быть считано целым количеством пакетных чтений. Например, если в FIFO находится 19 символов и пороговый уровень 4 символа, то в таком случае контроллер ПДП выполнит 4 пакетных чтения, а оставшиеся 3 символа будут считаны единичными запросами.

Каждый запрос остается активным до тех пор, пока контроллер ПДП не выставит соответствующий сигнал подтверждения DMACLR. После сброса сигнала запроса, запрос может быть снова выставлен контроллером UART в случае если ранее перечисленные условия выставления запроса все еще выполняются. Любые запросы к контроллеру ПДП не выставляются если контроллер UART выключен (бит UARTEN в регистре UARTCR равен 0) или соответ-

ствующий типу запроса бит управления интерфейсом ПДП находится в неактивном состоянии (биты TXDMAE и RXDMAE в регистре управления каналом ПДП UARTDMACR).

В режиме выключенных FIFO буферов (бит FEN регистра UARTCR_H равен 0) контроллер UART поддерживает только посимвольный обмен (чтение/запись пакета каналом ПДП запрещено). Соответственно, только символьные запросы UARTRXDMASREQ и UARTTXDMASREQ могут быть выданы контроллером UART.

В режиме включенных FIFO буферов (бит FEN регистра UARTCR_H равен 1) контроллер UART поддерживает и пакетный и посимвольный обмен. Тип запроса, который ожидается контроллером UART от контроллера ПДП зависит от текущего количества символов в FIFO и запрограммированного порогового уровня. Пороговые уровни принимающего и передающего FIFO задаются полями RXIFLSEL и TXIFLSEL регистра UARTIFLS соответственно. В таблице ниже приведено соответствие пороговых уровней выдачи пакетных запросов и абсолютного количества пустых/занятых ячеек передающего и принимающего FIFO (Таблица 5.104).

Таблица 5.104 – Пороговые уровни выдачи пакетных ПДП запросов

Пороговый уровень	FIFO передатчика (количество пустых ячеек)	FIFO приемника (количество занятых ячеек)
1/8	28	4
1/4	24	8
1/2	16	16
3/4	8	24
7/8	4	28

Также, в регистре управления каналом ПДП UARTDMACR существует бит DMAONERR который управляет выдачей запросов приемника к контроллеру ПДП в случае возникновения ошибок приема данных. Если бит DMAONERR установлен в 1, и в случае обнаружения какой-либо ошибки приема данных (активно прерывание UARTEINTR) запросы приемника UARTRXDMASREQ и UARTRXDMABREQ блокируются до тех пор, пока прерывание UARTEINTR находится в активном состоянии. Данный бит никак не влияет на выдачу ПДП запросов от передатчика.

В СБИС СНП ВП установлено следующее соответствие между выходами запросов контроллера UART и входами запросов от периферийных устройств контроллера ПДП(XDMAC).

Таблица 5.105 – Соответствие выходов ПДП запросов UART и входов XDMAC

XDMAC		UART	
DREQ[11]	вход	UART1TXDMABREQ	выход
DREQ[10]	вход	UART1TXDMASREQ	выход
DREQ[9]	вход	UART1RXDMABREQ	выход
DREQ[8]	вход	UART1RXDMASREQ	выход
DREQ[7]	вход	UART0TXDMABREQ	выход
DREQ[6]	вход	UART0TXDMASREQ	выход
DREQ[5]	вход	UART0RXDMABREQ	выход
DREQ[4]	вход	UART0RXDMASREQ	выход
DACK [11] | DACK [10]	выход	UART1TXDMACLR	вход
DACK [9] | DACK [8]	выход	UART1RXDMACLR	вход
DACK [7] | DACK [6]	выход	UART0TXDMACLR	вход
DACK [5] | DACK [4]	выход	UART0RXDMACLR	вход

5.14.3 Регистровая модель контроллера порта UART

Микросхема содержит 2 контроллера порта UART0 и UART1, базовые адреса для регистров управления равны FFF0_B000h и FFF0_C000h соответственно.

Запрещен доступ к регистрам по зарезервированным адресам, т.к. это может привести к непредвиденным результатам. Запрещена модификация зарезервированных полей регистров, а при чтении их значения должны игнорироваться, если в тексте описания не указано обратное.

Спецификация регистров представлена в таблице ниже:

Таблица 5.106 – Спецификация регистров контроллера порта UART

Смещение	Название	Тип	Значение после сброса	Размер, бит	Описание
0x000	UARTDR	ЧТ/ЗП	0x---	12/8	Регистр данных
0x004	UARTRSR/ UARTECR	ЧТ/ЗП	0x0	4/0	Регистр состояния приемника/ Сброс ошибки приемника
0x008-0x014	-	-	-	-	Зарезервировано
0x018	UARTFR	ЧТ	0b-10010---	9	Регистр флагов
0x01C	-	-	-	-	Зарезервировано
0x020	-	-	-	-	Зарезервировано
0x024	UARTIBRD	ЧТ/ЗП	0x0000	16	Регистр целой части коэффициента деления частоты
0x028	UARTFBRD	ЧТ/ЗП	0x00	6	Регистр дробной части коэффициента деления частоты
0x02C	UARTLCR_H	ЧТ/ЗП	0x00	8	Регистр управления линией
0x030	UARTCR	ЧТ/ЗП	0x0300	16	Регистр управления
0x034	UARTIFLS	ЧТ/ЗП	0x12	6	Регистр пороговых уровней заполнения FIFO
0x038	UARTIMSC	ЧТ/ЗП	0x000	11	Регистр управления маскированием прерываний
0x03C	UARTRIS	ЧТ	0x00-	11	Регистр состояния прерываний
0x040	UARTMIS	ЧТ	0x00-	11	Регистр маскированного состояния прерываний
0x044	UARTICR	ЗП	-	11	Регистр сброса прерываний
0x048	UARTDMACR	ЧТ/ЗП	0x00	3	Регистр управления каналом ПДП
0x04C-0x07C	-	-	-	-	Зарезервировано
0x080	UARTTCR	ЧТ/ЗП	0x0	3	Регистр управления тестированием
0x084	UARTITIP	ЧТ/ЗП	0x00	8	Входной регистр интеграционного тестирования
0x088	UARTITOP	ЧТ/ЗП	0x000	14	Выходной регистр интеграционного тестирования
0x08C	UARTTDR	ЧТ/ЗП	0x---	11	Регистр тестовых данных
0x090-0xFDC	-	-	-	-	Зарезервировано
0xFE0	UARTPeriphID0	ЧТ	0x11	8	Регистр UARTPeriphID0
0xFE4	UARTPeriphID1	ЧТ	0x10	8	Регистр UARTPeriphID1
0xFE8	UARTPeriphID2	ЧТ	0x34	8	Регистр UARTPeriphID2
0xFEC	UARTPeriphID3	ЧТ	0x00	8	Регистр UARTPeriphID3
0xFF0	UARTPCellID0	ЧТ	0x0D	8	Регистр UARTPCellID0
0xFF4	UARTPCellID1	ЧТ	0xF0	8	Регистр UARTPCellID1
0xFF8	UARTPCellID2	ЧТ	0x05	8	Регистр UARTPCellID2
0xFFC	UARTPCellID3	ЧТ	0xB1	8	Регистр UARTPCellID3

Далее приведено подробное описание каждого регистра.

5.14.3.1 Регистр данных UARTDR

Регистр данных UARTDR используется для передачи и приема данных по интерфейсу UART.

Операция записи в регистр инициирует передачу данных. Записываемые данные сначала попадают в буфер FIFO передатчика, а после этого слово данных предваряется стартовым битом, дополняется битом контроля четности (если режим контроля четности включен) и стоповым битом. Сформированное слово отправляется в линию передачи данных.

Полученные из линии связи данные считывается путем чтения из регистра UARTDR совместно с соответствующими битами состояния принятого слова данных. Информация о состоянии также может быть получена путем чтения регистра UARTRSR/UARTECR (Таблица 5.108).

Формат регистра UARTDR представлен в таблице ниже.

Таблица 5.107 – Формат регистра UARTDR

Бит	Название	Тип	Назначение
[15:12]	-	-	Зарезервировано
[11]	OE	ЧТ	Признак/ошибка переполнения буфера FIFO приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время, как буфер FIFO заполнен. Сбрасывается в 0 после того, как в приемном FIFO появится свободное место.
[10]	BE	ЧТ	Ошибка разрыва линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). В режиме включенного FIFO данная ошибка ассоциируется с символом на вершине(голове) FIFO очереди, т.е. с символом, который будет прочитан следующим. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита.
[9]	PE	ЧТ	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам бит EPS и SPS в регистре управления линией UARTLCR_H. В режиме включенного FIFO данная ошибка ассоциируется с символом на вершине(голове) FIFO очереди, т.е. с символом который будет прочитан следующим.
[8]	FE	ЧТ	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). В режиме FIFO данная ошибка ассоциируется с последним символом, поступившим в буфер.
[7:0]	DATA	ЧТ/ЗП	Принимаемые данные (чтение) Передаваемые данные (запись)

Примечание. Необходимо запрещать работу приемопередатчика (бит UARTEN регистра UARTCR) перед любым перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние в процессе передачи или приема символа, то перед остановкой он завершает выполняемую операцию.

5.14.3.2 Регистр состояния приемника/сброса ошибки UARTRSR/UARTECR

Состояние приемника может быть считано из регистра UARTRSR. В этом случае информация о состоянии признаков разрыва линии, ошибки контроля четности и ошибки в структуре кадра относится к последнему символу, считанному из регистра данных UARTDR. С другой стороны, признак переполнения приемного FIFO устанавливается немедленно после возникновения этого состояния (и не связан с последним считанным из регистра UARTDR байтом данных).

Запись в регистр UARTECR приводит к сбросу всех признаков ошибок переполнения, четности, структуры кадра, разрыва линии. Также все эти признаки устанавливаются в 0 после сброса устройства.

Формат регистра UARTRSR/UARTECR представлен в таблице ниже.

Таблица 5.108 –Регистр UARTRSR/UARTECR

Биты	Название	Тип	Назначение
[7:0]	-	ЗП	Запись в регистр сбрасывает признаки ошибок формирования кадра, проверки на четность, разрыва линии и переполнения буфера. Содержимое, записываемого слова, не имеет значения.
[7:4]	-	ЧТ	Зарезервировано, при чтении результат не определен
[3]	OE	ЧТ	Признак/ошибка переполнения буфера FIFO приемника. Бит устанавливается в 1 в случае, если на вход приемника поступают данные, в то время как буфер FIFO заполнен. Сбрасывается в 0 после записи в регистр UARTECR. Содержимое буфера FIFO приемника остается верным, так как были перезаписаны данные только во входном сдвиговом регистре. Центральный процессор должен считать данные для того, чтобы освободить буфер FIFO приемника.
[2]	BE	ЧТ	Ошибка разрыва линии. Устанавливается в 1 при обнаружении признака разрыва линии, то есть в случае наличия низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного слова данных (включая стартовый, стоповый биты и бит проверки на четность). Бит сбрасывается в 0 после записи в регистр UARTECR. В случае обнаружения разрыва линии в буфер загружается только один нулевой символ, прием данных возобновляется только после перехода линии в логическую 1 и последующего обнаружения корректного стартового бита.
[1]	PE	ЧТ	Ошибка контроля четности. Устанавливается в 1 в случае, если четность принятого символа данных не соответствует установкам бит EPS и SPS в регистре управления линией UARTLCR_H. Бит сбрасывается в 0 после записи в регистр UARTECR.
[0]	FE	ЧТ	Ошибка в структуре кадра. Устанавливается в 1 в случае, если в принятом символе не обнаружен корректный стоповый бит (корректный стоповый бит равен 1). Бит сбрасывается в 0 после записи в регистр UARTECR.

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. Вместе с тем, информация о состоянии приемника может быть получена непосредственно из регистра данных UARTDR.

5.14.3.3 Регистр флагов UARTFR

После сброса биты TXFF, RXFF и BUSY устанавливаются в 0, а биты TXFE и RXFE устанавливаются в 1.

Формат регистра UARTFR представлен в таблице ниже.

Таблица 5.109 – Формат регистра UARTFR

Биты	Название	Тип	Назначение
[15:9]	-	-	Зарезервировано. Не модифицировать. При чтении заполняются нулями
[8]	-	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[7]	TXFE	ЧТ	Буфер FIFO передатчика пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. В режиме включенного FIFO бит установлен, когда буфер FIFO передатчика пуст, а в режиме выключенного FIFO бит установлен, когда выходной буферный регистр пуст. Данный бит не дает никакой информации о наличии данных в выходном сдвиговом регистре передатчика
[6]	RXFF	ЧТ	Буфер FIFO приемника заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. В режиме включенного FIFO бит установлен, когда буфер FIFO приемника заполнен, а в режиме выключенного FIFO бит установлен когда входной буферный регистр заполнен.
[5]	TXFF	ЧТ	Буфер FIFO передатчика заполнен. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. В режиме включенного FIFO бит установлен, когда буфер FIFO передатчика заполнен, а в режиме выключенного FIFO бит установлен когда выходной буферный регистр заполнен.
[4]	RXFE	ЧТ	Буфер FIFO приемника пуст. Значение бита зависит от состояния бита FEN регистра управления линией UARTLCR_H. В режиме включенного FIFO бит установлен, когда буфер FIFO приемника пуст, а в режиме выключенного FIFO бит установлен когда входной буферный регистр пуст.
[3]	BUSY	ЧТ	UART занят. Бит установлен в 1 в состоянии, когда контроллер передает в линию данные. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Этот бит устанавливается в 1 сразу как только данные попадают в буфер FIFO передатчика, вне зависимости от состояния приемопередатчика, даже если он выключен (бит UARTEN регистра UARTCR).
[2]	-	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[1]	-	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[0]	CTS	ЧТ	Clear to Send. Инверсия состояния входного сигнала nUARTCTS

5.14.3.4 Регистр целой части коэффициента деления частоты UARTIBRD

Формат регистра UARTIBRD представлен в таблице ниже.

Таблица 5.110 – Формат регистра UARTIBRD

Биты	Название	Тип	Назначение
[15:0]	BAUDDIV_INT	ЧТ/ЗП	Целая часть коэффициента деления опорной частоты для установки скорости работы интерфейса UART. После сброса устанавливается в 0

5.14.3.5 Регистр дробной части коэффициента деления частоты UARTFBRD

Формат регистра UARTFBRD представлен в таблице ниже.

Таблица 5.111 – Формат регистра UARTFBRD

Биты	Название	Тип	Назначение
[5:0]	BAUDDIV_FRAC	ЧТ/ЗП	Дробная часть коэффициента деления частоты для установки скорости работы интерфейса UART. После сброса устанавливается в 0

Примечание. Формулы перевода значений, записываемых в регистры UARTIBRD и UARTFBRD, в реальную скорость работы в Бод см. в разделе 5.14.2.1. Содержимое регистров UARTIBRD и UARTFBRD обновляется только после завершения передачи/приема текущего символа.

5.14.3.6 Регистр управления линией UARTLCR_H

Данный регистр обеспечивает доступ к разрядам с 29 по 22 виртуального регистра управления линией UARTLCR. При сбросе все биты регистра UARTLCR_H обнуляются.

Формат регистра UARTLCR_H показан в таблице ниже.

Таблица 5.112 – Формат регистра UARTLCR_H

Биты	Название	Тип	Назначение
[15:8]	-	-	Зарезервировано. Не модифицировать. При чтении выдаются нули.
[7]	SPS	ЧТ/ЗП	Передача бита четности с фиксированным значением. 0 – запрещена; 1 – на месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. (При EPS=0 на месте бита четности передается 1, при EPS=1 – передается 0). Значение бита SPS не играет роли в случае, если формирование и проверка бита четности запрещена битом PEN.
[6:5]	WLEN	ЧТ/ЗП	Длина слова(символа) – количество передаваемых или принимаемых информационных бит в кадре: b11 – 8 бит, b10 – 7 бит, b01 – 6 бит, b00 – 5 бит
[4]	FEN	ЧТ/ЗП	Управление режимом работы буферов FIFO приемника и передатчика. 0 – режим FIFO выключен, входное и выходное FIFO становятся однобайтовыми буферными регистрами. 1 – режим FIFO включен, полный объем FIFO используется
[3]	STP2	ЧТ/ЗП	Режим передачи двух стоповых бит. 0 – один стоповый бит, 1 – два стоповых бита. Приемник не проверяет наличие дополнительного стопового бита в кадре
[2]	EPS	ЧТ/ЗП	Настройка типа четности при передаче/приеме кадров: 0 – odd parity, бит четности дополняет количество единиц в информационной части кадра до нечетного 1 – even parity, бит четности дополняет количество единиц в информационной части кадра до четного Значение этого бита не играет роли в случае, если бит PEN отключает контроль четности.
[1]	PEN	ЧТ/ЗП	Разрешение контроля четност. 0 – бит четности не добавляется в кадр и не проверяется 1 – бит четности добавляется в передаваемый кадр и проверяется в принимаемом кадре
[0]	BRK	ЧТ/ЗП	Выдача сигнала разрыва 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 ПО должно установить данный бит как минимум на время передачи двух полных кадров. В нормальном режиме функционирования бит должен быть установлен в 0.

Содержимое регистров UARTLCR_H, UARTIBRD и UARTFBRD совместно образует виртуальный 30-разрядный регистр управления линией UARTLCR, который обновляется по стробу, формируемому при записи в регистр UARTLCR_H.

ВНИМАНИЕ! Таким образом, для того, чтобы изменение параметров коэффициента деления частоты (регистров UARTIBRD и/или UARTFBRD) вступило в силу, после их изменения необходимо осуществить запись данных в регистр UARTLCR_H.

Примеры:

- Изменение значений трех регистров можно осуществить корректно двумя способами:
  - Запись UARTIBRD, запись UARTFBRD, запись UARTLCR_H;
  - Запись UARTFBRD, запись UARTIBRD, запись UARTLCR_H;
- Для того чтобы изменить значение лишь одного из регистров (UARTIBRD или UARTFBRD) необходимо выполнить следующие шаги:
  - Запись UARTIBRD (или UARTFBRD), запись UARTLCR_H.

В таблице ниже приведена таблица истинности для бит управления контролем четности SPS, EPS, PEN регистра управления линией UARTLCR_H.

Таблица 5.113 –Управление режимом контроля четности

PEN	EPS	SPS	Бит контроля четности
0	x	x	Не передается, не проверяется
1	1	0	Проверка четности символа (even parity)
1	0	0	Проверка нечетности символа (odd parity)
1	0	1	Бит четности постоянно равен 1
1	1	1	Бит четности постоянно равен 0

Примечания:

- Регистры UARTLCR_H, UARTIBRD, и UARTFBRD не должны изменяться:
  - При разрешенной работе приемопередатчика (бит UARTEN регистра UARTCR равен 1);
  - Во время завершения приема или передачи символа 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 ПО после остановки приемопередатчика при наличии данных в буферах FIFO, повторно перевело приемопередатчик в разрешенное состояние.

5.14.3.7 Регистр управления UARTCR

После сброса все биты регистра управления, за исключением битов 9 и 8 устанавливаются в нулевое состояние, биты 9 и 8 устанавливаются в единичное состояние.

Формат регистра UARTCR показан в таблице ниже.

Таблица 5.114 – Формат регистра UARTCR

Биты	Название	Тип	Назначение
[15]	CTSEn	ЧТ/ЗП	Разрешение CTS (Clear to Send) управления потоком. 1 – разрешено, данные передаются в линию только при активном значении входного сигнала nUARTCTS
[14]	RTSEn	ЧТ/ЗП	Разрешение RTS (Request to Send) управления потоком. 1 – разрешено, запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере FIFO приемника
[13]	-	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[12]	-	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[11]	RTS	ЧТ/ЗП	Инверсия выходного сигнала nUARTRTS. Запись 1 выставляет сигнал в активное состояние.
[10]	-	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[9]	RXE	ЧТ/ЗП	Разрешение приема. Установка бита в 1 разрешает работу приемника. Прием данных осуществляется со внешнего входа UARTRXD. В случае перевода приемопередатчика в запрещенное состояние в ходе приема данных, он завершает прием текущего символа перед остановкой
[8]	TXE	ЧТ/ЗП	Разрешение передачи. Установка бита в 1 разрешает работу передатчика. Передача осуществляется через внешний выход UARTTXD. В случае перевода приемопередатчика в запрещенное состояние в ходе передачи данных, он завершает передачу текущего символа перед остановкой
[7]	LBE	ЧТ/ЗП	Разрешение тестирования в режиме обратной петли. В этом режиме данные с выхода UARTTXD напрямую передаются на вход UARTRXD. 1 – режим включен, 0 – режим выключен. После сброса бит устанавливается в 0.
[6:3]	-	-	Зарезервировано. Не модифицировать. При чтении выдаются нули.
[2]	-	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[1]	-	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[0]	UARTEN	ЧТ/ЗП	Разрешение работы приемопередатчика: 0 – обмен запрещен. Перед остановкой завершается прием и передача текущих символов. 1 – обмен разрешен.

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.

Рекомендуется следующая последовательность действий для программирования регистров управления:

- Остановите работу приемопередатчика.
- Дождитесь окончания приема и/или передачи текущего символа данных.
- Очистите буфер передатчика путем сброса бита FEN регистра UARTLCR_H в 0.
- Измените настройки регистра UARTCR.
- Возобновите работу приемопередатчика.

5.14.3.8 Регистр пороговых уровней заполнения FIFO UARTIFLS

Данный регистр используется для задания пороговых уровней заполнения буферов FIFO передатчика и приемника, по достижению которых генерируется сигнал прерывания UARTTXINTR или UARTRXINTR, соответственно. Прерывание генерируется в момент перехода счетчика заполнения FIFO через заданное пороговое значение.

Также задаваемые пороги участвуют в логике генерации пакетных запросов к контроллеру ПДП (см. раздел 5.14.2.7).

После сброса в регистре устанавливается пороги, соответствующие заполнению буферов на половину. Формат регистра UARTIFLS показан в таблице ниже.

Таблица 5.115 – Формат регистра UARTIFLS

Биты	Название	Тип	Назначение
[15:6]	-	-	Зарезервировано. Не модифицировать. При чтении выдаются нули
[5:3]	RXIFLSEL	ЧТ/ЗП	Порог прерывания по заполнению буфера FIFO приемника: b000 = Буфер заполнен на ≥ 1/8 b001 = Буфер заполнен на ≥ 1/4 b010 = Буфер заполнен на ≥ 1/2 b011 = Буфер заполнен на ≥ 3/4 b100 = Буфер заполнен на ≥ 7/8 b101-b111 = резерв
[2:0]	TXIFLSEL	ЧТ/ЗП	Порог прерывания по заполнению буфера передатчика: b000 = Буфер заполнен на ≤ 1/8 b001 = Буфер заполнен на ≤ 1/4 b010 = Буфер заполнен на ≤ 1/2 b011 = Буфер заполнен на ≤ 3/4 b100 = Буфер заполнен на ≤ 7/8 b101-b111 = резерв

5.14.3.9 Регистр управления маскированием прерываний UARTIMSC

При чтении выдается текущее значение маски. При записи 1 производится установка маски, а при записи 0 сброс маски на соответствующее прерывание.

После сброса все биты регистра маски устанавливаются в нулевое состояние.

Формат регистра UARTIMSC показан в таблице ниже.

Таблица 5.116 – Формат регистра UARTIMSC

Биты	Название	Тип	Назначение
[15:11]	-	-	Зарезервировано. Не модифицировать. При чтении выдаются нули
[10]	OEIM	ЧТ/ЗП	Маска прерывания по переполнению буфера UARTOEINTR. 1 – установлена 0 – сброшена
[9]	BEIM	ЧТ/ЗП	Маска прерывания по разрыву линии UARTBEINTR 1 – установлена 0 – сброшена
[8]	PEIM	ЧТ/ЗП	Маска прерывания по ошибке контроля четности UARTPEINTR 1 – установлена 0 – сброшена
[7]	FEIM	ЧТ/ЗП	Маска прерывания по ошибке в структуре кадра UARTFEINTR 1 – установлена 0 – сброшена
[6]	RTIM	ЧТ/ЗП	Маска прерывания по таймауту приема данных UARTRTINTR 1 – установлена 0 – сброшена
[5]	TXIM	ЧТ/ЗП	Маска прерывания от передатчика UARTTXINTR 1 – установлена 0 – сброшена
[4]	RXIM	ЧТ/ЗП	Маска прерывания от приемника UARTRXINTR. 1 – установлена 0 – сброшена
[3]	DSRMIM	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[2]	DCDMIM	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[1]	CTSMIM	ЧТ/ЗП	Маска прерывания UARTCTSINTR по изменению состояния линии nUARTCTS 1 – установлена 0 – сброшена
[0]	RIMIM	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.

5.14.3.10 Регистр состояния прерываний UARTRIS

Этот регистр доступен только для чтения и содержит текущее состояние прерываний без учета маскирования. Данные, записываемые в регистр, игнорируются.

Предупреждение. После сброса все биты регистра, за исключением бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение битов прерывания по состоянию модема после сброса не определено.

Формат регистра UARTRIS представлен в таблице ниже.

Таблица 5.117 – Формат регистра UARTRIS

Биты	Название	Тип	Назначение
[15:11]	-	-	Зарезервировано. Не модифицировать. При чтении выдаются нули
[10]	OERIS	ЧТ	Состояние прерывания по переполнению буфера UARTOEINTR 1 – буфер приемника переполнен; 0 – буфер приемника не переполнен.
[9]	BERIS	ЧТ	Состояние прерывания по разрыву линии UARTBEINTR 1 – произошел разрыв линии приема; 0 – разрыва линии приема не происходило.
[8]	PERIS	ЧТ	Состояние прерывания по ошибке контроля четности UARTPEINTR 1 – возникла ошибка контроля четности; 0 – ошибки контроля четности не возникало.
[7]	FERIS	ЧТ	Состояние прерывания по ошибке в структуре кадра UARTFEINTR 1 – возникла ошибка в структуре кадра; 0 – ошибки в структуре кадра не возникало.
[6]	RTRIS	ЧТ	Состояние прерывания по таймауту приема данных UARTRTINTR[1] 1 – вышло время таймаута приема данных; 0 – время таймаута приема данных не вышло.
[5]	TXRIS	ЧТ	Состояние прерывания от передатчика UARTTXINTR 1 – возникло прерывание от передатчика; 0 – прерывания от передатчика нет.
[4]	RXRIS	ЧТ	Состояние прерывания от приемника UARTRXINTR 1 – возникло прерывание от приемника; 0 – прерывания от приемника не возникало.
[3]	DSRRMIS	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[2]	DCDRMIS	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[1]	CTSRMIS	ЧТ	Состояние прерывания UARTCTSINTR по изменению линии nUARTCTS 1 – возникло прерывание; 0 – прерывания не возникало.
[0]	RIRMIS	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.

[1] - Сигнал маски прерывания по таймауту используется в качестве разрешения перехода в режим пониженного энергопотребления. Поэтому чтение состояния прерывания по таймауту из регистров UARTMIS и UARTRIS даст одинаковый результат.

5.14.3.11 Регистр маскированного состояния прерываний UARTMIS

Этот регистр доступен только для чтения и содержит текущее состояние прерываний с учетом маскирования. Данные, записываемые в регистр, игнорируются.

После сброса все биты регистра, за исключением бит прерывания по состоянию модема (биты с 3 по 0), устанавливаются в 0. Значение бит прерывания по состоянию модема после сброса не определено.

Формат регистра UARTMIS представлен в таблице ниже.

Таблица 5.118 – Формат регистра UARTMIS

Биты	Название	Тип	Назначение
[15:11]	-	-	Зарезервировано. Не модифицировать. При чтении выдаются нули
[10]	OEMIS	ЧТ	Маскированное состояние прерывания по переполнению буфера UARTOEINTR
[9]	BEMIS	ЧТ	Маскированное состояние прерывания по разрыву линии UARTBEINTR
[8]	PEMIS	ЧТ	Маскированное состояние прерывания по ошибке контроля четности UARTPEINTR
[7]	FEMIS	ЧТ	Маскированное состояние прерывания по ошибке в структуре кадра UARTFEINTR
[6]	RTMIS	ЧТ	Маскированное состояние прерывания по таймауту приема данных UARTRTINTR
[5]	TXMIS	ЧТ	Маскированное состояние прерывания от передатчика UARTTXINTR
[4]	RXMIS	ЧТ	Маскированное состояние прерывания от приемника UARTRXINTR
[3]	DSRMMIS	ЧТ	Данный бит зарезервирован. Не модифицировать.
[2]	DCDMMIS	ЧТ	Данный бит зарезервирован. Не модифицировать.
[1]	CTSMMIS	ЧТ	Маскированное состояние прерывания UARTCTSINTR по изменению линии nUARTCTS
[0]	RIMMIS	ЧТ	Данный бит зарезервирован. Не модифицировать.

5.14.3.12 Регистр сброса прерываний UARTICR

Этот регистр доступен только для записи и предназначен для сброса признака прерывания по заданному событию путем записи 1 в соответствующий бит. Запись 0 игнорируется.

Формат регистра UARTICR представлен в таблице ниже.

Таблица 5.119 – Формат регистра UARTICR

Биты	Название	Тип	Назначение
[15:11]	-	-	Зарезервировано. Не модифицировать. При чтении выдаются нули
[10]	OEIC	ЗП	Сброс прерывания по переполнению буфера UARTOEINTR 1 – сброс прерывания; 0 – не влияет на состояние регистра.
[9]	BEIC	ЗП	Сброс прерывания по разрыву линии UARTBEINTR 1 – сброс прерывания; 0 – не влияет на состояние регистра.
[8]	PEIC	ЗП	Сброс прерывания по ошибке контроля четности UARTPEINTR 1 – сброс прерывания; 0 – не влияет на состояние регистра.
[7]	FEIC	ЗП	Сброс прерывания по ошибке в структуре кадра UARTFEINTR 1 – сброс прерывания; 0 – не влияет на состояние регистра.
[6]	RTIC	ЗП	Сброс прерывания по таймауту приема данных UARTRTINTR 1 – сброс прерывания; 0 – не влияет на состояние регистра.
[5]	TXIC	ЗП	Сброс прерывания от передатчика UARTTXINTR 1 – сброс прерывания; 0 – не влияет на состояние регистра.
[4]	RXIC	ЗП	Сброс прерывания от приемника UARTRXINTR 1 – сброс прерывания; 0 – не влияет на состояние регистра.
[3]	DSRMIC	ЗП	Данный бит зарезервирован. Не модифицировать.
[2]	DCDMIC	ЗП	Данный бит зарезервирован. Не модифицировать.
[1]	CTSMIC	ЗП	Сброс прерывания UARTCTSINTR по изменению линии nUARTCTS 1 – сброс прерывания; 0 – не влияет на состояние регистра.
[0]	RIMIC	ЗП	Данный бит зарезервирован. Не модифицировать.

5.14.3.13 Регистр управления каналом ПДП UARTDMACR

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Формат регистра UARTDMACR представлен в таблице ниже.

Таблица 5.120 – Формат регистра UARTDMACR

Биты	Название	Тип	Назначение
[15:13]	-	-	Зарезервировано. Не модифицировать. При чтении выдаются нули
[2]	DMAONERR	ЧТ/ЗП	Если бит установлен в 1, то в случае возникновения комбинированного прерывания по ошибке UARTEINTR блокируется выдача ПДП запросов от приемника UARTRXDMASREQ и UARTRXDMABREQ
[1]	TXDMAE	ЧТ/ЗП	Использование канала ПДП для передатчика. Если бит установлен в 1, то разрешено формирование запросов ПДП для обслуживания буфера FIFO передатчика
[0]	RXDMAE	ЧТ/ЗП	Использование канала ПДП для приемника. Если бит установлен в 1, то разрешено формирование запросов ПДП для обслуживания буфера FIFO приемника

5.14.3.14 Идентификатор периферийного устройства UARTPeriphID0-3

Регистр UARTPeriphID0-3 представляет собой четыре 8-битных регистра, предназначенных только для чтения, которые охватывают адресное пространство от 0xFE0 до 0xFEC. Регистры могут быть концептуально представлены как один 32-битный регистр.

В таблице ниже приведен формат полей этого регистра.

Таблица 5.121 – Поля идентификатора периферийного устройства. UARTPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для UART (PL011) 0x011
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока - 0x3 (r1p5)
Configuration[31:24]	Вариант конфигурации периферии (0x0)

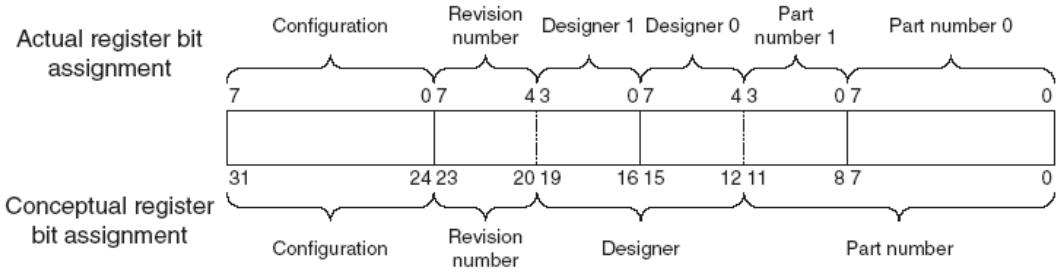


Рисунок 5.45 –Формат регистра UARTPeriphID0-3

Отметим, что периферийный блок занимает 4 Кб адресного пространства и обращение к периферийным регистрам должно быть 32-битным, независимо от реальной разрядности этих регистров.

5.14.3.15 PrimeCell идентификатор. UARTPCellID0-3

Регистры UARTPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры предназначены только для чтения и могут быть концептуально представлены как один 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. Регистр UARTPCellID0-3 установлен в значение 0xB105F00D. На рисунке ниже приведено распределение полей этого регистра.

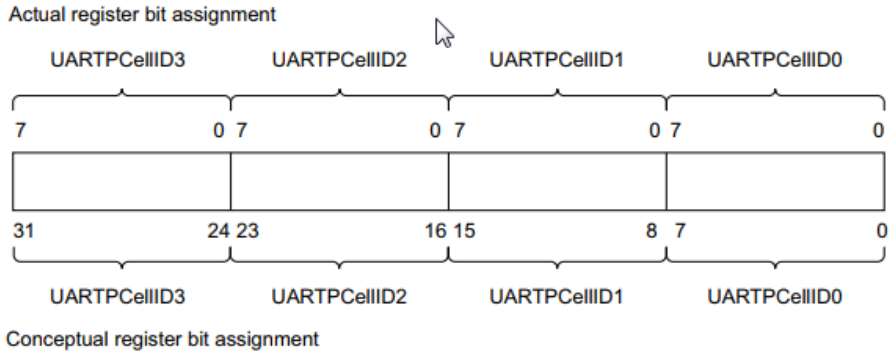


Рисунок 5.46 – Формат регистра UARTPCellID0-3

5.14.3.16 Регистр управления тестированием UARTTCR

Регистр доступен по чтению и записи. После сброса все биты регистра обнуляются.

Формат регистра UARTTCR представлен в таблице ниже.

Таблица 5.122 – Формат регистра UARTTCR

Биты	Название	Тип	Назначение
[15:3]	-	-	Зарезервировано. При чтении значение непредсказуемо.
[2]	SIRTEST	ЧТ/ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[1]	TESTFIFO	ЧТ/ЗП	Разрешение режима тестирования FIFO буферов. 1 – разрешено. Запись данных в регистр тестовых данных UARTTDR приведет к записи данных в FIFO приемника, а чтение из UARTTDR приводет к чтению данных из FIFO передатчика. 0 – запрещено. Данные не могут быть напрямую считаны/записаны из.в FIFO (режим нормального функционирования приемопередатчика).
[0]	ITEN	ЧТ/ЗП	Перевод контроллера UART в режим интеграционного тестирования. 1 – UART в режиме интеграционного тестирования 0 – UART в нормальном режиме

5.14.3.17 Входной регистр интеграционного тестирования UARTITIP

Входной регистр интеграционного тестирования UARTITIP доступен по чтению и записи. Формат регистра UARTITIP представлен в таблице ниже.

Таблица 5.123 – Формат регистра UARTITIP

Биты	Название	Тип	Назначение
[16:8]	-	-	Зарезервировано. При чтении значение непредсказуемо.
[7]	UARTTXDMACLR	ЧТ/ЗП	Операции записи в этот бит определяют значение внутреннего сигнала UARTTXDMACLR в режиме интеграционного тестирования. Операции чтения возвращают значение сигнала UARTTXDMACLR на выходе тестового мультиплексора.
[6]	UARTRXDMACLR	ЧТ/ЗП	Операции записи в этот бит определяют значение внутреннего сигнала UARTRXDMACLR в режиме интеграционного тестирования. Операции чтения возвращают значение сигнала UARTRXDMACLR на выходе тестового мультиплексора
[5]	nUARTRI	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[4]	nUARTDCD	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[3]	nUARTCTS	ЧТ	Операции чтения возвращают значение входного сигнала nUARTCTS
[2]	nUARTDSR	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[1]	SIRIN	ЧТ	Данный бит зарезервирован. Модифицировать запрещено.
[0]	UARTRXD	ЧТ	Операции чтения возвращают значение входного сигнала UARTRXD

5.14.3.1 Выходной регистр интеграционного тестирования UARTITOP

Выходной регистр интеграционного тестирования UARTITOP предназначен для контроля состояния на внутренних и внешних выводах контроллера UART в режиме интеграционного тестирования. Внешние выводы доступны только для записи, а внутренние выводы доступны для чтения и записи.

Формат регистра UARTITOP представлен в таблице ниже.

Таблица 5.124 – Формат регистра UARTITOP

Биты	Название	Тип	Назначение
[15]	UARTTXDMASREQ	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTTXDMASREQ. Операции чтения выдают значение UARTTXDMASREQ на выходе тестового мультиплексора.
[14]	UARTTXDMABREQ	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTTXDMABREQ. Операции чтения выдают значение UARTTXDMABREQ на выходе тестового мультиплексора.
[13]	UARTRXDMASREQ	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTRXDMASREQ. Операции чтения выдают значение UARTRXDMASREQ на выходе тестового мультиплексора.
[12]	UARTRXDMABREQ	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTRXDMABREQ. Операции чтения выдают значение UARTRXDMABREQ на выходе тестового мультиплексора.
[11]	UARTMSINTR	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTMSINTR. Операции чтения выдают значение UARTMSINTR на выходе тестового мультиплексора.
[10]	UARTRXINTR	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTRXINTR. Операции чтения выдают значение UARTRXINTR на выходе тестового мультиплексора.
[9]	UARTTXINTR	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTTXINTR. Операции чтения выдают значение UARTTXINTR на выходе тестового мультиплексора.
[8]	UARTRTINTR	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTRTINTR. Операции чтения выдают значение UARTRTINTR на выходе тестового мультиплексора.
[7]	UARTEINTR	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTEINTR. Операции чтения выдают значение UARTEINTR на выходе тестового мультиплексора.
[6]	UARTINTR	ЧТ/ЗП	Внутренний вывод. Операции записи определяют значение на UARTINTR. Операции чтения выдают значение UARTINTR на выходе тестового мультиплексора.
[5]	-	ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[4]	-	ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[3]	nUARTRTS	ЗП	Внешний вывод. Операции записи определяют значение на выходе nUARTRTS.
[2]	-	ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[1]	-	ЗП	Данный бит зарезервирован. Модифицировать запрещено.
[0]	UARTTXD	ЗП	Внешний вывод. Операции записи определяют значение на выходе UARTTXD.

5.14.3.2 Регистр тестовых данных UARTTDR

Регистр тестовых данных UARTTDR позволяет в режиме интеграционного тестирования записывать данные в буфер FIFO приемника и считывать данные из буфера FIFO передатчика напрямую. Эта тестовая функция активируется установкой бита TESTFIFO в регистре управления тестированием UARTTCR.

Формат регистра UARTTDR представлен в таблице ниже.

Таблица 5.125 – Формат регистра UARTTDR

Биты	Название	Тип	Назначение
[16:11]	-	-	Зарезервировано. При чтении значение непредсказуемо.
[10:0]	DATA	ЧТ/ЗП	Когда бит TESTFIFO установлен в 1, данные записываются в буфер FIFO приемника и считываются из буфера FIFO передатчика

5.14.4 Прерывания

В контроллере UART предусмотрено 8 маскируемых источников прерывания. Их комбинации формируют пять независимых выходных сигналов запроса на прерывание, а также один общий сигнал, представляющий собой комбинацию независимых по схеме ИЛИ.

Сигналы запроса на прерывание:

- UARTRXINTR – прерывание от приемника;
- UARTTXINTR – прерывание от передатчика;
- UARTRTINTR – прерывание по таймауту приемника;
- UARTMSINTR – прерывание от модемного интерфейса (не поддерживается);
- UARTEINTR – прерывание из-за ошибки передачи:
  - UARTOEINTR - переполнение буфера FIFO приемника;
  - UARTBEINTR - прерывание по разрыву линии;
  - UARTPEINTR - ошибка контроля четности;
  - UARTFEINTR - ошибка в структуре кадра.
- UARTINTR – логическое ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR.

Каждый из независимых сигналов запроса на прерывание может быть маскирован путем установки соответствующего бита в регистре управления маскированием прерываний UARTIMSC. Установка бита в 1 разрешает соответствующее прерывание, сброс в 0 – запрещает.

Прерывания от приемника UARTRXINTR и передатчика UARTTXINTR выведены отдельно от прерываний по изменению состояния устройства. Это позволяет использовать сигналы запроса UARTRXINTR и UARTTXINTR для обеспечения чтения и записи данных, согласованной с достижением заданного порога заполнения буферов FIFO приемника и передатчика.

Прерывание по обнаружению ошибке UARTEINTR формируется в случае возникновения той или иной ошибки приема данных (возможно несколько типов ошибок приема данных).

Признаки возникновения каждого из условий прерывания можно считать либо из регистра состояния прерываний UARTRIS, либо из регистра маскированного состояния прерываний UARTMIS.

ВНИМАНИЕ! Несмотря на то, что контроллер UART может формировать пять независимых выходных сигналов запроса на прерывание, к системному контроллеру прерываний подключено только одно комбинированное прерывание UARTINTR. Точную причину прерывания можно определить программно.

5.14.4.1 UARTRXINTR

Состояние прерывания от приемника может измениться в случае возникновения одного из следующих событий:

- В режиме включенных FIFO, количество данных в буфере FIFO приемника достигает заданного порогового уровня. В этом случае сигнал прерывания приемника переходит в активное состояние. Сигнал прерывания переходит в неактивное состояние после чтения данных из буфера приемника, когда количество данных станет меньше порогового уровня, либо после сброса прерывания.
- В режиме выключенных FIFO, объем FIFO приемника сокращен до одного слова (режим буферного регистра). Сигнал прерывания переходит в активное

состояние после заполнения этого буферного регистра, т.е. после приема одного символа данных. Сигнал прерывания переходит в неактивное состояние после одной операции чтения буфера приема, либо после сброса прерывания.

5.14.4.2 UARTTXINTR

Состояние прерывания от передатчика может измениться в случае возникновения одного из следующих событий:

- В режиме включенных FIFO, заполнение буфера FIFO передатчика меньше или равно заданному пороговому уровню. В этом случае сигнал прерывания передатчика переходит в активное состояние. Сигнал прерывания переходит в неактивное состояние после записи данных в буфер FIFO передатчика, когда его заполнение станет больше порогового уровня, либо после сброса прерывания.
- В режиме выключенных FIFO, объем FIFO передатчика сокращен до одного слова (режим буферного регистра). Сигнал прерывания переходит в активное состояние если этот буферный регистр пуст. Сигнал прерывания переходит в неактивное состояние после одной операции записи в буфер передатчика, либо после сброса прерывания.

Примечание. Передаваемые данные разрешено записывать в буфер FIFO передатчика либо до разрешения работы приемопередатчика и прерываний, либо после. В случае если работа приемопередатчика и прерывания от него разрешены до осуществления записи данных в буфер FIFO передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера FIFO передатчика, т.е. когда записанные данные будут переданы.

5.14.4.3 UARTRTINTR

Прерывание по таймауту приемника возникает в случае, если буфер FIFO приемника не пуст, и на вход приемника не поступало новых данных в течение периода времени, необходимого для приема 32 бит. Прерывание по таймауту снимается либо после считывания данных из буфера приемника до его опустошения, либо путем записи 1 в соответствующий бит регистра сброса прерывания UARTICR.

5.14.4.4 UARTEINTR

Прерывание по обнаружению ошибки передачи возникает в случае ошибки при приеме данных. Оно может быть вызвано рядом факторов:

- ошибка в структуре кадра;
- ошибка контроля четности;
- разрыв линии;
- переполнение приемного буфера.

Причину возникновения прерывания можно определить, прочитав содержимое регистра состояния прерываний UARTRIS, либо содержимое регистра маскированного состояния 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.

5.14.4.5 UARTINTR

Все описанные сигналы запроса на прерывание скомбинированы в общую линию путем объединения по схеме ИЛИ сигналов UARTRXINTR, UARTTXINTR, UARTRTINTR, UARTMSINTR и UARTEINTR с учетом маскирования.

ВНИМАНИЕ! В СБИС СНП ВП к системному контроллеру прерываний подключено только одно комбинированное прерывание UARTINTR. Точную причину прерывания можно определить программно.

5.15 Контроллер последовательного синхронного интерфейса SPI

Последовательный порт в микросхеме СНП-ВП имеет следующие характеристики:

- формат данных – Motorola SPI в режиме master
- поддержка полнодуплексного SPI mode 0,1,2 и 3
- программируемая частота обмена, максимальная частота обмена – FAPB/2= 40 МГц (T=25 нс)
- поддержка до 8 подключенных ведомых SPI устройств в режиме разделения времени, т.е. одновременно возможен обмен данными только с одним SPI устройством
- подключение к системной шине СБИС через AMBA APB интерфейс, используется 16 младших бит шины данных
- два раздельных 16-разрядных FIFO буфера глубиной 8 слов для приема и передачи данных
- программируемая длина SPI фрейма от 4 до 16 бит
- возможность работы по прерыванию
- поддержка DMA доступа

5.15.1 Устройство SPI порта СБИС СНП-ВП

В таблице ниже приведен список внешних выводов, относящихся к SPI порту.

Таблица 5.126 – Выводы микросхемы, входящие в состав SPI порта

Вывод	Тип буфера	Примечание
SPICLK	out	выход тактового сигнала SPI порта
SPITXD	out	выход данных SPI порта
SPIRXD	in (pullup 33K)	вход данных SPI порта
SPI_CS0 … SPI_CS7	out	выбор ведомого SPI устройства (сигнал с активным низким уровнем)

Выбор активного SPI slave устройства осуществляется путем предварительной записи бит [2:0] регистра SPICSCTRL в блоке ARMSC в соответствии с таблицей:

Таблица 5.127 – Выбор активного ведомого SPI устройства

ARMSC. SPICSCTRL[2:0]	Примечание
000	активное устройство подключено к SPI_CS0 (по умолчанию)
001	активное устройство подключено к SPI_CS1
010	активное устройство подключено к SPI_CS2
011	активное устройство подключено к SPI_CS3
100	активное устройство подключено к SPI_CS4
101	активное устройство подключено к SPI_CS5
110	активное устройство подключено к SPI_CS6
111	активное устройство подключено к SPI_CS7

На рисунке ниже приведена схема выбора активного SPI устройства:

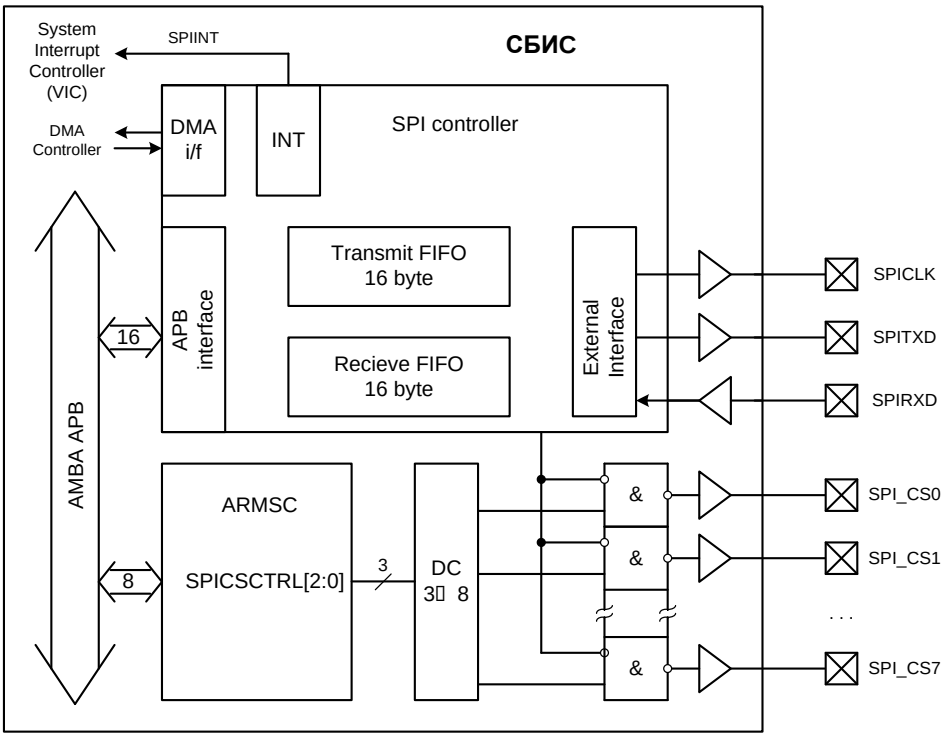


Рисунок 5.47 – Схема выбора активного SPI устройства

Контроллер SPI порта выполняет преобразование 16-разрядных данных, записываемых ядром ARM или DMA контроллером через шину APB, в последовательное однобитное представление, а также обратное преобразование. Внутри контроллера выполняется буферизация передаваемых и принимаемых данных с помощью двух раздельных 16-разрядных FIFO буферов глубиной 8 слов. Наполненность этих буферов отслеживается по прерыванию или методом программного поллинга.

Частота передачи и приема данных (F_{SPI}) программируется путем записи значений делителей системной частоты F_{APB} в поля SCR регистра SSPCR0 и поле CPSDVSR регистра SSPCPSR, по следующей формуле:

$$F_{SPI} = F_{APB}/[CPSDVR*(1+SCR)]$$

где CPSDVR = 2…254, а SCR = 0…255. Таким образом, возможна работа SPI интерфейса на частотах от 1.23 кГц до 40МГц.

Если предполагается использовать SPI flash память для начальной загрузки СБИС, то подключаемая микросхема памяти должна удовлетворять следующим условиям:

- поддержка SPI mode 3
- частота работы не менее 10 Гц, т.к. начальный делитель опорного синхросигнала, выставляемый загрузчиком, равен 8
- поддержка команды FAST_READ (код – 0x0B)
- загрузочная flash-память должна быть подключена к SPI_CS0 выводу СБИС

Подробное описание процесса начальной загрузки из SPI flash памяти см. п. 5.22.3.

5.15.2 Протокол передачи данных в различных режимах SPI порта

Работа SPI интерфейса возможна в 4 режимах, отличающихся друг от друга состоянием сигнала SPICLK в неактивном состоянии порта и активной фазой этого сигнала. Режим работы задается путем записи соответствующих значений в биты SPO и SPH регистра SSPCR0 SPI контроллера.

- бит SPO управляет состоянием выхода SPICLK в режиме ожидания. Если SPO = 0, то вывод SPICLK имеет низкий уровень при отсутствии передачи данных. Если SPO = 1, то вывод SPICLK имеет высокий уровень при отсутствии передачи данных.
- бит SPH управляет активной фазой сигнала SPICLK. Если SPH = 0, то данные защелкиваются по первому изменению (фронту или срезу) сигнала SPICLK. Если SPH = 1, то данные защелкиваются по второму изменению (фронту или срезу) сигнала SPICLK.

Ниже приведены временные диаграммы каждого из режимов.

5.15.2.1 SPI mode 0 (SPO=0, SPH=0)

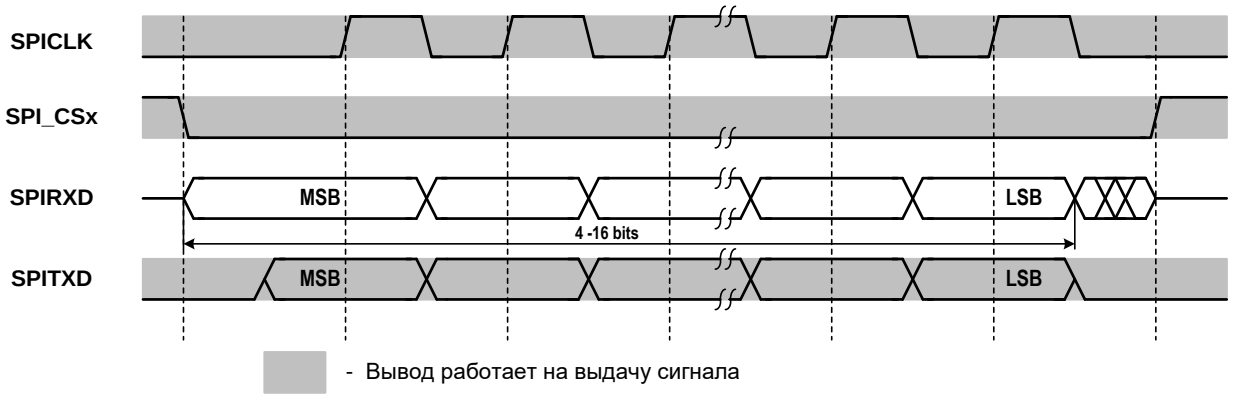


Рисунок 5.48 – SPI mode 0 (SPO=0, SPH=0) одиночная передача

Выдача данных происходит по срезу сигнала SPICLK, защелкивание входных данных – по фронту сигнала SPICLK. На рисунке ниже приведены временные диаграммы передачи, состоящей из нескольких пакетов.

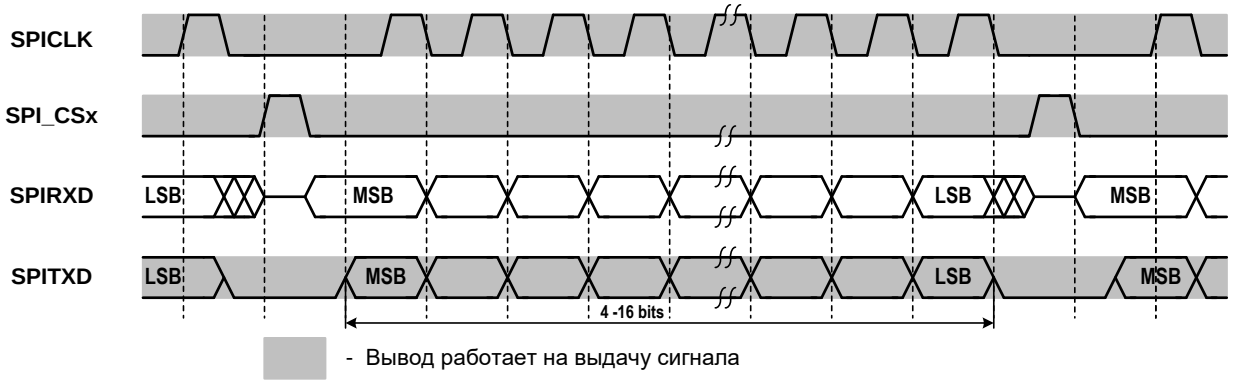


Рисунок 5.49 – SPI mode 0 (SPO=0, SPH=0) передача из нескольких пакетов

Т.е., между пакетами (словами) вывод SPI_CSx принимает значение 1 на протяжении полутакта синхросигнала SPICLK.

5.15.2.2 SPI mode 1 (SPO=0, SPH=1)

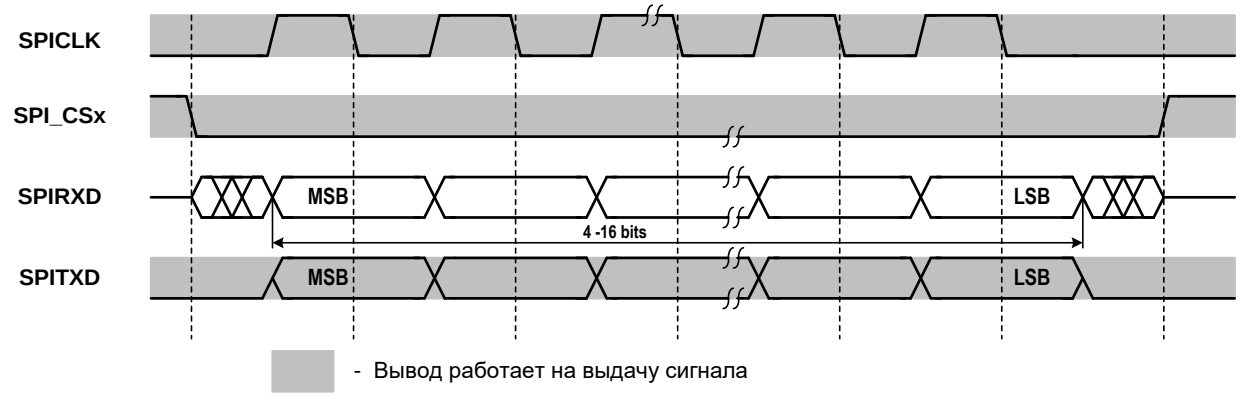


Рисунок 5.50 – SPI mode 1 (SPO=0, SPH=1) одиночная передача

Выдача данных происходит по фронту сигнала SPICLK, защелкивание входных данных – по срезу сигнала SPICLK. Сигнал SPI_CSx не меняет свое значение между словами в передаче из нескольких пакетов.

5.15.2.3 SPI mode 2 (SPO=1, SPH=0)

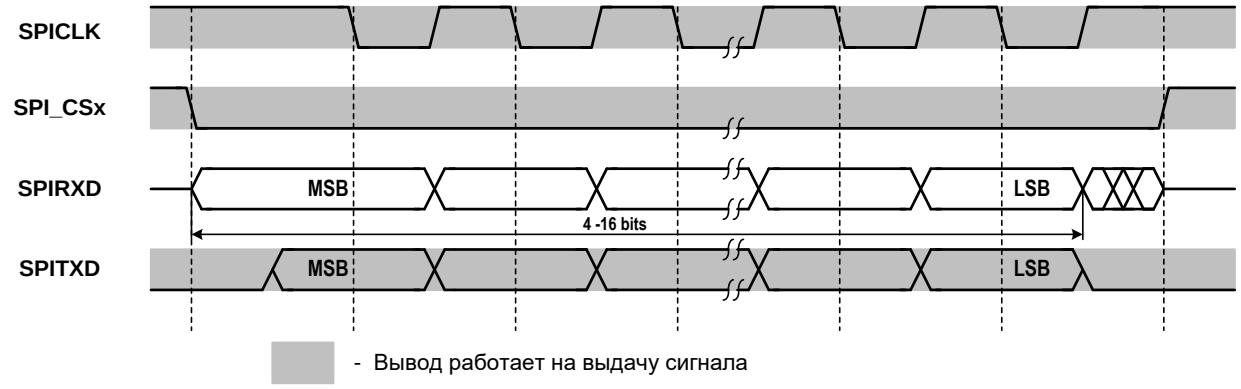


Рисунок 5.51 – SPI mode 2 (SPO=1, SPH=0) одиночная передача

Выдача данных происходит по фронту сигнала SPICLK, защелкивание входных данных – по срезу сигнала SPICLK. На рисунке ниже приведены временные диаграммы передачи, состоящей из нескольких пакетов.

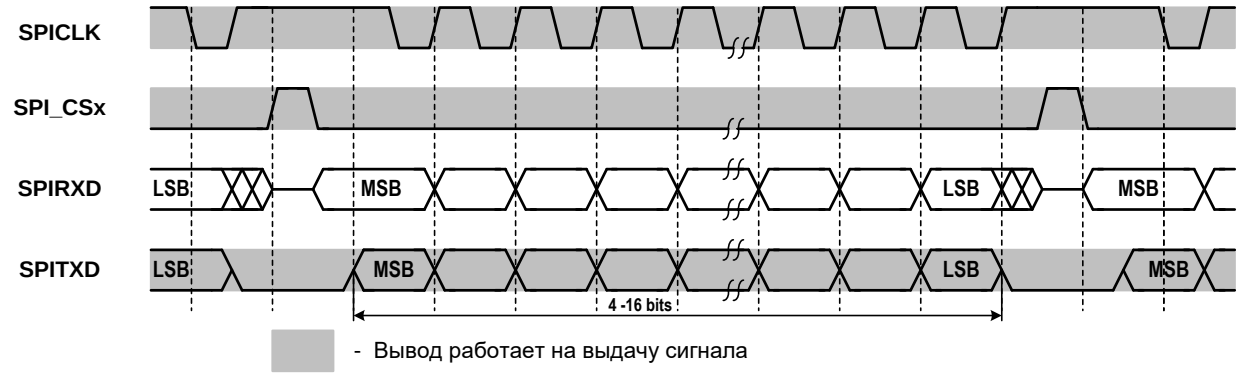


Рисунок 5.52 – SPI mode 2 (SPO=1, SPH=0) передача из нескольких пакетов

Т.е. между пакетами (словами) вывод SPI_CSx принимает значение 1 на протяжении полутакта синхросигнала SPICLK.

5.15.2.4 SPI mode 3 (SPO=1, SPH=1)

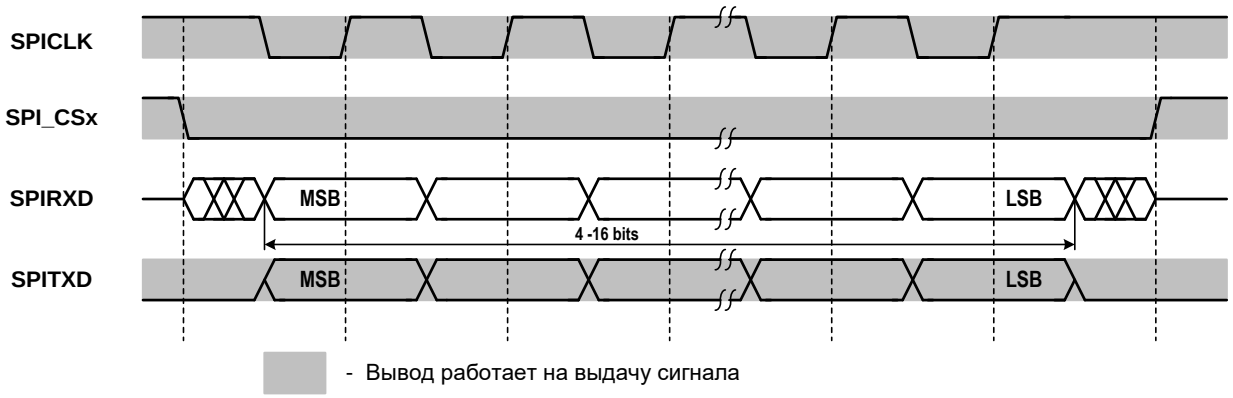


Рисунок 5.53 – SPI mode 3 (SPO=1, SPH=1) одиночная передача

Выдача данных происходит по срезу сигнала SPICLK, защелкивание входных данных – по фронту сигнала SPICLK. Сигнал SPI_CSx не меняет свое значение между словами в передаче из нескольких пакетов.

5.15.3 Регистровая модель контроллера порта SPI

Программно доступные регистры контроллера порта SPI расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение SSP Base = 0xFFF0D000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.128 – Спецификация регистров контроллера порта SPI

Адрес	Тип	Разрядность	Начальное значение	Имя	Описание
SSP Base +0x00	ЧТ/ЗП	16	0x0000	SSPCR0	Регистр управления 0
SSP Base +0x04	ЧТ/ЗП	4	0x0	SSPCR1	Регистр управления 1
SSP Base +0x08	ЧТ/ЗП	16	0x----	SSPDR	Регистр данных. SSPDR
SSP Base +0x0C	ЧТ	5	0x03	SSPSR	Регистр состояния. SSPSR
SSP Base +0x10	ЧТ/ЗП	8	0x00	SSPCPSR	Регистр множителя синхросигнала. SSPCPSR
SSP Base +0x14	ЧТ/ЗП	4	0x0	SSPIMSC	Регистр снятия или установки маски прерываний. SSPIMSC
SSP Base +0x18	ЧТ	4	0x8	SSPRIS	Регистр состояния прерывания до наложения маски. SSPRIS
SSP Base +0x1C	ЧТ	4	0x0	SSPMIS	Регистр состояния прерывания после наложения маски. SSPMIS
SSP Base +0x20	ЗП	4	0x0	SSPICR	Регистр сброса прерывания. SSPICR
SSP Base + 0x24	ЧТ/ЗП	2	0x0	SSPDMACR	Регистр управления DMA. SSPDMACR
SSP Base + 0x28 to 0xFDC	-	-	-	-	Зарезервировано
SSP base + 0xFEO	ЧТ	8	0x22	SSPPeriphID0	Идентификатор периферийного устройства. SSPPeriphID0 биты [7:0]
SSP base + 0xFE4	ЧТ	8	0x10	SSPPeriphID1	Идентификатор периферийного устройства. SSPPeriphID1 биты [15:8]
SSP base + 0xFE8	ЧТ	8	0x04	SSPPeriphID2	Идентификатор периферийного устройства. SSPPeriphID2 биты [23:16]
SSP base + 0xFEC	ЧТ	8	0x00	SSPPeriphID3	Идентификатор периферийного устройства. SSPPeriphID3 биты [31:24]
SSP base + 0xFF0	ЧТ	8	0x0D	SSPPCellID0	PrimeCell идентификатор. SSPPCellID0 биты [7:0]
SSP base + 0xFF4	ЧТ	8	0xF0	SSPPCellID1	PrimeCell идентификатор. SSPPCellID1 биты [15:8]
SSP base + 0xFF8	ЧТ	8	0x05	SSPPCellID2	PrimeCell идентификатор. SSPPCellID2 биты [23:16]
SSP base + 0xFFC	ЧТ	8	0xB1	SSPPCellID3	PrimeCell идентификатор. SSPPCellID3 биты [31:24]

Далее приведено подробное описание каждого регистра.

5.15.3.1 Регистр управления. SSPCR0

SSPCR0 – управляющий регистр, содержащий пять битовых полей, которые управляют различными функциями порта SSP. В таблице ниже приведены формат регистра SSPCR0.

Таблица 5.129 – Формат регистра SSPCR0

Биты	Название	Тип	Выполняемая функция
[15:8]	SCR	ЧТ/ЗП	Serial Clock Rate. Значение SCR используется для определения скорости приема и передачи блока SSP. Скорость вычисляется по следующей формуле: $\frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$, где CPSDVSR – значение от 2 до 254, программируемая регистром SSPCPSR. SCR может принимать значение от 0 до 255.
[7]	SPH	ЧТ/ЗП	SSPCLKOUT Phase. Фаза сигнала SSPCLKOUT
[6]	SPO	ЧТ/ЗП	SSPCLKOUT Polarity. Полярность SSPCLKOUT
[5:4]	FRF	ЧТ/ЗП	Frame Format. Формат данных: 00 – Motorola SPI формат данных 01 – не используется 10 – не используется 11 – не используется
[3:0]	DSS	ЧТ/ЗП	Data Size Select. Выбор размера данных: 0000 – зарезервировано 0001 – зарезервировано 0010 – зарезервировано 0011 – 4-битные данные 0100 – 5-битные данные 0101 – 6-битные данные 0110 – 7-битные данные 0111 – 8-битные данные 1000 – 9-битные данные 1001 – 10-битные данные 1010 – 11-битные данные 1011 – 12-битные данные 1100 – 13-битные данные 1101 – 14-битные данные 1110 – 15-битные данные 1111 – 16-битные данные

5.15.3.2 Регистр управления. SSPCR1

SSPCR1 – управляющий регистр, содержащий четыре битовых поля, которые управляют различными функциями порта SSP. В таблице ниже приведен формат регистра SSPCR1.

Таблица 5.130 – Формат регистра SSPCR1

Биты	Название	Тип	Выполняемая функция
[15:4]	-	-	Зарезервировано, при записи должен записываться нуль
[3]	SOD	ЧТ/ЗП	Slave-mode output disable. Бит используется только при работе SSP в режиме приема.
[2]	MS	ЧТ/ЗП	Master or Slave mode Select. Поддерживается работа только в режиме передачи (режим Master). Этот бит должен быть всегда равен 0 (значение по-умолчанию).
[1]	SSE	ЧТ/ЗП	Synchronous Serial Port Enable. Управление портом: 0 – порт неактивен 1 – порт активен
[0]	LBM	ЧТ/ЗП	Loop Back Mode. Управление тестовым кольцевым режимом: 0 – нормальный режим функционирования порта 1 – выходы передающего сдвигового регистра соединены с входами принимающего сдвигового регистра

5.15.3.3 Регистр данных. SSPDR

SSPDR – 16-битный регистр данных. Чтение из этого регистра возвращает слово данных из приемного FIFO, на которое в данный момент указывает счетчик чтения. Данные попадают в приемное FIFO автоматически из блока принимающей логики после приема полного SPI фрейма.

Запись в SSPDR приводит к записи данных в передающее FIFO, откуда автоматически перемещаются в передающий сдвиговый регистр. Далее данные последовательно сдвигаются наружу кристалла через вывод SPITXD, в соответствии с запрограммированной скоростью работы порта.

Когда разрядность передаваемых данных меньше, чем 16 бит, программист должен выравнивать данные по правому краю слова. Принимаемые данные выравниваются автоматически в принимающем буфере.

Таблица 5.131 – Формат регистра SSPDR

Биты	Название	Тип	Выполняемая функция
[15:0]	DATA	ЧТ/ЗП	Принимающее/передающее FIFO: Чтение – принимающего FIFO Запись – передающего FIFO Если разрядность данных меньше 16 бит, то данные должны быть выровнены по правому краю слова. Старшие неиспользуемые биты игнорируются автоматически. В режиме приема происходит автоматическое выравнивание по правому краю слова.

5.15.3.4 Регистр состояния. SSPSR

SSPSR – регистр, доступный только по чтению, в котором хранится состояние полноты FIFO и состояние занятости порта PrimeCell SSP. В таблице ниже приведен формат регистра SSPSR.

Таблица 5.132 – Формат регистра SSPSR

Биты	Название	Тип	Выполняемая функция
[15:5]	-	-	Зарезервирован, должен быть прописан нуль
[4]	BSY	ЧТ	Флаг занятости порта SSP: 0 = SSP простаивает 1 = SSP передает и/или принимает пакет данных
[3]	RFF	ЧТ	Сигнал полноты принимающего FIFO: 0 = принимающее FIFO не полное 1 = принимающее FIFO полное
[2]	RNE	ЧТ	Сигнал не пустоты принимающего FIFO: 0 = принимающее FIFO пустое 1 = принимающее FIFO не пустое
[1]	TNF	ЧТ	Сигнал не полноты передающего FIFO: 0 = передающее FIFO полное 1 = передающее FIFO не полное
[0]	TFE	ЧТ	Сигнал пустоты передающего FIFO: 0 = передающее FIFO не пустое 1 = передающее FIFO пустое

5.15.3.5 Регистр множителя синхросигнала. SSPCPSR

SSPCPSR- Регистр множителя синхросигнала, определяет делитель, на который входной сигнал SSPCLK должен быть поделен для дальнейшего использования. В таблице ниже приведен формат регистра SSPCPSR.

Таблица 5.133 – Формат регистра SSPCPSR

Биты	Название	Тип	Выполняемая функция
[15:8]	-	-	Зарезервирован, должен быть прописан нуль
[7:0]	CPSDVSR	ЧТ/ЗП	Clock Prescale Devisor. Значение делителя синхросигнала. Должен принимать четные значения от 2 до 254, в зависимости от частоты сигнала SSPCLK. Младшие значащие биты всегда принимают значение нуль при чтении

5.15.3.6 Регистр снятия или установки маски прерываний. SSPIMSC

SSPIMSC- регистр снятия или установки маски прерываний. Чтение регистра возвращает текущее значение маски, соответствующего прерывания. Запись 1 в соответствующий бит устанавливает маску, а запись 0 – снимает её. В таблице ниже приведен формат регистра SSPIMSC.

Таблица 5.134 – Формат регистра SSPIMSC

Биты	Название	Тип	Выполняемая функция
[15:4]	-	-	Зарезервирован, должен быть прописан нуль
[3]	TXIM	ЧТ/ЗП	Transmit Interrupt mask. Маска прерывания передающего FIFO(SSPTXINTR): 0 = Прерывание по заполнению TxFIFO более чем на половину замаскировано 1 = Прерывание по заполнению TxFIFO более чем на половину не маскировано
[2]	RXIM	ЧТ/ЗП	Receive Interrupt Mask. Маска прерывания принимающего FIFO(SSPRXINTR): 0 = Прерывание по заполнению Rx FIFO менее чем на половину замаскировано 1 = Прерывание по заполнению Rx FIFO менее чем на половину не маскировано
[1]	RTIM	ЧТ/ЗП	Receive Timeout Interrupt Mask. Маска прерывания по превышению timeout-периода чтения из приемного FIFO (SSPRTINTR). 0 = прерывание замаскировано 1 = прерывание не маскировано
[0]	RORIM	ЧТ/ЗП	Receive Overrun Interrupt Mask. Маска прерывания по попытке записи в полное Rx FIFO: 0 = прерывание замаскировано 1 = прерывание не маскировано

5.15.3.7 Регистр состояния прерывания до наложения маски. SSPRIS

SSPRIS – регистр состояния прерывания до наложения маски, программно доступен только по чтению. При чтении регистр содержит текущее состояние прерываний до наложения маски. В таблице ниже приведен формат регистра SSPIMSC.

Таблица 5.135 – Формат регистра SSPRIS

Биты	Название	Тип	Выполняемая функция
[15:4]	Зарезервировано	-	Зарезервирован, должен быть прописан нуль
[3]	TXRIS	ЧТ	Определяет состояние прерывания SSPTXINTR до наложения маски
[2]	RXRIS	ЧТ	Определяет состояние прерывания SSPRXINTR до наложения маски
[1]	RTRIS	ЧТ	Определяет состояние прерывания SSPRTINTR до наложения маски
[0]	RORRIS	ЧТ	Определяет состояние прерывания SSPRORINTR до наложения маски

5.15.3.8 Регистр состояния прерывания после наложения маски. SSPMIS

SSPMIS – Регистр состояния прерывания после наложения маски. Программно доступен только по чтению. При чтении регистр содержит текущее состояние прерываний после наложения маски. В таблице ниже приведен формат регистра SSPMIS.

Таблица 5.136 – Формат регистра SSPMIS

Биты	Название	Тип	Выполняемая функция
[15:4]	Зарезервировано	-	Зарезервирован, должен быть прописан нуль
[3]	TXMIS	ЧТ	Определяет состояние прерывания SSPTXINTR после наложения маски
[2]	RXMIS	ЧТ	Определяет состояние прерывания SSPRXINTR после наложения маски
[1]	RTMIS	ЧТ	Определяет состояние прерывания SSPRTINTR после наложения маски
[0]	RORMIS	ЧТ	Определяет состояние прерывания SSPRORINTR после наложения маски

5.15.3.9 Регистр сброса прерывания. SSPICR

SSPICR – регистр сброса прерывания, программно доступен только по записи. Запись единицы снимает соответствующее прерывание. Запись 0 не производит эффекта. В таблице ниже приведен формат регистра SSPICR.

Таблица 5.137 – Формат регистра SSPMIS

Биты	Название	Тип	Выполняемая функция
[15:2]	Зарезервировано	-	Зарезервирован, должен быть прописан нуль
[1]	RTIC	ЗП	Снимает прерывание SSPRTINTR
[0]	RORIC	ЗП	Снимает прерывание SSPRORINTR

5.15.3.10 Регистр управления DMA. SSPDMACR

SSPDMACR – регистр управления DMA, программно доступен на запись и на чтение. При сбросе все биты устанавливаются в нуль. В таблице ниже приведен формат регистра SSPDMACR.

Таблица 5.138 – Формат регистра SSPDMACR

Биты	Название	Тип	Выполняемая функция
[15:2]	Зарезервировано	-	Зарезервирован, должен быть прописан нуль
[1]	Transmit DMA Enable (TXDMAE)	ЧТ/ЗП	Если бит установлен в 1, то DMA-доступ к передающему FIFO разрешен.
[0]	Receive DMA Enable (RXDMAE)	ЧТ/ЗП	Если бит установлен в 1, то DMA-доступ к принимающему FIFO разрешен

5.15.3.11 Идентификатор периферийного устройства. SSPPeriphID0-3

Регистр SSPPeriphID0-3 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 ниже (см. Таблица 5.139) приведен формат полей этого регистра.

Таблица 5.139 – Поля идентификатора периферийного устройства SSPPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для SPI (PL022) 0x022
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 SPI (PL022) 0x0

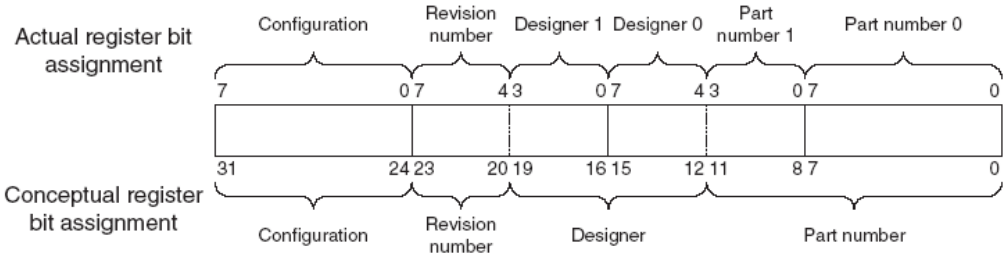


Рисунок 5.54 –Распределение полей регистра SSPPeriphID0-3

Обращение к периферийным регистрам должно быть 32-битным, независимо от их реальной разрядности.

5.15.3.12 PrimeCell идентификатор. SSPPCellID0-3

Регистры SSPPCellID0-3 – 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 SSPPCellID0-3 установлен в 0xB105F00D. На рисунке ниже приведено распределение полей этого регистра.

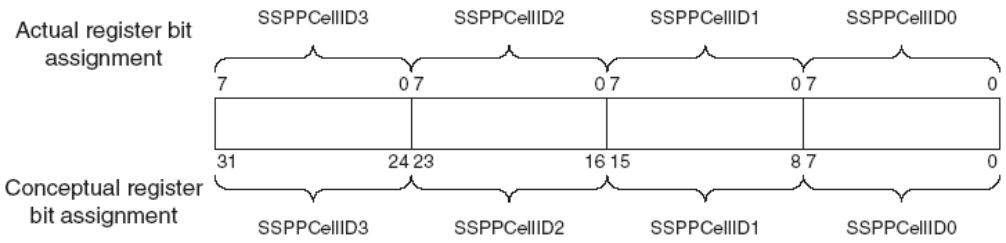


Рисунок 5.55 –Распределение полей регистра SSPPCellID0-3

5.15.4 Прерывания

Существует 5 прерываний, формируемых блоком SSP. Четыре из них являются маскируемыми с активным высоким уровнем:

- **SSPRXINTR**. Прерывание по необходимости разгрузки принимающего FIFO. Это прерывание устанавливается, когда принимающее FIFO содержит четыре или более слов данных.
- **SSPTXINTR**. Прерывание по необходимости обслуживания передающего FIFO. Это прерывание устанавливается, когда передающее FIFO содержит менее четырех слов данных. Программист может работать с передающим FIFO двумя способами:

Данные могут быть записаны в передающее FIFO до включения порта и до разрешения прерываний.

Или, наоборот, можно сначала включить порт и разрешить прерывания – тогда данные в передающее FIFO должны записываться подпрограммой обслуживания прерываний

- **SSPRORINTR**. Прерывание по переполнению принимающего FIFO. Это прерывание устанавливается, когда приемное FIFO полное и происходит прием данных, приводя к переполнению FIFO. Причем данные перезаписываются не в FIFO, а в приемном сдвиговом регистре.
- **SSPRTINTR**. Прерывание по истечении времени ожидания приема данных (timeout). Это прерывание устанавливается, когда принятые данные находятся в приемном FIFO дольше 32 тактов. Этот механизм дополнительно оповещает программиста о том, что необходимо забрать данные из приемного FIFO. Прерывание SSPRTINTR снимается, когда принимающее FIFO становится пустым, или если новые данные получены с SPI шины. Прерывание также может быть снято путем записи бита RTIC=1 в регистре SSPICR.

Каждое из четырех прерываний может быть замаскировано путем установления соответствующих бит в регистре SSPIMSC. Состояние прерывания до и после наложения маски можно узнать, прочитав SSPRIS и SSPMIS регистры.

Пятое прерывание **SSPINTR** представляет собой общее комбинированное прерывание. Выставляется только одновременно с одним из вышеописанных прерываний. На контроллер прерываний VIC заведено только это прерывание от блока SSP.

5.16 Контроллер шины USB2.0

Контроллер USB обеспечивает аппаратную поддержку обмена данными по шине USB и соответствует спецификации устройства типа USB 2.0 High-speed.

Контроллер имеет следующие характеристики:

- может работать как хост-контроллер (USB host) и как конечное устройство (USB device),
- поддержка режимов high-speed (480 Мбит/с) и full-speed (12 Мбит/с).

Особенности в режиме хост-контроллера:

- поддержка до 8 активных конечных точек,
- до 8 передач за 1 (микро)фрейм,
- до 15 подключенных устройств,
- встроенный корневой концентратор (root hub).

Особенности встроенного корневого концентратора:

- регистр состояния,
- аппаратная реализация последовательности сброса и определения скорости подключенного устройства.

Особенности в режиме конечного устройства:

- до 16 конечных точек USB (8 IN и 8 OUT),
- программная обработка стандартных запросов,
- общий размер буфера 32 кбайт.

5.16.1 Устройство контроллера USB

Контролер USB состоит из интерфейса процессорной шины, блока программно доступных регистров, блока памяти для буферов конечных точек или каналов, блока обработки протокола USB и блока физического интерфейса (PHY) (см. Рисунок 5.56).

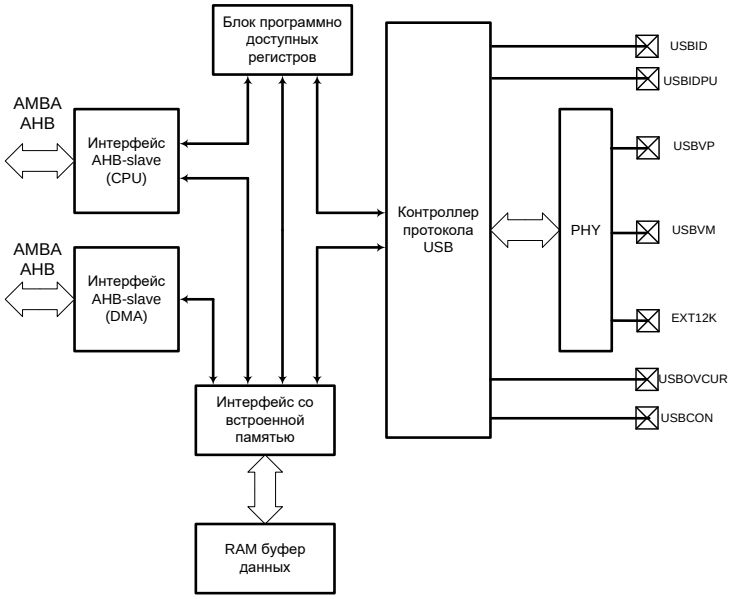


Рисунок 5.56 – Структурная схема контроллера USB

В таблице ниже (Таблица 5.140) приведен список внешних выводов, относящихся к контроллеру USB.

Таблица 5.140 – Выводы микросхемы, относящиеся к контроллеру USB

Вывод	Тип буфера	Примечание
USBCLKXI	I	Вывод опорного тактового сигнала 48 МГц
USBCLKXO	I/O	Вывод опорного тактового сигнала 48 МГц
DPPAD	I/O	Сигнал D+ шины USB
DMPAD	I/O	Сигнал D- шины USB
EXT12K	O	Вывод внешнего резистора 12 кОм
USBCON	I	Сигнал наличия питания Vbus шины USB
USBID	I	Выбор типа устройства USB: 0 – хост-контроллер (подключается к земле на разъёме mini-A), 1 – конечное устройство (остаётся неподключенным в разъёме mini-B)
USBIDPU	O	Управление подтягивающим резистором вывода USBID: 0 – подтягивающий резистор отключен, 1 – подтягивающий резистор подключен
USBOVCUR	I	Индикатор превышения тока: 0 – зафиксировано превышение тока; 1 – превышения тока нет.
USBDRVVBUS	O	Управление внешним источником питания USB: 0 – питание выключено; 1 – питание включено

Контроллер использует сигнал USBCON в режиме конечного устройства для того, чтобы определять наличие или отсутствие подключения к концентратору (hub) USB.

Вывод EXT12K используется для внутреннего источника тока и должен быть подключен к земле через резистор 12 кОм.

Вывод USBID определяет, в какой роли (хост-контроллер или конечное устройство) должен выступать данный контроллер при подключении к другому устройству USB. Если контроллер будет использоваться всегда в одной роли, то значение вывода USBID можно фиксировать (подключив его к земле или к питанию 3,3 В). Если роль устройства определяется типом коннектора (mini-AB), то для экономии потребления может использоваться схема с отключаемым подтягивающим резистором на линии USBID. В момент, когда контроллер имеет необходимость фиксировать значение на входе USBID, контроллер самостоятельно включает резистор с помощью вывода USBIDPU.

Индикатор превышения тока (USBOVCUR) используется в режиме хост-контроллера для оповещения ПО о ситуации превышения устройствами шины максимально допустимого значения силы тока.

5.16.2 Регистровая модель контроллера USB

Спецификация регистров контроллера USB представлена в таблице ниже (Таблица 5.141).

Регистры со смещением 8000h – 80FCh расположены во встроенном блоке памяти. В разных режимах (хост-контроллер или конечное устройство) спецификация этих регистров отличается. Их значение следует задавать при инициализации контроллера в соответствии с выбранной конфигурацией (режим работы, набор конечных точек и другое). При этом в зарезервированные биты следует явно прописывать значение 0.

По адресам со смещением 8100h – FFFFh расположен банк памяти, используемый для размещения в нём буферов данных конечных точек или каналов хост-контроллера. Формат банка задаётся программно с помощью регистров счётчиков конечных точек и каналов хост-контроллера.

Таблица 5.141 – Спецификация регистров контроллера USB

Адрес	Значение после инициализации	Имя	Описание
Общие регистры			
База+0000	02000002h	CONF	Регистр конфигурации контроллера
База+0004	00000000h	MODE	Регистр режима
База+0008	00000000h	INTEN	Регистр маски прерываний
База+000C	00000000h	INTS	Регистр запросов не прерывание
База+0040	00000000h	EPCMD0	Регистры команд конечных точек
База+0044	00000000h	EPCMD1	
База+0048	00000000h	EPCMD2	
База+004C	00000000h	EPCMD3	
База+0050	00000000h	EPCMD4	
База+0054	00000000h	EPCMD5	
База+0058	00000000h	EPCMD6	
База+005C	00000000h	EPCMD7	
База+0060	00000000h	EPCMD8	
База+0064	00000000h	EPCMD9	
База+0068	00000000h	EPCMD10	
База+006C	00000000h	EPCMD11	
База+0070	00000000h	EPCMD12	
База+0074	00000000h	EPCMD13	
База+0078	00000000h	EPCMD14	
База+007C	00000000h	EPCMD15	
Регистры хост-контроллера			
База+0100	00200000h	PORTSC	Регистр управления нисходящим портом
База+0104	00000000h	PORTSTSC	Регистр состояния нисходящего порта
База+0108	00000000h	HOSTEVENTSRC	Регистр состояния хост-контроллера
База+010C	00030000h	HOSTINTEN	Регистр управления прерываниями хост-контроллера
База+0110	00000000h	HCFRMIDX	Регистр счётчика фреймов
База+0114	00000000h	HCFRMINIT	Регистр длительности фрейма
База+0118	00000000h	HCCTRL	Регистр запуска хост-контроллера
База+011C	00000000h	HCSTLINK	Регистр стартового канала
Регистры конечного устройства			
База+0200	00000020h	DEVC	Регистр управления в режиме конечного устройства
База+0204	00020000h	DEVS	Регистр состояния конечного устройства
База+0208	00000000h	FADDR	Регистр адреса конечного устройства
База+020C	00000000h	TSTAMP	Регистр номера фрейма
Регистры функций OTG			
База+0300	-	OTGC	Регистр управления функциями OTG
База+0310	00000000h	OTGS	Регистр состояния функций OTG
База+0314	00000000h	OTGSTSC	Регистры изменения состояния функций OTG
База+0318	00000000h	OTGSTSFALL	
База+031C	00000000h	OTGSTSRISE	
База+0320	00000000h	OTGTC	Регистр управления таймером OTG
База+0324	00000000h	OTGT	Регистр таймера OTG
Регистры каналов ПДП			
База+0400	00000000h	DMAC1	Регистр управления каналом 1 ПДП
База+0404	00000000h	DMAS1	Регистр состояния кнала 1 ПДП
База+0408	00000000h	DMATCI1	Регистры счётчиков данных канала 1 ПДП
База+040C	00000000h	DMATC1	
База+0420	00000000h	DMAC2	Регистр управления каналом 1 ПДП
База+0424	00000000h	DMAS2	Регистр состояния кнала 1 ПДП
База+0428	00000000h	DMATCI2	Регистры счётчиков данных канала 1 ПДП
База+042C	00000000h	DMATC2	
Управление каналами в режиме хост-контроллера			

База+8000	-	HCEPCTRL1_0	Регистры управления каналами хост-контроллера
База+8008	-	HCEPCTRL1_1	
База+8010	-	HCEPCTRL1_2	
База+8018	-	HCEPCTRL1_3	
База+8020	-	HCEPCTRL1_4	
База+8028	-	HCEPCTRL1_5	
База+8030	-	HCEPCTRL1_6	
База+8038	-	HCEPCTRL1_7	
База+8004	-	HCEPCTRL2_0	
База+800C	-	HCEPCTRL2_1	
База+8014	-	HCEPCTRL2_2	
База+801C	-	HCEPCTRL2_3	
База+8024	-	HCEPCTRL2_4	
База+802C	-	HCEPCTRL2_5	
База+8034	-	HCEPCTRL2_6	
База+803C	-	HCEPCTRL2_7	
База+8040	-	HCEPCONF0	Регистры конфигурации каналов хост-контроллера
База+8044	-	HCEPCONF1	
База+8048	-	HCEPCONF2	
База+804C	-	HCEPCONF3	
База+8050	-	HCEPCONF4	
База+8054	-	HCEPCONF5	
База+8058	-	HCEPCONF6	
База+805C	-	HCEPCONF7	
База+8080	-	HCEPCOUNT0	Регистры счётчиков каналов хост-контроллера
База+8084	-	HCEPCOUNT1	
База+8088	-	HCEPCOUNT2	
База+808C	-	HCEPCOUNT3	
База+8090	-	HCEPCOUNT4	
База+8094	-	HCEPCOUNT5	
База+8098	-	HCEPCOUNT6	
База+809C	-	HCEPCOUNT7	
База+80A0	-	HCEPCOUNT8	
База+80A4	-	HCEPCOUNT9	
База+80A8	-	HCEPCOUNT10	
База+80AC	-	HCEPCOUNT11	
База+80B0	-	HCEPCOUNT12	
База+80B4	-	HCEPCOUNT13	
База+80B8	-	HCEPCOUNT14	
База+80BC	-	HCEPCOUNT15	
База+80C0	-	HCEPCOUNT16	
База+80C4	-	HCEPCOUNT17	
База+80C8	-	HCEPCOUNT18	
База+80CC	-	HCEPCOUNT19	
База+80D0	-	HCEPCOUNT20	
База+80D4	-	HCEPCOUNT21	
База+80D8	-	HCEPCOUNT22	
База+80DC	-	HCEPCOUNT23	
База+80E0	-	HCEPCOUNT24	
База+80E4	-	HCEPCOUNT25	
База+80E8	-	HCEPCOUNT26	
База+80EC	-	HCEPCOUNT27	
База+80F0	-	HCEPCOUNT28	
База+80F4	-	HCEPCOUNT29	
База+80F8	-	HCEPCOUNT30	
База+80FC	-	HCEPCOUNT31	
Управление конечными точками в режиме конечного устройства			
База+8000	-	EPCTRL0	Регистры управления конечными точками

База+8004	-	EPCTRL1	
База+8008	-	EPCTRL2	
База+800C	-	EPCTRL3	
База+8010	-	EPCTRL4	
База+8014	-	EPCTRL5	
База+8018	-	EPCTRL6	
База+801C	-	EPCTRL7	
База+8020	-	EPCTRL8	
База+8024	-	EPCTRL9	
База+8028	-	EPCTRL10	
База+802C	-	EPCTRL11	
База+8030	-	EPCTRL12	
База+8034	-	EPCTRL13	
База+8038	-	EPCTRL14	
База+803C	-	EPCTRL15	
База+8040	-	EPCONF0	Регистры конфигурации конечных точек
База+8044	-	EPCONF1	
База+8048	-	EPCONF2	
База+804C	-	EPCONF3	
База+8050	-	EPCONF4	
База+8054	-	EPCONF5	
База+8058	-	EPCONF6	
База+805C	-	EPCONF7	
База+8060	-	EPCONF8	
База+8064	-	EPCONF9	
База+8068	-	EPCONF10	
База+806C	-	EPCONF11	
База+8070	-	EPCONF12	
База+8074	-	EPCONF13	
База+8078	-	EPCONF14	
База+807C	-	EPCONF15	
База+8080	-	EPCOUNT0	Регистры счётчиков конечных точек
База+8084	-	EPCOUNT1	
База+8088	-	EPCOUNT2	
База+808C	-	EPCOUNT3	
База+8090	-	EPCOUNT4	
База+8094	-	EPCOUNT5	
База+8098	-	EPCOUNT6	
База+809C	-	EPCOUNT7	
База+80A0	-	EPCOUNT8	
База+80A4	-	EPCOUNT9	
База+80A8	-	EPCOUNT10	
База+80AC	-	EPCOUNT11	
База+80B0	-	EPCOUNT12	
База+80B4	-	EPCOUNT13	
База+80B8	-	EPCOUNT14	
База+80BC	-	EPCOUNT15	
База+80C0	-	EPCOUNT16	
База+80C4	-	EPCOUNT17	
База+80C8	-	EPCOUNT18	
База+80CC	-	EPCOUNT19	
База+80D0	-	EPCOUNT20	
База+80D4	-	EPCOUNT21	
База+80D8	-	EPCOUNT22	
База+80DC	-	EPCOUNT23	
База+80E0	-	EPCOUNT24	
База+80E4	-	EPCOUNT25	
База+80E8	-	EPCOUNT26	

База+80EC	-	EPCOUNT27	
База+80F0	-	EPCOUNT28	
База+80F4	-	EPCOUNT29	
База+80F8	-	EPCOUNT30	
База+80FC	-	EPCOUNT31	
Память данных			
База+8100 база+FFFF	-	RAM	Память буферов данных

5.16.2.1 Регистр конфигурации контроллера CONF

Формат регистра CONF приведён ниже (Таблица 5.142).

Таблица 5.142 – Формат регистра CONF

Биты	Тип	Название поля	Описание
[31:27]	-	Зарезервировано	
[26]	ЧТ	soft_reset_m	Отражение бита soft_reset
[25]	ЧТ	burst_wait_m	Отражение бита burst_wait
[24]	ЧТ	byte_order_m	Отражение бита byte_order
[23:3]	-	Зарезервировано	
[2]	ЧТ/ЗП	soft_reset	Программный сброс контроллера USB. Запись значения 1 производит программный сброс контроллера.
[1]	ЧТ/ЗП	burst_wait	Включает дополнительный такт ожидания на шине AHB: 0 – данные идут подряд, 1 – 1 такт ожидания вставляется в операциях на шине AHB.
[0]	ЧТ/ЗП	byte_order	Устанавливает порядок байтов (endianess) в 32-разрядном слове при обращении к контроллеру USB: 0 – младший байт в младшей части слова (little endian), 1 – младший байт в старшей части слова (big endian)

5.16.2.2 Регистр режима MODE

Формат регистра MODE приведён ниже (Таблица 5.143).

Таблица 5.143 – Формат регистра MODE

Биты	Тип	Название поля	Описание
[31:4]	-	Зарезервировано	
[3]	ЧТ/ЗП	dev_addr_load_mode	Используется для установки режима обновления адреса устройства USB. Значение 0 – поле func_addr регистра FADDR необходимо обновлять после завершения статусной фазы запроса SET_ADDRESS. Значение 1 – поле func_addr регистра FADDR обновляется в момент завершения статусной фазы запроса SET_ADDRESS автоматически в то значение, которое было записано заблаговременно.
[2]	ЧТ/ЗП	dev_int_mode	Управляет режимом запросов на прерывание по изменению состояния буферов данных. Для конечной точки 0: 0 – прерывание readyi_int генерируется, когда буфер опустошается после транзакции типа IN, прерывание readyo_int – когда первое данное поступает в буфер во время транзакции типа OUT 1 – прерывание readyi_int или readyo_int генерируется, когда происходит запись в бит bufwr или bufrd регистра EPCMD (в том случае, когда соответствующий буфер доступен для записи или чтения), прерывание readyi_int также генерируется в момент записи бита init. Для конечных точек 1-15: 0 – прерывание ready_int генерируется, когда буфер перестаёт быть полным (при транзакции типа IN), прерывание empty_int генерируется, когда буфер становится пустым (при транзакции типа IN), прерывание ready_int генерируется, когда первое данное поступает в буфер (при транзакции типа OUT), 1 – прерывание ready_int генерируется, когда происходит запись в бит bufwr или bufrd регистра EPCMD (в том случае, когда соответствующий буфер доступен для записи или чтения), в этом случае также запись в бит init производит генерацию прерываний empty_int и ready_int для транзакций типа IN и прерывания empty_int для транзакций типа OUT.
[1]	ЧТ/ЗП	dev_en	Запись значения 1 в бит dev_en переводит контроллер в режим конечного устройства.
[0]	ЧТ/ЗП	host_en	Запись значения 1 в бит host_en переводит контроллер в режим хост-контроллера.

Запись значений 1 одновременно в биты dev_en и host_en не допускается.

Биты dev_addr_load_mode и dev_int_mode работают только когда контроллер находится в режиме конечного устройства (dev_en=1).

5.16.2.3 Регистр маски прерываний INTEN

Формат регистра INTEN приведён ниже (Таблица 5.144).

Таблица 5.144 – Формат регистра INTEN

Биты	Тип	Название поля	Описание
[31:16]	ЧТ/ЗП	dev_ep_inten	Разрешение прерывания dev_ep_int
[15:10]	-	Зарезервировано	
[9]	ЧТ/ЗП	dma2_inten	Разрешение прерывания dma2_int
[8]	ЧТ/ЗП	dma1_inten	Разрешение прерывания dma1_int
[7:5]	-	Зарезервировано	
[4]	ЧТ/ЗП	cmd_inten	Разрешение прерывания cmd_int
[3]	ЧТ/ЗП	phy_err_inten	Разрешение прерывания phy_err_int
[2]	ЧТ/ЗП	otg_inten	Разрешение прерывания otg_int
[1]	ЧТ/ЗП	dev_inten	Разрешение прерывания dev_int
[0]	ЧТ/ЗП	host_inten	Разрешение прерывания host_int

Каждый бит регистра INTEN включает прерывание по установке соответствующего бита регистра INTS. Если разряд регистра INTEN имеет значение 1, то прерывание по установке соответствующего бита регистра INTS разрешено. Если разряд регистра INTEN имеет значение 0, то прерывание по установке соответствующего бита регистра INTS запрещено.

5.16.2.4 Регистр запросов на прерывание INTS

Формат регистра INTS приведён ниже (Таблица 5.145).

Таблица 5.145 – Формат регистра INTS

Биты	Тип	Название поля	Описание
[31:16]	ЧТ	dev_ep_int	Биты поля dev_ep_int инициируют запрос на прерывание от конечных точек контроллера. Причиной установки бита 16 (конечная точка 0) служит одно из следующих условий: готовность буфера для транзакции типа IN, - готовность буфера для транзакции типа OUT, - получение пакета PING, - выдача пакета NAK, - выдача пакета STALL.
			Причиной установки каждого из битов 17-31 (конечные точки 1-15 соответственно) служит одно из следующих условий: - готовность буфера для транзакции типа IN, - готовность буфера для транзакции типа OUT, - получение пакета PING, - выдача пакета NAK, - выдача пакета STALL. - буфер становится пустым. Поле dev_ep_int сбрасывается также при сбросе шины USB.
[15:10]	-	Зарезервировано	
[9]	ЧТ/ЗП	dma2_int	Биты dma1_int и dma2_int инициируют запросы на прерывание от каналов ПДП (1 и 2 соответственно). Причиной установки бита dma1_int или dma2_int служит одно из следующих условий: - канал ПДП завершил работу, - канал ПДП завершил работу с ошибкой, - канал ПДП был остановлен.
[8]	ЧТ/ЗП	dma1_int	
[7:5]	-	Зарезервировано	
[4]	ЧТ/ЗП	cmd_int	Бит cmd_int инициируют запрос на прерывание при завершении записи в регистр EPCMD.
[3]	ЧТ/ЗП	phy_err_int	Бит phy_err_int сигнализирует об аппаратном сбое блока физического кодирования/декодирования.
[2]	ЧТ	otg_int	Бит otg_int сигнализирует об одном из следующих событий: - состояние SE0 (низкий уровень на обеих сигнальных линиях USB) продолжительностью более 2 мс, - изменение уровня на входе USBID, - таймер OTG (регистр OTGT) досчитал до 0.
[1]	ЧТ	dev_int	Бит dev_int устанавливается при наступлении одного из следующих событий: - переход или выход из состояния пониженного энергопотребления (Suspend), - приём пакета SOF, - приём пакета SETUP, - сброс шины USB.
[0]	ЧТ	host_int	Бит host_en устанавливается при наступлении одного из следующих событий: - передача пакета SOF, - переполнение счётчика фреймов, - изменение состояния подключения, - ошибка подключения, - переход или выход из состояния пониженного энергопотребления (Suspend), - изменение уровня на входе USBOVCUR, завершение сброса шины USB, - завершена операция со всеми конечными точками.

Биты данного регистра устанавливаются аппаратно, что может вызывать запрос на прерывание к процессорной системе. Сброс битов, доступных только для чтения (dev_ep_int, otg_int, dev_int, host_int), производится также аппаратно при устранении причины установки. Сброс битов, достпных для записи (dma2_int, dma1_int, cmd_int, phy_err_int) производится программно путём записи в даный бит значения 0.

5.16.2.5 Регистры команд конечных точек EPCMD0-EPCMD15

Имеется 16 регистров EPCMD для каждой конечной точки или канала хост-контроллера.

Регистр EPCMD имеет разный формат в разных режимах контроллера. Формат регистра EPCMD для режима хост-контроллера (бит host_en регистра MODE установлен) приведён ниже (Таблица 5.146 и Таблица 5.147).

Таблица 5.146 – Формат регистра EPCMD для режима хост-контроллера

Биты	Тип	Название поля	Описание
[31]	ЧТ	epcmd_busy	Поле epcmd_busy показывает готовность регистра принять следующую запись: 0 – запись в регистр EPCMD разрешена, 1 – регистр занят. Для записи в регистр EPCMD необходимо дождаться, пока бит epcmd_busy станет равным 0, и только затем производить запись. Есть также 2 способа записи в данный регистр, в зависимости от записываемого значения бита write_en: при write_en=1 обновляются все поля регистра EPCMD, при write_en=0 обновляются только биты 0-8 и 18-22.
[30]	-	Зарезервировано	
[29:28]	ЗП	speed	Поле speed задаёт скорость канала: 00 – full speed, 01 – low speed, 10 – high speed, 11 – зарезервировано (не допускается устанавливать это значение).
[27:26]	ЗП	bnum	Поле bnum задаёт тип буферизации: 00 – одинарная, 01 – двойная, 10, 11 – зарезервировано (не допускается устанавливать это значение).
[25]	ЗП	sc	Поле sc задаёт тип пакета SPLIT (если используется): значение 0 – Start-Split, значение 1 – Complete-Split.
[24:23]	ЗП	et	Поле et задаёт тип транзакции USB: 00 – Control, 01 – Isochronous, 10 – Bulk, 11 – Interrupt.
[22:20]	-	Зарезервировано	
[19]	ЗП	status_clr	Запись значения 1 в бит status_clr сбрасывает поле status в регистре HCEPCTRL1
[18]	ЗП	errcnt_clr	Запись значения 1 в бит errcnt_clr сбрасывает поле errcnt в регистре HCEPCTRL1.
[17:16]	ЗП	sendpid	Поле sendpid устанавливает тип передачи: 00 – OUT, 01 – IN, 10 – SETUP, 11 – PING.
[15]	ЗП	nlinkinvalid	Бит nlinkinvalid устанавливает значение бита nlinkinvalid регистра HCEPCTRL1.
[14:12]	ЗП	nextlink	Поле nextlink устанавливает значение поля nextlink регистра HCEPCTRL1.
[11]	ЗП	write_en	0: разрешена запись полей регистра EPCMD[22:18] и EPCMD[8:0] 1: разрешена запись полей регистра EPCMD[29:0]
[10:9]	-	Зарезервировано	
[8]	ЗП	toggle_clr	Запись значения 1 в бит toggle_clr сбрасывает секвенсор пакетов данных в канале (бит toggle регистра HCEPCTRL1 в значение 0).
[7]	ЗП	toggle_set	Запись значения 1 в бит toggle_set устанавливает секвенсор пакетов данных в канале (бит toggle регистра HCEPCTRL1 в значение 1).
[6:5]	-	Зарезервировано	
[4]	ЗП	bufrd	Запись значения 1 в бит bufrd даёт сигнал о завершении чтения из буфера канала.
[3]	ЗП	bufwr	Запись значения 1 в бит bufwr даёт сигнал о завершении записи в буфер канала.

[2]	ЗП	init	Запись значения 1 в бит init инициализирует буфер канала.
[1]	ЗП	stop	Запись значения 1 в бит stop останавливает канал (устанавливает trans_en=0 в регистре HCEPCTRL1).
[0]	ЗП	start	Запись значения 1 в бит start запускает канал (устанавливает trans_en=1 в регистре HCEPCTRL1).

Таблица 5.147 – Формат регистра EPCMD для режима конечного устройства

Биты	Тип	Название поля	Описание
[31]	ЧТ	epcmd_busy	Поле epcmd_busy показывает готовность регистра принять запись: 0 – запись в регистр EPCMD разрешена, 1 – регистр занят. Для записи в регистр EPCMD необходимо дождаться, пока бит epcmd_busy станет равным 0, и только затем производить запись. Есть также 2 способа записи в данный регистр, в зависимости от записываемого значения бита write_en: при write_en=1 обновляются все поля регистра EPCMD, при write_en=0 обновляются только биты 0-8 и 18-22.
[30]	-	Зарезервировано	
[29:28]	ЗП	hiband	Поле hiband задаёт количество пакетов передаваемое или получаемое за один (микро)фрейм: 00 – 1, 01 – 1, 10 – 2, 11 – 3.
[27:26]	ЗП	bnum	Поле bnum задаёт тип буферизации: 00 – одинарная, 01 – двойная, 10, 11 – зарезервировано (не допускается устанавливать это значение).
[25]	ЗП	dir	Поле dir задаёт направление транзакции: 0 – OUT, 1 – IN.
[24:23]	ЗП	et	Поле et задаёт тип транзакции: 00 – зарезервировано (не допускается устанавливать данное значение), 01 – Isochronous, 10 – Bulk, 11 – Interrupt.
[22]	ЗП	nack_int_clr	Запись значения 1 в бит nack_int_clr снимает бит nack_int регистра EPCTRL (снимает причину прерывания по отправке пакета NAK).
[21]	ЗП	stalled_int_clr	Запись значения 1 в бит stalled_int_clr снимает бит stalled_int регистра EPCTRL (снимает причину прерывания по отправке пакета STALL).
[20]	ЗП	ping_int_clr	Запись значения 1 в бит ping_int_clr снимает бит ping_int регистра EPCTRL (снимает причину прерывания по приёму пакета PING).
[19]	ЗП	readyo_int_clr/empty_int_clr	Запись значения 1 в бит readyo_int_clr/empty_int_clr снимает бит readyo_int/empty_int регистра EPCTRL0/EPCTRL1-EPCTRL15 (снимает причину прерывания по готовности буфера конечной точки 0/1-15).
[18]	ЗП	readyi_int_clr/ready_int_clr	Запись значения 1 в бит readyi_int_clr/ready_int_clr снимает бит readyi_int/ready_int регистра EPCTRL0/EPCTRL1-EPCTRL15 (снимает причину прерывания по готовности буфера конечной точки 0/1-15).
[17]	-	Зарезервировано	
[16]	ЗП	nack_inten	Бит nack_inten включает прерывание по передаче пакета NAK: 1 – включено, 0 – выключено.
[15]	ЗП	stalled_inten	Бит stalled_inten включает прерывание по передаче пакета STALL: 1 – включено, 0 – выключено.
[14]	ЗП	ping_inten	Бит ping_inten включает прерывание по приёму пакета PING: 1 – включено, 0 – выключено.
[13]	ЗП	readyo_int_en/empty_inten	Бит readyo_int_en/empty_inten включает прерывание по установке бита readyo_int/empty_int регистра EPCTRL0/EPCTRL1-EPCTRL15.
[12]	ЗП	readyi_int_en/ready_inten	Бит readyi_int_en/ready_int_en включает прерывание по установке бита readyi_int/ready_int регистра EPCTRL0/EPCTRL1-EPCTRL15.
[11]	ЗП	write_en	

[10]	ЗП	nackresp	Запись значения 1 в бит nackresp инициирует выдачу ответа NAK на транзакцию Bulk IN и OUT.
[9]	ЗП	nullresp	Запись значения 1 в бит nullresp инициирует выдачу пустого пакета в ответ на транзакцию Bulk или Interrupt IN.
[8]	ЗП	toggle_clr	Запись значения 1 в бит toggle_clr сбрасывает секвенсор пакетов данных конечной точки (бит toggle регистра EPCTRL в значение 0).
[7]	ЗП	toggle_set	Запись значения 1 в бит toggle_set устанавливает секвенсор пакетов данных конечной точки (бит toggle регистра EPCTRL в значение 1).
[6]	ЗП	stall_clr	Запись значения 1 в бит stall_clr сбрасывает бит stall регистра EPCTRL.
[5]	ЗП	stall_set	Запись значения 1 в бит stall_set устанавливает сбит stall регистра EPCTRL.
[4]	ЗП	bufrd	Запись значения 1 в бит bufrd даёт сигнал о завершении чтения из буфера конечной точки.
[3]	ЗП	bufwr	Запись значения 1 в бит bufwr даёт сигнал о завершении записи в буфер конечной точки.
[2]	ЗП	init	Запись значения 1 в бит init инициализирует буфер конечной точки.
[1]	ЗП	stop	Запись значения 1 в бит stop останавливает конечную точку (устанавливает ep_en=0 в регистре EPCTRL).
[0]	ЗП	start	Запись значения 1 в бит start запускает конечную точку (устанавливает ep_en=1 в регистре EPCTRL).

5.16.2.6 Регистр управления нисходящим портом PORTSC

Формат регистра PORTSC приведён ниже (Таблица 5.148).

Таблица 5.148 – Формат регистра PORTSC

Биты	Тип	Название поля	Описание
[31]	ЧТ/ЗП	port_resume_req	Запись значения 1 в бит port_resume_req инициирует возвращение порта в активное состояние (Resume).
[30]	ЧТ/ЗП	port_suspend_req	Запись значения 1 в бит port_suspend_req инициирует переход порта в состояние пониженного энергопотребления (Suspend). Переход осуществляется в конце текущего фрейма.
[29]	ЧТ/ЗП	port_wakeup_req	Бит port_wakeup_req включает возможность внешнего устройства инициировать выход из режима пониженного энергопотребления (функция Remote Wakeup): 0 – выключено, 1 – включено (Remote Wakeup используется).
[28:27]	-	Зарезервировано	
[26]	ЧТ/ЗП	port_enable_req	Запись значения 1 в бит port_enable_req включает порт.
[25]	ЧТ/ЗП	port_reset_req	Запись значения 1 в бит port_reset_req инициирует сброс порта.
[24]	-	Зарезервировано	
[23]	ЧТ/ЗП	forcefs_req	Бит forcefs_req отключает механизм автоопределения high-speed устройств (chirp-последовательность): 0 – устройство пытается подключиться как high-speed, 1 – устройство остаётся full-speed.
[22]	ЧТ/ЗП	port_power_req	Бит port_power_req управляет подачей питания на шину Vbus порта (непосредственно управляет выводом USBDRVVBUS): 0 – питание отключено, 1 – питание включено.
[21]	ЧТ/ЗП	port_power_ctl_req	Бит port_power_ctl_req устанавливает способ отключения питания при повышенном потреблении: значение 1 – с помощью данного контроллера (сигналом USBDRVVBUS), значение 0 – с помощью внешнего предохранителя (в этом случае контроллер не использует вывод USBDRVVBUS).
[20:12]	-	Зарезервировано	
[11]	ЧТ	port_resuming_rhs	Бит port_resuming_rhs имеет значение 1 в момент возвращения из режима пониженного энергопотребления, значение 0 – в остальное время.
[10]	ЧТ	port_suspended_rhs	Бит port_suspended_rhs показывает, находится ли контроллер в режиме пониженного энергопотребления: значение 1 – находится (Sespend), 0 – в остальное время.
[9]	ЧТ	port_wakeup_rhs	Бит port_wakeup_rhs отображает значение бита port_wakeup_req.
[8]	ЧТ	port_high_speed_rhs	Бит port_high_speed_rhs показывает, подключено ли к порту устройство типа high-speed: значение 1 – подключенное устройство прошло процесс автоопределения скорости high-speed, 0 – устройство не прошло процесс автоопределения скорости high-speed.
[7]	ЧТ	port_low_speed_rhs	Бит port_low_speed_rhs показывает, подключено ли к порту устройство типа low-speed: значение 1 – к порту подключено устройство типа low-speed, 0 – в других случаях.
[6]	ЧТ	port_enable_rhs	Бит port_enable_rhs показывает, находится ли порт во включенном состоянии: значение 1 – порт включен и прошёл процедуру сброса, 0 – в других случаях.
[5]	ЧТ	port_reset_rhs	Бит port_reset_rhs показывает, находится ли порт в

			состоянии сброса: значение 1 – порт сбрасывается в данный момент, 0 – в других случаях.
[4]	ЧТ	port_connection_rhs	Бит port_connection_rhs показывает наличие подключения к порту: значение 1 – устройство (независимо от типа) подключено к порту, 0 – нет подключенного устройства.
[3]	ЧТ	forcefs_rhs	Бит forcefs_rhs отображает значение бита forcefs_req.
[2]	ЧТ	port_power_rhs	Бит port_power_rhs отображает значение бита port_power_req.
[1]	ЧТ	port_power_ctl_rhs	Бит port_power_ctl_rhs отображает значение бита port_power_ctl_req.
[0]	ЧТ	port_over_current_rhs	Отображает состояние вывода USBOVCUR

5.16.2.7 Регистр состояния нисходящего порта PORTSTSC

Формат регистра PORTSTSC приведён ниже (Таблица 5.149).

Таблица 5.149 – Формат регистра PORTSTSC

Биты	Тип	Название поля	Описание
[31:7]	-	Зарезервировано	
[6:5]	ЧТ	linestate	Поле linestate показывает текущее состояние выводов D+/D- (DPPAD – бит 5, DMPAD – бит 6).
[4]	ЧТ/ЗП	port_reset_c	Бит port_reset_c устанавливается в момент завершения сборса порта.
[3]	ЧТ/ЗП	port_ov_curr_c	Бит port_ov_curr_c устанавливается в момент изменения состояния вывода USBOVCUR.
[2]	ЧТ/ЗП	port_suspend_c	Бит port_suspend_c устанавливается в момент выхода из состояния пониженного энергопотребления (Suspend).
[1]	ЧТ/ЗП	port_enable_c	Бит port_enable_c устанавливается при выключении порта из-за ошибки подключения.
[0]	ЧТ/ЗП	port_connection_c	Бит port_connection_c устанавливается при подключении и отключении порта.

Биты регистра PORTSTSC[4:0] устанавливаются аппаратно и сбрасываются программно путём записи в них значения 0.

5.16.2.8 Регистр состояния хост-контроллера HOSTEVENTSRC

Формат регистра HOSTEVENTSRC приведён ниже (Таблица 5.150).

Таблица 5.150 – Формат регистра HOSTEVENTSRC

Биты	Тип	Название поля	Описание
[31:16]	-	Зарезервировано	
[15:8]	ЧТ/ЗП	trans_done	Поле trans_done показывает завершение работы каналов хост-контроллера. Бит поля устанавливается автоматически при завершении передачи соответствующим каналом хост-контроллера (когда сбрасывается бит trans_en регистра HCEPCTRL1). Сброс битов trans_done производится программно путём записи в них значения 0.
[2]	ЧТ	dport_evt	Бит dport_evt объединяет по ИЛИ биты port_reset_c, port_ov_curr_c, port_suspend_c, port_enable_c, port_connection_c.
[1]	ЧТ/ЗП	frameov	Бит frameov устанавливается в момент переполнения счётчика фреймов, сброс бит производится программно путём записи в него значения 0.
[0]	ЧТ/ЗП	sofstart	Бит sofstart устанавливается автоматически через заданное число микрофреймов (в поле sofevtinterval регистра HOSTINTEN).

5.16.2.9 Регистр управления прерываниями хост-контроллера HOSTINTEN

Формат регистра HOSTINTEN приведён ниже (Таблица 5.151).

Таблица 5.151 – Формат регистра HOSTINTEN

Биты	Тип	Название поля	
[31:19]	-	Зарезервировано	
[18:16]	ЧТ/ЗП	sofevtinterval	Поле sofevtinterval задаёт период, с которым устанавливается значение 1 в бите sofstart регистра HOSTEVENTSRC: • 000 – 1 микрофрейм (125 мкс), • 001 – 2 микрофрейма (250 мкс), • 010 – 4 микрофрейма (500 мкс), • 011 – 8 микрофреймов (1 мс, 1 фрейм), • 100 – 16 микрофреймов (2 мс), • 101 – 32 микрофрейма (4 мс), • 110 – 64 микрофрейма (8 мс), • 111 – 128 микрофреймов (16 мс).
[15:8]	ЧТ/ЗП	trans_done_inten	Разрешение прерывания по установке бита trans_done
[7]	-	Зарезервировано	
[6]	ЧТ/ЗП	port_reset_c_inten	Разрешение прерывания по установке бита port_reset_c
[5]	ЧТ/ЗП	port_ov_curr_c_inten	Разрешение прерывания по установке бита port_ov_curr_c
[4]	ЧТ/ЗП	port_suspend_c_inten	Разрешение прерывания по установке бита port_suspend_c
[3]	ЧТ/ЗП	port_enable_c_inten	Разрешение прерывания по установке бита port_enable_c
[2]	ЧТ/ЗП	port_connection_c_inten	Разрешение прерывания по установке бита port_connection_c
[1]	ЧТ/ЗП	frameov_inten	Разрешение прерывания по установке бита frameov
[0]	ЧТ/ЗП	sofstart_inten	Разрешение прерывания по установке бита sofstart

Биты 0-15 регистра HOSTINTEN включают прерывание по установке соответствующих битов регистра HOSTEVENTSRC и PORTSTSC.

Если разряд регистра HOSTINTEN имеет значение 1, то прерывание по установке соответствующего бита регистра HOSTEVENTSRC или PORTSTSC разрешено.

Если разряд регистра HOSTINTEN имеет значение 0, то прерывание по установке соответствующего бита регистра HOSTEVENTSRC или PORTSTSC запрещено.

5.16.2.10 Регистр счётчика фреймов HCFRMIDX

Формат регистра HCFRMIDX приведён ниже (Таблица 5.152).

Таблица 5.152 – Формат регистра HCFRMIDX

Биты	Тип	Название поля	Описание
[31:27]	-	Зарезервировано	
[26:16]	ЧТ	sofv	Поле sofv – старшие 11 разрядов счётчика frmidx и используется как номер фрейма в пакетах SOF
[15:14]	-	Зарезервировано	
[13:0]	ЧТ/ЗП	frmidx	Поле frmidx считает микрофреймы – инкрементируется каждые 125 мкс.

5.16.2.11 Регистр длительности фрейма HCFRMINIT

Формат регистра HCFRMINIT приведён ниже (Таблица 5.153).

Таблица 5.153 – Формат регистра HCFRMINIT

Биты	Тип	Название поля	Описание
[31:16]	ЧТ	frmcnt	Поле frmcnt показывает текущее значение обратного счётчика, отсчитывающего интервал микрофрейма.
[15:13]	-	Зарезервировано	
[12:0]	ЧТ/ЗП	frminit	Поле frminit задаёт интервал микрофрейма в периодах тактового сигнала 60 МГц. Значение данного поля после сброса – 7499 (1D4Bh) – соответствует стандортному значению 125 мкс. Изменять значение данного поля может понадобиться только в целях отладки.

5.16.2.12 Регистр запуска хост-контроллера HCCTRL

Формат регистра HCCTRL приведён ниже (Таблица 5.154).

Таблица 5.154 – Формат регистра HCCTRL

Биты	Тип	Название поля	Описание
[31:1]	-	Зарезервировано	
[0]	ЧТ/ЗП	hcrun	Бит hcrun производит запуск хост-контроллера: значение 1 – контроллер запущен, 0 – контроллер приостановлен.

5.16.2.13 Регистр стартового канала HCSTLINK

Формат регистра HCSTLINK приведён ниже (Таблица 5.155).

Таблица 5.155 – Формат регистра HCSTLINK

Биты	Тип	Название поля	Описание
[31:3]	-	Зарезервировано	
[2:0]	ЧТ/ЗП	startlink	Поле startlink задаёт номер канала (номер регистра HCEPCTRL1), с которого хост-контроллер начинает работу.

5.16.2.14 Регистр управления в режиме конечного устройства DEVC

Формат регистра DEVC приведён ниже (Таблица 5.156).

Таблица 5.156 – Формат регистра DEVC

Биты	Тип	Название поля	Описание
[31]	ЧТ/ЗП	status_ng_inten	Биты *_inten включают генерацию запросов на прерывание по установке соответствующих битов *_int регистра DEVS.
[30]	ЧТ/ЗП	status_ok_inten	
[29]	ЧТ/ЗП	usbrstb_inten	
[28]	ЧТ/ЗП	usbrste_inten	
[27]	ЧТ/ЗП	setup_inten	
[26]	ЧТ/ЗП	sof_inten	
[25]	ЧТ/ЗП	suspendb_inten	
[24]	ЧТ/ЗП	suspende_inten	
[23:17]	-	Зарезервировано	
[16]	ЧТ/ЗП	physusp	Бит physusp переводит блок физического интерфейса в состояние пониженного энергопотребления.
[15:6]	-	Зарезервировано	
[5]	ЧТ/ЗП	disconnect	Бит disconnect производит отключение устройства от порта USB (электрическим способом, снимая воздействия на порт концентратора).
[4]	-	Зарезервировано	
[3]	ЧТ/ЗП	enrmtwkup	Бит enrmtwkup разрешает режим Remote Wakeup.
[2]	ЧТ/ЗП	reqresume	Запись значения 1 в бит reqresume инициирует запрос Remote Wakeup.
[1]	-	Зарезервировано	
[0]	ЧТ/ЗП	reqspeed	Бит reqspeed устанавливает режим физического интерфейса: значение 0 – high-speed, значение 1 – full-speed.

5.16.2.15 Регистр состояния конечного устройства DEVS

Формат регистра DEVS приведён ниже (Таблица 5.157).

Таблица 5.157 – Формат регистра DEVS

Биты	Тип	Название поля	Описание
[31]	ЧТ/ЗП	status_ng_int	Бит status_ng_int устанавливается в момент ошибки управляющих транзакций USB. Отслеживаются следующие ситуации: • выдача пакета STALL в ходе фазы данных или статуса (Data Stage, Status Stage), • получение пакета SETUP до завершения предыдущей фазы статуса, • непустой пакет в фазе статуса, • таймаут, • неверное направление передач или неверный секвенсор данных в управляющей транзакции.
[30]	ЧТ/ЗП	status_ok_int	Бит status_ok_int устанавливается в момент корректного завершения фазы статуса управляющей транзакции USB.
[29]	ЧТ/ЗП	usbrstb_int	Бит usbrstb_int устанавливается в начале процедуры сброса USB.
[28]	ЧТ/ЗП	usbrste_int	Бит usbrste_int устанавливается в конце процедуры сброса USB.
[27]	ЧТ/ЗП	setup_int	Бит setup_int устанавливается в момент начала управляющей фазы (Setup Stage) управляющей транзакции USB.
[26]	ЧТ/ЗП	sof_int	Бит sof_int устанавливается в момент приёма пакета SOF или, если контроллер не принял пакет SOF, когда контроллер определяет начало (микро)фрейма с помощью собственного механизма.
[25]	ЧТ/ЗП	suspendb_int	Бит suspendb_int устанавливается в момент перехода в состояние пониженного энергопотребления.
[24]	ЧТ/ЗП	suspende_int	Бит suspende_int устанавливается в момент выхода из состояния пониженного энергопотребления.
[23:18]	-	Зарезервировано	
[17]	ЧТ	crtspeed	Бит crtspeed показывает текущую настройку скорости: значение 0 – high-speed, 1 – full-speed.
[16]	ЧТ	phyreset	Бит phyreset установлен во время сброса блока физического интерфейса.
[15:2]	-	Зарезервировано	
[1]	ЧТ	busreset	Бит busreset установлен во время сброса USB.
[0]	ЧТ	suspend	Бит suspend установлен на время режима пониженного энергопотребления.

Сброс битов *_int производится программно путём записи в них значения 0.

5.16.2.16 Регистр адреса конечного устройства FADDR

Формат регистра FADDR приведён ниже (Таблица 5.158).

Таблица 5.158 – Формат регистра FADDR

Биты	Тип	Название поля	Описание
[31:23]	-	Зарезервировано	
[22:16]	ЧТ	crt_func_addr	Поле crt_func_addr показывает текущий установленный адрес устройства на шине USB.
[15:9]	-	Зарезервировано	
[8]	ЧТ/ЗП	dev_configured	Бит dev_configured переводит устройство в состояние "сконфигурировано" (Configured). Это бит слудет устанавливать после обработки запроса SET_CONFIGURATION.
[7]	-	Зарезервировано	
[6:0]	ЧТ/ЗП	func_addr	Поле func_addr следует использовать для установки адреса USB устройства после обработки запроса SET_ADDRESS.

5.16.2.17 Регистр номера фрейма TSTAMP

Формат регистра TSTAMP приведён ниже (Таблица 5.159).

Таблица 5.159 – Формат регистра TSTAMP

Биты	Тип	Название поля	Описание
[31:11]	-	Зарезервировано	
[10:0]	ЧТ	timestamp	Поле timestamp показывает номер текущего фрейма (значение, принятое от хост-контроллера, или полученное с помощью собственного механизма).

5.16.2.18 Регистр управления функцими OTG OTGC

Формат регистра OTGC приведён ниже (Таблица 5.160).

Таблица 5.160 – Формат регистра OTGC

Биты	Тип	Название поля	
[31:10]	-	Зарезервировано	
[9]	ЧТ/ЗП	id_pull_up	Бит id_pull_up управляет включением подтягивающего резистора вывода USBID: значение 0 – подтягивающий резистор отключен от питания (ключ разомкнут), значение 1 – подтягивающий резистор подключен к питанию (ключ замкнут).
[8]	ЧТ/ЗП	dp_pull_down	Бит dp_pull_down управляет включением подтягивающего резистора вывода DPPAD: значение 0 – подтягивающий резистор отключен от земли (ключ разомкнут), значение 1 – подтягивающий резистор подключен к земле (ключ замкнут).
[7]	ЧТ/ЗП	dm_pull_down	Бит dm_pull_down управляет включением подтягивающего резистора вывода DMPAD: значение 0 – подтягивающий резистор отключен от земли (ключ разомкнут), значение 1 – подтягивающий резистор подключен к земле (ключ замкнут).
[6:0]	-	Зарезервировано	

5.16.2.19 Регистр состояния функций OTG OTGS

Формат регистра OTGS приведён ниже (Таблица 5.161).

Таблица 5.161 – Формат регистра OTGS

Биты	Тип	Название поля	Описание
[31:11]	-	Зарезервировано	
[10]	ЧТ	vbus_vld	Бит vbus_vld показывает подключение устройства к шине питания USB, значение соответствует выводу USBCON.
[9:7]	-	Зарезервировано	
[6]	ЧТ	id	Бит id показывает значение на выводе USBID: 0 – вывод USBID подключен к земле (подключен разъём mini-A), 1 – в остальных случаях (подтянут к питанию с помощью резистора, управляемого битом id_pull_up).
[5:1]	-	Зарезервировано	
[0]	ЧТ	otg_tmrout	Бит otg_tmrout устанавливается в момент, когда таймер OTG (управляемый регистрами OTGT и OTGTC) досчитал до нуля.

5.16.2.20 Регистры изменения состояния функций OTG OTGSTSC, OTGSTSFALL, OTGSTSRISE

Формат регистра OTGSTSC приведён ниже (Таблица 5.162).

Таблица 5.162 – Формат регистра OTGSTSC

Биты	Тип	Название поля	Описание
[31:11]	-	Зарезервировано	
[10]	ЧТ/ЗП	vbus_vld_c	Бит vbus_vld устанавливается в следующих случаях: • в момент перехода уровня USBCON из 0 в 1, если установлен бит vbus_vld_ren, • в момент перехода уровня USBCON из 1 в 0, если установлен бит vbus_vld_fen.
[9:7]	-	Зарезервировано	
[6]	ЧТ/ЗП	id_c	Бит id_c устанавливается в следующих случаях: • в момент перехода уровня USBID из 0 в 1, если установлен бит id_ren, • в момент перехода уровня USBID из 1 в 0, если установлен бит id_fen.
[5:1]	-	Зарезервировано	
[0]	ЧТ/ЗП	otg_tmrout_c	Бит otg_tmrout_c устанавливается в следующем случае: • в момент перехода бита otg_tmrout регистра OTGS из 0 в 1, если установлен бит otg_tmrout_ren,

Биты регистра OTGSTSC устанавливаются автоматически и сбрасываются программно путём записи в них значения 0.

Форматы регистров OTGSTSFALL и OTGSTSRISE приведены ниже (Таблица 5.163 и Таблица 5.164).

Таблица 5.163 – Формат регистра OTGSTSFALL

Биты	Тип	Название поля	Описание
[31:11]	-	Зарезервировано	
[10]	ЧТ/ЗП	vbus_vld_fen	Разрешение установки vbus_vld_c при изменении vbus_vld 0->1
[9:7]	-	Зарезервировано	
[6]	ЧТ/ЗП	id_fen	Разрешение установки id_c при изменении id 0->1
[5:0]	-	Зарезервировано	

Таблица 5.164 – Формат регистра OTGSTSRISE

Биты	Тип	Название поля	Описание
[31:11]	-	Зарезервировано	
[10]	ЧТ/ЗП	vbus_vld_ren	Разрешение установки vbus_vld_c при изменении vbus_vld 1->0
[9:7]	-	Зарезервировано	
[6]	ЧТ/ЗП	id_ren	Разрешение установки id_c при изменении id 1->0
[5:1]	-	Зарезервировано	
[0]	ЧТ/ЗП	otg_tmrout_ren	

5.16.2.21 Регистр управления таймером OTG OTGTC

Формат регистра OTGTC приведён ниже (Таблица 5.165).

Таблица 5.165 – Формат регистра OTGTC

Биты	Тип	Название поля	Описание
[31:16]	-	Зарезервировано	
[15:8]	ЧТ/ЗП	tmr_prescale	Поле tmr_prescale задаёт делитель частоты тактвого сигнала 60 МГц для формирования единицы отсчёта в 0,01 мс для таймера OTG. Значение после сброса – 149. Изменять данное значение может понадобиться только для отладки.
[7:2]	-	Зарезервировано	
[1]	ЧТ/ЗП	tmr_mode	Бит tmr_mode устанавливает режим работы таймера OTG: значение 0 – однократный запуск, 1 – постоянная работа.
[0]	ЧТ/ЗП	start_tmr	Бит start_tmr производит запуск таймера: значение 1 – таймер запущен, 0 – таймер остановлен.

5.16.2.22 Регистр таймера OTG OTGT

Формат регистра OTGT приведён ниже (Таблица 5.166).

Таблица 5.166 – Формат регистра OTGT

Биты	Тип	Название поля	Описание
[31:16]	-	Зарезервировано	
[15:0]	ЧТ/ЗП	tmr_init_val	Поле tmr_init_val задаёт период счёта таймера OTG в единицах отсчёта, заданных в поле tmr_prescale регистра OTGTC.

5.16.2.23 Регистры управления ПДП DMAC1, DMAC2

Контроллер содержит два канала ПДП: 1 и 2. Для каждого имеются регистры одинакового формата, представленного ниже (Таблица 5.167). Под символом X понимается 1 или 2 для соответствующего канала.

Таблица 5.167 – Формат регистров DMACX

Биты	Тип	Название поля	Описание
[31:27]	-	Зарезервировано	
[26:16]	ЧТ/ЗП	dmaX_blksize	Поле dmaX_blksize настраивает размер передачи в байтах за один запрос контроллеру ПДП.
[15:12]	-	Зарезервировано	
[11:8]	ЧТ/ЗП	dmaX_ep	Поле dmaX_ep устанавливает номер конечной точки, с которой будет работать данный канал ПДП.
[7:6]	-	Зарезервировано	
[5]	ЧТ/ЗП	dmaX_spr	Бит dmaX_spr устанавливает режим, в котором канал ПДП завершает свою работу при приёме короткого или пустого пакета (условие окончания передачи блока данных по USB).
[4]	ЧТ/ЗП	dmaX_int_empty	Бит dmaX_int_empty устанавливает, в какой момент устройство должно устанавливать статус завершения работы канала: значение 0 – в момент завершения операции с контроллером ПДП, значение 1 – в момент завершения операции с буфером конечной точки.
[3]	ЧТ/ЗП	dmaX_sendnull	Бит dmaX_sendnull устанавливает режим отправки пустого пакета в завершающей транзакции USB (работает, когда при передаче блока данных последний пакет получается максимального размера): значение 0 – пустой пакет не отправляется, значение 1 – отправляется. Обычно для завершения передачи блока данных по USB в таком случае необходимо отправлять пустой пакет.
[2]	ЧТ/ЗП	dmaX_mode	Бит dmaX_mode устанавливает режим работы с контроллером ПДП: значение 0 – контроллер USB устанавливает сигнал запроса не снимает его до получения сигнала подтверждения, значение 1 – контроллер USB устанавливает сигнал запроса на 1 такт.
[1]	ЧТ	dmaX_busy	Бит dmaX_busy показывает, что данный канал работает.
[0]	ЧТ/ЗП	dmaX_st	Запись значения 1 в бит dmaX_st производит запуск канала, запись значения 0 – остановку. Бит dmaX_st может сбрасывается автоматически, когда передача завершена или когда передача не может быть совершена (выключена соответствующая конечная точка или весь контроллер).

5.16.2.24 Регистры состояния ПДП DMAS1, DMAS2

Для каждого канала ПДП имеются регистры одинакового формата, представленного ниже (Таблица 5.168). Под символом X понимается 1 или 2 для соответствующего канала.

Таблица 5.168 – Формат регистров DMASX

Биты	Тип	Название поля	Описание
[31:2]	-	Зарезервировано	
[1]	ЧТ	dmaX_sp	Поле dmaX_sp показывает размер последнего принятого или переданного пакета: значение 0 – последний пакет был максимального размера, 1 – последний пакет был коротким или пустым.
[0]	ЧТ	dmaX_np	Поле dmaX_np показывает размер последнего принятого или переданного пакета: значение 0 – последний пакет не был пустым, 1 – последний пакет был пустым.

5.16.2.25 Регистры счётчиков данных ПДП DMATCI1, DMATCI2

Для каждого канала ПДП имеются регистры одинакового формата, представленного ниже (Таблица 5.169). Под символом X понимается 1 или 2 для соответствующего канал.

Таблица 5.169 – Формат регистров DMATCIX

Биты	Тип	Название поля	Описание
[31:0]	ЧТ/ЗП	dmaX_tci	Поле dmaX_tci задаёт общий размер данных в байтах, которые следует передать за один запуск канала ПДП.

5.16.2.26 Регистры счётчиков данных ПДП DMATC1, DMATC2

Для каждого канала ПДП имеются регистры одинакового формата, представленного ниже (Таблица 5.170). Под символом X понимается 1 или 2 для соответствующего канала.

Таблица 5.170 – Формат регистров DMATCX

Биты	Тип	Название поля	Описание
[31:0]	ЧТ/ЗП	dmaX_tc	Поле dmaX_tci задаёт общий размер переданных за запуск канала ПДП данных в байтах.

5.16.2.27 Регистр управления конечной точкой 0 EPCTRL0

Формат регистра EPCTRL0 приведён ниже (Таблица 5.171).

Таблица 5.171 – Формат регистра EPCTRL0

Биты	Тип	Название поля	Описание
[31]	-	Зарезервировано	
[30]	ЧТ	nack_int	Бит nack_int устанавливается при отправке пакета NAK. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом nack_int_clr.
[29]	ЧТ	stalled_int	Бит stalled_int устанавливается при отправке пакета STALL. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом stalled_int_clr.
[28]	ЧТ	ping_int	Бит ping_int устанавливается при получении пакета PING. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом ping_int_clr.
[27]	ЧТ	readyo_int	Бит readyo_int устанавливается в следующих случаях: • выдача подтверждения ACK в ответ на транзакцию SETUP, • бит fullo изменяет своё значение с 0 на 1. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом readyo_int_clr.
[26]	ЧТ	readyi_int	Бит readyi_int устанавливается в следующих случаях: • выдача подтверждения ACK в ответ на транзакцию SETUP, • бит emptyi изменяет своё значение с 0 на 1. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом readyi_int_clr.
[25:23]	-	Зарезервировано	
[22]	ЧТ	nack_inten	Биты nack_inten, stalled_inten, ping_inten, readyo_inten, readyi_inten разрешают запрос на прерывание по установке соответствующих битов.
[21]	ЧТ	stalled_inten	
[20]	ЧТ	ping_inten	
[19]	ЧТ	readyo_int_en	
[18]	ЧТ	readyi_int_en	
[17:14]	-	Зарезервировано	
[13]	ЧТ	toggle	Бит toggle покаывает текущее значение секвенсора.
[12]	ЧТ	stall	Бит stall показывает, находится ли конечная точка в остановленном состоянии (Stalled State).
[11]	ЧТ	fullo	Бит fullo показывает наличие данных в буфере конечной точки типа OUT.
[10]	ЧТ	emptyo	Бит emptyo показывает отсутствие данных в буфере конечной точки типа OUT.
[9]	ЧТ	fulli	Бит fulli показывает наличие данных в буфере конечной точки типа IN.
[8]	ЧТ	emptyi	Бит emptyi показывает отсутствие данных в буфере конечной точки типа IN.
[7:1]	-	Зарезервировано	
[0]	ЧТ	ep_en	Бит ep_en показывает, включена ли данная конечная точка.

5.16.2.28 Регистры управления конечными точками 1-15 EPCTRL1-EPCTRL15

Формат регистра EPCTRL приведён ниже (Таблица 5.172).

Таблица 5.172 – Формат регистра EPCTRL

Биты	Тип	Название поля	Описание
[31]	-	Зарезервировано	
[30]	ЧТ	nack_int	Бит nack_int устанавливается при отправке пакета NAK. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом nack_int_clr.
[29]	ЧТ	stalled_int	Бит stalled_int устанавливается при отправке пакета STALL. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом stalled_int_clr.
[28]	ЧТ	ping_int	Бит ping_int устанавливается при получении пакета PING. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом ping_int_clr.
[27]	ЧТ	empty_int	Бит empty_int устанавливается, когда буфер становится пустым. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом empty_int_clr.
[26]	ЧТ	ready_int	Бит ready_int устанавливается в следующих случаях: • для конечных точек типа IN – бит full изменяет своё значение с 0 на 1, • для конечных точек тип OUT – бит empty изменяет своё значение с 0 на 1. Для того чтобы сбросить данный бит необходимо произвести запись в регистр EPCMD с установленным битом ready_int_clr.
[25:23]	-	Зарезервировано	
[22]	ЧТ	nack_inten	Биты nack_inten, stalled_inten, ping_inten, empty_inten, ready_inten разрешают запрос на прерывание по установке соответствующих битов.
[21]	ЧТ	stalled_inten	
[20]	ЧТ	ping_inten	
[19]	ЧТ	empty_inten	
[18]	ЧТ	ready_inten	
[17]	ЧТ	nackresp	Бит nackresp показывает, отвечает ли данная конечная точка пакетом NAK на пакеты IN, OUT, PING.
[16]	ЧТ	nullresp	Бит nullresp показывает, отвечает ли данная конечная точка пустым пакетом на транзакции IN типа Bulk или Interrupt.
[15:14]	ЧТ	hiband	Поле hiband показывает количество пакетов, отправляемое за 1 (микро)фрейм.
[13]	ЧТ	toggle	Бит toggle покаывает текущее значение секвенсора.
[12]	ЧТ	stall	Бит stall показывает, находится ли конечная точка в остановленном состоянии (Stalled State).
[11]	ЧТ	full	Бит full показывает наличие данных в буфере конечной точки.
[10]	ЧТ	empty	Бит empty показывает отсутствие данных в буфере конечной точки.
[9:8]	ЧТ	phyptr	Поле phyptr показывает номер буфера (00 или 01), к которому в данный момент производится доступ со стороны USB.
[7:6]	ЧТ	appptr	Поле appptr показывает номер буфера (00 или 01), к которому в данный момент производится доступ со стороны процессорной системы.
[5:4]	ЧТ	bnum	Поле bnum показывает тип буферизации: 00 – одинарная, 01 – двойная, 10, 11 – зарезервировано (не допускается устанавливать это значение).
[3]	ЧТ	dir	Бит dir показывает направление передачи данных для данной конечной точки: значение 0 – OUT, значение 1 – IN.
[2:1]	ЧТ	et	Поле et показывает тип транзакции: 00 – зарезервировано

			(не допускается устанавливать данное значение), 01 – Isochronous, 10 – Bulk, 11 – Interrupt.
[0]	ЧТ	ep_en	Бит ep_en показывает, включена ли данная конечная точка.

5.16.2.29 Регистр конфигурации конечных точек EPCONF0-EPCONF15

Формат регистра EPCONF приведён ниже (Таблица 5.173).

Таблица 5.173 – Формат регистра EPCONF

Биты	Тип	Название поля	Описание
[31:29]	-	Зарезервировано	
[28:24]	ЧТ/ЗП	countidx	Поле countidx устанавливает номер (от 0 до 31) счётчика данных (регистр EPCOUNT) для данной конечной точки.
[23:13]	ЧТ/ЗП	size	Поле size устанавливает размер буфера в байтах для данной конечной точки. Максимальный размер пакета, передаваемого данной конечной точкой, равен размеру этого буфера. Допустимые значения: • в режиме full-speed для конечных точек типа Control – 8, 16, 32 или 64 байта, • в режиме full-speed для конечных точек типа Bulk – 8, 16, 32 или 64 байта, • в режиме full-speed для конечных точек типа Interrupt – от 1 до 64 байт, • в режиме full-speed для конечных точек типа Isochronous – от 1 до 1024 байт, • в режиме high-speed для конечных точек типа Control – 64 байта, • в режиме high-speed для конечных точек типа Bulk – 512 байт, • в режиме high-speed для конечных точек типа Interrupt – от 1 до 1024 байт, • в режиме high-speed для конечных точек типа Isochronous – от 1 до 1024 байт.
[12:0]	ЧТ/ЗП	base	Поле base задаёт начальный адрес буфера конечной точки во внутренней памяти контроллера. От физического адреса в байтах нужно отбросить 2 младших бита и 1 старший. Например, адрес – 8180h, поле base – 0060h; адрес – FF00h, поле base – 1FC0h.

5.16.2.30 Регистр счётчиков конечных точек EPCOUNT0-EPCOUNT31

Формат регистра EPCOUNT приведён ниже (Таблица 5.174).

Таблица 5.174 – Формат регистра EPCOUNT

Биты	Тип	Название поля	Описание
[31:27]	-	Зарезервировано	
[26:16]	ЧТ	phycnt	Поле phycnt показывает текущее состояние счётчика данных, переданных со стороны физического интерфейса. Поле обновляется вместе с отправкой подтверждения, сбрасывается в начале приёма пакета.
[15:11]	-	Зарезервировано	
[10:0]	ЧТ	appcnt	Поле appcnt показывает текущее состояние счётчика данных, записанных или прочитанных в буфер конечной точки.

5.16.2.31 Регистры управления каналами хост-контроллера HCEPCTRL1, HCEPCTRL2

Формат регистра HCEPCTRL1 приведён ниже (Таблица 5.175).

Таблица 5.175 – Формат регистра HCEPCTRL1

Биты	Тип	Название поля	Описание
[31]	ЧТ	nlinkinvalid	Поле nlinkinvalid показывает, следует ли контроллеру перейти к выполнению передачи в следующем канале. Значение 0 – запланированных передач больше нет, значение 1 – перейти к выполнению передачи канала с номером nextlink.
[30:28]	ЧТ	nextlink	Поле nextlink показывает номер канала, к выполнению передачи в котором следует приступить после выполнения передачи в данном канале.
[27:26]	ЧТ	speed	Поле speed показывает скорость канала: 00 – full speed, 01 – low speed, 10 – high speed, 11 – зарезервировано (не допускается устанавливать это значение).
[25:24]	ЧТ	sendpid	Поле sendpid показывает тип транзакции: 00 – OUT, 01 – IN, 10 – SETUP, 11 – PING. При использовании механизма PING поле автоматически меняет тип с PING на OUT и обратно в соответствии с ответом конечного устройства.
[23:22]	-	Зарезервировано	
[21:20]	ЧТ	nyetcnt	Поле nyetcnt считает количество принятых подряд пакетов NYET или MDATA в транзакциях Complete-Split. Условия инкремента: - при транзакции Complete-Split Interrupt IN принят пакет NYET или MDATA, - при транзакции Complete-Split Interrupt OUT принят пакет NYET. Условия сброса: - при транзакции Start-Split, - при приёме подтверждения ACK на транзакцию Complete-Split Interrupt OUT.
[19:16]	ЧТ	status	Поле status показывает результат транзакции. Бит 16 – Halt (ошибка более трёх раз). Бит 17 – Stall (конечное устройство ответило пакетом STALL). Бит 18 – пропущен микрофрейм (для транзакции был установлен бит fmtsel, но она попала в другой микрофрейм). Бит 19 зарезервирован.
[15:14]	ЧТ	errcnt	Поле errcnt показывает текущее состояние счётчика ошибок. При запуске передачи следует устанавливать значение 0, а когда количество ошибок превышает 3, то аппаратно снимается бит trans_en. Поле инкрементируется всегда, когда возникает необходимость повторной пересылки данных, кроме случаев, когда работает счётчик nyetcnt.
[13]	ЧТ	toggle	Бит toggle показывает текущее значение секвенсора данных.
[12]	ЧТ	sc	Поле sc задаёт тип пакета SPLIT (если используется): значение 0 – Start-Split, значение 1 – Complete-Split.
[11]	ЧТ	full	Бит full показывает, что буфер канала содержит данные.
[10]	ЧТ	empty	Бит empty показывает, что буфер канала не содержит данных.

Таблица 5.175 – Формат регистра HCEPCTRL1. Продолжение

Биты	Тип	Название поля	Описание
[9:8]	ЧТ	hcptr	Поле hcptr показывает номер буфера (00 или 01), к которому в данный момент производится доступ со стороны USB.
[7:6]	ЧТ	appptr	Поле appptr показывает номер буфера (00 или 01), к которому в данный момент производится доступ со стороны процессорной системы
[5:4]	ЧТ	bnum	Поле bnum показывает тип буферизации: 00 – одинарная, 01 – двойная, 10, 11 – зарезервировано (не допускается устанавливать это значение).
[3]	-	Зарезервировано	
[2:1]	ЧТ	et	Поле et задаёт тип транзакции USB: 00 – Control, 01 – Isochronous, 10 – Bulk, 11 – Interrupt.
[0]	ЧТ	trans_en	Поле trans_en, работает ли данный канал. Запуск и остановка осуществляется с помощью регистра EPCMD.

Формат регистра HCEPCTRL2 приведён ниже (Таблица 5.176).

Таблица 5.176 – Формат регистра HCEPCTRL2

Биты	Тип	Название поля	Описание
[31]	ЧТ/ЗП	intervalsel	Бит intervalsel задаёт единицы, в которых задан период передач типа Isochronous и Interrupt, если выбран циклический режим (fmtsel=0). Сам период задаётся в поле interval. Значение 0 бита intervalsel выбирает в качестве единицы периода один микрофрейм, значение 1 – один фрейм.
[30]	ЧТ/ЗП	fmtsel	Бит fmtsel задаёт режим запуска передач типа Isochronous и Interrupt. Значение 0 – циклический, значение 1 – по совпадению номера фрейма.
[29:20]	ЧТ/ЗП	interval	Поле interval задаёт период запуска передач типа Isochronous и Interrupt. В циклическом режиме (fmtsel=0) и при выборе единицы в один микрофрейм (intervalsel=0): 000 – передача каждый 1 микрофрейм, 001 – передача каждые 2 микрофрейма, 010 – передача каждые 4 микрофрейма, 011 – передача каждые 8 микрофреймов, В циклическом режиме (fmtsel=0) и при выборе единицы в один фрейм (intervalsel=1): 000 – передача каждый 1 фрейм, 001 – передача каждые 2 фрейма, 010 – передача каждые 4 фрейма, 011 – передача каждые 8 фреймов, 100 – передача каждые 16 фреймов, 101 – передача каждые 32 фрейма, 110 – передача каждые 64 фрейма, 111 – передача каждые 128 фреймов, В режиме по совпадению номера фрейма (fmtsel=1) и в режиме high-speed: каждый микрофрейм, в котором поле HCFRMIDX[9:0] совпадает с полем interval. В режиме по совпадению номера фрейма (fmtsel=1) и в режиме full-speed или low-speed: каждый микрофрейм, в котором поле HCFRMIDX[9:3] совпадает с полем interval[9:3].
[19:16]	ЧТ/ЗП	funcaddr	Поле funcaddr задаёт адрес конечного устройства, с которым производится транзакция. Можно задать номер от 0 до 7, так как данный хост-контроллер поддерживает только 8 устройств.
[15:12]	ЧТ/ЗП	endpnumber	Поле endpnumber задаёт адрес конечной точки, с которой производится транзакция. Значение от 0 до 15.
[11:8]	ЧТ/ЗП	hubaddr	Поле hubaddr задаёт адрес концентратора, к которому подключено устройство, с которым производится транзакция. Допустимые значения – от 0 до 15.
[7:4]	ЧТ/ЗП	hubportnumber	Поле hubportnumber задаёт номер порта концентратора, к которому подключено устройство, с которым производится транзакция. Допустимые значения – от 0 до 15.
[3:2]	-	Зарезервировано	
[1]	ЧТ/ЗП	startbit	Бит startbit устанавливет бит S в пакетах SPLIT.
[0]	ЧТ/ЗП	endbit	Бит endbit устанавливает бит E (для Start-Split) и U (для Complete-Split) в пакетах типа SPLIT.

5.16.2.32 Регистр конфигурации каналов хост-контроллера HCEPCONF0-HCEPCONF7

Формат регистра HCEPCONF приведён ниже (Таблица 5.177).

Таблица 5.177 – Формат регистра HCEPCONF

Биты	Тип	Название поля	Описание
[31:29]	-	Зарезервировано	
[28:24]	ЧТ/ЗП	countidx	Поле countidx устанавливает номер (от 0 до 31) счётчика данных (регистр HCEPCOUNT) для данного канала.
[23:13]	ЧТ/ЗП	size	Поле size устанавливает размер буфера в байтах для данного канала. Максимальный размер пакета, передаваемого данным каналом, равен размеру этого буфера. Допустимые значения: - в режиме low-speed для каналов типа Control – 8 байт, - в режиме low-speed для каналов типа Interrupt – от 1 до 8 байт, - в режиме full-speed для каналов типа Control – 8, 16, 32 или 64 байта, - в режиме full-speed для каналов типа Bulk – 8, 16, 32 или 64 байта, - в режиме full-speed для каналов типа Interrupt – от 1 до 64 байт, - в режиме full-speed для каналов типа Isochronous – от 1 до 1024 байт, - в режиме high-speed для каналов типа Control – 64 байта, - в режиме high-speed для каналов типа Bulk – 512 байт, - в режиме high-speed для каналов типа Interrupt – от 1 до 1024 байт, - в режиме high-speed для каналов типа Isochronous – от 1 до 1024 байт.
[12:0]	ЧТ/ЗП	base	Поле base задаёт начальный адрес буфера канала во внутренней памяти контроллера. От физического адреса в байтах нужно отбросить 2 младших бита и 1 старший. Например, адрес – 8180h, поле base – 0060h; адрес – FF00h, поле base – 1FC0h.

5.16.2.33 Регистр счётчиков каналов хост-контроллера HCEPCOUNT0-HCEPCOUNT31

Формат регистра HCEPCOUNT приведён ниже (Таблица 5.178).

Таблица 5.178 – Формат регистра HCEPCOUNT

Биты	Тип	Название поля	Описание
[31:30]	ЧТ	rdatapid	Поле rdatapid показывает идентификатор пакета полученного данного (последней успешной транзакции): 00 – DATA0, 01 – DATA1, 10 – DATA2, 11 – MDATA.
[29:27]	-	Зарезервировано	
[26:16]	ЧТ	hccnt	Поле hccnt показывает текущее состояние счётчика данных, переданных со стороны физического интерфейса. Поле обновляется вместе с отправкой подтверждения, сбрасывается в начале приёма пакета.
[15:11]	-	Зарезервировано	
[10:0]	ЧТ/ЗП	appcnt	Поле appcnt показывает текущее состояние счётчика данных, записанных или прочитанных в буфер канала.

5.16.3 Рекомендации по подключению интерфейса USB

Ниже на рисунке (см. Рисунок 5.57) приведены способы подключения СБИС СНП-ВП в качестве хост-контроллера, конечного устройства и универсальное подключение.

Для Формирования напряжения VBUS и контроля перегрузки по току рекомендуется использовать специализированные для этого микросхемы, например, TPS2557. Для контроля

напряжения на линии VBUS возможно использование резистивного делителя (для преобразования 5В в 3.3В).

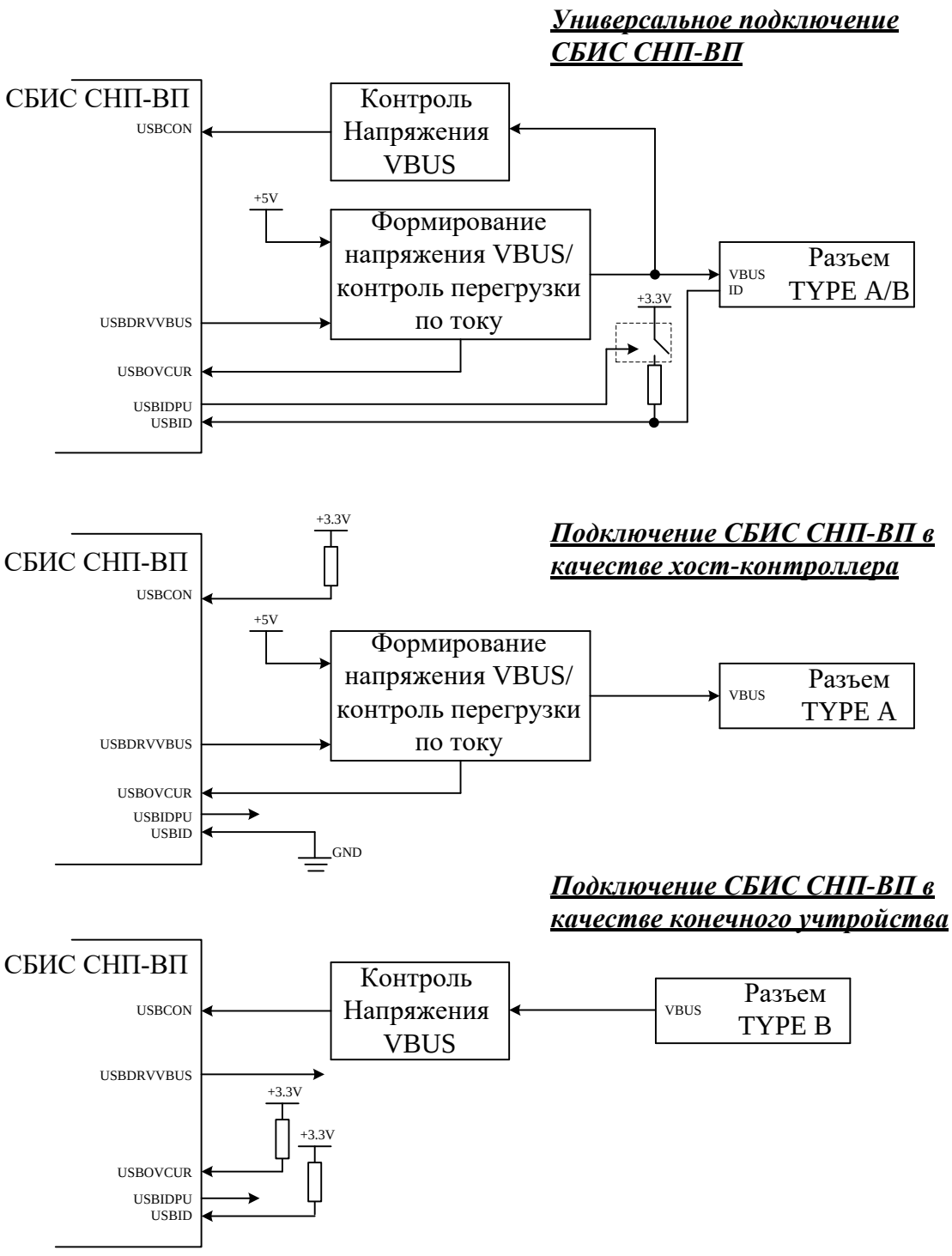


Рисунок 5.57 – Пример подключения устройств USB к СБИС СНП-ВП

Для формирования сигнала USBID возможно использование вывода 4 в разъемах USB mini или USB micro (в разъемах типа A данный вывод замкнут на землю, в разъемах типа B не задействован). Выход USBIDPU служит исключительно для экономии потребления и обеспечивает подключение резистора только на время идентификации подключения (хост-контроллер/конечное устройство).

5.17 Блок портов общего назначения GPIO

Блок портов общего назначения GPIO в СБИС СНП-ВП имеет следующие характеристики:

- 16 портов общего назначения
- подключение к системной шине СБИС через AMBA APB интерфейс, используются 16 младших бит шины данных
- максимальная частота работы Fsys/2

5.17.1 Устройство портов общего назначения

В таблице ниже приведен список внешних выводов, относящихся к портам GPIO:

Таблица 5.179 - Выводы микросхемы, входящие в состав портов общего назначения

Вывод	Тип буфера	Примечание
GPIO0 … GPIO15	inout	16 портов общего назначения

На рисунке (см. Рисунок 5.58) приведена внутренняя структура 1 бита блока портов общего назначения.

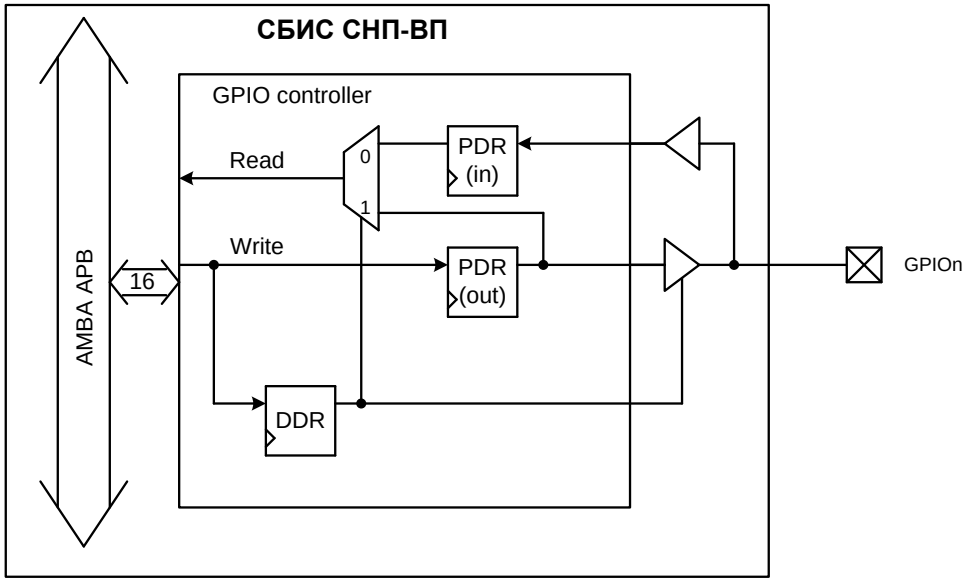


Рисунок 5.58 – Внутренняя структура бита блока портов GPIO

5.17.2 Регистровая модель контроллера портов GPIO

Программно доступные регистры контроллера портов общего назначения GPIO расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение GPIO Base = 0xFFF0A000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.180 - Спецификация регистров контроллера GPIO

Адрес	Тип	Разряд-ность	Начальное значение	Имя	Описание
GPIO Base + 0x00	ЧТ/ЗП	16	-	PDR0	Регистр порта данных
GPIO Base + 0x04 до GPIO Base + 0x0C	-	-	-	-	Зарезервировано (доступ запрещен)
GPIO Base + 0x10	ЧТ/ЗП	16	0x00	DDR0	Регистр направления передачи данных
GPIO Base + 0x14 до GPIO Base + 0xFFC	-	-	-	-	Зарезервировано (доступ запрещен)

Далее приведено подробное описание каждого регистра.

5.17.2.1 Регистр портов данных PDR

PDR – регистр для записи и чтения значений на внешних выводах GPIO. В таблице ниже приведен формат регистра PDR.

Таблица 5.181 - Формат регистра PDR

Биты	Название	Тип	Выполняемая функция
[31:16]	-	-	Зарезервировано
[15:0]	PDR	ЧТ/ЗП	Входные и выходные данные передаются через данный регистр. Биты PDR[15:0] соответствуют внешним выводам GPIO[15:0]. Направление передачи порта определяется соответствующими битами регистра DDR.

5.17.2.2 Регистр направления передачи данных DDR

DDR – регистр для задания направления передачи данных портов GPIO. В таблице ниже приведен формат регистра DDR.

Таблица 5.182 - Формат регистра DDR

Биты	Название	Тип	Выполняемые функции
[31:16]	-	-	Зарезервировано
[15:0]	DDR	ЧТ/ЗП	Задаёт направление передачи данных портов GPIO. Каждый бит регистра DDR управляет направлением, соответствующего ему GPIO порта. Биты DDR[15:0] соответствуют внешним выводам GPIO[15:0]. Хотя в начальный момент времени порты GPIO[15:0] установлены как входы, необходимо учитывать, что программа начального загрузчика, расположенного в ПЗУ имеет настройки, позволяющие ей использовать порты GPIO (см. п. 5.22.1).

5.17.3 Прерывания

Контроллер GPIO не имеет выводов прерываний.

5.18 Контроллер интерфейса CAN

Контроллер интерфейса CAN в СБИС СНП-ВП имеет следующие характеристики:

- Поддерживает CAN протокол версии 2.0 часть A и B;
- Программируемая частота передачи данных до 1МБит/с;
- Подключение к системной шине СБИС через AMBA APB интерфейс;
- Гибкость конфигурирования и возможность модернизации системы;
- Память для хранения 32 объектов сообщений (Message Objects) с их собственными:
  - 11-битными тегами идентификаторов, 29-битными тегами идентификаторов;
  - 11-битными масками идентификаторов, 29-битными масками идентификаторов;
  - 8-байтными буферами данных (статически размещенными);
  - Конфигурациями передачи (Tx), приема (Rx), буфера кадра или автоматического ответа;
  - Временными отметками
- Каждому сообщению (а не устройству) устанавливается свой приоритет;
- Каждый объект сообщений имеет свой собственный идентификатор;
- Допустимость нескольких ведущих устройств в сети;
- Непротиворечивость данных на уровне всей системы;
- Способность к обнаружению ошибок и сигнализации об их наличии;
- Автоматический повтор передачи сообщений, доставленных с ошибкой, сразу, как только сеть станет свободной;
- Режим запрета автоматического повтора передачи сообщений;
- Широковещательный прием сообщений с синхронизацией времени;
- Гарантированная величина паузы между двумя актами обмена;
- Режим программного возврата при самотестировании.

5.18.1 Устройство контроллера шины CAN

В таблице ниже приведен список внешних выводов СБИС, относящихся к контроллеру CAN:

Таблица 5.183- Выводы микросхемы, входящие в состав портов общего назначения

Вывод	Тип буфера	Примечание
CAN_TX	out	Передающий порт CAN интерфейса
CAN_RX	in	Принимающий порт CAN интерфейса

На рисунке (см. Рисунок 5.59) приведена внутренняя структура контроллера интерфейса CAN.

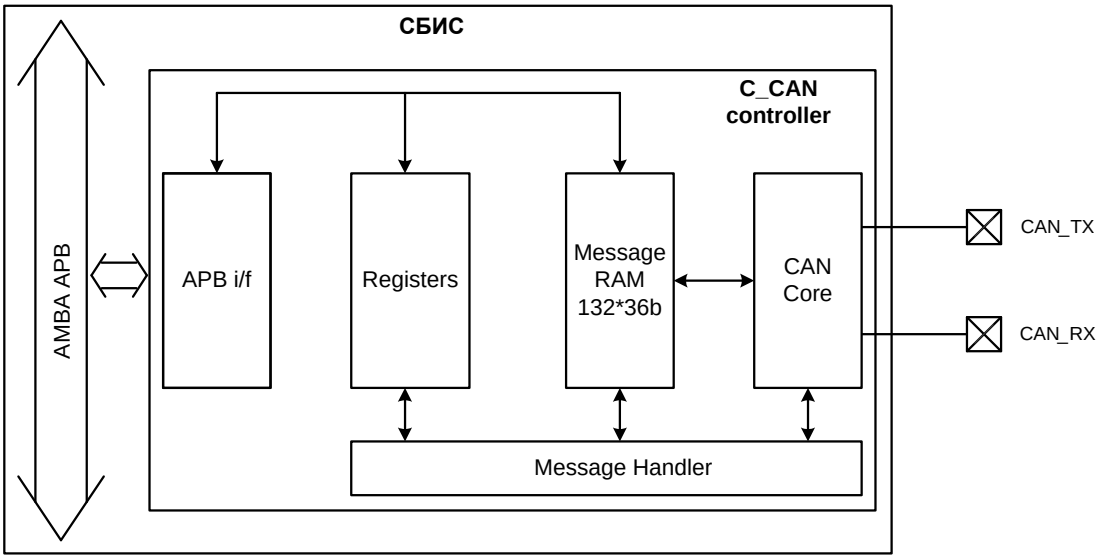


Рисунок 5.59 - Внутренняя структура контроллера шины CAN

- **CAN Core** – контроллер, реализованный в соответствии с протоколом CAN версии 2.0 A, B и стандартом ISO 11898-1, включающий блок преобразования для последовательного кода в параллельный и обратно.
- **Message RAM** – Блок памяти для хранения 32 объектов сообщений, масок идентификаторов.
- **Registers** – блок управляющих и конфигурационных регистров.
- **Message Handler** – блок управления, реализующий работу автомата состояния контролера. Автомат управляет передачей данных между блоком CAN_CORE (его приемно-передающим (Rx /Tx) сдвиговым регистром) и памятью Message RAM. Формирует программные прерывания.
- **APB Interface** – Блок для подключения к системной шине через AMBA APB интерфейс.

5.18.2 Протокол шины CAN

Протокол контроллера локальной сети является последовательным, широковещательным протоколом реального времени с очень высоким уровнем безопасности.

CAN – это международная стандартизированная последовательная шина системы (ISO 11898), обеспечивающая функциональные возможности уровня канала данных и физического канала согласно модели соединения открытых систем (ISO/OSI 7498).

Принципы:

- Информация на шине представлена в виде фиксированных сообщений различной, но ограниченной длины. Когда шина свободна, любой подключённый узел может начать передавать новое сообщение.
- Сообщение идентифицируется при помощи своего идентификатора. Идентификатор не указывает адресата сообщения, но описывает значение данных, а также определяет приоритет сообщения.
- Приоритет передаваемого сообщения сравнивается с приоритетом другого менее срочного сообщения, имеющего свой идентификатор. Приоритеты устанавливаются при проектировании системы и не могут быть изменены динамически.
- Посылая кадр удалённого запроса данных, узел может запрашивать данные у другого узла, которые тот должен послать в виде соответствующего кадра данных. Кадр данных и кадр удаленного запроса данных имеют один и тот же идентификатор.

- В системах CAN нет никакой информации относительно конфигурации системы (например, адрес узла). Это имеет несколько важных следствий:
    - Гибкость системы: Узлы могут быть добавлены к CAN-сети без изменения программного обеспечения и аппаратных средствах любого узла или прикладного уровня.
    - Групповая обработка: Любое число узлов может получать и одновременно воздействовать на одно и то же сообщение.
    - Непротиворечивость данных: Внутри CAN-сети гарантируется, что сообщение одновременно принято или всеми узлами или ни одним узлом.
- Когда шина свободна любой узел может начинать передавать сообщение. Узлу с сообщением более высокого приоритета, будет передано право на доступ к шине.
- Всякий раз, когда шина свободна, любой узел может начинать передавать сообщение. Если два или большее количество узлов начинают передавать сообщения в одно и то же время, конфликт доступа к шине решается поразрядным арбитражем, с использованием идентификатора. Механизм арбитража гарантирует, что ни информация, ни время не будут потеряны. Если кадр данных и кадр удаленного запроса данных с одинаковым идентификатором начинают передаваться одновременно, то кадр данных преобладает над кадром удаленного запроса данных. В течение арбитража каждый передатчик сравнивает уровень переданного бита с уровнем, который наблюдается на шине. Если эти уровни одинаковы, узел может продолжать передачу. Когда послан единичный уровень, а наблюдается нулевой уровень, это означает, что узел потерял право арбитража и не должен посылать больше ни одного бита.
- Для обнаружения ошибок применяются следующие меры:
    - Текущий контроль (передатчики сравнивают уровни передающихся битов с уровнями, обнаруженными на шине).
    - Циклический контроль по избыточности.
    - Контроль кадра сообщения.
- Искаженные кадры помечаются любым узлом, обнаружившим ошибку. Такие кадры прерываются и должны быть автоматически переданы заново.
- Шина состоит из канала обмена, который передаёт биты. Из этих данных может быть получена информация о пересинхронизации.
- Все приемники проверяют непротиворечивость получаемого сообщения, подтверждают непротиворечивость сообщения и помечают противоречивые сообщения.

5.18.3 Принципы работы CAN контроллера

Данный контроллер предоставляет полные аппаратные средства для удобной фильтрации приема и управления сообщениями. Для каждого передаваемого или получаемого сообщения настраиваются так называемый «объект сообщения», в котором хранится вся информация, относящаяся к сообщению: идентификатор, байты данных и т.д. Объекты сообщений хранятся в Message RAM. Все функции обработки сообщений (фильтрация, пересылка сообщений между CAN core и Message RAM, обработка внешних запросов, а также выставление прерывания) реализованы в блоке Message Handler.

5.18.3.1 Конфигурирование режимов работы CAN канала

Канал CAN может находиться в следующих режимах:

- Тестовый режим работы (Test Mode);
- Режим «Молчания» (Silent Mode);
- Режим кольцевой проверки (Loop Back Mode);

- Основной тестовый режим работы;
- Режим кольцевой проверки, совмещенный с режимом «Молчания» проверки (Silent Loop Back Mode).

5.18.3.1.1 Тестовый режим работы (Test Mode)

Для того, чтобы настроить контроллер для работы в тестовом режиме необходимо установить бит **Test** в регистре управления CAN контроллера. В данном режиме биты **Tx1**, **Tx0**, **LBack**, **Silent, Basic** регистра тестового режима программно доступны для записи. Бит **Rx** отслеживает состояние вывода **CAN_RX**, поэтому доступен только для чтения.

5.18.3.1.2 Режим «Молчания» (Silent Mode)

Для того, чтобы настроить контроллер для работы в режиме «Молчания» необходимо установить бит **Silent** в регистре управления CAN контроллера. В данном режиме контроллер может получать сообщения от других удаленных узлов, но отвечать только рецессивными битами по CAN шине, не возможна передача сообщения. Данный режим используется для анализа трафика по CAN шине, без влияния на передачу данных.

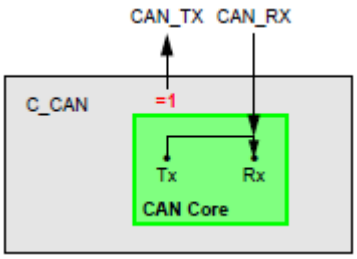


Рисунок 5.60 - Соединение внешних сигналов CAN контроллера, находящегося в режиме «Молчания»

5.18.3.1.3 Режим кольцевой проверки (Loop Back Mode)

Для того, чтобы настроить контроллер для работы в режиме «Молчания» необходимо установить бит LBack в регистре управления CAN контроллера. В данном режиме сообщение, посланное контроллером, принимается им же и сохраняется в приемном буфере, если предварительно прошло проверку. Данный режим используется для самотестирования.

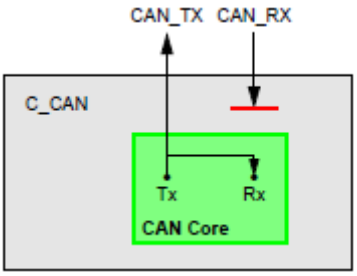


Рисунок 5.61 - Соединение внешних сигналов CAN контроллера, находящегося в режиме кольцевой проверки

5.18.3.1.4 Режим кольцевой проверки, совмещенный с режимом «Молчания» проверки (Silent Loop Back Mode)

Для того, чтобы контроллер работал совместно как и в режиме «Молчания», так и в режиме кольцевой проверки, необходимо установить биты **LBack** и **Silent** в регистре управления CAN контроллера. Данный режим используется для тестирования контроллера CAN, и позволяет полностью отключить внешние пины от системы.

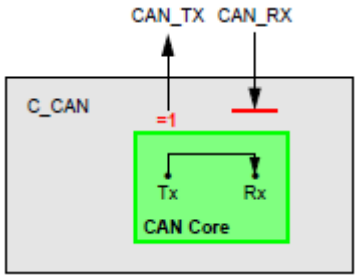


Рисунок 5.62 - Соединение внешних сигналов CAN контроллера, работающего в режимах кольцевой проверки и «Молчания»

5.18.3.1.5 Основной тестовый режим работы

Для того, чтобы настроить контроллер для работы в основном тестовом режиме, необходимо установить бит **Basic** в регистре управления CAN контроллера. В данном режиме отключена память Message RAM. Буферные регистры блока IF1 настроены, как передающий буфер. Для отправки сообщений из передающего буфера, должен быть установлен бит **Busy** в регистре запросов команд (IF1 Command Request). После передачи сообщения бит сбрасывается. Передача данных может быть прервана, если в процессе ожидания передачи бит **Busy** был сброшен.

Буферные регистры блока IF2 настроены, как принимающий буфер. Полученное сообщение сохраняется в приемном буфере (IF2 buffer) без предварительной проверки приемным фильтром.

5.18.3.2 Программная инициализация

Программная инициализация начинается с установки бита **Init** в регистре контроля CAN - контроллера. Этот бит может быть установлен программно микроконтроллером или автоматически после аппаратного сброса или после того, как CAN - контроллер будет переведен в состояние "отключен от шины".

Пока установлен бит **Init**:

- все передачи данных от / к CAN-контроллеру останавливаются.
- На линию передается рецессивный уровень (лог. 1).
- Счетчики ошибок EML не изменяются.
- Не изменяются конфигурационные регистры.

Для инициализации CAN – контроллера необходимо произвести следующие действия:

- настроить регистр BTR
- установить регистры глобальной маски (Global Mask Registers)
- проинициализировать каждый буфер сообщения.

Если буфер сообщения не нужен, то нужно сбросить бит **MsgVal** в регистре контроля этого буфера, т.е. объявить этот буфер недействительным. Это необходимо сделать во время инициализации. Установка бита **CCE** регистра управления CAN контроллера дополнительно разрешает изменять содержимое регистра BTR.

После завершения инициализации необходимо сбросить бит **Init**. После этого контроллер синхронизирует свою работу с обменом данными по шине CAN.

Инициализация буферов сообщений может производиться и при сброшенном бите **Init**, это так называемая инициализация “на лету”. Для изменения конфигурации буфера сообщения в течение нормальных операций, микроконтроллер должен сбросить бит **MsgVal** у данного буфера, т.е. объявить его недействительным. После завершения изменения конфигурации бит **MsgVal** должен быть снова установлен.

5.18.3.3 Объект Сообщений (Message Objects)

Объект сообщения (MO) является дескриптором кадра CAN и содержит всю информацию для обработки кадра. MO описывает сообщение CAN как объект.

Объекты сообщений хранятся в памяти Message RAM и должны быть проинициализированы процессором, иначе статус объекта должен быть объявлен недействительным (бит **MsgVal** регистра устанавливается равным 0). Так же должен быть настроен модуль битовой синхронизации (Bit Timing), прежде чем процессор сбросит бит **Init** в регистре управления CAN контроллера.

При конфигурировании MO в блоке интерфейсных регистров (IFx Control) программным способом задаются значения поля арбитража, поля маски, поля управляющих битов и поле данных. Значения регистра запроса команд (IFx Command Request), значение буферных регистров блоков IFx (IFx Message Buffer) записываются в память Message RAM в формате объекта сообщения (Message Object).

Полученные сообщения поступают на вход приемного фильтра. Если сообщение не отфильтровывается, то сохранятся в памяти Message RAM. Сообщения, запрошенные на передачу, загружаются в сдвиговый регистр блока CAN_Core, после чего предаются по шине CAN.

Полученные данные считываются процессором из приемного буфера блока IFx, так же процессор может модифицировать отправляемые данные, находящиеся в передающем буфере блока IFx.

В зависимости от программной настройки процессор может быть остановлен либо сообщением, принятым с шины CAN, либо событием ошибки.

5.18.3.4 Блок управления Message Handler контроллера CAN

Блок управления (Message Handler) контролирует обмен данными между приемопередающим сдвиговым регистром контроллера CAN и памятью Message RAM, а также между памятью и блоками IFx.

Особенности работы блока Message Handler:

- Запись данных из передающего буфера блока IFx в память Message RAM;
- Запись данных из памяти Message RAM в приемный буфер блока IFx;
- Запись данных из сдвигового регистра в память Message RAM;
- Запись данных из памяти Message RAM в сдвиговой регистр;
- Запись данных из сдвигового регистра в фильтр;
- Процедура поиска объекта сообщения в памяти;
- Управление флагами TxRqst;
- Управление прерываниями.

5.18.3.5 Передача сообщений CAN контроллером

Используя интерфейсные регистры блоков IFx, процессор в любое время может читать или писать сообщения. Блок Message Handler гарантирует корректность данных в случае одновременного доступа.

Когда процессор инициирует передачу данных между регистрами блока IFx и памятью Message RAM, устанавливается бит **Busy** регистра запроса команд (IFx Command Request). После завершения передачи данных бит сбрасывается. Регистр маски команд (IFx Command

Mask) соответствующего блока IFx определяет, будет ли объект сообщения (MO) передан целиком, или частично.

Пройдя приемный фильтр, полученные сообщения сохранятся в соответствующем MO. Сообщение сохранятся вместе с битами поля арбитража, поля DLC и с 8 байтами данных.

Структура памяти Message RAM не позволяет побитно переписывать значения бит MO, поэтому объект должен записываться в память целиком.

Если сообщения, которые должны быть отправлены, имеют одинаковые значение бит полей арбитража и управления, то процессор может изменять только значение полей данных, а не изменять сообщение целиком.

Объект сообщений должен быть сконфигурирован до прихода запроса на его отправку.

В одно и то же время могут прийти запросы на несколько объектов сообщений, тогда сообщения передаются последовательно, учитывая их приоритет. Действительность сообщения может быть изменена в любое время, даже если ожидается запрошенная передача.

В зависимости от конфигурации объекта сообщений передача сообщений может быть запрошена автономно при повторении кадра удаленного запроса данных (remote frame) с совпадающим идентификатором.

Если сдвиговый регистр блока CAN Core готов принять данные, и если нет передачи данных между регистрами блоков IFx и памятью Message RAM, то устанавливаются биты **MsgVal** регистра действительности сообщения (Message Valid) и биты **TxRqst** регистра запроса передачи (Transmission Request). В сдвиговый регистр записывается действительный объект сообщения с наивысшим приоритетом, начинается передача данных. Сбрасывается бит **NewDat** регистра новых данных (New Data Registers).

После успешной передачи, если нет новых записанных данных (**NewDat** = 0) с момента запуска передачи, сбрасывается соответствующий бит **TxRqst** регистр запроса передачи (Transmission Request). Если установлен бит **TxIE,** то будет установлен бит **IntPnd**, после успешной передачи. Если CAN проиграл арбитраж или произошла ошибка в процессе передачи, то сообщение будет отправлено снова, как только освободится CAN шина. Сообщения отправляются согласно установленным приоритетам.

5.18.3.6 Фильтрация принятого сообщения

Сообщение поступает в сдвиговый регистр контроллера CAN Core, после чего блок фильтра производит сравнение полей арбитража и поля управляющих битов принятого сообщения с загруженными значениями полей арбитража и поля маски первого объекта сообщения, хранящегося в памяти Message RAM. Процедура сравнения продолжается до тех пор, пока не будет найдено совпадение объекта сообщения, или пока не будет полностью проверена память Message RAM. Если произошло совпадение, то дальнейшая работа автомата состояния контроллера зависит от типа принятого сообщения (кадр данных, или кадр удаленного запроса).

5.18.3.6.1 Прием кадра данных

Сообщение из сдвигового регистра сохраняется в найденном объекте сообщения в памяти. Сообщение сохраняется в формате объекта (т.е. сохраняются также значение полей арбитража и DLC).

Если получены новые данные, то устанавливается бит **NewDat**. Сброс бита **NewDat** происходит, когда данные читаются процессором. Если к моменту приема сообщения бит NewDat все еще установлен, то установка бита **MsgLst** является признаком того, что предыдущие данные (предположительно невидимые процессору) считаются потерянными.

Если установлен бит **RxIE** и установлен бит **IntPnd**, то в регистре прерываний (Interrupt Register) прописывается адрес объекта сообщения.

Бит **TxRqst** сбрасывается для предотвращения передачи кадра удаленного запроса от данного объекта сообщения до тех пор, пока не будет получен запрошенный кадр данных.

5.18.3.6.2 Прием кадра удаленного запроса

Полученный кадр удаленного запроса может быть трех видов:

1) **Dir** = ‘1’ (режим передачи), **RmtEn** = ‘1’, **UMask** = ‘1’ или ’0’

При совпадении кадра формата удаленного запроса данной конфигурацией устанавливается бит **TxRqst** данного объекта сообщения. Остальная часть объекта сообщения остается неизменной.

2) **Dir** = ‘1’ (режим передачи), **RmtEn** = ‘0’, **UMask** = ’0’

При совпадении кадра формата удаленного запроса данной конфигурацией бит **TxRqst** данного объекта сообщения остается неизменным. Данный кадр игнорируется.

3) **Dir** = ‘1’ (режим передачи), **RmtEn** = ‘0’, **UMask** = ’1’

При совпадении кадра формата удаленного запроса данной конфигурацией бит **TxRqst** данного объекта сообщения сбрасывается. Поле арбитража и поле бит управления переписываются из сдвигового регистра в память в формате объекта сообщений. Устанавливается бит **NewDat** данного объекта. Поле данных объекта сообщений остается неизменным.

5.18.3.7 Приоритет полученных и отправленных сообщений

Приоритет объекта сообщений связан с номером сообщения. Объект с номером 1 имеет самый высокий приоритет, а объект с номером 32 имеет самый низкий приоритет. Если отправки ожидается более одного запроса, то обслуживание запросов происходит согласно приоритету соответствующего объекта сообщений.

5.18.3.8 Конфигурация объекта сообщения для передачи

На рисунке ниже показано, как объект сообщения для передачи должен быть инициализирован.

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

Рисунок 5.63 - Значения бит для инициализации передаваемого объекта сообщения

Поле арбитража состоит из идентификатора (**ID28-0**) и бита **Xtd**, определяющего тип сообщения. Для стандартного фрейма используется 11-битный идентификатор. Программным способом задаются **ID28 - ID18**, при этом **ID17 - ID0** остаются неизменными.

Если установлен бит **TxIE**, то бит **IntPnd** будет установлен после успешной передачи объекта сообщения.

Если установлен бит **RmtEn**, то при совпадении с полученным кадром удаленного запроса будет установлен бит **TxRqst**; кадр данных будет автоматически отправлен на полученный запрос.

Поле данных состоит из полей **DLC3-0**, **Data0-7**, биты **TxRqst** и **RmtEn** могут быть не установлены до прихода действительных данных.

Поле **Mask** состоит из битов **Msk28-0**, **UMask**, **MXtd**, и **Mdir**.

Бит **Dir** не должен быть замаскирован.

5.18.3.9 Конфигурация принятого объекта сообщения

На рисунке ниже показано, как принятый объект сообщения должен быть инициализирован.

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

Рисунок 5.64 - Инициализация объекта сообщения

Поле арбитража состоит из идентификатора (**ID28-0**) и бита **Xtd**, определяющего тип сообщения. Для стандартного фрейма используется 11-битный идентификатор. Программным способом задаются **ID28 - ID18**, при этом **ID17 - ID0** остаются неизменными.

Если установлен бит **RxIE**, то бит **IntPnd** будет установлен после успешного приема и сохранения объекта сообщения.

Поле данных состоит из поля **DLC3-0**. Когда сохраняются данные в формате объекта сообщения, также сохраняются биты поля **DLC**.

Поле **Mask** состоит из битов **Msk28-0**, **UMask**, **MXtd**, и **Mdir**.

Бит **Dir** не должен быть замаскирован.

5.18.4 Программная модель контроллера CAN

Программно доступные регистры контроллера портов CAN расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение CAN Base = 0xFFF05000 и общий размер 4 Кб. Регистры имеют разрядность 16 бит. Два блока интерфейсных регистров (IF1 и IF2) управляют доступом процессора к памяти Message RAM. Использование буферов данных в этих блоках позволяет избегать конфликтов доступа процессора к памяти Message RAM, возникающих при приеме и отправке сообщений.

Спецификация регистров представлена в таблице ниже:

Таблица 5.184 - Сводная таблица регистров CAN контроллера

Адрес	Название	Значение после сброса	Дополнительно
CAN Base +0x000	Регистр управления CAN контроллера	0x0001	
CAN Base +0x004	Регистр статуса CAN контроллера	0x0000	
CAN Base +0x008	Счетчик ошибок	0x0000	Только для чтения
CAN Base +0x00C	Регистр битовой синхронизации	0x2301	Запись возможна только, если установлен бит CCE
CAN Base +0x010	Статус прерываний	0x0000	Только для чтения
CAN Base +0x014	Регистр тестового режима	0x00 и 0br0000000 1)	Запись возможна только, если установлен бит Test
CAN Base +0x018	Регистр расширения регистра BRP	0x0000	Запись возможна только, если установлен бит CCE
CAN Base +0x01C	резерв	- 3)	
CAN Base +0x020	Регистр запросов команд блока IF1	0x0001	
CAN Base +0x024	Регистр маски команд блока IF1	0x0000	
CAN Base +0x028	1-ый Регистр маски блока IF1	0xFFFF	
CAN Base +0x02C	2-ой Регистр маски блока IF1	0xFFFF	
CAN Base +0x030	1-ый Регистр арбитража блока IF1	0x0000	
CAN Base +0x034	2-ой Регистр арбитража блока IF1	0x0000	
CAN Base +0x038	Регистр управления блока IF1	0x0000	
CAN Base +0x03C	1-ый Регистр данных Data A	0x0000	
CAN Base +0x040	2-ой Регистр данных Data A	0x0000	
CAN Base +0x044	1-ый Регистр данных Data B	0x0000	
CAN Base +0x048	2-ой Регистр данных Data B	0x0000	
CAN Base +0x04C-0x07C	резерв	- 3)	
CAN Base +0x080-0x0A8	регистры блока IF2	см примечание 2)	Набор регистров, как для блока IF1
CAN Base +0x0AC-0x0FC	резерв	- 3)	
CAN Base +0x100	1-ый Регистр запроса передачи	0x0000	Только для чтения
CAN Base +0x104	2-ой Регистр запроса передачи	0x0000	Только для чтения
CAN Base +0x108-0x11C	резерв	- 3)	
CAN Base +0x120	1-ый Регистр новых данных	0x0000	Только для чтения
CAN Base +0x124	2-ой Регистр новых данных	0x0000	Только для чтения
CAN Base +0x128-0x13C	резерв	- 3)	
CAN Base +0x140	1-ый Регистры отложенных прерываний	0x0000	Только для чтения
CAN Base +0x144	2-ой Регистры отложенных прерываний	0x0000	Только для чтения
CAN Base +0x148-0x15C	резерв	- 3)	
CAN Base +0x160	1-ый Регистр признаков действительности объектов сообщений	0x0000	Только для чтения
CAN Base +0x164	2-ой Регистр признаков действительности объектов сообщений	0x0000	Только для чтения
CAN Base +0x168-0x17C	резерв	- 3)	

Примечание:

1) r означает действительное значение для CAN_RX.

2) Группа интерфейсных регистров блока IF1 и блока IF2 имеют одинаковые функции.

3) Чтение резервных битов возвращает «0», кроме 2-ого регистра маски блоков IFx.

5.18.4.1 Описание аппаратного сброса

Значение регистров после аппаратного сброса приведено выше в таблице (Таблица 5.184). При аппаратном сбросе контроллер находится в состоянии отключения шины, уровень выхода CAN_TX устанавливается эквивалентным единичному уровню - "recessive".

После записи значения 0x0001 (Init = ‘1’) в регистр управления CAN контроллера начинается программная инициализация. Аппаратный сброс не изменяет данные, хранимые в памяти. После включения питания память Message RAM не определена.

5.18.4.2 Регистры CAN контроллера

Регистры CAN контроллера располагаются в блоке CAN Core. Регистровая модель контроллера программным способом управляет рабочими режимами контроллера, настраивает побитную синхронизацию CAN контроллера, а также предоставляет информацию о статусе контроллера.

5.18.4.2.1 Регистр управления CAN контроллера

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res	res	res	Test	CCE	DAR	res	EIE	SIE	IE	Init
r	r	r	r	r	r	r	r	rw	rw	rw	r	rw	rw	rw	rw

Рисунок 5.65 - Регистр управления CAN контроллера

Таблица 5.185 - Структура регистра управления CAN

Биты	Название	Тип	Функция	Описание
[0]	Init	ЗП/ЧТ	Инициализация	1 – Идет процесс инициализации
				0 – нормальный режим функционирования.
[1]	IE	ЗП/ЧТ	Разрешение прерывания	1 – разрешено. Прерывание IRQ_B устанавливается в состояние логического нуля и остается в данном состоянии пока обрабатываются все отложенные прерывания
				0 – Прерывания модуля IRQ_B остаются в состоянии логической единицы.
[2]	SIE	ЗП/ЧТ	Разрешение генерации прерывания при изменении состояния.	1 – Разрешен. Прерывание будет сгенерировано, как только успешно завершится передача, или если на шине CAN обнаружена ошибка.
				0 – не разрешено изменение статуса по прерыванию.
[3]	EIE	ЗП/ЧТ	Разрешение генерации прерывания при ошибке.	1 – Разрешена. Изменение бита Boff, или бита Ewarn в регистре статуса (Status Register), вызовет генерацию микроконтроллером прерывания
				0 – Не разрешена. Никакого прерывания по ошибке не будет сформировано
[4]	Res	ЧТ	Зарезервировано	Зарезервировано
[5]	DAR	ЗП/ЧТ	Запрет автоматического повтора передачи сообщений	1 – повторная передача запрещена
				0 – разрешена повторная передача нарушенного сообщения
[6]	CCE	ЗП/ЧТ	Разрешение изменения конфигурации	1 – регистр побитовой синхронизации программно доступен процессору на запись. Разрешает доступ микроконтроллеру к регистрам BTR0 и BTR1
				0 - регистр побитовой синхронизации программно не доступен процессору на запись запрещает доступ микроконтроллеру к регистрам BTR0 и BTR1
[7]	Test	ЗП/ЧТ	Тестовый режим.	1 – Этот бит используется для тестирования при производстве микроконтроллера.
				0 – При выполнении нормальных операций этот бит должен быть сброшен.
[15:8]	res	ЧТ	Зарезервированы	Зарезервированы

Последовательность действий контроллера в состоянии "отключения от шины" не может быть сокращена путем установки, или сброса бита Init. Если контроллер находится в данном состоянии, бит Init будет установлен сам по себе, при этом будут остановлены все транзакции на шине. После сброса процессором бита Init, устройство будет ожидать появления 129 событий BusIdle (129 * 11 последовательных рецессивных битов), прежде чем будет возобновлены нормальные операции на шине. В конце инициализации должны быть сброшены счетчики ошибок.

После сброса бита Init, каждый раз, когда отслеживается последовательность 11 рецессивных битов, в регистре статуса контроллера CAN поле LEC прописывается значением Bit0Error, позволяя узлу контролировать появление на шине доминантных битов, или проверять нарушения шины, так же позволяет контролировать протекание последовательности действий контроллера в состоянии "отключения от шины".

5.18.4.2.2 Регистр статуса CAN контроллера

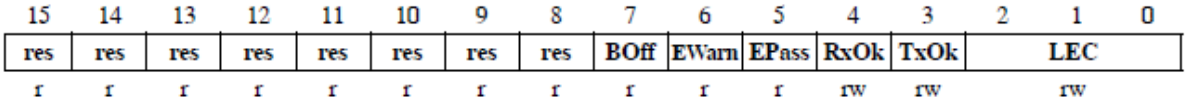


Рисунок 5.66 - Регистр статуса CAN контроллера

Таблица 5.186 - Структура регистра статуса CAN Контроллера

Биты	Название	Тип	Функция	Описание	
[2:0]	LEC	ЗП/ЧТ	Код последней ошибки. Это поле показывает тип последней ошибки, произошедшей на шине CAN. Если сообщение было принято (передано) без ошибок, это поле равно 0. Код "7" не используется.	0 0 0 – Нет ошибок	-
				0 0 1 – Ошибка заполнения	Было принято больше чем 5 одинаковых бит в той части сообщения, где это не разрешено.
				0 1 0 – Ошибка формы	Фиксированный формат части принятого сообщения имеет не правильный формат.
				0 1 1 – Ошибка подтверждения	Переданное сообщение не было подтверждено другим узлом.
				1 0 0 – Ошибка единичного бита	В течение передачи сообщения (исключая поле арбитража), узел посылал единичный уровень, а проконтролировал на шине нулевой уровень.
				1 0 1 – Ошибка нулевого бита	В течение передачи сообщения (исключая поле арбитража), узел посылал нулевой уровень, а проконтролировал на шине единичный уровень
				1 1 0 – Ошибка CRC	В принятом сообщении не правильная CRC-сумма.
				111 – Не используется	Запись значения «7» в данное поле не приведет к каким-либо событиям на шине CAN.
[3]	TxOk	ЗП/ЧТ	Сообщение успешно отправлено	Устанавливается, если последняя передача сообщения была успешно завершена (при передаче не было ошибок и было принято подтверждение хотя бы от одного узла). Этот бит не сбрасывается CAN - контроллером!!	
[4]	RxOk	ЗП/ЧТ	Сообщение успешно получено	Устанавливается, если было успешно принято сообщение. Этот бит не сбрасывается CAN - контроллером!!	
[5]	EPass	ЧТ	Состояние пассивной ошибки	1 – Контроллер CAN находится в состоянии пассивной ошибки	
				0 - Контроллер CAN не находится в состоянии пассивной ошибки	
[6]	EWarn	ЧТ	Статус предупреждения	Устанавливается, если хотя бы один из счетчиков ошибок (EML) стал больше, чем 96.	
[7]	BOff	ЧТ	Статус контроллера «отключения от шины»	Устанавливается, если произошел переход CAN - контроллера в состояние "отключен от шины".	
[15:8]	res	ЧТ	Зарезервированы	Зарезервированы	

Поле **LEC** сохраняет код последней произошедшей ошибки на шине CAN. Это поле кодируется «0», когда сообщение передается/принимается без ошибки.

Чтение старшего байта этого регистра очистит значение младшего байта регистра прерывания. Для того чтобы это не произошло, используйте байтовую адресацию.

5.18.4.2.3 Статус прерываний

Имеются различные прерывания:

- Прерывание по завершению получения (установлены биты **RxOk**, **SIE**, прописано поле LEC в регистре статуса);
- Прерывание по завершению передачи (установлены биты **TxOk**, **SIE**, прописано поле LEC в регистре статуса);
- Прерывание по обнаружению ошибки;
- Прерывание по заполнению буфера кадра;
- Прерывание по отключению от шины (установлены биты **BOff**, **EWarn** регистра статуса, в регистре управления установлен EIE);

Контроллер не перейдет в состояния прерывания при изменении бита **EPass** или при записи бит **TxOk**, **RxOk**, или LEC.

Чтение регистра статуса сбрасывает значение идентификатора прерывания в регистре прерывания, если были отложенные прерывания.

5.18.4.2.4 Счетчик ошибок

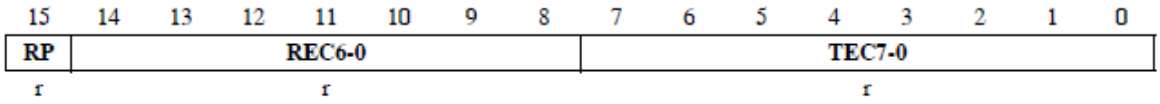


Рисунок 5.67 - Регистр счетчика ошибок

Таблица 5.187 - Структура регистра статуса CAN Контроллера

Биты	Название	Тип	Функция	Описание
[7:0]	TEC7-0	ЧТ	Счетчик отправленных ошибок	Текущее значение счетчика. Счетчик принимает значения в диапазоне от 0 до 255.
[14:8]	REC6-0	ЧТ	Счетчик полученных ошибок (REC)	Текущее значение счетчика. Счетчик принимает значения в диапазоне от 0 до 127.
[15]	RP	ЧТ	Получение пассивной ошибки	1 – Значение счетчика REC достигло уровня пассивной ошибки, согласно спецификации CAN
				0 – Значение счетчика REC ниже уровня пассивной ошибки

5.18.4.2.5 Регистр битовой синхронизации (BTR)

Согласно CAN – спецификации битовое время разделено на четыре сегмента:

- SYNC сегмент, сегмент времени распространения, сегмент TSEG1 и сегмент TSEG2.
- SYNC сегмент всегда равен одному кванту времени.
- Сегмент времени распространения, и сегмент TSEG1 объединены в один временной интервал, который программируется полем TSEG1 регистра BTR и определяет время перед точкой считывания.
- Сегмент TSEG2 определяет время после точки считывания.

Длины этих сегментов (кроме SYNC сегмента) программируются через регистр BTR.

Примечание: Для правильного определения длин этих сегментов смотрите CAN – спецификацию.

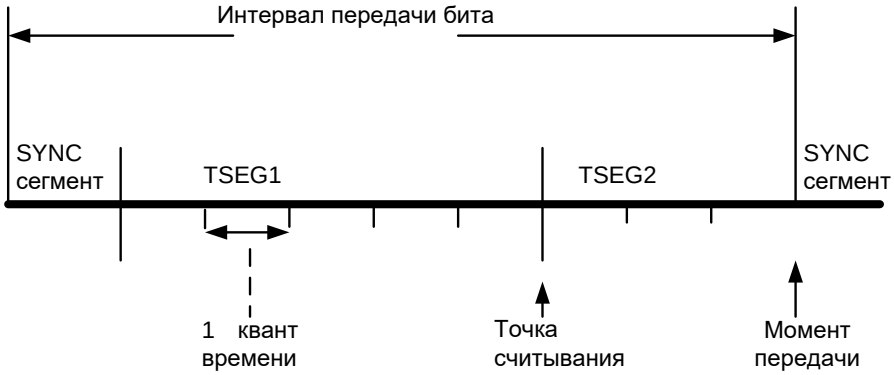


Рисунок 5.68 - Параметры битовой синхронизации

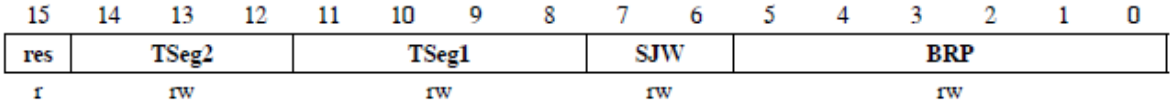


Рисунок 5.69 - Регистр BTR

Таблица 5.188 - Структура регистра BTR

Биты	Название	Тип	Функция	Описание
[5:0]	BRP	ЗП/ЧТ	Делитель скорости передачи	Значение делителя частоты осциллятора. Определяет число квантов времени сегмента. Действительные значения поля [0…63]
[7:6]	SJW	ЗП/ЧТ	Шина перехода синхронизации	Действительные значения поля [0…3]
[11:8]	TSeg1	ЗП/ЧТ	Размер временного сегмента перед точкой считывания	Определяет размер временного сегмента перед точкой считывания Действительные значения поля [1…15]
[14:12]	TSeg2	ЗП/ЧТ	размер временного сегмента после точки считывания	Определяет размер временного сегмента после точки считывания Действительные значения поля [1…7]
[15]	res	ЧТ	Зарезервировано	Зарезервировано

С частотой сигнала CAN_CLK, равной 8 МГц, контроллер CAN может передавать данные со скоростью 500 кбит/с. Запись в регистры возможна тогда, когда установлены биты CCE и Init регистра управления CAN контроллера.

5.18.4.2.6 Регистр тестового режима

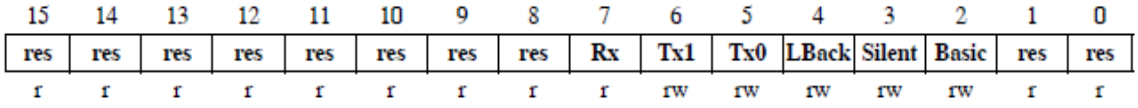


Рисунок 5.70 - Регистр TR

Таблица 5.189 - Структура регистра TR

Биты	Название	Тип	Функция	Описание
[1:0]	res	ЧТ		
[2]	Basic	ЗП/ЧТ	Основной режим работы	1–блок интерфейсных регистров IF1 используется как передающий буфер, блок интерфейсных регистров IF2 используется как приемный буфер.
				0 – Основной режим не используется.
[3]	Silent	ЗП/ЧТ	Режим молчания	1 - Работает в режиме молчания
				0 – Работает в нормальном режиме
[4]	LBack	ЗП/ЧТ	Режим кольцевой проверки	1 - Разрешает кольцевой проверки
				0 – Не разрешает кольцевой проверки
[6:5]	[Tx1, Tx0]	ЗП/ЧТ	Контролирует вывод CAN_TX	00 – Значение при аппаратном сбросе. CAN_TX контролируется контроллером CAN
				01 – наблюдается момент времени - точки выборки, на шине CAN_TX, те момент времени, в котором уровень шины считывается и интерпретируется как значение соответствующего бита (Sample Point).
				10 – устанавливается доминантный уровень на CAN_TX
				11 – устанавливается рецессивный уровень на CAN_TX
[7]	Rx	ЧТ	Отслеживает текущее состояние CAN_RX вывода	1 – Рецессивный уровень шины CAN (CAN_RX = 1)
				0 - Доминантный уровень шины CAN (CAN_RX = 0)
[15:8]	res	ЧТ	Зарезервированы	Зарезервированы

Программная запись в регистр тестового режима разрешена, когда установлен бит Test в регистре управления контроллера CAN. Нарушение обмена сообщений происходит, если комбинация бит [Tx1, Tx0] не равна значению «00».

5.18.4.2.7 Регистр расширения регистра BRP

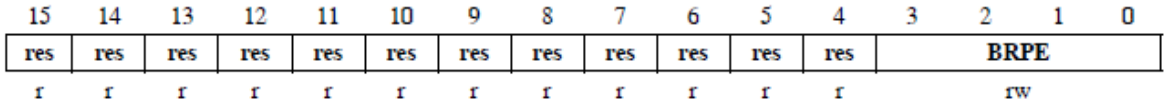


Рисунок 5.71 - Регистр расширения регистра BRP

Таблица 5.190 - Структура регистра расширения регистра BRP

Биты	Название	Тип	Функция	Описание
[3:0]	BRPE	ЗП/ЧТ	Расширение значения делителя скорости передачи	Программным способом BRPE может быть увеличен до значения 1023.
[15:4]	res	ЧТ	Зарезервированы	Зарезервированы

5.18.4.3 Набор интерфейсных регистров блока IF1 и блока IF2

Программная модель контроллера CAN составлена для двух блоков IFx. Работа блока IF1 и IF2 аналогична, за исключением работы в основном тестовом режиме. Регистры одного блока

могут быть настроены на передачу данных в память Message RAM, в то время как регистры другого блока могут быть настроены на прием данных из памяти Message RAM. Программная модель двух блоков интерфейсных регистров приведена ниже (Таблица 5.192). Каждый набор регистров состоит из буферных регистров, предназначенных для хранения сообщений, и регистров управления каждого блока. Регистр маски команд задает направление передачи данных, а также определяет, какие биты поля объекта сообщения будут отправлены.

Таблица 5.191 - Набор интерфейсных регистров блоков IF1 и IF2

Адрес	Набор регистров блока IF1	Адрес	Набор регистров блока IF2
CAN Base + 0x20	Регистр запросов команд блока IF1	CAN Base + 0x80	Регистр запросов команд блока IF2
CAN Base + 0x24	Регистр маски команд блока IF1	CAN Base + 0x84	Регистр маски команд блока IF2
CAN Base + 0x28	1-ый Регистр маски блока IF1	CAN Base + 0x88	1-ый Регистр маски блока IF2
CAN Base + 0x2C	2-ой Регистр маски блока IF1	CAN Base + 0x8C	2-ой Регистр маски блока IF2
CAN Base + 0x30	1-ый Регистр арбитража блока IF1	CAN Base + 0x90	1-ый Регистр арбитража блока IF2
CAN Base + 0x34	2-ой Регистр арбитража блока IF1	CAN Base + 0x94	2-ой Регистр арбитража блока IF2
CAN Base + 0x38	Регистр управления блока IF1	CAN Base + 0x98	Регистр управления блока IF2
CAN Base + 0x3C	1-ый Регистр данных Data A блока IF1	CAN Base + 0x9C	1-ый Регистр данных Data A блока IF2
CAN Base + 0x40	2-ой Регистр данных Data A блока IF1	CAN Base + 0xA0	2-ой Регистр данных Data A блока IF2
CAN Base + 0x44	1-ый Регистр данных Data B блока IF1	CAN Base + 0xA4	1-ый Регистр данных Data B блока IF2
CAN Base + 0x48	2-ой Регистр данных Data B блока IF1	CAN Base + 0xA8	2-ой Регистр данных Data B блока IF2

5.18.4.3.1 Регистры запросов команд блоков IFx

Передача начинается с того, что процессор прописывает номер сообщения в регистре запросов команд. Одновременно с записью автоматически устанавливается бит Busy, а сигнал CAN_WAIT_B принимает значение логического нуля, уведомляя процессор, что началась передача сообщения. Передача данных между блоком IFx и памятью Message RAM занимает от 3 до 6 тактов синхросигнала CAN_CLK. Бита Busy сбрасывается, CAN_WAIT_B принимает значение логической единицы.

	15	14	13	12	11	10	9	8	7	6	5 4 3 2 1 0
Регистр запросов команд блока IF1 (адрес 0x20)	Busy	res	res	res	res	res	res	res	res	res	Message Number
Регистр запросов команд блока IF2 (адрес 0x80)	Busy	res	res	res	res	res	res	res	res	res	Message Number
	ЧТ	ЧТ	ЧТ	ЧТ	ЧТ	ЧТ	ЧТ	ЧТ	ЧТ	ЧТ	ЧТ/ЗП

Рисунок 5.72 - Регистры запросов команд блоков IFx

Таблица 5.192 - Структура регистров запросов команд блоков IFx

Биты	Название	Тип	Функция	Описание
[5:0]	Message Number	ЗП/ЧТ	Номер сообщения	0x01-0x20: Действующий номер сообщения. Объект сообщения выбирается из памяти Message RAM для передачи данных
				0x00: Недействительный номер, интерпретируется как 0x20
				0x21 – 0x3F: Недействительный номер, интерпретируется как 0x01- 0x1F
[14:6]	res	ЧТ	Зарезервированы	Зарезервированы
[15]	Busy	ЧТ	Флаг занятости	Устанавливается в «1», если произошла запись в данный регистр
				Сбрасывается, когда завершается чтение, или запись.

Примечание: Если Message Number прописывается недействительным значением, то номер сообщения интерпретируется в действующее значение, затем происходит передача объекта сообщения.

5.18.4.3.2 Регистры маски команд блоков IFx

Управляющие биты регистра маски команд задают направление передачи данных и определяют, будут ли буферные регистры настроены как приемные буферы или как передающие.

	15 14 13 12 11 10 9 8	7	6	5	4	3	2	1	0
Регистры маски команд блока IF1 (адрес 0x24)	res	WR/RD	Mask	Arb	Control	ClrIntPnd	TxRqst/ NewDat	Data A	Data B
Регистры маски команд блока IF2 (адрес 0x84)	res	WR/RD	Mask	Arb	Control	ClrIntPnd	TxRqst/ NewDat	Data A	Data B
	ЧТ	ЧТ/ЗП	ЧТ/ЗП	ЧТ/ЗП	ЧТ/ЗП	ЧТ/ЗП	ЧТ/ЗП	ЧТ/ЗП	ЧТ/ЗП

Рисунок 5.73 - Регистры маски команд блоков IFx

Таблица 5.193 - Структура регистров маски команд блоков IFx

Биты	Название	Тип	Функция	Описание
[15:8]	res	ЧТ	Зарезервированы	Зарезервированы
[7]	WR/RD	ЗП/ЧТ	Запись/Чтение	1 – Запись. Данные передаются из буферных регистров выбранного блока в память в формате объекта сообщения. Адрес записи задается в регистре запросов команд
				0 – Чтение. Данные передаются из памяти в формате объекта сообщения в буферные регистры выбранного блока. Адрес чтения задается в регистре запросов команд
Запись*				
[6]	Mask	ЗП/ЧТ	Доступ к битам маски	1 – В объект сообщения передаются биты: Identifier Mask + MDir + MXtd
				0 – Биты маски в объекте сообщения не меняются
[5]	Arb	ЗП/ЧТ	Доступ к битам арбитража	1 - В объект сообщения передаются биты: Identifier + Dir + Xtd + MsgVal
				0 - Биты арбитража в объекте сообщения не меняются
[4]	Control	ЗП/ЧТ	Доступ к битам управления	1 – Передаются биты управления в объект сообщения
				0 – Биты остаются неизменными
[3]	ClrIntPnd	ЗП/ЧТ	Сброс бита ожидающего прерывания	При записи в объект сообщения этот бит игнорируется.
[2]	TxRqst/NewDat	ЗП/ЧТ	Доступ к биту запроса передачи данных	1 – устанавливается бит TxRqst
				0 – значение бита остается неизменным Примечание: Если передача настраивается путем программной записи бита TxRqst/NewDat в регистре маски управления блоков IFx, то бит TxRqst игнорируется в регистре управления сообщением блока IFx.
[1]	Data A	ЗП/ЧТ	Доступ к 0-3 байтам данных	1 – Передача с 0 по 3 байт данных в объект сообщения
				0 – Данные не меняются
[0]	Data B	ЗП/ЧТ	Доступ к 4-7 байтам данных	1 - Передача с 4 по 7 байт данных в объект сообщения
				0 - Данные не меняются
Чтение*				
[6]	Mask	ЗП/ЧТ	Доступ к битам маски	1 – В буферные регистры блоков IFx передаются биты: Identifier Mask + MDir + MXtd
				0 – Биты маски не меняются
[5]	Arb	ЗП/ЧТ	Доступ к битам арбитража	1 - В буферные регистры блоков IFx передаются биты: Identifier + Dir + Xtd + MsgVal
				0 - Биты арбитража не меняются
[4]	Control	ЗП/ЧТ	Доступ к битам управления	1 – Биты управления передаются буферные регистры блоков IFx
				0 – Биты остаются неизменными
[3]	ClrIntPnd	ЗП/ЧТ	Сброс бита ожидающего прерывания	1 – сброс бита IntPnd в объекте сообщения
				0 - бита IntPnd не меняется
[2]	TxRqst/NewDat	ЗП/ЧТ	Доступ к биту новых данных	1 – сброс бита NewDat в объекте сообщения
				0 – значение бита NewDat остается неизменным Примечание: Объект сообщения, доступный для чтения, так же может быть доступным для сброса управляющих бит IntPnd и NewDat. Значения этих бит, переданных в управляющие регистры блоков IFx определяют состояние контроллера перед сбросом.
[1]	Data A	ЗП/ЧТ	Доступ к 0-3 байтам	1 – Передача с 0 по 3 байт данных буферные реги-

			данных	стры блоков IFx
				0 – Данные не меняются
[0]	Data B	ЗП/ЧТ	Доступ к 4-7 байтам данных	1 - Передача с 4 по 7 байт данных буферные регистры блоков IFx
				0 - Данные не меняются

Примечание: Функциональное значение бит [6:0] зависит от направления передачи данных.

5.18.4.4 Буферные регистры блоков IFx

Биты буферных регистров блоков IFx зеркально отражаются в памяти в формате объекта сообщения.

5.18.4.4.1 Регистры маски блоков IFx (IFx Mask Registers)

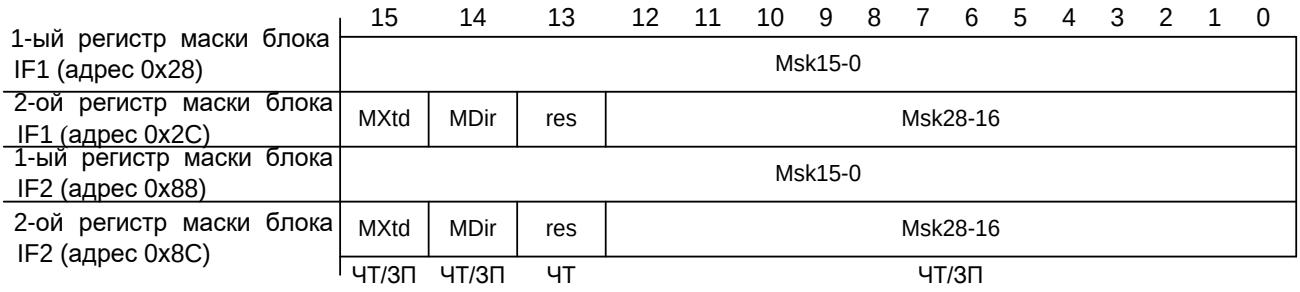


Рисунок 5.74 - Буферные регистры блоков IFx

5.18.4.4.2 Регистры арбитража блоков IFx

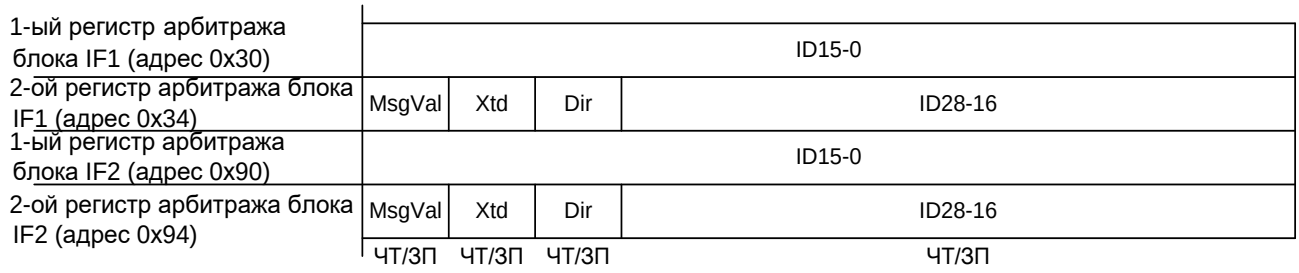


Рисунок 5.75 - Регистры арбитража блоков IFx

5.18.4.4.3 Регистры управления блоков IFx

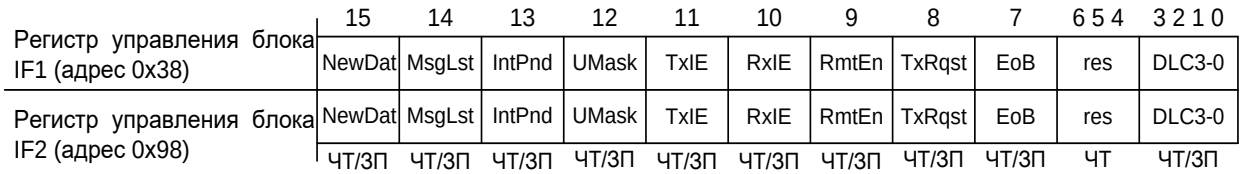


Рисунок 5.76 - Регистры управления блоков IFx

5.18.4.4.4 Регистры данных Data A и Data B

Байты данных сообщений хранятся в буферных регистрах блоков IFx в следующем порядке:

	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Регистр данных Data A1 (адрес 0x3C)	Data(1)	Data(0)
Регистр данных Data A2 (адрес 0x40)	Data(3)	Data(2)
Регистр данных Data B1 (адрес 0x44)	Data(5)	Data(4)
Регистр данных Data B2 (адрес 0x48)	Data(7)	Data(6)
Регистр данных Data A1 (адрес 0x9C)	Data(1)	Data(0)
Регистр данных Data A2 (адрес 0xA0)	Data(3)	Data(2)
Регистр данных Data B1 (адрес 0xA4)	Data(5)	Data(4)
Регистр данных Data B1 (адрес 0xA8)	Data(7)	Data(6)
	ЧТ/ЗП	ЧТ/ЗП

Рисунок 5.77 - Структура байт данных, хранящихся в буферных регистрах блоков IFx

По спецификации CAN шины, если передается/принимается кадр данных, то сначала передается/принимается нулевой байт данных, последним передается/принимается седьмой байт данных.

5.18.4.5 Объект сообщений, хранимый в памяти

В памяти хранится 32 объекта памяти. Процессор не может напрямую обращаться к объекту сообщений, хранящихся в памяти Message RAM. Для предотвращения конфликтов доступа к памяти между процессором и контроллером CAN, поэтому доступ процессором реализуется посредством двух блоков IFx. На рисунке (см ниже) приведена структура объекта сообщения.

Message Object												
UMask	**Msk28-0**	**MXtd**	**MDir**	**EoB**	**NewDat**		**MsgLst**	**RxIE**	**TxIE**	**IntPnd**	**RmtEn**	**TxRqst**
MsgVal	**ID28-0**	**Xtd**	**Dir**	**DLC3-0**	**Data 0**	**Data 1**	**Data 2**	**Data 3**	**Data 4**	**Data 5**	**Data 6**	**Data 7**

Рисунок 5.78 - Структура объекта сообщения

Таблица 5.194 - Структура Буферные регистры блоков IFx

Название	Функция	Описание	Примечание
MsgVal	Действительность сообщения	1 – Объект сообщений сконфигурирован, может обрабатываться блоком Message Handler 0 – Сообщение игнорируется блоком Message Handler	Бит должен быть сброшен процессором во всех неиспользуемых объектах сообщений при инициализации. Прежде чем будет сброшен бит **Init** в регистре управления контроллера CAN. Так же этот бит должен быть сброшен, прежде чем будут изменены следующие поля: Идентификатор **Id28-0**, управляющие биты **Xtd**, **Dir**, **DLC3-0**.
UMask	Использование маски при процедуре фильтрации	1 – Используется маска при процедуре фильтрации (**Msk28-0**, **MXtd** и **MDir**) 0 – Маска игнорируется	Если бит установлен в единицу, то биты маски объекта сообщения должны быть программно настроены в течение инициализации, прежде чем будет установлен бит MsgVal.
ID28-0	Идентификатор сообщения	ID28- ID0: 29-битный идентификатор (расширенный формат данных); ID28- ID18: 11-битный идентификатор (стандартный формат данных);	-
Msk28-0	Маска идентификатора	1 – соответствующий бит идентификатора используется при фильтрации 0 - соответствующий бит идентификатора в объекте сообщений не сравнивается при фильтрации.	-
Xtd	Расширенный идентификатор	1 – В объекте сообщений используется 29-битный расширенный идентификатор 0 - В объекте сообщений используется 119-битный стандартный идентификатор	-
MXtd	Маска расширенного идентификатора	1 – соответствующий бит маски расширенного идентификатора используется при фильтрации 0 - соответствующий бит маски расширенного идентификатора не сравнивается при фильтрации.	Когда используется стандартный 11-битный идентификатор в объекте сообщений, тогда идентификаторы принятых кадров данных записываются в [28-18] биты. На вход фильтра поступают именно эти биты, вместе с битами маски.
Dir	Направление сообщения	1- Передача. Если установлен **TxRqst** соответствующее сообщение будет передано. При принятии пакета удаленного запроса данных с соответствующим идентификатором, будут установлены, биты **TxRqst** и **RmtEn** данного буфера. 0 - Прием. Если установлен TxRqst, то будет послан пакет удаленного запроса данных. При принятии сообще-	

		ния с соответствующим идентификатором, сообщение будет сохранено в этот буфер приема.	
MDir	Маска направления сообщения	1 –бит Dir используется при фильтрации	-
		0 - бит Dir не сравнивается при фильтрации.	
EoB	Признак последнего данного в FIFO Buffer	Устанавливается в «1», если задан только один объект сообщения, или если объект сообщения является последним, хранимым в FIFO Buffer.	Этот бит используется для организации очереди в FIFO Buffer для объектов сообщения. Если задан только один объект сообщения, то бит должен быть установлен в «1».
		Устанавливается в «0» для всех объектов, хранимых в FIFO Buffer, кроме последнего.	
NewDat	Признак новых данных	1 – Данные в объекте сообщений были изменены либо блоком Message Handler, либо процессором	-
		0 – не было изменений данных с того момента, как в последний раз был сброшен данный бит.	
MsgLst	Сообщение потеряно (только для объекта сообщения, настроенного на прием)	Показывает, что блоком Message Handler было сохранено новое сообщение в данный объект сообщения, когда был установлен бит NEWDAT, т.е. предыдущий кадр был потерян.	-
		0 – Не было потери данных	
RxIE	Разрешает прерывание после приема	Определяет, установится ли бит IntPnd после успешного принятия сообщения	-
TxIE	Разрешение прерывания после передачи	Определяет, установится ли бит IntPnd после успешной передачи.	-
IntPnd	Отложенные прерывания	1 – данный объект сообщения является источником прерывания.	Идентификатор прерывания в регистре прерывания укажет на этот объект сообщения, если не было других боле высокоприоритетных прерываний.
		0 - данный объект сообщения не является источником прерывания.	
RmtEn	Признак приема кадра удаленного запроса	1 – При приеме кадра удаленного запроса, устанавливается бит TxRqst	-
		0 - При приеме кадра удаленного запроса, бит TxRqst не изменяется	
TxRqst	Запрос на передачу	1 - Показывает, что требуется передача данного сообщения микроконтроллером или удаленным узлом.	-

		0 – Сообщении больше не ожидает отправки	
DLC3-0	Код длины данных	Действительные значения 0…8	DLC объекта сообщения должен совпадать с DLC всех других объектов, посланных от разных удаленных узлов и имеющих такой же идентификатор. При сохранении кадра данных поле DLC прописывается значением полученного сообщения.
Data 0	0 байт данных	Первый байт данных кадра данных	Передача данных из сдвигового регистра начинается с Data 0, Data 7 передается последним. Кадр данных полностью сохраняется в объекте сообщений. Если DLC задана меньше 8, то остальные байты перезаписываются неопределенными сообщениями
Data 1	1 байт данных	Второй байт данных кадра данных	
Data 2	2 байт данных	Третий байт данных кадра данных	
Data 3	3 байт данных	Четвертый байт данных кадра данных	
Data 4	4 байт данных	Пятый байт данных кадра данных	
Data 5	5 байт данных	шестой байт данных кадра данных	
Data 6	6 байт данных	Седьмой байт данных кадра данных	
Data 7	7 байт данных	Восьмой байт данных кадра данных	

Значения, прописанные в полях регистров арбитража (ID28-0, Xtd, Dir), определяют идентификатор и направление сообщения (прием, или передача). Также совместно со значениями полей регистров маски (Msk28-0, MXtd, MDir), они используются при процедуре фильтрации. Полученное сообщение хранится в объекте сообщения, имеющего такой же идентификатор, что и сообщение. Также в зависимости от направления сообщения хранятся полученные данные: если прием, то сообщение имеет формат кадра данных; если передача, то сообщение имеет формат удаленного запроса данных. Расширенный формат идентификатора сохраняется в объекте сообщений только при Xtd=1, для стандартного формат идентификатора Xtd=0. Если принятое сообщение совпадает более чем с одним действительным объектом сообщения, оно сохраняется с более низким номером сообщения

5.18.4.6 Регистры блока Message Handler

Все регистры данного блока доступны только по чтению. Содержимое регистров показывает внутреннее состояние работы CAN контроллера.

5.18.4.6.1 Регистр прерывания

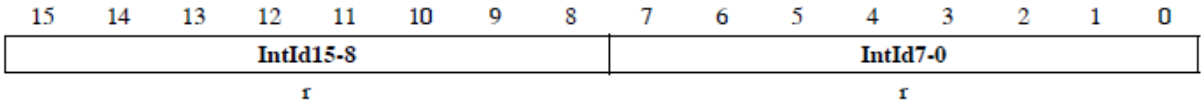


Рисунок 5.79 - Регистр прерывания

Таблица 5.195 - Структура регистра прерывания

Биты	Название	Тип	Функция	Описание
[15:0]	IntId	Чт	Идентификатор прерывания	0x0000: нет ожидающих прерываний
				0x0001-0x0020: Номер объекта, где возникло прерывание
				0x0021-0x7FFF: не используется
				0x8000: статусное прерывание
				0x8001-0xFFFF: не используется

Для того чтобы определить причину прерывания, необходимо прочесть регистр статуса прерывания (CAN Interrupt Status Register). Идентификатор прерывания IntId в регистре прерывания показывает причину прерывания. Если прерывания нет, значение INTID равно 0.

Прерывания сбрасываются процессором. Если значение поля IntId отлично от 0x0000, устанавливается бит IE в управляющем регистре, сигнал прерывания (IRQ_B) от CAN – модуля вызовет соответствующий обработчик прерывания. Сигнал прерывания остается активным пока значение INTID не равно 0, или пока не отключена генерация данного прерывания (CSR.IE=0).

Статусное прерывание является самым высокоприоритетным. Прерывания с наименьшим номером обладают наибольшим приоритетом.

Значение IntId будет также модифицировано, если обработался соответствующий запрос. Это может произойти, если сбросить флаг IntPnd данного объекта сообщения, или прочесть состояние в регистре управления CAN контроллера (в случае если прерывание вызвано изменением регистра состояния).

Прерывание статуса сбрасывается при чтении регистра статуса.

5.18.4.6.2 Регистр запроса передачи

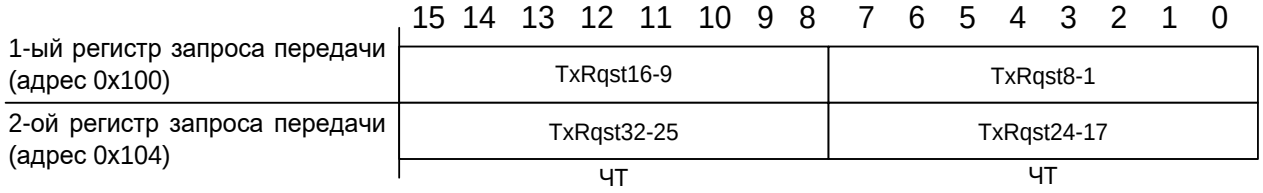


Рисунок 5.80 - Регистр прерывания

Таблица 5.196 - Структура регистра прерывания

Биты	Название	Тип	Функция	Описание
[31:0]	TxRqst32-1	Чт	Биты запроса передачи	1 - Показывает, что требуется передача данного сообщения микроконтроллером или удаленным узлом.
				0 – Сообщении больше не ожидает отправки

В регистре прописываются биты TxRqst всех 32 объектов сообщения. По установленным битам данного регистра, определяется, какой объект сообщения ожидает отправки. Биты TxRqst могут быть программно установлены или сброшены либо процессором, либо CAN контролером после получения кадра удаленного запроса, либо после успешной передачи данных.

5.18.4.6.3 Регистр новых данных

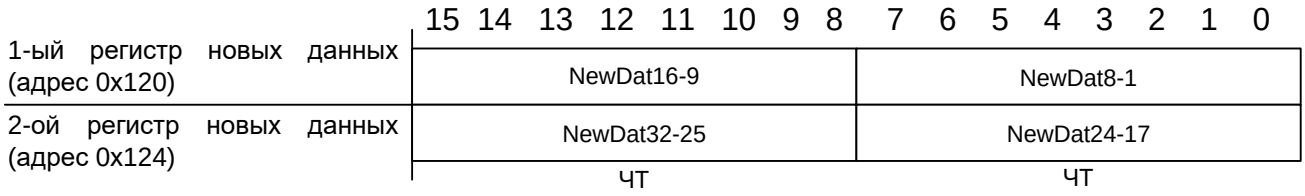


Рисунок 5.81 - Регистр новых данных

Таблица 5.197 - Структура регистра новых данных

Биты	Название	Тип	Функция	Описание
[31:0]	NewDat32-1	Чт	Биты новых данных	1 – Данные в объекте сообщений были изменены либо блоком Message Handler, либо процессором
				0 – не было изменений данных с того момента, как в последний раз был сброшен данный бит.

В регистре прописываются биты NewDat всех 32 объектов сообщения. По установленным битам данного регистра определяется какие данные были модифицированы. Биты NewDat могут быть программно установлены, или сброшены либо процессором, либо CAN контролером после получения кадра данных, либо после успешной передачи данных.

5.18.4.6.4 Регистры отложенных прерываний

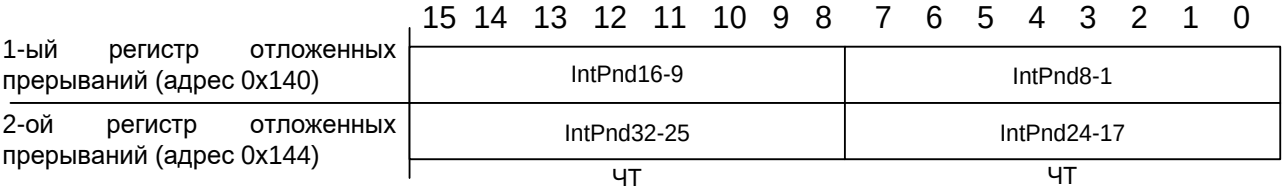


Рисунок 5.82 - Регистры отложенных прерываний

Таблица 5.198 - Структура регистров отложенных прерываний

Биты	**Название**	**Тип**	**Функция**	**Описание**
[31:0]	IntPnd32-1	Чт	Биты новых данных	1 – данный объект сообщения является источником прерывания.
				0 - данный объект сообщения не является источником прерывания.

В регистре прописываются биты **IntPnd** всех 32 объектов сообщения. По установленным битам данного регистра определяется в каком объекте сообщения возникло прерывание. Биты **IntPnd** могут быть программно установлены или сброшены либо процессором, либо CAN контролером после получения кадра данных, либо после успешной передачи данных. Поле **IntId** регистра прерываний прописывается номером объекта сообщения.

5.18.4.6.5 Регистр признаков действительности объектов сообщений

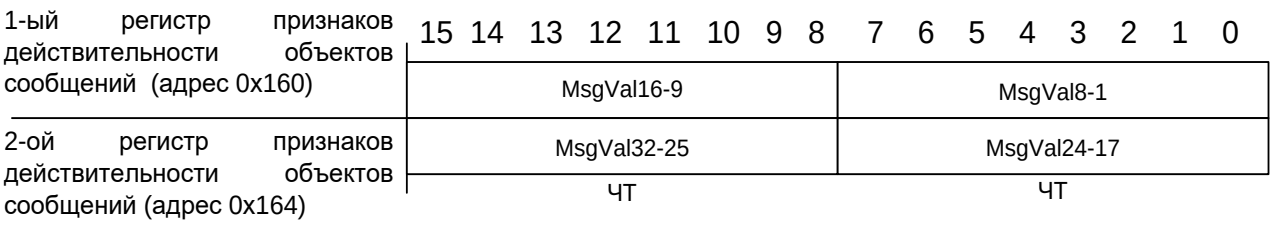


Рисунок 5.83 - Регистр признаков действительности объектов сообщений

Таблица 5.199 - Структура регистров признаков действительности объектов сообщений

Биты	Название	Тип	Функция	Описание
[31:0]	MsgVal32-1	Чт	Действительности данных	1 – Объект сообщений сконфигурирован, может обрабатываться блоком Message Handler
				0 – Сообщение игнорируется блоком Message Handler

В регистре прописываются биты MsgVal всех 32 объектов сообщения. По установленным битам данного регистра определяется какой объект сообщения является действительным. Бит MsgVal отдельного объекта сообщения может быть программно установлен/сброшен процессором (через IFx Message Interface регистры).

5.19 Контроллер интерфейса I2C

Контролер полностью соответствует спецификации шины I2C. Блок контроллера интерфейса I2C в СБИС СНП-ВП имеет следующие характеристики:

- Ведущий может выступать как в роли ведущего-передатчика, так и в роли ведущего-приемника;
- Ведомый может выступать как в роли ведомого-передатчика, так и в роли ведомого-приемника;
- Для устранения коллизий, которые могут возникать при подключении более одного ведущего, разработана процедура арбитража;
- Поддерживается процедура синхронизации устройств;
- Обнаружение адреса Slave-устройства (поддерживается только 7-битная адресация);
- Обнаружение адреса общего вызова;
- Опознавание направления передачи;
- Повторение генерации и обнаружение состояния START
- Обнаружение ошибок на шине;
- Поддерживается стандартный режим (100 кбит/с) и быстрый режим работы (400 кбит/с);
- Не поддерживается совместимость с CBUS-устройствами.
- Подключение к системной шине СБИС через AMBA APB интерфейс

5.19.1 Устройство контроллера интерфейса I2C

I2C – двунаправленная шина с последовательной передачей данных. При передаче данных одно устройство является «ведущим», которое инициирует передачу данных и формирует сигналы синхронизации. Другое устройство «ведомое», которое может начать передачу данных только по команде ведущего шины. Основные термины, связанные с шиной I2C, представлены в таблице ниже.

Таблица 5.200 - Основные термины, связанные с шиной I2C

Термин	Описание
Передатчик	Устройство, передающее данные по шине I2C.
Приемник	Устройство, принимающее данные с шины I2C.
Ведущий (Master)	Устройство, инициирующее передачу данных, формирующее тактовый сигнал и заканчивающее передачу данных
Ведомый (Slave)	Устройство, к которому обращается ведущий.
Многомастерная сеть (Multi-master)	Более чем один ведущий на шине. Несколько ведущих могут пытаться передать данные без разрушения текущего сообщения.
Арбитраж	Процедура, гарантирующая, что, если больше чем один ведущий одновременно пытается управлять шиной, только один из них получит полное управление шиной и передача данных будет осуществлена без ошибок
Синхронизация	Процедура синхронизации тактовых сигналов от двух или более устройств на шине

Физическая шина I2C представляет собой две сигнальные линии, одна из которых (SCL) предназначена для передачи тактового сигнала, а вторая (SDA) для обмена данными. В таблице ниже приведен список внешних выводов СБИС, относящихся к контроллеру I2C:

Таблица 5.201-- Выводы микросхемы, относящиеся к контроллеру I2C

Вывод	Тип буфера	Примечание
SDA	inout	Последовательная линия данных I2C
SCL	inout	Линия синхронизации I2C

На рисунке ниже (см. Рисунок 5.84) приведена внутренняя структура контроллера I2C.

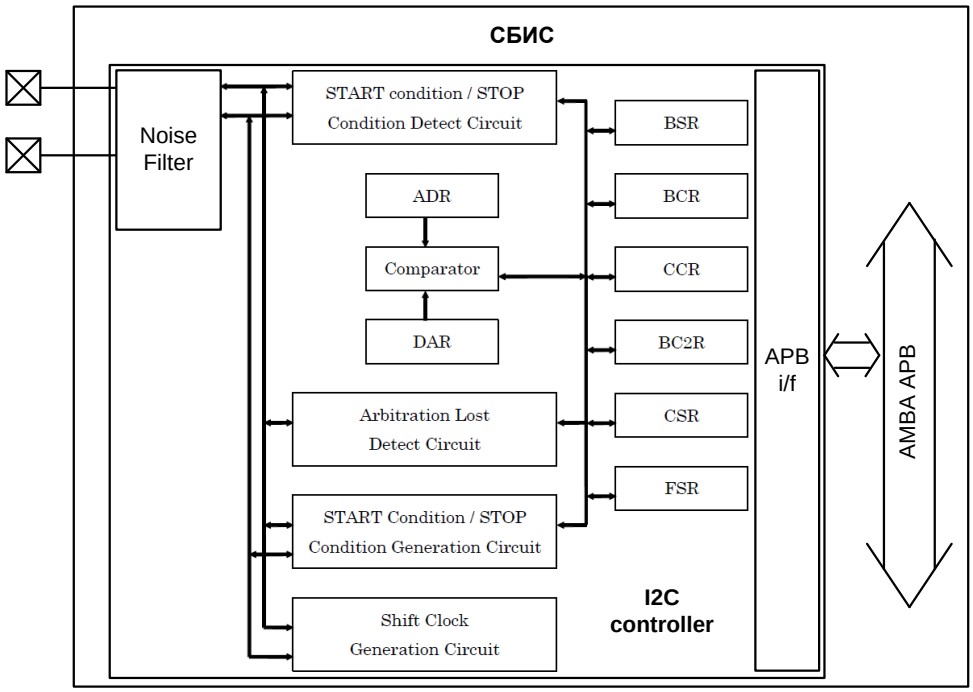


Рисунок 5.84 – Внутренняя структура контроллера I2C

START condition/STOP condition detect circuit - блок, который в зависимости от изменения состояний сигнальных линий SDA и SCL, распознает START и STOP состояния.

START condition/STOP condition generation circuit- блок, который изменяет состояние сигнальных линий, для формирования состояний START и STOP.

Arbitration lost detect circuit - блок, отслеживающий проигрыш арбитража. Выходные данные, отправленные по шине SDA, сравниваются с принятыми данными по входной шине SDA. Если они не совпадают, то формируется признак потери шины.

Shift clock generation circuit - блок, формирующий необходимое количество тактов для передачи последовательных данных, а также контролирующий количество тактов сигнала SCL. Значение времени задается в регистре управления синхросигналом (CCR).

Comparator - компаратор, который проверяет принятый адрес на соответствие значению адреса регистра или на соответствие значению глобальному адресу.

ADR: 7-битный регистр, используется для задания адреса SLAVE.

DAR: 8-битный регистр, используется для последовательной передачи данных.

BSR: 8-битный регистр, определяет состояние шины I2C. В данном регистре отслеживаются следующие признаки:

- Повторное определение состояния START;
- Обнаружение проигрыша арбитража;
- Подтверждение передачи данных;
- Распознавание адреса;
- Обнаружение адреса общего вызова;

- Распознавание первого байта.

BCR: 8-битный регистр, управляет шиной I2C и прерываниями. В данном регистре отслеживаются следующие признаки:

- Запрос прерываний/ разрешение прерываний;
- Формирование состояния START;
- Определение состояния Master/Slave;
- Формирование разрешения подтверждения.

CCR: 7-битный регистр, в котором задается частота, с которой будут последовательно переданы данные. В данном регистре отслеживаются следующие признаки:

- Разрешение операции передачи данных;
- Установка частоты синхросигнала;
- Выбор режима работы: стандартный, или быстрый.

Noise filter: блок фильтрации помех.

BC2R: С помощью данного регистра проверяется состояние линии после прохождения фильтра помех, принудительно устанавливает линию в состояние логического нуля «L».

CSR: Регистр расширения регистра CCR, определяющий биты делителя частоты CS.

FSR: В этом регистре задается частота синхросигнала шины APB, для СБИС СНП-ВП эта частота F_{HPMP}=80МГц по умолчанию.

5.19.2 Программная модель контроллера интерфейса I2C

Программно доступные регистры контроллера шины I2C расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение I2C Base = 0xFFF04000 и общий размер 4 Кб. Регистры имеют разрядность 8 бит. Спецификация регистров представлена в таблице ниже:

Таблица 5.202 - Сводная таблица регистров контроллера интерфейса I2C

Адрес	Тип доступа	Обозначение	Описание
Base + 0x00	ЧТ	BSR	Регистр статуса шины
Base + 0x04	ЧТ/ЗП	BCR	Регистр управления шиной
Base + 0x08	ЧТ/ЗП	CCR	Регистр управления тактовым сигналом
Base + 0x0C	ЧТ/ЗП	ADR	Регистр адреса
Base + 0x10	ЧТ/ЗП	DAR	Регистр данных
Base + 0x14	ЧТ/ЗП	CSR	Регистр определения дополнительных бит для управления периодом тактового сигнала.
Base + 0x18	ЧТ/ЗП	FSR	Регистр определяющий внутреннюю частоту тактового сигнала контроллера
Base + 0x1C	ЧТ/ЗП	BC2R	Дополнительный регистр управления шиной.
Base + 0x20 to 0xFFC	-	Зарезервировано	-

5.19.2.1 Регистр *статуса* шины (BSR)

D7	D6	D5	D4	D3	D2	D1	D0
BB	RSC	AL	LRB	TRX	AAS	GCA	FBT

Рисунок 5.85 - Регистр статуса шины

Таблица 5.203 – Структура регистра BSR

Биты	Название	Значение при сбросе	Функция
[7]	BB	0	Признак занятости шины. 0 – Обнаружено состояние STOP
			1 - Обнаружено состояние START (Шина занята)
[6]	RSC	0	Признак повторения состояния START. 0 – повторяющее состояние START не обнаружено
			1- Пока шина занята, выявляется повторяющее состояние START.
			Примечание: Данный бит может быть сброшен, если при установленном бите INT произошли следующие события: Бит INT был сброшен; Если в режиме Slave не было обращения по адресу; При обнаружении состояния START, когда контроллер находится в нерабочем состоянии; При обнаружении состояния STOP.
[5]	AL	0	Признак проигрыша арбитража. 0 - Не обнаружен проигрыш арбитража
			1 – В процессе передачи данных устройством, находящимся в состоянии Master, был проигран арбитраж, или другое устройство использовало шину, бит MSS был прописан значением «1».
			Примечание: Если биты AL и MSS прописаны «1», то Master не происводит передачу данных. Этот бит сбрасывается если установленный бит INT был программно сброшен.
[4]	LRB	0	0 – Получено подтверждение (ACK).
			1 – Подтверждение не получено (NAK).
			Примечание: Данный бит сбрасывается при обнаружении состояния START или STOP.
[3]	TRX	0	Данный бит определяет состояние передаваемых данных. 0 – Принимаемые данные.
			1 – Передаваемые данные.
[2]	AAS	0	Признак обнаружения адреса. 0 – В режиме Slave не было обращения по адресу.
			1 - В режиме Slave было обращение по адресу.
			Примечание: Данный бит сбрасывается при обнаружении состояния START или STOP.
[1]	GCA	0	Признак обнаружения адреса общего вызова. 0 - В режиме Slave не было получено адреса общего вызова.
			1 - В режиме Slave был получен адрес общего вызова.
			Примечание: Данный бит сбрасывается при обнаружении состояния START или STOP.
[0]	FBT	0	Признак первого переданного байта. 0 - Полученные данные не являются 1-ым байтом.
			1 - Полученные данные являются 1-ым байтом (адрес данных).
			Примечание: Даже если обнаружено состояние START, данный бит устанавливается в «1». Бит FBT сбрасывается, если: Установленный бит INT программно сбрасывается, В режиме Slave не было обращения по адресу.

5.19.2.2 Регистр управления шиной (BCR)

D7	D6	D5	D4	D3	D2	D1	D0
BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT

Рисунок 5.86 - Регистр управления шиной

Таблица 5.204– Структура регистра BCR

Биты	Название	Значение при сбросе	Функция
[7]	BER	0	Флаг запроса прерывания, формирующегося при возникновении ошибки на шине. При записи «0» – Сброс флага. При чтении «0» – Ошибка на шине не обнаружена.
			При записи «1» - не имеет смысла. При чтении «1» - Если в процессе передачи данных обнаруживается некорректное состояние START, или STOP.
			Примечание: Если установлен данный бит, то бит EN регистра CCR сбрасывается, контроллер прекращает работу, прерывается передача данных.
[6]	BEIE	0	Разрешение прерывания, формирующегося при возникновении ошибки на шине. 0 – прерывание не разрешено.
			1- прерывание разрешено.
			Примечание: Прерывание формируется, если данный бит и BER установлены в «1».
[5]	SCC	0	Сформировано состояние START при записи «0» – не имеет значения.
			при записи «1» - В режиме Master состояние START заново формируется. Бит автоматически сбрасывается, после его установки.
[4]	MSS	0	Признак работы контроллера либо в состоянии Master, либо в состоянии Slave. 0 – После того, как сформировано состояние STOP, прекращается передача данных, контроллер переходит в состояние Slave.
			1 – Контроллер переходит в состояние Master, когда сформировано состояние START, начинает передачу данных.
			Примечание: Если в состоянии Master при передаче данных был проигран арбитраж, то данный бит сбрасывается и контроллер переходит в состояние Slave. В случае «многомастерной» сети, запрещается контроллеру, а также другим передатчикам одновременно передавать адрес общего вызова, а также запрещается потеря арбитража при передаче 2-ого байта, или следующих после него байтов.
[3]	ACK	0	Разрешение подтверждения принимаемых данных. 0 – Подтверждение не было сформировано.
			1 - Подтверждение сформировано.
			Примечание: В состоянии Slave при приеме адреса данных данный бит не активен.
[2]	GCAA	0	Разрешение подтверждения принимаемого адреса общего вызова. 0 - Подтверждение не было сформировано.
			1 - Подтверждение сформировано.
			Примечание: Если значение данного бита равно «0», то даже если будет получен адрес общего вызова, передача, а так же прерывание не будут сформированы.
[1]	INTE	0	Разрешение прерывания. 0 – прерывание не разрешено.
			1 – прерывание разрешено.
			Примечание: Если данный бит и бит INT установлены, то будет сформировано прерывание по окончаню передачи.
[0]	INT	0	Флаг запроса прерывания по окончанию передачи. При записи «0» - сброс флага запроса прерывания по окончанию передачи. При чтении «0» - Передача еще не закончена.
			При записи «1» - не имеет смысла. При чтении «1». Данный бит может быть установлен, если закончена

			передача одного байта и получено подтверждение (ACK), а так же если выполнено одно из следующих условий: Контроллер управляет шиной; В приемнике произошло совпадение при обращении по адресу; Был получен адрес общего вызова (Только когда GCAA=1); Проигрыш арбитража; Есди другой узел системы использует шину, при этом обнаруживается состояние START;
			Примечание: Если данный бит установлен, то линия SCL удерживается в состоянии логического нуля. При сбросе бита INT линия SCL освобождается для последующей передачи байтов. Данный бит сбрасывается: В состоянии Master, при установки бита SCC для повторного формирования состояния START; При сбросе бита MSS для формирования состояния STOP.

Если биты SCC, MSS и INT одновременно прописываются, то возникает неопределенность в том, какое событие должно быть сформировано: передача следующего байта, формирование START или формирование STOP. Для разрешения неопределенности введена следующая приоритетность действий:

- При одновременной записи INT=0 и MSS=0 – больший приоритет у записи бита MSS, т.е. будет сформировано состояние STOP;
- При одновременной записи INT=0 и SCC=1 – больший приоритет у записи бита SCC, т.е. будет сформировано состояние START;
- Запрещено одновременно прописывать SCC=1 и MSS=0.

5.19.2.3 Регистр управления тактовым сигналом (CCR)

D7	D6	D5	D4	D3	D2	D1	D0
—	FM	EN	CS4	CS3	CS2	CS1	CS0

Рисунок 5.87 - Регистр управления тактовым сигналом

Таблица 5.205– Структура регистра CCR

Биты	Название	Значение при сбросе	Функция
[7]	-	-	При чтении всегда возвращает «1»
[6]	FM	0	Установка быстрого режима работы. 0 – Стандартный режим работы
			1 – Быстрый режим работы.
[5]	EN	0	Разрешение проведения операций. 0 – Операция не разрешена.
			1 - Операция разрешена.
			Примечание: Сброс данного бита, сбрасывает регистры BCR и BSR, за исключением битов BER и BEIE.
[4:0]	CS4-0	-	Биты устанавливают интервал частот для формирования тактовых сигналов. Данные тактовые сигналы формируются путем деления частоты системного тактового сигнала. Конфигурация регистра CSR позволяет изменять значение системного синхросигнала. Если регистр CSR не используется, частота тактового сигнала (fscl) и настройка делителя (m) приведены ниже.

Частота шины I2C вычисляется по формулам:

F_{I2C}(МГц) = $F_{НРМР}$(МГц)/[2*m+2] (Стандартный режим F_{I2C}(макс)=100 кбит/с)
F_{I2C}(МГц) = $F_{НРМР}$(МГц)/[int(1.5*m)+2] (Быстрый режим F_{I2C}(макс)=400 кбит/с)

где m= 9…96. В таблице ниже (Таблица 5.206) приведена связь между значением делителя m и значениями CS[10:0], при этом в битах CS10-CS5 – нули (регистр CSR – не используется).

Пример расчета для стандартного режима:
Системная частота контроллера FHPMP = 14 МГц.
Регистр CSR не используется. (CS10-5: 6'b000000).
fscl (МГц) = 14(МГц)/((2*69)+2);
m=69.
fscl (МГц) = 0,1 МГц = 100 кГц.
CS4-0 = 5'b00100.
Пример расчета для быстрого режима:
Системная частота контроллера FHPMP = 14 МГц.
Регистр CSR не используется. (CS10-5: 6'b000000).
fscl (МГц) = 14(МГц)/((1.5*22)+2);
m=22.
fscl (МГц) = 400 кГц.
CS4-0 = 5'b10101.

Таблица 5.206 - Конфигурация тактовых сигналов CS4-0 при стандартном и быстром режиме работы контроллера

*CS10-CS5	CS4	CS3	CS2	CS1	CS0	m	
						FM=0 (Стандартный режим)	FM=1 (Ускоренный режим)
000000	0	0	0	0	0	65	Запрещена установка
000000	0	0	0	0	1	66	Запрещена установка
000000	0	0	0	1	0	67	Запрещена установка
000000	0	0	0	1	1	68	Запрещена установка
000000	0	0	1	0	0	69	Запрещена установка
000000	0	0	1	0	1	70	Запрещена установка
000000	0	0	1	1	0	71	Запрещена установка
000000	0	0	1	1	1	72	Запрещена установка
000000	0	1	0	0	0	73	9
000000	0	1	0	0	1	74	10
000000	0	1	0	1	0	75	11
000000	0	1	0	1	1	76	12
000000	0	1	1	0	0	77	13
000000	0	1	1	0	1	78	14
000000	0	1	1	1	0	79	15
000000	0	1	1	1	1	80	16
000000	1	0	0	0	0	81	17
000000	1	0	0	0	1	82	18
000000	1	0	0	1	0	83	19
000000	1	0	0	1	1	84	20
000000	1	0	1	0	0	85	21
000000	1	0	1	0	1	86	22
000000	1	0	1	1	0	87	23
000000	1	0	1	1	1	88	24
000000	1	1	0	0	0	89	25
000000	1	1	0	0	1	90	26
000000	1	1	0	1	0	91	27
000000	1	1	0	1	1	92	28
000000	1	1	1	0	0	93	29
000000	1	1	1	0	1	94	30
000000	1	1	1	1	0	95	31
000000	1	1	1	1	1	96	32

** Значения CS10-CS5 устанавливаются в регистре CSR.*

*** Диапазон, которому принадлежит системная частота контроллера FHPMP, задается в регистре FSR (описан ниже).*

5.19.2.4 Регистр адреса (ADR)

D7	D6	D5	D4	D3	D2	D1	D0
—	A6	A5	A4	A3	A2	A1	A0

Рисунок 5.88 - Регистр адреса

Таблица 5.207 – Структура регистра ADR

Биты	Название	Значение при сбросе	Функция
[7]	-	-	При чтении всегда возвращает «1»
[6:0]	A6-0	-	Адрес Slave. В состоянии Slave, после получения адреса данных, данный регистр сравнивается с регистром DAR. Если происходит совпадение значений, то Master посылается подтверждение (ACK).

5.19.2.5 Регистр данных (DAR)

D7	D6	D5	D4	D3	D2	D1	D0
D7	D6	D5	D4	D3	D2	D1	D0

Рисунок 5.89 - Регистр данных

Таблица 5.208 – Структура регистра DAR

Биты	Название	Функция
[7:0]	D7-0	Биты данных. Данные передаются в порядке MSB. При записи данные попадают в буфер глубиной равной двум. Если установлен признак занятости шины (BB=1), данные записываются в регистры для последующей побайтной передачи данных. Чтение данных из регистров возможно когда установлен бит INT.

5.19.2.6 Регистр определения дополнительных бит для управления периодом тактового сигнала (CSR)

D7	D6	D5	D4	D3	D2	D1	D0
TST1	TST0	CS10	CS9	CS8	CS7	CS6	CS5

Рисунок 5.90 - Регистр определения дополнительных бит для управления периодом тактового сигнала

Таблица 5.209 – Структура регистра CSR

Биты	Название	Значение при сбросе	Функция
[7:6]	[TST1: TST0]	00	0 – нормальный режим; 1 – тестовый режим.
[5:0]	CS10-5	6’b000000	Данные биты расширяют формат бит CS4-0 регистра CCR. CS10-5 = 6’b000000 (значение по умолчанию). Изменение значений данных бит приводит к изменению тактовой частоты синхросигнала.

При использовании регистра CSR частота шины I2C вычисляется по формулам приведенным ниже, где m = (CS10-0+1), CS10-0 – биты из регистров CSR и CCR:

$$F_{I2C}(МГц) = F_{НРМР}(МГц)/[2*m+2] \quad (Стандартный\ режим\ F_{I2C}(макс)=100\ кбит/с)$$
$$F_{I2C}(МГц) = F_{НРМР}(МГц)/[int(1.5*m)+2] \quad (Быстрый\ режим\ F_{I2C}(макс)=400\ кбит/с)$$

5.19.2.7 Регистр, определяющий внутреннюю частоту тактового сигнала (FSR)

D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	—	FS3	FS2	FS1	FS0

Рисунок 5.91 - Регистр, определяющий внутреннюю частоту тактового сигнала

Таблица 5.210 – Формат регистра FSR

Биты	Название	Значение при сбросе	Функция
[7:4]	-	-	При чтении всегда возвращает «0000».
[3:0]	FS3-0	0001	Биты предназначены для определения внутренней частоты тактового сигнала в МГц. 0000 – установка запрещена; 0001 – 14 ≤ Частота < 20; 0010 – 20 ≤ Частота < 40; 0011 – 40 ≤ Частота < 60; 0100 – 60 ≤ Частота < 80; 0101 – 80 ≤ Частота < 100; 0110 – 100 ≤ Частота < 120; 0111 – 120 ≤ Частота < 140; 1000 – 140 ≤ Частота < 160; 1001 – 160 ≤ Частота < 180; 1010 – 180 ≤ Частота < 200; 1011 – 200 ≤ Частота < 220; 1100 – не определено; 1101 – не определено; 1110 – не определено; 1111 – не определено;

5.19.2.8 Дополнительный регистр управления шиной (BC2R)

D7	D6	D5	D4	D3	D2	D1	D0
—	—	SDAS	SCLS	—	—	SDAL	SCLL

Рисунок 5.92 - 2-ой регистр управления шиной

Таблица 5.211 – Структура регистра BC2R

Биты	Название	Значение при сбросе	Функция
[7:6]	-	-	При чтении всегда возвращает «0».
[5]	SDAS	-	Статус линии SDA. 0 – линия в состоянии логического нуля «L»; 1 – линия в состоянии логического единицы «H».
[4]	SCLS	-	Статус линии SDL. 0 – линия в состоянии логического нуля «L»; 1 – линия в состоянии логического единицы «H».
[3:2]	-	-	При чтении всегда возвращает «00».
[1]	SDAL	0	0 – выход SDAO работает в нормальном режиме; 1 – выход SDAO принудительно установлен в состояние логического нуля «L».
[0]	SCLL	0	0 – выход SDLO работает в нормальном режиме; 1 – выход SDLO принудительно установлен в состояние логического нуля «L».

5.19.3 Функционирование контроллера интерфейса I2C

5.19.3.1 Состояние START

Если установлен бит MSS, без признака занятости шины (BB=0), контроллер переходит в состояние Master, одновременно формируется состояние START. В состоянии Master, даже если шина занята (BB=1), при установке бита SCC, формируется повторение состояния START.

Два условия для формирования состояния START:

1. При незанятой шине (MSS=0, BB=0, INT=0, AL=0) устанавливается бит MSS;

2. Если в состоянии Master происходит прерывание (MSS=1, BB=1, INT=1, AL=0), то устанавливается бит SCC.

Если другие Master используют шину (MSS=0, BB=1), при этом устанавливается бит MSS, устанавливается бит AL, что означает проигрыш арбитража. Если, помимо условий 1 и 2 (перечисленных выше), происходят какие-либо другие события (даже если установлены биты MSS и CSS), то формирования состояния START не произойдет.

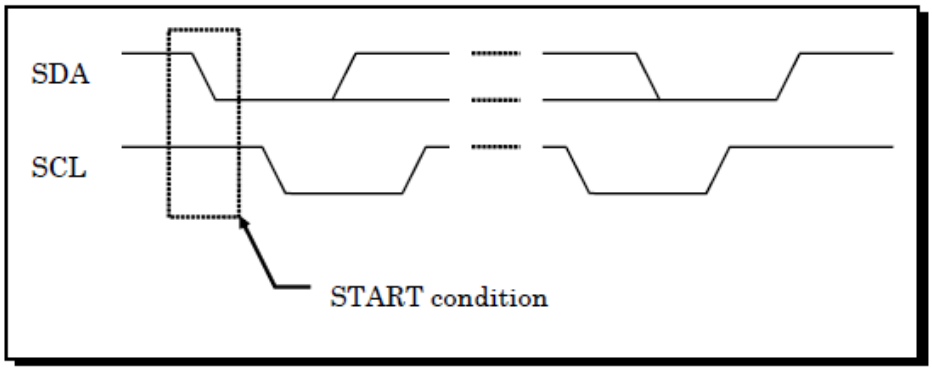


Рисунок 5.93 – Формирование состояния START на шине I2C

Сигнальная линия SDA переходит из состояния логической «1» в состояние логического «0». Сигнальная линия SCL остается в состоянии логической «1».

5.19.3.2 Состояние STOP

В режиме Master (MSS=1), при сбросе бита MSS, контроллер переходит в состояние STOP. Контроллер начинает работать в режиме Slave.

Условие формирования состояния STOP: должно произойти прерывание в состоянии Master на шине (т.е. MSS=1, BB=1, INT=1, AL=1) и сброс бита MSS. Формирование состояния STOP не может быть выполнено при других условиях.

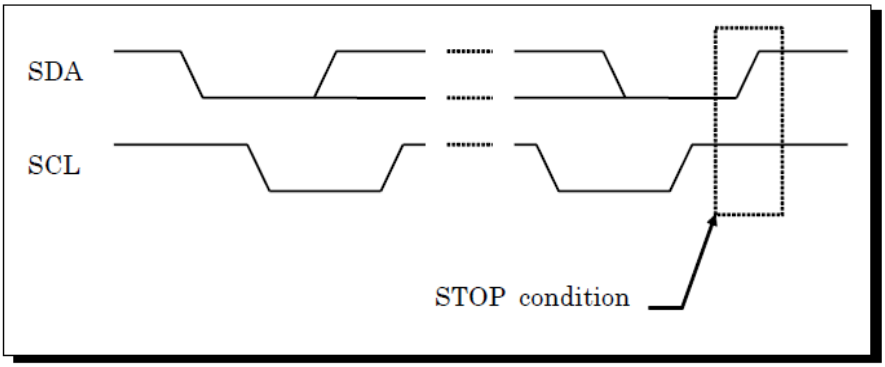


Рисунок 5.94 – Формирование состояния STOP на шине I2C

Сигнальная линия SDA переходит из состояния логической «0» в состояние логического «1». Сигнальная линия SCL остается в состоянии логической «1».

5.19.3.3 Адресация

В режиме Master, после формирования состояния START, BB устанавливается равным «1», TRX=1, данные регистра DAR поступают на шину в порядке следования MSB. После передачи адреса, должно прийти подтверждение приема адреса от Slave. После получения подтверждения, бит bit0 регистра DAR инвертируется и сохраняется, как бит TRX.

В режиме Slave, после формирования состояния START, BB устанавливается равным «1», TRX=0, принимаемые данные сохраняются в DAR регистре. После получения адреса происходит сравнение регистров DAR и ADR. Если значения совпадают, то AAS устанавливается равным «1» и формируется подтверждение приема адреса. После этого bit0 регистра DAR сохраняется, как бит TRX.

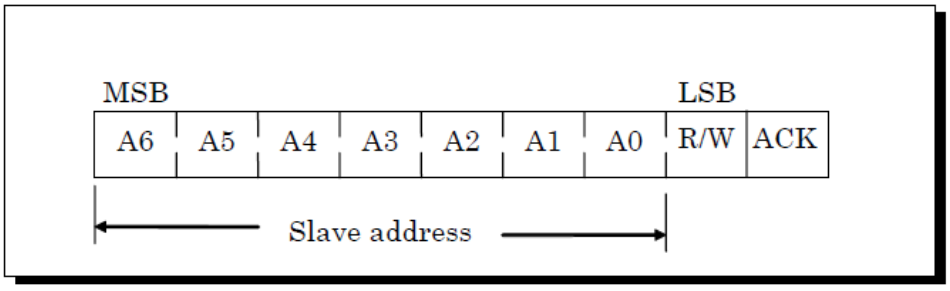


Рисунок 5.95 - Формат передачи адреса приемнику

Таблица 5.212 - Карта памяти адресов приемника

Принимаемый адрес	R/W	Пояснение
0000000	0	Адрес общего вызова
0000000	1	Байт состояния START
0000001	X	Адрес CBUS
0000010	X	Зарезервировано
0000011	X	Зарезервировано
00001XX	X	Код передатчика (используется в режиме скоростной передачи) [1*]
0001XXX – 1110XXX	X	Доступный адрес приемника
11110XX	X	10-битный адрес приемника[1*]
11111XX	X	Зарезервировано

Примечание:
Контроллер не поддерживает режим скоростной передачи (High-speed).
Контроллер не поддерживает 10-битную адресацию приемника.

5.19.3.4 Синхронизация линии SCL

Когда несколько устройств с интерфейсом I2C одновременно работают в режиме Master, используя шину SCL, в этом случае каждое устройство считывает состояние линии SCL и автоматически подстраивает время владения линии SCL в соответствии со временем самого медленного устройства.

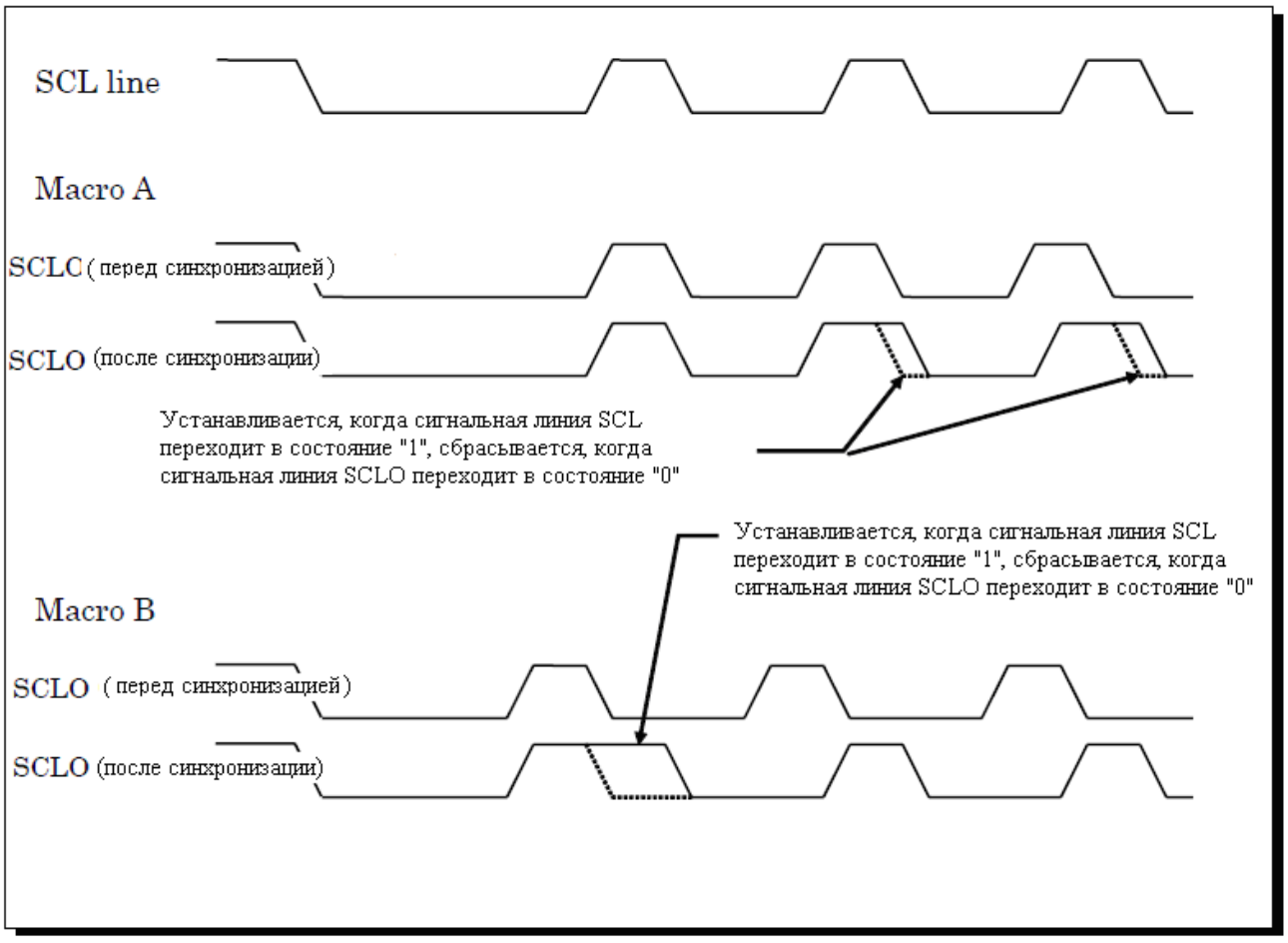


Рисунок 5.96 – Синхронизация линии SCL

5.19.3.5 Арбитраж

Процедура арбитража происходит, если несколько устройств, работающих в режиме Master, передают данные. Если устройство, работающее в режиме Master, формирует высокий уровень данных, а на линии SDA данные установлены в состояние логического нуля, то Master проигрывает арбитраж, устанавливается бит AL.

Пока другое устройство использует шину (BB=1), попытка сформировать состояние START приводит к тому, что Master проигрывает арбитраж, устанавливается бит AL.

Если шина не занята (BB=0) и устанавливается бит MSS и если в течение времени формирования состояния START данное состояние было сформировано другим Master, то данный Master теряет арбитраж, устанавливается бит AL.

При установке бита Al, биты MSS, TRX сбрасываются, устройство переходит в режим Slave.

К моменту, когда Master проигрывает арбитраж, он перестает владеть линией SDA. Однако Master не перестает владеть линией SCL до тех пор, пока не закончится передача одного байта и не будет сброшено прерывание.

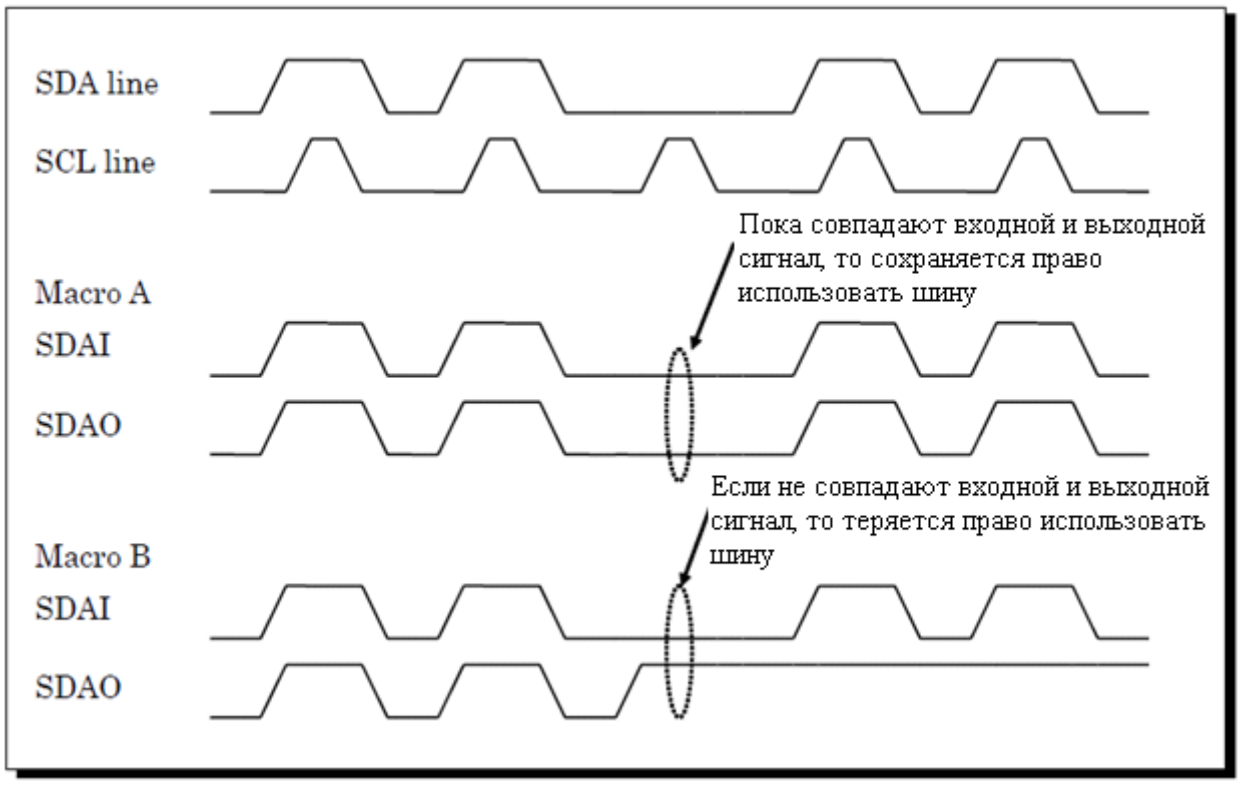


Рисунок 5.97 – Процедура арбитража

5.19.3.6 Подтверждение

9 бит данных является битом подтверждения (ACK) или неподтверждения (NAK). Если бит 9 находится в состоянии логического нуля – ACK, если находится в состоянии логической единицы – NAK. Признак подтверждения формируется приемником и передается передатчику.

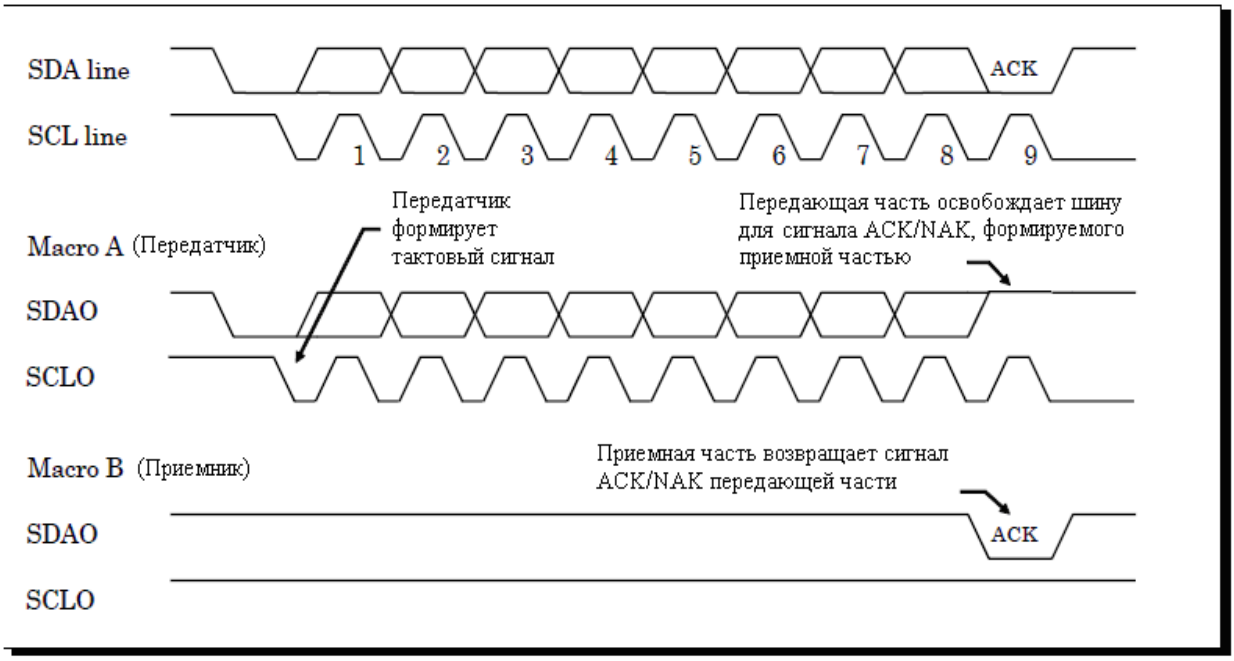


Рисунок 5.98 - Формирование признака подтверждения приема данных

5.19.3.7 Инициализация

См. Примечания в конце раздела 5.19.3.10.

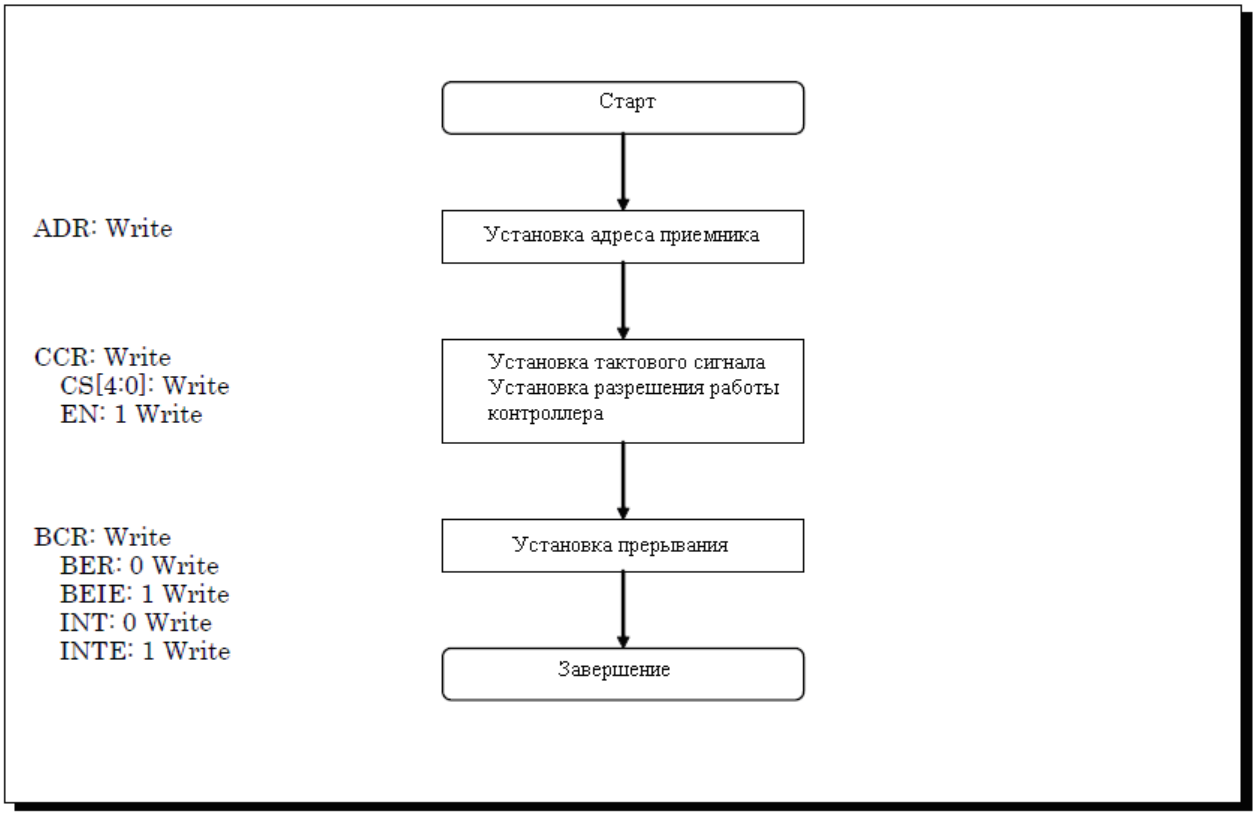


Рисунок 5.99 – Диаграмма инициализации контроллера I2C

5.19.3.8 Передача одного байта от Master к Slave

Ниже приведена диаграмма процесса передачи одного байта от Master к Slave (Рисунок 5.100). На рисунке также показаны изменения в конфигурации регистров передатчика (Master) и приемника (Slave). См. Примечания в конце раздела 5.19.3.10.

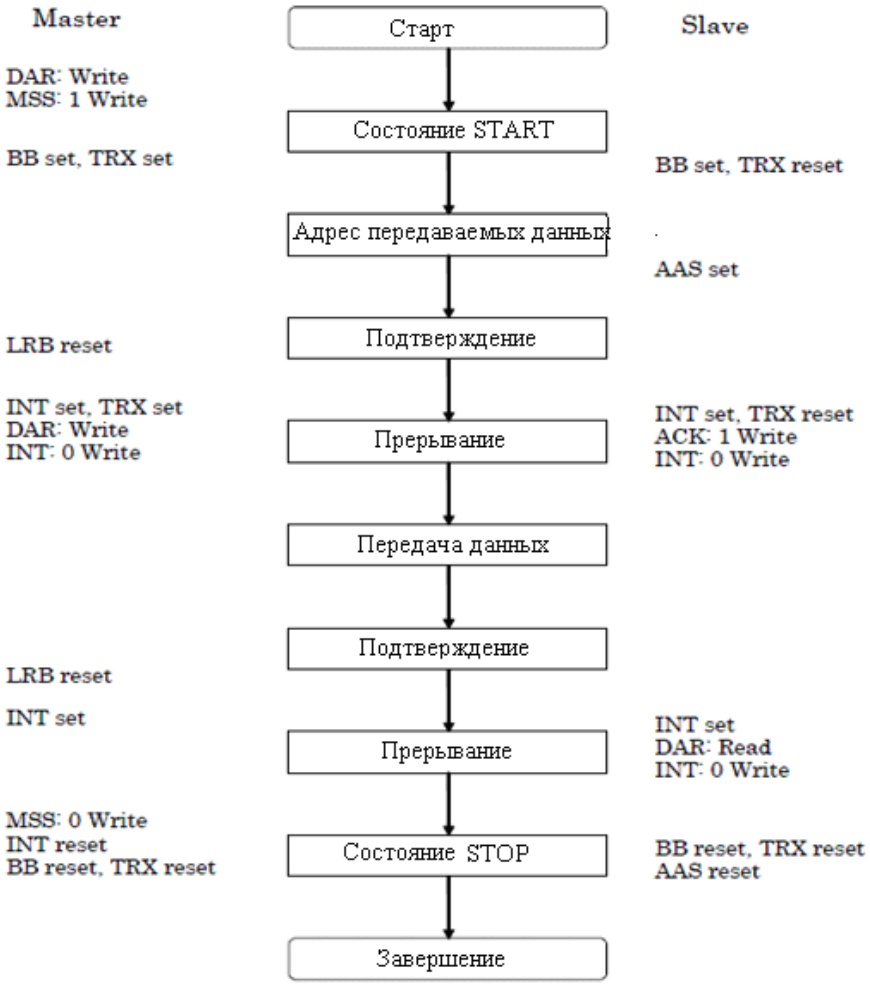


Рисунок 5.100 - Диаграмма процесса передачи одного байта от Master к Slave

5.19.3.9 Передача одного байта от Slave к Master

Ниже приведена диаграмма процесса передачи одного байта от Slave к Master (Рисунок 5.101). На рисунке также показаны изменения в конфигурации регистров передатчика (Master) и приемника (Slave). См. Примечания в конце раздела 5.19.3.10.

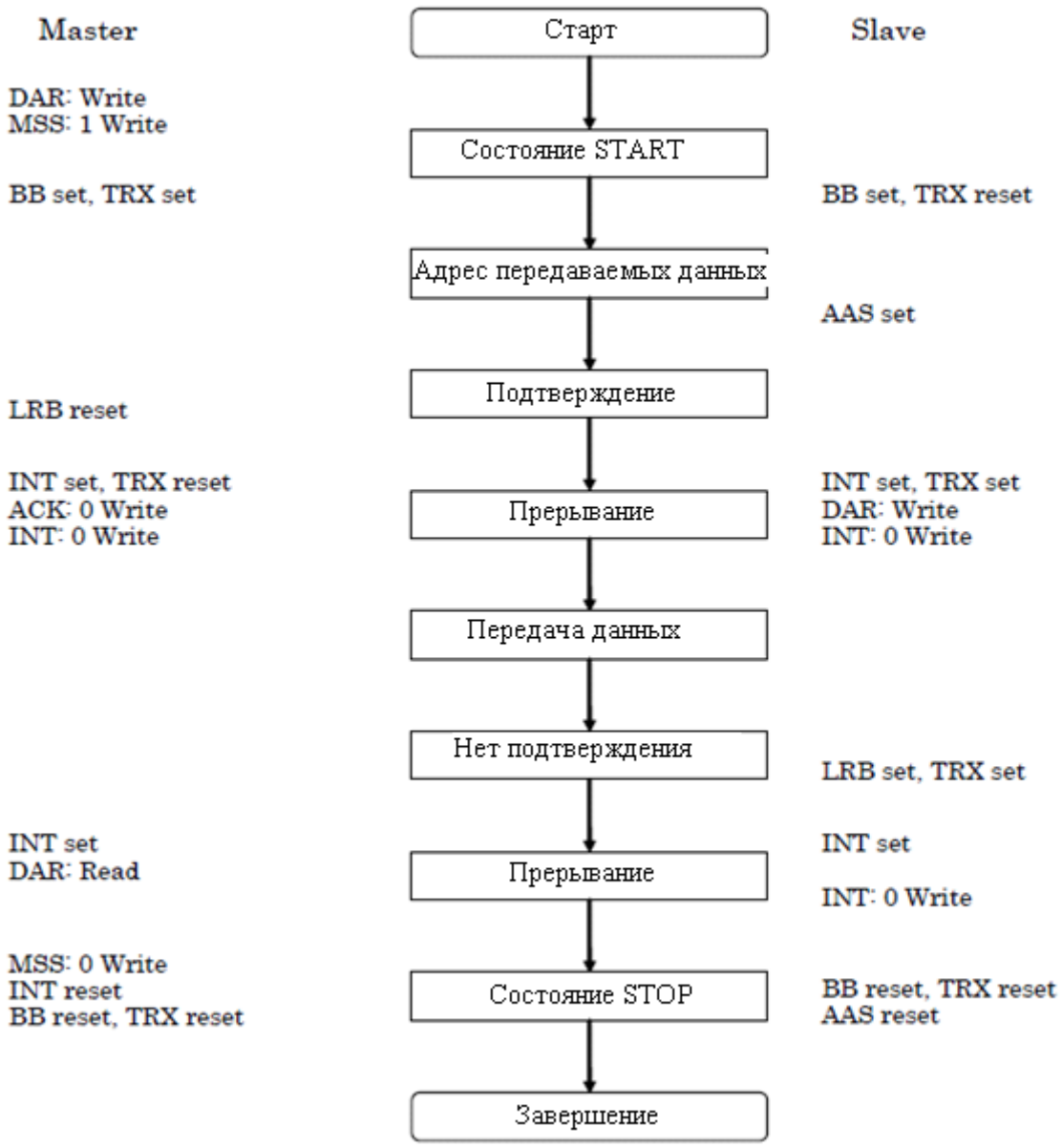


Рисунок 5.101 - Диаграмма процесса передачи одного байта от Slave к Master

5.19.3.10 Восстановление контроллера после ошибки шины

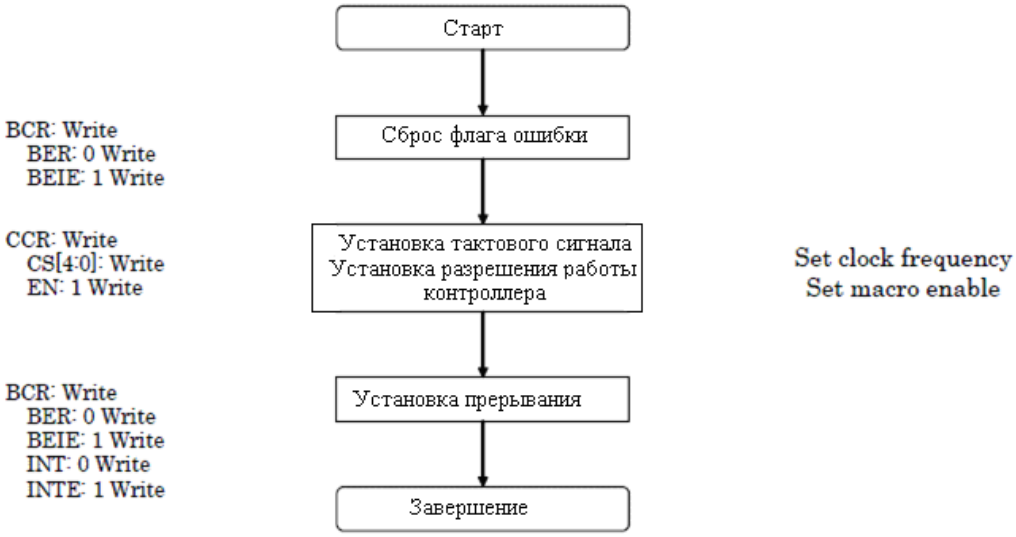


Рисунок 5.102 - Диаграмма процесса восстановления контроллера после ошибки шины

Примечания.

1) На рисунках выше запись типа «MSS: 1 Write» или типа «INT: 0 Write» обозначает, что надо произвести запись в бит соответствующего регистра единицы или нуля, тогда как запись типа «BB set» или «INT reset» обозначает, что I2C устройство, вследствие каких-либо событий, изменило свою конфигурацию и установило бит соответствующего регистра в 1 (set) или в 0 (reset).

2) При записи значений в биты регистров надо соблюдать осторожность, чтобы не перезаписать значения бит, которые не предполагалось перезаписывать (должна выполняться операция read-modify-write).

5.20 Контроллер интерфейса ETHERNET

СБИС СНП-ВП содержит MAC контроллер (GRETH), который поддерживает сетевой протокол Ethernet, а также поддерживает стандарты шин AMBA AHB и AMBA APB. Скорость передачи контроллера составляет 10/100 Мбит/сек в полудуплексном и дуплексном режимах. Интерфейс AMBA APB используется для конфигурации и управления контроллером, интерфейс AMBA AHB используется для передачи данных. Поток данных обрабатывается DMA-каналами. Один DMA–канал предназначен для передачи данных, другой - для приема данных. Оба канала делят интерфейс AHB. Интерфейс Ethernet поддерживает не зависящий от среды передачи интерфейс RMII (сокращенный формат интерфейса MII). Данный интерфейс должн быть соединен с устройством физического уровня (PHY). GRETH также обеспечивает доступ к интерфейсу MII Management (MDIO интерфейс), который используется для настройки блока PHY.

Аппаратно в контроллере поддерживается интерфейс Ethernet Debug Communication Link (EDCL). EDCL основан на протоколе UDP/IP, и применяется для удаленной отладки системы.

Блок-схема внутренней структуры GRETH приведена ниже (см. Рисунок 5.103).

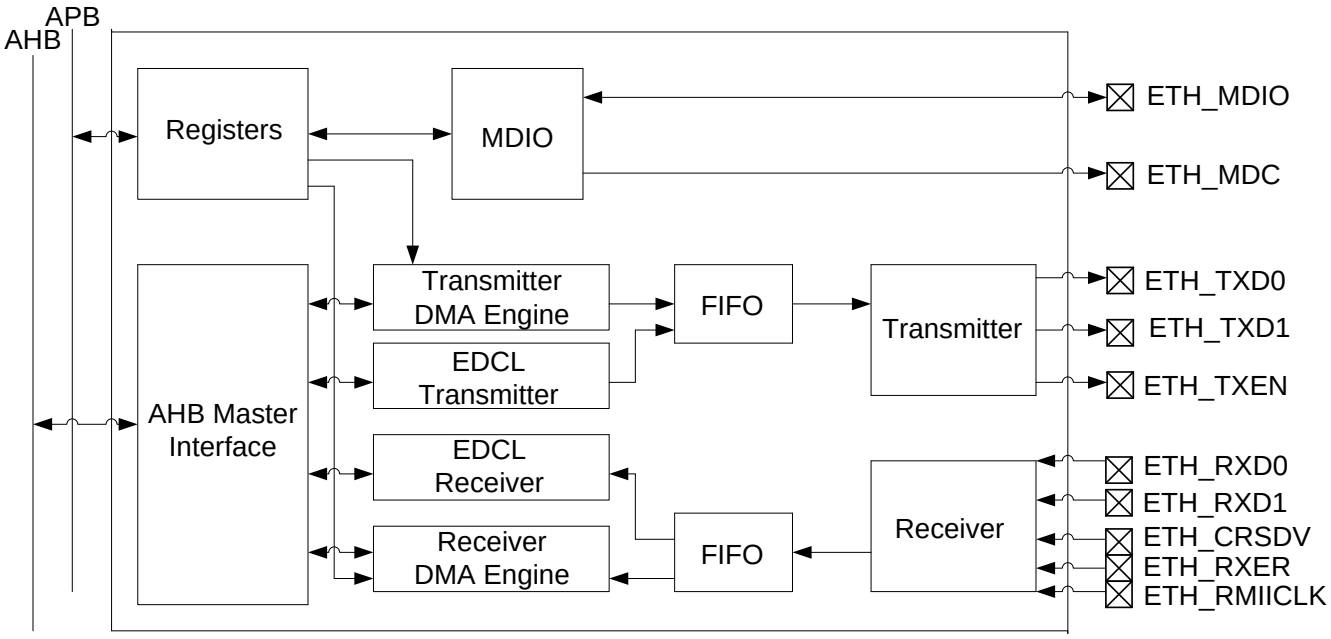


Рисунок 5.103 - Блок-схема внутренней структуры GRETH

5.20.1 Принципы работы контроллера GRETH

5.20.1.1 Обзор системы

Контроллер GRETH состоит из 3-х функциональных блоков: 2-х DMA-каналов, MDIO-интерфейса и блока удаленной отладки EDCL.

Основная функциональность контроллера соответствует функциональности DMA-каналов, которые осуществляют передачу данных между шиной AHB и сетевым протоколом Ethernet. Один DMA-канал используется для передачи данных, другой DMA-канал - для приема данных. Конфигурирование DMA-каналов осуществляется с помощью регистров, программно доступных через APB-интерфейс.

MDIO-интерфейс используется для доступа к конфигурационным и статусным регистрам одного или более PHY-микросхем, подключенных к MAC. Управление работой данного интерфейса осуществляется посредством шины APB.

Блок EDCL обеспечивает доступ к системному коммутатору HPMS через шину AHB. Блок позволяет производить удаленную отладку программного обеспечения. Для этого совместно используется протоколы UDP, IP, ARP с протоколом пользовательского уровня (custom application layer). EDCL не имеет программно-доступных регистров и работает всегда параллельно с DMA-каналами.

СБИС СНП-ВП поддерживает подключение PHY-микросхем только с интерфейсом RMII, который является усеченной по количеству выводов версией интерфейса MII. Данные, принятые с шины AHB, поступают в блок передатчика Ethernet, а оттуда, посредством интерфейса RMII, поступают на шину Ethernet. Соответственно данные, принятые с шины Ethernet, сохраняются в блоке приемника Ethernet, а оттуда поступают на шину AHB. Оба этих блока имеют буфера памяти типа FIFO для передачи потоков данных.

Канал связи EDCL и DMA-каналы делят блоки приемника Ethernet Receiver и передатчика Ethernet Transmitter.

5.20.1.2 Поддержка протоколов

GRETH–контроллер разработан согласно IEEE стандарту 802.3-2002 и стандарту 802.3Q-2003. Дополнительный уровень управления не поддерживается. Это означает, что пакеты с типом 0x8808 (единственные определенные в настоящее время управляющие пакеты) отбрасываются.

5.20.1.3 Синхронизация

GRETH–контроллер имеет 3 домена тактовых сигналов: AHB, Ethernet-приемник, Ethernet-передатчик. Тактовые сигналы Ethernet-приемника и Ethernet-передатчика генерируются внешним PHY и поступают в ядро посредством RMII интерфейса. Три домена тактовых сигналов асинхронны друг другу. Все сигналы полностью синхронизованы внутри ядра контроллера.

Оба рабочих режима: дуплексный и полудуплексный, поддерживаются и могут работать со скоростью 10 или 100 Мбит.

5.20.1.4 Интерфейс передатчика TxDMA

Интерфейс DMA-передатчика используется для передачи данных по сети Ethernet. При передаче данных используются дескрипторы, располагаемые в памяти.

5.20.1.4.1 Установка дескриптора

Дескриптор состоит из двух 32-х разрядных слов и показан на рисунке ниже (см.

Рисунок 5.104 и Рисунок 5.105). Количество байт, которые должны быть переданы, устанавливаются в поле длины (LENGHT), поле адреса (ADDRESS) адресует данные. Адрес должен быть выровнен с точностью до слова (32 бита). Если бит разрешения прерывания IE установлен, прерывание будет сгенерировано при отправке пакета (необходимо, чтобы бит прерывания передатчика в управляющем регистре также был установлен). Прерывание будет сгенерировано независимо от того, был ли пакет передан успешно или нет. Бит возврата (WR) также должен быть установленным перед передачей.

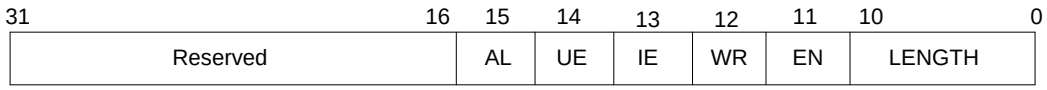


Рисунок 5.104 - Дескриптор передатчика GRETH (слово 0)

Таблица 5.213 – Описание полей слова 0 дескриптора передатчика GRETH

Биты	Название	Функция
[31:16]	-	Не используется.
[15]	Attempt Limit Error (AL)	Пакет не был передан, поэтому было достигнуто максимальное количество попыток.
[14]	Underrun Error (UE)	Пакет был некорректно передан из-за ошибки недозагрузки выходного FIFO.
[13]	Interrupt enable (IE)	Разрешение прерываний. Прерывание было сгенерировано, когда пакет от этого дескриптора был послан, при условии, что бит разрешения прерывания передатчика установлен в управляющем регистре. Прерывание формируется независимо от того, успешно был передан пакет или он был завершен с ошибкой.
[12]	Wrap(WR)	Если устанавливается 1, то указателю дескриптора присваивается начальное значение (0) после того, как дескриптор был использован. Если этот бит не установлен, указатель будет увеличен на 8. Указатель автоматически возвращается в начальное состояние (0), когда достигнута граница дескриптора 1 кБ.
[11]	Enable(EN)	Установка 1, активирует дескриптор. Устанавливается одним из самых последних.
[10:0]	LENGTH	Количество байт, которое должно быть передано.

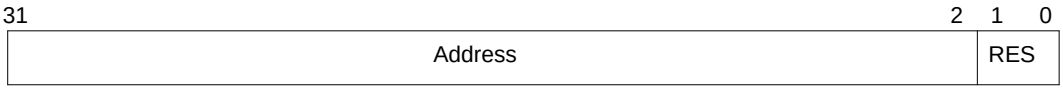


Рисунок 5.105 - Дескриптор передатчика GRETH (слово 1)

Таблица 5.214 - Описание полей слова 1 дескриптора передатчика GRETH

Биты	Название	Функция
[31:2]	Address (ADDRESS)	Указатель на область памяти, откуда данные будут загружены.
[1:0]	RES	Не используется

Для активации дескриптора необходимо установить бит EN, после этого дискриптор не должен перезаписываться до момента, пока GRETH сам не сбросит бит EN.

5.20.1.4.2 Начало передачи

Для начала передачи недостаточно только активировать дескриптор. Сначала в контроллере GRETH должен быть задан указатель области памяти, содержащей дескрипторы. Соответствующий указатель задается в регистре указателя дескриптора передатчика (Transmit descriptor pointer). Адрес должен быть выровнен по границе 1кБ. 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 адрес будет использован GRETH-контроллером, поле указателя будет увеличено на 8, чтобы указать на следующий дескриптор. Указатель будет автоматически возвращен обратно в начальное состояние (0), когда будет достигнута следующая граница 1кБ (используется дескриптор со смещением адреса 0x3F8). Для того чтобы указатель перешел в начальное состояние (0) до достижения границы 1кБ, в слове 0 дескриптора передачика должен быть установлен бит WR.

Регистр указателя дескриптора передатчика программно доступен для записи, но необходимо делать это с осторожностью. Запрещено записывать поле **DESCPNT** при активной передаче данных.

Последний шаг для активизации передачи - это установка бита разрешения передачи (transmit enable) в управляющем регистре (Control register). Таким образом, GRETH- контроллеру сообщается о наличие активных дескрипторов в таблице дескрипторов. Данный бит всегда должен устанавливаться при активизации новых дескрипторов, даже если передачи уже инициированы. Дескрипторы должны быть активированы прежде, чем бит разрешения передачи (transmit enable bit) в управляющем регистре (Control register) будет установлен.

5.20.1.4.3 Обработка дескриптора после передачи

Когда передача пакета закончена, статус записывается в слово 0 соответствующего дескриптора. Признак ошибки отсутствия данных (Underrun Error) устанавливается, если передающее FIFO стало пустым до того, как пакет был полностью передан (например, из-за того, что у GRETH-контроллера недостаточно приоритета для доступа к определенной области адресного пространства, чтобы изменить приоритет см. 5.2.1.7 и Таблица 5.3). Признак ошибки исчерпания количества попыток (Attempt Limit Error) устанавливается, если произошло больше столкновений, чем разрешено. Пакет считается успешно переданным, если оба этих бита равны 0. Другие разряды слова 0 дескриптора устанавливаются в 0 после передачи, слово 1 дескриптора остается нетронутым.

Разряд разрешения (EN) является признаком возможности повторного использования дескриптора. Дескриптор возможно использовать повторно после того, как этот бит был сброшен GRETH-контроллером.

В регистре статуса контроллера GRETH (Status register) имеются 3 бита, которые определяют состояние контроллера:

- Ошибка передатчика (Transmission Error) устанавливается каждый раз, когда передача заканчивается с ошибкой (когда, по крайней мере, один из 2 статусных битов в передаваемом дескрипторе был установлен).
- Прерывание передатчика (Transmitter Interrupt) устанавливается каждый раз, когда передача закончилась успешно.
- Признак ошибки передатчика AHB (Transmitter AHB Error) устанавливается, когда происходит ошибка на шине AHB, или во время чтения дескриптора, или во время чтения пакета данных. Любые инициированные передачи будут прекращены, и передатчик будет остановлен. Передатчик может быть активирован снова, если будет задан бит разрешения передачи (transmit enable) в регистре управления (Control register).

5.20.1.4.4 Установка данных передачи

Данные для передачи должны быть помещены, начиная с адреса, указанного в поле адреса дескриптора. Вместе с данными в буфере данных должен храниться заголовок пакета Ethernet, GRETH не генерирует его автоматически. В конце каждого пакета автоматически добавляются 4 байта контрольной суммы (CRC) пакета Ethernet. Каждый дескриптор будет посылаться одиночным Ethernet-пакетом. Если поле размера в дескрипторе больше, чем 1514B, то пакет не будет передан.

5.20.1.5 Интерфейс приемника RxDMA

Интерфейс DMA-приемника используется для приема данных по сетевому протоколу Ethernet. При приеме данных используются дескрипторы, определенные в памяти.

5.20.1.5.1 Установка дескрипторов

Дескриптор состоит из двух 32-х разрядных слов и показан на рисунке ниже (см. Рисунок 5.106 и Рисунок 5.107). Поле адреса слова 1 указывает на буфер памяти, в котором должны быть сохранены принятые данные. Буфер должен быть выровнен по границе слова. Максимальный размер буфера составляет максимальное значение, задаваемое при настройке, плюс

размер заголовка. Если разряд разрешения прерывания (IE) установлен, то прерывание будет сгенерировано, когда пакет будет принят в буфер (разряд прерывания приемника в управляющем регистре должен быть установлен). Прерывание генерируется независимо от правильности принятия пакета. Бит возврата (WR) должен быть установлен прежде, чем будет активирован дескриптор.

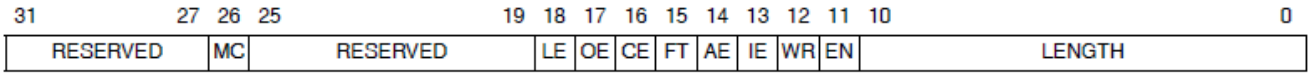


Рисунок 5.106 - Дескриптор приемника GRETH (слово 0)

Таблица 5.215 - Описание полей слова 0 дескриптора приемника GRETH

Биты	Название	Функция
[31:27]	-	Не используется.
[26]	Multicast address (MC)	Признак того, что адрес в заголовке пакета был Multicast адресом
[25:19]	-	Не используется.
[18]	Length error (LE)	Признак ошибки несовпадения длины/типа пакета в заголовке с действительным числом принятых байтов.
[17]	Overrun error (OE)	Признак переполнения приемного FIFO. Кадр данных был некорректно принят из-за переполнения FIFO.
[16]	CRC error (CE)	Признак ошибки CRC. Ошибка контрольной суммы (CRC) была обнаружена в данном кадре.
[15]	Frame too long (FT)	Был принят кадр, который превышает максимальный размер. Лишняя часть будет отброшена.
[14]	Alignment error (AE)	Ошибка выравнивания. Нечетное число полубайтов было принято.
[13]	Interrupt enable (IE)	Разрешение прерываний. Установка бита разрешает формирование прерывания после принятия пакета дескриптора. Прерывание генерируется независимо от того, успешно был принят пакет, или он был завершен с ошибкой
[12]	Wrap(WR)	Устанавка 1, присваивает указателю дескриптора начальное значение (0) после использования дескриптора. Если этот бит не установлен, то указатель инкрементируется на 8. Указатель автоматически принимает начальное значение (0), при достижении границы дескриптора 1кВ.
[11]	Enable(EN)	Устанавливается в 1 для активизации дескриптора. Это поле должно записываться самым последним (после всех остальных полей дескриптора).
[10:0]	LENGTH	Количество байт, которое должно быть передано.

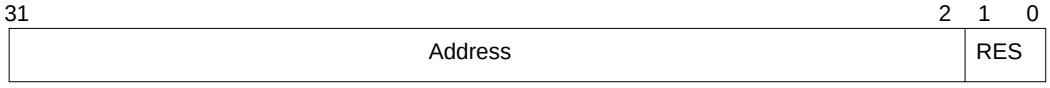


Рисунок 5.107 - Дескриптор приемника GRETH (слово 1)

Таблица 5.216 - Описание полей слова 1 дескриптора приемника GRETH

Биты	Название	Функция
[31:2]	Address (ADDRESS)	Указатель на область памяти, куда данные будут загружены.
[1:0]	RES	Не используется.

5.20.1.5.2 Начало приема

Для начала приема данных недостаточно только активировать дескриптор. Сначала в GRETH-контроллере должен быть задан указатель области памяти, содержащей дескрипторы.

Соответствующий указатель задается в регистре указателя дескриптора приемника (Receiver descriptor pointer). Адрес должен быть выровнен по границе 1кБ. 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 помещается по базовому адресу и, когда он будет использован GRETH-контроллером, поле указателя будет увеличено на 8, чтобы указать на следующий дескриптор. Указателю будет автоматически присвоено начальное значение (0), когда будет достигнута следующая граница 1кБ (использован дескриптор со смещением адреса 0x3F8). Для того, чтобы указатель перешел в начальное состояние (0) до достижения границы в 1кБ, в дескрипторах должен быть установлен бит WR.

Регистр указателя дескриптора приемника (Receiver descriptor pointer) программно доступно для записи, но необходимо делать это с осторожностью. Запрещено производить заись при приеме данных.

Последний шаг для активизации приема - это установка бита разрешения приема (RE) в управляющем регистре (Control register). Таким образом, GRETH- контроллеру сообщается о возможности считать первый дескриптор и ожидать входной пакет.

5.20.1.5.3 Обработка дескриптора после приема

После окончания приема, GRETH-контроллер сбрасывает бит активации (EN) в слове 0 дискриптора приемника, и другие управляющие биты (WR, IE). Поле длины (LENGTH) определяет количество принятых байт. По адресу, определенному в слове 1 дискриптора передатчика, сохраняются следующие части пакета Ethernet: MAC-адрес приемника, MAC-адрес источника, тип и поля данных Ethernet пакета. Разряды 18-14 в слове 0 дескриптора приемника являются статусными разрядами, определяющими различные ошибки при приеме данных (Таблица 5.215). Все пять разрядов сбрасываются после приема данных, если нет ошибок.

Если MAC-адрес принятого пакета является недопустимым, то бит IA регистра статуса (Status register) будет установлен в 1.

Пакеты размера больше максимального размера (1514B) вызывают установку признака FT в слове 0 дискриптора приемника, при этом не гарантируется, что в поле длины (LENGTH) содержится правильное значение количества принятых байтов. Счетчик останавливается после записи последнего байта блока максимального размера.

5.20.1.5.4 Прием с ошибками на шине AHB

Если происходит ошибка на шине AHB во время чтения дескриптора или сохранения данных, то устанавливается бит ошибки при приеме на шине AHB (RA) в регистре статуса (Status register) и приемник останавливается. Приемник может быть активирован снова, если будет установлен бит разрешения передачи (RE) в управляющем регистре (Control register).

5.20.1.6 MDIO интерфейс

Интерфейс MDIO обеспечивает доступ к конфигурационным и статусным регистрам PHY-микросхемы, используя двухпроводной интерфейс, являющийся частью интерфейса RMII. GRETH-контроллер обеспечивает полную поддержку MDIO интерфейса

MDIO интерфейс может быть использован для управления от 1 до 32 PHY, содержащих от 1 до 32-х 16-разрядных регистров. Операция чтения запускается, если в регистр MDIO Control записан адрес конкретного PHY, адрес нужного регистра, установлен бит чтения. После запуска операции чтения устанавливается разряд занятости (Busy bit) в регистре MDIO Control. Когда операция чтения заканчена, сбрасывается разряд занятости (Busy bit). Если операция была успешной, бит LF равен 0, и в поле данных находятся считанные данные. При ошибочном завершении операции бит LF устанавливается в 1. В этом случае поле данных не определено.

Для выполнения операции записи надо в регистр управления MDIO (MDIO Control) записать нужные 16 бит данных, записать адрес PHY, записать адрес регистра, для которго нужно произвести операцию записи, и установить бит записи (WR). Когда операция записи начинает-

ся, бит занятости (BU) устанавливается в 1, когда операция записи заканчивается, бит записи устанавливается в 0. Если операция была успешной, бит LF равен 0.

5.20.1.7 Интерфейс для удаленной отладки (Ethernet Debug Communication Link)

Ethernet Debug Communication Link (EDCL) обеспечивает доступ к шине AHB посредством сетевого протокола Ethernet. Совместно используются UDP, IP, ARP протоколы и протокол прикладного пользовательского уровня. Протокол прикладного уровня использует ARQ алгоритм для обеспечения передачи команд AHB. Передача по чтению или по записи может быть сгенерирована по любому адресу на шине AHB

5.20.1.7.1 Функционирование EDCL

EDCL получает пакеты параллельно с приемным каналом DMA MAC-контроллера. EDCL использует отдельный MAC адрес: 0xEC1766AEDE20, который позволяет различать пакеты для EDCL от пакетов, предназначенных DMA-каналу MAC-контроллера. EDCL также имеет отдельный IP адрес: 0xC0A80101 (значение при загрузке). При этом младшие 4 бита EDCL IP-адреса и EDCL MAC-адреса устанавливаются в соответствии со значениями на входах BOOTM0, BOOTM1, BOOTM2 (Таблица 5.217). Значения EDCL IP-адреса можно программно изменить, используя регистр EDCL IP. Значения EDCL MAC адреса изменить программно нельзя.

Таблица 5.217 - Соответствия IP и MAC адресов входам BOOTM

BOOTM[2:0]	4 lsb EDCL IP и EDCL MAC
0x0, 0x1, 0x2, 0x3	0x1
0x4	0xA
0x5	0xB
0x6	0xC
0x7	0xD

Так как ARP пакеты используют широковещательный адрес Ethernet протокола, в данном случае IP адрес позволяет различать ARP пакеты интерфейса EDCL от пакетов, поступающих в DMA-канал. Пакеты для EDCL, не обрабатываются приемными каналами DMA.

После проверки пакетов на корректность, выполняется операция на шине AHB. Ответы посылаются автоматически блоком передатчиком EDCL при завершении операции.

5.20.1.7.2 EDCL протокол

EDCL получает формат кадров Ethernet, содержащие IP или ARP данные. ARP обрабатываются согласно спецификации протокола без всяких исключений.

IP пакеты передают команды шины AHB. EDCL ожидает формат кадра Ethernet, содержащий IP, UDP протоколы и специфические части прикладного уровня EDCL. На рисунке ниже (см Рисунок 5.108) показан IP пакет, требуемый для EDCL интерфейса.

Ethernet Header	IP Header	UDP Header	2B Offset	4B Control word	4B Address	Data 0-242 4B Words	Ethernet CRC

Рисунок 5.108 - IP пакет, требуемый для EDCL интерфейса

Для успешной связи по EDCL интерфейсу требуется следующее: корректный MAC адрес приемника, поле типа протокола Ethernet, содержащее 0x806 (ARP) или 0x800 (IP) и коректный IP–адрес приемника. Контрольная сумма IP–заголовка и идентификационные поля не проверяются. Существует несколько ограничений для полей IP-заголовка. Поле протокола UDP

должно быть всегда равным 0x11. Длина и контрольная сумма – единственные IP–поля, изменяемые при 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 приемника. Контрольная сумма UDP не используется, и при ответах поле контрольной суммы равно 0.

UDP поле данных содержит поля прикладного уровня интерфейса EDCL. На рисунке (см. Рисунок 5.109) показаны поля протокола прикладного уровня в пакетах, принятых посредством EDCL интерфейса, которые соответствуют полям 2B Offset и 4B Control word IP пакета, требуемого для EDCL интерфейса (Рисунок 5.108).

16-bit Offset	14-bit Sequence number	1-bit R/W	10-bit Length	7-bit Unused

Рисунок 5.109 - Поля прикладного уровня в принимаемых кадрах

16-битное смещение используется для того, чтобы выровнять в памяти оставшуюся часть данных прикладного уровня по границе, кратной слову, и может быть установлено любым значением. Поле R/W определяет, будет выполняться чтение или запись. Поле длины содержит количество байтов, которое нужно считать или записать. Если R/W равен 1, поле данных, показанное на рисунке ниже (см. Рисунок 5.108), содержит данные, которые должны быть записаны. Если R/W равен 0, то поле данных пусто в принимаемых пакетах.

На Рисунок 5.110 показаны поля прикладного уровня ответов от EDCL.

16-bit Offset	14-bit Sequence number	1-bit ACK/NACK	10-bit Length	7-bit Unused

Рисунок 5.110 - 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. Значение внутреннего счетчика в ответном кадре хранится в поле номера последовательности. Если номер последовательности совпадает, внутренний счетчик увеличивается, значение счетчика запоминается в поле номера последовательности и поле ACK/NAK сбрасывается при ответе. Поле длины (LENGTH) всегда сбрасывается для кадров с ACK/NAK=1. Неиспользуемые поля не проверяются и копируются при ответе. Так могут быть установлены, например, некоторые дополнительные разряды идентификатора, если необходимо.

5.20.1.7.3 Размер буфера EDCL

Блок EDCL содержит внутренние блоки памяти, предназначенные для хранения полученных пакетов в процессе передачи данных. Общий размер этих блоков памяти – 2kB. Максимальное количество пакетов, которое можно обрабатывать, - 4. Максимальный размер каждого пакета - 512kB (получается сложением размера заголовка и CRC поля – 56B и максимального размера данных в пакете – 456B). Если отправляется большее количество пакетов данных, до прихода подтверждения, то происходит потеря пакетов.

5.20.2 Программные драйверы

Драйверы для MAC поддержаны следующими операционными системами: RTEMS, eCos, uClinux и Linux-2.6. Драйверы находятся в свободном доступе в виде полного исходного кода по GPL лицензии от Gaisler Research”s web site (http://gaisler.com).

5.20.3 Программная модель контроллера GRETH

Программно доступные регистры контроллера GRETH расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение ETH Base = 0xFFF06000 и общий размер 4 Кб. Регистры имеют разрядность 32 бит. Спецификация регистров представлена в таблице ниже (см.
Таблица 5.218).

Таблица 5.218 – регистры контроллера GRETH

APB address offset	Register
0x0	Control regiseter
0x4	Status register
0x8	MAC Address MSB
0xC	MAC Address LSB
0x10	MDIO Control
0x14	Transmit descriptor pointer
0x18	Receiver descriptor pointer
0x1C	EDCL IP
0x20	Hash table msb
0x24	Hash table lsb
0x28	EDCL MAC address MSB
0x2C	EDCL MAC address LSB
0x10000-0x107FC	Transmit RAM buffer debug access
0x20000-0x207FC	Receiver RAM buffer debug access
0x30000-0x3FFFC	EDCL buffer debug access

5.20.3.1 Регистр управления контроллера GRETH (Control register)

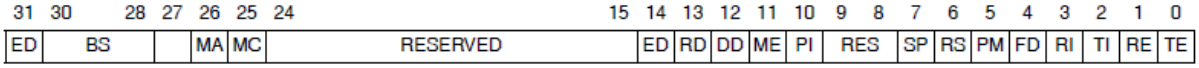


Рисунок 5.111 – Управляющий регистр контроллера GRETH

Таблица 5.219 - Структура управляющего регистра контроллера GRETH

Биты	Название	Функция
[31]	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]	-	Не используется.
[26]	MDIO interrupts available (MA)	Устанавливается в 1, когда ядро контроллера поддерживает прерывания от MDIO блока. Бит доступен только для чтения.
[25]	Multicast available (MC)	1 - ядро контроллера поддерживает прием широковещательного адреса для пакета данных (multicast).
[24:15]	-	Зарезервированы.
[14]	EDCL Disable (ED)	При установки бита происходит отключение EDCL, при сбросе подключается EDCL. Начальное значение устанавливается равным 0.
[13]	RAM debug enable (RD)	0 – режим отладки памяти заблокирован.
[12]	Disable duplex detection (DD)	Если FSM не может завершить обнаружение EDCL, то интерфейс MDIO будет находиться в состоянии занятости и будет недоступен. Если требуется программный доступ к MDIO, то FSM может быть остановлен, и как только сбросится бит занятоси MDIO, интерфейс MDIO становится доступным. FSM не может быть заново запущен.
[11]	Multicast enable (ME)	Разрешение приема широковещательного адреса для пакета данных (multicast).
[10]	PHY status change interrupt enable (PI)	Разрешает прерывания при изменениях статусов PHY-блоков.
[9:8]	-	Не используется.
[7]	Speed (SP)	Устанавливает текущий режим скорости. 0= 10 Mbit, 1=100 Mbit. Используется только в режиме RMII (rmii=1). Значение по умолчанию автоматически считывается из PHY-блока после аппаратного сброса. Значение при сбросе “1”.
[6]	Reset (RS)	1, записанная в этот разряд, обнуляет ядро GRETH-контроллера. Бит сбрасывается автоматически.
[5]	Promiscuous mode (PM)	Если бит установлен, GRETH-контроллер работает в смешанном режиме, который означает, что он будет получать все пакеты независимо от адреса назначения.
[4]	Full duplex (FD)	Если бит установлен, GRETH-контроллер работает в дуплексном режиме, иначе он работает в полудуплексе. Значение при сбросе “0”.
[3]	Receiver Interrupt (RI)	Разрешает прерывания от блока приемника. Прерывание будет сгенерировано каждый раз, когда передан пакет, бит установливается в 1. Прерывание генерируется независимо от того, был ли пакет принят успешно, или завершен с ошибкой. Значение при сбросе “0”.
[2]	Transmitter Interrupt (TI)	Разрешает прерывания от блока передатчика. Прерывание будет сгенерировано каждый раз, когда принят пакет, бит установливается в 1. Прерывание генерируется независимо от того, был ли пакет передан успешно, или завершен с ошибкой Значение при сбросе “0”.
[1]	Receive enable (RE)	Бит устанавливается в 1 при необходимости начать прием входящих пакетов. Пока этот разряд 1, GRETH-контроллер будет считывать новые дескрипторы, и как только он встретит неактивный дескриптор, он остановится, бит RE сбросится. При необходимости продолжить прием входящих пакетов (и при наличии правильного вида дескрипторов приемника), бит RE надо установить сново. Значение при сбросе “0”.
[0]	Transmit enable (TE)	Бит устанавилвается в 1 при необходимости начать передачу пакетов. Пока этот разряд 1, GRETH-контроллер будет считывать новые дескрипторы и как только он встретит неактивный дескриптор, он остановится, бит TE сбросится. При необходимости продолжить передачу (и при наличии правильного вида дескрипторов передатчика), бит TE надо установить заново. Значение при сбросе “0”.

5.20.3.2 Регистр статуса контроллера GRETH (Status register)

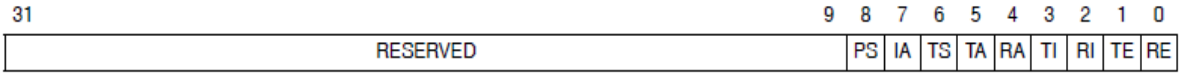


Рисунок 5.112 –регистр статуса контроллера GRETH

Таблица 5.220 - Структура регистра статуса контроллера GRETH

Биты	Название	Функция
[31:9]	-	Не используется.
[8]	PHY status changes (PS)	Устанавливается каждый раз когда обнаруживается изменение статуса PHY-блока.
[7]	Invalid address (IA)	Был принят пакет с адресом, не допустимый для MAC. Сбрасывается, когда записывается 1. Значение при ресете “0”.
[6]	Too small (TS)	Был принят пакет меньше минимального размера. Сбрасывается, когда записывается 1. Значение при ресете “0”.
[5]	Transmitter AHB error (TA)	Обнаружена ошибка на шине AHB при работе DMA-передатчика. Сбрасывается, когда записывается 1. Значение не обнуляется.
[4]	Receiver AHB error (RA)	Обнаружена ошибка на шине AHB 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.

5.20.3.3 Регистр, определяющий старшие биты (MSB) MAC-адреса (MAC Address MSB)

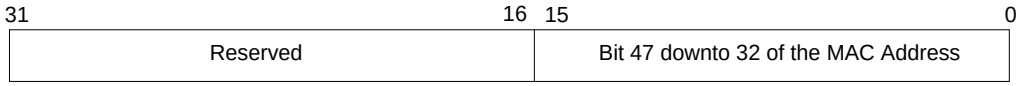


Рисунок 5.113 - Регистр, определяющий старшие биты (MSB) MAC-адреса

Таблица 5.221 - Структура регистра, определяющая старшие биты (MSB) MAC-адреса

Биты	Название	Функция
[31:16]	-	Не используется.
[15:0]	Bit 47-32	Два старших значащих байта MAC – адреса. Не обнуляются

5.20.3.4 Регистр, определяющий младшие биты (LSB) MAC-адреса (MAC Address LSB)

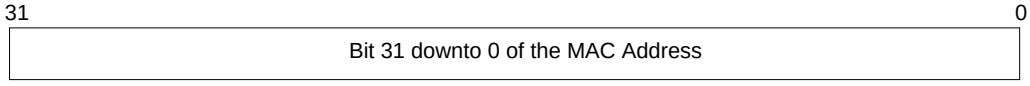


Рисунок 5.114 - Регистр, определяющий младшие биты (LSB) MAC-адреса

Таблица 5.222 - Структура регистра, определяющая младшие биты (LSB) MAC-адреса

Биты	Название	Функция
[31:0]	Bit 31-0	Четыре младших значащих байта MAC – адреса. Не обнуляются.

5.20.3.5 Регистр управления/статуса интерфейса MDIO контроллера GRETH (MDIO Control/Status)

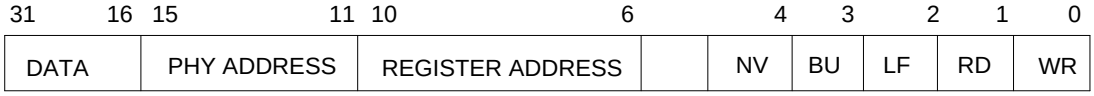


Рисунок 5.115 - Регистр управления/статуса интерфейса MDIO контроллера GRETH

Таблица 5.223 - Структура регистра управления/статуса интерфейса MDIO контроллера GRETH

Биты	Название	Функция
[31:16]	Data (DATA)	При операции чтения содержит считанные данные. При операции записи содержит данные, которые будут переданы. Значение при сбросе - 0x0000.
[15:11]	PHY address (PHYADDR)	Поле содержит адрес PHY-блока, который будет выбиран во время операции чтения или записи. Значение при сбросе - 0x0000.
[10:6]	Register address (REGADDR)	Это поле содержит адрес регистра, который будет выбиран во время операции чтения или записи. Значение при сбросе - 0x0000.
[5]	-	Не используется.
[4]	Not valid (NV)	Когда операция закончена (BUSY=0), этот бит указывает были ли приняты валидные данные (содержит ли поле данных правильные данные). Значение при сбросе “0”.
[3]	Busy (BU)	Когда операция выполняется, этот разряд установлен в 1. Как только операция закончена этот разряд сбрасывается. Значение при сбросе “0”.
[2]	Linkfail (LF)	Когда операция завершается (BUSY=0) этот бит устанавливается, если произошла ошибка на интерфейсе MDIO. Значение при сбросе “1”.
[1]	Read (RD)	Начать операцию чтения. Данные хранятся в поле данных. Значение при сбросе “0”.
[0]	Write (WR)	Начать операцию записи. Данные берутся из поля данных. Значение при сбросе “0”.

5.20.3.6 Регистр базового адреса таблицы дескрипторов передатчика GRETH- контроллера (Transmit descriptor pointer)

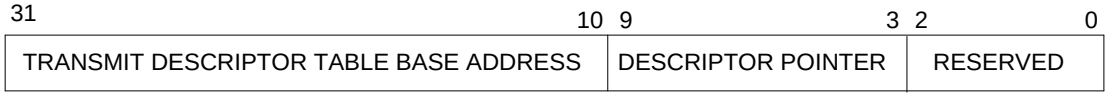


Рисунок 5.116 - Регистр базового адреса таблицы дескрипторов передатчика GRETH-контроллера

Таблица 5.224 - Структура регистра базового адреса таблицы дескрипторов передатчика

Биты	Название	Функция
[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 MAC-контроллером протокола Ethernet (если не установлен бит WR в дескрипторе передатчика).
[2:0]	-	Не используется

5.20.3.7 Регистр базового адреса таблицы дескрипторов приемника GRETH-контроллера (Receiver descriptor pointer)

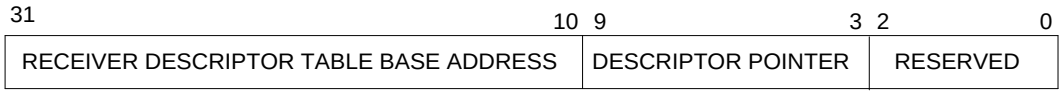


Рисунок 5.117 - Регистр базового адреса таблицы дескрипторов приемника GRETH-контроллера

Таблица 5.225 - Структура регистра базового адреса таблицы дескрипторов приемника

Биты	Название	Функция
[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 MAC-контроллером протокола Ethernet (если не установлен бит WR в дескрипторе передатчика).
[2:0]	-	Не используется

5.20.3.8 Регистр IP блока EDCL GRETH-контроллера (EDCL IP)

Рисунок 5.118 - Регистр IP блока EDCL GRETH- контроллера

Таблица 5.226 - Структура регистра IP блока EDCL

Биты	Название	Функция
[31:0]	EDCL IP адрес	После сброса содержит EDCL IP 0xC0A80101 (при этом младшие четыре бита могут быть другими (Таблица 5.217)). Может быть перезаписан.

5.20.3.9 Регистр старших битов hash таблицы GRETH-контроллера (Hash table msb)

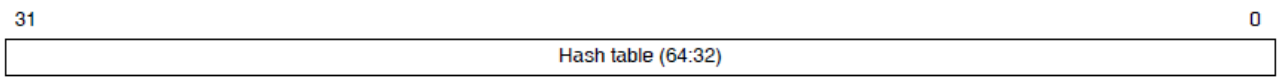


Рисунок 5.119 - Регистр старших битов hash таблицы GRETH- контроллера

Таблица 5.227 - Структура регистра старших битов hash таблицы

Биты	Название	Функция
[31:0]	Hash table (64:32)	Значение бит с 64 по 32 hash таблицы.

5.20.3.10 Регистр младших битов hash таблицы GRETH-контроллера (Hash table lsb)

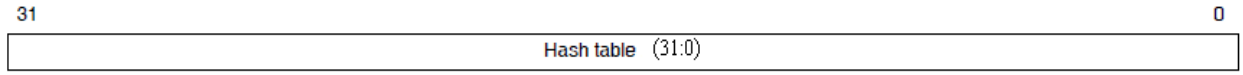


Рисунок 5.120 - Регистр младших битов hash таблицы GRETH- контроллера

Таблица 5.228 - Структура регистра младших битов hash таблицы

Биты	Название	Функция
[31:0]	Hash table (31:0)	Значение бит с 31 по 0 hash таблицы.

5.20.3.11 Регистр старших бит MAC-адреса для EDCL GRETH-контроллера (EDCL MAC address MSB)

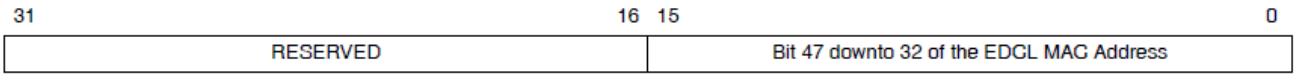


Рисунок 5.121 - Регистр старших бит MAC-адреса для EDCL GRETH-контроллера

Таблица 5.229 - Структура регистра старших бит MAC-адреса для EDCL

Биты	Название	Функция
[31:16]	-	Зарезервировано
[15:0]	Bit 47 downto 32 of the EDCL MAC Address	Содержит два старших байта MAC-адреса EDCL 0xEC17, доступен только для чтения.

5.20.3.12 Регистр младших бит MAC-адреса для EDCL GRETH-контроллера (EDCL MAC address LSB)

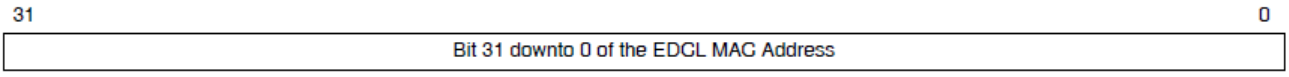


Рисунок 5.122 - Регистр младших бит MAC-адреса для EDCL GRETH-контроллера

Таблица 5.230 - Структура регистра младших бит MAC-адреса для EDCL

Биты	Название	Функция
[31:0]	Bit 31 downto 0 of the EDCL MAC Address	Содержит четыре младших байта MAC-адреса EDCL 0x66AEDE20 (младшие четыре бита могут быть другими (Таблица 5.217)), доступен только для чтения.

5.21 Контроллер внешней динамической памяти LPDDR2

Интерфейс с внешней памятью LPDDR2 управляется двумя контроллерами:

- Контроллер логического уровня (EMIC)
- Контроллер физического уровня и DLL (PHYC)

Ниже приведены основные параметры контроллеров:

- Тип поддерживаемой памяти LPDDR2 SDRAM;
- Частота шины 200 МГц;
- Разрядность внешней шины данных, бит – 32;
- Количество сигналов выбора микросхемы (chip select) – 2;
- Количество банков в микросхеме памяти – 8;

Программирование контроллеров осуществляется процессорным ядром ARM. Перед тем, как производить обращения к внешней памяти, необходимо инициализировать оба контроллер.

5.21.1 Отображение адресов

В карте памяти NMU внешней памяти отведено 2000_0000h адресов с доступом к 32 разрядным данным. В карте памяти ARMU внешней памяти отведено 8000_0000h адресов с доступом к 8 разрядным данным. Таким образом, максимально доступно 2 ГБ памяти.

30	29	27	26	12	11	2	1	0
Chip Select	Bank		Row		Column		Datapath	

Рисунок 5.123 - Состав адреса максимальной ширины при обращении из ARMU

28	27	25	24	10	9	0
Chip Select	Bank		Row		Column	

Рисунок 5.124 - Состав адреса максимальной ширины при обращении из NMU

Последовательность полей Chip Select, Bank, Row, Column, Datapath неизменна. Для использования микросхем памяти меньшего размера, есть возможность уменьшения ширины полей адреса с помощью параметров контроллера EMIC – COL_DIFF, ROW_DIFF и BANK_DIFF. При этом остальные поля будут сдвинуты вправо на соответствующую величину.

Например: при необходимости подключения памяти размером 64 МБ с 12 строками и 8 столбцами следует записать следующие значения параметров:

COL_DIFF = 2

ROW_DIFF = 3

Тогда контроллер EMIC будет разбивать адрес на поля следующим образом:

25	24	22	21	10	9	2	1	0
Chip Select	Bank		Row		Column		Datapath	

Рисунок 5.125 - Состав адреса уменьшенной ширины при обращении из NMU

5.21.2 Регистровая модель контроллера EMIC

В данном разделе описываются программно доступные регистры блока контроллера динамической памяти. Таблица 5.231 содержит список программно доступных регистров контроллера EMIC. Программно доступные регистры контроллера расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение EMIC Base = 0xFFF70000 и общий размер 64 Кб. Спецификация регистров приведена в таблице ниже.

- RW = чтение/запись.
- RD = только чтение.
- WR = только запись.
- RW+ = программное чтение/запись, внутреннее разрешение записи. Эти параметры могут быть считаны или записаны программно, но также могут быть изменены внутренней логикой.
- RW_D = чтение/запись. Эти параметры будут установлены в 0x0 по системному сбросу, но будут изменены на не нулевые значения в следующем после системного сброса такте.

Таблица 5.231 – Список регистров контроллера EMIC

Название	AXI Адрес	Controller Адрес	Тип	Параметры, содержащиеся в регистре
DENALI_CTL_00	0x00	0x00	RD RW RW	VERSION DRAM_CLASS START
DENALI_CTL_01	0x04	0x01	RD RD RD RD	READ_DATA_FIFO_DEPTH MAX_CS_REG MAX_COL_REG MAX_ROW_REG
DENALI_CTL_02	0x08	0x02	RD RD RD RD	ASYNC_CDC_STAGES WRITE_DATA_FIFO_PTR_WIDTH WRITE_DATA_FIFO_DEPTH READ_DATA_FIFO_PTR_WIDTH
DENALI_CTL_03	0x0c	0x03	RD RD RD RD	AXI0_WRCMD_PROC_FIFO_LOG2_DEPTH AXI0_WRFIFO_LOG2_DEPTH AXI0_RDFIFO_LOG2_DEPTH AXI0_CMDFIFO_LOG2_DEPTH
DENALI_CTL_04	0x10	0x04	RW	TINIT
DENALI_CTL_05	0x14	0x05	RW	TINIT3
DENALI_CTL_06	0x18	0x06	RW	TINIT4
DENALI_CTL_07	0x1c	0x07	RW RW	NO_AUTO_MRR_INIT TINIT5
DENALI_CTL_08	0x20	0x08	RW RW RW	NO_CMD_INIT TDLL INITAREF
DENALI_CTL_09	0x24	0x09	RW RW RW RW	TCCD TBST_INT_INTERVAL WRLAT CASLAT_LIN
DENALI_CTL_10	0x28	0x0a	RW RW RW RW	TWTR TRAS_MIN TRC TRRD
DENALI_CTL_11	0x2c	0x0b	RW RW RW RW	TMRD TRTP TFAW TRP
DENALI_CTL_12	0x30	0x0c	RW RW RW	TCKE TRAS_MAX TMOD
DENALI_CTL_13	0x34	0x0d	RW RW RW RW	TWR TRCD WRITEINTERP TCKESR
DENALI_CTL_14	0x38	0x0e	RW RW RW RW	TRAS_LOCKOUT CONCURRENTAP AP TMRR
DENALI_CTL_15	0x3c	0x0f	RW RW RW_D RW	REG_DIMM_ENABLE TRP_AB BSTLEN TDAL
DENALI_CTL_16	0x40	0x10	RW RW WR	TREF_ENABLE RESERVED AREFRESH
DENALI_CTL_17	0x44	0x11	RW RW	TREF TRFC
DENALI_CTL_18	0x48	0x12	RW	TREF_INTERVAL

DENALI_CTL_19	0x4c	0x13	RW RW	TXSR TPDEX
DENALI_CTL_20	0x50	0x14	RW RW RW	SREFRESH_EXIT_NO_REFRESH PWRUP_SREFRESH_EXIT TXSNR
DENALI_CTL_21	0x54	0x15	RW RW RW RW	CKSRE LOWPOWER_REFRESH_ENABLE CKE_DELAY ENABLE_QUICK_SREFRESH
DENALI_CTL_22	0x58	0x16	RD RD WR RW	LP_ARB_STATE LP_STATE LP_CMD CKSRX
DENALI_CTL_23	0x5c	0x17	RW RW RW	LP_AUTO_MEM_GATE_EN LP_AUTO_EXIT_EN LP_AUTO_ENTRY_EN
DENALI_CTL_24	0x60	0x18	RW RW RW	LP_AUTO_SR_MC_GATE_IDLE LP_AUTO_SR_IDLE LP_AUTO_PD_IDLE
DENALI_CTL_25	0x64	0x19	RW	RESERVED
DENALI_CTL_26	0x68	0x1a	RW+	WRITE_MODEREG
DENALI_CTL_27	0x6c	0x1b	RW+ RD	READ_MODEREG MRW_STATUS
DENALI_CTL_28	0x70	0x1c	RD RD	AUTO_TEMPCHK_VAL PERIPHERAL_MRR_DATA
DENALI_CTL_29	0x74	0x1d	RW RW	MR0_DATA_0 REFRESH_PER_AUTO_TEMPCHK
DENALI_CTL_30	0x78	0x1e	RW RW	MR2_DATA_0 MR1_DATA_0
DENALI_CTL_31	0x7c	0x1f	RW RW	MR3_DATA_0 MRSINGLE_DATA_0
DENALI_CTL_32	0x80	0x20	RW RW RD	MR17_DATA_0 MR16_DATA_0 MR8_DATA_0
DENALI_CTL_33	0x84	0x21	RW RW	MR1_DATA_1 MR0_DATA_1
DENALI_CTL_34	0x88	0x22	RW RW	MRSINGLE_DATA_1 MR2_DATA_1
DENALI_CTL_35	0x8c	0x23	RW RD RW	MR16_DATA_1 MR8_DATA_1 MR3_DATA_1
DENALI_CTL_36	0x90	0x24	RW_D RW RW	ZQINIT LONG_COUNT_MASK MR17_DATA_1
DENALI_CTL_37	0x94	0x25	RW RW	ZQCS ZQCL
DENALI_CTL_38	0x98	0x26	RW WR	ZQ_ON_SREF_EXIT ZQ_REQ
DENALI_CTL_39	0x9c	0x27	RW	ZQ_INTERVAL
DENALI_CTL_40	0xa0	0x28	RW RW RD	NO_ZQ_INIT ZQRESET ZQ_IN_PROGRESS
DENALI_CTL_41	0xa4	0x29	RW RW RW RW	COL_DIFF ROW_DIFF BANK_DIFF ZQCS_ROTATE
DENALI_CTL_42	0xa8	0x2a	RW RW RW	ADDR_CMP_EN COMMAND_AGE_COUNT AGE_COUNT

			RW_D	APREBIT
DENALI_CTL_43	0xac	0x2b	RW RW RW RW	PRIORITY_EN PLACEMENT_EN BANK_SPLIT_EN ADDR_COLLISION_MPM_DIS
DENALI_CTL_44	0xb0	0x2c	RW RW RW RW	W2R_SPLIT_EN CS_SAME_EN RW_SAME_PAGE_EN RW_SAME_EN
DENALI_CTL_45	0xb4	0x2d	RW RW RW RW	DISABLE_RD_INTERLEAVE SWAP_EN NUM_Q_ENTRIES_ACT_DISABLE DISABLE_RW_GRUP_W_BNK_CONFLICT
DENALI_CTL_46	0xb8	0x2e	RW RW RW RW	LPDDR2_S4 REDUC CS_MAP INHIBIT_DRAM_CMD
DENALI_CTL_47	0xbc	0x2f	WR RD RW RW	CTRLUPD_REQ CONTROLLER_BUSY IN_ORDER_ACCEPT Q_FULLNESS
DENALI_CTL_48	0xc0	0x30	RD RW	INT_STATUS CTRLUPD_REQ_PER_AREF_EN
DENALI_CTL_49	0xc4	0x31	RW WR	INT_MASK INT_ACK
DENALI_CTL_50	0xc8	0x32	RD	OUT_OF_RANGE_ADDR
DENALI_CTL_51	0xcc	0x33	RD RD RD	OUT_OF_RANGE_SOURCE_ID OUT_OF_RANGE_TYPE OUT_OF_RANGE_LENGTH
DENALI_CTL_52	0xd0	0x34	RD	PORT_CMD_ERROR_ADDR
DENALI_CTL_53	0xd4	0x35	RW_D RW_D RD RD	R2W_DIFFCS_DLY R2R_DIFFCS_DLY PORT_CMD_ERROR_TYPE PORT_CMD_ERROR_ID
DENALI_CTL_54	0xd8	0x36	RW_D RW RW_D RW_D	R2W_SAMECS_DLY R2R_SAMECS_DLY W2W_DIFFCS_DLY W2R_DIFFCS_DLY
DENALI_CTL_55	0xdc	0x37	RW RW RW RW	TDQSCK_MIN TDQSCK_MAX W2W_SAMECS_DLY W2R_SAMECS_DLY
DENALI_CTL_56	0xe0	0x38	RD RW RW RW	CKE_STATUS AXI0_FIFO_TYPE_REG AXI0_W_PRIORITY AXI0_R_PRIORITY
DENALI_CTL_57	0xe4	0x39	RD RW RW	TDFI_PHY_WRLAT DLL_RST_ADJ_DLY DLL_RST_DELAY
DENALI_CTL_58	0xe8	0x3a	RW RD RW_D RD	DRAM_CLK_DISABLE TDFI_RDDATA_EN TDFI_PHY_RDLAT UPDATE_ERROR_STATUS
DENALI_CTL_59	0xec	0x3b	RW RD	TDFI_CTRLUPD_MAX TDFI_CTRLUPD_MIN
DENALI_CTL_60	0xf0	0x3c	RW RW	TDFI_PHYUPD_TYPE1 TDFI_PHYUPD_TYPE0
DENALI_CTL_61	0xf4	0x3d	RW RW	TDFI_PHYUPD_TYPE3 TDFI_PHYUPD_TYPE2

DENALI_CTL_62	0xf8	0x3e	RW	TDFI_PHYUPD_RESP
DENALI_CTL_63	0xfc	0x3f	RW	TDFI_CTRLUPD_INTERVAL
DENALI_CTL_64	0x100	0x40	RW RW_D RW RW	TDFI_DRAM_CLK_DISABLE TDFI_CTRL_DELAY WRLAT_ADJ RDLAT_ADJ
DENALI_CTL_65	0x104	0x41	RW	TDFI_DRAM_CLK_ENABLE

Ниже приведено описание параметров, определяющих работу контроллера EMIC.

Таблица 5.232 – DENALI_CTL_00 (Controller Address 0x00, AXI Address 0x00)

Название	Биты	Значение по умолчанию	Диапазон	Описание
VERSION	31:16	0x2041	0x2041	Содержит номер версии контроллера. Только для чтения.
DRAM_CLASS	11:8	0x0	0x0-0xf	Определяет режим работы контроллера. 'b0101 – LPDDR2 остальные значения зарезервированы.[3)]
START	0	0x0	0x0-0x1	Инициирует обработку команд контроллером. Установите в 1 для обработки. До тех пор контроллер не будет выполнять никаких действий, кроме записи и чтения параметров и приема трафика с внутренней шины. Окончание запуска контроллера отслеживается по 3 биту параметра INT_STATUS.[3)]

Таблица 5.233 – DENALI_CTL_01 (Controller Address 0x01, AXI Address 0x04)

Название	Биты	Значение по умолчанию	Диапазон	Описание
READ_DATA_ FIFO_DEPTH	31:24	0x00	0x0-0xff	Отражает глубину очереди чтения контроллера. Только для чтения.[5)]
MAX_CS_REG	17:16	0x2	0x0-0x2	Содержит максимальное число доступных CS. Только для чтения.[5)]
MAX_COL_REG	11:8	0xa	0x0-0xa	Содержит максимальную ширину адреса столбцов в микросхемах DRAM. Только для чтения. [5)]
MAX_ROW_REG	3:0	0xf	0x0-0xf	Содержит максимальную ширину адресной шины памяти. Только для чтения.[5)]

Таблица 5.234 – DENALI_CTL_02 (Controller Address 0x02, AXI Address 0x08)

Название	Биты	Значение по умолчанию	Диапазон	Описание
ASYNC_CDC_ STAGES	31:24	0x00	0x0-0xff	Показывает число задержек синхронизатора, определенных для асинхронного пересечения границ. Только для чтения.[5)]
WRITE_DATA_ FIFO_PTR_WIDTH	23:16	0x00	0x0-0xff	Показывает ширину указателя глубины очереди записи данных ядра контроллера. Только для чтения.[5)]
WRITE_DATA_ FIFO_DEPTH	15:8	0x00	0x0-0xff	Показывает глубину очереди записи данных ядра контроллера. Только для чтения.[5)]
READ_DATA_ FIFO_PTR_WIDTH	7:0	0x00	0x0-0xff	Показывает ширину указателя глубины очереди чтения данных ядра контроллера. Только для чтения.[5)]

Таблица 5.235 – DENALI_CTL_03 (Controller Address 0x03, AXI Address 0x0C)

Название	Биты	Значение по умолчанию	Диапазон	Описание
AXI0_WRCMD_ PROC_FIFO_ LOG2_DEPTH	31:24	0x00	0x0-0xff	Показывает глубину FIFO обработки команд записи порта AXI 0. Значение является log2 глубины. Только для чтения.[5)]
AXI0_WRFIFO_ LOG2_DEPTH	23:16	0x00	0x0-0xff	Показывает глубину FIFO записи данных порта AXI 0. Значение является log2 глубины. Только для чтения.[5)]
AXI0_RDFIFO_ LOG2_DEPTH	15:8	0x00	0x0-0xff	Показывает глубину FIFO чтения данных порта AXI 0. Значение является log2 глубины. Только для чтения.[5)]
AXI0_CMDFIFO_ LOG2_DEPTH	7:0	0x00	0x0-0xff	Показывает глубину FIFO команд порта AXI 0. Значение является log2 глубины. Только для чтения.[5)]

Таблица 5.236 – DENALI_CTL_04 (Controller Address 0x04, AXI Address 0x10)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TINIT	23:0	0x000000	0x0-0xffffff	Время инициализации памяти TINIT1 в тактах.[1)]

Таблица 5.237 – DENALI_CTL_05 (Controller Address 0x05, AXI Address 0x14)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TINIT3	23:0	0x000000	0x0-0xffffff	Время инициализации памяти TINIT3 в тактах.[1)]

Таблица 5.238 – DENALI_CTL_06 (Controller Address 0x06, AXI Address 0x18)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TINIT4	23:0	0x000000	0x0-0xffffff	Время инициализации памяти TINIT4 в тактах.[1)]

Таблица 5.239 – DENALI_CTL_07 (Controller Address 0x07, AXI Address 0x1c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
NO_AUTO_ MRR_INIT	24	0x0	0x0-0x1	Запрет MRR команд во время инициализации. Установить в 1 для запрета.[4)]
TINIT5	23:0	0x000000	0x0-0xffffff	Время инициализации памяти TINIT5 в тактах.[1)]

Таблица 5.240 – DENALI_CTL_08 (Controller Address 0x08, AXI Address 0x20)

Название	Биты	Значение по умолчанию	Диапазон	Описание
NO_CMD_INIT	24	0x0	0x0-0x1	Этот параметр должен быть установлен в 0
TDLL	23:8	0x0000	0x0-0xffff	Этот параметр должен быть установлен в 0
INITAREF	3:0	0x0	0x0-0xf	Этот параметр должен быть установлен в 0

Таблица 5.241 – DENALI_CTL_09 (Controller Address 0x09, AXI Address 0x24)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TCCD	28:24	0x00	0x1-0x1f	DRAM CAS-to-CAS минимальная задержка в тактах.[1)]
TBST_INT_INTERVAL	18:16	0x0	0x1-0x7	Период попыток повторения команды, заблокированной в результате передачи burst команды памяти.
WRLAT	10:8	0x0	0x0-0x7	DRAM WRLAT значение в тактах. Задержка между выдачей команды записи и выдачей записываемых данных.[1)]
CASLAT_LIN	4:0	0x00	0x0-0x1f	Устанавливает задержку в тактах от отправки команды чтения до получения данных из контроллера. Бит (0) - это инкремент на полтакта и старшие биты определяют CAS задержку для контроллера.[1)]

Таблица 5.242 – DENALI_CTL_10 (Controller Address 0x0a, AXI Address 0x28)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TWTR	29:24	0x00	0x0-0x3f	DRAM TWTR значение в тактах. Интервал между командой записи и командой чтения.[1)]
TRAS_MIN	23:16	0x00	0x0-0xff	DRAM TRAS_MIN значение в тактах. Минимальное время активации строки в памяти.[1)]
TRC	15:8	0x00	0x0-0xff	DRAM TRC значение в тактах. Интервал между активными командами для одного банка памяти.[1)]
TRRD	7:0	0x00	0x0-0xff	DRAM TRRD значение в тактах. Интервал между активациями разных банков памяти.[1)]

Таблица 5.243 – DENALI_CTL_11 (Controller Address 0x0b, AXI Address 0x2c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TMRD	28:24	0x00	0x0-0x1f	TMRW значение в тактах. Минимальный интервал между командами чтения/записи контрольных регистров памяти.[1)]
TRTP	18:16	0x0	0x0-0x7	DRAM TRTP значение в тактах. Интервал между командой чтения (READ) и командой закрытия (PRECHARGE) банка памяти.[1)]
TFAW	13:8	0x00	0x0-0x3f	DRAM TFAW значение в тактах.[1)]
TRP	4:0	0x00	0x0-0x1f	DRAM TRP значение в тактах. Интервал между командой закрытия (PRECHARGE) и активацией (ACTIVATE) банка памяти.[1)]

Таблица 5.244 – DENALI_CTL_12 (Controller Address 0x0c, AXI Address 0x30)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TCKE	26:24	0x0	0x0-0x7	Минимальная ширина полки CKE в тактах.[1)]
TRAS_MAX	23:8	0x0000	0x0-0xffff	DRAM TRAS_MAX значение в тактах. Максимальное время активации строки в памяти.[1)]
TMOD	7:0	0x00	0x0-0xff	Интервал между командой чтения/записи контрольных регистров памяти и другими командами.[1)]

Таблица 5.245 – DENALI_CTL_13 (Controller Address 0x0d, AXI Address 0x34)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TWR	29:24	0x00	0x0-0x3f	DRAM TWR значение в тактах.[1)]
TRCD	23:16	0x00	0x0-0xff	DRAM TRCD значение в тактах. RAS-to-CAS интервал.[1)]
WRITEINTERP	8	0x0	0x0-0x1	Разрешает контроллеру прерывать burst запись в микросхемы DRAM с помощью команды чтения. Установить в 1 для разрешения прерывания.
TCKESR	7:0	0x00	0x0-0xff	Минимальный интервал, в течение которого должен быть снят сигнал CKE во время собственной регенерации (self REFRESH).[1)]

Таблица 5.246 – DENALI_CTL_14 (Controller Address 0x0e, AXI Address 0x38)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TRAS_LOCKOUT	24	0x0	0x0-0x1	Разрешает контроллеру выполнять команды авто-закрытия (auto PRECHARGE) до окончания параметра TRAS_MIN. Установить в 1 для разрешения.
CONCURRENTAP	16	0x0	0x0-0x1	Разрешает контроллеру выдавать команды другим банкам, пока банк в состоянии авто-закрытия (auto PRECHARGE). Установить в 1 для разрешения.
AP	8	0x0	0x0-0x1	Разрешение режима авто-закрытия (auto PRECHARGE) контроллера. Установить в 1 для разрешения.
TMRR	3:0	0x0	0x0-0xf	DRAM TMRR значение в тактах.[1)]

Таблица 5.247 – DENALI_CTL_15 (Controller Address 0x0f, AXI Address 0x3c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
REG_DIMM_ ENABLE	24	0x0	0x0-0x1	Разрешение регистровых DIMM операций контроллера. Установить в 1 для разрешения.
TRP_AB	20:16	0x00	0x0-0x1f	DRAM TRP значение в тактах. Интервал между закрытием (PRECHARGE) и активацией (ACTIVATE) всех банков памяти.[1)]
BSTLEN	10:8	0x2	0x0-0x7	Закодированная длина burst, отправляемого в микросхемы DRAM во время инициализации. 'b010 – 4 слова 'b011 – 8 слов остальные значения зарезервированы
TDAL	5:0	0x00	0x0-0x3f	DRAM TDAL значение в тактах. Если значение не указано в спецификации памяти, присвоить значение TWR+TRP.[1)]

Таблица 5.248 – DENALI_CTL_16 (Controller Address 0x10, AXI Address 0x40)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TREF_ENABLE	16	0x0	0x0-0x1	Разрешение команд автоматической регенерации (auto REFRESH) памяти. Установить в 1 для разрешения.
RESERVED	8	0x0	0x0-0x1	Резерв. Должно быть установлено в 0x0.
AREFRESH	0	0x0	0x0-0x1	Инициирует автоматическую регенерацию по окончании текущего burst. Установить в 1 для запуска. Все активные банки будут закрыты. Только для записи.

Таблица 5.249 – DENALI_CTL_17 (Controller Address 0x11, AXI Address 0x44)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TREF	31:16	0x0000	0x0-0xffff	DRAM TREF значение в тактах. Средний интервал между автоматическими регенерациями (auto REFRESH).[1)]
TRFC	9:0	0x000	0x0-0x3ff	DRAM TRFC значение в тактах. Время регенерации (REFRESH).[1)]

Таблица 5.250 – DENALI_CTL_18 (Controller Address 0x12, AXI Address 0x48)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TREF_INTERVAL	15:0	0x0000	0x0-0xffff	Интервал между регенерациями (REFRESH) для разных сигналов CS (разных микросхем памяти). При значении 0 регенерация происходит одновременно.[1)]

Таблица 5.251 – DENALI_CTL_19 (Controller Address 0x13, AXI Address 0x4c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TXSR	31:16	0x0000	0x0-0xffff	DRAM TXSR значение в тактах. Интервал от выхода из режима регенерации до следующей активной команды (требующей тактовый сигнал).[1)]
TPDEX	15:0	0x0000	0x0-0xffff	DRAM TXP значение в тактах. Время выхода из режима пониженного энергопотребления.[1)]

Таблица 5.252 – DENALI_CTL_20 (Controller Address 0x14, AXI Address 0x50)

Название	Биты	Значение по умолчанию	Диапазон	Описание
SREFRESH_ EXIT_ NO_REFRESH	24	0x0	0x0-0x1	Запрет автоматической регенерации (auto REFRESH) после выхода из режима собственной регенерации (self REFRESH). Установить в 1 для запрета.
PWRUP_ SREFRESH_ EXIT	16	0x0	0x0-0x1	Разрешение включения через режим собственной регенерации в отличие от полной инициализации памяти. Установить в 1 для разрешения.
TXSNR	15:0	0x0000	0x0-0xffff	DRAM TXSNR значение в тактах. Интервал от выхода из режима собственной регенерации до следующей команды, не требующей тактовый сигнал.[1)]

Таблица 5.253 – DENALI_CTL_21 (Controller Address 0x15, AXI Address 0x54)

Название	Биты	Значение по умолчанию	Диапазон	Описание
CKSRE	31:24	0x00	0x0-0xff	Интервал удержания тактового сигнала после входа в режим собственной регенерации (self REFRESH).
LOWPOWER_ REFRESH_ ENABLE	17:16	0x0	0x0-0x3	Разрешение собственной регенерации в режиме пониженного потребления. Бит (0) контролирует cs0, бит (1) - cs1 и т.д. Установить каждый бит в 1 для запрета.[7)]
CKE_DELAY	10:8	0x0	0x0-0x7	Дополнительные такты задержки.
ENABLE_ QUICK_ SREFRESH	0	0x0	0x0-0x1	Разрешение пользователю прерывать инициализацию памяти для входа в режим собственной регенерации. Установить в 1 для разрешения.

Таблица 5.254 – DENALI_CTL_22 (Controller Address 0x16, AXI Address 0x58)

Название	Биты	Значение по умолчанию	Диапазон	Описание
LP_ARB_ STATE	27:24	0x0	0x0-0xf	Отчет о состоянии арбитра. Биты (2:0) показывают какой интерфейс управляет модулем контроля питания. Бит (3) показывает имеет ли программный интерфейс активную блокировку арбитра. Для битов (2:0), значение 0 показывает отключение модуля, значение 1 показывает программный контроль интерфейса, значение 2 показывает контроль интерфейса с внешней ножки, значение 3 показывает автоматический контроль интерфейса, значение 4 показывает динамический контроль питания через чип селект интерфейс, значение 5 показывает, что контроллер под контролем. Только для чтения.[7)]
LP_ STATE	21:16	0x20	0x0-0x3f	Состояние параметров управления питанием. (4:0) показывают текущее состояние питания. 5'b00000 начальное (Idle) 5'b00001 активное пониженное (Active Power-Down) 5'b00010 активное пониженное с отключением синхросигнала памяти (Active Power-Down With Memory Clock Gating) 5'b00011 закрытое пониженное (Precharge Power-Down) 5'b00100 закрытое пониженное с отключением синхросигнала памяти (Precharge Power-Down With Memory Clock Gating) 5'b00101 собственная регенерация (Self-Refresh) 5'b00110 собственная регенерация с отключением синхросигнала памяти (Self-Refresh With Memory Clock Gating) 5'b00111 собственная регенерация с отключением синхросигнала памяти и контроллера (Self-Refresh With Memory And Controller Clock Gating) остальные значения зарезервированы (5) показывает, что переходной процесс завершен и биты (4:0) актуальны. Только для чтения.[7)]
LP_CMD	15:8	0x00	0x0-0xff	Интерфейс программных запросов команд управления питанием. (0) установить 1 для выхода из любого режима (1) установить 1 для входа в указанный режим (4:2) тип режима потребления 3'b000 активный пониженный (Active Power-Down) 3'b001 закрытый пониженный (Precharge Power-Down) 3'b010 собственная регенерация (Self-Refresh) остальные значения зарезервированы (5) установить 1 для отключения синхросигнала памяти (6)установить 1 для отключение синхросигнала контроллера (7) команда блокировки. Только для записи. [7)]
CKSRX	7:0	0x00	0x0-0xff	Интервал удержания тактового сигнала после выхода из режима собственной регенерации. [7)]

Таблица 5.255 – DENALI_CTL_23 (Controller Address 0x17, AXI Address 0x5c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
LP_AUTO_ MEM_ GATE_EN	17:16	0x0	0x0-0x3	Разрешение отключения синхросигнала памяти при входе в энергосберегающий режим через автосчетчики. (0) при отключении питания (1) при собственной регенерации (self REFRESH). Установить каждый бит в 1 для разрешения. 7)
LP_AUTO_ EXIT_EN	10:8	0x0	0x0-0x7	Разрешение автоматического выхода из любого указанного режима пониженного потребления при поступлении в очередь команд команды записи или чтения. (0) активный или закрытый пониженный (1) собственная регенерация (self REFRESH) (2) собственная регенерация с отключением синхросигнала памяти и контроллера (self REFRESH With Memory And Controller Clock Gating). Установить каждый бит в 1 для разрешения. 7)
LP_AUTO_ ENTRY_EN	2:0	0x0	0x0-0x7	Разрешение автоматического входа в любое состояние по питанию по истечению соответствующего таймера бездействия (0) активный или закрытый пониженный (1) собственная регенерация (self REFRESH) (2) собственная регенерация с отключением синхросигнала памяти и контроллера (self REFRESH With Memory And Controller Clock Gating). Установить каждый бит в 1 для разрешения. 7)

Таблица 5.256 – DENALI_CTL_24 (Controller Address 0x18, AXI Address 0x60)

Название	Биты	Значение по умолчанию	Диапазон	Описание
LP_AUTO_ SR_MC_ GATE_IDLE	31:24	0x00	0x0-0xff	Число длинных последовательностей до того, как контроллер переведет память в собственную регенерацию с отключением синхросигнала 7)
LP_AUTO_ SR_IDLE	23:16	0x00	0x0-0xff	Число длинных последовательностей до того, как контроллер переведет память в собственную регенерацию. 7)
LP_AUTO_ PD_IDLE	11:0	0x000	0x0-0xfff	Определяет время бездействия до того как, контроллер переведет память в состояние активного пониженного питания. 7)

Таблица 5.257 – DENALI_CTL_25 (Controller Address 0x19, AXI Address 0x64)

Название	Биты	Значение по умолчанию	Диапазон	Описание
RESERVED	5:0	0x00	0x0-0x3f	Резерв для будущего использования. Должно быть установлено в 0x0.

Таблица 5.258 – DENALI_CTL_26 (Controller Address 0x1a, AXI Address 0x68)

Название	Биты	Значение по умолчанию	Диапазон	Описание
WRITE_ MODEREG	25:0	0x0000000	0x0-0x3ffffff	Регистр режима записи данных в контрольные регистры микросхемы DRAM. (25) установить 1 для запуска записи (24) установить 1 для записи по всем CS, иначе значение берется из (15:8) (23) запись одиночного регистра под номером (7:0) (18) запись MR16, MR17 (17) запись MR0, MR1, MR2, MR3 (16) запись MR0, MR1, MR2, MR3, MR16, MR17 (15:8) номер CS для записи, используется только при снятом (24) (7:0) номер регистра для записи, используется только при установленном (23) [4)]

Таблица 5.259 – DENALI_CTL_27 (Controller Address 0x1b, AXI Address 0x6c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
READ_ MODEREG	24:8	0x00000	0x0-0x1ffff	Чтение определенного регистра режима памяти из определенной микросхемы, когда установлен стартовый бит. (16) запускает чтение (15:8) номер CS для чтения (7:0) определяют номер контрольного регистра памяти для чтения [4)]
MRW_ STATUS	7:0	0x00	0x0-0xff	Статус регистра режима записи памяти. Установка бита (0) показывает ошибку программирования параметра WRITE_MODEREG. Только чтение. [4)]

Таблица 5.260 – DENALI_CTL_28 (Controller Address 0x1c, AXI Address 0x70)

Название	Биты	Значение по умолчанию	Диапазон	Описание
AUTO_ TEMPCHK_ VAL	23:16	0x00	0x0-0xff	MR4 данные для всех микросхем, доступных автоматическим MRR командам, с 4 битами на CS. (3:0) соответствуют cs0 (7:4) соответствуют cs1 и т.д. Значение показывает OP7, OP2, OP1 и OP0 биты. Только чтение. [4)]
PERIPHERAL_ MRR_DATA	15:0	0x0000	0x0-0xffff	Данные и чип, возвращаемые по запросу READ_MODEREG контрольного регистра памяти. (7:0) показывают данные чтения (15:8) показывают микросхему. Только чтение. [4)]

Таблица 5.261 – DENALI_CTL_29 (Controller Address 0x1d, AXI Address 0x74)

Название	Биты	Значение по умолчанию	Диапазон	Описание
MR0_DATA_0	30:16	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти 0 для CS0. [4)]
REFRESH_PER_ AUTO_TEMPCHK	15:0	0x0000	0x0-0xffff	Число длинных последовательностей между автоматическими командами чтения регистра режима памяти MR4. [4)]

Таблица 5.262 – DENALI_CTL_30 (Controller Address 0x1e, AXI Address 0x78)

Название	Биты	Значение по умолчанию	Диапазон	Описание
MR2_DATA_0	30:16	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти 2 для CS 0. [4)]
MR1_DATA_0	14:0	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти 1 для CS 0. [4)]

Таблица 5.263 – DENALI_CTL_31 (Controller Address 0x1f, AXI Address 0x7c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
MR3_DATA_0	30:16	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти 3 для CS 0. [4)]
MRSINGLE_ DATA_0	14:0	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти при одиночной записи для CS0.[4)]

Таблица 5.264 – DENALI_CTL_32 (Controller Address 0x20, AXI Address 0x80)

Название	Биты	Значение по умолчанию	Диапазон	Описание
MR17_DATA_0	23:16	0x00	0x0-0xff	Данные для программирования в контрольный регистр памяти 17 для CS0. [4)]
MR16_DATA_0	15:8	0x00	0x0-0xff	Данные для программирования в контрольный регистр памяти 16 для CS0. [4)]
MR8_DATA_0	7:0	0x00	0x0-0xff	Данные чтения из MR8 для CS0. Только чтение. [4)]

Таблица 5.265 – DENALI_CTL_33 (Controller Address 0x21, AXI Address 0x84)

Название	Биты	Значение по умолчанию	Диапазон	Описание
MR1_DATA_1	30:16	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти 1 для CS1. [4)]
MR0_DATA_1	14:0	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти 0 для CS1. [4)]

Таблица 5.266 – DENALI_CTL_34 (Controller Address 0x22, AXI Address 0x88)

Название	Биты	Значение по умолчанию	Диапазон	Описание
MRSINGLE_ DATA_1	30:16	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти при одиночной записи для CS1. [4)]
MR2_DATA_1	14:0	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти 2 для CS1. [4)]

Таблица 5.267 – DENALI_CTL_35 (Controller Address 0x23, AXI Address 0x8c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
MR16_DATA_1	31:24	0x00	0x0-0xff	Данные для программирования в контрольный регистр памяти 16 для CS1. 4)
MR8_DATA_1	23:16	0x00	0x0-0xff	Данные чтения из MR8 для CS1. Только чтение. 4)
MR3_DATA_1	14:0	0x0000	0x0-0x7fff	Данные для программирования в контрольный регистр памяти 3 для CS1. 4)

Таблица 5.268 – DENALI_CTL_36 (Controller Address 0x24, AXI Address 0x90)

Название	Биты	Значение по умолчанию	Диапазон	Описание
ZQINIT	27:16	0x000	0x0-0xfff	Интервал времени, необходимый для ZQ-калибровки при инициализации.2)
LONG_COUNT_MASK	12:8	0x00	0x0-0x1f	Показывает величину длинной последовательности. Поддерживаются только значения 0x00 (1024 такта), 0x10 (512 тактов), 0x18 (256 тактов), 0x1C (128 тактов), 0x1E (64 такта) and 0x1F (32 такта).
MR17_DATA_1	7:0	0x00	0x0-0xff	Данные для программирования в контрольный регистр памяти 17 для CS1. 4)

Таблица 5.269 – DENALI_CTL_37 (Controller Address 0x25, AXI Address 0x94)

Название	Биты	Значение по умолчанию	Диапазон	Описание
ZQCS	27:16	0x000	0x0-0xfff	Интервал времени, необходимый для быстрой ZQ-калибровки.2)
ZQCL	11:0	0x000	0x0-0xfff	Интервал времени, необходимый для долгой ZQ-калибровки.2)

Таблица 5.270 – DENALI_CTL_38 (Controller Address 0x26, AXI Address 0x98)

Название	Биты	Значение по умолчанию	Диапазон	Описание
ZQ_ON_SREF_EXIT	11:8	0x0	0x0-0xf	Определяет тип ZQ калибровки, выполненной на выходе из собственной регенерации (0) выбирает ZQCS (1) – ZQCL (2) – ZQINIT (3) – ZQRESET Установить один из этих битов в 1 для разрешения.2)
ZQ_REQ	3:0	0x0	0x0-0xf	Запрос пользователя на выполнение ZQ калибровки. (0) запрос ZQCS (1) – ZQCL (2) – ZQINIT (3) – ZQRESET Установить один из этих битов в 1 для запуска. Только запись.2)

Таблица 5.271 – DENALI_CTL_39 (Controller Address 0x27, AXI Address 0x9c)

Название	Биты	Значение по умолчанию	Диапазон	Описание
ZQ_INTERVAL	31:0	0x00000000	0x0-0xffffffff	Число длинных последовательностей допустимое между автоматическими ZQCS командами (быстрыми ZQ-калибровками). Сброс параметра запрещает автоматические ZQ-калибровки.[2)]

Таблица 5.272 – DENALI_CTL_40 (Controller Address 0x28, AXI Address 0xa0)

Название	Биты	Значение по умолчанию	Диапазон	Описание
NO_ZQ_INIT	24	0x0	0x0-0x1	Запрет ZQ операций во время инициализации. Установить в 1 для запрета.[2)]
ZQRESET	19:8	0x000	0x0-0xfff	Число тактов, необходимых для ZQRESET команды.[2)]
ZQ_IN_PROGRESS	0	0x0	0x0-0x1	Показывает, что ZQ команда сейчас выполняется. Значение 1 показывает, что команда выполняется. Только для чтения.[2)]

Таблица 5.273 – DENALI_CTL_41 (Controller Address 0x29, AXI Address 0xa4)

Название	Биты	Значение по умолчанию	Диапазон	Описание
COL_DIFF	27:24	0x0	0x0-0xf	Разница между максимальной и используемой шириной адресации столбцов. [5)]
ROW_DIFF	18:16	0x0	0x0-0x7	Разница между максимальной и используемой шириной адресации строк. [5)]
BANK_DIFF	9:8	0x0	0x0-0x3	Кодированное число банков в микросхеме(ах) DRAM. 'b00 – все 8 банков используются 'b01 – 4 банка используются 'b10 – 2 банка используются 'b11 – зарезервировано [5)]
ZQCS_ROTATE	0	0x0	0x0-0x1	Выбирает, будет ли ZQCS калибровать только один CS или все CS. Установить в 1 для калибровки только одного CS и их ротации.[2)]

Таблица 5.274 – DENALI_CTL_42 (Controller Address 0x2a, AXI Address 0xa8)

Название	Биты	Значение по умолчанию	Диапазон	Описание
ADDR_CMP_EN	24	0x0	0x0-0x1	Разрешает обнаружение конфликтов адресов, как правило для постановки команд в очередь. Установить в 1 для разрешения. [5)]
COMMAND_ AGE_COUNT	23:16	0x00	0x0-0xff	Начальное значение индивидуальных счетчиков устаревания команды. [5)]
AGE_COUNT	15:8	0x00	0x0-0xff	Начальное значение мастер счетчика скорости устаревания команд. [5)]
APREBIT	3:0	0xa	0x0-0xf	Расположение бита авто закрытия (auto PRECHARGE) в адресе DRAM.

Таблица 5.275 – DENALI_CTL_43 (Controller Address 0x2b, AXI Address 0xac)

Название	Биты	Значение по умолчанию	Диапазон	Описание
PRIORITY_ EN	24	0x0	0x0-0x1	Разрешение приоритета как правила построения очереди команд. Установить в 1 для разрешения. 5)
PLACEMENT_ EN	16	0x0	0x0-0x1	Разрешение логики построения очереди команд. Установить в 1 для разрешения. 5)
BANK_ SPLIT_EN	8	0x0	0x0-0x1	Разрешение разделения банков как правила построения очереди команд. Установить в 1 для разрешения. 5)
ADDR_ COLLISION_ MPM_DIS	0	0x0	0x0-0x1	Запрещение расширения обнаружения конфликтов адресов, используя микромаски страниц для построения очереди команд и их выборки. Установить в 1 для запрещения. 5)

Таблица 5.276 – DENALI_CTL_44 (Controller Address 0x2c, AXI Address 0xa0)

Название	Биты	Значение по умолчанию	Диапазон	Описание
W2R_ SPLIT_EN	24	0x0	0x0-0x1	Разрешение разделения команд чтения от записи для одного CS как правила для построения очереди команд. 5)
CS_SAME_ EN	16	0x0	0x0-0x1	Разрешение группировки CS во время группировки чтения/записи как правила для построения очереди команд. Верно только при установке параметра RW_SAME_EN. Установить в 1 для разрешения. 5)
RW_SAME_ PAGE_EN	8	0x0	0x0-0x1	Разрешение группировки страниц во время группировки чтения/записи как правила для построения очереди команд. Верно только при установке параметра RW_SAME_EN. Установить в 1 для разрешения. 5)
RW_SAME_ EN	0	0x0	0x0-0x1	Разрешение группировки чтения/записи как правила для построения очереди команд. Установить в 1 для разрешения. 5)

Таблица 5.277 – DENALI_CTL_45 (Controller Address 0x2d, AXI Address 0xb4)

Название	Биты	Значение по умолчанию	Диапазон	Описание
DISABLE_RD_ INTERLEAVE	24	0x0	0x0-0x1	Запрещение чередования читаемых данных для команд одного порта независимо от ID запроса. 5)
SWAP_EN	16	0x0	0x0-0x1	Разрешение перестановки команд при их выборке из очереди. Установить в 1 для разрешения. 5)
NUM_Q_ ENTRIES_ ACT_ DISABLE	10:8	0x0	0x0-0x7	Число входов в очередь, в которых ACT запросы будут запрещены. Установка в X запретит ACT запросы от X входов нижних в очереди команд. 5)
DISABLE_ RW_GROUP_ W_BNK_ CONFLICT	1:0	0x0	0x0-0x3	Запрещение постановки в группу чтения/записи, когда группировка приводит к коллизии банков. Бит (0) контролирует постановку следующей за конфликтующей командой, бит (1) – постановку через одну от конфликтующей команды. Установить каждый бит в 1 для запрещения. 5)

Таблица 5.278 – DENALI_CTL_46 (Controller Address 0x2e, AXI Address 0xb8)

Название	Биты	Значение по умолчанию	Диапазон	Описание
LPDDR2_S4	24	0x0	0x0-0x1	Определяет тип используемого DRAM LPDDR2 устройства. Установить в 1 для LPDDR2-S4.
REDUC	16	0x0	0x0-0x1	Уменьшение ширины используемых данных до 16 бит. Установить в 1 для разрешения.
CS_MAP	9:8	0x0	0x0-0x3	Определяет активный CS.
INHIBIT_DRAM_CMD	1:0	0x0	0x0-0x3	Запрещает выполнение типов команд из очереди команд. Установить в 0 для разрешения любых команд, в 1 для запрета чтения/записи и команд банкам, в 2 для запрета MRR и периферийных MRR команд, в 3 для запрета команд MMR и чтения/записи. 5)

Таблица 5.279 – DENALI_CTL_47 (Controller Address 0x2f, AXI Address 0xbc)

Название	Биты	Значение по умолчанию	Диапазон	Описание
CTRLUPD_REQ	24	0x0	0x0-0x1	Объявляет, что контроллер инициировал сигнал запроса обновления dfi_ctrlupd_req. Установить в 1 для запуска. Только запись.8)
CONTROLLER_BUSY	16	0x0	0x0-0x1	Показывает, что контроллер обрабатывает команду. Очищает все порты для внешних транзакций. Значение 1 показывает, что контроллер занят. Только чтение.
IN_ORDER_ACCEPT	8	0x0	0x0-0x1	Заставляет контроллер принимать команды в порядке, в котором они помещены в очередь команд. 5)
Q_FULLNESS	2:0	0x0	0x0-0x7	Величина, определяющая полноту очереди команд. 5)

Таблица 5.280 – DENALI_CTL_48 (Controller Address 0x30, AXI Address 0xc0)

Название	Биты	Значение по умолчанию	Диапазон	Описание
INT_STATUS	21:8	0x0000	0x0-0x3fff	Регистр статуса прерываний контроллера. Только чтение. (13) – логическое или всех других битов (12) – перезапуск DLL завершен (11) – изменение сигнала dfi_init_complete после инициализации (неправильная работа PHY) (10) – текущая команда прервана пользовательской командой INHIBIT_DRAM_CMD (9) – программно запущенная запись контрольных регистров памяти завершена (8) – нарушен температурный режим (7) – автоматической чтение контрольного регистра памяти MR4 зафиксировало смену температуры или частоты регенерации (6) – программно запущенное чтение контрольных регистров памяти завершено (5) – ошибка обновления PHY (4) – выход из режима пониженного энергопотребления завершен (3) – инициализация контроллера завершена (2) – ошибка команды записи/чтения на внутренней шине (1) – множественный выход за допустимые границы адреса (0) – выход за допустимые границы адреса [3)]
CTRLUPD_REQ_ PER_AREF_EN	0	0x0	0x0-0x1	Разрешает автоматическое инициирование контроллером обновления (dfi_ctrlupd_req) после каждой регенерации. Установить в 1 для разрешения.[8)]

Таблица 5.281 – DENALI_CTL_49 (Controller Address 0x31, AXI Address 0xc4)

Название	Биты	Значение по умолчанию	Диапазон	Описание
INT_MASK	29:16	0x0000	0x0-0x3fff	Маска регистра прерываний INT_STATUS. Установить в 1 для маскирования соответствующего прерывания. [3)]
INT_ACK	12:0	0x0000	0x0-0x1fff	Очистка регистра прерываний INT_STATUS. [3)]

Таблица 5.282 – DENALI_CTL_50 (Controller Address 0x32, AXI Address 0xc8)

Название	Биты	Значение по умолчанию	Диапазон	Описание
OUT_OF_ RANGE_ADDR	30:0	0x00000000	0x0-0x7fffffff	Адрес команды, вышедший за допустимые пределы и вызвавший (0) или (1) прерывание. Только чтение.

Таблица 5.283 – DENALI_CTL_51 (Controller Address 0x33, AXI Address 0xcc)

Название	Биты	Значение по умолчанию	Диапазон	Описание
OUT_OF_RANGE_SOURCE_ID	23:16	0x00	0x0-0xff	ID источника команды, которая вызвала прерывание (0) или (1). Только чтение.
OUT_OF_RANGE_TYPE	13:8	0x00	0x0-0x3f	Тип команды, которая вызвала прерывание (0) или (1). Только чтение.
OUT_OF_RANGE_LENGTH	6:0	0x00	0x0-0x7f	Длина команды, которая вызвала прерывание (0) или (1). Только чтение.

Таблица 5.284 – DENALI_CTL_52 (Controller Address 0x30, AXI Address 0xd0)

Название	Биты	Значение по умолчанию	Диапазон	Описание
PORT_CMD_ERROR_ADDR	30:0	0x00000000	0x0-0x7fffffff	Адрес команды, которая вызвала прерывание (2). Только чтение.

Таблица 5.285 – DENALI_CTL_53 (Controller Address 0x31, AXI Address 0xd4)

Название	Биты	Значение по умолчанию	Диапазон	Описание
R2W_DIFFCS_DLY	26:24	0x1	0x0-0x7	Дополнительная задержка, вставляемая между чтениями и записями, для разных чип селектов. Программировать в не нулевое значение.
R2R_DIFFCS_DLY	18:16	0x1	0x0-0x7	Дополнительная задержка, вставляемая между чтениями, для разных CS. Программировать в не нулевое значение.
PORT_CMD_ERROR_TYPE	9:8	0x0	0x0-0x3	Тип ошибки и доступа, которые вызвали прерывание (2). (0) – FIXED команда получена по шине. Только чтение.
PORT_CMD_ERROR_ID	7:0	0x00	0x0-0xff	ID источника команды, которая вызвала прерывание (2). Только чтение.

Таблица 5.286 – DENALI_CTL_54 (Controller Address 0x32, AXI Address 0xd8)

Название	Биты	Значение по умолчанию	Диапазон	Описание
R2W_SAMECS_DLY	26:24	0x2	0x0-0x7	Дополнительная задержка, вставляемая между чтениями и записями, для одинаковых CS. Программировать в не нулевое значение.
R2R_SAMECS_DLY	18:16	0x0	0x0-0x7	Дополнительная задержка, вставляемая между чтениями, для одинаковых CS. Любое значение поддерживается, включая 0x0.
W2W_DIFFCS_DLY	10:8	0x1	0x0-0x7	Дополнительная задержка, вставляемая между записью и записью, для разных CS. Программировать в не нулевое значение.
W2R_DIFFCS_DLY	2:0	0x1	0x0-0x7	Дополнительная задержка, вставляемая между записями и чтениями, для разных CS. Допускается программирование в зависимости от системы памяти.

Таблица 5.287 – DENALI_CTL_55 (Controller Address 0x33, AXI Address 0xdc)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDQSCK_MIN	25:24	0x0	0x0-0x3	Минимальное значение интервала tDQSCK.[1)]
TDQSCK_MAX	18:16	0x0	0x0-0x7	Максимальное значение интервала tDQSCK.[1)]
W2W_SAMECS_ DLY	10:8	0x0	0x0-0x7	Дополнительная задержка, вставляемая между записями, для одинаковых CS. Любое значение поддерживается, включая 0x0.
W2R_SAMECS_ DLY	2:0	0x0	0x0-0x7	Дополнительная задержка, вставляемая между записями и чтениями, для одинаковых CS.

Таблица 5.288 – DENALI_CTL_56 (Controller Address 0x30, AXI Address 0xe0)

Название	Биты	Значение по умолчанию	Диапазон	Описание
CKE_STATUS	25:24	0x0	0x0-0x3	Регистровый доступ к сигналу cke_status. Только чтение.
AXI0_FIFO_ TYPE_REG	17:16	0x0	0x0-0x3	Соотношение доменов синхросигнала между AXI порт 0 и ядром контроллера. 2'd0 – если асинхронны 2’d1 – псевдосинхронные с соотношением частот порт/ядро равным 2/1 2’d2 – псевдосинхронные с соотношением частот порт/ядро равным 1/2 2’d3 синхронности.
AXI0_W_ PRIORITY	8	0x0	0x0-0x1	Приоритет команд записи 0 порта AXI. 0 - наивысший приоритет.
AXI0_R_ PRIORITY	0	0x0	0x0-0x1	Приоритет команд чтения 0 порта AXI. 0 - наивысший приоритет.

Таблица 5.289 – DENALI_CTL_57 (Controller Address 0x31, AXI Address 0xe4)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDFI_PHY_ WRLAT	28:24	Calc Value	0x0-0x1f	Хранит рассчитанный временной параметр DFI tPHY_WRLAT (в тактах DFI PHY), максимальное число тактов между командой записи и dfi_wrdata_en установкой. Только чтение. [6)]
DLL_RST_ ADJ_DLY	23:16	0x00	0x0-0xff	Минимальное число тактов после установки мастер задержки в DLL до установки сигнала сброса DLL dll_rst_n. Этот сигнал не используется и параметр можно игнорировать.
DLL_RST_ DELAY	15:0	0x0000	0x0-0xffff	Минимальное число тактов, на которое необходимо удерживать сигнал сброса DLL dll_rst_n. Этот сигнал не используется и параметр можно игнорировать.

Таблица 5.290 – DENALI_CTL_58 (Controller Address 0x32, AXI Address 0xe8)

Название	Биты	Значение по умолчанию	Диапазон	Описание
DRAM_CLK_ DISABLE	25:24	0x0	0x0-0x3	Установленное значение сигнала dfi_dram_clk_disable. (0) контролирует cs0 (1) контролирует cs1 и т.д. Установить каждый бит в 1 для запрещения. 6)
TDFI_ RDDATA_EN	20:16	Calc Value	0x0-0x1f	Содержит расчетный временной параметр DFI tRDDATA_EN (в тактах DFI PHY), максимальное число тактов между командой чтения и установкой dfi_rddata_en. Только чтение. 6)
TDFI_PHY_ RDLAT	12:8	0x06	0x0-0x1f	Определяет DFI tPHY_RDLAT временной параметр (в тактах DFI PHY), максимальное число тактов между установкой dfi_rddata_en и установкой dfi_rddata_valid. 6)
UPDATE_ ERROR_ STATUS	6:0	0x00	0x0-0x7f	Идентифицирует источник любой ошибки DFI MC- или PHY-инициированного обновления. Значение 1 показывает временную ошибку соответствующего временного параметра. Только чтение. 6)8)

Таблица 5.291 – DENALI_CTL_59 (Controller Address 0x33, AXI Address 0xec)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDFI_ CTRLUPD_ MAX	23:8	0x0000	0x0-0xffff	Определяет временной параметр DFI tCTRLUPD_MAX (в тактах DFI), максимальное количество тактов установки сигнала dfi_ctrlupd_req. Если запрограммирован в ненулевое значение, то временная ошибка вызовет прерывание и бит (1) установит в параметре UPDATE_ERROR_STATUS. 6)8)
TDFI_ CTRLUPD_ MIN	3:0	0x4	0x0-0xf	Показывает временной параметр DFI tCTRLUPD_MIN (в тактах DFI), минимальное количество тактов установки сигнала dfi_ctrlupd_req. Только чтение. 6)8)

Таблица 5.292 – DENALI_CTL_60 (Controller Address 0x30, AXI Address 0xf0)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDFI_ PHYUPD_ TYPE1	31:16	0x0000	0x0-0xffff	Определяет временной параметр DFI tPHYUPD_TYPE1 (в тактах DFI), максимальное число тактов установки сигнала dfi_phyupd_req после dfi_phyupd_ack для dfi_phyupd_type 1. Если запрограммирован в ненулевое значение, то временная ошибка вызовет прерывание и бит (3) установит в параметре UPDATE_ERROR_STATUS. 6)8)
TDFI_ PHYUPD_ TYPE0	15:0	0x0000	0x0-0xffff	Определяет временной параметр DFI tPHYUPD_TYPE0 (в тактах DFI), максимальное число тактов установки сигнала dfi_phyupd_req после dfi_phyupd_ack для dfi_phyupd_type 0. Если запрограммирован в ненулевое значение, то временная ошибка вызовет прерывание и бит (2) установит в параметре UPDATE_ERROR_STATUS. 6)8)

Таблица 5.293 – DENALI_CTL_61 (Controller Address 0x31, AXI Address 0xf4)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDFI_ PHYUPD_ TYPE3	31:16	0x0000	0x0-0xffff	Определяет временной параметр DFI tPHYUPD_TYPE3 (в тактах DFI), максимальное число тактов установки сигнала dfi_phyupd_req после dfi_phyupd_ack для dfi_phyupd_type 3. Если запрограммирован в ненулевое значение, то временная ошибка вызовет прерывание и бит (5) установит в параметре UPDATE_ERROR_STATUS. 6)8)
TDFI_ PHYUPD_ TYPE2	15:0	0x0000	0x0-0xffff	Определяет временной параметр DFI tPHYUPD_TYPE2 (в тактах DFI), максимальное число тактов установки сигнала dfi_phyupd_req после dfi_phyupd_ack для dfi_phyupd_type 2. Если запрограммирован в ненулевое значение, то временная ошибка вызовет прерывание и бит (4) установит в параметре UPDATE_ERROR_STATUS. 6)8)

Таблица 5.294 – DENALI_CTL_62 (Controller Address 0x32, AXI Address 0xf8)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDFI_ PHYUPD_ RESP	15:0	0x0000	0x0-0xffff	Определяет временной параметр DFI tPHYUPD_RESP (в тактах DFI), максимальное число тактов между установкой сигнала dfi_phyupd_req и установкой dfi_phyupd_ack. Если запрограммирован в ненулевое значение, то временная ошибка вызовет прерывание и бит (6) установит в параметре UPDATE_ERROR_STATUS. 6)8)

Таблица 5.295 – DENALI_CTL_63 (Controller Address 0x33, AXI Address 0xfc)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDFI_ CTRLUPD_ INTERVAL	31:0	0x00000000	0x0-0xffffffff	Определяет временной параметр DFI tCTRLUPD_INTERVAL (в тактах DFI), максимальное число тактов между установками сигнала dfi_ctrlupd_req. Если запрограммирован в не нулевое значение, то временная ошибка вызовет прерывание и бит (0) установит в параметре UPDATE_ERROR_STATUS. 6)8)

Таблица 5.296 – DENALI_CTL_64 (Controller Address 0x34, AXI Address 0x100)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDFI_ DRAM_ CLK_ DISABLE	27:24	0x0	0x0-0xf	Определяет временной параметр DFI tDRAM_CLK_DISABLE (в тактах DFI), задержка между установкой сигнала dfi_dram_clock_disable и отключением синхросигнала памяти. 6)
TDFI_ CTRL_ DELAY	19:16	0x2	0x0-0xf	Определяет временной параметр DFI tCTRL_DELAY (в тактах DFI), задержка между командой заряда DFI и командой памяти. 6)
WRLAT_ ADJ	12:8	0x00	0x0-0x1f	Поправка к времени записи PHY.
RDLAT_ ADJ	4:0	0x00	0x0-0x1f	Поправка к времени чтения PHY.

Таблица 5.297 – DENALI_CTL_65 (Controller Address 0x35, AXI Address 0x104)

Название	Биты	Значение по умолчанию	Диапазон	Описание
TDFI_ DRAM_ CLK_ ENABLE	3:0	0x0	0x0-0xf	Определяет временной параметр DFI tDRAM_CLK_ENABLE (в тактах DFI), задержка между снятием сигнала dfi_dram_clk_disable и включением синхросигнала памяти. 6)

Все параметры контроллера EMIC можно приблизительно разделить на группы:

1) временные параметры внешней памяти. Должны соответствовать спецификации на используемую память

2) параметры ZQ калибровки импеданса входных и выходных буферов – не функциональны в этом контроллере. ZQ калибровка управляется PHYC контроллером

3) параметры программного управления

4) параметры для обращения к контрольным регистрам внешней памяти

5) параметры обработки команд процессора

6) параметры интерфейса между контроллером и PHY – DFI interface

7) параметры переключения режимов питания LP – Low Power

8) параметры Update взаимодействия контроллера и PHY – не функциональны в этом контроллере

5.21.3 Регистровая модель контроллера физического уровня PHYC

Программно доступные регистры контроллера расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение PHYC Base = 0xFFF01400 и общий размер 4 Кб. Спецификация регистров приведена в таблице ниже.

Таблица 5.298 – Спецификация регистров PHYC

Адрес	Название регистра	Значение после сброса	Доступ	Описание
Base + 0x00h	DLL_CTRL	2002h	RW	управление DLL и цепочкой сдвига
Base + 0x04h	DRV_CTRL	7FFFh	RW	контроль выходного сопротивления выводов интерфейса LPDDR2
Base + 0x08h	RST_CTRL	E0h	RW	начальные настройки PHY
Base + 0x0Ch	PWR_CTRL	10h	RW	включение/выключение выходных буферов интерфейса LPDDR2
Base + 0x10h	PZ_DBG	0000h	RW	управление Z-калибровкой
Base + 0x14h	RST_REG	37h	RW	сброс PHY
Base + 0x18h	INIT_REG	0h	RW	инициализация PHY
Base + 0x1Ch	DDRC_STATUS_REG	00h	RD	состояние контроллера EMIC

Далее приведено подробное описание регистров.

5.21.3.1 Регистр DLL_CTRL

Данный регистр выбирает режим работы физической части интерфейса LPDDR2. Формат регистра приведен в таблице ниже (Таблица 5.299). Возможны два режима работы интерфейса

LPDDR2: с использованием DLL (режим DLL) и с использованием регистра сдвига (режим VDL). Выбор режима осуществляется из значения тактовой частоты интерфейса:

- до 100МГц используется режим VDL;
- от 125 МГц до 200 МГц используется режим DLL.

В режиме VDL поле **IVDLSEL** должно иметь значение 3'b111. Поля **TIME_INIT_DLL, IDSCXEN, IUDSET** можно оставить без изменений. Увеличение поля **IUDSET** требует увеличение параметра **TIME_INIT_DLL**.

Таблица 5.299 – DLL_CTRL – Регистр управления DLL и цепочкой сдвига

разряды регистра	названия поля	описание	значение по умолчанию
[15]	IVDLEN	Выбор между DLL и сдвиговыми регистрами 0 - DLL используется для сдвига 1 - используется сдвиговый регистр (VDL mode)	1'b0
[14:12]	TIME_INIT_DLL	задает время инициализации DLL 000 - 4064 такта 001 - 8160 тактов 010 - 16352 такта 011 - 32736 тактов 100 - 65504 тактов 101 - 131040 тактов 110 - 262112 тактов 111 - 524256 тактов	3'b010
[11:9]	IVDLSEL	задержка цепи 000 - самая маленькая; ... 111 - самая большая.	3'b000
[8]	IDSCXEN	разрешение изменения IDEGSL, ISFT[2:0] 0 - при изменении IDEGSL, ISFT[2:0] будет изменена задержка; 1 - при изменении IDEGSL, ISFT[2:0] не будет изменена задержка.	1'b0
[7]	IDEGSL	IDEGSL, ISFT[2:0] - задает сдвиг линии задержки относительно входа. 0 000 - 0 градусов 0 001 - 11.25 градусов 0 010 - 22.5 градуса ..... 1 000 - 90 градусов 1 001 - 101.25 градуса ..... 1 111 - 168.75 градусов.	1'b0
[6:4]	ISFT[2:0]		3'b000
[3:0]	IUDSET	определяет порог для фильтра обратной связи. Чем больше значение, тем стабильней будет захват и меньше будет влиять джиттер входного сигнала, но тем больше будет время выхода PLL на рабочий режим.	4'b0010

5.21.3.2 Регистр DRV_CTRL

Данный регистр позволяет настроить выходное сопротивление буферов интерфейса LPDDR2. Формат регистра приведен в таблице ниже (Таблица 5.300) Значение выходного со-

противления буфера должно быть как можно ближе к волновому сопротивлению линии (как правило, 50 Ом), так как при этом отражения в линии будут минимальны.

Таблица 5.300 – DRV_CTRL – Регистр контроля выходного сопротивления выводов интерфейса LPDDR2

разряды регистра	названия поля	описание
[14]	ICDRVOFS_CK0	определяют выходное сопротивление CK0 ICDRVOFS_CK0, ICDRV_ CK0[1], ICDRV_ CK0[0] 011b - 80 Ом 100b - 60 Ом 101b - 48 Ом 110b - 40 Ом 111b - 34 Ом
[13:12]	ICDRV_CK0[1:0]	
[11]	ICDRVOFS_CKE	определяют выходное сопротивление CKE ICDRVOFS_CKE, ICDRV_ CKE[1], ICDRV_ CKE[0] 011b - 80 Ом 100b - 60 Ом 101b - 48 Ом 110b - 40 Ом 111b - 34 Ом
[10:9]	ICDRV_CKE[1:0]	
[8]	ICDRVOFS_CSn	определяют выходное сопротивление CS ICDRVOFS_CSn, ICDRV_CSn[1], ICDRV_CSn[0] 011b - 80 Ом 100b - 60 Ом 101b - 48 Ом 110b - 40 Ом 111b - 34 Ом
[7:6]	ICDRV_CSn[1:0]	
[5]	ICDRVOFS_CA	определяют выходное сопротивление CA ICDRVOFS_CA, ICDRV_CA[1], ICDRV_CA[0] 011b - 80 Ом 100b - 60 Ом 101b - 48 Ом 110b - 40 Ом 111b - 34 Ом
[4:3]	ICDRV_CA[1:0]	
[2]	ICDRVOFS_DQ	определяют выходное сопротивление DQ, DM, DQS ICDRVOFS_DQ, ICDRV_DQ[1], ICDRV_DQ[0] 011b - 80 Ом 100b - 60 Ом 101b - 48 Ом 110b - 40 Ом 111b - 34 Ом
[1:0]	ICDRV_DQ[1:0]	

5.21.3.3 Регистры RST_REG и INIT_REG

Данные регистры управляют начальной инициализацией контроллера PHYC и Z-калибровкой.

Z-калибровка - процесс при котором с помощью эталонного внешнего резистора происходит точная подстройка выходного сопротивления буферов интерфейса LPDDR2. Данную калибровку необходимо проводить один раз перед началом работы.

Начальная инициализация возможна двумя способами: программным и автоматическим. Программный способ инициализации заключается в записи последовательности необходимых значений в регистр RST_REG (

Таблица 5.301). При этом программист отвечает за соблюдение всех необходимых временных параметров. Временные диаграммы инициализации с временными ограничениями приведены на рисунках ниже (см. Рисунок 5.126 и Рисунок 5.127). После окончания инициали-

зации должен быть установлен сигнал CACKDQDQS_INIT_COMPETE, который говорит о окончании инициализации. Автоматический способ инициализации запускается установкой соответствующего бита регистра INIT_REG (Таблица 5.302). После этого все управляющие сигналы, которые содержит регистр RST_REG, формируются в нужной последовательности с соблюдением всех временных интервалов.

Таблица 5.301 – RST_REG – Регистр сброса PHY

разряды регистра	названия поля	описание	значение по умолчанию
[7]	ISUSPZQ	режим низкого потребления ячейкой 0 - режим нормального потребления 1 - режим пониженного потребления	1'b0
[6]	IPZQ_INIT_UPD	начало инициализации	1'b0
[5]	IPZQRST	сброс ZQ ячейки	1'b1
[4]	IRST	сброс ZQ ячейки	1'b1
[3]	CACKDQDQS_INIT_COMPETE	закончена инициализация адресной части	1'b0
[2]	IDLLRST	сброс DLL	1'b1
[1]	IUSRRST	аналоговый сброс	1'b1
[0]	IRESET	системный сброс PHY	1'b1

Таблица 5.302 – INIT_REG – Регистр инициализации PHY

разряды регистра	названия поля	описание	значение по умолчанию
[2]	IRXDQLATENCYINIT	требует установки из 0 в 1 после инициализации контроллера PHYC	1'b0
[1]	INIT_PHY	инициализация PHY (автоматический режим)	1'b0
[0]	INIT_ZQ	инициализация калибровки (автоматический режим)	1'b0

После установки бита INIT_PHY регистра INIT_REG происходит необходимая автоматическая установка сигналов CACKDQDQS_INIT_COMPETE, IDLLRST, IUSRRST, IRESET. Данные сигналы могут быть установлены путем записи в регистр RST_REG программно.

После установки бита INIT_ZQ происходит необходимая автоматическая установка сигналов ISUSPZQ, IPZQ_INIT_UPD, IPZQRST, IRST. Данные сигналы могут быть установлены путем записи в регистр RST_REG программно.

Когда одновременно установлены сигналы ISUSPZQ и CACKDQDQS_INIT_COMPETE, происходит установка сигнала DFI_INIT_COMPLETE.

В процессе работы возможна повторная инициализация путем повторной установки INIT_PHY, INIT_PZQ. Установка возможна как по отдельности, так и обоих разрядов сразу.

Бит IRXDQLATENCYINIT следует установить программно после инициализации микросхем памяти.

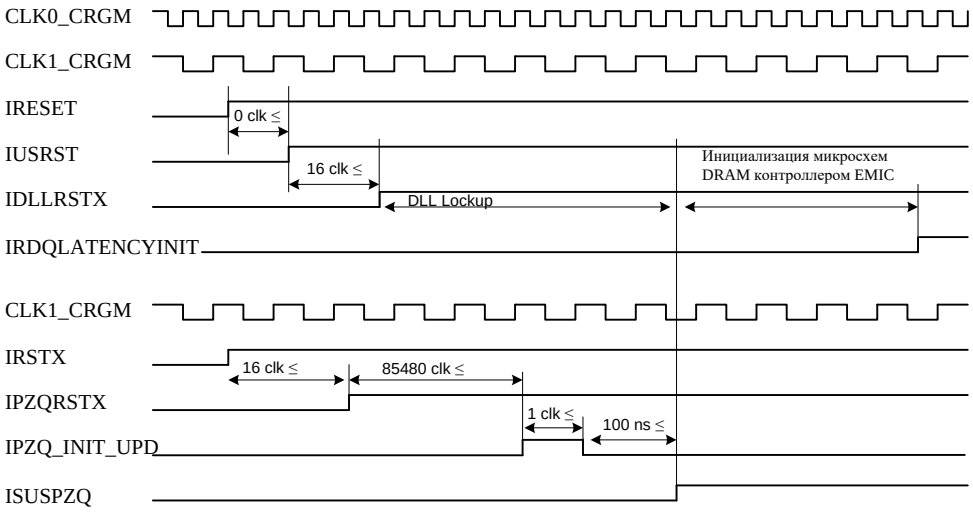


Рисунок 5.126 – инициализация PHYC в режиме DLL

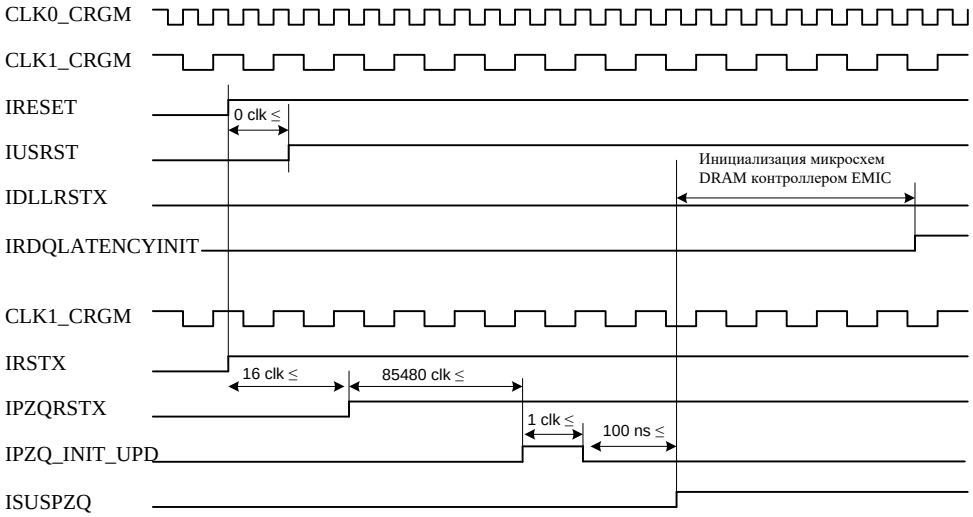


Рисунок 5.127 – инициализация PHYC в режиме VDL

5.21.3.4 Регистры, не требующие программной настройки, PZ_DBG, RST_CNTRL и PWR_CTRL

Регистры RST_CNTRL и PWR_CTRL служат для управления буферами. Формат регистров приведен в таблицах ниже (Таблица 5.303 и Таблица 5.304). Данные регистры не требуют программной модификации во время работы, а так же начальной инициализации. После системного сброса все поля регистров принимают необходимые для работы значения.

Регистры PZ_DBG служит для тестовых целей и в функциональном режиме не используется. Формат регистра приведен в таблице ниже (Таблица 5.305).

Таблица 5.303 – RST_CTRL – Регистр управления выходными буферами

разряды регистра	названия поля	описание	значение по умолчанию
[7]	IDQSGATE_EN	DQS gating control 0 - gating control not used 1 - gating control used	1'b1
[6]	IRXENMASK	0 - DQS mask enable 1- DQS mask disable	1'b1
[5]	IDQSDRIVE	Управляются ли выходы DQS во время, кода нет чтения. 0 - выходы не управляются 1 - выходы управляются	1'b1
[4]	IDQDRIVE	Управляются ли выходы DQ во время, когда нет чтения. 0 - выходы не управляются 1 - выходы управляются	1'b0
[3:2]	ICSCLPR	определяет выход OTXCS после и во время системного сброса ICSCLPR[0] =0 => OTXCS[0] =L; ICSCLPR[0] =1 => OTXCS[0] =H ICSCLPR[1] =0 => OTXCS[1] =L; ICSCLPR[1] =1 => OTXCS[1] =H	2'b0 асинхронный сброс
[1:0]	ICKECLPR	определяет выход OTXCKE после и во время системного сброса ICKECLPR[0] =0 => OTXCKE[0] =L; ICKECLPR[0] =1 => OTXCKE[0] =H ICKECLPR[1] =0 => OTXCKE[1] =L; ICKECLPR[1] =1 => OTXCKE[1] =H	2'b0 асинхронный сброс

Таблица 5.304 – PWR_CTRL – Регистр управления питанием буферов

разряды регистра	названия поля	описание	значение по умолчанию
[4]	ISUSPD_CK1	Управление буфером CK1 0 - буфер работает 1 - буфер не работает	1'b1
[3]	ISUSPD_CK0	Управление буфером CK0 0 - буфер работает 1 - буфер не работает	1'b0
[2]	ISUSPD_CACS	Управление буферами CA и CS 0 - буфер работает 1 - буфер не работает	1'b0
[1]	ISUSPR_DQ_OFF	динамическое управление SUSPR для буферов DQ 0 - управление разрешено; 1 - управление запрешено;	1'b0
[0]	ISUSPR_DQS_OFF	динамическое управление SUSPR для буферов DQS 0 - управление разрешено; 1 - управление запрешено;	1'b0

Таблица 5.305 – PZ_DBG – Регистр управления отладкой ZQ-калибровки

разряды регистра	названия поля	описание	значение по умолчанию
[13]	OPZQCAL_TRIG_EN	разрешение выдачи времени поиска PZQ блока	1'b0
[12]	OPZQCAL_ERR_EN	разрешение выдачи ошибки калибровки	1'b0
[11]	IDEBUG_SEL_PZQ		1'b0
[10]	IPZQUSR_USE		1'b0
[9:5]	IPZQDP_USR		5'b0
[4:0]	IPZQDN_USR		5'b0

Таблица 5.306 – DDRC_STATUS_REG – регистр состояния контроллера LPDDR2

разряды регистра	названия поля	описание
[5]	q_almost_full	показывает, что очередь команд достигла значения *q_fullness*
[4]	refresh_in_process	1 - контроллер выполняет команду refresh
[3:2]	cke_status	показывает что память находится в режиме self-refresh или питание отключено.
[1]	controller_busy	1 - контроллер выполняет команду чтения или записи
[0]	port_busy	1 - порт обслуживает команду записи или чтения

5.21.4 Стандартный алгоритм инициализации контроллеров

- Запись необходимых параметров в регистры контроллера EMIC за исключением параметра START
- Запуск работы контроллера EMIC
- Запись необходимых параметров в регистры контроллера PHYC
- Запуск работы контроллера PHYC
- Ожидание окончания инициализации контроллера PHYC
- Ожидание окончания инициализации контроллера EMIC
- Установка сигнала IRXDQLATENCYINIT

5.21.5 Режимы пониженного потребления

Во многих случаях желательно минимизировать мощность, потребляемую внешней памятью и управляющим контроллером. Это достигается программированием контроллера EMIC. Всего есть семь режимов пониженного потребления. Их описание приведено ниже в порядке возрастания экономии энергии.

5.21.5.1 Активный пониженный (Active Power-Down)

Контроллер переводит внешнюю динамическую память в режим пониженного потребления, пока любая строка активна. В этом режиме и контроллер, и память полностью работоспособны, но сигнал CKE снят. При этом память входит в активный или закрытый режим пониженного потребления в зависимости от наличия открытых строк.

Если параметр LOWPOWER_REFRESH_ENABLE снят, то, находясь в текущем режиме, контроллер будет продолжать переводить память в состояние регенерации. При этом сигнал CKE будет устанавливаться на период регенерации. Регенерация включает в себя команду закрытия (PRECHARGE ALL) всех банков, поэтому, по ее окончании память войдет в закрытый режим, а сигнал CKE снимется снова.

5.21.5.2 Активный пониженный с отключением синхросигнала памяти (Active Power-Down With Memory Clock Gating)

Данный режим представляет собой активный пониженный, дополненный отключением тактового сигнала. При регенерации синхросигнал CK0 будет подключен до установки CKE.

5.21.5.3 Закрытый пониженный (Precharge Power-Down)

Контроллер переводит внешнюю динамическую память в режим пониженного потребления, если все банки не используются. Если хотя бы одна строка активна, то перед входом в режим контроллер закроет все банки (PRECHARGE ALL).

Если параметр LOWPOWER_REFRESH_ENABLE снят, то, находясь в текущем режиме, контроллер будет продолжать переводить память в состояние регенерации. При этом сигнал CKE будет устанавливаться на период регенерации. По окончании регенерации сигнал CKE снимется снова.

5.21.5.4 Закрытый пониженный с отключением синхросигнала памяти (Precharge Power-Down With Memory Clock Gating)

Данный режим представляет собой закрытый пониженный, дополненный отключением тактового сигнала. При регенерации синхросигнал CK0 будет подключен до установки CKE.

5.21.5.5 Собственная регенерация (Self-Refresh)

Контроллер переводит внешнюю динамическую память в режим собственной регенерации. При этом тактовые сигналы контроллера и памяти включены, сигнал CKE снят. В данном режиме память регенерирует самостоятельно.

5.21.5.6 Собственная регенерация с отключением синхросигнала памяти (Self-Refresh With Memory Clock Gating)

Контроллер переводит внешнюю динамическую память в режим собственной регенерации, отключает тактовый сигнал CK0. При выходе из этого состояния тактовый сигнал будет подан снова.

5.21.5.7 Собственная регенерация с отключением синхросигнала памяти и контроллера (Self-Refresh With Memory And Controller Clock Gating)

Режим с наименьшим потреблением энергии. Контроллер переводит внешнюю динамическую память в режим собственной регенерации, отключает тактовый сигнал CK0, а также собственный тактовый сигнал.

При выходе из этого режима все тактовые сигналы будут подключены снова. Если автоматический выход из этого состояния разрешен, новая запись или чтение выведут память в активное состояние.

В этот режим нельзя входить во время записи или чтения данных. Необходимо предварительно убедиться, что контроллер не выполняет никаких команд, посредством чтения параметра CONTROLLER_BUSY.

5.21.6 Управление режимами пониженного потребления

Есть два способа переключения между режимами питания – программный и автоматический. Выбор между ними осуществляется схемой арбитража. Программное управление имеет больший приоритет, чем автоматическое. Параметр LP_ARB_STATE показывает используемый способ.

5.21.6.1 Программное управление

Программный интерфейс использует параметр LP_CMD для входа и выхода из любого режима. Результат можно наблюдать в регистрах LP_ARB_STATE и LP_STATE. Запрос на переход в более глубокий режим экономии исполняется. Запрос на переход в менее глубокий режим (за исключением выхода в рабочий режим) игнорируется.

Программный интерфейс имеет возможность блокировки схемы арбитража посредством записи (7) бита LOCK в параметре LP_CMD. Это дает уверенность, что между чтением режима и командой программного изменения, он не будет изменен автоматически.

5.21.6.2 Автоматическое управление

Контроллер EMIC поддерживает автоматический вход в любой режим пониженного потребления, основанный на программируемых параметрах и отслеживании отсутствия передачи данных. Для каждого режима есть свой счетчик, начинающий работу при бездействии контроллера. Соответствие режимов и параметров приведено в таюлице ниже (Таблица 5.307). Как и при программном управлении, возможен переход только в менее потребляющий режим.

Автоматический интерфейс также поддерживает автоматический выход из любого режима пониженного потребления, разрешенного в параметре LP_AUTO_EXIT_EN. Если выход из текущего состояния разрешен, то он осуществится при получении контроллером команды записи или чтения.

Таблица 5.307 – Параметры автоматического управления

Режим потребления	Разрешающий параметр	Параметр счетчика
Активный пониженный	LP_AUTO_ENTRY_EN [0] = 1’b1 LP_AUTOMEM_GATE_EN [0] = 1’b0	LP_AUTO_PD_IDLE
Активный пониженный с отключением синхросигнала памяти	LP_AUTO_ENTRY_EN [0] = 1’b1 LP_AUTOMEM_GATE_EN [0] = 1’b1	LP_AUTO_PD_IDLE
Закрытый пониженный	LP_AUTO_ENTRY_EN [0] = 1’b1 LP_AUTOMEM_GATE_EN [0] = 1’b0	LP_AUTO_PD_IDLE
Закрытый пониженный с отключением синхросигнала памяти	LP_AUTO_ENTRY_EN [0] = 1’b1 LP_AUTOMEM_GATE_EN [0] = 1’b1	LP_AUTO_PD_IDLE
Собственная регенерация	LP_AUTO_ENTRY_EN [1] = 1’b1 LP_AUTOMEM_GATE_EN [1] = 1’b0	LP_AUTO_SR_IDLE
Собственная регенерация с отключением синхросигнала памяти	LP_AUTO_ENTRY_EN [1] = 1’b1 LP_AUTOMEM_GATE_EN [1] = 1’b1	LP_AUTO_SR_IDLE
Собственная регенерация с отключением синхросигнала памяти и контроллера	LP_AUTO_ENTRY_EN [2] = 1’b1	LP_AUTO_SR_MC_GATE_IDLE

5.21.7 Проверка правильности адреса

При использовании памяти не максимального размера, ширина адреса должна быть уменьшена с помощью параметров BANK_DIF, ROW_DIFF и COL_DIFF. При этом появляется возможность обращения процессорного ядра по несуществующему адресу. Контроллер EMIC способен отслеживать такие ситуации.

При получении команды с адресом, выходящим за допустимые границы, контроллер установит параметры, перечисленные в таблице ниже (Таблица 5.308), заблокирует их последующую запись, а также выдаст соответствующее прерывание (если оно не маскировано). Чтение параметров разблокирует их и сбросит. Если от момента установки параметров до момента их чтения были получены новые команды с ошибочным адресом, то будет сообщено о множественной ошибке.

Несмотря на несоответствие адреса нужным границам, команды будут исполняться, чтобы избежать зависания внутренней шины. Команды чтения при этом вернут случайные данные. Команды записи могут переписать информацию во внешней памяти.

Таблица 5.308 – Параметры проверки правильности адреса

Название параметра	Описание параметра
OUT_OF_RANGE_ADDR [30:0]	адрес, вышедший за разрешенные границы
OUT_OF_RANGE_SOURCE_ID [7:0]	ID источника ошибочной команды
OUT_OF_RANGE_LENGTH [6:0]	общее количество байт данных в ошибочной команде
OUT_OF_RANGE_TYPE [5:0]	тип ошибочной команды 'b000000 – Non-Exclusive Write 'b000001 – Non-Exclusive Read 'b000010 – Non-Exclusive Masked Write 'b000100 – Wrapped Write 'b000101 – Wrapped Write 'b000110 – Wrapped Masked Write 'b001000 – Exclusive Write 'b001001 – Exclusive Write 'b001010 – Exclusive Masked Write 'b010000 – Flushed Write 'b100000 – Non-Exclusive Write with Auto-Precharge 'b100001 – Non-Exclusive Read with Auto-Precharge 'b100010 – Non-Exclusive Masked Write with Auto-Precharge 'b100100 – Wrapped Write with Auto-Precharge 'b100101 – Wrapped Read with Auto-Precharge 'b100110 – Wrapped Masked Write with Auto-Precharge 'b101000 – Exclusive Write with Auto-Precharge 'b101001 – Exclusive Read with Auto-Precharge 'b101010 – Exclusive Masked Write with Auto-Precharge 'b110000 – Flushed Write with Auto-Precharge

5.21.8 Взаимодействие с регистрами внешней памяти

Контроллер EMIC предоставляет возможность взаимодействия с контрольными регистрами внешней памяти. При запуске EMIC автоматически исполняются команды записи содержимого MRx_REGx параметров в соответствующие контрольные регистры памяти – Mode Registers (MRx).

Помимо автоматической, возможна ручная программная запись в регистры MRx.

Программной записью управляет параметр WRITE_MODEREG. Данные для записи, в зависимости от установки, могут быть взяты из параметров MRx_DATA_x и MRSINGLE_DATA_x. По завершении записи будет установлен бит (9) в параметре INT_STATUS и выдано прерывание (если не маскировано). В случае ошибки записи будет установлено соответствующее значение в параметр MRW_STATUS.

Программным чтением управляет параметр READ_MODEREG. По завершении чтения будет установлен бит (6) в параметре INT_STATUS и выдано прерывание (если не маскировано). Считанные данные будут доступны в параметре PERIPHERAL_MRR_DATA.

Памяти типа LPDDR2 имеют встроенный датчик температуры, и ее значение хранится в контрольном регистре MR4. Контроллеру можно разрешить автоматическое считывание этого регистра, установив параметр периода чтения REFRESH_PER_AUTO_TEMPCHK в ненулевое значение. Считанное значение температуры будет доступно в параметре AUTO_TEMPCHK_VAL. При изменении температуры будет установлен бит (7) в параметре INT_STATUS и выдано прерывание (если не маскировано). При критически низкой или высокой температуре будет установлен бит (8) в параметре INT_STATUS и выдано прерывание (если не маскировано).

Подробное описание контрольных регистров памяти и содержащихся в них значений приведено в документации на память.

5.22 Загрузочное ПЗУ BROM

СБИС СНП-ВП имеет банк памяти ПЗУ объёмом 4 Кб, который содержит программу начального загрузчика. Начальная загрузка микросхемы возможна через интерфейсы SPI, UART0, UART1 и Ethernet. Выбор режима загрузки осуществляется внешними входами BOOTM2, BOOTM1 и BOOTM0 в соответствии с таблицей:

Таблица 5.309 – Варианты начальной загрузки СБИС СНП-ВП

BOOTM2	BOOTM1	BOOTM0	Интерфейс для загрузки
0	0	0	SPI
0	0	1	UART0
0	1	0	UART1
0	1	1	Зарезервировано
1	-	-	Ethernet

Начальный загрузчик – это программа, которая начинает выполняться после снятия внешнего сигнала сброса и инициализации схемы ФАПЧХ внутри СБИС СНП-ВП. Задача этой программы загрузить во внутреннюю память процессорного ядра ARM следующую исполняемую программу и передать ей управление. Этой следующей программой может быть, например, загрузчик операционной системы.

Двоичный образ начального загрузчика (см. Рисунок 5.128(а)) находится в доступной только для чтения памяти (ROM, адреса 0x00040000-0x00040FFF), начиная с адреса 0x00040000. Первые восемь команд начального загрузчика составляют таблицу векторов прерываний. Каждый вектор таблицы – это одна команда перехода на соответствующую номеру вектора процедуру обработки прерывания.

Двоичный образ загружаемой программы (см. Рисунок 5.128(б)) в процессе загрузки помещается во внутреннюю память микросхемы. Для SPI, UART0, UART1 двоичный образ загружаемой программы должен содержать в последнем слове CRC32 свертку всех остальных слов. Для SPI, кроме того, двоичный образ загружаемой программы должен содержать в шестом слове служебное слово загрузчика (см. Рисунок 5.128).

Начальная загрузка происходит следующим образом. После снятия внешнего сигнала сброса и инициализации схемы ФАПЧХ, устанавливается начальное отображение памяти (см. п. 3.2, отображение «Before Remap»), при котором область памяти 0x00040000 – 0x00040FFF отображается на область памяти 0x00000000 – 0x00000FFF, и управление передается команде с адресом 0x00000000. Эта команда выполняет переход на точку входа начального загрузчика.

Начальный загрузчик выбирает устройство для считывания двоичного образа программы, считывает этот образ в область памяти, начиная с адреса 0xC0000000, меняет отображение памяти (см. п. 3.2, отображение «After Remap») так, что область памяти 0xC0000000 – 0xC003FFFF отображается на область памяти 0x00000000 – 0x0003FFFF, и передает управление команде с адресом 0xC0000000. Работа начального загрузчика на этом закончена. Схематическое представление двоичных образов начального загрузчика и загружаемой программы приведено на рисунке ниже (см. Рисунок 5.128).

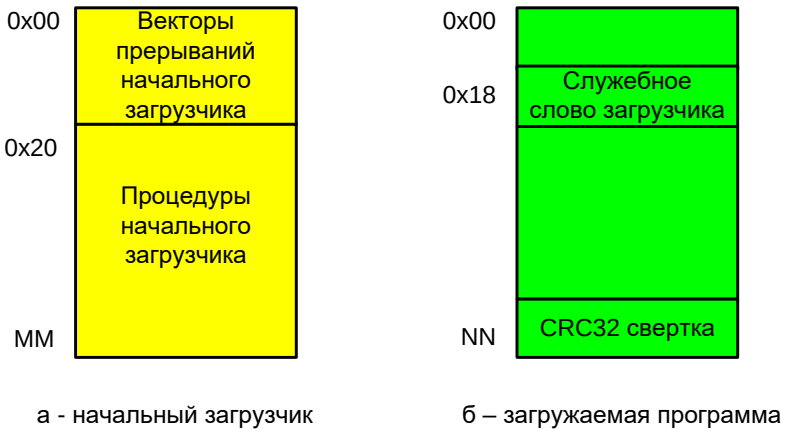


Рисунок 5.128 – Двоичные образы начального загрузчика и загружаемой программы

Изменение состояния памяти в процессе загрузки программы приведено на рисунке ниже (см. Рисунок 5.129).

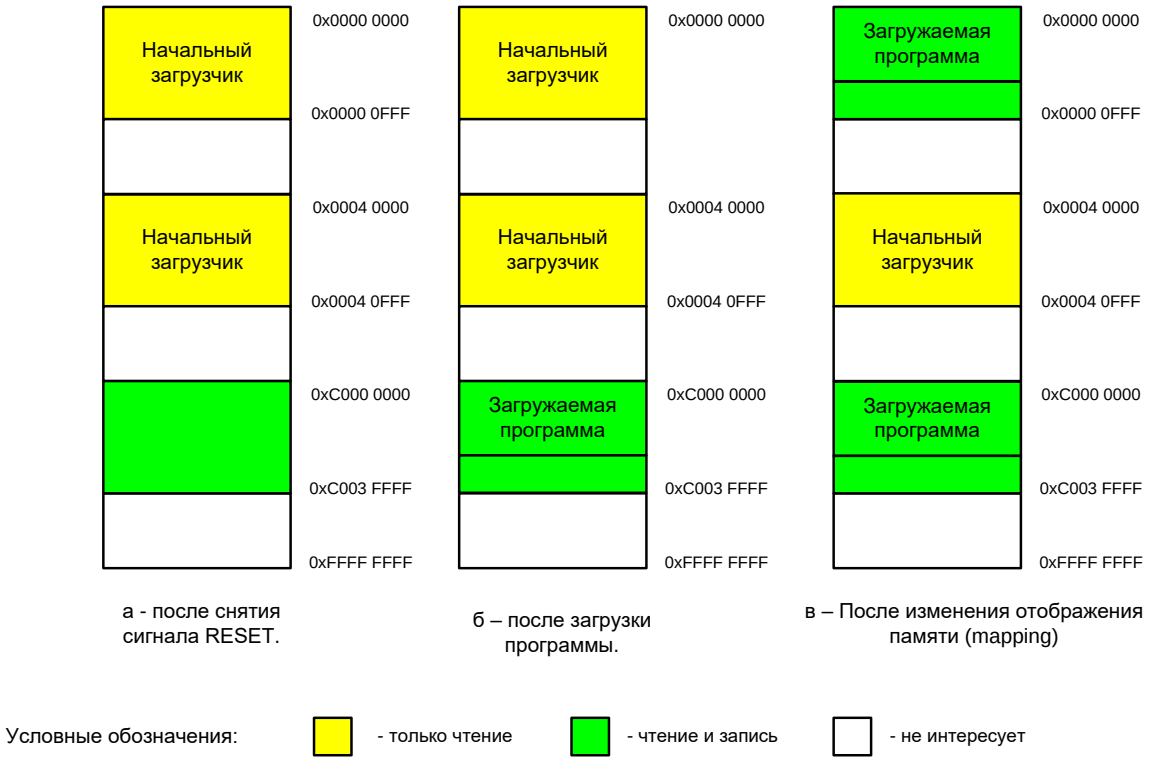


Рисунок 5.129 – Последовательность состояний памяти в процессе загрузки

5.22.1 Алгоритм работы начального загрузчика

Работа начального загрузчика зависит от состояния четырех внешних входов микросхемы BOOTM1, BOOTM2, TST_I1 и TST_I2. Алгоритм работы начального загрузчика приведен на рисунке ниже (см. Рисунок 5.130).

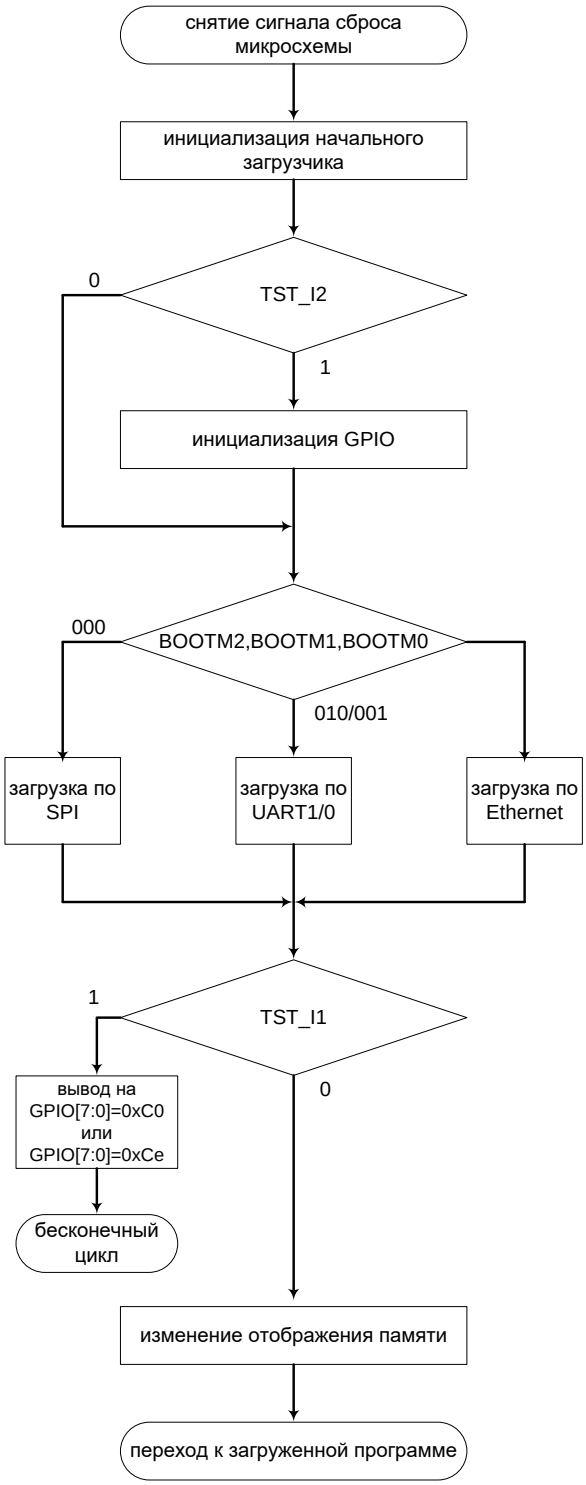


Рисунок 5.130 – Алгоритм работы начального загрузчика

5.22.2 Состояние процессора перед началом исполнения загруженной программы

Непосредственно перед началом исполнения загруженной начальным загрузчиком программы значения всех регистров процессорного ядра ARM, кроме PC и CPSR, считаются неопределенными. Регистры PC и CPSR имеют следующие значения.

Регистр PC.

PC = 0xC0000000

Регистр CPSR.

Определены значения следующих разрядов регистра CPSR.

[4:0] = b10011 – Supervisor mode
[5] = 0 – Thumb state
[6] = 0 – IRQ disable
[7] = 0 – FIQ disable
[8] = 0 – Imprecise abort disable
[9] = 0 – Endianess
[24] = 0 – Java state

Значения остальных разрядов регистра CPSR не определены. Пример загружаемой программы

В этом разделе приведена возможная схема программы, которая правильно работает после того, как она загружена и запущена начальным загрузчиком. При описании использован ассемблер ARM.

В результате отображения памяти, используемого при работе программы, загружаемая программа оказывается в двух адресных пространствах одновременно. Первое адресное пространство начинается с адреса 0x00000000, а второе – с адреса 0xC0000000. Программа компилируется в предположении, что она находится во втором адресном пространстве.

Первый фрагмент программы содержит вектора прерываний и располагается в памяти, начиная с адреса 0xC0000000. Этот фрагмент написан так, что он правильно работает в обоих адресных пространствах. Для этого вместо команд перехода использованы команды загрузки в регистр PC значений, адреса которых задаются смещениями относительно адресов соответствующих команд загрузки.

```
AREA Vect, CODE, READONLY
   IMPORT Reset_Handler
   IMPORT Undefined_Handler
   IMPORT SWI_Handler
   IMPORT Prefetch_Handler
   IMPORT Abort_Handler
   IMPORT IRQ_Handler
   IMPORT FIQ_Handler

ENTRY
   LDR PC, Reset_Addr
   LDR PC, Undefined_Addr
   LDR PC, SWI_Addr
   LDR PC, Prefetch_Addr
   LDR PC, Abort_Addr
   NOP ; На месте этого вектора в режиме загрузки по SPI
      ; должно находиться служебное слово загрузчика.
   LDR PC, IRQ_Addr
   LDR PC, FIQ_Addr

Reset_Addr      DCD Reset_Handler
Undefined_Addr  DCD Undefined_Handler
SWI_Addr        DCD SWI_Handler
Prefetch_Addr   DCD Prefetch_Handler
Abort_Addr      DCD Abort_Handler
            DCD 0; Reserved vector
IRQ_Addr        DCD IRQ_Handler
```

```
FIQ_Addr        DCD FIQ_Handler
END
```

На написание второго фрагмента программы ограничений не накладывается и, в общем случае, он может работать только во втором адресном пространстве.

```
AREA MainCode, CODE, READONLY
;
;   Далее следует остальной код программы
;
....
....
;   Сюда добавляется свертка crc32
END
```

5.22.3 Начальная загрузка через порт SPI

Загрузка через SPI предполагает загрузку программы из SPI флэш-памяти. Если предполагается этот способ начальной загрузки, то подключаемая микросхема памяти должна удовлетворять следующим условиям:

- поддержка SPI mode 3,
- частота работы не менее $F_{HPMP}/8$ = 10 МГц, т.к. начальный делитель опорного синхросигнала, выставляемый загрузчиком – 8,
- поддержка команды FAST_READ (код – 0x0B),
- загрузочная flash-память должна быть подключена к выводу SPI_CS0,
- максимальный объем flash-памяти 16Мбайт.

5.22.3.1 Считывание данных из флэш-памяти

При считывании данных из флэш-памяти загрузчик передает и принимает данные через SPI 16-разрядными словами. Ниже каждое такое слово представлено как пара <старший байт> <младший байт>.

Для считывания фрагмента данных из флэш-памяти загрузчик через SPI передает следующую последовательность слов:

```
Cm a2
a1 a0
xx xx
xx xx
...
...
```

Значения байтов передаваемой последовательности слов таковы:

- cm – команда быстрого чтения из флэш-памяти (FAST_READ = 0x0B);
- a2, a1, a0 – 24-разрядный адрес первого считываемого из флэш-памяти байта;
- xx – байты с любым значением.

В ответ загрузчик получает через SPI следующую последовательность слов:

```
Xx xx
xx xx
xx b0
b1 b2
b3 b4
...
...
```

Значения байтов принимаемой последовательности слов таковы:

- b0, b1, b2... – следующие друг за другом байты из флэш-памяти;
- xx – байты с любым значением.

Число принятых слов совпадает с числом переданных.

5.22.3.2 Последовательность загрузки через порт SPI

Загрузка через SPI выполняется в два этапа. На первом этапе SPI инициализируется с делителем частоты равным 8, и из флэш-памяти читаются 24 байта (шесть 32-х разрядных слов), начиная с адреса 0x00000000. Последнее из этих шести слов трактуется как служебное слово загрузчика (Рисунок 5.131). Проверяется содержимое служебного слова, и если оно правильное, выполняется второе чтение. В противном случае происходит аварийное прекращение загрузки.

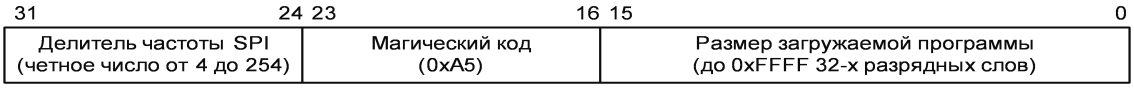


Рисунок 5.131 – Формат служебного слова загрузчика

Начальный загрузчик проверяет, что

- делитель частоты SPI - четное число в пределах от 4 до 254;
- значение поля ‘магический код’ – 0xA5;
- размер загружаемой программы больше 8 и меньше или равно 0xFFFF 32-х разрядных слов.

Перед вторым чтением SPI инициализируется с делителем частоты, равным значению соответствующего поля служебного слова. После этого из флэш-памяти читаются (N – 6) 32-х разрядных слов, начиная с адреса 0x00000018, где N – размер образа загружаемой программы, взятый из служебного слова.

5.22.4 Начальная загрузка через порт UART

Загрузка через UART выполняется в два этапа. На первом этапе выбирается скорость обмена, а на втором происходит чтение данных.

5.22.4.1 Выбор скорости обмена

Загрузчик может определить скорость интерфейса хост-компьютера, если данные передаются со скоростью 460.800, 230.400, 115.200, 57.600, 38.400, 28.800, 19.200, 14.400, 9.600 или 4.800 бод.

Хост-компьютер может работать на любой скорости из указанного ряда. Начальный загрузчик микросхемы сам подстраивается под скорость хост-компьютера. Алгоритм работы хост-компьютера:

- отправляет символ 0x00 с периодом не менее 4 мс,
- когда ответ получен, отправляет символ 0xAA,
- приступает к передаче данных.

5.22.4.2 Передача данных

Данные для загрузки через UART передаются хостом блоками произвольной длины. Каждый блок – это массив 32-разрядных слов. Блоки записываются начальным загрузчиком в

память последовательно, один за другим в порядке поступления, начиная с адреса 0xC0000000. Порядок действий хост-компьютера при передаче одного блока:

- передача пакета заголовка блока,
- прием пакета первого ответа,
- передача пакета данных блока,
- прием пакета второго ответа.

Перечисленные шаги выполняются обязательно, независимо от значения ответа. Однако, если пакет заголовка блока и/или пакет первого ответа принят с ошибкой, вместо пакета данных блока передается короткий фиктивный пакет данных с неверной контрольной суммой.

Если все четыре шага выполнены без ошибки, хост переходит к передаче следующего блока. В противном случае все четыре шага повторяются для старого блока.

Пакет заголовка блока – это массив из четырёх 32-разрядных слов.

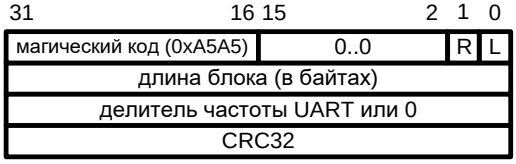


Рисунок 5.132 - Пакет заголовка блока

R – признак повторной передачи блока.

L – признак последнего блока.

Если делитель частоты UART не равен нулю, то перед отправкой ответа начальный загрузчик перестроит частоту UART на заданное значение.

Пакет данных блока – это массив 32-разрядных слов, размер которого определяется полем «длина блока» в отправленном ранее заголовке.

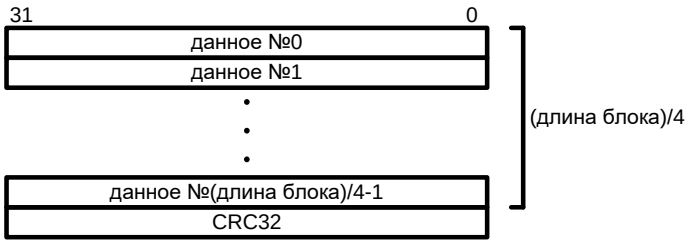


Рисунок 5.133 - Пакет данных блока

Пакет ответа – это массив из двух 32-разрядных слов.

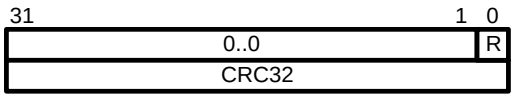


Рисунок 5.134 - Пакет ответа

R = 0 – последний пакет принят начальным загрузчиком без ошибок.

R = 1 – последний пакет принят начальным загрузчиком с ошибкой.

5.22.5 Начальная загрузка через порт Ethernet

СБИС СНП-ВП может быть загружена с помощью интерфейса Ethernet посредством протокола EDCL (Ethernet Debug Communication Link). Протокол EDCL позволяет произвольным образом записывать данные в память микросхемы.

В режиме загрузки по Ethernet загрузка происходит по следующему алгоритму:

- начальный загрузчик записывает в регистр, определяющий младшие биты (LSB) MAC-адреса (см. главу 5.20.3.4) значение 0,
- внешнее устройство по протоколу EDCL производит загрузку программы во внутреннюю память СБИС,
- внешнее устройство по протоколу EDCL записывает ненулевое значение в регистр, определяющий младшие биты (LSB) MAC-адреса,
- начальный загрузчик обнаруживает, что значение регистра изменилось, и переходит к исполнению загруженной программы.

5.22.6 Использование GPIO для индикации состояния начального загрузчика

Использование GPIO для индикации состояния начального загрузчика зависит от состояния внешних входов микросхемы TST_I2 и TST_I1. Если TST_I2 = 0, то загрузчик конфигурирует все разряды GPIO как входы и индикации состояния начального загрузчика не происходит.

Если TST_I2 = 1, то в момент старта загрузчик конфигурирует разряды GPIO [7:0] как выходы. Остальные разряды GPIO конфигурируются как входы.

Если TST_I2 = 1, то

- значение 0x00 на GPIO[7:0] говорит о том, что процесс загрузки стартовал.

Если TST_I1 = 1, то

- значение 0xC0 говорит об успешном завершении загрузки
- значение 0xCe говорит об ошибке загрузки, где e – код ошибки.

Возможные коды ошибки загрузки:

1 – ошибка определения скорости обмена по UART,
2 – «магический код» не найден в образе при загрузке по SPI,
3 – неправильный размер загружаемого образа при загрузке по SPI,
4 – неправильная контрольная сумма загружаемого образа (любой тип загрузки),
5 – неправильный размер пакет при загрузке по UART.

6 Блок аналого-цифрового преобразования (ADCU)

Блок аналогово-цифрового преобразования (**ADCU**) состоит из четырёх 10-разрядных АЦП, работающих на частоте до 90 МГц. Структурная схема блока показана ниже (см. Рисунок 6.1).

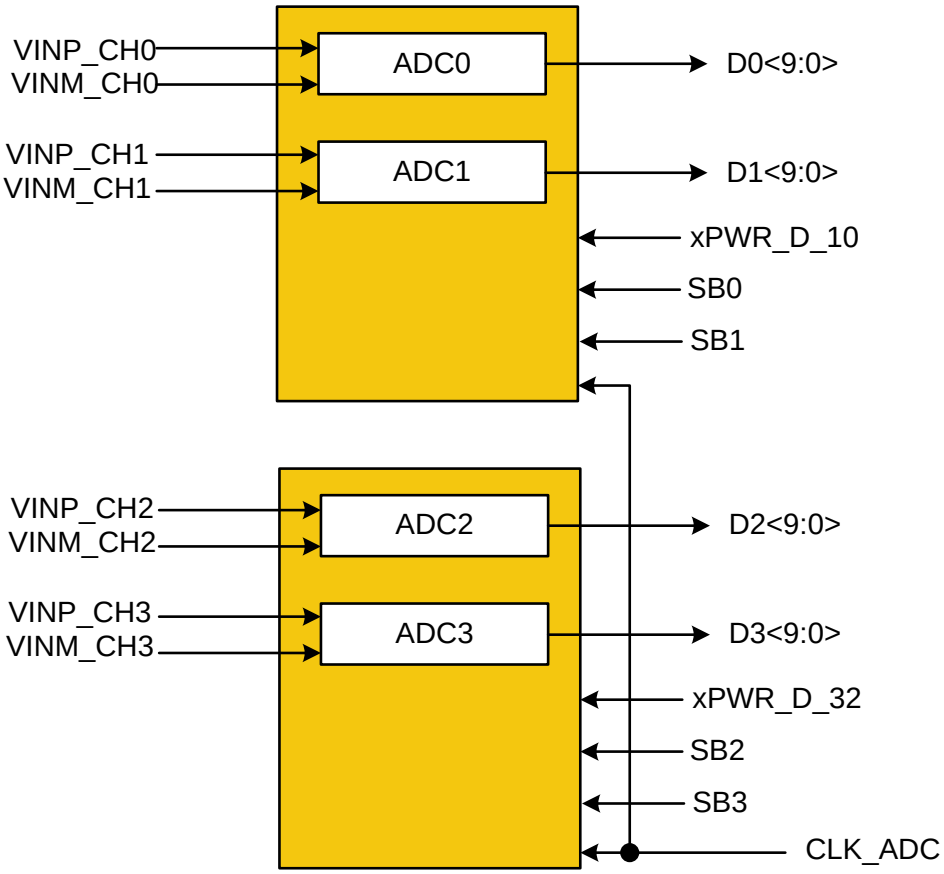


Рисунок 6.1 – Структурная схема блока ADCU

Четыре АЦП объединены в блоки попарно: ADC0, ADC1 и ADC2, ADC3. На вход каждой пары могут быть поданы независимые аналоговые сигналы или квадратурные составляющие I и Q одного сигнала. Для этого каждая пара АЦП имеет общий источник напряжения смещения и общую систему питания, что обеспечивает идентичные характеристики преобразования. Выходы АЦП подаются на DDC.

При помощи внутренних регистров управления можно переводить АЦП в режим ожидания или снять питание. Управление осуществляется процессорной системой ARMU путем программирования необходимого регистра в системном контроллере ARMSC (см. 5.11.2.7).

7 Блок аппаратных каналов преобразования частоты DDC

Блок **DDC** состоит из 12 каналов (см. Рисунок 7.1):

- 8 каналов (с 0-го по 7-й) осуществляющих обработку оцифрованных данных поступающих от АЦП или с двухбитных входов SIGNx/MAGNx. Данные каналы между собой ничем не отличаются. Канал осуществляет выбор источника входных данных, комплексное гетеродинирование с последующим децимированием (на дециматоре HC1) и нормализацией. Входные данные могут быть как комплексные (RE, IM), так и действительные. Схема накопления и упаковки (СНУ) формирует 64-х разрядные слова для записи в память. Данные упаковываются либо в 8-ми битовый, либо в 4-х битовый, либо в 2-х битовый комплексный формат. Каналы требуют настройки, которая заключается в записи управляющих регистров;
- 4 канала (с 8-го по 11-й) прямой записи в память. Данные каналы осуществляют упаковку данных с АЦП в 64-х разрядные слова для записи в память. В одном 64-х разрядном слове содержится четыре отсчета АЦП. Упаковка осуществляется путем расширения знакового разряда до 16 бит. Каждый канал осуществляет обработку соответствующего ему АЦП:
  - канал 8 производит запись в память данных с АЦП0;
  - канал 9 производит запись в память данных с АЦП1;
  - канал 10 производит запись в память данных с АЦП2;
  - канал 11 производит запись в память данных с АЦП3;

  Данным каналам требуется только настройка запуска/остановки, а так же настройка адреса куда будет осуществляться запись;
- 1 канал (12-й канал) осуществляет обработку данных из памяти. Канал выполняет функцию КИХ фильтра и требует загрузки коэффициентов фильтра.

Запуск, останов, изменение параметров каналов, исключая канал с КИХ-фильтром, осуществляется строго по рабочим временным меткам, формируемым на базе основного временного интервала (вырабатывает блок интервальных таймеров **ITU** (см. главу 9)). Рабочие временные метки делят входной поток данных с АЦП или с двухбитных входов SIGNx/MAGNx на кадры (данные между двумя соседними рабочими временными метками).

Работа блока **DDC** происходит следующим образом. На фоне обработки текущего кадра, управляющая программа осуществляет анализ и новую настройку каналов путем записи настроек в теневые регистры. После настройки всех необходимых регистров по приходу в канал временной метки, управляющая информация настройки канала переписывается из теневых регистров в рабочие (основные).

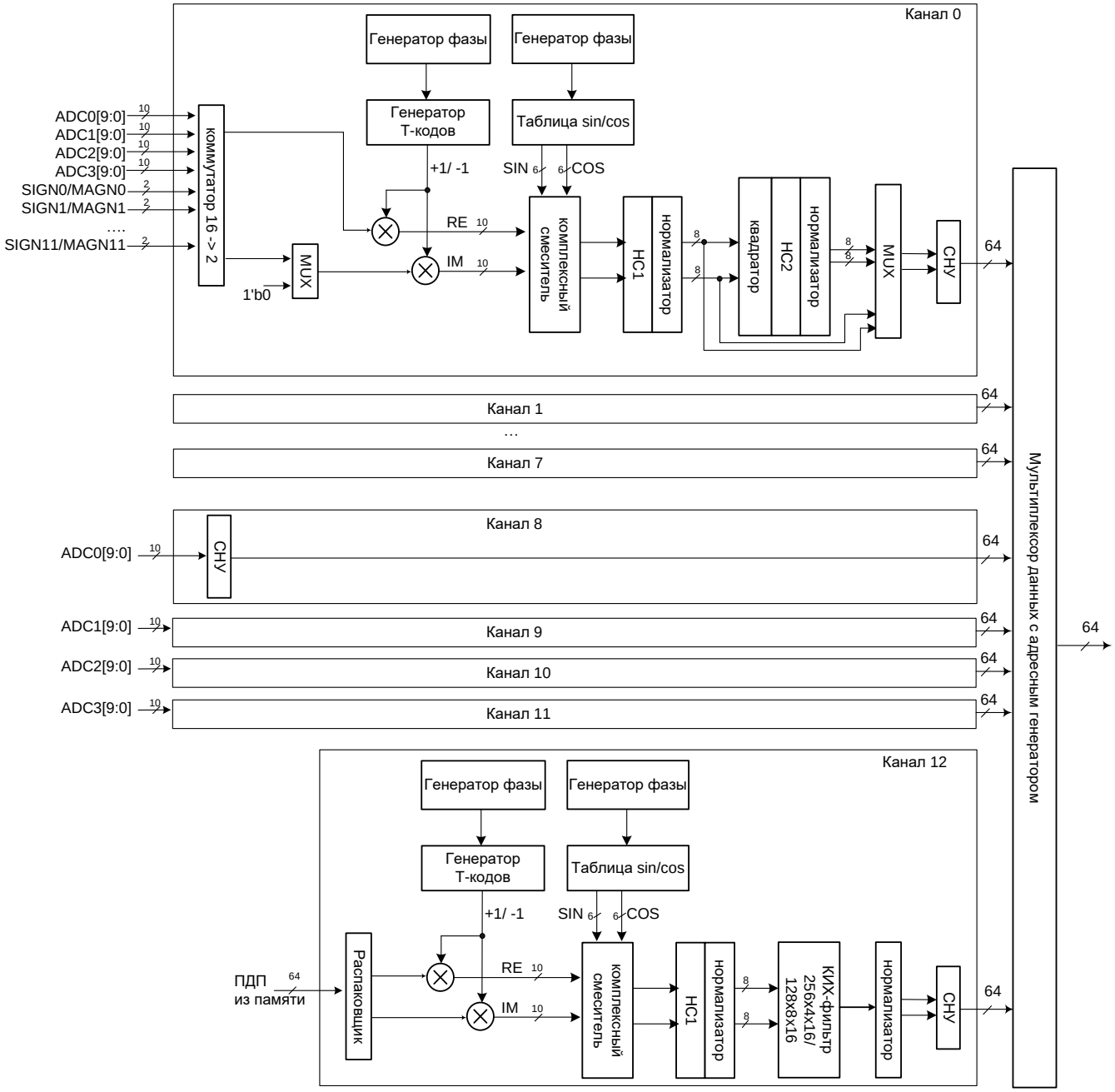


Рисунок 7.1 – Структурная схема блока DDC

7.1 Аппаратный канал, осуществляющий обработку оцифрованных данных (канал 0... канал 7)

Структурная схема канала приведена на рисунке ниже (см. Рисунок 7.2).

Канал обеспечивает обработку входных данных, поступающих с АЦП или двухбитовых входов SIGNx/MAGNx с последующей упаковкой в 64-х разрядные слова для записи в память. Канал состоит из:

- входного коммутатора;
- генератора Т-кодов (генератор псевдо случайных последовательностей);
- гетеродина;
- комплексного смесителя;

- дециматоров HC1, HC2
- нормализаторов H1, H2;
- квадратора;
- системы накопления и упаковки (СНУ);
- схемы управления каналом.

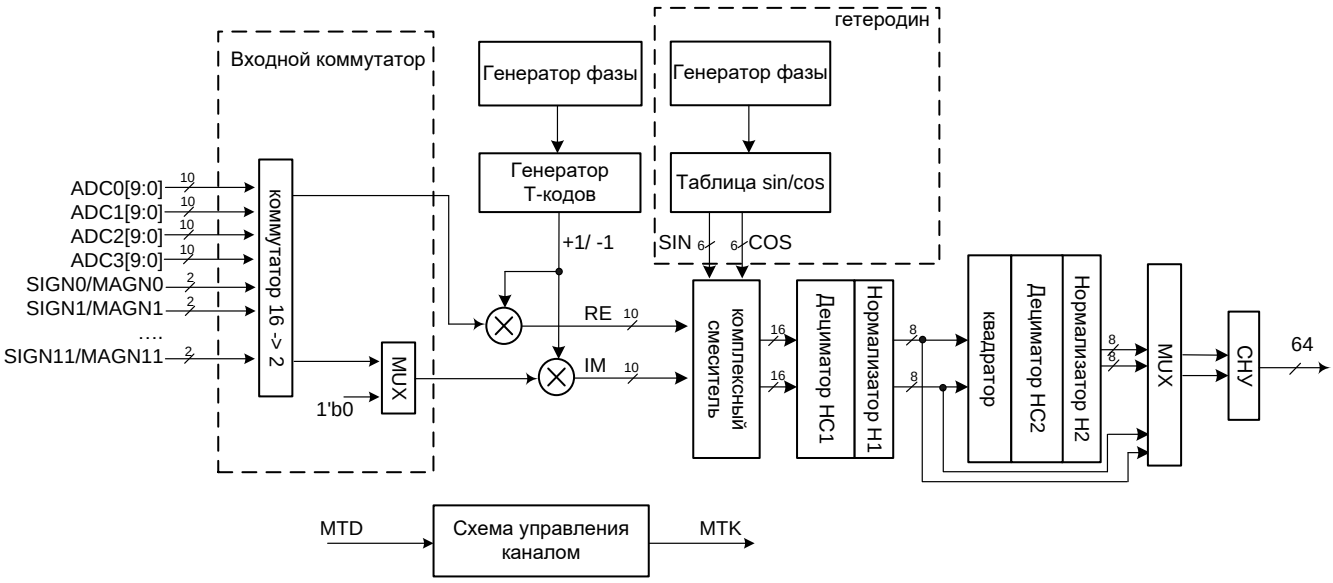


Рисунок 7.2 – Структурная схема канала осуществляющего обработку оцифрованных данных

Работа с каналом состоит из следующих действий:

- программная настройка канала;
- запуск канала;
- перестройка канала в процессе работы;
- останов канала.

Запуск, останов, изменение параметров каналов, осуществляется строго по рабочим временным меткам (**MTK**). Рабочие временные метки формируются на базе основного временного интервала (**MTD**), формируемого блоком интервальных таймеров **ITU** (см. главу 9). Программно настраивается сколько основных временных интервалов содержится между рабочими временными метками. Минимальное время между рабочими временными метками равно одному основному временному интервалу.

Для того чтобы по рабочей временной метке произошел запуск или изменение настроек канала, необходимо:

- указать в регистре **CH_CNTRL** какие регистры следует изменить;
- установить бит регистра **START** соответствующий каналу, который надо запустить или перестроить.

7.1.1 Входной коммутатор канала

Входной коммутатор обеспечивает выбор источника входных данных. Источниками входных данных могут быть данные с АЦП0,..., АЦП3 или 2-х разрядные данные с цифровых входов СБИС SIGN0/MAGN0,..., SIGN11/MAGN11. Данные могут интерпретироваться как действительные или комплексные. Если входные данные интерпретируются как комплексные, то происходит выбор двух источников: один источник для мнимой части, второй источник для действительной части комплексного данного. Управление коммутатором осуществляется регистром **CH_SELR**. Помимо этого, в данном блоке (на рисунке не указано) имеется возможность

задать для данных от аналого-цифровых преобразователей смещение на некоторую постоянную величину, которая задается в дополнительном коде. Для каждого АЦП задается свое значение (в регистре **ADCC**x). Коррекция данных производится в режиме без насыщения с увеличением разрядности результата.

7.1.2 Схема формирования синуса и косинуса (гетеродин)

Схема состоит из 32-х разрядного регистра текущей фазы FT, регистра частоты CH_FQ теневого и рабочего (теневой регистр не показан на рисунке), регистра начального смещения CH_FB, сумматора и таблицы (см. Рисунок 7.3).

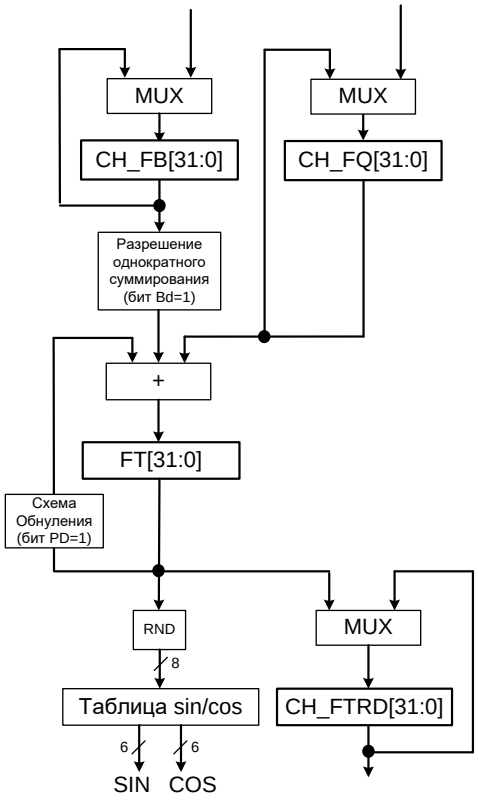


Рисунок 7.3- Структурная схема гетеродина

В каждом такте работы схемы происходит суммирование регистров FT и CH_FQ с записью результата в регистр FT.

При старте/перестройке канала возможно:

- обнуление обратной связи регистра текущей фазы FT, если установлен бит **Pd** регистра **CH_CNTRL**;
- однократное добавление регистра начального смещения **CH_FB**, если установлен бит **Bd** регистра **CH_CNTRL**. Это обеспечивает корректировку фазы при старте и в процессе работы канала.

Старшие разряды [31:23] регистра **FT** поступают в блок округления **RND** (механизм округления будет описан ниже см. главу 7.1.8). Результат после округления поступает на вход таблицы, в которой в соответствии с поступающими разрядами происходит выдача значения синуса и косинуса. Разрядность выходного синуса и косинуса 6 бит.

По приходу рабочей метки происходит запись регистра текущей фазы **FT** в регистр **CH_FTRD**. Регистр доступен **CH_FTRD** по чтению для контроля текущей фазы с момент прихода рабочей метки.

Значения **CH_FB** и **CH_FQ** задаются программно. Формула для расчета частоты синуса и косинуса и обратно приведены ниже:

$$f = fd*code * / 2^{32},$$
$$code = [f * 2^{32} /fd],$$

где f – частота гетеродина, выраженная в Герцах; code – значение регистра FQ[31:0], fd - частота оцифровки входных данных, выраженная в Герцах.

Формулы для вычисления смещения фазы гетеродина **φ**, выраженной в радианах, в зависимости от кода в регистре CH_FB[31:0] и обратно приведены ниже:

$$\varphi = code * 2*\pi / 2^{32},$$
$$code = [\varphi * 2^{32} /(2*\pi)],$$

где code – значение регистра **CH_FB**[31:0].

7.1.3 Генератор Т-кодов

Генератор Т-кодов предназначен для формирования псевдослучайной последовательности заданной длины и с заданным периодом. Для формирования используется схема на сдвиговых регистрах. Выход генератора Т-кодов осуществляет инверсию входных данных. Если выход генератора Т-кодов равен 1, то инверсия происходит, если выход генератора Т-кодов равен 0, то инверсии нет (см. Рисунок 7.2).

Генератор Т-кодов состоит из:

- генератора фазы (**TG**), который формирует управляющий сигнал (**SHIFT**) для формирования нового элемента последовательности;
- основного генератора, построенного на сдвиговых регистрах (**SRG**);
- схемы модуляции старшими разрядами генератора времени (**SM**);

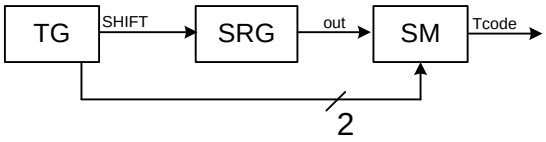


Рисунок 7.4 – Структурная схема генератора Т-кодов

Генератор фазы **TG** (см. Рисунок 7.5) состоит из регистров начального смещения (**TFB**), частоты (**TFQ**) и текущей фазы (**TFT**). В каждом такте работы происходит суммирование регистров **TFT** и **TFQ** с записью результата в регистр **TFT**. Если при суммировании возникает перенос из 31-го разряда, то формируется сигнал **SHIFT**. Также старшие два разряда регистра **TFT** поступают на схему модуляции **SM**. При старте/перестройке канала возможно:

- обнуление обратной связи регистра текущей фазы **TFT**, если установлен бит **Pt** регистра **CH_CNTRL**;
- однократное добавление регистра начального смещения **TFB**, если установлен бит **Bt** регистра **CH_CNTRL**. Это обеспечивает корректировку фазы при старте и в процессе работы канала.

Значения **TFQ** и **TFB** задаются программно. Формулы для расчета частоты смены символов ПСП приведены ниже:

$$f = fd*code * / 2^{32},$$
$$code = [f * 2^{32} /fd],$$

где f – частота смены символов ПСП, выраженная в Герцах; code – значение регистра TFQ[31:0], fd - частота оцифровки входных данных, выраженная в Герцах.

Формулы для вычисления смещения фазы внутри символа **φ**, выраженной в радианах, в зависимости от кода в регистре **TFB** и обратно приведены ниже:

$$\varphi = code * 2*\pi / 2^{32},$$

$$code = [\varphi * 2^{32} /(2*\pi)],$$

где code – значение регистра **TFB**.

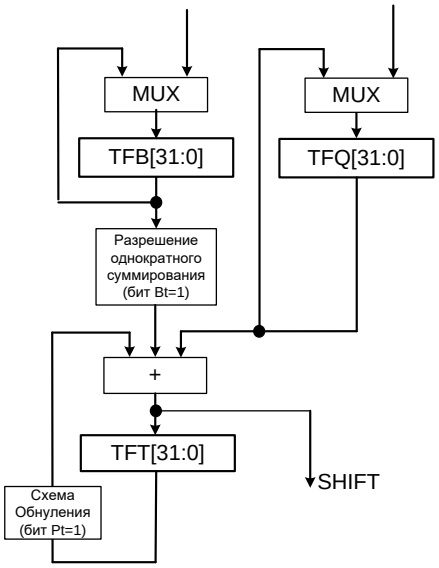


Рисунок 7.5- Структурная схема генератора фазы (TG)

Основной генератор (**SRG**) (см. Рисунок 7.6) по сигналу **SHIFT** формирует новый символ Т-кодов. Символ получается путем суммирования "по модулю 2" старших разрядов сдвиговых регистров TS0 и TS1.

По сигналу **SHIFT** от блока **TG** происходит сдвиг регистров **TS0** и **TS1** в сторону старших разрядов. Младший разряд сдвигового регистра получается сложением "по модулю 2" всех разрядов сдвигового регистра, умноженных на маску **TM**x. Маска **TM0** маскирует разряды регистра **TS0**, **TM1** - **TS1**. Помимо этого значение обратной связи может инвертироваться. Данная функция настраивается программно регистром управления **CH_ TCR**.

Также возможна работа генератора с использованием счетчика длины последовательности. В данном режиме происходит инкрементирование регистра TCGx при появлении сигнала **SHIFT**. Когда возникает сигнал переноса **carry** из старшего разряда, происходит перезапись регистра **TISx** в **TSx** (вместо сдвига) и регистра **STCGx** в **TCGx**. После этого работа продолжается.

При старте канала в сдвиговые регистры **TSx**, регистры маски **TMx**, регистры длины последовательности **STCGx**, регистр счетчика длины последовательности **TCGx** и регистры **TISx** происходит загрузка данных из теневых регистров (на схеме теневые регистры обозначены постфиксом _sh), которые должны быть инициализированы программно перед стартом канала. Запись непосредственно в рабочие регистры невозможна. При работе генератора по каждой рабочей метке происходит запись регистра TCGx в регистр TCGx_RD, который доступен для чтения. Таким образом, возможен контроль состояния регистра текущей длины в момент прихода рабочей метки.

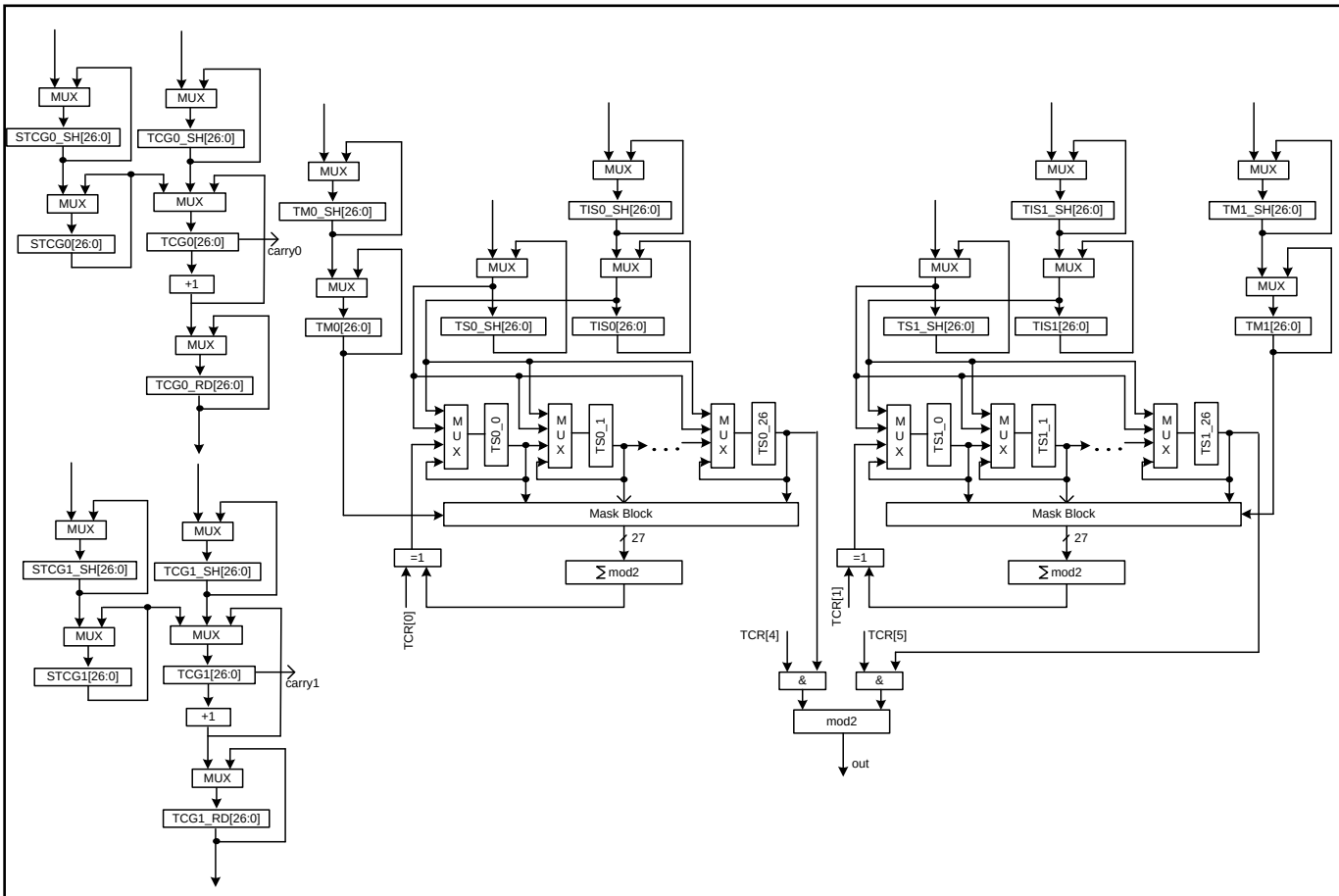


Рисунок 7.6 – Структурная схема генератора SRG

Управление режимами работы генератора осуществляется при помощи регистра **TCR**. Выход основного генератора поступает **out** на схему модуляции **SM** (см. Рисунок 7.7).

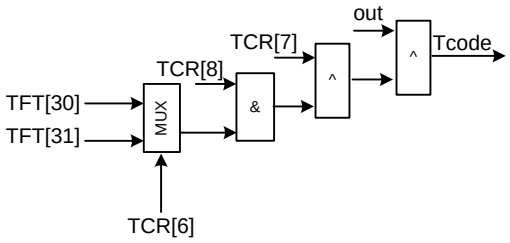


Рисунок 7.7 – Структурная схема схемы модуляции SM

Данная схема позволяет модулировать выход основоного генератора **SRG** меадром (BOC-модуляция). В качестве меандра может быть выбран 31-й или 30-й разряд регистра **TFT** генератора фазы. Управление схемой осуществляется при помощи регистра **TCR**.

7.1.4 Комплексный смеситель

Комплексный смеситель выполняет комплексное умножение входных данных с коммутатора (с учетом инверсии знака генератором Т-кодов) и значений синуса и косинуса поступающих с гетеродина.

AU_RE[16:0] = DATA_re[10:0]*cos[5:0] - DATA_im[10:0]*sin[5:0];
AU_IM[16:0] = DATA_re[10:0]*sin[5:0] + DATA_im[10:0]*cos[5:0];

где AU_RE и AU_IM - результаты арифметических операций комплексного смесителя, которые поступают на дециматор HC1; DATA - данные после инверсии знака генератором Т-кодов; sin, cos - значение синуса и косинуса, поступающее с гетеродина.

7.1.5 Дециматор HC1 и HC2

Дециматор HC1 обеспечивает уменьшение темпа входного потока данных. Блок осуществляет децимацию входных отсчетов - прореживание входных данных (выборка каждого i-того отсчета).

Децимация может работать в двух режимах:

- децимация не прекращается по границе кадра;
- децимация прекращается по границе кадра данных и начинается заново. Если закончившаяся децимация выполнилась для меньшего числа данных чем было запрограммировано, то полученный "битый" результат пропадает.

Управление блоком осуществляется программно при помощи регистра **CH_H1CNTR.**

7.1.6 Блоки нормализации H1 и H2

Блоки нормализации расположены после дециматоров, служат для уменьшения разрядности данных после арифметических блоков. Задача блоков нормализации – выбор восьми смежных разрядов аккумулятора и передача их дальше на обработку.

Выбор разрядов входных данных определяется программно с помощью поля **norm** регистров настройки **CH_H1NORM** и **CH_H2NORM**. Значению 0 соответствует выбор 1-го и 0-го разрядов аккумулятора и дополнение их шестью нулями (см. Рисунок 7.8), значению 1 – выбор трёх младших разрядов и т. д.

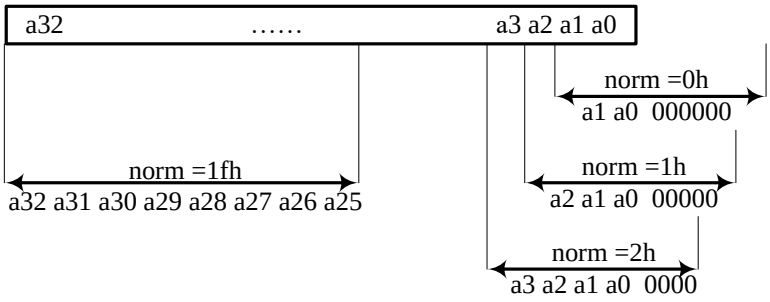


Рисунок 7.8 – Выбор разрядов при нормализации в зависимости от поля norm

Помимо этого в блоке нормализации предусмотрены дополнительные функции упрощающие работу:

- в процессе нормализации предусмотрены различные режимы округления результата (см. главу 7.1.8);
- программное включение/выключение насыщения. Настройка насыщения играет роль в случаях переполнения – ситуации, когда хотя бы один из старших отбрасываемых разрядов не равен старшему (знаковому) из выбранных восьми. Если включен режим насыщения, то при переполнении на выходе блока нормализации будет максимальное положительное (+127) или максимальное отрицательное (-128) число;
- подсчет числа переполнений в текущем кадре для каждой составляющей комплексного сигнала. После окончания обработки кадра число переполнений может быть считано программно и по полученным результатам может быть проведена коррекция настройки нормализатора;

7.1.7 Блок квадратора

Данный блок может выполнять следующие функции:

- возведение в квадрат комплексных входных данных:
  D_RE[15:0] = DATA_re[7:0]* DATA_re[7:0] - DATA _im[7:0]*DATA _im[7:0];
  D_IM[16:0] = 2*DATA_re[7:0]*DATA _im[7:0],
  где D_RE и D_IM - результаты арифметических операций квадратора, которые поступают на дециматор HC2; DATA_re, DATA_im - входные данные на входе квадратора;
- вычисление мощности:
  D_RE[15:0] = DATA_re[7:0]* DATA_re[7:0] + DATA _im[7:0]*DATA _im[7:0];
  D_IM[16:0] = *значение не определено***,**
  где D_RE и D_IM - результаты арифметических операций квадратора, которые поступают на дециматор HC2; DATA_re, DATA_im - входные данные на входе квадратора.

Выбор режима работы задается программно регистром **CH_H2CNTR**.

7.1.8 Округление результатов вычислений в каналах

Многократные операции умножения, проводимые над данными, приводят к тому, что разрядность данных растёт. Округление результатов вычислений в аппаратных каналах предварительной обработки сигналов позволяет снизить разрядность данных за счёт точности.

В канале округление производится в блоках двух типов: в блоках нормализации (см. главу 7.1.6) и в блоке фазы гетеродина (см. Рисунок 7.2, на котором данный блок обозначен как **RND**). В обоих случаях применяются одни и те же способы округления:

- к ближайшему значению – старший отбрасываемый бит прибавляется к результату;
- в большую сторону – к результату всегда прибавляется 1;
- к нулю – простое отбрасывание не используемых бит.

Выбор типа округления настраивается программно.

7.1.9 Схема накопления и упаковки СНУ

Схема накопления и упаковки (СНУ) формирует из входных комплексных 8-ми разрядных (для каждой составляющей комплексного сигнала) данных 64-х разрядные слова для записи в память, а так же адрес памяти, куда будут записаны данные. Действительные и мнимые части комплексных данных в упакованном 64-х разрядном слове располагаются в одном слове (см. Рисунок 7.9).

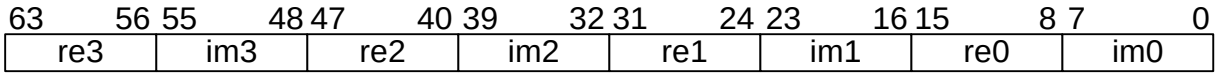


Рисунок 7.9 – Упаковка данных в 64-х разрядные слова для записи в память

Упаковка в 64-х разрядное слово может осуществляться по 8 разрядов (как на рисунке выше), по 4 разряда и по 2 разряда. При упаковке по 4 и по 2 разряда выбираются старшие биты. Настройка упаковки задается программно в регистре **CH_PACK**.

Запись упакованных данных в память осуществляется, используя 2 буфера. Т.е. сначала записываются данные в буфер0, потом в буфер1, потом опять в буфер0, затирая старые данные и т.д. Смена буфера происходит при смене кадра (приходу рабочей метки). Это позволяет обрабатывать данные в одном буфере, пока другой буфер заполняется. Если при этом буфера выделены в разных банках, то не создается конфликтов при обращении к памяти.

Начальные адреса буферов задается программно, используя регистры **CH_ADR_EV** и **CH_ADR_ODD**. Какой буфер использовать при старте канала задается полем **e/o** в регистре **CH_MSCR**.

В одном буфере данные пишутся последовательно, начиная со второго 64-х разрядного слова. В первом 64-х разрядном слове располагается заголовок (**header**), который содержит информацию о буфере. Заголовок формируется по завершающей для данного кадра рабочей метке и записывается в первое слово. Формат заголовка приведен на рисунке ниже:

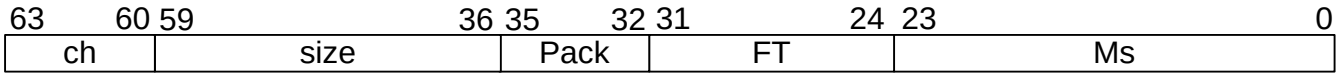


Рисунок 7.10 – Формат заголовка кадра

Поле **Ms** (разряды с 23 по 0) – значение номера основного временного интервала от интервального таймера (значение регистра **MICR** блока **ITU**);

Поле **FT** (разряды с 24 по 31) – старшие восемь разрядов регистра текущей фазы гетеродина **FT** на момент прихода рабочей метки;

Поле **Pack** (разряды с 32 по 35) – количество используемых 4-х битных фрагментов в последнем 64 разрядном слове (1 - использовано 4 бита, 2 - использовано 8 бит, ... , 31 - использовано 60 бит, 0 - использованы 64 бита). Это связано с тем, что по приходу рабочей метки происходит смена кадра и при этом часть последнего 64-х разрядного слова может быть не использована;

Поле **size** (разряды с 36 по 59) – количество 64 разрядных слов, переданных **DDC** в память для данного кадра с учетом "неполных" слов и заголовка;

Поле **ch** (разряды с 60 по 63) – номер канала (с 12 по 0), который записал данный заголовок.

!!! Если в кадре не было ни одной записи данных в память, то заголовок тоже записан не будет!!!

7.2 Аппаратный канал с КИХ-фильтром, осуществляющий обработку данных из памяти (канал 12)

Структурная схема канала приведена на рисунке ниже (см. Рисунок 7.11).

Канал обеспечивает обработку входных данных из памяти с последующей упаковкой в 64-х разрядные слова для записи в память. Канал состоит из:

- Распаковщика;
- генератора Т-кодов (генератор псевдо случайных последовательностей);
- гетеродина;
- комплексного смесителя;
- дециматор НС1;
- нормализаторов Н1, Н2;
- аппаратного КИХ-фильтра;
- системы накопления и упаковки (СНУ);

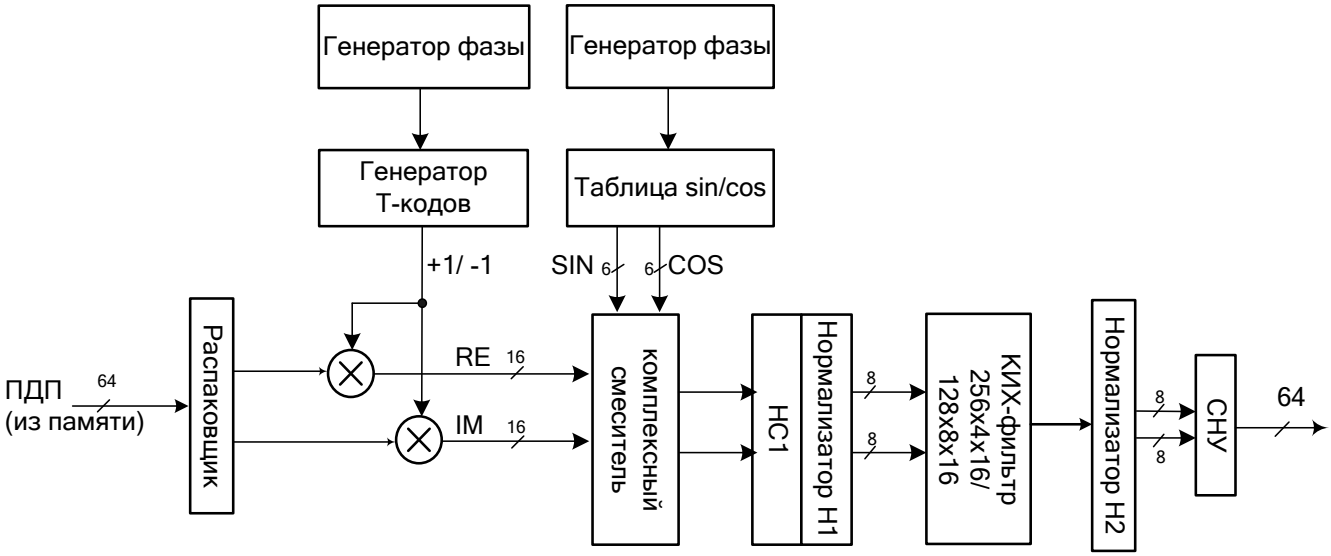


Рисунок 7.11 – Структурная схема канала с КИХ-фильтром

Работа с каналом состоит из следующих действий:

- программная настройка канала;
- настройка канала ПДП "память-DDC" (см. главу 11.2) для подкачки данных;
- запуск ПДП.

Канал воспринимает данные от одного запуска ПДП как один кадр. Результат обработки записывается в память, включая заголовок кадра.

7.2.1 Распаковщик

Данный блок обеспечивает выделение из входного 64-х разрядного слова данных для обработки в канале. Поддерживаются следующие форматы входных данных (см. Рисунок 7.12):

- комплексные 2-х битные данные;
- комплексные 4-х битные данные;
- комплексные 8-х битные данные;
- действительные 11-ти битные данные.

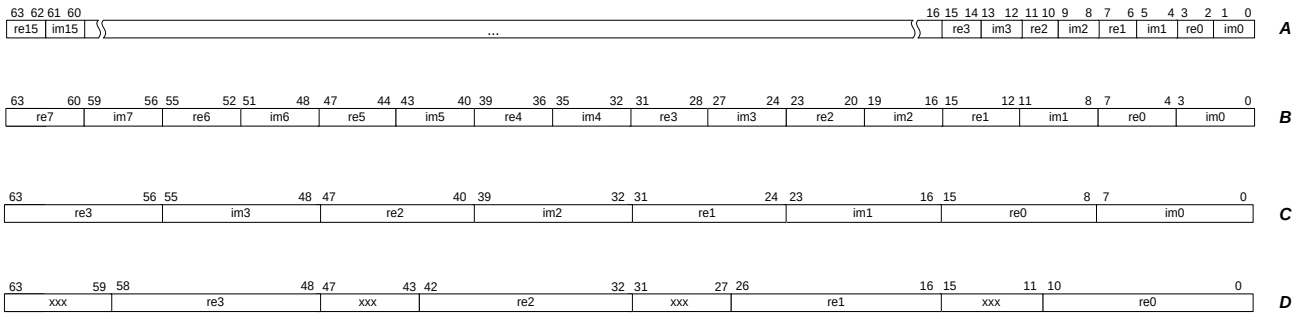


Рисунок 7.12 – Поддерживаемые распаковщиком форматы входного 64-х разрядного слова: A - комплексные 2-х битные данные; B - комплексные 4-х битные данные; C - комплексные 8-битные данные D - действительные 11-ти битные данные;

Все описанные форматы идентичны формату упаковки схемой накопления и упаковки аппаратных каналов, осуществляющих обработку оцифрованных данных (канал 0,..., канал 7), поэтому записанные в память данные каналами обработки могут быть сразу отправлены на обработку в данный канал (*необходимо только исключить первое слово заголовка*).

Выбор формата входных данных осуществляется регистром **F_PACK**.

7.2.2 Схема формирования синуса и косинуса (гетеродин)

Схема состоит из 32-х разрядного регистра текущей фазы **F_FT**, регистра частоты **F_FQ**, сумматора и таблицы (см. Рисунок 7.13).

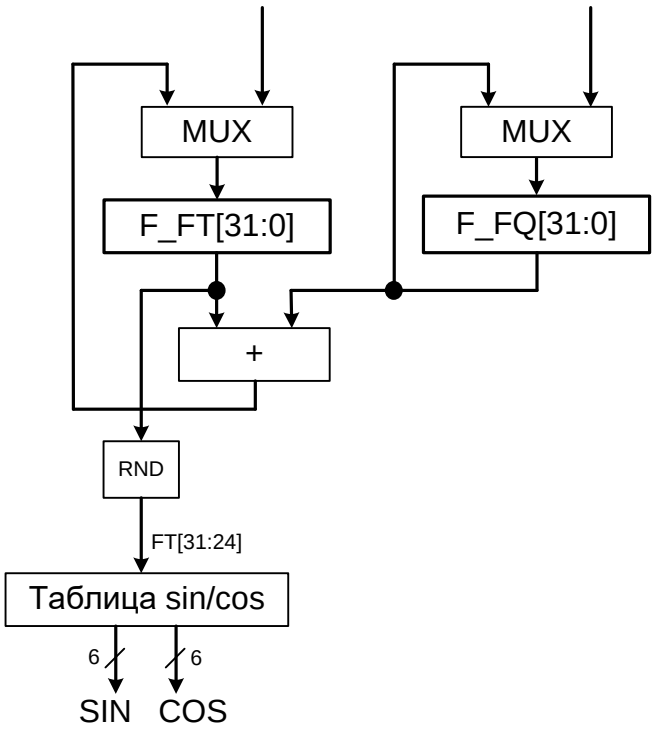


Рисунок 7.13 - Структурная схема гетеродина

В каждом такте работы схемы происходит суммирование регистров **F_FT** и **F_FQ** с записью результата в регистр F_FT.

Старшие разряды [31:22] регистра **F_FT** поступают в блок округления **RND** (механизм округления описан выше см. главу 7.1.8). Результат после округления поступает на вход таблицы, в которой в соответствии с поступающими разрядами происходит выдача значения синуса и косинуса. Разрядность выходного синуса и косинуса 6 бит.

От схемы гетеродина в канале0 - канале7 данная схема отличается:

- отсутствием теневых регистров;

- отсутствием регистра начального смещения **FB**, в данном варианте доступна запись непосредственно в регистр **F_FT**;

Управление блоком осуществляется при помощи регистров **F_SELR**, **F_FT** и **F_FQ**, значение которых задается программно.

7.2.3 Генератор Т-кодов

Функции и устройство данного генератора Т-кодов идентичны схеме генератора Т-кодов аппаратных каналов, осуществляющих обработку оцифрованных данных (канал 0,..., канал 7) (см. главу 7.1.3) кроме генератора фазы (см. Рисунок 7.14).

От схемы гетератора фазы в канале 0 - канале 7 данная схема отличается:

- отсутствием теневых регистров;

- отсутствием регистра начального смещения **CH_TFB**, в данном варианте доступна запись непосредственно в регистр **F_TFT**;

Управление генератором Т-кодов осуществляется при помощи регистров **F_TFT**, **F_TFQ**, **F_STCG0**, **F_STCG1**, **F_TCG0**, **F_TCG1**, **F_TM0**, **F_TM1**, **F_TS0**, **F_TS1**, **F_TIS0** и **F_TIS1**, значение которых задается программно.

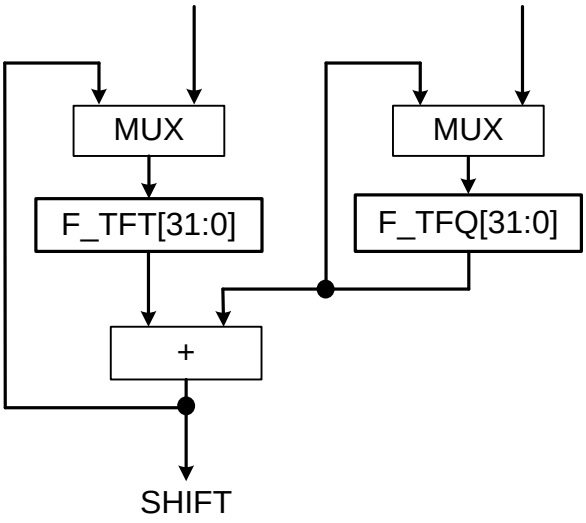


Рисунок 7.14 - Структурная схема генератора фазы Т-кодов

7.2.4 Комплексный смеситель

Комплексный смеситель выполняет комплексное умножение входных данных с коммутатора (с учетом инверсии знака генератором Т-кодов) и значений синуса и косинуса, поступающих с гетеродина. Смеситель идентичен смесителю в каналах, осуществляющих обработку оцифрованных данных (канал 0,..., канал 7) (см. главу 7.1.3).

7.2.5 Блок дециматора НС1

Блок НС1 идентичен блоку НС1 в каналах, осуществляющих обработку оцифрованных данных (канал 0,..., канал 7) (см. главу 7.1.5). Отличается только тем, что децимация всегда прекращается по границе кадра. В данном случае границей кадра являются последние данные. Управление НС1 осуществляется при помощи регистра **F_H1CNTR.**

7.2.6 Блоки нормализации Н1 и Н2

Нормализаторы идентичны нормализаторам в каналах, осуществляющих обработку оцифрованных данных (канал 0,..., канал 7) (см. главу 7.1.6). Управление нормализаторами осуществляется при помощи регистров **F_H1NORM** (нормализатор Н1) **F_H2NORM** (нормализатор Н2).

7.2.7 Блок КИХ-фильтра

Блок КИХ-фильтра состоит их сдвигового регистра данных D_0 - $D_{N-1,}$ регистров весов W_0 - $W_{N-1,}$ комплексных умножителей и сумматоров (см. Рисунок 7.15).

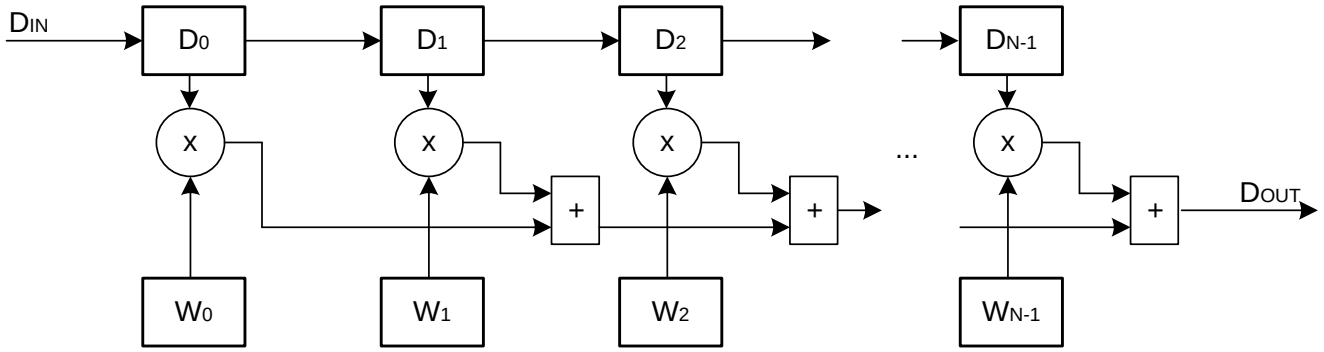


Рисунок 7.15 - Структурная схема блока КИХ-фильтра

Блок КИХ-фильтра выполняет следующее арифметическое действие:

$$Dout = \sum_{i=0}^{N-1} W_i * D_i \text{ , где}$$

Di - входные комплексные данные в блок КИХ-фильра (поступают с нормализатора Н1) разрядностью 8 или 4 бита для каждой составляющей комплексного сигнала; Wi - комплексные веса КИХ -фильтра разрядностью 16 бит для каждой составляющей комплексного сигнала; Dout - комплексный результат фильтрации (поступает в блок нормализатора Н2). N- порядок фильтра (128 или 256). Возможны два варианта фильтра:

- разрядность входных данных Di - 4 бита для каждой составляющей комплексного сигнала, разрядность весов 16 бит для каждой составляющей комплексного сигнала, порядок фильтра - 256;
- разрядность входных данных Di - 8 бит для каждой составляющей комплексного сигнала, разрядность весов 16 бит для каждой составляющей комплексного сигнала, порядок фильтра - 128;

Выбор конфигурации осуществляется программно с помощью регистра **F_PACK**. Веса задаются программно при помощи регистров **F_Wx**.

7.2.8 Схема накопления и упаковки СНУ

Схема накопления и упаковки (СНУ) формирует из входных комплексных 8-ми разрядных (для каждой составляющей комплексного сигнала) данных 64-х разрядные слова для записи в память, а так же адрес памяти, куда будут записаны данные. Упаковка данных и формат заголовка кадра такие же как и в каналах, осуществляющих обработку оцифрованных данных (канал 7,..., канал 0) (см. главу 7.1.9).

Запись упакованных данных в память осуществляется, используя только один буфер, адрес которого задается в регистре **F_ADR**. Это является отличием данной схемы от схемы СНУ в каналах, осуществляющих обработку оцифрованных данных (канал 0,..., канал 7).

7.3 Прерывания блока DDC

Блок формирует два прерывания (**INT_CH_DDC**, **INT_PRG_DDC**) процессорным системам **NMU** и **ARMU**.

Прерывание **INT_CH_DDC** формируется при ошибках в процессе работы каналов:

- неправильное программирование генратора Т-кодов;
- потеря данных.

Потеря данных возникает в том случае, когда пропускная способность необходимая для записи данных была не достигнута. Например, данная ситация возможна при одновременном обращении к одному банку памяти каналов DDC и процессорных ядер. В данном примере может произойти потеря данных, записываемых каналом.

Неправильное программирование генератора Т-кодов может возникнуть при однократном добавлении смещения (**TFB**) в генераторе времени. Так как смещение воспринимается как число со знаком, то в результате добавления может возникнуть ситуация, когда надо использовать предыдущее значение генератора Т-кодов. Вычисление предыдущего значение невозможно. Поэтому формируется признак ошибки о неправильном программировании генератора Т-кодов.

Прерывание **INT_PRG_DDC** формируется при успешном завершении записи кадра в память каналом.

Информация о канале, которым было вызвано прерывание определяется регистрами **HWM** и **INTR**. Дополнительную информацию о записи кадра содержит регистр статуса **HWM**:

- запись кадра в область памяти, определяемую регистром **CH_ADR_EV**, прошла успешно (формируется прерывание **INT_PRG_DDC**);
- запись кадра в область памяти, определяемую регистром **CH_ADR_ODD**, прошла успешно (формируется прерывание **INT_PRG_DDC**);
- при записи в память произошла ошибка (потеря данных) (формируется прерывание **INT_CH_DDC)**.

Если возникла ошибка, то для возобновления работы канала необходимо сбросить соответствующие биты регистров **HWM** и **INTR**. Сброс осуществляется автоматически при чтении регистров с учетом разрешения сброса (регистр **IE**).

7.4 Конфигурационные регистры блока DDC

В таблицах ниже (Таблица 7.1- Таблица 7.4) приведены младшие адреса регистров блока DDC в адресном пространстве NMU. Старшие разряды адресы всегда равны ***000Ch*** (т.е. адрес имеет вид ***000C_xxxxh***). Адресация в процессорной системе ARMU получается умножением адреса NMU на 4.

Таблица 7.1 – младшие разряды адреса регистров общего назначения в адресном пространстве NMU

название регистра	тип доступа*	адрес, hex
START	wr	070F
STOP	wr	070E
ADCC0	wr	0700
ADCC1	wr	0701
ADCC2	wr	0702
ADCC3	wr	0703
PRST	wr	070D
HWM	rd	0681
HWM	rd/reset	0680
IE	wr/rd	0683
INTR	rd/reset	0682

*wr - регистр доступен по записи;
rd - регистр доступен по чтению.

Таблица 7.2 – младшие разряды адреса регистров управления каналами с 0 по 7 в адресном пространстве NMU

название регистра	тип доступа	адрес регистра в канале 0, hex	адрес регистра в канале 1, hex	адрес регистра в канале 2, hex	адрес регистра в канале 3, hex	адрес регистра в канале 4, hex	адрес регистра в канале 5, hex	адрес регистра в канале 6, hex	адрес регистра в канале 7, hex
CH_MSCR	rd/wr	000D	002D	004D	006D	008D	00AD	00CD	00ED
CH_CNTRL	rd/wr	000F	002F	004F	006F	008F	00AF	00CF	00EF
CH_SELR	rd/wr	000C	002C	004C	006C	008C	00AC	00CC	00EC
CH_FB	rd/wr	0000	0020	0040	0060	0080	00A0	00C0	00E0
CH_FQ	rd/wr	0001	0021	0041	0061	0081	00A1	00C1	00E1
CH_FTRD	rd	0002	0022	0042	0062	0082	00A2	00C2	00E2
CH_H1CNTR	rd/wr	0008	0028	0048	0068	0088	00A8	00C8	00E8
CH_H2CNTR	rd/wr	0009	0029	0049	0069	0089	00A9	00C9	00E9
CH_H1NORM	rd/wr	000A	002A	004A	006A	008A	00AA	00CA	00EA
CH_H2NORM	rd/wr	000B	002B	004B	006B	008B	00AB	00CB	00EB
CH_PACK	rd/wr	000E	002E	004E	006E	008E	00AE	00CE	00EE
CH_ADR_EV	wr	0400	0402	0404	0406	0408	040A	040C	040E
CH_ADR_ODD	wr	0401	0403	0405	0407	0409	040B	040D	040F
CH_TFB	rd/wr	0004	0024	0044	0064	0084	00A4	00C4	00E4
CH_TFQ	rd/wr	0005	0025	0045	0065	0085	00A5	00C5	00E5
CH_TFTRD	rd	0006	0026	0046	0066	0086	00A6	00C6	00E6
CH_STCG0	rd/wr	0010	0030	0050	0070	0090	00B0	00D0	00F0
CH_STCG1	rd/wr	0018	0038	0058	0078	0098	00B8	00D8	00F8
CH_TCG0	rd/wr	0011	0031	0051	0071	0091	00B1	00D1	00F1
CH_TCG1	rd/wr	0019	0039	0059	0079	0099	00B9	00D9	00F9
CH_TM0	rd/wr	0014	0034	0054	0074	0094	00B4	00D4	00F4
CH_TM1	rd/wr	001C	003C	005C	007C	009C	00BC	00DC	00FC
CH_TS0	rd/wr	0013	0033	0053	0073	0093	00B3	00D3	00F3
CH_TS1	rd/wr	001B	003B	005B	007B	009B	00BB	00DB	00FB
CH_TIS0	rd/wr	0012	0032	0052	0072	0092	00B2	00D2	00F2
CH_TIS1	rd/wr	001A	003A	005A	007A	009A	00BA	00DA	00FA

Таблица 7.3 – младшие разряды адреса регистров управления каналами с 8 по 11 в адресном пространстве NMU

название регистра	тип доступа	адрес регистра в канале 8, hex	адрес регистра в канале 9, hex	адрес регистра в канале 10, hex	адрес регистра в канале 11, hex
CH_ADR_EV	wr	0410	0412	0414	0416
CH_ADR_ODD	wr	0411	0413	0415	0417

Таблица 7.4 – младшие разряды адреса регистров управления канала с КИХ-фильтром (канал 12) в адресном пространстве NMU

название регистра	тип доступа	адрес регистра в канале 12, hex
F_FQ	rd/wr	0601
F_FT	rd/wr	0600
F_PACK	rd/wr	060E
F_SELR	rd/wr	060C
F_H1CNTR	rd/wr	0608
F_H1NORM	rd/wr	060A
F_H2NORM	rd/wr	060B
F_ADR	wr	0419
F_TFT	rd/wr	0604
F_TFQ	rd/wr	0605
F_STCG0	rd/wr	0610
F_STCG1	rd/wr	0618
F_TCG0	rd/wr	0611
F_TCG1	rd/wr	0619
F_TM0	rd/wr	0614
F_TM1	rd/wr	061C
F_TS0	rd/wr	0613
F_TS1	rd/wr	061B
F_TIS0	rd/wr	0612
F_TIS1	rd/wr	061A
F_W0	wr	0800
F_W1	wr	0801
F_W2	wr	0802
F_W3	wr	0803
....		
F_W253	wr	08fd
F_W254	wr	08fe
F_W255	wr	08ff

Далее приводится подробное описание всех конфигурационных регистров блока DDC.

7.4.1 Регистры общего назначения

7.4.1.1 Регистр запуска каналов START

Данный регистр управляет запуском и разрешением перестройки каналов с номерами 0..11. Формат регистра приведен на рисунке (см Рисунок 7.16), значения полей приведены в таблице ниже (Таблица 7.5).

11	10	9	8	7	6	5	4	3	2	1	0
c11	c10	c9	c8	c7	c6	c5	c4	c3	c2	c1	c0

Рисунок 7.16 - Формат регистра START

Таблица 7.5 – Назначение полей регистра START

Биты	Название	Выполняемая функция
[31:12]	Reserved	Зарезервированы.
[11]	c11	1b – канал 11 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 11 после прихода метки не будет
[10]	c10	1b – канал 10 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 10 после прихода метки не будет
[9]	c9	1b – канал 9 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 9 после прихода метки не будет
[8]	c8	1b – канал 8 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 8 после прихода метки не будет
[7]	c7	1b – канал 7 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 7 после прихода метки не будет
[6]	c6	1b – канал 6 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 6 после прихода метки не будет
[5]	c5	1b – канал 5 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 5 после прихода метки не будет
[4]	c4	1b – канал 4 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 4 после прихода метки не будет
[3]	c3	1b – канал 3 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 3 после прихода метки не будет
[2]	c2	1b – канал 2 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 2 после прихода метки не будет
[1]	c1	1b – канал 1 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 1 после прихода метки не будет
[0]	c0	1b – канал 0 будет запущен или разрешена перестройка уже работающего канала после прихода метки 0b – команды запуска для канала 0 после прихода метки не будет

Регистр доступен для побитной установки. Сброс происходит автоматически по приходу рабочей метки.

Установленный бит разрешает запускать канал в момент прихода рабочей метки. Если канал уже работает, то установленный бит разрешает перестройку конфигурационных регистров, определяемых регистром ***CH_CNTRL*** (*некоторые регистры могут быть перестроины без установки бита в регистре START (см. описание регистра CH_CNTRL*).

Существует ограничение – нельзя после записи регистра **START** производить настройку канала, после записи регистра **START** должна идти рабочая метка.

7.4.1.2 Регистр остановки каналов STOP

11	10	9	8	7	6	5	4	3	2	1	0
c11	c10	c9	c8	c7	c6	c5	c4	c3	c2	c1	c0

Рисунок 7.17 - Формат регистра STOP

Данный регистр управляет остановкой каналов с номерами 0..11. Формат регистра приведен на рисунке (см. Рисунок 7.17), значения полей приведены в таблице ниже (Таблица 7.6).

Таблица 7.6 – Назначение полей регистра STOP

Биты	Название	Выполняемая функция
[31:12]	Reserved	Зарезервированы.
[11]	c11	1b – будет остановлен канал 11 после прихода метки 0b – команды остановки для канала 11 после прихода метки не будет
[10]	c10	1b – будет остановлен канал 10 после прихода метки 0b – команды остановки для канала 10 после прихода метки не будет
[9]	c9	1b – будет остановлен канал 9 после прихода метки 0b – команды остановки для канала 9 после прихода метки не будет
[8]	c8	1b – будет остановлен канал 8 после прихода метки 0b – команды остановки для канала 8 после прихода метки не будет
[7]	c7	1b – будет остановлен канал 7 после прихода метки 0b – команды остановки для канала 7 после прихода метки не будет
[6]	c6	1b – будет остановлен канал 6 после прихода метки 0b – команды остановки для канала 6 после прихода метки не будет
[5]	c5	1b – будет остановлен канал 5 после прихода метки 0b – команды остановки для канала 5 после прихода метки не будет
[4]	c4	1b – будет остановлен канал 4 после прихода метки 0b – команды остановки для канала 4 после прихода метки не будет
[3]	c3	1b – будет остановлен канал 3 после прихода метки 0b – команды остановки для канала 3 после прихода метки не будет
[2]	c2	1b – будет остановлен канал 2 после прихода метки 0b – команды остановки для канала 2 после прихода метки не будет
[1]	c1	1b – будет остановлен канал 1 после прихода метки 0b – команды остановки для канала 1 после прихода метки не будет
[0]	c0	1b – будет остановлен канал 0 после прихода метки 0b – команды остановки для канала 0 после прихода метки не будет

Регистр доступен для побитной установки. Сброс происходит автоматически по приходу рабочей метки.

Установленный бит останавливает соответствующий канал по первой рабочей метке.

7.4.1.3 Регистры управления калибровкой АЦП ADCC0,..., ADCC3

Регистр ADCC0 задает величину смещения, которое добавляется к данным, поступаемым с входа АЦП0. Регистры ADCC1, ADCC2, ADCC3 имеют ту же функцию, но применительно к входам ADC1, ADC2, ADC3 соответственно. Формат регистров приведен на рисунке (см Рисунок 7.18), значения полей приведены в таблице ниже (Таблица 7.7).

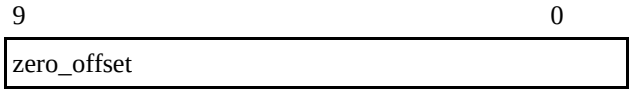


Рисунок 7.18 - Формат регистра ADCCx

Таблица 7.7 – Назначение полей регистра ADCCx

Биты	Название	Выполняемая функция
[31:10]	Reserved	Зарезервированы.
[9:0]	zero_offset	значение смещения в дополнительном коде.

Все регистры доступны только по записи.

7.4.1.4 Регистр программного сброса блока DDC PRST

Запись любых данных в регистр **PRST** осуществляет сброс блока **DDC**: каналы остановлены, все регистры обнулены. Состояние идентично состоянию после включения питания. Регистр доступен по записи.

7.4.1.5 Регистр разрешения сброса регистров статуса IE

Регистр **IE** определяет разряды регситров **INTR** и **HWM**, которые будут сброшены автоматически при чтении. Формат регистров приведен на рисунке ниже (см. Рисунок 7.19), значения полей приведены в таблице ниже (Таблица 7.8).

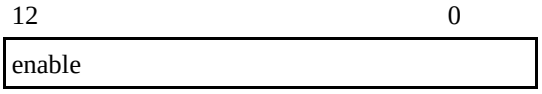


Рисунок 7.19 - Формат регистра IE

Таблица 7.8 – Назначение полей регистра IE

Биты	Название	Выполняемая функция
[31:13]	reserved	Зарезервированы.
[12]	enable_CH12	1 - автоматический сброс разрядов для канала 12 разрешен; 0 - автоматический сброс разрядов для канала 12 запрещен;
[11]	enable_CH11	1 - автоматический сброс разрядов для канала 11 разрешен; 0 - автоматический сброс разрядов для канала 11 запрещен;
[10]	enable_CH10	1 - автоматический сброс разрядов для канала 10 разрешен; 0 - автоматический сброс разрядов для канала 10 запрещен;
[9]	enable_CH9	1 - автоматический сброс разрядов для канала 9 разрешен; 0 - автоматический сброс разрядов для канала 9 запрещен;
[8]	enable_CH8	1 - автоматический сброс разрядов для канала 8 разрешен; 0 - автоматический сброс разрядов для канала 8 запрещен;
[7]	enable_CH7	1 - автоматический сброс разрядов для канала 7 разрешен; 0 - автоматический сброс разрядов для канала 7 запрещен;
[6]	enable_CH6	1 - автоматический сброс разрядов для канала 6 разрешен; 0 - автоматический сброс разрядов для канала 6 запрещен;
[5]	enable_CH5	1 - автоматический сброс разрядов для канала 5 разрешен; 0 - автоматический сброс разрядов для канала 5 запрещен;
[4]	enable_CH4	1 - автоматический сброс разрядов для канала 4 разрешен; 0 - автоматический сброс разрядов для канала 4 запрещен;
[3]	enable_CH3	1 - автоматический сброс разрядов для канала 3 разрешен; 0 - автоматический сброс разрядов для канала 3 запрещен;
[2]	enable_CH2	1 - автоматический сброс разрядов для канала 2 разрешен; 0 - автоматический сброс разрядов для канала 2 запрещен;
[1]	enable_CH1	1 - автоматический сброс разрядов для канала 1 разрешен; 0 - автоматический сброс разрядов для канала 1 запрещен;
[0]	enable_CH0	1 - автоматический сброс разрядов для канала 0 разрешен; 0 - автоматический сброс разрядов для канала 0 запрещен;

Регистр доступен по чтению и записи.

7.4.1.6 Регистр прерываний INTR

Регистр **INTR** содержит информацию о сформированных прерываниях в каналах 0-7, возникших при неправильном программировании генератора Т-кодов.

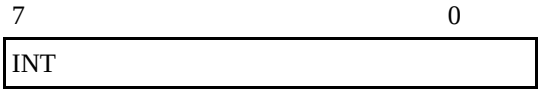


Рисунок 7.20 - Формат регистра INTR

Таблица 7.9 – Назначение полей регистра INTR

Биты	Название	Выполняемая функция
[31:8]	reserved	Зарезервированы.
[7]	INT _CH7	0 - нет ошибки программирования генератора Т-кодов в канале 7; 1 - есть ошибка программирования генератора Т-кодов в канале 7;
[6]	INT _CH6	0 - нет ошибки программирования генератора Т-кодов в канале 6; 1 - есть ошибка программирования генератора Т-кодов в канале 6;
[5]	INT _CH5	0 - нет ошибки программирования генератора Т-кодов в канале 5; 1 - есть ошибка программирования генератора Т-кодов в канале 5;
[4]	INT _CH4	0 - нет ошибки программирования генератора Т-кодов в канале 4; 1 - есть ошибка программирования генератора Т-кодов в канале 4;
[3]	INT _CH3	0 - нет ошибки программирования генератора Т-кодов в канале 3; 1 - есть ошибка программирования генератора Т-кодов в канале 3;
[2]	INT _CH2	0 - нет ошибки программирования генератора Т-кодов в канале 2; 1 - есть ошибка программирования генератора Т-кодов в канале 2;
[1]	INT _CH1	0 - нет ошибки программирования генератора Т-кодов в канале 1; 1 - есть ошибка программирования генератора Т-кодов в канале 1;
[0]	INT _CH0	0 - нет ошибки программирования генератора Т-кодов в канале 0; 1 - есть ошибка программирования генератора Т-кодов в канале 0;

Регистр доступен по чтению. После чтения происходит автоматический сброс всех разрядов, сброс которых разрешен регистром **IE**.

7.4.1.7 Регистр статуса каналов HWM

Регистр **HWM** содержит информацию о работе каналов блока **DDC**.

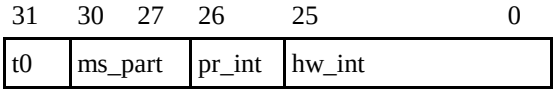


Рисунок 7.21 - Формат данных регистра HWM

Таблица 7.10 – Назначение полей регистра HWM

Биты	Название	Выполняемая функция
[31]	t0	младший разряд регистра **MICR** блока **ITU** на момент чтения регистра.
[30:27]	ms_part	часть основного временного интервала блока **ITU** на момент чтения регистра 0000b - идет первая часть 1/16 основного временного интервала **Tmtd** была на момент чтения регистра; 0001b - идет вторая часть 1/16 основного временного интервала **Tmtd** была на момент чтения регистра; .... 1111b - идет шестнадцатая (последняя) часть 1/16 основного временного интервала **Tmtd** была на момент чтения регистра;
[26]	pr_int	1 - в регистре **INTR** есть хотя бы одна единица; 0 - в регистре **INTR** нет единиц.
[25:24]	hw_int_12	статус канала 12
[23:22]	hw_int_11	статус канала 11
[21:20]	hw_int_10	статус канала 10
[19:18]	hw_int_9	статус канала 9
[17:16]	hw_int_8	статус канала 8
[15:14]	hw_int_7	статус канала 7
[13:12]	hw_int_6	статус канала 6
[11:10]	hw_int_5	статус канала 5
[9:8]	hw_int_4	статус канала 4
[7:6]	hw_int_3	статус канала 3
[5:4]	hw_int_2	статус канала 2
[3:2]	hw_int_1	статус канала 1
[1:0]	hw_int_0	статус канала 0

Статус канала может принимать следующие значения:

- 00b — новой информации не было;
- 01b — запись кадра в область памяти, определяемую регистром **CH_ADR_EV**, прошла успешно (формируется прерывание **INT_PRG_DDC**);
- 10b — запись кадра в область памяти, определяемую регистром **CH_ADR_ODD**, прошла успешно (формируется прерывание **INT_PRG_DDC**);
- 11b — при записи в память произошла потеря данных.

При потере данных необходимо сбросить при помощи чтения соответствующие разряды поля hw_int (источник прерывания)

Регистр доступен по чтению и по чтению со сбросом. При чтении со сбросом после чтения происходит автоматический сброс разрядов поля **hw_int**, сброс которых разрешен регистром **IE**.

7.4.2 Регистры управления каналами, осуществляющими обработку оцифрованных данных (канал 0, ..., канал 7)

7.4.2.1 Регистр программирования рабочей метки CH_MSCR

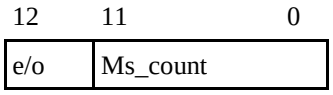


Рисунок 7.22 - Формат регистра CH_MSCR

Данный регистр управляет формированием рабочих меток. Формат регистра приведен на рисунке(см. Рисунок 7.22), значения полей приведены в таблице ниже (Таблица 7.11).

Таблица 7.11 – Назначение полей регистра CH_MSCR

Биты	Название	Выполняемая функция
[31:13]	Reserved	Зарезервированы.
[12]	e/o	Бит e/o задает буфер в памяти, в который будет производиться запись первого кадра после рабочей прихода метки: 0 – начальный адрес буфера определяется регистром **CH_ADR_ODD**; 1 – начальный адрес буфера определяется регистром **CH_ADR_EV**;
[11:0]	Ms_count	Период рабочей метки 0h - Tmtk = Tmtd; 1h - Tmtk = 2Tmtd**;** 2h - Tmtk = 3Tmtd**;** ... FFFh - Tmtk = 4096Tmtd**;** Tmtk - период рабочей метки; Tmtd - период основного временного интервала, формируемый блоком **ITU**

Регистр доступен по чтению и записи. Запись осуществляется в теневой регистр, который будет переписан в рабочий по первой рабочей метке при наличии бита **Ms** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре START. Чтение осуществляется из рабочего регистра.

Рабочая метка задается для каждого канала осуществляющего обработку оцифрованных данных поступающих от АЦП или с двухбитных входов SIGNx/MAGNx, (канал 7,..., канал 0).

Бит e/o при записи определяет в какую область памяти будут записываться данные после запуска канала, при чтении определяет область памяти куда будет производиться запись данных в следующем кадре.

7.4.2.2 Регистр управления CH_CNTRL

Данный регистр указывает какие настройки канала должны измениться по приходу рабочей метки. Формат регистра приведен на рисунке (см. Рисунок 7.23), значения полей приведены в таблице ниже (Таблица 7.12).

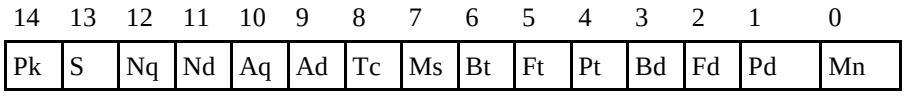


Рисунок 7.23 - Формат регистра CH_CNTRL

Таблица 7.12 – Назначение полей регистра CH_CNTRL

Биты	Название	Выполняемая функция
[31:15]	Reserved	Зарезервированы.
[14]	Pk	1 - регистр **CH_PACK** будет переписан из теневого в рабочий при установленном бите в регистре **START** по первой рабочей метке. 0 - значение регистра **CH_PACK** не изменится.
[13]	S	1 - регистр **CH_SELR** будет переписан из теневого в рабочий при установленном бите в регистре **START** по первой рабочей метке. 0 - значение регистра **CH_SELR** не изменится.
[12]	Nq	1b: регистр управления **CH_H2NORM** нормализатором H2 будет переписан из теневого в рабочий при установленном бите в регистре **START** по первой рабочей метке. 0b: не переписывать теневой регистр **CH_H2NORM** в рабочий.
[11]	Nd	1b: регистр управления **CH_H1NORM** нормализатором H1 будет переписан из теневого в рабочий при установленном бите в регистре **START** по первой рабочей метке. 0b: не переписывать теневой регистр **CH_H1NORM** в рабочий.
[10]	Aq	1b: регистр управления **CH_H2CNTR** дециматором HC2 будет переписан из теневого в рабочий при установленном бите в регистре START по первой рабочей метке. 0b: не переписывать теневой регистр CH_H2CNTR в рабочий.
[9]	Ad	1b: регистр управления **CH_H1CNTR** дециматором HC1 будет переписан из теневого в рабочий при установленном бите в регистре **START** по первой рабочей метке. 0b: не переписывать теневой регистр **CH_H1CNTR** в рабочий.
[8]	Tc	1b: регистр управления **CH_TCR** генератором T-кодов будет переписан из теневого в рабочий при установленном бите в регистре **START** по первой рабочей метке. 0b: не переписывать теневой регистр **CH_TCR** в рабочий.
[7]	Ms	1b: регистр управления **CH_MSCR** рабочей меткой будет переписан из теневого в рабочий при установленном бите в регистре **START** по первой рабочей метке. 0b: не переписывать теневой регистр **CH_MSCR** в рабочий.
[6]	Bt	1: разрешить однократное суммирование регистра начального смещения генератора фазы схемы T-кодов **TFB** по первой рабочей метке. 0 : однократное суммирование регистра начального смещения генератора фазы схемы T-кодов **TFB** не производить
[5]	Ft	1: регистр частоты генератора фазы схемы T-кодов **TFQ** будет переписан из теневого в рабочий по первой рабочей метке; 0 : не переписывать теневой регистр частоты генератора фазы схемы T-кодов **TFQ**.
[4]	Pt	1 : обнулить выход регистра текущей фазы генератора фазы схемы T-кодов TFT по первой рабочей метке. 0 : обнуление регистра текущей фазы генератора фазы схемы T-кодов TFT не будет
[3]	Bd	1: разрешить однократное суммирование регистра начального смещения гетеродина FB по первой рабочей метке. 0 : однократное суммирование регистра начального смещения гетеродина FB не производить
[2]	Fd	1: регистр частоты гетеродина FQ будет переписан из теневого в рабочий по первой рабочей метке. 0 : не переписывать теневой регистр частоты гетеродина FQ в рабочий
[1]	Pd	1 : обнулить выход регистра текущей фазы гетеродина FT по первой рабочей метке. 0 : обнуление регистра текущей фазы гетеродина FT не будет
[0]	Mn	1: рабочей меткой для регистров START, STOP и бит Pd, Fd, Bd, Pt, Ft, Bt становится ближайший сигнал основного временного интервала от интервального таймера. 0 : формирование рабочей метки осуществляется в соответствии с настройкой регистра CH_MSCR

Регистр доступен по чтению и записи.

Запись рабочих регистров конфигурации канала из теневых произойдет только если установлен соответствующий бит в данном регистре и установлен соответствующий данному каналу бит в регистре **START**. После прихода рабочей метки происходит сброс всех битов.

При помощи бита **Mn** предусмотрена возможность делать коррекцию канала по ближайшему сигналу основного временного интервала, а не только по рабочей метке. Т.е. ближайший сигнал основного временного интервала становится рабочей меткой. При этом рабочая метка,

сформированная при Mn=1, может оказывать влияние только на запуск, останов и биты **Pd**, **Fd**, **Bd**, **Pt**, **Ft**, **Bt**. Например, регистр **CH_H1CNTR** не может быть переписан из теневого в рабочий по ближайшей рабочей метке, сформированной при Mn=1 и Ms=0.

7.4.2.3 Регистр выбора входных данных CH_SELR

Данный регистр осуществляет выбор источника входных данных, формат перекодировки сигналов SIGNx/MAGNx и тип округления старших восьми разрядов гетеродина данных. Формат регистра приведен на рисунке (см Рисунок 7.24), значения полей приведены в таблице ниже (Таблица 7.13). Регистр доступен по чтению и записи.

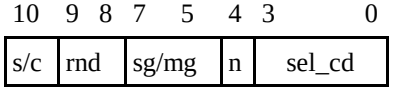


Рисунок 7.24 - Формат данных регистра CH_SELR

Таблица 7.13 – Назначение полей регистра CH_SELR

<table>
<tr><th>Биты</th><th>Назва-ние</th><th colspan="2">Выполняемая функция</th></tr>
<tr><td>[31:10]</td><td>Res.</td><td colspan="2">Зарезервированы.</td></tr>
<tr><td>[10]</td><td>s/c</td><td colspan="2">sin/cos – настройка выходов гетеродина sin[5:0] и cos[5:0]:
1 - sin[5:0] =0; cos[5:0] = 1;
0 - выходы sin[5:0] и cos[5:0] устанавливается согласно значению счетчика гетеродина.</td></tr>
<tr><td>[9:8]</td><td>rnd</td><td colspan="2">Тип округления восьми старших ([31:24]) разрядов гетеродина:
00b – к ближайшему значению (23й разряд регистра гетеродина FT суммируется со старшими разрядами FT[31:24]);
01b – округление в большую сторону с учетом знака (плюс единица всегда);
1xb – округление к нулю (простое отбрасывание).</td></tr>
<tr><td>[7:5]</td><td>sg/mg</td><td colspan="2">Тип перекодировки разрядов SIGN/MAGN:</td></tr>
<tr><td></td><td></td><td>000b:
SIGN MAGN значение
0 0 +3
0 1 +1
1 0 -1
1 1 -3</td><td>001b:
SIGN MAGN значение
0 0 +1
0 1 +2
1 0 -1
1 1 -2</td></tr>
<tr><td>[4]</td><td>n</td><td colspan="2">Признак комплексных данных на входе канала:
1b – входные данные являются комплексными;
0b – входные данные являются действительными (мнимая часть входных данных обнуляется)</td></tr>
<tr><td>[3:0]</td><td>sel_cd</td><td>0000b – Re = ADC0, Im = ADC1;
0001b – Re = ADC2, Im = ADC3;
0010b – Re = S0/M0, Im = S1/M1;
0011b – Re = S2/M2, Im = S3/M3;
0100b – Re = S4/M4, Im = S5/M5;
0101b – Re = S6/M6, Im = S7/M7;
0110b – Re = S8/M8, Im = S9/M9;
0111b – Re = S10/M10, Im = S11/M11;</td><td>1000b – Re = ADC1, Im = ADC0;
1001b – Re = ADC3, Im = ADC2;
1010b – Re = S1/M1, Im = S0/M0;
1011b – Re = S3/M3, Im = S2/M2;
1100b – Re = S5/M5, Im = S4/M4;
1101b – Re = S7/M7, Im = S6/M6;
1110b – Re = S9/M9, Im = S8/M8;
1111b – Re = S11/M11, Im = S10/M10;</td></tr>
</table>

Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **S** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре **START**. Чтение осуществляется из рабочего регистра.

7.4.2.4 Регистр частоты гетеродина CH_FQ

Данный регистр задает частоту гетеродина. Формат регистра приведен на рисунке (см. Рисунок 7.25 – Формат регистра CH_FQ), значения полей приведены в таблице ниже (Таблица 7.14).

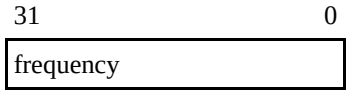


Рисунок 7.25 – Формат регистра CH_FQ

Таблица 7.14 – Назначение полей регистра CH_FQ

Биты	Название	Выполняемая функция
[31:0]	frequency	значение частоты гетеродина данных

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Fd** в регистре **CH_CNTRL**. Чтение осуществляется из рабочего регистра.

7.4.2.5 Регистр смещения фазы гетеродина CH_FB

Данный регистр задает смещение фазы. Формат регистра приведен на рисунке (см. Рисунок 7.30), значения полей приведены в таблице ниже (Таблица 7.15).

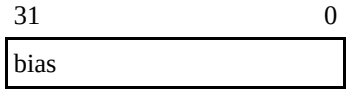


Рисунок 7.26 – Формат регистра CH_FB

Таблица 7.15 – Назначение полей регистра CH_FB

Биты	Название	Выполняемая функция
[31:0]	bias	значение смещения фазы

Регистр доступен по записи и чтению. Однократное суммирование данного регистра со значением текущей фазы происходит по рабочей метке при наличии бита **Bd** в регистре **CH_CNTRL**.

7.4.2.6 Регистр управления децимацией CH_H1CNTR

Данный регистр управляет децимацией в канале блока HC1 (см. Рисунок 7.2). Формат регистра приведен на рисунке (см Рисунок 7.27), значения полей приведены в таблице ниже (Таблица 7.16).

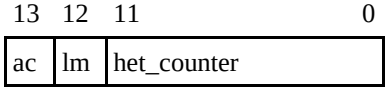


Рисунок 7.27 - Формат регистра CH_H1CNTR

Таблица 7.16 – Назначение полей регистра CH_H1CNTR

Биты	Название	Выполняемая функция
[31:14]	Reserved	Зарезервированы.
[13]	ac	0b – в данное поле всегда необходимо записывать 0
[12]	lm	Определяет влияние границы кадра на прореживание: 0b - по границе кадра прореживание начинается заново. Не до конца накопленные данные продают; 1b – граница кадра не прерывает процесс накопления/прореживания
[11:0]	het_counter	Период прореживания входного комплексного сигнала. Задается в прямом коде. 0 - прореживания не происходит; 1 - происходит прореживание двух элементов; 2 - происходит прореживание трех элементов; ..... 4095 - происходит накопление(децимация) 4096 элементов.

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Ad** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре **START**. Чтение осуществляется из рабочего регистра.

7.4.2.7 Регистр управления накоплением/прореживанием CH_H2CNTR

Данный регистр управляет накоплением в канале дециматора HC2, данные на который поступают с квадратора и режимом работы квадратора.

Формат регистра приведен на рисунке (см. Рисунок 7.28), значения полей приведены в таблице ниже (Таблица 7.17).

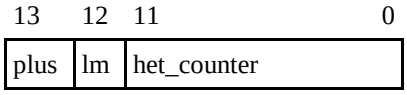


Рисунок 7.28 - Формат регистра CH_H2CNTR

Таблица 7.17 – Назначение полей регистра CH_H2CNTR

Биты	Название	Выполняемая функция
[31:14]	Reserved	Зарезервированы.
[13]	plus	Определяет функцию квадратора: 0h - комплексное возведение в квадрат; 1h - вычисление мощности сигнала;
[12]	lm	Определяет влияние границы кадра на прореживание: 0b - по границе кадра прореживание начинается заново. Не до конца прореженные данные продают; 1b – граница кадра не прерывает процесс прореживания
[11:0]	het_counter	Период накопления/прореживания входного комплексного сигнала. Задается в прямом коде. 0 - накопления/прореживания не происходит; 1 - происходит накопление(прореживание) двух элементов; 2 - происходит накопление(прореживание) трех элементов; ..... 4095 - происходит накопление(децимация) 4096 элементов.

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Aq** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре **START**. Чтение осуществляется из рабочего регистра.

7.4.2.8 Регистр управления нормализатором CH_H1NORM

Данный регистр управляет настройкой нормализатора H1, расположенного после децимато-ра HC1. Формат регистра приведен на рисунке (см Рисунок 7.29), значения полей приведены в таблице ниже (Таблица 7.18).

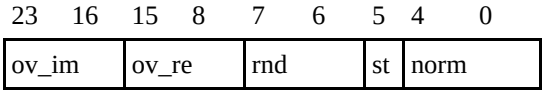


Рисунок 7.29 - Формат регистра CH_H1NORM

Таблица 7.18 – Назначение полей регистра CH_H1NORM

Биты	Название	Выполняемая функция
[31:24]	undefined	Зарезервированы.
[23:16]	ov_im	Количество переполнений в нормализаторе по мнимой составляющей в предыдущем кадре. Поле игнорируется при записи и имеет смысл только при чтении
[15:8]	ov_re	Количество переполнений в нормализаторе по действительной составляющей в предыдущем кадре. Поле игнорируется при записи и имеет смысл только при чтении
[7:6]	rnd	Тип округления результата нормализации: 00b – округление к ближайшему значению; 01b – округление в большую сторону с учетом знака (плюс единица всегда); 1xb – округление к нулю (простое отбрасывание).
[5]	st	0 – режим насыщения выключен; 1 – режим насыщения включен.
[4:0]	norm	выбор разрядов при нормализации 00h - выделяются 2 младших разряда результата и дополняются шестью нулями; 01h - выделяются 3 младших разряда результата и дополняются пятью нулями; … 06h - выделяются 8 младших разрядов результата; … 1fh – выделяются разряды с 32го по 25й результата

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Nd** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре **START**. Чтение осу-ществляется из рабочего регистра.

7.4.2.9 Регистр управления нормализатором CH_H2NORM

Данный регистр управляет настройкой нормализатора H2, расположенного после децимато-ра HC2. Формат регистра и назначение полей такое же как и у регистра **CH_H1NORM** (см. главу 7.4.2.8).

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Nq** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре **START**. Чтение осу-ществляется из рабочего регистра.

7.4.2.10 Регистр управления схемой упаковки данных CH_PACK

Данный регистр управляет упаковкой обработанных данных в канале в 64-х разрядные слова для записи в память. Формат регистра приведен на рисунке (см. Рисунок 7.30), значения полей приведены в таблице ниже (Таблица 7.19).

2	1	0
q/h	pack	

Рисунок 7.30 - Формат регистра CH_PACK

Таблица 7.19 – Назначение полей регистра CH_PACK

Биты	Название	Выполняемая функция
[31:3]	Reserved	Зарезервированы.
[2]	q/h	Управление входным коммутатором блока упаковки: 1b – входные данные поступают с нормализатора квадратора HC2; 0b – входные данные поступают с нормализатора гетеродина HC1.
[1:0]	pack	Управление упаковкой комплексных данных: 00b – упаковка выключена – запись в память не производится; 01b – упаковка 2-х битовая; 10b – упаковка 4-х битовая; 11b – упаковка8-ми битовая (байтовая).

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Pk** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре **START**. Чтение осуществляется из рабочего регистра.

7.4.2.11 Регистр начального адреса четного CH_ADR_EV и нечетного CH_ADR_ODD буферов

Данные регистры определяет начальные адреса в памяти буферов для записи данных. Формат регистров приведен на рисунке (см. Рисунок 7.31), значения полей приведены в таблице ниже (Таблица 7.20).

31	0
Memory_address	

Рисунок 7.31 - Формат регистров CH_ADR_EV и CH_ADR_ODD

Таблица 7.20 – Назначение полей регистров CH_ADR_EV и CH_ADR_ODD

Биты	Название	Выполняемая функция
[31:0]	Memory_address	начальный адрес буфера в памяти (адресное пространство в адресном пространстве NMU)

Регистры доступны только по записи. Так как **DDC** заполняет память только 64-х битными словами, то младший разряд адреса должен быть равен нулю. Производить запись необходимо обязательно в два регистра (в регистр **CH_ADR_EV** и **CH_ADR_ODD**) и обязательно в строгой последовательности: сначала запись **CH_ADR_EV**, следующая команда - запись в **CH_ADR_ODD.** Запись только одного регистра запрещена (например, загрузить начальный адрес только четного кадра и сохранить старое значение начального адреса нечетного кадра нельзя).

7.4.2.12 Регистр управления схемой Т-кодов CH_TCR

Данный регистр задает режим работы генератора Т-кодов. Формат регистра приведен на рисунке (см. Рисунок 7.32), значения полей приведены в таблице ниже (Таблица 7.21).

8	7	6	5	4	3	2	1	0
Z_M	M_I	MN	Z_1	Z_0	LE_1	LE_0	IE_1	IE_0

Рисунок 7.32 - Формат регистра CH_TCR

Таблица 7.21 – Назначение полей команды CH_TCR

Биты	Название	Выполняемая функция
[8]	Z_M	Режим модуляции меандром: 0b - модуляции нет; 1b - модуляция есть;
[7]	M_I	Управление инверсией мудулирующего сигнала 0b - инверсии нет; 1b -инверсия есть.
[6]	MN	Выбор сигнала для модуляции: 0b - модуляция осуществляется 30-м разрядом регистра **CH_TFT**; 1b - модуляция осуществляется 31-м разрядом регистра **CH_TFT**.
[5]	Z_1	Обнуление выхода TS1_26 0 - обнулить выход TS1_26 1 - не обнулять выход TS1_26
[4]	Z_0	Обнуление выхода TS0_26 0 - обнулить выход TS0_26 1 - не обнулять выход TS0_26
[3]	LE_1	Использование счетчика длины последовательности TCG1 0 - счетчик длины последовательности не используется; 1 - счетчик длины последовательности используется.
[2]	LE_0	Использование счетчика длины последовательности TCG0 0 - счетчик длины последовательности не используется; 1 - счетчик длины последовательности используется.
[1]	IE_1	Инвертирования обратной связи рабочего сдвигового регистра TS1 0 - обратная связь не инвертируется 1 - обратная связь инвертируется
[0]	IE_0	Инвертирования обратной связи рабочего сдвигового регистра TS0 0 - обратная связь не инвертируется 1 - обратная связь инвертируется

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Tc** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре **START**. Чтение осуществляется из рабочего регистра.

7.4.2.13 Регистр частоты генератора фазы схемы Т-кодов CH_TFQ

Данный регистр задает частоту генератора фазы схемы Т-кодов. Формат регистра приведен на рисунке (см. Рисунок 7.33), значения полей приведены в таблице ниже (Таблица 7.22).

31	0
frequency	

Рисунок 7.33 – Формат регистра CH_TFQ

Таблица 7.22 – Назначение полей регистра CH_TFQ

Биты	Название	Выполняемая функция
[31:0]	frequency	значение частоты смены символа данных

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Ft** в регистре **CH_CNTRL**. Чтение осуществляется из рабочего регистра.

7.4.2.14 Регистр смещения фазы генератора фазы схемы Т-кодов CH_TFB

Данный регистр задает смещение фазы. Формат регистра приведен на рисунке (см. Рисунок 7.34), значения полей приведены в таблице ниже (Таблица 7.23).

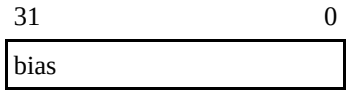


Рисунок 7.34 – Формат регистра CH_TFB

Таблица 7.23 – Назначение полей регистра CH_FB

Биты	Название	Выполняемая функция
[31:0]	bias	значение смещения фазы

Регистр доступен по записи и чтению. Однократное суммирование данного регистра со значением текущей фазы происходит по рабочей метке при наличии бита **Bt** в регистре **CH_CNTRL**.

7.4.2.15 Регистры длины последовательности схемы Т-кодов CH_STCG0, CH_STCG1

Данные регистры задают длину последовательности. Формат регистров приведен на рисунке (см. Рисунок 7.35), значения полей приведены в таблице ниже (Таблица 7.24).

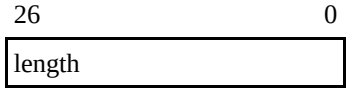


Рисунок 7.35 – Формат регистров CH_STCGx

Таблица 7.24 – Назначение полей регистра CH_STCGx

Биты	Название	Выполняемая функция
[26:0]	length	длина последовательности Т-кодов. Чтобы задать число слов, равное N, необходимо задать число (-N) в дополнительном коде.

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит при старте канала. Чтение осуществляется из рабочего регистра.

7.4.2.16 Регистры текущей длины последовательности схемы Т-кодов CH_TCG0, CH_TCG1

Данные регистры задают длину последовательности, используемую при старте канала. Это необходимо когда последовательность начинается не с первого элемента. Формат регистров приведен на рисунке (см. Рисунок 7.36), значения полей приведены в таблице ниже (Таблица 7.25).

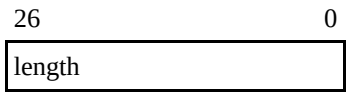


Рисунок 7.36 – Формат регистров CH_TCGx

Таблица 7.25 – Назначение полей регистра CH_TCGx

Биты	Название	Выполняемая функция
[26:0]	length	длина последовательности Т-кодов при старте канала. Чтобы задать число слов, равное N, необходимо задать число (-N) в дополнительном коде.

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит при старте канала. Чтение осуществляется из рабочего регистра, содержащего текущее значение длины на момент прихода рабочей метки (TCG_RD см. Рисунок 7.6)

7.4.2.17 Регистры маски схемы Т-кодов CH_TM0, CH_TM1

Данные регистры задают маску в схеме обратной связи сдвигового регистра. Формат регистров приведен на рисунке (см. Рисунок 7.37), значения полей приведены в таблице ниже (Таблица 7.26).

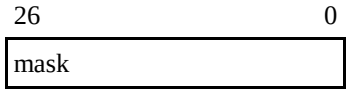


Рисунок 7.37 – Формат регистров CH_TMx

Таблица 7.26 – Назначение полей регистра CH_TMx

Биты	Название	Выполняемая функция
[26:0]	mask	определяет какие разряды сдвигового регистра участвуют в формировании младшего разряда. Побитно с регистром TSx фомирует логическое "И"

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит при старте канала. Чтение осуществляется из рабочего регистра.

7.4.2.18 Сдвиговые регистры схемы Т-кодов CH_TS0, CH_TS1

Старшие разряды сдвиговых регистров формируют выход генератора Т-кодов. Формат регистров приведен на рисунке (см. Рисунок 7.38), значения полей приведены в таблице ниже (Таблица 7.27).

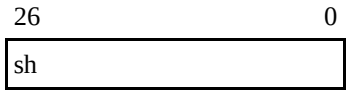


Рисунок 7.38 – Формат регистров CH_TSx

Таблица 7.27 – Назначение полей регистра CH_TSx

Биты	Название	Выполняемая функция
[26:0]	sh	значение сдвигового регистра

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит при старте канала, инициализируя рабочий регистр начальным значением. Чтение осуществляется из рабочего регистра.

7.4.2.19 Регистры CH_TIS0, CH_TIS1

Данные регистры используются при работе с использованием счетчика длины последовательности для инициализации сдвиговых регистров при достижении последнего элемента последовательности. Формат регистров приведен на рисунке (см. Рисунок 7.39), значения полей приведены в таблице ниже (Таблица 7.28).

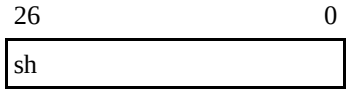


Рисунок 7.39 – Формат регистров CH_TSx

Таблица 7.28 – Назначение полей регистра CH_TSx

Биты	Название	Выполняемая функция
[26:0]	sh	значение сдвигового регистра для инициализации при достижении последнего элемента последовательности

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит при старте канала. Чтение осуществляется из рабочего регистра.

7.4.3 Регистры управления аппаратного канала с КИХ-фильтром (канал 12)

7.4.3.1 Регистр частоты гетеродина F_FQ

Данный регистр задает частоту гетеродина. Регистр доступен по записи и чтению. Назначение и формат регистра полностью совпадают с регистром CH_FQ. Отличие состоит только в отсутствии теневого регистра. Запись происходит в рабочий регистр, поэтому менять значение данного регистра во время работы канала запрещено.

7.4.3.2 Регистр смещения фазы гетеродина F_FT

Данный регистр задает начальное значение фазы. Формат регистра приведен на рисунке (см. Рисунок 7.40), значения полей приведены в таблице ниже (Таблица 7.29).

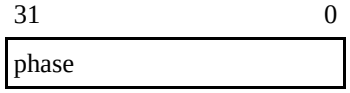


Рисунок 7.40 – Формат регистра F_FT

Таблица 7.29 – Назначение полей регистра F_FT

Биты	Название	Выполняемая функция
[31:0]	phase	значение начальной фазы гетеродина данных

Регистр доступен по записи и чтению. Запись происходит в рабочий регистр, поэтому менять значение данного регистра во время работы канала запрещено.

7.4.3.3 Регистр управления схемой упаковки данных F_PACK

Данный регистр управляет распаковкой данных поступающих из памяти, конфигурацией фильтра и упаковкой обработанных данных в канале в 64-х разрядные слова для записи в память. Формат регистра приведен на рисунке (см. Рисунок 7.41), значения полей приведены в таблице ниже (Таблица 7.30).

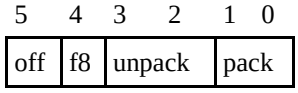


Рисунок 7.41 - Формат регистра F_PACK

Таблица 7.30 – Назначение полей регистра F_PACK

Биты	Название	Выполняемая функция
[31:5]	Reserved	Зарезервированы.
[5]	off	Логически изолирует фильтр: 0b – фильтр подключен; 1b – фильтр логически изолирован от остальной части схемы;
[4]	f8	Конфигурирует фильтр: 0b – фильтр выполняет функцию КИХ фильтра над 4-х разрядными входными комплексными данными и имеет порядок 256; 1b – фильтр выполняет функцию КИХ фильтра над 8-ми разрядными входными комплексными данными и имеет порядок 128;
[3:2]	unpack	Управляет распаковкой входных данных: 00b – входные данные комплексные 2-х битные; 01b – входные данные комплексные 4-х битные; 10b – входные данные комплексные 8-х битные; 11b – входные данные действительные 11-ти битные.
[1:0]	pack	Управляет упаковкой обработанных данных для записи в память: 00b – упаковка выключена – не происходит запись в память; 01b – упаковка 2-х битовая; 10b – упаковка 4-х битовая; 11b – упаковка 8-ми битовая (байтовая).

Регистр доступен по записи и чтению. Запись происходит в сразу в рабочий регистр поэтому менять значение данного регистра во время работы канала запрещено.

7.4.3.4 Регистр выбора входных данных F_SELR

Данный регистр осуществляет тип округления старших восьми разрядов гетеродина данных. Формат регистра приведен на рисунке (см. Рисунок 7.42), значения полей приведены в таблице ниже (Таблица 7.31). Регистр доступен по чтению и записи.

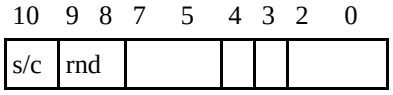


Рисунок 7.42 - Формат регистра F_SELR

Таблица 7.31 – Назначение полей регистра F_SELR

Биты	Название	Выполняемая функция
[31:10;7:0]	Res.	Зарезервированы.
[10]	s/c	sin/cos – настройка выходов гетеродина sin[5:0] и cos[5:0]: 1 - sin[5:0] =0; cos[5:0] = 1; 0 - выходы sin[5:0] и cos[5:0] устанавливается согласно значению счетчика гетеродина.
[9:8]	rnd	Тип округления восьми старших ([31:24]) разрядов гетеродина: 00b – к ближайшему значению ((23й разряд регистра гетеродина FT суммируется со старшими разрядами FT[31:24]; 01b – округление в большую сторону с учетом знака (плюс единица всегда); 1xb – округление к нулю (простое отбрасывание).

Регистр доступен по чтению и записи. Запись происходит сразу в рабочий регистр, поэтому менять значение данного регистра во время работы канала запрещено.

7.4.3.5 Регистр управления децимацией F_H1CNTR

Данный регистр управляет децимацией блока HC1 (см. Рисунок 7.11). Формат регистра приведен на рисунке (см. Рисунок 7.43), значения полей приведены в таблице ниже (Таблица 7.32).

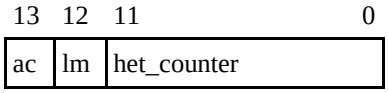


Рисунок 7.43 - Формат регистра F_H1CNTR

Таблица 7.32 – Назначение полей регистра F_H1CNTR

Биты	Название	Выполняемая функция
[31:14]	Reserved	Зарезервированы.
[13]	ac	0b - при записи данное поле должно быть 0
[11:0]	het_counter	Период прореживания входного комплексного сигнала. Задается в прямом коде. Если 0, то накопления/прореживания не происходит.

Регистр доступен по записи и чтению. Запись происходит в теневой регистр. Запись из теневого в рабочий регистр происходит по рабочей метке при наличии бита **Ad** в регистре **CH_CNTRL** и при установленном для данного канала бите в регистре **START**. Чтение осуществляется из рабочего регистра.

7.4.3.6 Регистр управления нормализатором F_H1NORM

Данный регистр управляет настройкой нормализатора H1, расположенного после дециматора HC1. Формат регистра и назначение полей такое же как и у регистра **CH_H1NORM** (см. главу 7.4.2.8).

Регистр доступен по чтению и записи. Запись происходит сразу в рабочий регистр, поэтому менять значение данного регистра во время работы канала запрещено.

7.4.3.7 Регистр управления нормализатором F_H2NORM

Данный регистр управляет настройкой нормализатора H2, расположенного после КИХ-фильтра. Формат регистра и назначение полей такое же как и у регистра **CH_H1NORM** (см. главу 7.4.2.8).

Регистр доступен по чтению и записи. Запись происходит сразу в рабочий регистр, поэтому менять значение данного регистра во время работы канала запрещено.

7.4.3.8 Регистры весов КИХ-фильтра F_W0,....,F_W256

Данные регистры содержат веса КИХ-фильтра. Формат регистров приведен на рисунке (см. Рисунок 7.44), значения полей приведены в таблице ниже (Таблица 7.33). Регистры доступны по записи.

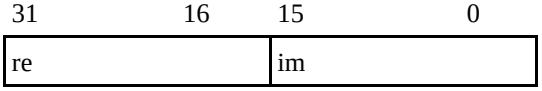


Рисунок 7.44 - Формат регистров F_Wx

Таблица 7.33 – Назначение полей регистров F_Wx

Биты	Название	Выполняемая функция
[31:16]	re	вещественная часть комплексного коэффициента КИХ-фильтра
[15:0]	im	мнимая часть комплексного коэффициента КИХ-фильтра

Загружать коэффициенты можно в любом порядке. Для фильтра 256-го порядка загружать надо 256 регистров F_W0 - F_W255, для фильтра 128-го порядка необходимо загрузить

128 весов в регистры F_W0 - F_W127 *и те же самые веса загрузить в регистры F_W128 - F_W255!!!*

7.4.3.9 Регистр начального адреса буфера записи данных F_ADR

Данный регистр определяет начальный адрес буфера в памяти для записи данных. Формат регистра приведен на рисунке (см. Рисунок 7.45), значения полей приведены в таблице ниже (Таблица 7.34).

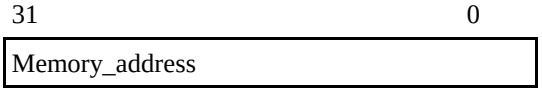


Рисунок 7.45 - Формат регистра F_ADR

Таблица 7.34 – Назначение полей регистра F_ADR

Биты	Название	Выполняемая функция
[31:0]	Memory_address	начальный адрес буфера в памяти (адресное пространство в адресном пространстве NMU)

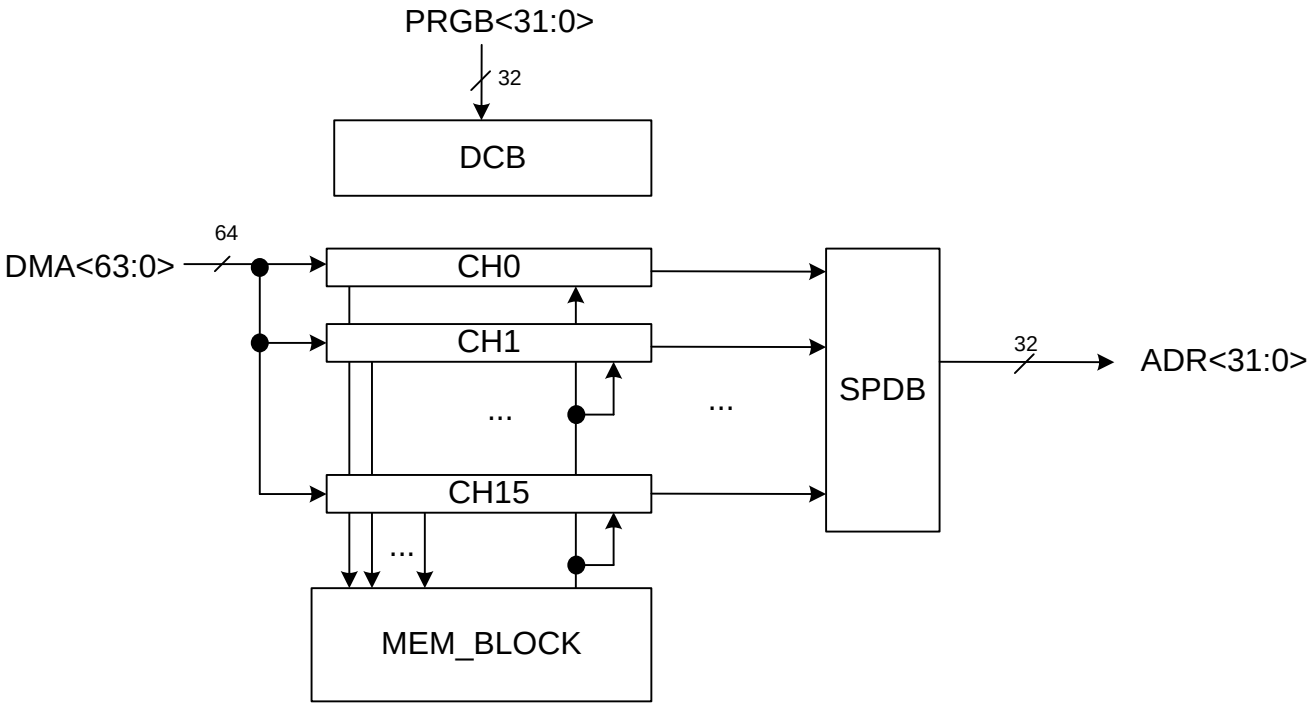
Регистр доступен только по записи. Так как **DDC** заполняет память только 64-х битными словами, то младший разряд адреса должен быть равен нулю.

7.4.3.10 Регистры управлениея генератором Т-кодов

Регистры управления генератором Т-кодов в канале с КИХ-фильтром (**F_TFT**, **F_TFQ**, **F_STCG0**, **F_STCG1**, **F_TCG0**, **F_TCG1**, **F_TM0**, **F_TM1**, **F_TS0**, **F_TS1**, **F_TIS0** и **F_TIS1)** имеют такой же формат и назначение как регистры управления генератором Т-кодов (**CH_TFB**, **CH_TFQ**, **CH_STCG0**, **CH_STCG1**, **CH_TCG0**, **CH _TCG1**, **CH _TM0**, **CH_TM1**, **CH_TS0**, **CH_TS1**, **CH_TIS0** и **CH_TIS1)** в каналае 0 - канале 7 (см. главы 7.4.2.12 - 7.4.2.19).

8 Блок аппаратных корреляторов CORB

Блок CORB состоит из 16 аппаратных каналов (CHx), блока памяти для хранения таблиц ПСП (MEM_BLOCK), схемы приоритетов и анализа зависимых каналов (SPDB) и декодера входных каманд (DCB).



CHx - аппаратный канал корреляции
MEM_BLOCK – блок памяти для хранения таблицы генератора ПСП
SPDB – схема приоритетов и анализ зависимых каналов
DCB - декодер входных команд

Рисунок 8.1- Общая структура блока аппаратных корреляторов

DCB - декодер входных команд. Декодер осуществляет декодирование команды и формирует управляющие сигналы для записи или чтения программно доступных регистров блока, а также запись данных для таблицы в MEM_BLOCK. Запись и чтение возможна всеми процессорными системами (NMU1, NMU2, ARMU). Запись и чтение программно доступных регистров осуществляется только 32-х разрядными словами.

CHx - аппаратный канал, осуществляющий вычисление корреляционной функции между входными данными DMA[63:0] и псевдослучайной последовательностью. Псевдослучайная последовательность формируется генератором ПСП, который находится внутри канала.

SPDP - схемы приоритетов и анализа зависимых каналов. Схема осуществляет выбор одного запроса от аппаратных каналов для подкачки данных и выдачу адреса ячейки памяти, из которой надо осуществить подкачку данных. Канал, запрос от которого был обработан, получает наименьший приоритет. При нескольких активных запросах от каналов выбирается запрос от канала, который имеет больший приоритет.

MEM_BLOCK - блок памяти для хранения таблиц ПСП. Данный блок содержит память объемом 2048x32 бит, которая предназначена для хранения таблиц, используемых при формировании ПСП. Количество таблиц от 1 до 16.

Порядок работы с блоком CORB заключается в начальной настройке необходимого аппаратного канала и его запуск. После этого канал работает самостоятельно. Как только будет обработанно заданное число входных данных, канал останавливается. Останов канала может быть определен чтением конфигурационных регистров или по прерыванию.

8.1 Аппаратный канал CH

Структура аппаратного канала приведена на рисунке ниже.

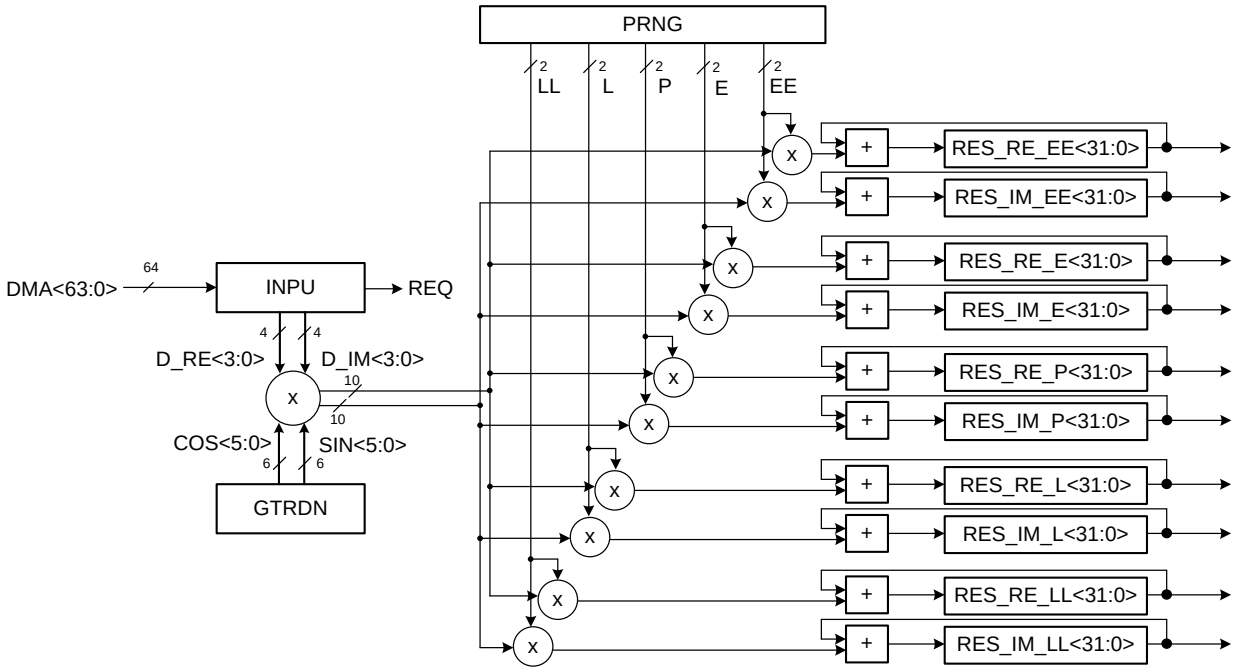


Рисунок 8.2-Структура аппаратного канала

Аппаратный канал состоит из входного устройства (INPU), гетеродина (GTRDN) и генератора псевдослучайных последовательностей (PRNG).

INPU формирует запрос на подкачку новых данных по мере надобности. Подкачка данных производится по последовательным адресам 64-х разрядными словами. Начальный адрес, с которого будет выполнена подкачка, и количество 64-х разрядных слов задается программно.

Получаемые данные DMA[63:0] представляют собой 64-х разрядные слова, в которых упакованы комплексные отсчеты. Поддерживается два вида упаковки: 2+2 бита и 4+4 бита (см. Рисунок 8.3). INPU производит распаковку входных данных и выдает распакованные комплексные отсчеты, которые поступают на умножитель.

63 62	61 60		7 6	5 4	3 2	1 0
D_RE15	D_IM15	….	D_RE1	D_IM1	D_RE0	D_IM0
D_RE31	D_IM31	….	D_RE17	D_IM17	D_RE16	D_IM16

2-х битная упаковка

63 60	59 56		15 12	11 8	7 4	3 0
D_RE7	D_IM7	….	D_RE1	D_IM1	D_RE0	D_IM0
D_RE15	D_IM15	….	D_RE9	D_IM9	D_RE8	D_IM8

4-х битная упаковка

Рисунок 8.3-Формат упакованных данных

8.1.1 Гетеродин (GTRDN)

Гетеродин состоит из 32-х разрядного регистра текущей фазы FT, регистра частоты FQ, сумматора и таблицы (см. Рисунок 8.4).

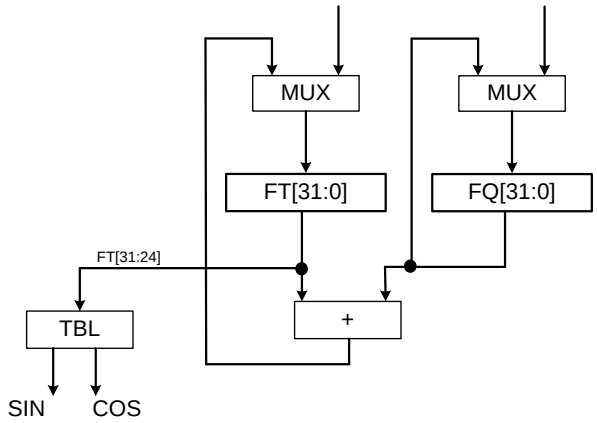


Рисунок 8.4- Структурная схема гетеродина

В каждом такте работы гетеродина происходит суммирование регистров FT и FQ с записью результата в регистр FT. Старшие разряды регистра FT поступают на вход таблицы, в которой в соответствии с поступающими разрядами происходит выдача значения синуса и косинуса. Поддерживается два вида таблицы:

- на вход подаются разряды FT[31:26], разрядность выходного синуса и косинуса 4 бита;
- на вход подаются разряды FT[31:24], разрядность выходного синуса и косинуса 6 бит.

Выбор таблицы производится программно. Начальные значения FT и FQ задаются программно. Формула для расчета частоты синуса и косинуса и обратно приведены ниже:

$$f = fd*code * / 2^{32},$$

$$code = [f * 2^{32} /fd],$$

где f – частота гетеродина, выраженная в герцах; code – значение регистра FQ[31:0], fd - частота оцифровки комплексных отсчетов, выраженная в герцах.

Формулы для вычисления смещения фазы гетеродина **φ**, выраженной в радианах, в зависимости от кода в регистре FT[31:0] и обратно приведены ниже:

$$\varphi = code * 2*\pi / 2^{32},$$

$$code = [\varphi * 2^{32} /(2*\pi)],$$

где code – значение регистра FQ[31:0].

8.1.2 Генератор псевдослучайных последовательностей (PRNG).

Генератор псевдослучайных последовательностей предназначен для формирования псевдослучайной последовательности (P) заданной длины и с заданным периодом, а также двух опережающих (E, EE) и двух запаздывающих (L, LL) последовательностей по отношению к основной.

PRNG может работать в двух режимах:

- генератор, построенный на сдвиговых регистрах (SRG);
- табличный генератор (TBLG).

Выбор режима работы производится программно.

Структурная схема генератора представлена на рисунке ниже (см. Рисунок 8.5).

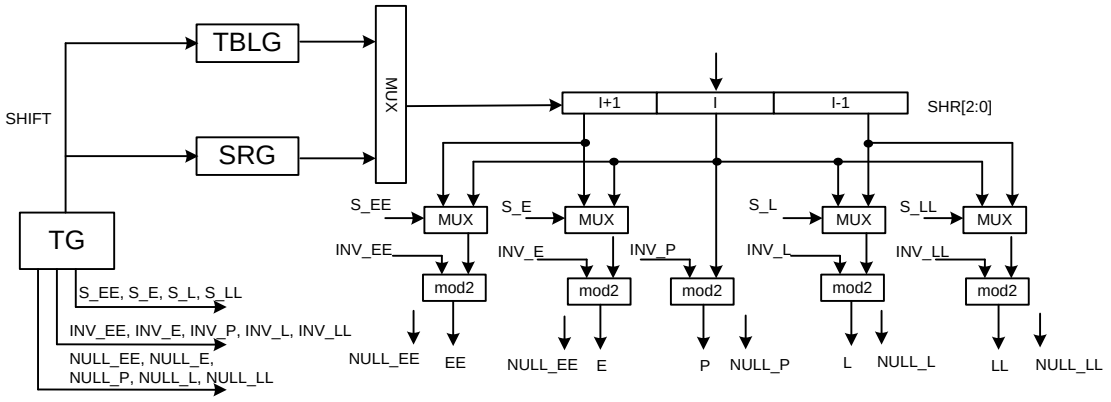


Рисунок 8.5- Структурная схема генератора псевдослучайных последовательностей (PRNG)

Также в состав PRNG входит генератор времени (TG) и регистр сдвига (SHR).

Генератор времени формирует управляющие сигналы для:

- формирования нового элемента ПСП (SHIFT);
- формирования значений E, EE, L, LL (S_EE, S_E, S_L, S_LL);
- формирования инверсии для BOC-модуляции (INV_EE, INV_E, INV_P, INV_L, INV_LL);
- обнуления значений E, EE, P, L, LL, используемых при кодовом уплотнении (NULL_EE, NULL_E, NULL_P, NULL_L, NULL_LL).

По сигналу shift происходит формирование нового элемента ПСП, который записывается в старший разряд сдвигового регистра SHR[2:0]. В самом сдвиговом регистре происходит сдвиг, и самый старый элемент ПСП выкидывается. При помощи сигналов управления S_E и S_EE происходит выбор между SHR[2] и SHR[1] для последовательности E и EE, при помощи сигналов управления S_L и S_LL происходит выбор между SHR[1] и SHR[0] для последовательности L и LL.

Первоначальную инициализацию регистра SHR[2:0] может выполнить программист путем программной записи в этот регистр, а так же может быть произведена автоматическая инициализация. При автоматической инициализации, до того как начнется штатная работа, происходит вычисление трех элементов ПСП, которыми инициализируется регистр SHR.

Генератор времени. Структурна схема генератора времени приведена на рисунке ниже (см. Рисунок 8.6)

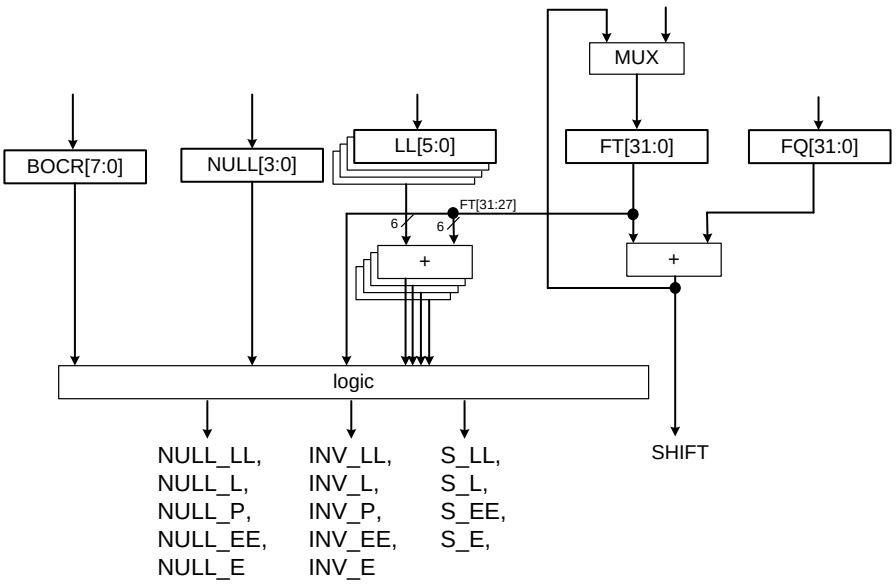


Рисунок 8.6- Структурная схема генератора времени (TG)

TG состоит из 32-х разрядного регистра текущей фазы FT, регистра частоты FQ, сумматора, регистров управления (BOCR, NULL, E,EE,L,LL) и комбинационной логики.

В каждом такте работы происходит суммирование регистров FT и FQ с записью результата в регистр FT. Если при суммировании возникает перенос из 31-го разряда, то формируется сигнал SHIFT. Также старшие разряды регистра FT поступают на вход сумматоров, где происходит суммирование с регистрами E и EE и вычитание регистров L и LL. По результатам суммирования и вычитания формируются остальные сигналы управления.

Начальные значения FT и FQ задаются программно. Формулы для расчета частоты смены символов ПСП приведены ниже:

$$f = fd*code * / 2^{32},$$

$$code = [f * 2^{32} /fd],$$

где f – частота смены символов ПСП, выраженная в герцах; code – значение регистра FQ[31:0], fd - частота оцифровки комплексных отсчетов, выраженная в герцах.

Формулы для вычисления смещения фазы внутри символа φ, выраженной в радианах, в зависимости от кода в регистре FT[31:0] и обратно приведены ниже:

$$\varphi = code * 2*\pi / 2^{32},$$

$$code = [\varphi * 2^{32} /(2*\pi)],$$

где code – значение регистра FT.

Табличное формирование ПСП (TBLG). Функциональная схема табличного формирования приведена на рисунке ниже. В наличии есть 65536 однобитовых ячеек. Таблица может занимать соседние ячейки. Регистр FIRST содержит адрес первой ячейки, регистр LAST содержит адрес последней ячейки, регистр CA содержит адрес текущей ячейки. После поступления сигнала SHIFT на выходе табличного генератора выбирается символ по адресу CA. После этого значение CA инкрементируется. Если CA = LAST, то вместо инкремента в CA записывается значение FIRST.

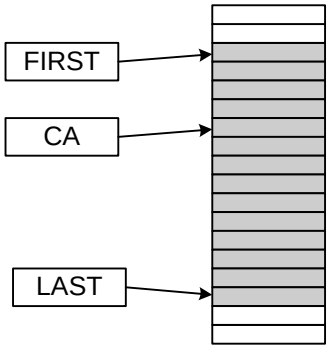


Рисунок 8.7- Схема табличного формирования ПСП

Формирование ПСП на сдвиговых регистрах(SRG). Функциональная схема данного генератора приведена на рисунке ниже (см. Рисунок 8.8). Результат получается путем суммирования "по модулю 2" старших разрядов сдвиговых регистров TS0 и TS1. Начальные значения сдвиговых регистров TS0 и TS1 загружаются программно. По сигналу SHIFT от блока TG происходит сдвиг регистров TS0 и TS1 в сторону старших разрядов. Младший разряд сдвигового регистра получается сложением "по модулю 2" всех разрядов сдвигового регистра, умноженных на маску TMx. Маска TM0 маскирует разряды регистра TS0, TM1 - TS1. Регистры TM0 и TM1 загружаются программно перед началом работы генератора. Помимо этого значение обратной связи может инвертироваться. Данная функция настраивается программно полем TCR[1] и TCR[0] регистра OPR.

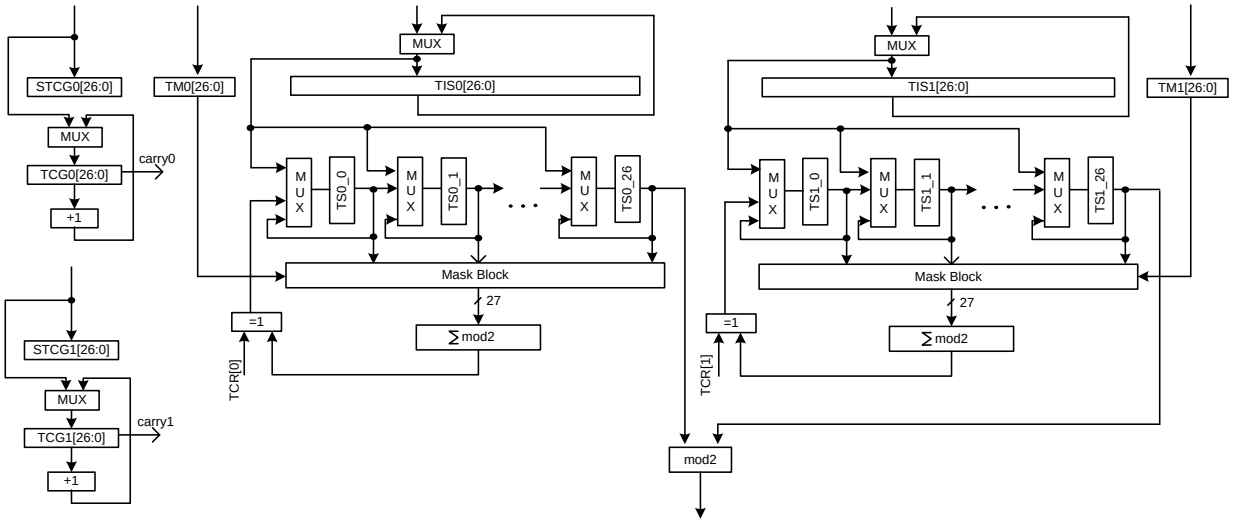


Рисунок 8.8- Функциональная схема формирования ПСП на сдвиговых регистрах

Также возможна работа генератора с использованием счетчика длины последовательности. В данном режиме происходит инкрементирование регистра TCGx при появлении сигнала SHIFT. Когда возникает сигнал переноса carry из старшего разряда, происходит перезапись регистра TISx в TSx (вместо сдвига) и регистра STCGx в TCGx. После этого работа продолжается. Значение регистров STCGx и TCGx записывается программно перед началом работы генератора.

8.2 Схема приоритетов и анализа зависимых каналов (SPDP)

SPDP осуществляет выбор одного запроса от аппаратных каналов для подкачки данных и выдачу адреса ячейки памяти, из которой надо осуществить подкачку данных. Канал, запрос от которого был обработан, получает наименьший приоритет. При нескольких активных запросах от каналов выбирается запрос от канала, который имеет больший приоритет.

При упаковке данных, получаемых по каналу DMA, 2+2 бита пропускной способности входной шины хватает для одновременной работы 16 каналов без простоев (в каждом 64-х разрядном слове 16 отсчетов, т.е. канал работает с одним 64-х разрядным словом 16 тактов. За это время производится подкачка данных для остальных каналов).

При упаковке данных, получаемых по каналу DMA, 4+4 бита пропускной способности входной шины не хватает для одновременной работы 16 каналов без простоев. Чтобы производительность введен механизм зависимых каналов. В случае если несколько каналов используют один пакет данных, то механизм позволяет произвести подкачку только один раз для всех каналов. Если данный механизм не использовать, то каждый канал будет подкачивать себе данные независимо от других и тем самым входная шина будет больше загружена. На рисунках ниже проиллюстрировано использование механизма зависимых каналов для трех каналов использующих одни и те же входные данные

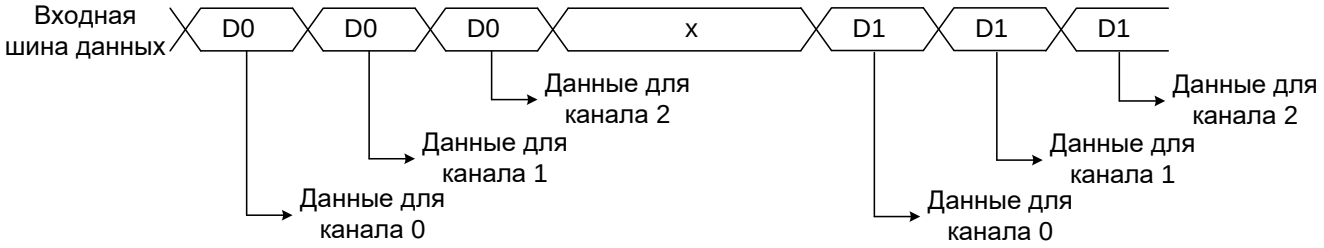


Рисунок 8.9- Механизм зависимых каналов не используется

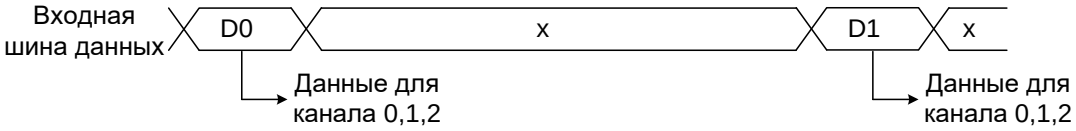


Рисунок 8.10- Механизм зависимых каналов используется

Для использования механизма зависимых каналов необходимо выделить группу каналов, которые используют одни и те же входные данные. Ниже приведены условия, которым должна обладать выбранная группа каналов:

- номера каналов использующих механизм зависимых каналов должны располагаться последовательно (1,2,3 или 4,5). Выбор непоследовательной нумерации запрещен (например, 1,5, 8);
- канал с наименьшим номером является основным. Адреса для подкачки данных будут формироваться этим каналом. Каналы с остальными номерами являются зависимыми;
- все каналы должны иметь первый элемент обработки в одном и том же 64-х разрядном слове. Номер первого элемента обработки внутри 64-х разрядного слова у каждого канала свой;

- количество 64-х разрядных слов для обработки каждый канал может иметь свое (значение регистра **CNTR**). Но основной канал должен иметь наибольшее число из всей группы.

Каналы, удовлетворяющие данным критериям, могут использовать механизм зависимых каналов. Назначение основных и зависимых каналов программируется регистром **DR**. У основной канала должен быть записан регистр **CADR**, у зависимых каналов значение данного регистра игнорируется. После этого производится остальная настройка каналов и каналы запускаются. Остановка каналов происходит независимо (зависит от значения регистра **CNTR** в каждом канале), но основной канал останавливается последним, так как у него должно быть максимально значение регистра **CNTR** (или такое же как и у остальных).

8.3 Программная модель CORB

Адресное пространство CORB доступно по адресам 000D_0000h - 000D_FFFFh при 32-х битной адресации. Адресное пространство регистров CORB состоит из трех частей:

- регистры общего назначения (для всех каналов);
- регистры канала;
- ячейка памяти табличного генератора ПСП.

Ниже приведен формат младших разрядов адреса для блока **CORB**. Для адресации процессорной системой ARMU необходимо добавить два 2b'00 со стороны младших разрядов.

12 11	10 … 7	6 … 0
0 0	Адрес ячейки памяти	
1 0	Номер канала	Код регистра
1 1	Код регистра общ. назначен.	

Рисунок 8.11- Формирования адресов для блока CORB

Ниже в таблице приведены адреса регстров общего назначения.

Таблица 8.1 – Адреса регистров общего назначения

	Адрес вектор в пространстве NMU, hex	Адрес вектор в пространстве ARMU, hex	Чтение	Запись
1	000D_1800	0034_6000	WR	WRset
2	000D_1802	0034_6008	INTCHR	INTCHRrst
3	000D_1803	0034_600C	INTCHR	INTCHRset
4	000D_1804	0034_6010	MASK	MASK
5	000D_1805	0034_6014	INTREQ	INTREQrst
6	000D_1806	0034_6018	DR	DRrst
7	000D_1807	0034_601C	DR	DRset
	000D_1808	0034_6020	ERR_REG	ERR_EN

В таблицах ниже (Таблица 8.2, Таблица 8.3) приведены младшие адреса регистров каналов в адресном пространстве NMU. Старшие разряды адресы всегда равны ***000D*** (т.е. адрес имеет вид ***000D_xxxx***). Адресация в процессорной системе ARMU получается умножением адреса NMU на 4.

Таблица 8.2 – младшие разряды адреса регистров каналов с 0 по 7 в адресном пространстве NMU

	тип операции		номер канала							
	запись	чтение	0	1	2	3	4	5	6	7
1	CADR_DMA	CADR_DMA	1000	1080	1100	1180	1200	1280	1300	1380
2	CNTR_DMA	CNTR_DMA	1001	1081	1101	1181	1201	1281	1301	1381
3	UTR_DMA	UTR_DMA	1002	1082	1102	1182	1202	1282	1302	1382
4	GET_CNTRL	GET_CNTRL	1003	1083	1103	1183	1203	1283	1303	1383
5	GET_FT	GET_FT	1004	1084	1104	1184	1204	1284	1304	1384
6	GET_FQ	GET_FQ	1005	1085	1105	1185	1205	1285	1305	1385
7	SHCR	SHCR	1006	1086	1106	1186	1206	1286	1306	1386
8	TG_FQ	TG_FQ	1007	1087	1107	1187	1207	1287	1307	1387
9	RES_AU	RD_EE_RE	1008	1088	1108	1188	1208	1288	1308	1388
10		RD_EE_IM	1009	1089	1109	1189	1209	1289	1309	1389
11		RD_E_RE	100A	108A	110A	118A	120A	128A	130A	138A
12		RD_E_IM	100B	108B	110B	118B	120B	128B	130B	138B
13		RD_P_RE	100C	108C	110C	118C	120C	128C	130C	138C
14		RD_P_IM	100D	108D	110D	118D	120D	128D	130D	138D
15		RD_L_RE	100E	108E	110E	118E	120E	128E	130E	138E
16		RD_L_IM	100F	108F	110F	118F	120F	128F	130F	138F
17	*запрещено*	RD_LL_RE	1010	1090	1110	1190	1210	1290	1310	1390
18	*запрещено*	RD_LL_IM	1011	1091	1111	1191	1211	1291	1311	1391
19	TG_FT	TG_FT	1012	1092	1112	1192	1212	1292	1312	1392
20	BOCR	BOCR	1013	1093	1113	1193	1213	1293	1313	1393
21	CNTRL_gen	CNTRL_gen	1014	1094	1114	1194	1214	1294	1314	1394
22	STCG0	STCG0	1015	1095	1115	1195	1215	1295	1315	1395
23	TCG0	TCG0	1016	1096	1116	1196	1216	1296	1316	1396
24	CADR_TBL	CADR_TBL	1017	1097	1117	1197	1217	1297	1317	1397
25	TIS0	TIS0	1018	1098	1118	1198	1218	1298	1318	1398
26	FADR_TBL	FADR_TBL	1019	1099	1119	1199	1219	1299	1319	1399
27	TIS1	TIS1	101A	109A	111A	119A	121A	129A	131A	139A
28	LADR_TBL	LADR_TBL	101B	109B	111B	119B	121B	129B	131B	139B
29	TS0	TS0	101C	109C	111C	119C	121C	129C	131C	139C
30	TS1	TS1	101D	109D	111D	119D	121D	129D	131D	139D
31	OPR	OPR	101E	109E	111E	119E	121E	129E	131E	139E
32	TM0	TM0	101F	109F	111F	119F	121F	129F	131F	139F
33	TM1	TM1	1020	10A0	1120	11A0	1220	12A0	1320	13A0
34	STCG1	STCG1	1021	10A1	1121	11A1	1221	12A1	1321	13A1
35	TCG1	TCG1	1022	10A2	1122	11A2	1222	12A2	1322	13A2

Таблица 8.3 – младшие разряды адреса регистров каналов с 8 по 15 в адресном пространстве NMU

	тип операции		номер канала							
	запись	чтение	8	9	10	11	12	13	14	15
1	CADR_DMA	CADR_DMA	1400	1480	1500	1580	1600	1680	1700	1780
2	CNTR_DMA	CNTR_DMA	1401	1481	1501	1581	1601	1681	1701	1781
3	UTR_DMA	UTR_DMA	1402	1482	1502	1582	1602	1682	1702	1782
4	GET_CNTRL	GET_CNTRL	1403	1483	1503	1583	1603	1683	1703	1783
5	GET_FT	GET_FT	1404	1484	1504	1584	1604	1684	1704	1784
6	GET_FQ	GET_FQ	1405	1485	1505	1585	1605	1685	1705	1785
7	SHCR	SHCR	1406	1486	1506	1586	1606	1686	1706	1786
8	TG_FQ	TG_FQ	1407	1487	1507	1587	1607	1687	1707	1787
9	обнуление всех ре-	EE_RE	1408	1488	1508	1588	1608	1688	1708	1788
10	гистров корреляции	EE_IM	1409	1489	1509	1589	1609	1689	1709	1789
11		E_RE	140A	148A	150A	158A	160A	168A	170A	178A
12		E_IM	140B	148B	150B	158B	160B	168B	170B	178B
13		P_RE	140C	148C	150C	158C	160C	168C	170C	178C
14		P_IM	140D	148D	150D	158D	160D	168D	170D	178D
15		L_RE	140E	148E	150E	158E	160E	168E	170E	178E
16		L_IM	140F	148F	150F	158F	160F	168F	170F	178F
17	*запрещено*	LL_RE	1410	1490	1510	1590	1610	1690	1710	1790
18	*запрещено*	LL_IM	1411	1491	1511	1591	1611	1691	1711	1791
19	TG_FT	TG_FT	1412	1492	1512	1592	1612	1692	1712	1792
20	BOCR	BOCR	1413	1493	1513	1593	1613	1693	1713	1793
21	CNTRL_gen	CNTRL_gen	1414	1494	1514	1594	1614	1694	1714	1794
22	STCG0	STCG0	1415	1495	1515	1595	1615	1695	1715	1795
23	TCG0	TCG0	1416	1496	1516	1596	1616	1696	1716	1796
24	CADR_TBL	CADR_TBL	1417	1497	1517	1597	1617	1697	1717	1797
25	TIS0	TIS0	1418	1498	1518	1598	1618	1698	1718	1798
26	FADR_TBL	FADR_TBL	1419	1499	1519	1599	1619	1699	1719	1799
27	TIS1	TIS1	141A	149A	151A	159A	161A	169A	171A	179A
28	LADR_TBL	LADR_TBL	141B	149B	151B	159B	161B	169B	171B	179B
29	TS0	TS0	141C	149C	151C	159C	161C	169C	171C	179C
30	TS1	TS1	141D	149D	151D	159D	161D	169D	171D	179D
31	OPR	OPR	141E	149E	151E	159E	161E	169E	171E	179E
32	TM0	TM0	141F	149F	151F	159F	161F	169F	171F	179F
33	TM1	TM1	1420	14A0	1520	15A0	1620	16A0	1720	17A0
34	STCG1	STCG1	1421	14A1	1521	15A1	1621	16A1	1721	17A1
35	TCG1	TCG1	1422	14A2	1522	15A2	1622	16A2	1722	17A2

8.3.1 Регистры общего назначения

8.3.1.1 Регистр WR

Формат регистра приведен на рисунке ниже.

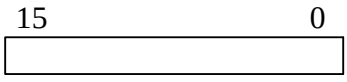


Рисунок 8.12- Формат регистра WR

WR (Work Register) ***(R/W)*** доступен по чтению и для побитной установки. Сброс происходит автоматически по завершению работы. Порядковый номер бита соответствует номеру канала и содержит информацию о работе канала:

0 - i-й канал остановлен;
1 - i-й канал работает.

8.3.1.2 Регистр DR

Формат регистра приведен на рисунке ниже.

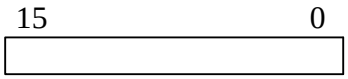


Рисунок 8.13- Формат регистра DR

DR (Depend Register) доступен по чтению, а также для побитной установки и сбросу. Порядковый номер бита соответствует номеру канала и содержит информацию о том яляется ли канал зависимым:

0 - i-й канал является зависимым от i-1 канала;
1 - i-й канал не зависим от других каналов.

8.3.1.3 Регистр INTCHR

Формат регистра приведен на рисунке ниже.

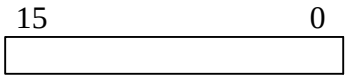


Рисунок 8.14- Формат регистра INTCHR

INTCHR (Interrupt Channel Register) доступен по чтению и для побитной установки и сброса. Порядковый номер бита соответствует номеру канала. Установка бита происходит по окончанию работы канала (бит регистра WR обнуляется)

0 - нет запроса на прерывание;
1 - есть запрос на прерывание.

8.3.1.4 Регистр MASKR

Формат регистра приведен на рисунке ниже.

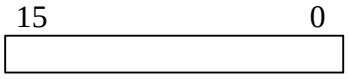


Рисунок 8.15 - Формат регистра MASKR

MASKR - регистр маски прерываний, доступен по чтению и записи. Порядковый номер бита соответствует номеру канала

0 - прерывание не маскируется;
1 - прерывание маскируется.

8.3.1.5 Регистр INTREQ

Формат регистра приведен на рисунке ниже.

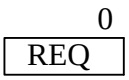


Рисунок 8.16 - Формат регистра INTREQ

Регистр запроса на прерывание для блока CORB, доступен по чтению и для сброса. Установка происходит, если установлен бит регистра INCHR и не происходит его маскирования, т.е. соответствующий разряд регистра MASKR равен нулю. Любая программная запись вызывает сброс прерывания, при этом если источник в регистре INTCHR не был сброшен, то в следующем такте прерывание опять возникнет.

8.3.1.6 Регистр ERR_REG

Формат регистра приведен на рисунке ниже.

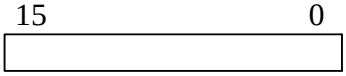


Рисунок 8.17 - Формат регистра ERR_REG

ERR_REG (Error Register) ***(R)*** доступен по чтению. Каждый бит регистра содержит информацию об ошибочном программировании соответствующего канала. Регистрируются ошибки запуска зависимых каналов и неправильного значения регистра UTR

0 – ошибка не обнаружена

1 – ошибка обнаружена, канал остановлен

Значение по умолчанию: 0

8.3.1.7 Регистр ERR_EN

Формат регистра приведен на рисунке ниже.

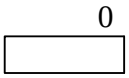


Рисунок 8.18 - Формат регистра ERR_EN

ERR_EN (Error Enable) ***(W)*** доступен по записи. Регистр разрешения работы схемы защиты от неправильного программирования.

0 – схема не работает

1 – схема работает

Значение по умолчанию: 0

8.3.2 Регистры канала для обращения за данными

8.3.2.1 Регистр CADR_DMA

Формат регистра приведен на рисунке ниже.

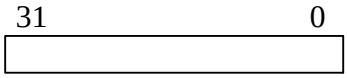


Рисунок 8.19 - Формат регистра CADR_DMAx

CADR_DMAx (Current Address Register) доступен по чтению и по записи. Регистр содержит текущий адрес для обращения к памяти за данными. Адресация осуществляется по карте памяти процессорных систем NMU (адрес выражается в 32-х разрядных словах) . После окончания работы канала данный регистр содержит адрес, следующий после последнего обращения. 0-й разряд всегда равен нулю. Данные должны быть выравнены по 64-х разрядной границе. У зависимых каналов данный регистр не используется.

8.3.2.2 Регистр CNTR_DMA

Формат регистра приведен на рисунке ниже.

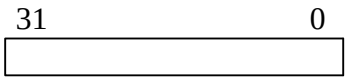


Рисунок 8.20 - Формат регистра CNTR_DMAx

CNTR_DMAx (Counter Register) доступен по чтению и по записи. Регистр содержит количество 64-х разрядных слов для канала обработки в прямом коде. 0- одно слово, 1 - два слова и т.д. При достижении нуля происходит остановка выборки данных и после обработки уже выбранных данных канал останавливается. Данный регистр работает и в случае зависимого канала.

8.3.2.3 Регистр UTR_DMA

Формат регистра приведен на рисунке ниже.

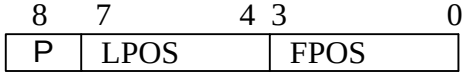


Рисунок 8.21 - Формат регистра UTR_DMAx

UTR_DMAx (Unpack Type Register) ***(R/W)*** доступен по чтению и по записи. Регистр содержит информацию для распаковки входных 64-х разрядных слов:

P - тип упаковки: 0 - 2+2 бита; 1 - 4+4 бита;

LPOS[4:0] - номер элемента в последнем 64-х разрядном слове, который будет обработан последним (0-первый элемент, 1-второй элемент и т.д.);

FPOS[4:0] - номер элемента в первом 64-х разрядном слове, с которого будет начата обработка (0-первый элемент, 1-второй элемент и т.д.).

8.3.3 Регистры канала для управления гетеродином

8.3.3.1 Регистр GET_FQ

Формат регистра приведен на рисунке ниже.

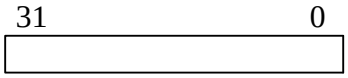


Рисунок 8.22 - Формат регистра GET_FQ

GET_FQ доступен по чтению и по записи. Регистр содержит значения частоты гетеродина.

8.3.3.2 Регистр GET_FT

Формат регистра приведен на рисунке ниже.

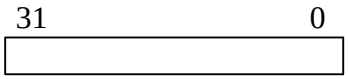


Рисунок 8.23 - Формат регистра GET_FT

GET_FT доступен по чтению и по записи. Регистр содержит текущее значение фазы. После остановки канала в регистре содержится следующее (еще не обработанное) значение фазы.

8.3.3.3 Регистр GET_CNTRL

Формат регистра приведен на рисунке ниже.

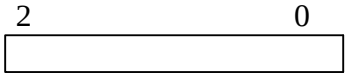


Рисунок 8.24 - Формат регистра GET_CNTRL

GET_CNTRL доступен по чтению и по записи. Регистр содержит настройку для выбора таблицы преобразования:

000b – sin/cos. Преобразование 6 старших разрядов гетеродина в 4 разряда sin/cos.

001b – sin=0,cos=1,

010b – sin=1,cos=0,

011b – sin=0,cos=0.

1xxb – sin/cos. Преобразование 8 старших разрядов гетеродина (после накопления) в 6 разрядов sin/cos.

100b – с округлением to_-inf (просто все отбрасывается);

101b – с округлением to_+inf (к результату прибавляется 1, если (R + S) > 0);

110b – с округлением to_nearest (к результату прибавляется 1, если R = 1);

111b – с округлением to_nearest_even (к результату прибавляется 1, если R & (L + S) > 0).

(L - последний бит округляемого, R - первый отбрасываемый бит, S остальные отбрасываемые биты)

8.3.4 Регистры канала для управления генератором времени универсального генератора псевдослучайных последовательностей

8.3.4.1 Регистр TG_FQ

Формат регистра приведен на рисунке ниже.

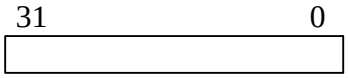


Рисунок 8.25 - Формат регистра TG_FQ

TG_FQ доступен по чтению и по записи. Регистр содержит значения частоты генератора блока TG.

8.3.4.2 Регистр TG_FT

Формат регистра приведен на рисунке ниже.

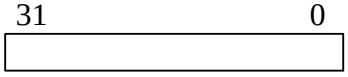


Рисунок 8.26 - Формат регистра TG_FT

TG_FT доступен по чтению и по записи. Регистр содержит текущее значение фазы. После остановки канала в регистре содержится следующее (еще не обработанное) значение фазы.

8.3.4.3 Регистр BOCR

Формат регистра приведен на рисунке ниже.

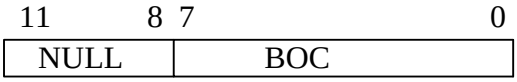


Рисунок 8.27 - Формат регистра BOCR

BOCR доступен по чтению и по записи. Регистр отвечает за BOC-модуляцию и обнуление части входных данных при кодовом уплотнении.

Поле BOC:

0000_0000 – BOC не работает

xxxx_xxx1 – BOC работает, инверсия в интервале фазы [0 : 2π/8) радиан;

xxxx_xx1x – BOC работает, инверсия в интервале фазы [2π/8 : 2π*2/8) радиан;

xxxx_x1xx – BOC работает, инверсия в интервале фазы [2π*2/8 : 2π*3/8) радиан;

xxxx_1xxx – BOC работает, инверсия в интервале фазы [2π*3/8 : 2π*4/8) радиан;

xxx1_xxxx – BOC работает, инверсия в интервале фазы [2π*4/8 : 2π*5/8) радиан;

xx1x_xxxx – BOC работает, инверсия в интервале фазы [2π*5/8 : 2π*6/8) радиан;

x1xx_xxxx – BOC работает, инверсия в интервале фазы [2π*6/8 : 2π*7/8) радиан;

1xxx_xxxx – BOC работает, инверсия в интервале фазы [2π*7/8 : 2π*8/8) радиан.

Поле NULL (используется при уплотнении кода):

0000 – обнуления нет

xxx1 – обнуление в интервале фазы [0 : 2π/4) радиан;

xx1x – обнуление в интервале фазы [2π/4: 2π*2/4) радиан;

x1xx – обнуление в интервале фазы [2π*2/4: 2π*3/4) радиан;

1xxx – обнуление в интервале фазы [2π*3/4: 2π*4/4) радиан.

8.3.4.4 Регистр SHCR

Формат регистра приведен на рисунке ниже.

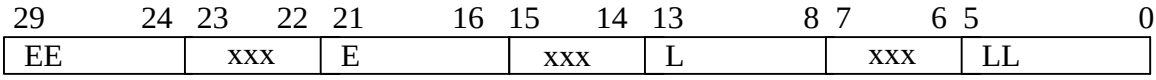


Рисунок 8.28 - Формат регистра SHCR

SHCR доступен по чтению и по записи. Регистр содержит даные о смещении опережающих и отстающих ПСП относительно основной.

Поля LL[5:0], L[5:0] определяют значение фазы, на которое будет отставать сигнал LL, L:

000000 - значение LL(L) совпадает с P;

000001b - LL (L) отстает от P на 2π/64 радиан;

000010b - LL (L) отстает от P на 2π*2/64 радиан;

.....

111111b - LL (L) отстает от P на 2π*63/64 радиан;

Поля EE[5:0], E[5:0] определяют значение фазы, на которое будет опережать сигнал EE, E:

000000 - значение EE(E) совпадает с P;

000001b - EE (E) опережает P на 2π/64 радиан;

000010b - EE (E) опережает P на 2π*2/64 радиан;

.....

111111b - EE (E) опережает P на 2π*63/64 радиан;

8.3.4.5 Регистр CNTRL_gen

Формат регистра приведен на рисунке ниже

Рисунок 8.29 - Формат регистра CNTRL_gen

Регистр доступен по чтению и записи, определяет работу универсального генератора псевдослучайных последовательностей.

TPL[1:0] определяет работу генератора

0x - генератор не работает, на выходе 0;

10 - используется генератор на сдвиговых регистрах

11 - используется табличный генератор;

EPL[2:0], EPL_TYPE[1:0] - определяет тип запуска генератора:

EPL, 00 - полная программная загрузка EPL;

PLx, 01 - программная загрузка PL; вычисление E

Lxx, 10 - программная загрузка L; вычисление EP

xxx 11 - нет программной загрузки; вычисление EPL.

При чтении регистра поле EPL содержит текущее значение EPL[2:0] в регистре сдвига, EPL_TYPE[1:0] содержит информацию о текущих вычислениях:

00- генератор будет вычислять следующее значение после E,

01- генератор будет вычислять значение E;

10- генератор будет вычислять значение EP;

11- генератор будет вычислять значение EPL;

Пояснение про запуск генератора псевдослучайных последовательностей. Генератор псевдослучайных последовательностей может вычислять только следующие значения и не может вычислять предыдущие значения. Но, например, когда используется L составляющая необходимо получить предыдущее значение. В данном случае можно настроить запуск генератора с учетом этого и запустить с предыдущего значения (EPL_TYPE[1:0] = 2'b11) или прописать предыдущее значение L программно (поле EPL[2:0] = 3'bLxx). Аналогично со значением P и L. При работе генератора происходит вычисление сдедущего значения после текущего (необходимо для E составляющей). Таким образом, после остановки генератора значение регистров состояния (TG_FT, CADR_TBL и т.д) уже указывают на ячейку, которая будет дальше чем P значение. Чтобы генератор можно было запустить с этого же места необходимо считать регистр CNTRL_gen вместе с регистрами состояния и при следующем щапуске прописать регистр CNTRL_gen считанным ранее значением.

8.3.5 Регистры канала для табличного формирования ПСП

8.3.5.1 Регистр FADR_TBL

Формат регистра приведен на рисунке ниже.

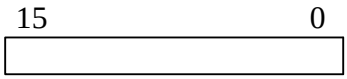


Рисунок 8.30 - Формат регистра FADR_TBL

FADR_TBL доступен по чтению и по записи. Регистр содержит адрес начала таблицы. Адресация битовая.

8.3.5.2 Регистр LADR_TBL

Формат регистра приведен на рисунке ниже.

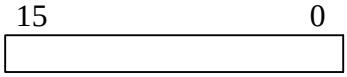


Рисунок 8.31 - Формат регистра LADR_TBL

LADR_TBL доступен по чтению и по записи. Регистр содержит адрес конца таблицы. Адресация битовая.

8.3.5.3 Регистр CADR_TBL

Формат регистра приведен на рисунке ниже.

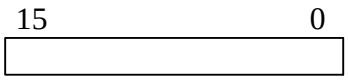


Рисунок 8.32 - Формат регистра CADR_TBL

CADR_TBL доступен по чтению и по записи. Регистр содержит текущий адрес элемента таблицы. Адресация битовая.

8.3.6 Регистры канала для формирования ПСП на сдвиговых регистрах

8.3.6.1 Регистр TCG0, TCG1

Формат регистра приведен на рисунке ниже.

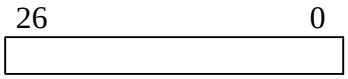


Рисунок 8.33 - Формат регистра TCG0, TCG1

TCGx [26:0] – 27-разрядный программно доступный на чтение и запись регистр, хранящий текущее значение счётчика длины последовательности TCG [26:0]. TCG0 управляет сдвиговыми регистрами TIS0,TS0; TCG1 - TIS1,TS1

8.3.6.2 Регистр STCG0, STCG1

Формат регистра приведен на рисунке ниже.

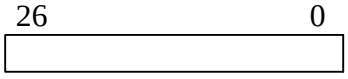


Рисунок 8.34 - Формат регистра STCG0, STCG1

STCGx [26:0] – 27-разрядный программно доступный на чтение и запись регистр, хранящий период длины последовательности при использовании счетчика длины. Для того чтобы задать период равный N, в регистр STCGx необходимо записать число -N в дополнительном коде.

8.3.6.3 Регистр TM0, TM1

Формат регистра приведен на рисунке ниже.

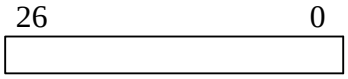


Рисунок 8.35 - Формат регистра TMx

TMx[26:0] – 27-разрядный программно-доступный на запись и чтение регистр маски обратной связи.

Если i-й разряд 1 - то данный разряд будет учитываться в обратной связи; 0- разряд будет заблокирован.

8.3.6.4 Регистр TIS1[26:0], TIS0[26:0]

Формат регистра приведен на рисунке ниже.

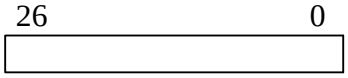


Рисунок 8.36 - Формат TIS1, TIS0

TISx[26:0] – 27-разрядный программно доступный на запись и чтение регистр начального значения псевдослучайной последовательности. Ипользуется при работе счетчика длины последовательности.

8.3.6.5 Регистр TS1[26:0], TS0[26:0]

Формат регистра приведен на рисунке ниже.

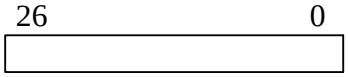


Рисунок 8.37 - Формат TS1, TS0

TSx[26:0] – 27-разрядный программно доступный на запись и чтение сдвиговый регистр регистр псевдослучайной последовательности.

8.3.6.6 Регистр OPR

Формат регистра приведен на рисунке ниже.

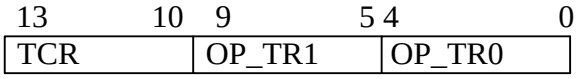


Рисунок 8.38 - Формат OPR

Регистр OPR доступен по чтению и по записи. Регистр формирует управление сдвиговым регистром:

OP_TR0[4:0] - определяет номер элемента сдвигового регистра TS0, с которого будет сформирован выход; 0 - выход с элемента TS0[0], 1- выход с элемента TS0[1] и т..д.

OP_TR1[4:0] - определяет номер элемента сдвигового регистра TS1, с которого будет сформирован выход.

TCR[0] - определяет обратную связь регистра TS0

TCR[1] - определяет обратную связь регистра TS1

0 - не инвертировать обратную связь;

1 - инвертировать обратную связь.

TCR[2] - использовать счетчик длины последовательности или нет для TS0:

0 - счетчик длины последовательности не используется;

1 - счетчик длины последовательности используется;

TCR[3] - использовать счетчик длины последовательности или нет для TS1:

0 - счетчик длины последовательности не используется;

1 - счетчик длины последовательности используется;

8.3.7 Регистры канала, содержащие результат корреляции P_RE, P_IM, E_RE, E_IM, EE_RE, EE_IM, L_RE, L_IM, LL_RE, LL_IM

Формат регистров приведен на рисунке ниже.

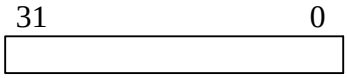


Рисунок 8.39 - Формат регистра P_RE, P_IM, ...

P_RE, E_RE, EE_RE, L_RE, LL_RE регистры доступны по чтению и сбросу. Регистры содержат результаты корреляции действительной составляющей для текущей, опережающих и запаздывающих последовательностей.

P_IM, E_IM, EE_IM, L_IM, LL_IM регистры доступны по чтению и сбросу. Регистры содержат результаты корреляции мнимой составляющей для текущей, опережающих и запаздывающих последовательностей.

Запись в любой из этих регистров вызывает сброс всех регистров результата корреляции канала.

9 Интервальный таймер для формирования временных интервалов и синхронизации с внешним событием (ITU)

Блок **ITU** предназначен для формирования временных интервалов (основного и дополнительного) и импульсов с заданной задержкой и длительностью (программируемый генератор импульсов). Данный блок работает на тактовой частоте ACLK.

9.1 Принцип работы блока ITU

Структурная схема блока интервальных таймеров приведена ниже (см. Рисунок 9.1).

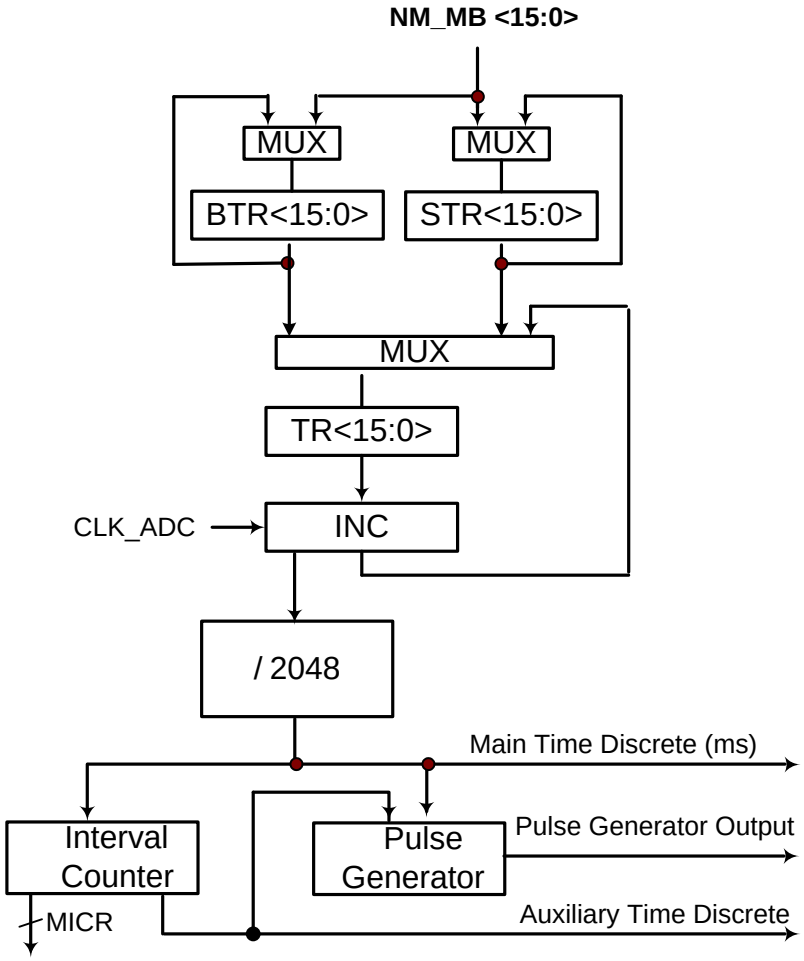


Рисунок 9.1 - Структурная схема блока ITU

Величина временного интервала программно задается в 16-разрядном теневом регистре **STR** в дополнительном коде. Каждый такт содержимое рабочего регистра **TR** инкрементируется, когда значение **TR** достигнет нуля, выдается импульс на делитель (**/2048**) и происходит перезапись значения из регистра **STR** в регистр **TR**. Исключение составляет только первая перезапись после формирования основного временного интервала. При первой перезаписи происходит запись регситра **BTR** в регистр **TR**.

/2048 – делитель, предназначенный для деления частоты поступающих на него импульсов в 2048 раза. На выходе делителя формируется основной временной интервал (**Main Time Discr**

ete). Таким образом,

Main Time Discrete = Tbtr + Tstr * 2047.

При формировании основного временного интервала формируется сигнал запроса на прерывание по основному временному интервалу INT_MS_ITU, который поступает на входы контроллеров внешних прерываний процессорных систем NMU0, NMU1 и ARMU.

Сигнал основного интервала времени подается на интервальный счетчик **Interval Counter**. Структурная схема интервального счетчика представлена на рисунке ниже (см. Рисунок 9.2).

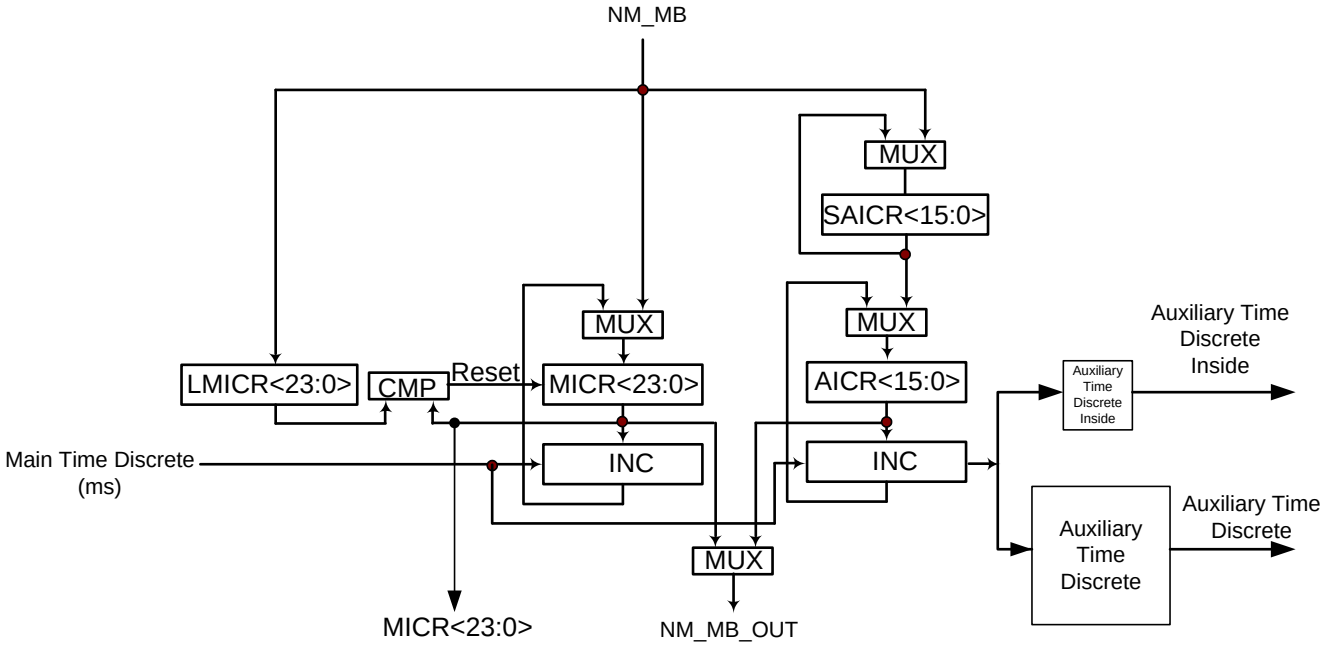


Рисунок 9.2 - Структурная схема интервального счетчика

Интервальный счетчик состоит из основного и вспомогательного. Основной счетчик осуществляет циклический счет основных временных интервалов. Его регистр **MICR (Main Interval Count Register)** доступен программисту по чтению и по записи. Содержимое регистра **MICR** также записывается в заголовок пакета данных при изменении состояния вывода **Main Time Discrete (ms)** во время работы каналов ПДП “DDC - память”. По сигналу системного сброса счетчик принимает значение 000000h. Другой программно-доступный регистр - **LMICR (Last Main Interval Count Register)** – служит для обнуления **MICR** при достижении им заданного значения. После системного сброса содержимое **LMICR** равно всем единицам (FFFFFFh), и данное значение не влияет на работу **MICR**.

Дополнительный счетчик осуществляет формирование дополнительного временного интервала – **Auxiliary Time Discrete** для вторичной обработки информации, например 10мс, 100мс или 1с. Счётчик содержит программно доступные регистры: на запись – теневой **SAICR (Shadow Auxiliary Interval Count Register)**, на чтение – рабочий регистр **AICR (Auxiliary Interval Count Register)**. Каждый основной интервал времени содержимое рабочего регистра **AICR** инкрементируется, когда значение **AICR** достигнет нуля, формируется однотактовый импульс **Auxiliary Time Discrete** для подачи на программируемый генератор импульсов и 16-тактовый для подачи на внешний вывод микросхемы **ITO**. При формировании дополнительного временного интервала формируется сигнал запроса на прерывание по дополнительному временному интервалу INT_AUX_ITU, который поступает на входы контроллеров внешних прерываний процессорных систем NMU0, NMU1 и ARMU.

Для синхронизации нескольких микросхем по положительному фронту внешнего сигнала **ITSTROB** текущее состояние делителя (13 разрядов), регистра **TR** (16 разрядов) и регистра **MICR** (24 разряда) записываются в регистр **SPECR**, который программно доступен по чтению. Кроме того, в этот момент взводится бит **F** в регистре **CR (Condition Register),** который программно доступен на чтение и обнуляется после чтения регистра **SPECR**.

Сигналы основного и дополнительного интервалов времени поступают также на программируемый генератор импульсов. Основное назначение программируемого генератора импульсов заключается в формировании импульса заданной длительности, фронт которого с высокой точностью синхронизован с эталонной шкалой времени. Структурная схема генератора представлена ниже (см. Рисунок 9.3). Значение программно доступного регистра **SELR** определяет, какой сигнал будет использоваться для запуска генератора (основной или дополнительный временной интервал).

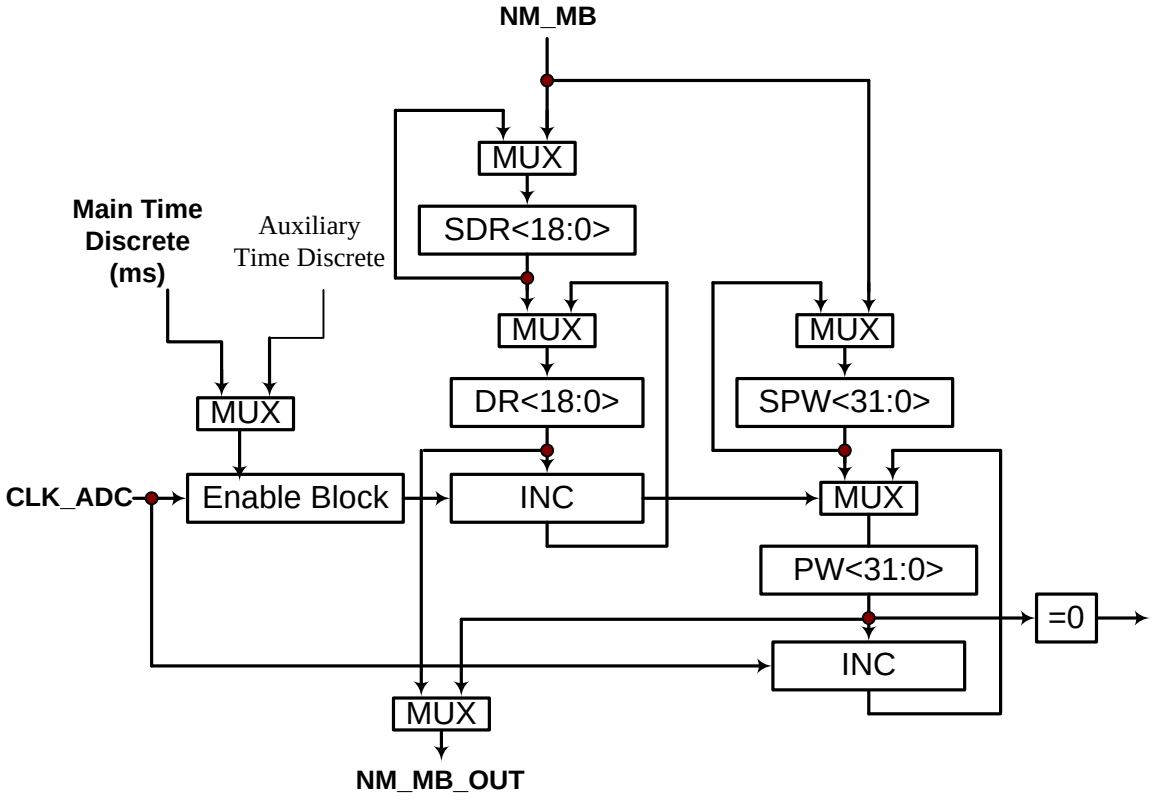


Рисунок 9.3- Структурная схема программируемого генератора импульсов

Для выдачи указанного импульса в программно доступный регистр **SDR (Shadow Delay Register)** записывается величина задержки в дополнительном коде. После запуска генератора содержимое теневого регистра **SDR** переписывается в регистр **DR**, после чего значение регистра **DR** инкрементируется до тех пор, пока **DR** не обнулится. После этого запускается схема формирования импульса заданной длительности. В её состав входят программно доступные регистры: на запись – теневой **SPW (Shadow Pulse Width)**, на чтение – рабочий регистр **PW (Pulse Width)**. Для формирования импульса нужной длительности в регистр **SPW** записывается величина длительности в дополнительном коде. После запуска схемы формирования импульса заданной длительности содержимое **SPW** переписывается в **PW**, и **PW** инкрементируется до тех пор, пока не сформируется перенос из старшего разряда, и после этого **PW** не обнулится. Схема формирования импульса заданной длительности формирует на выходе программируе-

мого генератора импульс до тех пор, пока содержимое **PW** не равно нулю. Выход программируемого генератора импульсов поступает на внешний вывод микросхемы.

Чтобы повторно сформировать импульс, необходимо осуществить запись в регистр **SDR**. Запись можно производить сразу после запуска генератора.

По началу выдачи сигнала импульса заданной длительности формируется запрос на прерывание INT_STRB_AUX, который подается на вход контроллеров внешних прерываний процессорных систем **NMU1**, **NMU2** и **ARMU**.

9.2 Конфигурационные регистры блока ITU

Управление блоком **ITU** может осуществляться процессорной системой **NMU1, NMU2** или процессорной системой ARMU. Чтение и запись конфигурационных регистров блока **ITU** производится скалярными командами ввода/вывода и равносильно чтению и записи в определенные ячейки памяти. Адреса регистров блока **ITU** для процессорной системы **NMU1** и **NMU2** совпадают и приведены ниже в таблице (Таблица 9.1).

Таблица 9.1 – Адреса регистров блока ITU в адресном пространстве NMU1 и NMU2

	источник	Адрес вектор в пространстве NMU, hex	Адрес вектор в пространстве ARMU, hex	Регистр доступен на	
				Чтение	Запись
1	STR	000F_0000	003C_0000		+
2	TR	000F_0000	003C_0000	+	
3	SAICR	000F_0002	003C_0008		+
4	AICR	000F_0004	003C_0010	+	
5	LMICR	000F_0006	003C_0018		+
6	MICR	000F_0008	003C_0020	+	+
7	SPECR_LO	000F_000A	003C_0028	+	
8	SPECR_HI	000F_000C	003C_0030	+	
9	CR	000F_000E	003C_0038	+	
10	SDR	000F_0010	003C_0040		+
11	DR	000F_0010	003C_0040	+	
12	SPW	000F_0012	003C_0048		+
13	PW	000F_0012	003C_0048	+	
14	BTR	000F_0014	003C_0050	+	+
15	SELR	000F_0016	003C_0058		+

Теневой регистр **STR** доступен на запись, на чтение доступен рабочий регистр TR. Регистр BTR доступен на чтение и запись. Формат регистров приведен ниже (см. Рисунок 9.4).

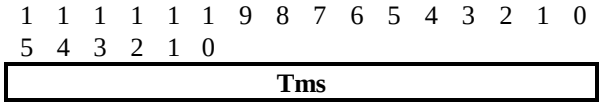


Рисунок 9.4 – Формат регистра STR, TR, BTR

Поле **Tms** (15-0 разряды) определяет период основного временного интервала, величина которого в тактах выражается формулой:

Main Time Discrete = Tbtr + Tstr * 2047.

Значение **Tbtr** и **Tstr** задается в дополнительном коде со знаком минус в поле **Tms**. Например, если надо сформировать основной интервал длительностью 2050 тактов, то **Tstr=1**, **Tbtr = 3**. Соответственно STR = -1 = FFFFh, BTR = -3 = FFFDh.

Регистры **SAICR** и **AICR** – теневой и рабочий регистры, определяющие период дополнительного временного интервала. Регистр **SAICR** доступен на запись, регистр **AICR** доступен на чтение. Формат регистров приведен ниже (см. Рисунок 9.5).

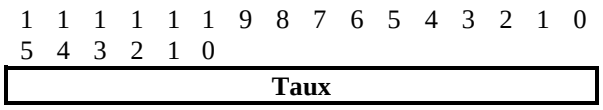


Рисунок 9.5 – Формат регистров SAICR и AICR

Поле **Taux** (15-0 разряды) определяет период дополнительного временного интервала. Период дополнительного временного интервала выражается в количестве основных временных интервалов. Количетсво задается в дополнительном коде со знаком минус в поле **Taux**.

После системного сброса значение поля Taux принимает значение FF9Ch, что соответствует периоду дополнительного временного интервала равного 100 периодам основного веременного интервала.

Регистр **MICR** – 24-х разрядный счетчик основных временных интервалов, доступен на запись и чтение. Регистр содержит текущий номер основного временного интервала. После системного сброса значение регистра равно нулю.

Регистр **LMICR** – 24-х разрядный регистр, определяющий сброс регистра **MICR,** доступен на запись. После системного сброса значение регистра имеет значение FFFFFFh.

Регистр **SPECR** состоит из двух частей старшей **SPECR_HI** и младшей **SPECR_LO**. Обе части доступны на чтение. Формат регистров SPECR_HI и SPECR_LO приведен ниже (см. Рисунок 9.6 Рисунок 9.7)

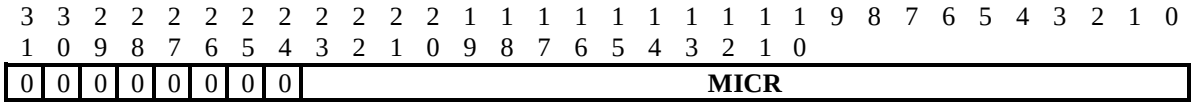


Рисунок 9.6 – Формат регистра SPECR_LO

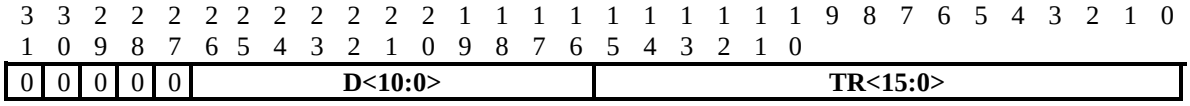


Рисунок 9.7 – Формат регистра SPECR_HI

Поле **MICR** содержит значение регистра **MICR** в момент прихода положительного фронта сигнала **ITSTROB**.

Поле **TR<15:0>** содержит значение регистра **TR** в момент прихода положительного фронта сигнала **ITSTROB**.

Поле **D<10:0>** содержит значение счетчика делителя в момент прихода положительного фронта сигнала **ITSTROB**.

Регистр состояния **CR (Condition Register)** доступен на чтение. Формат регистра приведен ниже (см. Рисунок 9.8).

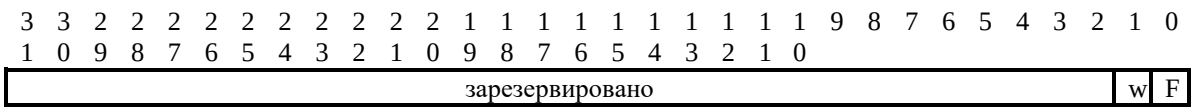


Рисунок 9.8 – Формат регистра CR

Бит **F** (0-й разряд) показывает, что производилась синхронизация с помощью сигнала **ITSTROB**. Бит устанавливается, когда приходит положительный фронт **ITSTROB**. Сброс происходит после чтения регистра **SPECR_HI** или **SPECR_LO**.

Бит **w** (1-й разряд) информирует о работе программируемого генератора импульсов. Бит устанавливается, когда происходит запись в регистр **SDR**, показывая, что генератор будет формировать импульс после прихода сигнала запуска. Сброс бита происходит после прихода запускающего импульса, показывая, что может быть произведена запись регистра **SDR** для следующего импульса.

Регистр **SELR** определяет, по какому сигналу будет запускаться программируемый генератор импульсов. Регистр доступен на запись. Формат регистра **SELR** приведен ниже (см. Рисунок 9.9).

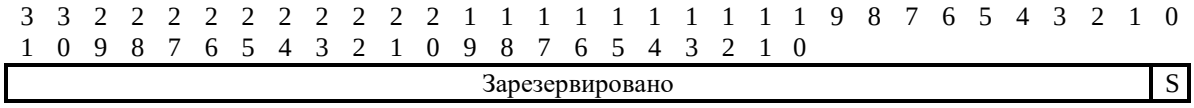


Рисунок 9.9 – Формат регистра SELR

Поле **S**:

0 – запускающий сигнал – сигнал основного временного интервала.

1 – запускающий сигнал – сигнал дополнительного временного интервала.

Регистры **SDR** и **DR** – теневой и рабочий регистры, определяющие задержку перед выдачей импульса программируемым генератором импульсов. Регистр **SDR** доступен на запись, регистр **DR** доступен на чтение. Формат регистров приведен ниже (см. Рисунок 9.10).

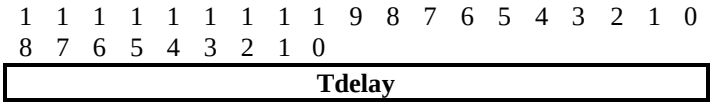


Рисунок 9.10 – Формат регистров SDR и DR

Поле **Tdelay** (15-0 разряды) определяет время задержки. Задержка задается в дополнительном коде. T = Tdelay+1.

Регистры **SPW** и **PW** – теневой и рабочий регистры, определяющие длительность импульса программируемого генератора импульсов. Регистр **SPW** доступен на запись, регистр **PW** доступен на чтение. Формат регистров приведен ниже (см. Рисунок 9.11).

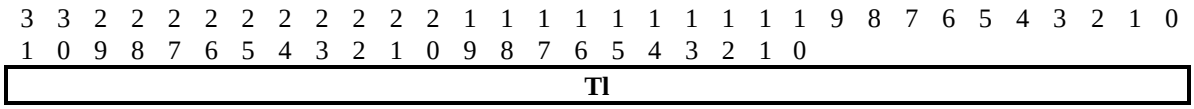


Рисунок 9.11 – Формат регистров SPW и PW

Поле **Tl** (15-0 разряды) определяет длительность импульса, выраженную в тактах ACLK. Необходимое значение записывается со знаком минус в дополнительном коде в поле **Tl**.

10 Генератор дальномерного кода для калибровки (CALB)

СБИС СНП-ВП содержит генератор дальномерного кода, работающий на тактовой частоте ACLK, для калибровки радио тракта. Данный генератор аналогичен генератору дальномерного кода в блоке корреляторов формирующему ПСП на сдвиговых регистрах. Отличие состоит в том, что нет формирования запаздывающих и опережающих последовательностей. Формируется только точная (P) последовательность, которая поступает на внешнией выводу микросхемы:

- TCOD выход псевдослучайной последовательности;
- TCOD_NULL признак обнуления значения используемого при кодовом уплотнении

10.1 Конфигурационные регистры блока CALB

Управление блоком **CALB** может осуществляться процессорной системой **NMU1, NMU2** или процессорной системой ARMU. Чтение и запись конфигурационных регистров блока **CALB** производится скалярными командами ввода/вывода и равносильно чтению и записи в определенные ячейки памяти. Адреса регистров блока для процессорной системы **NMU1** и **NMU2** совпадают и приведены ниже в таблице (Таблица 10.1).

Таблица 10.1 – Адреса регистров блока CALB в адресном пространстве NMU1,NMU2, ARMU

	источник	Адрес вектор в пространстве NMU, hex	Адрес вектор в пространстве ARMU, hex	Регистр доступен на	
				Чтение	Запись
1	START	000F_0018	003C_0060		+
2	STOP	000F_001A	003C_0068		+
3	STATUS	000F_0018 000F_001A	003C_0060 003C_0068	+	
4	FQ	000F_001C	003C_0070	+	+
5	FT	000F_001E	003C_0078	+	+
6	BOCR	000F_0020	003C_0080	+	+
7	OPR	000F_0022	003C_0088	+	+
8	STCG0	000F_0024	003C_0090	+	+
9	TCG0	000F_0026	003C_0098	+	+
10	STCG1	000F_0028	003C_00A0	+	+
11	TCG1	000F_002A	003C_00A8	+	+
12	TIS0	000F_002C	003C_00B0	+	+
13	TS0	000F_002E	003C_00B8	+	+
14	TIS1	000F_0030	003C_00C0	+	+
15	TS1	000F_0032	003C_00C8	+	+
16	TM0	000F_0034	003C_00D0	+	+
17	TM1	000F_0036	003C_00D8	+	+

Регистр **START** доступен на запись. Формат регистра приведен ниже (см. Рисунок 10.1).

31 … 1	0
Зарезервировано	St

Рисунок 10.1 – Формат регистра START

Запись единицы в поле St говорит о том, что по первой метке основного временного интервала будет запущен **CALB.** Чтение данного поля происходит при чтении регистра **STATUS**.

Регистр **STOP** доступен на запись. Формат регистра приведен ниже (см. Рисунок 10.2).

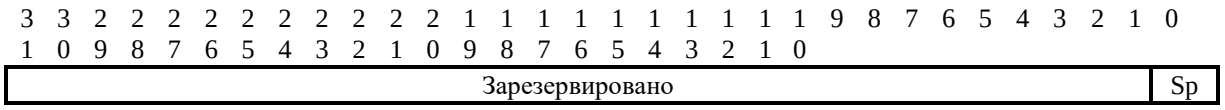


Рисунок 10.2 – Формат регистра STOP

Запись единицы в поле Sp говорит о том, что по первой метке основного временного интервала будет остановлен **CALB.** Чтение данного поля происходит при чтении регистра **STATUS**.

Регистр **STATUS** доступен на чтение. Формат регистра приведен ниже (см. Рисунок 10.3).

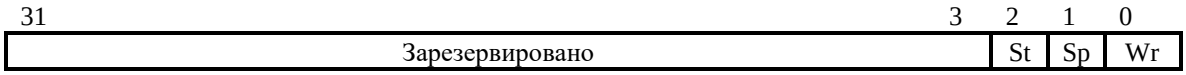


Рисунок 10.3 – Формат регистра STATUS

Поле **St** соответствует одноименному полю регистра **START**, Поле **Sp** соответствует одноименному полю регистра **STOP,** поле **WR** показывает состояние блока на момент чтения:

0 - блок CALB не работает;

1 - блок CALB работает.

11 Контроллер ПДП (DMAC)

Контроллер ПДП процессорных систем NMU1 и NMU2 обеспечивает прямой доступ в память блокам DDC, блокам коммуникационных портов, а также обеспечивает передачи между блоками памяти СБИС. Блок обрабатывает следующие каналы ПДП:

канал DDC-память (DW) – запись данных от DDC,
канал память-DDC (DR) – чтение данных для DDC,
канал память-CORB (COR) – чтение данных для CORB,
канал память-память (MM) – передачи память-память (2 виртуальных канала),
канал CP0-память (CP0WR) – запись данных от коммуникационного порта 0,
канал CP1-память (CP1WR) – запись данных от коммуникационного порта 1,
канал память-CP0 (CP0RD) – чтение данных для коммуникационного порта 0,
канал память-CP1 (CP1RD) – чтение данных для коммуникационного порта 1.

Блок доступен процессорным системам NMU1, NMU2 и ARMU как периферийное устройство. Контроллер имеет независимые интерфейсы передачи данных с каждым обслуживаемым устройством (см. Рисунок 11.1).

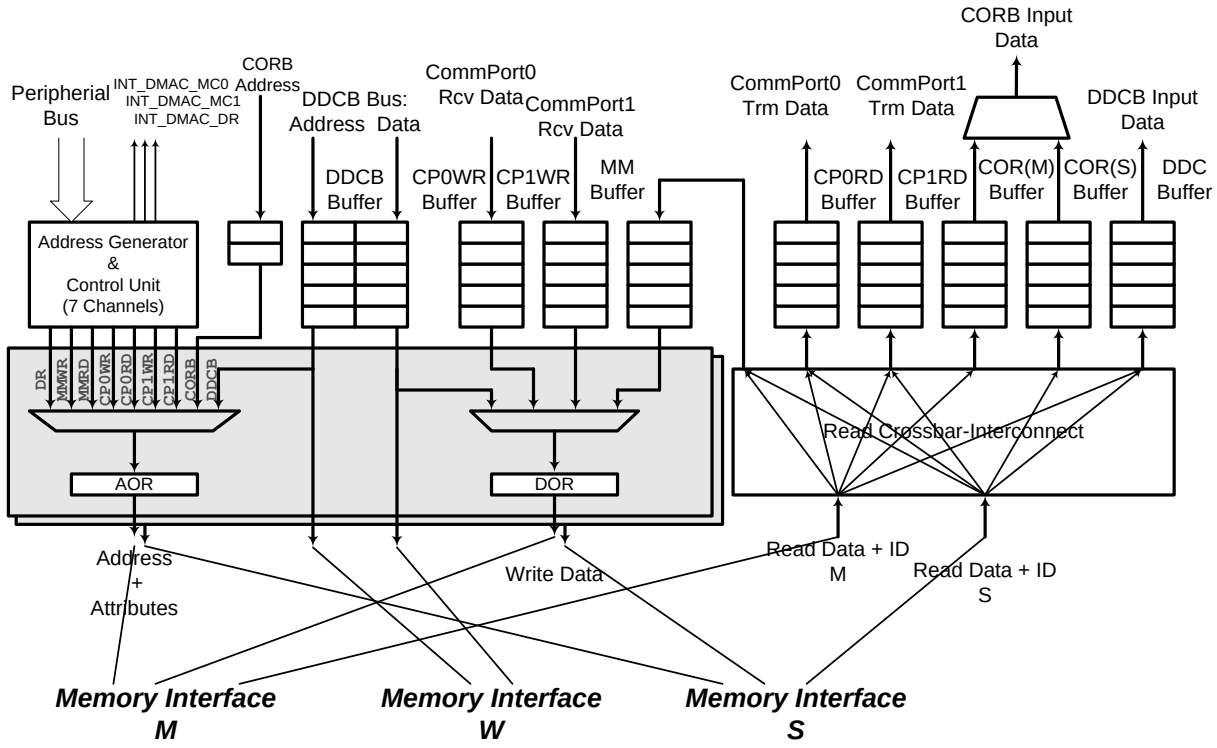


Рисунок 11.1 - Структурная схема контроллера ПДП

Имеются 3 интерфейса доступа в память (M, S и W), разделённые по адресному пространству: адресное пространство 0x10000000-0xFFFFFFFF доступно контроллеру ПДП через интерфейс M, адресное пространство 0x00000000-0x0FFFFFFF – интерфейсы S и W. Доступ в память производится 64-разрядными словами.

11.1 Канал DDC-память

Канал DDC-память, требующий наибольшей пропускной способности, имеет выделенный только для этого канала интерфейс W доступа в память процессорных систем NMU1 и NMU2 и

в блок общей памяти SMU. В интерфейсе M данный канал имеет фиксированный наивысший приоритет. Генерация адресов для записи в память и запуск канала производятся блоком DDC самостоятельно – контроллер ПДП не имеет программно доступных регистров, связанных с данным каналом.

11.2 Канал память-DDC

Канал память-DDC производит чтение данных для блока DDC. Источником данных могут быть любые блоки памяти, доступные для процессорных систем NMU1 и NMU2.

При обращении по интерфейсу M канал использует пакетные запросы (описано ниже).

11.3 Канал память-COR

Канал память-COR производит чтение данных для блока CORB. Генерация адресов для чтения из памяти и запуск канала производятся блоком CORB самостоятельно – контроллер ПДП не имеет программно доступных регистров, связанных с данным каналом. Контроллер производит чтение данных через интерфейсы M или S в зависимости от того, в какое адресное пространство попадает очередной адрес. Блок CORB может выдавать адреса из разных адресных пространств вперемешку, поэтому для временного хранения данных в канале COR имеются 2 буфера данных: COR(M) – для данных с шины M и COR(S) – для данных с шины S. Данные передаются блоку CORB в том порядке, в котором были выданы соответствующие адреса.

11.4 Канал память-память

Канал память-память работает следующим образом: производит чтение данных из памяти-источника, далее – по мере прихода данных в буфер – канал производит запись данных в память-приёмник. Источником и приёмником данных могут быть любые блоки памяти, доступные для процессорных систем NMU1 и NMU2.

Канал состоит из двух виртуальных каналов, каждый из которых имеет независимый набор программно доступных регистров. Если во время работы одного виртуального канала был запущен другой, то второй канал ожидает, когда первый канал будет программно переведён в нерабочее состояние (сброшен бит CPL).

При обращении по интерфейсу M канал использует пакетные запросы (описано ниже).

11.5 Каналы CP0-память, память-CP0, CP1-память и память-CP1

Данные каналы предназначены для обслуживания коммуникационных портов СБИС. В контроллере ПДП все 4 канала независимы, их функции:

CP0-память – приём данных с коммуникационного порта 0 и запись в память,

CP1-память – приём данных с коммуникационного порта 1 и запись в память,

память-CP0 – чтение из памяти и передача данных по коммуникационному порту 0,

память-CP1 – чтение из памяти и передача данных по коммуникационному порту 1.

Приёмный и передающий каналы одного коммуникационного порта могут запускаться независимо, при этом арбитраж шины производится интерфейсным блоком коммуникационного порта.

Блок управления питанием СБИС (PWRC) может инициировать запуск каналов CP0-память и CP1-память (независимо друг от друга). Данная функция используется для начальной

загрузки процессорных систем NMU1 и NMU2. Последовательность работы в процессе загрузки такова:

блок управления питанием делает запрос на запуск канала,

если соответствующий канал CPx-память работал, то схема дожидается, когда этот канал придёт в состояние бездействия (EN=0, CPL=0 или FSMCPxWR=0001),

канал CPx-память запускается со значениями регистров CPxWR_CNT=256 и CP0WR_ADDR=0x00200000 (нулевой адрес процессорной системы NMU1) и CP1WR_ADDR=0x00280000 (нулевой адрес процессорной системы NMU2), а также со снятой маской прерывания INT_DMAC_MCx (при этом устанавливается значение 0 в регистре CPxWR_MASK),

канал CPx-память принимает данные с интерфейса коммуникационного порта и записывает в память,

в момент записи последнего данного канал CPx-память выдаёт блоку управления питанием подтверждение загрузки и устанавливает запрос на прерывание INT_DMAC_MCx,

канал CPx-память снимает сигналы подтверждения и запроса на прерывание и переводит канал в состояние бездействия.

Далее описанная процедура может повторяться произвольное количество раз.

11.6 Пакетные запросы

Особенностью каналов память-память и память-DDC является использование в интерфейсе M пакетных запросов (как чтения, так и записи). Размер пакета – всегда 4 64-разрядных слова, причём адрес всегда выровнен, то есть первый запрос в пакете всегда имеет адрес, кратный 8 (дополнительный множитель 2 – за счёт того, что в одном слове – 2 адресуемых 32-разрядных данных). Если адрес и размер передачи не выровнены под размер пакета, то канал использует одиночные транзакции в начале и в конце передачи. Для эффективной реализации пакетной передачи в транзакцию шины добавлена фаза арбитража: когда канал собирается выдать пакет запросов, блок ПДП выставляет сигнал псевдозапроса. Псевдозапрос резервирует в шинном коммутаторе место под пакет данных, что приводит к замедлению доступа процессорных ядер NMC к адресному пространству 0x10000000-0xFFFFFFFF. Выдачу псевдозапросов можно отключить программно в регистре CORE_STATUS блока ПДП.

11.7 Программная модель контроллера ПДП

Каналы ПДП настраиваются с помощью программно доступных регистров. Регистры контроллера расположены в адресном пространстве процессорных систем NMU1 и NMU2, начиная с базового адреса 0x000E0000, в адресном пространстве ARMU – с адреса 0x00380000.

Каждый канал управляется своим набором регистров, который отображается в адресном пространстве как выровненный блок из 16 32-разрядных слов, структура этого набора – общая для всех каналов. Набор регистров канала состоит из:

регистра управления (CSR),

регистра-счётчика данных (CNT),

регистра начального адреса для чтения (в каналах, производящих чтение),

регистра начального адреса для записи (в каналах, производящих запись),

регистра маски прерывания.

Для канала память-память имеется 2 набора регистров (2 виртуальных канала). Благодаря этому 2 независимых процесса (на разных ядрах системы) могут совместно использовать канал ПДП память-память, не проводя между собой арбитража. Каждый процесс может иметь свой независимый программный интерфейс. Физический канал делится между двумя запусками по

очереди. Разница между наборами только в генерации запроса на прерывание. Виртуальный канал 0 может генерировать прерывание только процессорной системе NMU1, виртуальный канал 1 – только NMU2.

Не указанные в таблице адреса зарезервированы – запись по зарезервированным адресам никак не влияет на устройство, при чтении выдаётся неспецифицированное значение. Неиспользованные старшие разряды имеющихся регистров при чтении возвращают 0.

Таблица 11.1 - Программная модель блока ПДП

Название регистра	Адрес в пространстве NMU	Адрес в пространстве ARMU	Разрядность	Доступ
Регистры канала память-DDC				
D_CSR	0x000E0000	0x00380000	2	ЧТ/ЗП
D_MASK	0x000E0002	0x00380008	1	ЗП
D_CNT	0x000E0004	0x00380010	16	ЗП
D_RDADDR	0x000E0008	0x00380020	32	ЗП
Регистры канала память-память (виртуальный канал 0)				
MM0_CSR	0x000E0010	0x00380040	2	ЧТ/ЗП
MM0_MASK	0x000E0012	0x00380048	1	ЗП
MM0_CNT	0x000E0014	0x00380050	1	ЗП
MM0_RDADDR	0x000E0018	0x00380058	1	ЗП
MM0_WRADDR	0x000E001A	0x00380060	1	ЗП
Регистры канала память-память (виртуальный канал 1)				
MM1_CSR	0x000E0020	0x00380080	2	ЧТ/ЗП
MM1_MASK	0x000E0022	0x00380088	1	ЗП
MM1_CNT	0x000E0024	0x00380090	32	ЗП
MM1_RDADDR	0x000E0028	0x00380098	32	ЗП
MM1_WRADDR	0x000E002A	0x003800A8	32	ЗП
Регистры прёмного канала коммуникационного порта 0				
CP0WR_CSR	0x000E0030	0x003800C0	2	ЧТ/ЗП
CP0WR_MASK	0x000E0032	0x003800C8	1	ЗП
CP0WR_CNT	0x000E0034	0x003800D0	32	ЗП
CP0WR_WRADDR	0x000E003A	0x003800E8	32	ЗП
Регистры передающего канала коммуникационного порта 0				
CP0RD_CSR	0x000E0040	0x00380100	6	ЧТ/ЗП
CP0RD_MASK	0x000E0042	0x00380108	1	ЗП
CP0RD_CNT	0x000E0044	0x00380110	32	ЗП
CP0RD_RDADDR	0x000E0048	0x00380120	32	ЗП
Регистры прёмного канала коммуникационного порта 1				
CP1WR_CSR	0x000E0050	0x00380140	2	ЧТ/ЗП
CP1WR_MASK	0x000E0052	0x00380148	1	ЗП
CP1WR_CNT	0x000E0054	0x00380150	32	ЗП
CP1WR_WRADDR	0x000E005A	0x00380168	32	ЗП
Регистры передающего канала коммуникационного порта 1				
CP1RD_CSR	0x000E0060	0x00380180	6	ЧТ/ЗП
CP1RD_MASK	0x000E0062	0x00380188	1	ЗП
CP1RD_CNT	0x000E0064	0x00380190	32	ЗП
CP1RD_RDADDR	0x000E0068	0x003801A0	32	ЗП
Регистр состояния				
CORE_STATUS	0x000E0070	0x003801C0	32	ЧТ/ЗП

11.7.1 Регистр управления каналом (CSR)

Биты EN и CPL отображают состояние канала:
EN=0, CPL=0 – нерабочее состояние – канал не запущен,
EN=1, CPL=0 – канал работает,
EN=0, CPL=1 – канал завершил работу и ожидает перевода в нерабочее состояние,
EN=1, CPL=1 – такое состояние невозможно.

После аппаратного сброса канал находится в нерабочем состоянии EN=0, CPL=0. Запись в другие регистры канала разрешена только в нерабочем состоянии. Запуск канала переводит его в состояние EN=1, CPL=0. Когда завершается пердача данных, канал переходит в состояение EN=0, CPL=1 и остаётся в нём. Возврат канала в нерабочее состояние следует производить программно.

Виртуальные каналы память-память могут запускаться независимо, но работают на единственном физическом канале по очереди. Канал, в котором запись в бит EN=1 произошла раньше, запускается первым. Второй канал в этом случае ждёт полного завершения работы первого канала и начнёт передачу своих данных, только когда первый канал будет переведён в нерабочее состояние EN=0, CPL=0. Поля MM1ACTV, RQMM0 и RQMM1 регистра CORE_STATUS показывают, какой канал работает в данный момент, а какой канал находится в режиме ожидания.

Бит EN (0-й разряд) – бит запуска канала. Запись значения 1 запускает канал (только если он находился в состоянии EN=0, CPL=0), при чтении возвращается статус канала: 0 – канал не работает, 1 – канал работает.

Бит CPL (1-й разряд) – бит завершения работы канала. Запись значения 1 сбрасывает бит CPL и снимает запрос на прерывание по завершению работы данного канала; при чтении выдаётся статус: 1 – канал завершил работу, 0 – канал ещё не завершил работу или не был запущен.

Запись одновременно EN=1 и CPL=1 не допускается.

Поле RATE (5-2 разряды) – управляет скоростью работы в передающем канале коммуникационного порта: доступно для записи только в передающих каналах коммуникационных портов (память-CP0, память-CP1). Данное поле осуществляет деление тактового сигнала CLK1 блока CRGS (см. 5.4) при формировании сигнала CxSTRBn и выходных данных. В таблице ниже приведены допустимые скорости обмена.

Таблица 11.2 - Скорость передачи коммуникационного порта

значение поля RATE	**Частота передачи, Мбайт/с (значение при CLK1=320 МГц)**	**максимальная частота сигнала CxSTRBn (значение при CLK1=320 МГц)**
0000	**запрещенное значение**	
0001	CLK1/2 (160)	CLK1/4 (80)
0010	CLK1/3 (106,66)	CLK1/6 (53,33)
0011	CLK1/4 (80)	CLK1/8 (40)
0100	CLK1/5 (64)	CLK1/10 (32)
0101	CLK1/6 (53,33)	CLK1/12 (26,67)
0110	CLK1/7 (45,71)	CLK1/14 (22,86)
0111	CLK1/8 (40)	CLK1/16 (20)
1000	CLK1/9 (35,56)	CLK1/18 (17,78)
1001	CLK1/10 (32)	CLK1/20 (16)
1010	CLK1/11 (29,09)	CLK1/22 (14,54)
1011	CLK1/12 (26,67)	CLK1/24 (13,83)
1100	CLK1/13 (24,61)	CLK1/26 (12,3)
1101	CLK1/14 (22,86)	CLK1/28 (11,43)
1110	CLK1/15 (21.33)	CLK1/30 (10,67)
1111	CLK1/16 (20)	CLK1/32 (10)

Принимающие каналы коммуникационных портов не имеют настройки скорости, так как приём ведётся на скорости передатчика. ***При этом частота тактового сигнала CLK1 блока CRGS должна быть больше частоты сигнала CxSTRBn.***

Бит CDC (6-й разряд) – режим работы двунаправленных буферов при передаче шины коммуникационного порта. Значение 0 – унаследованный режим (от микросхем с процессором NeuroMatrix Core). Если две микросхемы, подключенные по коммуникационному порту, рабо-

тают в унаследованном режиме, то короткое время буферы сигналов XCSTRB в обоих микросхемах работают на выход. Значение 1 – режим с высокоимпедансным состоянием. Если две микросхемы, подключенные по коммуникационному порту, работают в режиме с высокоимпедансным состоянием, то короткое время буферы сигналов XCSTRB в обеих микросхемах работают на вход.

Бит CFGEN (7-й разряд) – разрешение записи в биты RATE и CDC. Для того чтобы при записи в регистр значение полей RATE и CDC обновлялось, необходимо в 7-й разряд записывать значение 1.

11.7.2 Регистр маски запроса на прерывание (MASK)

Бит MASK (0-й разряд) – маска запроса на прерывание. Запись значения 1 маскирует запрос на прерывание от данного канала, запись значения 0 – разрешает генерацию запроса на прерывание от данного канала.

Если канал CPx-память исполняет процесс загрузки, то бит MASK в регистре CPxWR_MASK сбрасывается в 0.

11.7.3 Регистр-счётчик данных (CNT)

В регистр CNT записывается размер передачи в 64-разрядных словах. Чтобы задать число слов для передачи в канале, равное N, необходимо записать в регистр CNT число (-1 * N) в дополнительном коде. Единице соответствует код FFFF_FFFFh, числу 2^{32} соответствует код 0000_0000h.

Не допускается настройка счётчика в каналах память-память (MMx_CNT) и в канале память-DDC (D_CNT) на передачу слов в количестве меньшем 5, если хотя бы один из адресов MMx_WRADDR или MMx_RDADDR принадлежит пространству 0x10000000-0xFFFFFFFF.

11.7.4 Регистр начального адреса чтения (RDADDR)

Запись в регистр RDADDR задаёт начальный (наименьший) адрес передаваемого массива. Данный регистр имеется только у каналов, осуществляющих чтение из памяти: память-DDC, память-память, память-CP0, память-CP1. В остальных каналах на месте данного регистра в структуре находится зарезервированный адрес.

Адрес задаётся в 32-разрядных словах (адрес в формате NMU), младший разряд адреса должен быть равен 0.

11.7.5 Регистр начального адреса записи (WRADDR)

Запись в регистр WRADDR задаёт начальный (наименьший) адрес записываемого в память массива. Данный регистр имеется только у каналов, осуществляющих запись в память: память-память, CP0-память, CP1-память. В остальных каналах на месте данного регистра в структуре находится зарезервированный адрес, запись в который любого значения никак не влияет на устройство.

Адрес задаётся в 32-разрядных словах (адрес в формате NMU), младший разряд адреса должен быть равен 0.

11.7.6 Регистр состояния блока ПДП (CORE_STATUS)

Регистр состояния блока ПДП показывает состояния каналов блока ПДП, а также статус виртуальных каналов (текущий активный виртуальный канал). Регистр CORE_STATUS также содержит биты отключения псевдозапросов в интерфейсе M.

Бит WRPRI (0-й разряд) – включает механизм псевдозапросов (1 – включено, 0 – выключено) при записи через интерфейс М. Доступен на чтение и запись.

Бит RDPRI (1-й разряд) – включает механизм псевдозапросов (1 – включено, 0 – выключено) при чтении через интерфейс М. Доступен на чтение и запись.

Поле FSMD (разряды 5-2) – состояние канала память-DDC: 0001 – не работает, 0010 – производит чтение и передачу данных, 0100 – все чтения завершены, производит только передачу данных, 1000 – работа завершена, ожидание сброса.

Поле FSMMM (разряды 9-6) – состояние физического канала память-память: 0001 – не работает, 0010 – производит чтение и запись, 0100 – все чтения завершены, производит только запись, 1000 – работа завершена, ожидание сброса.

Поле FSMCP0WR (разряды 13-10) – состояние принимающего канала коммуникационного порта 0: 0001 – не работает, 0010 – производит запись, 1000 – работа завершена, ожидание сброса.

Поле FSMCP0RD (разряды 17-14) – состояние передающего канала коммуникационного порта 0: 0001 – не работает, 0010 – производит чтение и передачу, 0100 – все чтения завершены, производит только передачу, 1000 – работа завершена, ожидание сброса.

Поле FSMCP1WR (разряды 21-18) – состояние принимающего канала коммуникационного порта 1: 0001 – не работает, 0010 – производит запись, 1000 – работа завершена, ожидание сброса.

Поле FSMCP1RD (разряды 25-22) – состояние передающего канала коммуникационного порта 1: 0001 – не работает, 0010 – производит чтение и передачу, 0100 – все чтения завершены, производит только передачу, 1000 – работа завершена, ожидание сброса.

Бит MM1ACTV (26-й разряд) – текущий работающий виртуальный канал в канале память-память: 1 – канал MM1, 0 – канал MM0. Если ни один из виртуальных каналов не работает (FSMMM=0001), то значение данного бита не определено.

Бит RQMM0 (27-й разряд) – показывает, что виртуальный канал MM0 ожидает своей очереди запуска на физическом канале: 1 – виртуальный канал ожидает своей очереди, 0 – не ожидает или работает.

Бит RQMM1 (28-й разряд) – показывает, что виртуальный канал MM1 ожидает своей очереди запуска на физическом канале: 1 – виртуальный канал ожидает своей очереди, 0 – не ожидает или работает.

Бит BOOT0 (29-й разряд) – показывает, что канал CP0-память выполняет процесс загрузки.

Бит BOOT1 (30-й разряд) – показывает, что канал CP1-память выполняет процесс загрузки.

Бит ACTV (31-й разряд) – показывает, что в контроллере ПДП выполняется какая-либо операция, значение 0 означает готовность контроллера к отключению тактового сигнала.

11.8 Прерывания

Блок ПДП может выдавать запросы на прерывание процессорным системам NMU1, NMU2 и ARMU:

INT_DMAC_DR – запрос на прерывание по завершению работы канала память-DDC,

INT_DMAC_MC0 – запрос на прерывание по завершению работы каналов память-память (виртуальный канал 0), память-CP0, CP0-память,

INT_DMAC_MC1 – запрос на прерывание по завершению работы каналов память-память (виртуальный канал 1), память-CP1, CP1-память.

Запрос INT_DMAC_DR отправляется всем трём процессорным системам (NMU1, NMU2 и ARMU). Запрос INT_DMAC_MC0 отправляется процессорным системам NMU1 и ARMU. Запрос INT_DMAC_MC1 отправляется процессорным системам NMU2 и ARMU.

Запрос на прерывание INT_DMAC_DR производится, когда установлен бит CPL регистра D_CSR, и при этом сброшен соответствующий бит маски регистра D_MASK.

Запрос на прерывание INT_DMAC_MC0 производится, когда

установлен бит CPL регистра MM0_CSR и сброшен бит маски MM0_MASK или

установлен бит CPL регистра CP0WR_CSR и сброшен бит маски CP0WR_MASK (в т.ч. в конце процесса загрузки) или

установлен бит CPL регистра CP0RD_CSR и сброшен бит маски CP0RD_MASK.

Запрос на прерывание INT_DMAC_MC1 производится, когда

установлен бит CPL регистра MM1_CSR и сброшен бит маски MM1_MASK или

установлен бит CPL регистра CP1WR_CSR и сброшен бит маски CP1WR_MASK (в т.ч. в конце процесса загрузки) или

установлен бит CPL регистра CP1RD_CSR и сброшен бит маски CP1RD_MASK.

Если причиной установки запроса INT_DMAC_MCx было завершение процесса загрузки, то запрос сбрасывается самостоятельно, когда контроллер (PWRC), управляющий загрузкой, снимает запрос LOAD_CP_NMUx.

11.9 Байтовый коммуникационный порт

Синхронные байтовые коммуникационные порты ввода/вывода предназначены для высокоскоростного обмена данными между процессорами или между процессором и внешним устройством по типу “точка - точка”. СБИС СНП-ВП в своем составе содержит два коммуникационных порта. По устройству и функциональным возможностям оба порта идентичны.

Основные характеристики каждого из коммуникационных портов:

- порт должен обеспечивать полудуплексную побайтную передачу 64-разрядных слов;
- непосредственная коммутация процессоров с помощью 8 линий данных и 4 линий управления;
- передача в обе стороны с производительностью до 160 Мбайт/сек (при тактовой частоте работы процессора равной 320 МГц);
- возможность работы со скоростью 107 Мбайт/сек, 80 Мбайт/сек;
- синхронный обмен с выставлением строба и данных от передатчика к приёмнику для увеличения скорости обмена, причём данные меняются по каждому фронту переключения строба (по типу Double Data Rate);
- формирование сигнала готовности от приёмника передатчику для синхронизации их работы;
- автоматический асинхронный арбитраж шины данных между двумя процессорами.

11.9.1 Структурная схема интерфейсной части контроллера ПДП и коммуникационных портов

Общая схема интерфейсной части контроллера ПДП процессорных систем NMU1 и NMU2 и байтовых коммуникационных портов представлена на рисунке ниже (см. Рисунок 11.2).

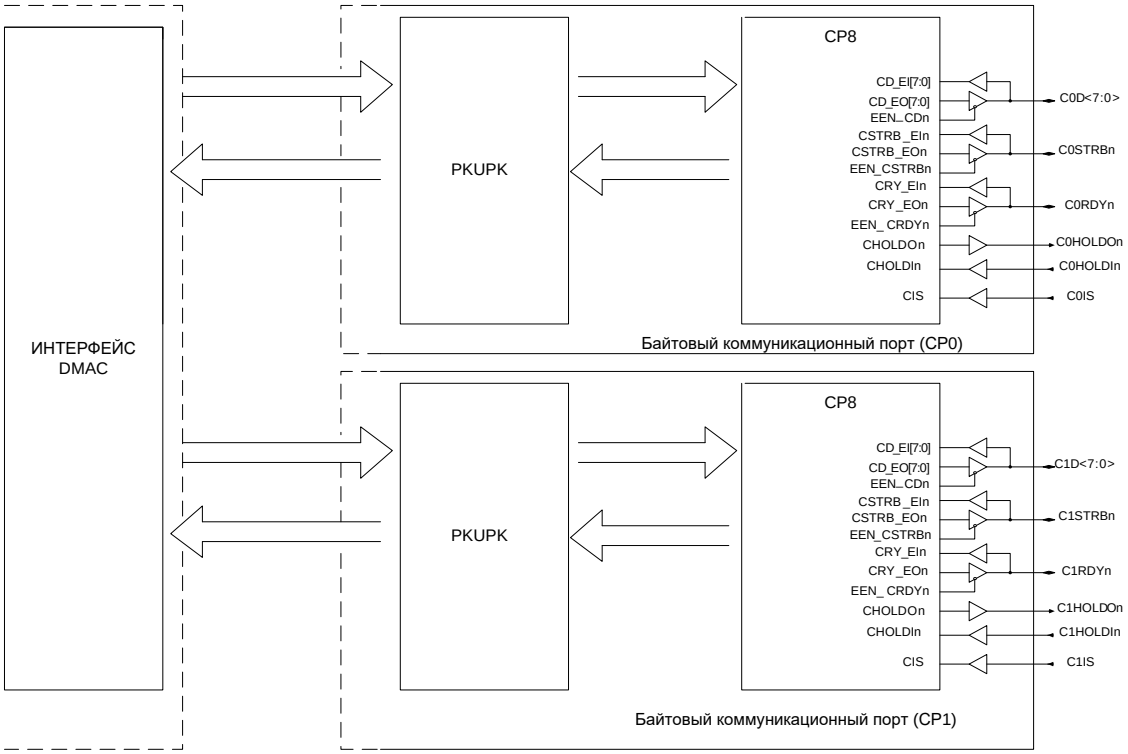


Рисунок 11.2 - Общая схема интерфейсной части контроллера ПДП и байтовых коммуникационных портов

Коммуникационный порт обеспечивает полудуплексную побайтную передачу 64-разрядных слов между КП и памятью в режиме ПДП под управлением контроллера ПДП процессорных систем NMU1 и NMU2.

DMAC_CommPort – интерфейсная часть блока ПДП. ПДП обрабатывает 4 канала, предназначенные для обслуживания коммуникационных портов СБИС:

- CP0-память (CP0WR) – канал приема коммуникационного порта CP0,
- CP1-память (CP1WR) – канал приема коммуникационного порта CP1,
- память-CP0 (CP0RD) – канал передачи коммуникационного порта CP0,
- память-CP1 (CP1RD) – канал передачи коммуникационного порта CP1.

PKUPK – блок упаковки/распаковки данных.

CP8 – интерфейсный блок коммуникационного порта.

11.9.2 Внешние выводы коммуникационного порта

В каждый момент времени коммуникационный порт может находиться в одном из двух состояний: в режиме приема или в режиме передачи. В режиме передачи порт является активным устройством, выдающим на шину данные и стробирующие сигналы. В режиме приема порт является пассивным устройством, ожидающим прихода данных. Функциональное назначение выводов коммуникационного порта в зависимости от его состояния описано ниже (см. Таблица 11.3).

Таблица 11.3 - Функциональное описание выводов коммуникационного порта

Обозначение 1) 2)	Кол-во	Тип 3)	Функциональное назначение	
			Ком. порт в режиме приема”	Ком. порт в режиме передачи
CxD	8	I/O	Входы данных	Выходы данных
CxSTRBn	1	I/O	Входной строб данных	Выходной строб данных
CxRDYn	1	I/O	Выход готовности к приему следующего байта	Вход готовности к приему следующего байта
CxHOLDIn	1	I	Вход разрешения на передачу шины от внешнего устройства	Вход запроса внешнего устройства на захват шины
CxHOLDOn	1	O	Выход запроса порта на передачу шины	Выход разрешения порта на передачу шины
CxIS	1	I	Состояние коммуникационного порта после системного сброса	Состояние коммуникационного порта после системного сброса

Примечания:

1) При обозначении выводов символ “x” для коммуникационного порта CP0 принимает значение 0, а для коммуникационного порта CP1 принимает значение 1.
2) При обозначении выводов символ “n” понимается инверсное значение сигнала
3) Для выводов со знаком инверсии активным является низкий уровень сигнала.
4) Используемые обозначения типов выводов:
   I – вход,
   O - выход,
   I/O – двунаправленный вывод

Состояние коммуникационного порта в зависимости от состояния конфигурационных входов описаны ниже (см. Таблица 11.4).

Таблица 11.4 - Состояние коммуникационного порта после системного сброса

Обозначение	Функциональное назначение	Описание	
CxIS	Состояние коммуникационного порта после системного сброса	0 - 1 -	Порт в режиме приема Порт в режиме передачи

11.9.3 Организация обмена данными по коммуникационному порту

Перед запуском канала коммуникационного порта необходимо задать начальный адрес и размер пакета передаваемых или принимаемых данных. Значения соответствующих регистров задаются в блоке контроллера ПДП процессорных систем NMU1 и NMU2.

Скорость передачи данных на внешнем интерфейсе коммуникационного порта также задаётся в блоке контроллера ПДП (DMAC). Для этого используется поле RATE регистра управления передающего канала (CPxRD_CSR).

После старта канала передачи устройство управления стремится перевести коммуникационный порт в режим передачи. Если КП находится в режиме приема, то запускается процедура арбитража шины коммуникационного порта. Детально процедура арбитража шины описана в разделе 11.9.4. После окончания процесса арбитража устройство управления выдает запрос к памяти процессора, переупаковывает считанное из памяти 64-разрядное слово в пакет из восьми байтов и, в случае готовности приемника, выдает пакет на шину коммуникационного порта (CxD[7:0]) ***старшим байтом вперед***.

Каждый выдаваемый на шину коммуникационного порта байт сопровождается изменением состояния на выходе CxSTRBn. В зависимости от частоты работы коммуникационного порта изменение состояния вывода CxSTRBn сдвинуто относительно выдачи байта данных на время, равное одному или двум периодам, или трем периодам тактового сигнала процессора. Это позволяет использовать фронты сигнала CxSTRBn для фиксации данных на приемном конце.

Сигнал готовности приемника (CxRDYn) фиксируется процессором по положительному фронту сигнала PCLK и анализируется при выдаче каждого байта данных.

После старта канала приема устройство управления находится в режиме ожидания данных с шины коммуникационного порта (CxD[7:0]). Получаемые данные фиксируются процессором как по фронту, так и по срезу сигнала строба данных (CxSTRBn). После получения каждых восьми байтов происходит переупаковка данных в 64- разрядное слово и выдается запрос на запись в память процессора.

Если канал приема не готов принять очередной байт данных, то сигнал готовности приемника (CxRDYn) снимается. Передатчик, зафиксировав отсутствие готовности приемника, должен сразу же прекратить выдачу данных и изменение сигнала CxSTRBn. Однако прекращение передачи данных при отсутствии готовности приемника не происходит мгновенно после снятия сигнала CxRDYn из-за задержек распространения сигналов от передатчика к приемнику и обратно. Поэтому в канале приема реализован буфер, позволяющий принять данные, отправленные передатчиком до фиксации неготовности приемника. Размер этого буфера накладывает ограничения на время распространения сигналов. Корректная работа канала приема гарантируется, если время распространения сигнала между приемником и передатчиком не более двух периодов тактового сигнала процессора (PCLK).

Временные диаграммы обмена по коммуникационному порту представлены на рисунке ниже (см. Рисунок 11.3).

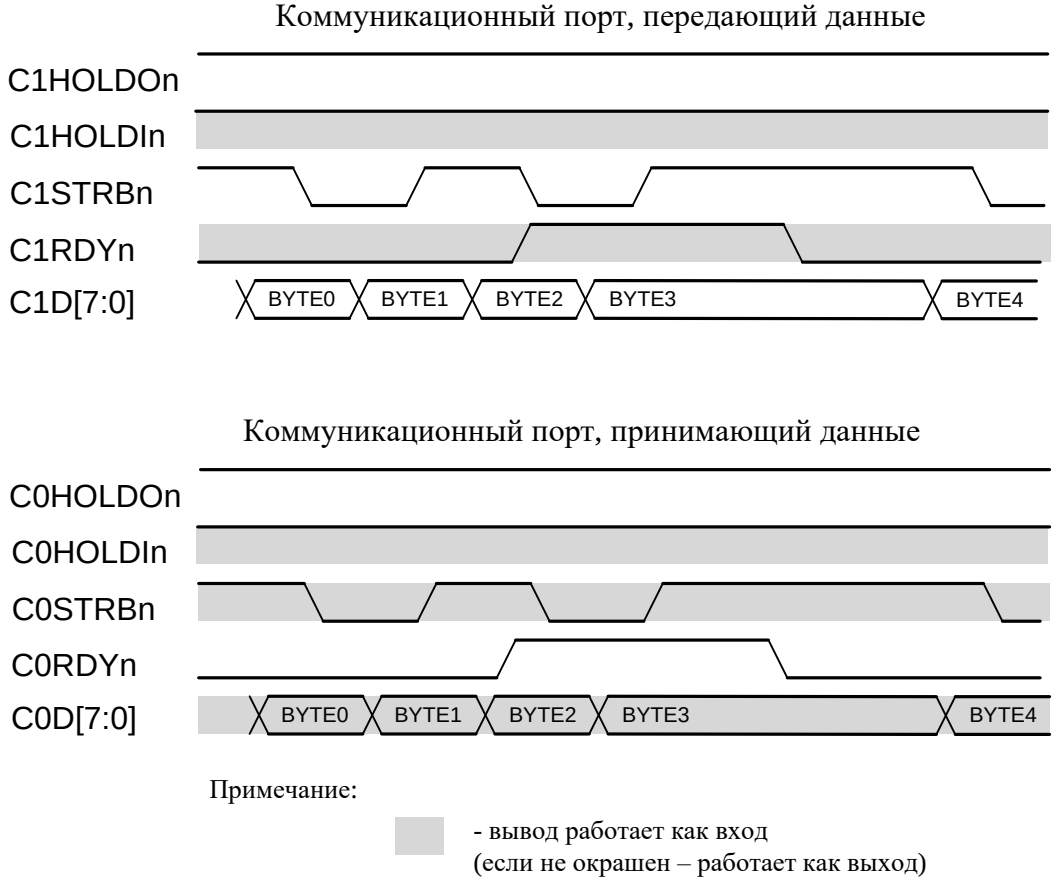


Рисунок 11.3- Временные диаграммы обмена по коммуникационному порту

11.9.4 Арбитраж шины коммуникационного порта

Процедура арбитража шины позволяет изменить состояние коммуникационного порта с приема на передачу и обратно. Управление процедурой осуществляется с помощью внешних выводов CxHOLDIn и CxHOLDOn. Функциональное назначение этих выводов в различных состояниях коммуникационного порта представлено ранее в таблице (см. Таблица 11.3).

Существует два режима работы порта при передаче шины:

- унаследованный режим (от микросхем с процессором NeuroMatrix Core). Если две микросхемы, подключенные по коммуникационному порту, работают в унаследованном режиме, то короткое время буферы сигналов XCSTRB в обоих микросхемах работают на выход;
- режим с высокоимпедансным состоянием. Если две микросхемы, подключенные по коммуникационному порту, работают в режиме с высокоимпедансным состоянием, то короткое время буферы сигналов XCSTRB в обеих микросхемах работают на вход;

Выбор режима осуществляется полем CDC регистра CSR (см. 11.7.1)

Временные диаграммы арбитража шины в унаследованном режиме представлены ниже на рисунке (Рисунок 11.4).

Коммуникационный порт, находящийся в режиме приема, запрашивает шину при запуске канала передачи порта (установке битов EN в регистре CPxRD_CSR) независимо от того, идет в данный момент прием данных или нет. С этой целью на выходе C0HOLDOn устанавливается низкий уровень сигнала.

Если коммуникационный порт, передающий данные, закончил выдачу (бит CPL в регистре CPxRD_CSR установлен) и у него инициализирован канал приема данных (бит EN в регистре CPxWR_CSR установлен), то, получив запрос на захват шины (низкий уровень C1HOLDI на рисунке), он выставляет сигнал разрешения на передачу шины(низкий уровень C1HOLDO на рисунке).

Для корректного выполнения арбитража шины необходимо, чтобы все переданные данные были записаны портом-приемником в память процессора. Если этого не произойдет, то процедура арбитража останется незаконченной, даже если разрешение на захват шины получено.

После получения сигнала разрешения на захват шины (низкий уровень на входе C0HOLDIn на рисунке) порт-приемник переводит вывод C0RDYn в состояние “на прием”, вывод C0STRBn в положение “на вывод“ и снимает запрос на захват шины (высокий уровень C0HOLDOn на рисунке).

Зафиксировав снятие запроса на захват шины, порт-передатчик переводит шину данных C1D и вывод C1STRBn в положение “на прием”, вывод C1RDYn в состояние “на вывод” и снимает сигнал разрешения на передачу шины.

В свою очередь порт-приемник, зафиксировав снятие разрешения на передачу шины (высокий уровень на входе C0HOLDIn), переводит выводы данных в состояние “на выдачу”. С этого момента он становится передатчиком и начинает осуществлять выдачу данных.

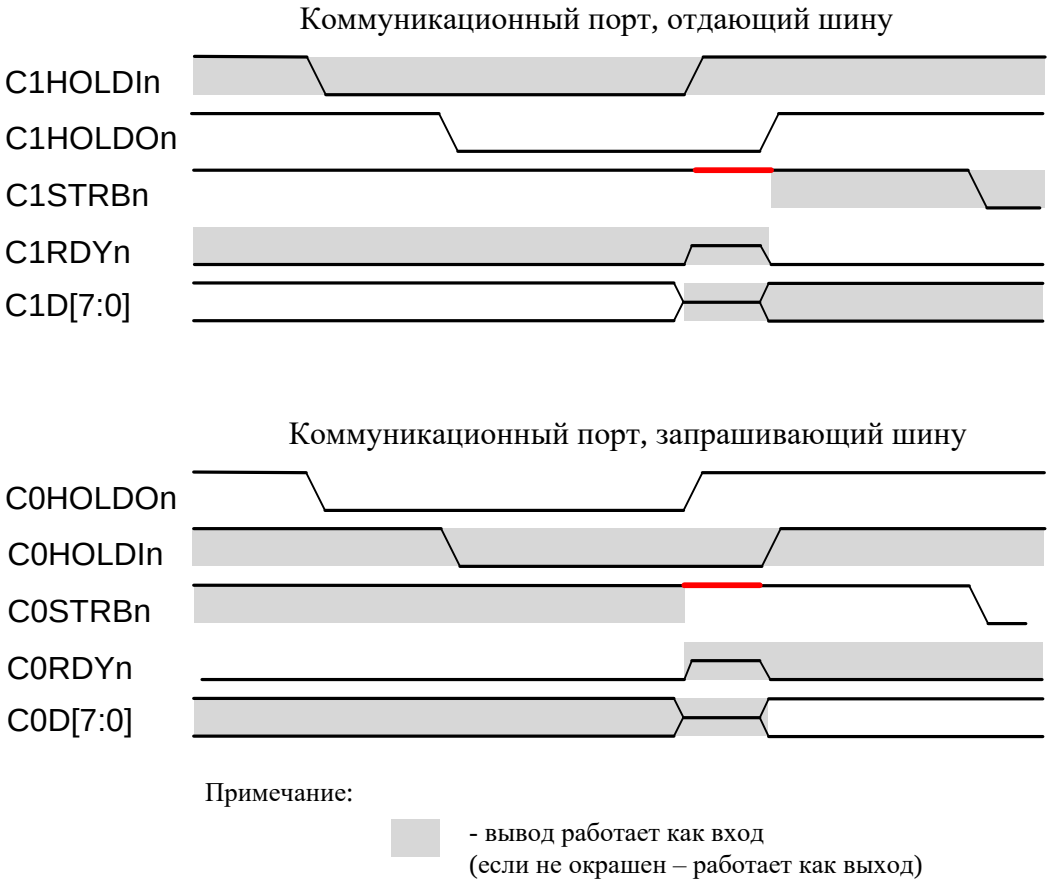


Рисунок 11.4 - Временные диаграммы арбитража шины коммуникационного порта в унаследованном режиме

Красным на рисунке выше (см. Рисунок 11.4) выделен момент времени, когда сигналы C0STRBn и C1STRBn выботаются друг другу навстречу.

На рисунке ниже (см. Рисунок 11.5) показана передача шины, используя высокоимпедансное состояние. Красным выделен момент времени, когда сигналы C1STRBn и C0STRBn находятся в высокоимпедансном состоянии. При использовании данного режима на плату необходимо подключать подтягивающий резистор к питания 3.3В.

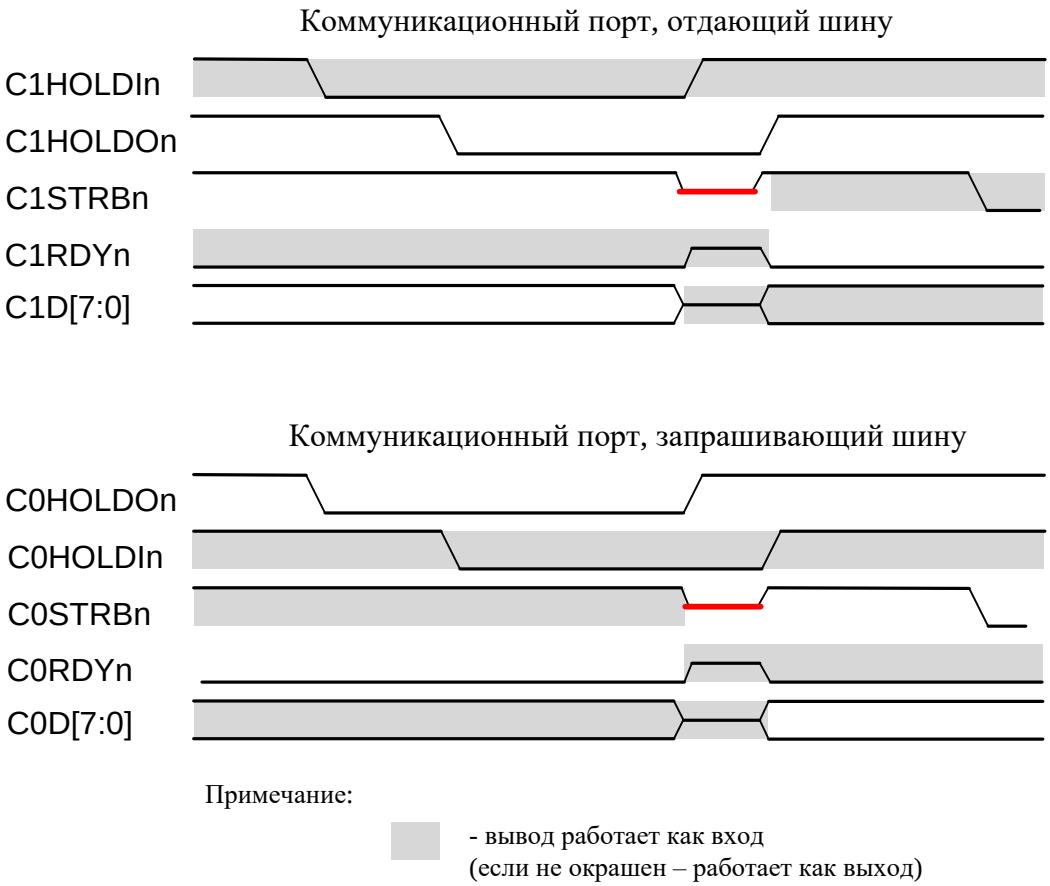


Рисунок 11.5 - Временные диаграммы арбитража шины коммуникационного порта в режиме перехода с высокоимпедансным состоянием

12 Электрические и конструктивные характеристики СБИС СНП-ВП

12.1 Состав и расположение внешних выводов СБИС СНП-ВП.

СБИС СНП-ВП изготавливается в 544-выводном пластиковом корпусе PBGA. Расположение внешних выводов микросхемы представлено ниже (см. Рисунок 12.1).

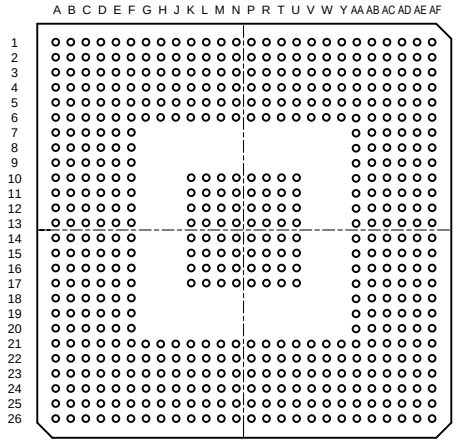


Рисунок 12.1 - Расположение внешних выводов СБИС ЦУПП

В таблице ниже (Таблица 12.1) представлены выводы микросхемы, отсортированные по их функциональному назначению.

Таблица 12.1 – Выводы микросхемы, отсортированные по функциональному назначению

Обозначение[1)]	Кол-во	Тип[2)]	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
				Интерфейс с внешней шиной LPDDR2	
D31…D0	32	I/O		Шина данных с SDRAM памятью типа LPDDR2	E4, E3, F5, F2, G1, H6, H5, H2, W2, W5, W6, Y1, AA2, AA5, AB3, AB4, J6, J5, J2, J1, K6, L6, L5, L2, T2, T5, T6, U6, V1, V2, V5, V6
CA9…CA0	10	O(Z)		Шина адреса	M5, M4, M3, M2, N1, P1, R2, R3, R4, R5
DQS_P3... DQS_P0	4	O(Z)		дефференциальный строб данных (прямая фаза)	G3, Y4, K4, U4
DQS_N3... DQS_N0	4	I/O		дефференциальный строб данных (инверсная фаза)	G4, Y3, K3, U3
DM3, … DM0	4	I/O		Сигналы маскирования данных при записи	G6, Y6, K1, U1
CS1...CS0	1	O(Z)		Выбор банка внешней памяти	P3, P4
CKE1, CKE0	2	O(Z)		Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления	N4, N3
ZQRES	1	AI		калибровачный резистор	F6
CK_P, CK_N	2	O		Выход тактового сигнала внешней шины	P6, N6
VREF1, … VREF0	4	AI		Вход напряжения смещения	H4, W4
MCLKXI		I		Вход кварца для контроллера внешней памяти	F8
MCLKXO		I/O		Вход кварца для контроллера внешней памяти	D6
MPLLBYPASS		I		Вход отключения PLL. Должен быть подключен к логическому нулю для того чтобы MPLL работала.	A6
MPLLSTOP		O		Сигнал отключения внешнего CLK для MPLL	B6
				SPI интерфейс	
SPICLK	1	O	8	Выход тактового сигнала	U21
SPITXD	1	O	8	Выход передаваемых данных	T23
SPIRXD	1	I,U		Вход принимаемых данных	U22
SPI_CS7, …, SPI_CS0	8	O	4	Выбор микросхемы/устройства SPI	U25, U24, V26, V24, V23, W25, Y24, W23
				Порты общего назначения	
GPIO15, …, GPIO0	16	I/O, Sh	4	Программируемы входы/выходы	AF7, AE7, AC7, AB7, AA7, AE6, AD6, AC6, AF5, AD5, AC5, AF4, AE4, AD4, AE3, AF2
				UART порт 0	
UART0TXD	1	O	4	Выход передаваемых данных	T25
UART0RXD	1	I, Sh		Вход принимаемых данных	R23
XCTS0	1	I, Sh		Разрешение передачи данных	T22
XRTS0	1	O	4	Запрос на передачу данных	T26
				UART порт 1	
UART1TXD	1	O	4	Выход передаваемых данных	R26
UART1RXD	1	I, Sh		Вход принимаемых данных	R24
XCTS1	1	I, Sh		Разрешение передачи данных	P24
XRTS1	1	O	4	Запрос на передачу данных	R25
				CAN интерфейс	
CAN_TX				Выход передаваемых данных	P26
CAN_RX				Вход принимаемых данных	P25

Продолжение таблицы 12.1

Обозначение[1)]	Кол-во	Тип[2)]	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
				USB2.0 (full speed) интерфейс	
DPPAD	1	I/O		Линия D+ шины USB	AF11
DMPAD	1	I/O		Линия D- шины USB	AF12
USBCLKXI	1	I		Вход P для подключения тактового сигнала 48 МГц	AA14
USBCLKXO	1	I/O		Вход M для подключения тактового сигнала 48 МГц	AA12
EXT12K	1	O		вывод внешнего резистора	AA9
USBID	1	I		идентификация типа USB устройств	AB13
USBIDPU	1	O	4	управление резистором pull-up	AC14
USBOVCUR	1	I		индикация превышения допустимого тока	AC11
USBDRVVBUS	1	O	4	управление питанием шиной USB	AD13
USBCON	1	I		Сигнал наличия питания Vbus шины USB	AC13
				I2C интерфейс	
SDA	1	I/O		Последовательная линия данных I2C	AA26
SCL	1	I/O		Линия синхронизации I2C	Y26
				Ethernet интерфейс	
ETH_TXD1, ETH_TXD0	2	O	6	выходные данные	AD16, AF16
ETH_RXD1, ETH_RXD0	2	I		входные данные	AC16,AC17
ETH_TXEN	1	O	6	разрешение передачи	AD15
ETH_CRSDV	1	I		обнаружение несущей	AC15
ETH_RMIICLK	1	I		входной тактовый сигнал 50 МГц	AA16
ETH_EN	1	I		разрешение ethernet интерфейса 0 - Ethernet не работает 1 - Ethernet работает	AB15
ETH_MDC	1	O	6	выходной тактовый сигнал для микросхемы физического интерфейса	AE14
ETH_MDIO	1	I/O	6	шина управления физическим интерфейсом	AF15
				Выводы счетчика реального времени	
RTCCLKXI	1	I		Вход A для подключения часового кварца 32768 Гц	P21
RTCCLKXO	1	I/O		Вход B для подключения часового кварца 32768 Гц	M21
RTCINT	1	O	1	Выход прерывания по будильнику	N26
RTCPORn	1	I		Сброс RTC	N23
RTCMPWR	1	I		Признак основного питания	N24
				Выводы блока интервальных таймеров	
ITSTROB	1	I, Sh		Вход строба записи значения интервального таймера	F7
IT_1SO	1	O	4	Выход программируемого генератора импульсов	C7
ITO	1	O	4	Выход сигнала дополнительного временного интервала	D7
TCOD_NULL	1	O	4	Выход генератора ПСП для калибровки	A7
TCOD	1	O	4	Выход генератора ПСП для калибровки	B7

Продолжение таблицы 12.1

Обозначение[1)]	Кол-во	Тип[2)]	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
				Выводы от блока предварительной обработки.	
ACLK	1	I		Тактовый сигнал для АЦП и SIGNx/MAGNx	F12
SIGN11,..., SIGN0	12	I		Цифровые входы навигационных данных (знаковый разряд)	D8, A8, D9, B9, E10, B10, C11, A10, E12, C12, C13, E13
MAGN11 ,…, MAGN0	12	I		Цифровые входы навигационных данных (значимый разряд)	E8, B8, E9, C9, F10, C10, D11, A11, F11, D12, B12, D13
				JTAG-интерфейс	
TDO	1	O(Z)	4	Выход данных тестового порта	AB26
TDI	1	I		Вход данных тестового порта	AC25
TCK	1	I		Тактовый сигнал тестового порта	AB24
TMS	1	I		Сигнал выбора режима тестового порта	Y21
XTRST	1	I,U		Сброс тестового порта	W21
				Отладочный интерфейс ядра ARM	
RTCK	1	O	4	Выход синхросигнала тестового порта	AA22
XSRST	1	I/O, U	4	Программный сброс от отладчика	AD26
				Коммуникационный порт 0	
C0D7...C0D0	8	I/O	6	Шина данных	F23, G24, G25, G26, H23, H24, H25, H26
C0STRBn	1	I/O	6	Строб данных	G21
C0RDYn	1	I/O		Сигнал готовности	J24
C0HOLDOn	1	O	4	Запрос на захват шины	J23
C0HOLDIn	1	I		Разрешение на передачу шины	G22
C0IS	1	I		Инициализация после сброса	H21
				Коммуникационный порт 1	
C1D7...C1D0	8	I/O	6	Шина данных	M26, L25, L24, L23, K26, K25, K24, K23
C1STRBn	1	I/O	6	Строб данных	K21
C1RDYn	1	I/O		Сигнал готовности	J25
C1HOLDOn	1	O	4	Запрос на захват шины	J26
C1HOLDIn	1	I		Разрешение на передачу шины	K22
C1IS	1	I		Инициализация после сброса	J21
				Тестовый порт NMC	
TD7,…, TD4	4	O	6	Выход тестового порта ядра NMC3 процессорной системы NMU2	C26, D24, D25, E23
TD3,…, TD0	4	O	6	Выход тестового порта ядра NMC3 процессорной системы NMU1	E24, E26, F24, F26
				Общее управление	
SCLKXI	1	I		Вход тактового сигнала	AA18
SCLKXO	1	I		Вход тактового сигнала	AA20
XRESET	1	I, Sh		Системный сброс.	AF20
XINT3, …, XINT0	4	I, Sh		Входы внешних прерываний	AE20, AF19, AD19, AB19
WDT	1	O	4	Выход сторожевого таймера	AD20
BOOTM2,…, BOOTM0[3)]	3	I		Сигналы управления выбора устройства начальной загрузкой	AC21, AD21, AF21
BOOTM_NM1, BOOTM_NM0	2	I		Сигнал управления начальной загрузкой ядер NMC	AF22, AB21
TST_I2,..., TST_I0	3	I		Выбор типа загрузки	AC22, AD22, AE22

Продолжение таблицы 12.1

Обозначение[1)]	Кол-во	Тип[2)]	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
				Тестовые выводы	
TSTMODE	1	I		Тестовый вход для тестирования СБИС на фабрике. Должен быть подключен к логическому нулю.	AC24
VPD	1	I		Отключение pullup/pulldown резисторов для тестирования. Должен быть подключен к логическому нулю.	C25
				Выводы управления PLL	
SPLLBYPASS	1	I		Вход отключения PLL. Должен быть подключен к логическому нулю.	AC18
SPLLSTOP	1	O	4	Сигнал отключения внешнего тактового сигнала	AC19
				Выводы блока АЦП	
VIP_CH0, VIM_CH0	2	AI		Дифференциальные входы АЦП0	F19, F18
VIP_CH1, VIM_CH1	2	AI		Дифференциальные входы АЦП1	D19, D20
VIP_CH2, VIM_CH2	2	AI		Дифференциальные входы АЦП2	D16, D17
VIP_CH3, VIM_CH3	2	AI		Дифференциальные входы АЦП3	F16, F15
VRP_CH23, VRP_CH01	2	AO		вывод для внешнего конденсатора	A18, B17
VRM_CH23, VRM_CH01	2	AO		вывод для внешнего конденсатора	A17, B18
VRC_CH23, VRC_CH01	2	AO		вывод для внешнего конденсатора	A16, B19
VDDI	40	S		Напряжение питания ядра +1,2В	AA11, AA13, AA15, AA17, AA19, AA21, AA6, AA8, F13, F21, F9, K10, K11, K13, K14, K16, K17, L10, L13, L14, L17, N10, N11, N16, N17, P10, P11, P16, P17, T10, T13, T14, T17, U10, U11, U13, U14, U16, U17, V21

Продолжение таблицы 12.1

Обозначение[1)]	Кол-во	Тип[2)]	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
Питание					
VDE1	63	S		Напряжение питания буферов ввода/вывода +3,3В	A12, A2, A23, A25, A4, A9, AA25, AB11, AB17, AB22, AB23,AB25,AB8, AC12, AC20, AC9, AD14, AD18, AD23, AD25, AD8, AE10, AE12, AE15, AE17, AE19, AE2, AE21, AE24, AE26, AE5, AF1, AF13, AF23, AF25, AF3, AF8, B1, B24, B26, B3, C2, C23, C4, C6, C8, D10, D23, E22, E6, F25, H22, L22, L26, M24, N25, P23, T24, U26, V22, W24, W26, Y23
VDE2	30	S		Напряжение питания буферов интерфеса LPDDR2 +1,2В	AA1, AA3, AB2, AB5, AC1, AC3, AD2, D1, D3, E2, F1, F3, G5, H1, H3, J4, K2, L4, M1, M6, N2, P2, R1, R6, T4, U2, V4, W1, W3, Y5
AVD12	9	S		Напряжение питания аналогового блока АЦП 1,2В	C16 ,C17, C18, C19, C20 ,D15, D18, E15, E18
RTC_VDD	2	S		Напряжение питания домена RTC +1,2В	N21, M22, P22, R21
AVD33	2	S		Напряжение питания аналогового блока формирования опорного напряжения для АЦП 3,0В	A21, B14
AVD12_SPLL, AVD12_MPLL	2	S		Напряжение питания для PLL +1,2В	AF18, B5
AVDF2	1	S		Аналоговое напряжение питания для USB +1,2В	AD11
AVDP	1	S		напряжение питания PLL для USB +1,2В	AF9
AVDF1	2	S		Напряжение питания для USB +3,3В	AB9, AB10
VSS	123	S		Цифровая земля	A1, A24, A26, A3, AA23, AA24, AA4, AB1, AB12, AB14,AB16, AB18, AB20, AB6, AC2, AC23, AC26, AC4, AC8, AD1, AD12, AD17, AD24, AD3, AD7, AD9, AE1, AE11, AE13, AE16, AE23, AE25, AE8, AF10,AF14,AF17, AF24, AF26, AF6, B11, B2, B23, B25, B4, C1, C22, C24, C3, C5, D2, D22, D26, D4, D5, E1, E11, E21, E25, E5, E7, F22, F4, G2, G23, J22, J3, K12, K15, K5, L1, L11, L12, L15, L16, L21, L3, M10, M11, M12, M13, M14, M15, M16, M17, M23, M25, N12, N13, N14, N15, N22, N5, P12, P13, P14, P15, P5, R10, R11, R12, R13, R14, R15, R16, R17, R22, T1, T11, T12, T15, T16, T21, T3, U12, U15, U23, U5, V25, V3, W22, Y2, Y22, Y25
AVS12	23	S		Аналоговая земля аналогового блока АЦП	A13, A15, A19, A20, A22, B13, B15, B16, B20, B22, C14, C15, C21, D14, D21, E14, E16, E17, E19, E20, F14, F17, F20
AVS33	2	S		Аналоговая земля блока формирования опорного напряжения для АЦП	A14, B21
AVS12_SPLL, AVS12_MPLL	2	S		Земля для PLL	AE18, A5
AVSF1	3	S		Аналоговая земля для USB	AD10, AC10, AA10
AVSP	1	S		Аналоговая земля для PLL USB	AE9

Примечания: *1) Для выводов со знаком инверсии активным является низкий уровень сигнала.*

2) Используемые обозначения типов выводов:

I – вход,

O – выход,

O(Z) – выход с высокоимпедансным состоянием,

I/O – двунаправленный вывод,

S – питание.

AI – аналоговый вход

AO – аналоговый выход

Sh – буфер с триггером Шмидта

U – на выводе стоит pull-up резистор

D – на выводе стоит pull-down резистор

3) Кодировка выводов BOOTM2, BOOTM1, BOOTM0:

000 – загрузка с SPI;

001 – загрузка с UART0;

010 – загрузка с UART1;

011 – зарезервировано;

1xx – загрузка по Ethernet;

12.2 Конструктивные характеристики

СБИС СНП-ВП изготавливается в 544-выводном пластиковом корпусе типа Ball Grid Array. Внешний вид корпуса микросхемы представлен ниже (см. Рисунок 12.2 – Рисунок 12.4) с указанием сведений о габаритных и установочных размерах.

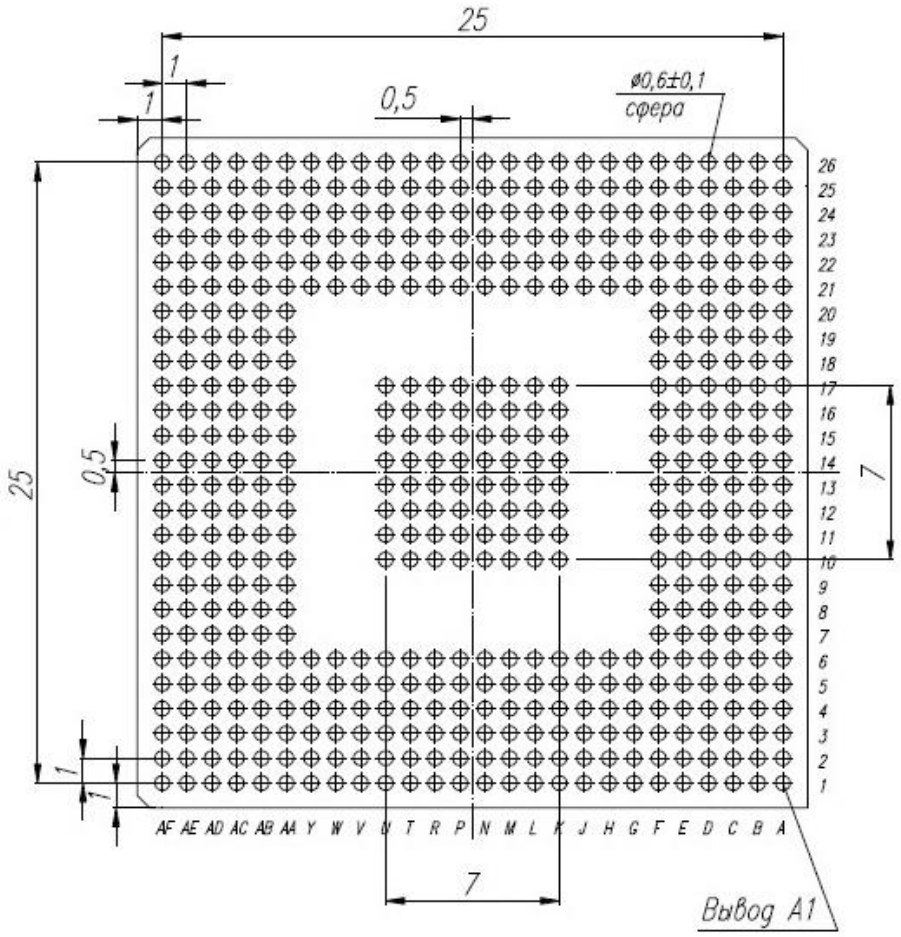


Рисунок 12.2 – Вид корпуса со стороны выводов

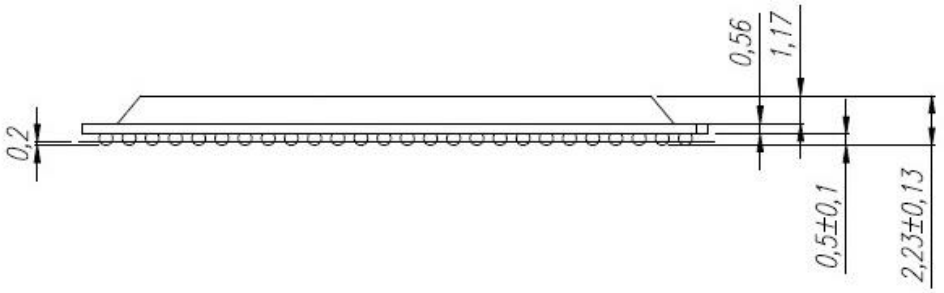


Рисунок 12.3 – Вид корпуса сбоку

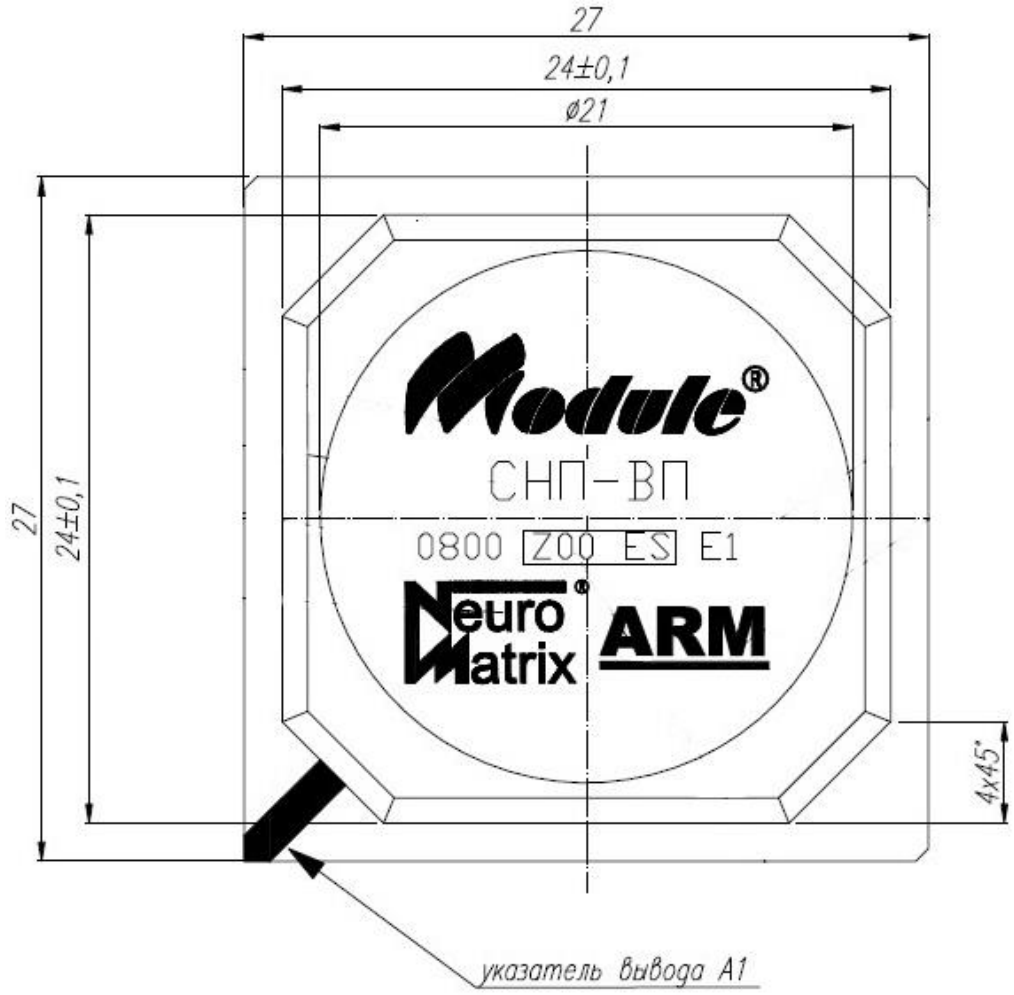


Рисунок 12.4 – Вид корпуса сверху

Основные характеристики корпуса:
Механические характеристики:

- размер – 27×27 мм2;
- количество выводов – 544
- шаг выводов – 1 мм;
- размер вывода – 0,60 мм.
- Масса СБИС – 2,7г

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 – 13,8 °С/Вт;
- Тепловое сопротивление при скорости обдува 1 м/с – 11,7 °С/Вт;
- Тепловое сопротивление при скорости обдува 2 м/с – 11,0 °С/Вт.

12.3 Электрические характеристики

Электрические параметры микросхемы соответствуют нормам, приведенным ниже в таблицах (Таблица 12.2, Таблица 12.3).

Таблица 12.2 – Электрические параметры микросхемы при приемке и поставке

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Норма параметра			Температура среды, °С
			не менее	номинал	не более	
Выходное напряжение низкого уровня КМОП буфера, В	U_{OL1}	I_{OL}=2 мА U_{CC3}=3,6 В U_{CC2}=1,3 В	-	-	0,4	от минус 40 до +85
Выходное напряжение высокого уровня КМОП буфера, В	U_{OH1}	I_{OH}= -2 мА U_{CC3}=3,0 В U_{CC2}=1,1 В	2,0	-	-	от минус 40 до +85
Выходное напряжение низкого уровня HSUL12 буфера, В	U_{OL2}	I_{OL}=0.1 мА U_{CC1}=1,3 В U_{CC2}=1,3 В	-	-	0,1U_{CC1}	от минус 40 до +85
Выходное напряжение высокого уровня HSUL12 буфера, В	U_{OH2}	I_{OL}=-0.1 мА U_{CC1}=1,14 В U_{CC2}=1,1 В	0,9U_{CC1}	-	-	от минус 40 до +85
Ток утечки низкого уровня на входе КМОП буфера, мкА	I_{LIL}	U_{IL}=0 В U_{CC3}=3,6 В U_{CC2}=1,3 В	-	-	5,0	от минус 40 до +85
Ток утечки высокого уровня на входе КМОП буфера, мкА	I_{LIH}	U_{IH}=3,6 В U_{CC3}=3,6 В U_{CC2}=1,3 В	-	-	5,0	от минус 40 до +85
Ток потребления статический, мА	I_{CC1}	f© = 0 МГц U_{CC1}=1,3 В U_{CC2}=1,3 В U_{CC3}=3,6 В	-	-	300,0	от минус 40 до +85
Ток потребления статический, мА	I_{CC2}	f© = 0 МГц U_{CC1}=1,3 В U_{CC2}=1,3 В U_{CC3}=3,6 В	-	-	50,0	от минус 40 до +85
Максимальная частота следования импульсов тактовых сигналов, МГц	f_{C1}	U_{CC1}=1,14 В U_{CC2}=1,1 В U_{CC3}=3,0 В	-	-	40	от минус 40 до +85
Максимальная частота следования импульсов тактовых сигналов, Гц	f_{C2}	U_{CC1}=1,14 В U_{CC2}=1,1 В U_{CC3}=3,0	-	-	32768	от минус 40 до +85
Максимальная частота следования импульсов тактовых сигналов, МГц	f_{C3}	U_{CC1}=1,14 В U_{CC2}=1,1 В U_{CC3}=3,0	-	-	48,0	от минус 40 до +85

Таблица 12.3 – Электрические параметры АЦП

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Норма параметра			Температура среды, °С
			не менее	номинал	не более	
Разрядность преобразования АЦП, бит	N	F_{IN} =1,0МГц F_S =90 МГц		10	-	от минус 50 до +85
Частота преобразования, МГц	F_S		-	-	90	от минус 50 до +85
Дифференциальная нелинейность АЦП, LSB[1)]	DNL	F_{IN} =1,0МГц F_S =90 МГц		-	±1	от минус 50 до +85
Интегральная нелинейность АЦП, LSB	INL	F_{IN} =1,0МГц F_S =90 МГц	-	-	±3	от минус 50 до +85
Смещение нуля, мВ	U_{OFS}	F_{IN} =1,0МГц F_S =90 МГц	-14,7	-	+14,7	от минус 50 до +85
Амплитуда дифференциального входного сигнала [2)], В	V_{INDL}[1)]	F_{IN} =1,0МГц F_S =90МГц		0,6		от минус 50 до +85
Амплитуда дифференциального входного сигнала [2)], В	V_{INDH}[1)]	F_{IN} =1,0МГц F_S =90МГц		1,2		от минус 50 до +85
Входное сопротивление, кОм	Z_{INDC}	F_S =90 МГц		13		от минус 50 до +85
Отношение сигнал/(шум +искажения), dBc	SNDR	Fs=90МГц Fin=20МГц	50			от минус 50 до +85
Отношение мощности сигнала к мощности 5 гармоник, dBc	THD	Fs=90МГц Fin=20МГц	52			от минус 50 до +85
Динамический диапазон, dBFc	SFDR	Fs=90МГц Fin=20МГц	54			от минус 50 до +85
Развязка между АЦП, dBFc	XTALK	Fs=90МГц Fin=20МГц	-40			от минус 50 до +85

Примечания:

1) 1 LSB = 1,17мВ;
2) АЦП имеет дифференциальный вход. Под амплитудой Vind понимается размах сигнала, равный учетверенной амплитуде входного синусоидального однофазного сигнала.

Значения предельно-допустимых и предельных электрических режимов эксплуатации микросхемы в диапазоне рабочих температур среды приведены ниже в таблице (Таблица 12.4).

Таблица 12.4 – Нормы предельно-допустимых и предельных электрических режимов эксплуатации микросхемы

Наименование параметра режима, единица измерения	Буквенное обозначение	Предельно – допустимый режим			Предельный режим	
		не менее	норм.	Не более	не менее	не более
Напряжение питания буферов ввода/вывода интерфейса LPDDR2 (VDE2), В	U_{CC1}	1,14	1,2	1,3	-0,5	2,5
напряжение смещения питания буферов ввода/вывода интерфейса LPDDR2	V_{REF}	$0,49U_{CC1}$	$0,5U_{CC1}$	$0,51U_{CC1}$		
Напряжение питания ядра (VDDI), В	U_{CC2}	1,1	1,2	1,3	-0,5	1,8
Напряжение питания питания буферов ввода/вывода, USB (VDE1, AVDF1), В	U_{CC3}	3,0	3,3	3,6	-0,5	4,0
Напряжение питания аналогового блока АЦП (AVD12), В	U_{ADC}	1,1	1,2	1,3	-0,5	1,8
Напряжение питания аналогового блока АЦП (AVD33), В	U_{REF}	2,7	3,0	3,6		
Напряжение питания для PLL (AVD12), В		1,1	1,2	1,3	-0,5	1,8
Входное напряжение высокого уровня, В	U_{IH}	2,0	-	$U_{CC3}+0,3$	-	-
Входное напряжение низкого уровня, В	U_{IL}	-0,3	-	$0,3U_{CC3}$	-	-
Напряжение на стандартных входах, В	U_{IN}	-0,3	-	$U_{CC3}+0,3$	-0,5	3,6
Выходное напряжение, В	U_{OUT}	-0,3	-	U_{CC3}	-0,5	3,6
Входной ток, мкА	I_{IN}	-	-	5	-	5
Выходной ток высокого уровня, мА 4- миллиамперный выход 6- миллиамперный выход 8- миллиамперный выход	I_{OH}	-	-	-4	-	-13
		-	-	-6	-	-13
		-	-	-8	-	-13
Выходной ток низкого уровня, мА 4- миллиамперный выход 6- миллиамперный выход 8- миллиамперный выход	I_{OL}	-	-	4	-	13
		-	-	6	-	13
		-	-	8	-	13

12.4 Порядок подачи питания при включении

Последовательность подачи напряжений питания и сигналов при включении:

1) Напряжение питания для внутренней логики VDDI;
2) Напряжение питание для аналоговых блоков AVD33, AVD12;
3) Напряжение питания для буферов ввода/вывода: VDE1, VDE2, VDE3;
4) Тактовый сигнал CLKIN0, CLKIN1;
5) системный сброс XRESET.

Последовательность подачи остальных сигналов не регламентируется. Выключение напряжений питания производится в порядке, обратном включению.

13 Рекомендации по включению микросхемы

13.1 Рекомендации по включению АЦП

На рисунке ниже приведена рекомендованная схема подключения конденсаторов к выводам СБИС, относящихся к АЦП.

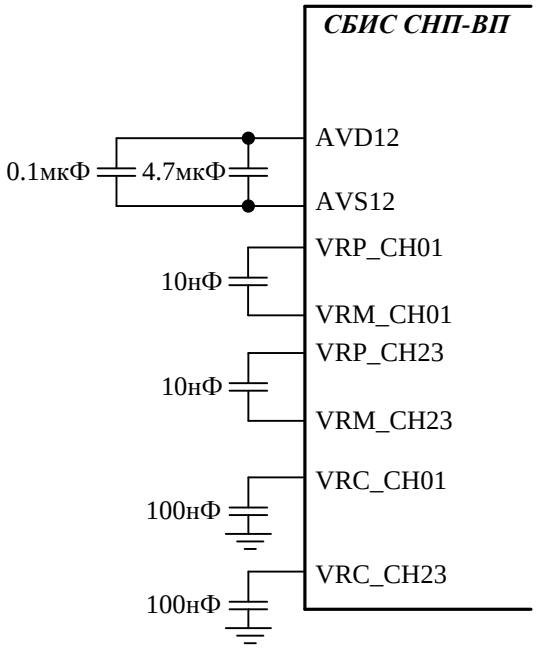


Рисунок 13.1 – Рекомендованная схема подключения АЦП

- между питанием и землей используйте два керамических конденсатора 0.1мкФ и 4.7 мкФ;
- между выводами VRPx и VRMx ставьте конденсатор 10 нФ как можно ближе к выводам микросхемы;
- между выводами VRCx и землей ставьте конденсатор 100 нФ как можно ближе к выводам микросхемы.

Ниже приведены некоторые характеристики выводов

Таблица 13.1 – Характеристики выводов АЦП

параметр	обозначение	минимальное значение	типовое значение	максимальное значение	единицы измерения
входная емкость VIPx, VIMx	Cin		0,96		пФ
справочное напряжение (высокое)	VRP	0,78	0,91	0,98	В
справочное напряжение (низкое)	VRM	0,18	0,31	0,38	В
напряжение средней точки	VRC	U_{ADC}/2 - 6%	0,61	U_{ADC}/2 + 6%	В

Далее приведены возможные примеры подключения к АЦП источника сигнала.

На рисунке ниже (Рисунок 13.2) приведен пример одного из вариантов согласования входа АЦП микросхемы К1888ВС018 в случае несимметричного подключения источника сигнала на промежуточной частоте, например с генератора, имеющего радиочастотный выход 50 Ом.

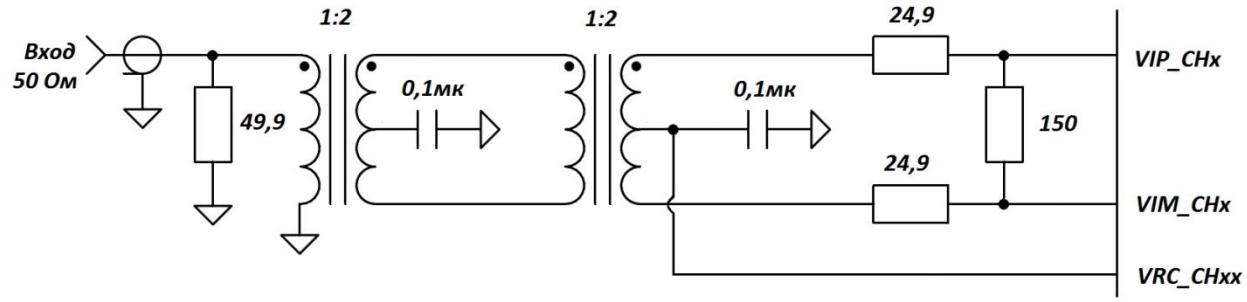


Рисунок 13.2 –Схема подключения АЦП вариант 1

Стоит обратить внимание на вывод VRC_CHxx, задающий смещение на входе АЦП. Индекс xx обозначает номера каналов и может быть 01 или 23. Соответственно подключение к выводу смещения осуществляется попарно для 0,1 каналов и 2,3 каналов. Для обеспечения лучшей производительности при высоких входных частотах два трансформатора соединены последовательно, как показано на рисунке 1, чтобы минимизировать гармонические искажения четного порядка. Первый трансформатор преобразовывает асимметричный сигнал в дифференциальный сигнал, при этом заземленный вход на первичной обмотке ухудшает амплитудный баланс на вторичной обмотке из-за паразитной емкостной связи между обмотками. Второй трансформатор улучшает амплитудный баланс, и таким образом выравнивает гармонические искажения четного порядка. Однако необходимо учитывать тот факт, что реальная полоса пропускания на входе АЦП будет меньше в 1,4 раза.

$$\Delta F = \frac{\Delta F\text{трансформатора}}{\sqrt{2}}$$

В случае применения радиочастотного приёмника с дифференциальным выходом перед входом АЦП, рекомендуется согласование подключения, приведенное ниже (см. Рисунок 13.3). В данном примере приводится согласование входов АЦП через трансформаторную развязку с соотношением импедансов 1:2. Выбор трансформатора осуществляется в зависимости от требуемой полосы пропускания, схемы подключения, вносимых потерь и ожидаемого размаха пик-пик дифференциального сигнала. Рабочий допустимый размах пик-пик дифференциального входного сигнала на входе АЦП задается программно и может быть равен либо 0,6 В, либо 1,2 В.

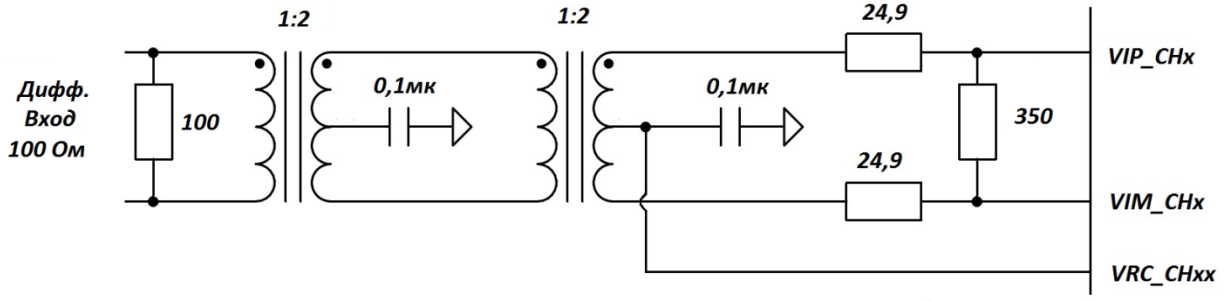


Рисунок 13.3 –Схема подключения АЦП вариант 2

13.2 Рекомендации по проектированию печатной платы для USB интерфейса

Рекомендации для сигнальных линий DMPAD/DPPAD:

- линии DMPAD/DPPAD должны иметь одинаковую длину. Допускаемый разброс длины 1.5 мм;
- линию между выводами микросхемы DMPAD/DPPAD и разъемом следует делать как можно короче. Допустимое время распространение сигнала между разъемом и микросхемой:
  - на стороне хост-контроллера не более 3 нс;
  - на стороне конечного устройства не более 1 нс;
- волновое дифференциальное сопротивление линии Zdiff= 90 Ом ± 10% @ 240МГц;
- линии DMPAD/DPPAD должны быть проведены рядом друг с другом. Расстояние между линиями и ширина линий должна быть постоянна по всей длине;
- запрещено делать углы для линий DMPAD/DPPAD больше 45 градусов;
- рекомендуется сделать защиту линий DMPAD/DPPAD шинами земли. Расстояние от линий DMPAD/DPPAD до защитных линий земли должно быть не меньше ширины линии DMPAD/DPPAD;
- трассировка других сигналов над/под линиями DMPAD/DPPAD запрещена;
- В слоях под/над линиями DMPAD/DPPAD делайте экраны из земли. Данные экраны должны перекрываться с защитными линиями земли, сделанными в том же слое что и сигналы DMPAD/DPPAD;
- лучше всего трассировать линии DMPAD/DPPAD в самом верхнем слое;
- по возможности избегать отверстий (VIA), так как это приводит к отражениям;

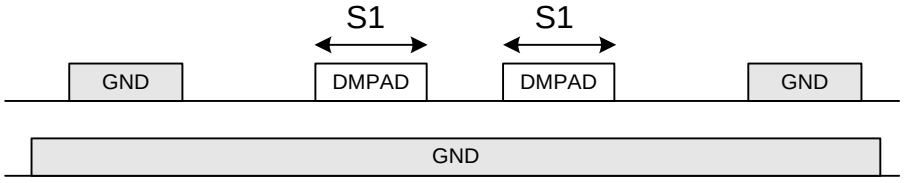


Рисунок 13.4 – пример трассировки DMPAD/DPPAD

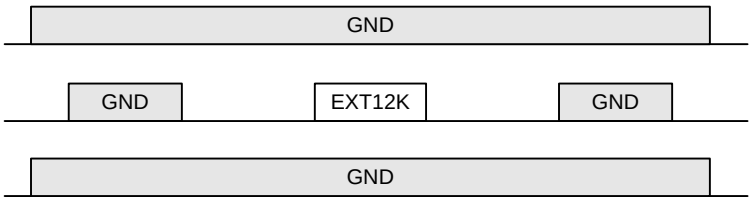


Рисунок 13.5 – экранирование линии EXT12K

Рекомендации по соединению вывода EXT12K с резистором:

- вывод EXT12K необходимо соединить с резистором, второй вывод которого должен быть подсоединен к земле;
- значение резистора 12 КОм ± 1%;
- линия от вывода микросхемы до резистора должна быть как можно короче;

- паразитная емкость линии между выводом EXT12K и резистором должна быть не более 20 пФ;
- для минимизации перекрестных наводок на данную линию связи необходимо экранировать землей ее с двух сторон в том же слое, а так же в слое выше и ниже (см. Рисунок 13.7).

Прочие рекомендации:

- необходимо избегать наводок на выводы питания блока USB (AVDF1, AVDF2, AVDP) и выводы земли блока USB (AVSF1, AVSP);
- использовать отдельный фильтр по питанию для каждого вывода (AVDF1, AVDF2, AVDP);
- не использовать питание блока USB для питания других блоков;
- помехи по земле блока USB должны быть меньше 25 мВ;
- точность тактового сигнала 48 МГц должна быть меньше ±200ppm, джиттер меньше ±50 пс.

13.3 Рекомендации по проектированию печатной платы для LPDDR2 интерфейса

Рассматривается два основных варианта подключения микросхем памяти к СБИС СНП-ВП: подключение одной микросхемы и подключение двух микросхем.

Вариант подключения одной микросхемы памяти приведен на рисунке ниже.

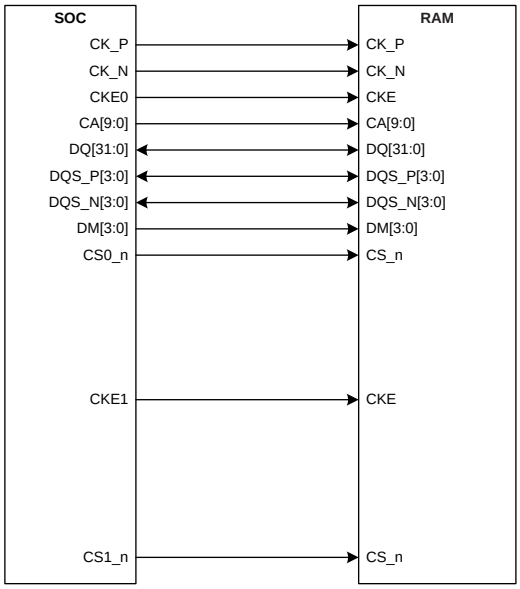


Рисунок 13.6 – Подключение одной микросхемы к СБИС СНП-ВП

При этом рекомендуется, чтобы параметры трассировки укладывались в следующие значения:

- длина линии: максимальная 50 мм, минимальная 35 мм;
- волновое сопротивление линии: Zo = 50 Ом, скорость распространения сигнала Td = 6.5 ns/m;
- разница в длинах дифференциального сигнала CK_P и CK_N = ±20 mils;
- разница в длинах внутри одной DQS группы: ±50 mils;
- разница в длинах внутри группы управляющих сигналов: ±50 mils;
- разница в длинах между группами: 100 mils;

Вариант подключения двух микросхем памяти приведен на рисунке ниже.

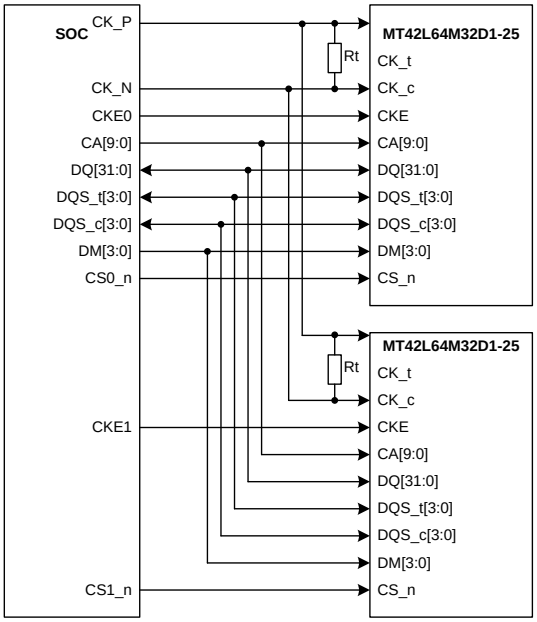


Рисунок 13.7 –Подключение двух микросхемы к СБИС СНП-ВП

При этом рекомендуется, чтобы параметры трассировки укладывались в следующие значения:

- длина линии до точки ветвления: максимальная 25 мм, минимальная 15 мм;
- длина линии после точки ветвления: максимальная 25 мм, минимальная 20 мм;
- волновое сопротивление линии до ветвления: Zo = 50 Ом, скорость распространения сигнала Td = 6.5 ns/m;
- волновое сопротивление линии после ветвления: Zo = 100 Ом, скорость распространения сигнала Td = 6.5 ns/m;
- Rt = 200 Ohm;
- разница в длинах дифференциального сигнала CK_P и CK_N = ±20 mils;
- разница в длинах внутри одной DQS группы: ±50 mils;
- разница в длинах внутри группы управляющих сигналов: ±50 mils;
- разница в длинах между группами: 100 mils.

Ниже приведены общие требования, соблюдение которых поможет выполнить вышеописанные требования. Для дифференциального тактового сигнала CK_P и CK_N старайтесь делать трассировку таким образом, чтобы сигналы имели одинаковую задержку, для этого:

- длина для сигналов CK_P и CK_N должна быть одинаковой;
- используйте один и тот же сигнальный слой для трассировки CK_P и CK_N. Это позволит минимизировать разброс волнового сопротивления для сигналов CL_P и CK_N;
- минимизируйте разброс параметров линий, делая одинаковое число переходных отверстий по каждой линии CK_P и CK_N;
- используйте одинаковое расстояние между проводниками (wiring pitch) для сигналов CK_P и CK_N. Это снизит разброс влияния перекрестных наводок.

Группируйте каждые 8 бит данных D, DQS_P, DQS_N и DM в одну группу:

DQS_P0/ DQS_N0 => DQ0 - DQ7, DM0 - первая группа
DQS_P1/ DQS_N1 => DQ8 - DQ15, DM1 - вторая группа
DQS_P2/ DQS_N2 => DQ16- DQ23, DM2 - третья группа
DQS_P3/ DQS_N3 => DQ24- DQ31, DM3 - четвертая группа

В каждой байтовой группе при трассировке печатной платы минимизируйте разницу задержек между каждым выводом группы DQ, D, DQS_P, DQS_N и DM, а затем добивайтесь минимальной разницы задержек между группами.

Трассировка сигналов CK_P/ CK_N и DQS_P/ DQS_N особенно важна, так как эти управляющие сигналы являются опорными для формирования остальных сигналов микросхемы. Экранируйте данные цепи шинами земли. Это снизит влияние перекрестных наводок от других сигнальных выводов. Аналогичные действия рекомендуются и для сигнала SCKE. Используйте всю свободную от проводников площадь сигнальных слоев для создания земляных шин и областей, при этом с помощью множественных переходных отверстий соедините их с внутренними земляными экранами печатной платы.

Для адресных сигналов и сигналов управления (CS, CKE) старайтесь делать трассировку таким образом, чтобы длина проводников этих сигналов была одинаковая.

Сигнал VREF определяет логический порог срабатывания для входных сигналов. Этот уровень должен иметь постоянное значение. Для обеспечения высокой стабильности VREF рекомендуется следующее:

- Экранируйте данные цепи шинами земли. Это снизит влияние перекрестных наводок от других сигнальных выводов.
- Используйте более широкие проводники для сигналов VREF. Это уменьшит сопротивление цепи и, соответственно, шум. Если возможно, используйте для этого отдельный слой.
- В зависимости от индуктивности или емкости корпуса, VREF может иметь LC резонансную частоту, на которой могут возникнуть колебания. Печатная плата должна быть спроектирована таким образом, чтобы можно было установить дополнительный внешний конденсатор для изменения резонансной частоты.

Используйте конденсаторы для снижения помех по питанию. Размещайте конденсаторы как можно ближе к выводам микросхемы. Используйте высоко- и низкочастотные конденсаторы на цепях питания.

Для одной DQ группы используйте один слой для трассировки.

При применении проводников змеевидной формы располагайте параллельные сегменты проводника на большем расстоянии (wiring pitch) друг от друга. Это минимизирует собственные перекрестные наводки, влияющие на задержку проводника.

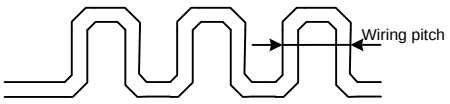


Рисунок 13.8 – Проводник змеевидной формы

В точках разветвления сигналов, подсоединенных к нескольким микросхемам (например, тактовый сигнал), может иметь место эффект отражения. Это возникает из-за различия в волновом сопротивлении. Для предотвращения отражения используйте подстроечный резистор для выравнивания волнового сопротивления.

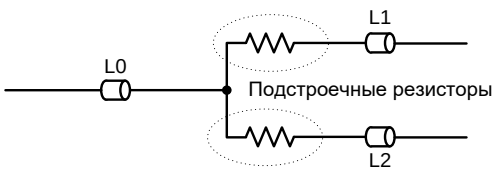


Рисунок 13.9 – Использование подстроечных резисторов

13.4 Рекомендации по реализации схемы начального сброса

Ниже приведены примеры реализации схемы сброса. Пунктиром на рисунках обозначены дополнительные элементы, которые можно поставить для повышения помехозащищенности на входе XTRST. Это решение нельзя применить к сбросу XSRST, т.к. он двунаправленный. Также необходимо помнить, что буфера XSRST и XTRST имеют встроенные pullup резисторы.

Минимальная длительность активного состояния сигнала сброса XRESET – 3 такта синхросигнала XCLK, т.е. около 40 нс. Минимальная длительность активного состояния сигнала сброса XSRST – 7 тактов внутреннего синхросигнала CCLK, т.е. около 25 нс.

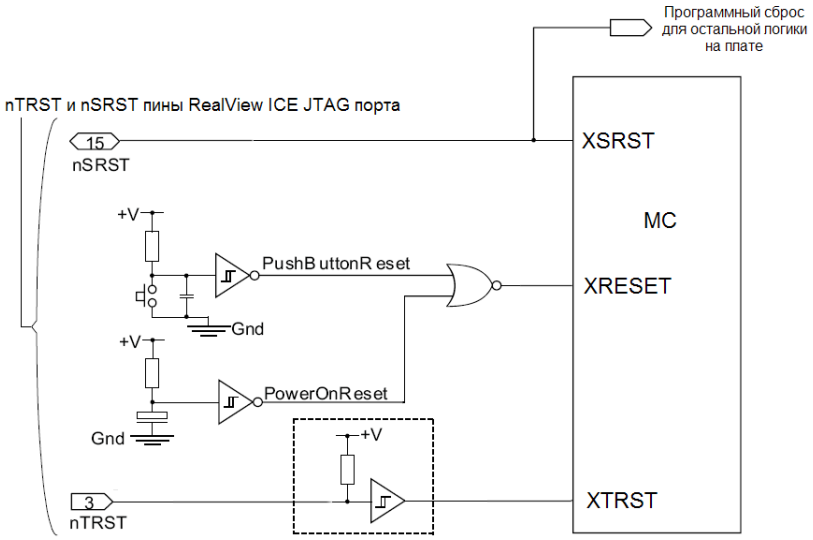


Рисунок 13.10 – Пример реализации схемы сброса

Возможно использовать специализированную микросхему монитора питания для реализации схемы сброса, например – MAX823:

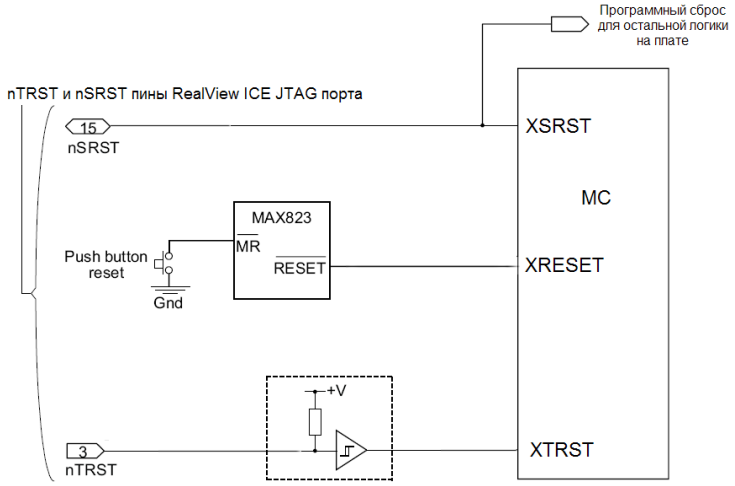


Рисунок 13.11 – Пример реализации схемы сброса с использованием микросхемы MAX823

13.5 Рекомендации по включению коммуникационных портов

При соединении коммуникационных портов следует уделять внимание согласованию линии. Чем больше частота работы коммуникационного порта, тем сильнее встает вопрос согласования. Выходное сопротивление буферов ввода-вывода коммуникационного порта составляет около 35 Ом. Ниже на рисунке приведен пример согласованной линии.

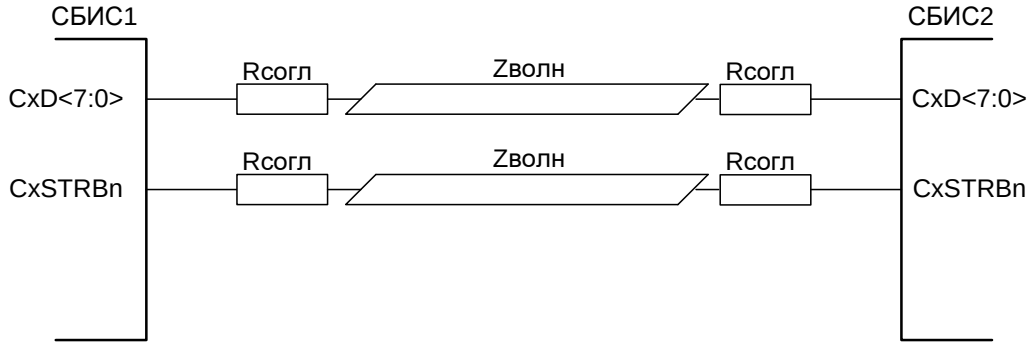


Рисунок 13.12 – Пример согласования коммуникационных портов

В данном примере значение согласующих резисторов Rсогл выбирается по формуле:

Rсогл = Zволн - Rбуф, где Zволн - волновое сопротивление линии, Rбуф - выходное сопротивление буферов-ввода вывода коммуникационного порта, которое составляет 35 Ом. Резисторы согласования должны располагаться как можно ближе к выводам микросхемы.

Лист регистрации изменений

Изм.	Номера страниц				Всего страниц в документе	№ документа	Входящий № сопроводительного докум. и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					
4		все				ЮФКВ.466-2015		Косоруков	29.12.2015