Удостоверен

ЮФКВ.431282.020РЭ-УД

Микросхема интегральная 1879ВМ8Я

Руководство по эксплуатации

ЮФКВ.431282.020РЭ

Инв.№подл.	Подп. и дата	Взам.инв.№	Инв.№дубл.	Подп. и дата	Справ.№
31406-3	24.04.2018	31406-2			

Содержание

1 Введение в архитектуру микросхемы 1879ВМ8Я 7
1.1 Основные отличительные особенности микросхемы 1879ВМ8Я 7
1.2 Основные характеристики и назначение микросхемы 8
1.2.1 Характеристики микросхемы 8
1.2.2 Области применения микросхемы 1879ВМ8Я 9
1.3 Общая структура микросхемы 1879ВМ8Я 10
1.3.1 Функциональный состав мультипроцессорной системы 10
1.3.2 Шинная структура и карта памяти микросхемы 1879ВМ8Я 13
1.3.3 Коммуникационные порты и система каналов межпроцессорного обмена 17
1.3.4 Система межпроцессорных прерываний 19
1.3.5 Функциональные выводы 21
2 Архитектура и структура процессорного ядра NMC4 29
2.1 Основные архитектурные особенности процессорного ядра NMC4 29
2.1.1 Обобщённая структура процессорной системы цифровой обработки сигналов на базе процессорного ядра NMC4 29
2.1.2 Статический VLIW 31
2.1.3 Многотактовые векторные команды и векторно-конвейерная организация вычислений (динамический VLIW) 33
2.1.4 Особенности работы конвейера команд при обмене данными с памятью 34
2.1.5 Единый адресный генератор процессорного ядра 36
2.1.6 Аппаратная вершина системного стека 36
2.1.7 Система прерываний 37
2.1.7.1 Типы прерываний 37
2.1.7.2 Внутренние и внешние прерывания процессорного ядра 38
2.2 Управляющее RISC-ядро 40
2.2.1 Структура RISC-ядра 40
2.2.2 Основные режимы работы RISC-ядра и методы адресации памяти 42
2.2.3 Регистр слова состояния процессора PSWR 42
2.2.4 Регистр запросов на прерывание INTR 43
2.3 Матрично-векторный сопроцессор арифметики с плавающей точкой 46
2.3.1 Базовые операции матрично-векторного сопроцессора 46
2.3.2 Структура матрично-векторного сопроцессора 47
2.3.3 Форматы векторных данных 49
2.3.3.1 Данные 49
2.3.3.2 64-разрядные слова упакованных данных в памяти 49
2.3.4 Центральный блок управления сопроцессором 50
2.3.5 Процессорная ячейка 55
2.3.6 Блок упаковки и распаковки данных 61
2.4 Система команд процессорного ядра NMC4 63
2.4.1 Форматы команд, задающих пересылку данных типа «регистр управляющего RISC-процессора – память» 65
2.4.2 Форматы команд пересылки данных типа “регистр-регистр» между регистрами управляющего RISC-процессора 67
2.4.3 Форматы команд модификации адресных регистров управляющего RISC-процессора 67
2.4.4 Форматы команд управления RISC-процессора 68

Удостоверен ЮФКВ.431282.020РЭ-УД

2.4.5 Форматы поля КОП СК (код операции скалярной команды управляющего RISC-процессора).....70
2.4.6 Формат поля КОП СК, задающего операцию сдвига.....71
2.4.7 Формат поля КОП СК, задающего логическую операцию.....73
2.4.8 Формат поля КОП СК, задающего арифметическую операцию.....74
2.4.9 Поле выбора адресного регистра управляющего RISC-процессора.....75
2.4.10 Поле $R_{ист/пр-к}$ в командах пересылки данных управляющего RISC-процессора.....76
2.4.11 Форматы команд, задающих пересылку данных типа «регистр сопроцессора арифметики с плавающей точкой - память».....79
2.4.12 Форматы команд пересылки данных типа "регистр-регистр» между регистрами сопроцессора арифметики с плавающей точкой.....79
2.4.13 Арифметические векторные команды сопроцессора арифметики с плавающей точкой.....80
2.4.14 Векторные команды ввода-вывода сопроцессора арифметики с плавающей точкой.....84
2.4.15 Векторные команды пересылки данных для сопроцессора арифметики с плавающей точкой...87
2.4.16 Поле $R_{ист/пр-к}$ в командах пересылки данных сопроцессора арифметики с плавающей точкой...89
3 Архитектура и структура вычислительного узла на базе процессорного ядра NMC4 (NMPU).....91
3.1 Структурная схема NMPU.....91
3.2 Карта памяти процессорного узла NMPU.....94
3.3 Системный интегратор и организация доступа в память.....95
3.3.1 Интерфейс системного интегратора с внешней памятью.....97
3.3.2 Интерфейс системного интегратора с периферийными устройствами.....98
3.4 Внутренняя память процессорного устройства NMPU.....98
3.5 Блок предвыборки команд IFU и управление кэш-памятью команд.....101
3.5.1 Программно доступные регистры блока управления кэш-памятью.....102
3.6 Блок защиты памяти MPU.....105
3.6.1 Разделение внутренней памяти NMPU на защищаемые сегменты.....107
3.6.2 Программно доступные регистры блока защиты памяти.....108
3.6.3 Прерывания.....110
3.7 Блок коммуникационного порта.....111
3.7.1 Приёмная часть блока коммуникационного порта.....113
3.7.1.1 Структура приемной части блока коммуникационного порта.....113
3.7.1.2 Принципы работы.....114
3.7.2 Передающая часть блока коммуникационного порта.....116
3.7.2.1 Структура передающей части блока коммуникационного порта.....116
3.7.2.2 Принципы работы.....117
3.7.3 Адресный генератор блока коммуникационного порта.....119
3.7.4 Программная модель передающей части блока коммуникационного порта.....121
3.7.5 Программная модель приёмной части блока коммуникационного порта.....125
3.7.6 Прерывания блока коммуникационного порта.....128
3.8 Мост "системный интегратор - AXI".....130
3.8.1 Особенности работы моста.....132
3.8.2 Эксклюзивные операции.....133
3.8.3 Ошибки обращения.....134
3.8.4 Программно доступные регистры моста "системный интегратор - AXI".....135
3.8.5 Прерывания.....138
3.9 Мост "системный интегратор - периферийная шина NMPU".....139
3.10 Блок периферийных устройств (PU) процессорного узла NMPU.....141
3.10.1 Общие сведения.....141
3.10.2 Системный контроллер NMPU.....141
3.10.2.1 Программно доступные регистры системного контроллера NMPU.....142
3.10.3 Контроллер внешних прерываний NMPU.....144
3.10.3.1 Внешние прерывания процессорной системы NMPU.....145
3.10.3.2 Режимы подтверждения запросов.....146
3.10.3.3 Приоритеты запросов.....147
3.10.3.4 Программно доступные регистры контроллера внешних прерываний.....147
3.10.4 Блок таймеров NMPU.....150
3.10.4.1 Программно доступные регистры блока таймеров.....151
3.10.4.2 Принцип работы таймеров.....152
4 Структура процессорного кластера PC.....154

4.1 Структурная схема процессорного кластера PC 154
4.1.1 Основные потоки данных в PC-кластере 158
4.1.2 Шинная структура процессорного кластера PC 159
4.2 Карта памяти CPU 160
4.3 Управляющий скалярный процессор ARM Cortex-A5 162
4.4 Внутренняя память кластера PC 164
4.4.1 Общие сведения о механизме эксклюзивного доступа 164
4.4.2 Общий порядок эксклюзивных операций 165
4.4.3 Эксклюзивные операции ядра ARM 166
4.4.4 Эксклюзивные операции NMPU 166
4.5 Контроллеры коммуникационных портов CP0…CP3 169
4.5.1 Программная модель передающей части контроллера коммуникационного порта 170
4.5.2 Программная модель приёмной части контроллера коммуникационного порта 171
4.5.3 Прерывания контроллера коммуникационного порта 171
4.6 Коммутаторы каналов связей LC0…LC3 172
4.6.1 Структура коммутатора каналов связи 172
4.6.2 Программно доступные регистры коммутаторов каналов связи 173
4.6.3 Коммутация портов 175
4.7 Контроллер ПДП память-память MDMAC 176
4.7.1 Программно доступные регистры контроллера MDMAC 176
4.7.1.1 Прерывания от контроллера MDMAC 180
4.8 Внешний высокоскоростной линк EL 181
4.8.1 Основные характеристики 181
4.8.2 Структура внешнего высокоскоростного линка EL 181
4.8.3 Контроллер линка EL 183
4.8.3.1 Общие сведения 183
4.8.3.2 Программно доступные регистры контроллера 184
4.8.3.3 Стандартные алгоритмы работы с MAC 191
4.8.3.4 Принципы работы MAC 193
4.8.4 Режимы работы 194
4.8.5 Сообщения 196
4.8.6 Ошибка при приеме 200
4.8.7 CRC_8 201
4.8.8 Сброс протокола 203
4.8.9 Блок Aligner 203
4.8.10 Блок DeSkewer 204
4.8.11 Типовые последовательности 205
4.9 Контроллер интерфейса с внешней динамической памятью EMI CPU 208
4.9.1 Программно доступные регистры интерфейса с внешней динамической памятью EMI CPU 209
4.10 Блок защиты памяти MPU 256
4.10.1 Разделение внутренней памяти на защищаемые сегменты 257
4.10.2 Программно доступные регистры блока защиты памяти 258
4.10.3 Прерывания 261
4.11 Контроллер прерываний GIC 263
4.11.1 Устройство системного контроллера прерываний 263
4.11.2 Запросы на прерывание 264
4.11.3 Интерфейс GIC с процессорным ядром ARM 267
4.11.4 Группы прерываний 268
4.11.5 Состояния обработки прерываний 268
4.11.6 Вложенные прерывания 269
4.11.7 Регистровая модель системного контроллера прерываний GIC 269
4.11.8 Порядок работы с GIC 284
4.12 Блок сдвоенных таймеров DIT 286
4.12.1 Устройство блока сдвоенных таймеров 286
4.12.2 Регистровая модель блока сдвоенных таймеров 287
4.12.3 Прерывания 294
4.13 Универсальный блок интерфейсов общего назначения GPIO/EXTINT 295
4.13.1 Устройство универсального блока интерфейсов общего назначения 295
4.13.2 Программно доступные регистры универсального блока интерфейсов общего назначения 297

4.13.3 Конфигурация регистров универсального блока интерфейсов общего назначения ... 298
4.13.4 Описание работы блока ... 300
4.13.4.1 Настройка выводов ... 300
4.13.4.2 Функционирование ... 300
4.14 Системный контроллер процессорного кластера CLSC ... 304
4.14.1 Программно доступные регистры системного контроллера CLSC ... 304
4.15 Генератор тактовых сигналов и сигналов сброса CRGH ... 307
4.15.1 Особенности CRGH ... 307
4.15.2 Программно доступные регистры генератора тактовых сигналов и сигналов сброса CRGH. 309
4.15.3 Особенности работы CRGH ... 312
4.15.3.1 APB-интерфейс ... 312
4.15.3.2 Генерация сигналов сброса ... 312
4.15.3.3 Генерация тактовых сигналов ... 317
5 Структура микросхемы 1879ВМ8Я ... 321
5.1 Структурная схема микросхемы 1879ВМ8Я ... 321
5.1.1 Основные потоки данных в микросхеме 1879ВМ8Я ... 324
5.1.2 Шинная структура микросхемы 1879ВМ8Я. ... 326
5.1.3 Карта памяти микросхемы 1879ВМ8Я ... 326
5.1.4 Система межпроцессорной функциональной синхронизации ... 331
5.1.4.1 Прерывания между вычислительными узлами 1879ВМ8Я ... 331
5.1.4.2 Синхронизация на основе ячеек общей памяти (семафоров) ... 332
5.2 Управляющий скалярный процессор ARM Cortex-A5 ... 333
5.3 Внутренняя память CCPU ... 333
5.4 Контроллер ПДП коммуникационных портов CP ... 333
5.5 Контроллер ПДП память-память MDMAC ... 333
5.6 Контроллер интерфейса с внешней динамической памятью EMI CCPU ... 333
5.7 Блок защиты памяти MPU ... 333
5.8 Системный контроллер прерываний GIC ... 333
5.9 Блок сдвоенных таймеров DIT ... 336
5.10 Универсальный блок интерфейсов общего назначения GPIO/EXTINT ... 336
5.11 Контроллер интерфейса ETHERNET ... 337
5.11.1 Обзор системы ... 338
5.11.2 Поддержка протоколов ... 338
5.11.3 Синхронизация ... 339
5.11.4 Интерфейс передатчика TxDMA ... 339
5.11.5 Интерфейс приемника RxDMA ... 342
5.11.6 MDIO интерфейс ... 344
5.11.7 Интерфейс для удаленной отладки (Ethernet Debug Communication Link) ... 345
5.11.8 RMII интерфейс ... 348
5.11.9 Программные драйверы ... 348
5.11.10 Регистровая модель контроллера GRETH ... 348
5.12 Высокоскоростной последовательный интерфейс PCIe ... 356
5.12.1 Устройство интерфейса PCIe ... 356
5.12.2 Программная модель ... 358
5.12.2.1 Описание программно доступных стандартных конфигурационных регистров (режим End Point) ... 366
5.12.2.2 Описание программно доступных стандартных конфигурационных регистров (режим Root Port) ... 397
5.12.2.3 Регистры локального управления ... 432
5.12.2.4 Регистры управления AXI портом ... 453
5.12.2.5 Регистры управления контроллером ПДП ... 458
5.12.3 Базовое преобразование адресов ... 466
5.12.3.1 Изменение адресов AXI шины ... 466
5.12.3.2 Изменение адресов PCIe шины ... 466
5.12.3.3 Встроенный контроллер прямого доступа ... 467
5.12.4 Управление прерываниями ... 472
5.12.4.1 Исходящие прерывания ... 472
5.12.4.2 Входящие прерывания ... 472
5.13 Системный контроллер CSC ... 474
5.13.1 Внешние выводы системного контроллера CSC ... 474
5.14 Контроллер синхронного последовательного интерфейса SPI ... 481
5.14.1 Внешние выводы SPI интерфейса ... 481

5.14.2 Программно доступные регистры контроллера SPI *482*
5.14.3 Протокол передачи данных в различных режимах работы SPI интерфейса *491*
5.14.3.1 Формат передачи данных спецификации Motorola SPI 491
5.14.3.2 Формат передачи данных спецификации Texas Instruments SPI 494
5.14.3.3 Формат передачи данных спецификации National Semiconductors Microwire 495
5.14.4 Работы контроллера ПДП интерфейса *496*
5.14.5 Прерывания контроллера SPI *498*
5.15 Сторожевой таймер WDT 500
5.15.1 Организация работы сторожевого таймера *500*
5.15.2 Программно доступные регистры блока сторожевого таймера *500*
5.16 Контроллер измерения температуры кристалла с температурным датчиком Temp Sens 505
5.16.1 Программно доступные регистры контроллера измерения температуры *505*
5.16.2 Работа с контроллером измерения температуры кристалла *508*
5.16.3 Прерывания контроллера измерения температуры на кристалле *509*
5.17 Загрузочное ПЗУ BROM и начальная загрузка микросхемы 1879ВМ8Я 510
5.17.1 Загрузка хост-контроллером *512*
5.17.2 Начальная загрузка через порт SPI *513*
5.17.3 Использование GPIOC для индикации состояния начального загрузчика *515*
5.18 Генераторы тактовых сигналов и сигналов сброса CRGL0…CRGL3 516
5.19 Генератор тактовых сигналов и сигналов сброса CCRG 518
5.20 Последовательный порт доступа для тестовых и отладочных данных DAP 519
5.20.1 Внешние выводы последовательного порта доступа для тестовых и отладочных данных DAP 519
5.20.2 Рекомендуемая схема подключения DAP *520*
6 Электрические, динамические и конструктивные характеристики микросхемы 1879ВМ8Я 523
6.1 Состав и расположение внешних выводов микросхемы 1879ВМ8Я 523
6.2 Конструктивные характеристики 538
6.3 Электрические характеристики 540
6.4 Временные характеристики 542
6.4.1 Временные диаграммы и временные параметры тактовых сигналов и сигналов общего назначения 542
6.5 Порядок подачи питания при включении 545

1 Введение в архитектуру микросхемы 1879ВМ8Я

Микросхема интегральная 1879ВМ8Я (далее по тексту – микросхема) представляет собой высокопроизводительную гетерогенную многопроцессорную систему на кристалле (СнК), в состав которой входят 16 процессорных ядер NeuroMatrix Core 4 и 5 ядер ARM Cortex-A5. Она предназначается для использования в качестве универсальной платформы для построения широкого класса систем цифровой обработки широкополосных сигналов в радиолокации, навигации и связи.

1.1 Основные отличительные особенности микросхемы 1879ВМ8Я

Отличительными особенностями микросхемы 1879ВМ8Я являются:

- Гетерогенный характер мультипроцессорной системы. Использование процессорных ядер NMC4 с отечественной архитектурой NeuroMatrix для достижения требуемой производительности в 512 GFLOP/s и процессорных ядер Cortex-A5 с системой команд ARM для управления системой.
- Иерархическое построение мультипроцессорной системы, состоящей из управляющего процессорного узла (ПУ) и 4 процессорных кластеров, каждый из которых в свою очередь состоит из управляющего ПУ и 4 вычислительных ПУ (NMPU) на базе процессорного ядра NMC4.
- Широкий доступ процессорных ядер к данным за счет большого количества банков внутренней памяти в каждом ПУ (4 банка в управляющем ПУ и 8 банков в NMPU) и наличия пяти 32-разрядных интерфейсов с внешней памятью типа DDR3.
- Использование контроллеров ПДП в каждом управляющем ПУ для эффективного обмена данными между внешними и внутренними банками памяти.
- Единое адресное пространство в 4 GB для всех процессоров СнК, что обеспечивается развитой системой AXI-шин и шинных коммутаторов.
- Развитая система широких каналов межпроцессорного обмена типа «точка-точка» для эффективной передачи информации между различными парами ПУ.
- Многоярусная система межпроцессорных прерываний для синхронизации вычислительных процессов, протекающих в различных ПУ.
- Аппаратная поддержка эксклюзивных операций при обращении к внутренней памяти любого ПУ, что позволяет организовать работу с общей памятью для нескольких процессорных ядер.

- Псевдо-двухпортовая организация банков внутренней памяти, позволяющая выполнять по два обращения каждом такте для большинства методов адресации.
- Система поддержки мониторинга выполнения и отладки программ с использованием стандартных отладочных средств фирмы ARM.

Высокопроизводительное процессорное ядро NMC4 с отечественной архитектурой NeuroMatrix, позволяющее выполнять до 32 операций за один процессорный такт благодаря реализации принципа динамического VLIW (Very Long Instruction Words) и векторно-конвеерного выполнения команд. Формат данных с плавающей точкой – 32 бита в соответствии со стандартом IEEE754. Также поддерживается обработка 64-разрядных данных в формате с плавающей точкой двойной точности, при этом достигается производительность до 8-ми операций за такт.

1.2 Основные характеристики и назначение микросхемы

Микросхема представляет собой высокопроизводительную гетерогенную многопроцессорную систему на кристалле. Она предназначена для использования в качестве основного или дополнительного вычислителя в вычислительных системах, интенсивно применяющих цифровую обработку сигналов. Его архитектура позволяет эффективно решать широкий круг задач, включая различные векторно-матричные вычислительные операции, вычисление преобразования Фурье, Адамара и прочих, цифровую фильтрацию, цифровую коммутацию. Микросхема может быть использована в качестве базового элемента при построении многопроцессорных параллельных вычислительных систем супер-производительности.

1.2.1 Характеристики микросхемы

В данном разделе приводятся основные характеристики, дающие представление об особенностях архитектуры и производительности микросхемы 1879ВМ8Я:

- Количество 32-разрядных процессорных ядер ARM Cortex A5 – 5.
- Рабочая частота процессоров Cortex A5- 800МГц.
- Количество процессорных ядер NMC4 с архитектурой NeuroMatrix- 16.
- Рабочая частота процессоров NMC4 - 1000МГц.
- Суммарный объем банков внутренней памяти – 76 Мбит.

- Пиковая производительность всех ядер NMC4 – до 512 GFLOPS/с в формате плавающей точки одинарной точности и 128 GFLOPS/с в формате плавающей точки двойной точности.
- Количество 32-разрядных интерфейсов с внешней памятью типа DDR3 – 5.
- Суммарная пропускная способность интерфейсов с внешней памятью - 256 Гбит/с.
- Количество внешних дуплексных интерфейсов межпроцессорного обмена – 5 х 4.
- Суммарная пропускная способность внешних интерфейсов межпроцессорного обмена – 160 Гбит/с.
- Количество внутренних 64-разрядных дуплексных каналов межпроцессорного обмена – 44.
- Суммарная пропускная способность внутренних каналов межпроцессорного обмена – 5120 Гбит/с.

1.2.2 Области применения микросхемы 1879ВМ8Я

Микросхема предназначена для обработки больших объемов сигнальной информации в режиме реального времени в составе встраиваемой вычислительной техники радиолокационных, навигационных и связных систем супер-производительности.

1.3 Общая структура микросхемы 1879ВМ8Я

1.3.1 Функциональный состав мультипроцессорной системы

Микросхема 1879ВМ8Я представляет собой высокопроизводительную гетерогенную многопроцессорную систему на кристалле (СнК), в состав которой входят 21 процессорный узел (ПУ), 5 интерфейсов с внешней памятью типа DDR3, интерфейс с хост-процессором на базе PCIe2.0 и четыре высокоскоростных линка для связи с внешними процессорными системами.

Каждый ПУ построен на базе процессорного ядра NMC4 или Cortex-A5 и включает в себя от 4 до 8 банков внутренней памяти объемом 16Kx32 бита каждый, 4 коммуникационных порта и ряд периферийных устройств. Система содержит 16 идентичных ПУ на базе NMC4, которые обеспечивают суммарную производительность системы в 512 GFLOP/s, и 5 ПУ на базе Cortex-A5, предназначенных для управления системой. Множество из 20 ПУ разбито на четыре одинаковых процессорных кластера по четыре ПУ на базе NMC4 и одного управляющего ПУ на базе Cortex-A5. К каждому кластеру подключена внешняя память типа DDR3. Еще один ПУ на базе Cortex-A5 осуществляет общее управление системой.

Рисунок 1.1 содержит общую структуру микросхемы 1879ВМ8Я. Здесь использованы следующие обозначения основных устройств системы:

NMPUij (NeuroMatrix Processing Unit) – j-й ПУ (j = 0,…,3) i-го кластера PCi (i = 0,…,3) на базе NMC4. Каждый такой узел содержит 8 банков внутренней памяти, работает на частоте 1000 MHz и может выполнять до 32 операций с плавающей точкой одинарной точности в каждом процессорном такте.

CPUi (Cortex Processing Unit) – управляющий ПУ i-го кластера PCi (i = 0,…,3) на базе Cortex-A5. Каждый такой узел содержит 4 банка внутренней памяти, работает на частоте 800 MHz, осуществляет управление элементами кластера, обеспечивает доступ к DDR3-памяти по 32-разрядной шине данных и обмен данными с внешними устройствами по высокоскоростному линку ELi.

PCi (Processing Cluster) – i-й процессорный кластер (i = 0,…,3), состоящий из 4-х ПУ на базе NMC и управляющего процессорного узла на базе Cortex-A5. Суммарная производительность кластера составляет 128 GFLOP/s, а суммарный объем внутренней памяти – 18 Мбит.

CCPU (Central Cortex Processing Unit) – центральный управляющий процессорный узел на базе Cortex-A5. Узел содержит 4 банка внутренней памяти, работает на частоте 800 MHz, осуществляет общее управление системой и обеспечивает доступ к DDR3-памяти по 32-разрядной шине данных.

DDRk – один из пяти (k = 0,…,4) интерфейсов с внешней памятью типа DDR3 с 32-разрядной шиной данных, обеспечивающей пропускную способность 6,4 ГБайта/с. Максимальный объем внешней памяти, подключаемой к такому интерфейсу, составляет 256M x 32 бит.

ELi (External Link) – внешний 4-проводной дуплексный линк, работающий на частоте 5 ГГц и предназначенный для обмена данными с другими процессорными системами. Каждый такой линк обеспечивает пропускную способность в 2 ГБайта/с в каждую сторону.

HI (Host Interface) – интерфейс с хост-процессором, реализованный на стандартном 4-канальном дуплексном PCIe2.0 интерфейсе.

Рисунок 1.1 – Основные процессорные и периферийные узлы микросхемы 1879ВМ8Я

Таким образом, пиковая производительность СнК составляет 516 GFLOPS/c, суммарная пропускная способность интерфейсов с внешней DDR-памятью составляет 32 ГБайта/с, а внешних линков – 20 ГБайт/с.

1879ВМ8Я имеет внутреннюю память суммарным объемом 76 Мбит и 5 интерфейсов с внешней DDR-памятью суммарным объемом до 40,96 Гбит. Вся эта память включена в единое адресное пространство, доступное всем процессорам системы.

Для обмена информацией между ПУ применяется два механизма:

- Программный доступ каждого процессора к любой ячейке общего адресного пространства, выполняемый по системе AXI-шин и шинных коммутаторов, соединяющей все процессоры и банки памяти СнК и построенной в соответствии с AMBA-протоколом.

- Обмен пакетами данных в режиме прямого доступа к памяти (ПДП) между любыми банками внутренней памяти, входящими в состав различных ПУ, через коммуникационные порты и систему каналов межпроцессорного обмена (КМО).

1.3.2 Шинная структура и карта памяти микросхемы 1879ВМ8Я

Система AXI-шин и шинных коммутаторов, соединяющей все процессорные узлы представлена ниже (см. Рисунок 1.2). Эта система построена в соответствии с AMBA-протоколом.

Каждый ПУ с ядром NMC (NMPUxx) имеет два 64-разрядных внешних порта (порт slave и порт master), подключаемых к AXI шинам и работающих на частоте процессора NMC (1000 MHz). Через порт master процессор NMC данного ПУ получает доступ ко всей внешней (относительно ПУ) памяти, а порт slave обеспечивает доступ других процессоров к внутренней памяти данного ПУ.

Каждый ПУ с ядром Cortex-A5 (CPUx), входящий в состав кластера, имеет две пары 64-разрядных внешних портов slave и master. Одна пара портов slave и master работает на частоте 1000 MHz и обеспечивает подключение ресурсов всех NMPUxx данного кластера к остальной системе. Вторая пара портов slave и master работает на частоте 800 MHz и обеспечивает подключение всех ресурсов данного кластера к остальной системе. Через порт master 1000 MHz управляющий процессор данного кластера или любой процессор, не входящий в состав этого кластера, получают доступ к внутренней памяти любого NMPUxx данного кластера. Порт slave 1000 MHz обеспечивает доступ любого процессора NMC данного кластера к внутренней памяти управляющего ПУ этого кластера, внешней DDR-памяти этого кластера или всей памяти, находящейся за пределами кластера. Через порт master 800 MHz любой процессор данного кластера получает доступ ко всей памяти, находящейся за пределами кластера. Порт slave 800 MHz обеспечивает доступ любого процессора, не входящего в состав этого кластера, ко всей памяти данного кластера включая его внешнюю DDR-память и внутреннюю память каждого ПУ кластера.

Рисунок 1.2 – Система межпроцессорных AXI-шин

Центральный ПУ с ядром Cortex-A5 (CCPU) также имеет пару 64-разрядных внешних портов slave и master, работающих на частоте 800 MHz. Через порт master центральный процессор Cortex-A5 получает доступ ко всей памяти, находящейся за пределами этого ПУ, а порт slave обеспечивает доступ других процессоров к внутренней или внешней DDR-памяти данного ПУ.

В каждом кластере имеется AXI-коммутатор, к которому подключены все высокоскоростные (1000 MHz) AXI-порты всех ПУ данного кластера. Данный коммутатор обеспечивает эффективный доступ каждого NMC к внутренней памяти соседних процессоров, входящих в состав этого кластера, а также обмен информацией между ресурсами NMPUxx данного кластера и другими ресурсами СнК.

В состав СнК входит центральный AXI-коммутатор, к которому подключены внешние AXI-порты всех процессорных кластеров, центрального ПУ и интерфейса PCIe2.0. Данный узел работает на частоте 800 MHz и обеспечивает обмен информацией между основными узлами системы.

Для обеспечения работоспособности представленной выше системы AXI-шин в СнК предусмотрено единое адресное пространство всех элементов памяти, входящих в систему или подключенных к ней. Рисунок 1.3 описывает карту памяти для различных процессоров СБИС.

Здесь для обозначения 6 старших областей памяти используются обозначения узлов, содержащие ячейки памяти с указанными адресами.

Оба используемых в СнК типа процессоров NMC и Cortex-A5 формируют 32-разрядные адреса. Однако они различаются дискретом адресации. NMC адресуют 32-разрядные слова, а в Cortex-A5 реализована байтовая адресация. Поэтому единое адресное пространство в СнК охватывает 4GB. Единый для обоих типов процессоров формат адреса совпадает с форматом адреса, формируемого Cortex-A5. Для перехода к этому единому формату адреса, генерируемые NMC, модифицируются следующим образом: добавляются два младших нулевых разряда, два старших разряда отбрасываются. При этом адресуемое NMC пространство сокращается с 4 до 1 G 32-разрядных слов.

Карта памяти микросхемы 1879ВМ8Я содержит 7 основных областей.

На центральный AXI-коммутатор поступают только адреса 6 старших областей памяти. Каждая из этих областей охватывает 512 MB внутренней памяти одного из 6 основных устройств, подключенных к центральному AXI-коммутатору. На рисунке 1.3 для обозначения этих областей используются имена соответствующих устройств (PCIe, CCPU, PC0, … , PC3). Поэтому адресное пространство в области трех старших GB одинаковое для любого типа процессора, используемого в СнК.

Область младших адресов в 1 GB отведена для адресации ячеек памяти, находящихся в том же основном устройстве, что и источник адреса. NMC, Cortex-A5, управляющий кластером, и центральный Cortex-A5 имеют различные организации области младших адресов, как показано на рисунке 1.3. Здесь использованы следующие обозначения отдельных полей памяти:

Рисунок 1.3 – Карта памяти микросхемы 1879ВМ8Я

NMMBi (16K x 32) – i-й банк внутренней памяти (i = 0,…,7) NMPU;

CMBj (16K x 32) – j-й банк внутренней памяти (j = 0,…,3) CPU ;

CCMBj (32K x 32) – j-й банк внутренней памяти (j = 0,…,3) CCPU;

REG_NMPU (128K x 32) – регистры внешних устройств NMPU;

REG_CPU (198K x 32) – регистры внешних устройств CPU;

REG_CCPU (198K x 32) – регистры внешних устройств CCPU;

NMPUj (128K x 32) – внутренняя память j-го NMPU кластера;

DDR_CPU (255M x 32) – DDR-память кластера;

DDR_CCPU ((256M-256K) x 32) – DDR-память центрального управляющего ПУ.

СнК рассчитана на применение в каждом кластере и центральном ПУ внешней DDR-памяти объемом до 256M x 32. Причем, при использовании DDR-памяти объемом 256M x 32 в кластере будут недоступны 1M младших ячеек, а в центральном ПУ – 256K младших ячеек.

В запросах, поступающих в основные узлы (CCPU, PC0, …, PC3) с центрального AXI-коммутатора 29–й разряд адреса принудительно обнуляется. В результате этого формат области памяти CCPU для всех процессоров совпадает с форматом младшей половины первой области центрального процессора CCPU, а формат каждой из четырех старших областей (PC0,…, PC3) совпадает с форматом младшей половины первой области управляющего процессора кластера CPU. Поэтому любому процессору при обращении к DDR-памяти другого кластера будет доступно не более 127M ее ячеек (младшие 1M ячеек недоступны по причине, указанной выше).

Еще одно ограничение в адресуемости заключается в том, что регистры внешних устройств в каждом NMPU доступны только своему процессору NMC.

1.3.3 Коммуникационные порты и система каналов межпроцессорного обмена

Для эффективного межпроцессорного обмена пакетами данных в СнК реализована сеть простых дуплексных каналов с высокой пропускной способностью. Каждый канал межпроцессорного обмена (КМО) обеспечивает соединение типа «точка-точка», связывает два ПУ и имеет очень простой состав шин. КМО содержит два симметричных набора шин, работающих на встречных курсах. Для каждого направления передачи используется 64-разрядная шина данных, сопровождающий ее сигнал строба, и встречный сигнал готовности к приему данных.

Каждый КМО соединяет два коммуникационных порта (КП), расположенных в разных ПУ. КП с одной стороны подключен к КМО, а с другой - к AXI-шине, через которую в

режиме ПДП осуществляется обмен пакетами данных с банками внутренней памяти ПУ. В состав КП входят интерфейс с КМО, интерфейс с портом master к AXI-шине, и два канала ПДП (на чтение и на запись). Каждый канал ПДП содержит счетчик текущего адреса, счетчик числа передаваемых слов, буфер данных и регистр управления каналом.

Рисунок 1.4. представляет схему межпроцессорного обмена в микросхеме 1879ВМ8Я.

Рисунок 1.4 – Схема каналов межпроцессорного обмена

В состав каждого ПУ входят четыре КП. Кроме того в состав управляющих ПУ на базе Cortex-A5 входят по четыре коммутатора КМО (LCxx). Состояние коммутаторов задается управляющим Cortex-A5, путем программирования регистра состояния, входящего в состав коммутатора. К каждому из трёх коммутаторов LCx1- LCx3 подключены три КМО – один от КП данного ПУ и два внешних. Коммутатор имеет три состояния, в каждом из которых осуществляется соединение двух КМО.

К четвёртому коммутатору LCx0, помимо трёх КМО, подключён внешний высокоскоростной порт ELCx. Каждый такой внешний порт имеет 4 внешних дуплексных линии, реализован на основе физической части интерфейса PCIe2.0 и обеспечивает пропускную способность 2GB/s в каждом направлении. Эти порты предназначены для обмена информацией с другими микросхемами при построении различных супер-производительных многокристальных систем.

Каждый NMPU подключен к четырем КМО, через которые он напрямую связан со всеми ПУ своего кластера.

Каждый CPU, управляющий работой кластера, через свои коммутаторы подключен к восьми КМО – четыре используются для связи с NMPU своего кластера, а четыре других для связи с другими управляющими ПУ.

Центральный управляющий ПУ (CCPU) так же подключен к четырём КМО для связи со всеми CPU.

КМО, подключенные к NMPU работают на частоте 1000 MHz, а каналы, подключенные к управляющим ПУ (CPUx, CCPU) - 800 MHz.

1.3.4 Система межпроцессорных прерываний

Для синхронизации вычислительных процессов, протекающих в различных ПУ, предусмотрена многоярусная система межпроцессорных прерываний (см. Рисунок 1.5).

По своей организации данная система напоминает описанную выше систему КМО с той разницей, что вместо каждого КМО используется две встречных пары сигналов прерывания, различающихся по уровню приоритета. Эти сигналы формируются путем программной установки определенных бит в специальном внешнем регистре одного ПУ и поступают в контроллер прерываний другого ПУ.

Каждый NMPU формирует 4 пары прерываний, поступающих на другие ПУ своего кластера, и принимает 4 пары прерываний от этих же ПУ.

Каждый ПУ, управляющий кластером, получает 4 пары прерываний от NMPU своего кластера и 4 пары прерываний от других управляющих ПУ. Он же в ответ выдает 4 пары прерываний для NMPU своего кластера и 4 пары прерываний для других управляющих ПУ.

Центральный ПУ (CCPU) формирует 4 пары прерываний для ПУ, управляющих кластером, и принимает 4 пары прерываний от этих же ПУ, а также 8 внешних прерываний с входов микросхемы.

Рисунок 1.5 – Схема межпроцессорных прерываний

1.3.5 Функциональные выводы

Микросхема 1879ВМ8Я имеет 568 функциональных выводов, назначение которых приведено в таблице 1-1.

Таблица 1.1 - Функциональные выводы микросхемы 1879ВМ8Я

Обозначение [1)]	Кол-во	Тип[2)]	Функциональное назначение
Общее управление (8 выводов)			
CLK_REFI	1	I	Вход опорного тактового сигнала
CLK_REFO	1	O	Выход опорного тактового сигнала
XRSTn	1	I	Системный сброс
BOOTM0, BOOTM1, BOOTM2	3	I	Режим начальной загрузки и тестирования процессора
TM	1	I	Режим тестирования процессора
VPD	1	I	Управление статическим током
Управление спящим режимом процессора (2 вывода)			
WKUPRQ	1	I	Запрос на вывод процессора из спящего режима
WKUPAK	1	O	Разрешение на снятие запроса на вывод процессора из спящего режима
JTAG интерфейс (5 выводов)			
BS_TDO	1	O(Z)	Выход данных тестового порта JTAG
BS_TDI	1	I	Вход данных тестового порта JTAG
BS_TCK	1	I	Тактовый сигнал тестового порта JTAG
BS_TMS	1	I	Выбор режима тестирования JTAG
BS_TRSTn	1	I	Сброс тестового порта JTAG
Debug интерфейс (6 выводов)			
TDO_SWO	1	O(Z)	Выход данных порта JTAG отладочного модуля
TDI	1	I	Вход данных тестового порта JTAG
TCK_SWCLK	1	I	Тактовый сигнал тестового порта отладочного модуля
TMS_SWDIO	1	I/O	Выбор режима работы отладочного модуля
TRSTn	1	I	Сброс порта JTAG отладочного модуля
SRSTn	1	I/O	Сброс отладочного модуля

Продолжение таблицы 1.1

Обозначение [1)]	Кол-во	Тип[2)]	Функциональное назначение
Выводы SPI интерфейса (7 выводов)			
SPI_CLK	1	O	Синхросигнал SPI интерфейса
SPI_TXD	1	O	Передаваемые данные
SPI_RXD	1	I	Принимаемые данные
SPI_CS0n	1	O	Выбор приемника 0
SPI_CS1n	1	O	Выбор приемника 1
SPI_CS2n	1	O	Выбор приемника 2
SPI_CS3n	1	O	Выбор приемника 3
Выводы Ethernet интерфейса (14 выводов)			
EDCLA0, EDCLA1, EDCLA2, EDCLA3	4	I	Выводы конфигурации MAC адреса
ETH_ENABLE	1	I	Конфигурационный вход наличия микросхемы физического уровня
ETH_TXD0, ETH_TXD1	2	O	Выходные данные интерфейса
ETH_TXEN	1	O	Разрешение передачи данных
ETH_RXD0, ETH_RXD1	2	I	Входные данные интерфейса
ETH_CRSDV	1	I	Признак обнаружения несущей
ETH_REFCLK	1	1	Входной опорный тактовый сигнал 50МГц.
ETH_MDC	1	O	Выходной тактовый сигнал для микросхемы физического уровня
ETH_MDIO	1	I/O	Шина управления микросхемой физического уровня
Выводы Watch DOG таймера (1 вывод)			
WDT	1	O	Выход сторожевого таймера
Выводы портов общего назначения управляющей части (8 выводов)			
GPIOC0, …, GPIOC7	8	I/O	Программируемы входы/выходы

Продолжение таблицы 1.1

Обозначение [1]	Кол-во	Тип[2]	Функциональное назначение
Выводы DDR контроллера управляющей части (76 выводов)			
DDRC_CK, DDRC_CKn	2	O	Тактовый сигнал DDR памяти
DDRC_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления
DDRC_A0, …, DDRC_A14	15	O	Шина адреса памяти
DDRC_BA0, …, DDRC_BA2	3	O	Выбор адресуемых банков в микросхемах памяти
DDRC_CSn	1	O	Выбор внешней памяти
DDRC_RASn	1	O	Строб адреса строки внешней памяти
DDRC_CASn	1	O	Строб адреса столбца внешней памяти
DDRC_WEn	1	O	Признак операции записи
DDRC_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти
DDRC_DM0, …, DDRC_DM3	4	O	Сигналы маскирования байтов данных при операциях записи
DDRC_DQ0, …, DDRC_DQ31	32	I/O	Шина данных
DDRC_DQS0, DDRC_DQS0n, …, DDRC_DQS3, DDRC_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных
DDRC_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти
DDRC_ATB0, DDRC_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса
DDRC_CAL0	1	I	Вывод для подключения калибровочного резистора
DDRC_PLL_TESTO_P, DDRC_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса
Выводы контроллера PCI Express (21 вывод)			
PCI_CLKREF_P, PCI_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера PCI Express
PCI_TXP0, PCI_TXN0, …, PCI_TXP3, PCI_TXN3	8	O	Парафазные выходы данных
PCI_RXP0, PCI_RXN0,…, PCI_RXP3, PCI_RXN3	8	I	Парафазные входы данных
PCI_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)
PCI_ ATB0, PCI_ ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса

Продолжение таблицы 1.1

Обозначение [1]	Кол-во	Тип[2]	Функциональное назначение
Выводы кластера 0			
Выводы портов общего назначения (8 выводов)			
GPIO00, …, GPIO07	8	I/O	Программируемы входы/выходы
Выводы DDR контроллера (76 выводов)			
DDR0_CK, DDR0_CKn	2	O	Тактовый сигнал DDR памяти
DDR0_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления
DDR0_A0, …, DDR0_A14	15	O	Шина адреса памяти
DDR0_BA0, …, DDR0_BA2	3	O	Выбор адресуемых банков в микросхемах памяти
DDR0_CSn	1	O	Выбор внешней памяти
DDR0_RASn	1	O	Строб адреса строки внешней памяти
DDR0_CASn	1	O	Строб адреса столбца внешней памяти
DDR0_WEn	1	O	Признак операции записи
DDR0_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти
DDR0_DM0, …, DDR0_DM3	4	O	Сигналы маскирования байтов данных при операциях записи
DDR0_DQ0, …, DDR0_DQ31	32	I/O	Шина данных
DDR0_DQS0, DDR0_DQS0n, …, DDR0_DQS3, DDR0_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных
DDR0_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти
DDR0_ATB0, DDR0_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса
DDR0_CAL0	1	I	Вывод для подключения калибровочного резистора
DDR0_PLL_TESTO_P, DDR0_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса
Выводы последовательного порта кластера (21 вывод)			
CP0_CLKREF_P, CP0_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера PCI Express
CP0_TXP0, CP0_TXN0, …, CP0_TXP3, CP0_TXN3	8	O	Парафазные выходы данных
CP0_RXP0, CP0_RXN0,…, CP0_RXP3, CP0_RXN3	8	I	Парафазные входы данных
CP0_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)
CP0_ ATB0, CP0_ ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса

Продолжение таблицы 1.1

Обозначение [1]	Кол-во	Тип[2]	Функциональное назначение
Выводы кластера 1			
Выводы портов общего назначения (8 выводов)			
GPIO10, …, GPIO17	8	I/O	Программируемы входы/выходы
Выводы DDR контроллера (76 выводов)			
DDR1_CK, DDR1_CKn	2	O	Тактовый сигнал DDR памяти
DDR1_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления
DDR1_A0, …, DDR1_A14	15	O	Шина адреса памяти
DDR1_BA0, …, DDR1_BA2	3	O	Выбор адресуемых банков в микросхемах памяти
DDR1_CSn	1	O	Выбор внешней памяти
DDR1_RASn	1	O	Строб адреса строки внешней памяти
DDR1_CASn	1	O	Строб адреса столбца внешней памяти
DDR1_WEn	1	O	Признак операции записи
DDR1_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти
DDR1_DM0, …, DDR1_DM3	4	O	Сигналы маскирования байтов данных при операциях записи
DDR1_DQ0, …, DDR1_DQ31	32	I/O	Шина данных
DDR1_DQS0, DDR1_DQS0n, …, DDR1_DQS3, DDR1_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных
DDR1_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти
DDR1_ATB0, DDR1_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса
DDR1_CAL0	1	I	Вывод для подключения калибровочного резистора
DDR1_PLL_TESTO_P, DDR1_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса
Выводы последовательного порта кластера (21 вывод)			
CP1_CLKREF_P, CP1_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера PCI Express
CP1_TXP0, CP1_TXN0, …, CP1_TXP3, CP1_TXN3	8	O	Парафазные выходы данных
CP1_RXP0, CP1_RXN0,…, CP1_RXP3, CP1_RXN3	8	I	Парафазные входы данных
CP1_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)
CP1_ATB0, CP1_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса

Продолжение таблицы 1.1

Обозначение [1]	Кол-во	Тип[2]	Функциональное назначение
Выводы кластера 2			
Выводы портов общего назначения (8 выводов)			
GPIO20, …, GPIO27	8	I/O	Программируемы входы/выходы
Выводы DDR контроллера (76 выводов)			
DDR2_CK, DDR2_CKn	2	O	Тактовый сигнал DDR памяти
DDR2_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления
DDR2_A0, …, DDR2_A14	15	O	Шина адреса памяти
DDR2_BA0, …, DDR2_BA2	3	O	Выбор адресуемых банков в микросхемах памяти
DDR2_CSn	1	O	Выбор внешней памяти
DDR2_RASn	1	O	Строб адреса строки внешней памяти
DDR2_CASn	1	O	Строб адреса столбца внешней памяти
DDR2_WEn	1	O	Признак операции записи
DDR2_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти
DDR2_DM0, …, DDR2_DM3	4	O	Сигналы маскирования байтов данных при операциях записи
DDR2_DQ0, …, DDR2_DQ31	32	I/O	Шина данных
DDR2_DQS0, DDR2_DQS0n, …, DDR2_DQS3, DDR2_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных
DDR2_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти
DDR2_ATB0, DDR2_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса
DDR2_CAL0	1	I	Вывод для подключения калибровочного резистора
DDR2_PLL_TESTO_P, DDR2_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса
Выводы последовательного порта кластера (21 вывод)			
CP2_CLKREF_P, CP2_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера PCI Express
CP2_TXP0, CP2_TXN0, …, CP2_TXP3, CP2_TXN3	8	O	Парафазные выходы данных
CP2_RXP0, CP2_RXN0,…, CP2_RXP3, CP2_RXN3	8	I	Парафазные входы данных
CP2_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)
CP2_ATB0, CP2_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса

Продолжение таблицы 1.1

Обозначение [1)]	Кол-во	Тип[2)]	Функциональное назначение
Выводы кластера 3			
Выводы портов общего назначения (8 выводов)			
GPIO30, …, GPIO37	8	I/O	Программируемы входы/выходы
Выводы DDR контроллера (76 выводов)			
DDR3_CK, DDR3_CKn	2	O	Тактовый сигнал DDR памяти
DDR3_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления
DDR3_A0, …, DDR3_A14	15	O	Шина адреса памяти
DDR3_BA0, …, DDR3_BA2	3	O	Выбор адресуемых банков в микросхемах памяти
DDR3_CSn	1	O	Выбор внешней памяти
DDR3_RASn	1	O	Строб адреса строки внешней памяти
DDR3_CASn	1	O	Строб адреса столбца внешней памяти
DDR3_WEn	1	O	Признак операции записи
DDR3_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти
DDR3_DM0, …, DDR3_DM3	4	O	Сигналы маскирования байтов данных при операциях записи
DDR3_DQ0, …, DDR3_DQ31	32	I/O	Шина данных
DDR3_DQS0, DDR3_DQS0n, …, DDR3_DQS3, DDR3_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных
DDR3_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти
DDR3_ATB0, DDR3_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса
DDR3_CAL0	1	I	Вывод для подключения калибровочного резистора
DDR3_PLL_TESTO_P, DDR3_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса

Продолжение таблицы 1.1

Обозначение [1]	Кол-во	Тип[2]	Функциональное назначение
Выводы последовательного порта кластера (21 вывод)			
CP3_CLKREF_P, CP3_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера PCI Express
CP3_TXP0, CP3_TXN0, …, CP3_TXP3, CP3_TXN3	8	O	Парафазные выходы данных
CP3_RXP0, CP3_RXN0,…, CP3_RXP3, CP3_RXN3	8	I	Парафазные входы данных
CP3_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)
CP3_ATB0, CP3_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса

Примечания:

1) Для обозначения выводов с активным низким уровнем сигнала в качестве последнего символа имени используется префикс «n».
2) Используемые обозначения типов выводов:
   I – вход,
   O – выход,
   I/O – двунаправленный вывод.

2 Архитектура и структура процессорного ядра NMC4

2.1 Основные архитектурные особенности процессорного ядра NMC4

2.1.1 Обобщённая структура процессорной системы цифровой обработки сигналов на базе процессорного ядра NMC4

Требование обработки потоков данных в реальном времени с производительностью, сравнимой с производительностью универсальных процессоров, и потреблением, характерным для встроенных систем, привело к созданию в ЗАО НТЦ «Модуль» архитектуры NeuroMatrix, обладающей следующими основными особенностями:

векторно-конвейерный принцип выполнения операций, позволяющий одним потоком команд задавать большое количество параллельно выполняемых операций (динамический VLIW);

наличие одного или нескольких векторных сопроцессоров, работающих под управлением одного RISC-процессора и имеющих свои шины ввода/вывода данных;

использование внешних адресных генераторов, что обеспечивает эффективное использование адресных регистров ядра при адресации векторных данных;

конвейер с очередью команд на ступени выборки операндов из памяти, обеспечивающий эффективную работу с банками внутренней и внешней памяти, имеющими различную глубину конвейера;

выдача и приём данных на одной и той же ступени конвейера (Late Write), что резко снижает количество конфликтов по данным;

использование команд, задающих несколько операций (статический VLIW).

Обобщенная структура процессора цифровой обработки сигналов на базе процессорного ядра NMCx (NeuroMatrix Core), обладающая всеми вышеперечисленными свойствами, приведена на рисунке 2.1.

Рисунок 2.1 - Обобщенная структура процессора цифровой обработки сигналов на базе процессорного ядра NeuroMatrix

В состав процессора цифровой обработки сигналов входят следующие основные блоки:

NMCx – процессорное ядро NeuroMatrix, содержащее управляющий RISC-процессор и несколько сопроцессоров, которые выполняют цифровую обработку данных, предварительно загруженных в память системы.

RISC-процессор решает следующие основные задачи:

декодирует и выполняет команды, считываемые из памяти по шине команд (Instruction Bus),

осуществляет через шину скалярных данных (Scalar Data Bus) конфигурирование сопроцессоров, настраивая их на обработку данных,

выставляет на внутреннюю шину инструкций (Internal Instruction Bus) команды оперативного управления сопроцессорами,

формирует и выставляет на адресную шину (Address Bus) адреса команд, а также скалярных и векторных данных.

Internal Memory Bank – банки внутренней памяти.

External Memory Interface – интерфейсы с внешней памятью, подключенной к соответствующим внешним шинам системы.

Instruction Address Generator – генератор последовательных адресов команд, содержимое которого может быть изменено произвольным образом при выполнении различных команд перехода.

Data Address Generator – генераторы адресов векторных данных, загружаемые через адресную шину начальным адресом, смещением адреса и количеством повторов операции ввода/вывода.

Address Switch – коммутатор адресов, который осуществляет пересылку адресов, формируемых адресными генераторами, и адресов, поступающих от внешних каналов прямого доступа к памяти, в банки внутренней памяти и в интерфейсы с внешней памятью.

Data Bus Switch – коммутатор шин данных, обеспечивающий обмен данными между процессорным ядром и памятью системы, а также обмен данными между каналами DMA и банками памяти.

Instruction Fetch Unit – блок выборки команд, который выстраивает в единую очередь команды, считываемые из внутренней или внешней памяти системы.

Peripheral Bus Interface – интерфейс с шиной периферийных устройств, через которую процессор осуществляет инициализацию, настройку и оперативное управление внешними устройствами, а также обмен данными с различными портами ввода/вывода информации.

Количество и тип сопроцессоров, количество банков внутренней памяти и внешних шин системы, а также количество генераторов адресов векторных данных определяются требуемой производительностью системы, что позволяет легко наращивать производительность вычислительных систем в зависимости от класса решаемых задач с сохранением программной совместимости внутри семейства процессоров.

2.1.2 Статический VLIW

Команды процессора цифровой обработки сигнала делятся на две основные группы: команды управляющего RISC-процессора (см. Рисунок 2.2) и команды сопроцессоров (см. Рисунок 2.3). Каждая из этих групп, в свою очередь, делится на команды скалярные, т.е. обычные RISC-команды, и векторные, которые задают многократно одни и те же действия над векторами данных, что эквивалентно аппаратной поддержке коротких циклов.

31	30 … 16	15 … 0
0	M O V E	O P E R

63 … 32
C O N S T

а) Кодировка скалярных команд управляющего RISC-процессора

31	30 … 18	17 … 13	12 … 0
0	V M O V E	COUNT	V O P E R

б) Кодировка векторных команд управляющего RISC-процессора

Рисунок 2.2 – Кодировка команд управляющего RISC-процессора

31	30 … 16	15 … 0
1	VPMOVE	OPER

63 … 32
C O N S T

а) Кодировка скалярных команд сопроцессора арифметики с плавающей точкой

31	30 … 18	17 … 13	12 … 0
1	VPMOVE	COUNT	VPOPER

б) Кодировка векторных команд сопроцессора арифметики с плавающей точкой

Рисунок 2.3 – Кодировка команд сопроцессоров

Скалярные команды управляющего RISC-процессора содержат следующие поля: MOVE – задаёт ввод/вывод данных с одновременной модификацией адресных регистров, условные переходы/переходы к подпрограмме и возвраты из подпрограммы/прерывания; OPER – определяет арифметическую, логическую операцию или операцию сдвига. Эти команды могут использовать 32-разрядные константы (поле CONST), которые могут грузиться в регистры или использоваться для задания адреса или смещения при обращении к памяти.

Векторные команды управляющего RISC-процессора имеют похожие поля: VMOVE – задаёт ввод/вывод векторных данных, VOPER – определяет арифметическую или логическую операцию над векторными данными, COUNT – определяет число повторов выполнения данной команды (от 1 до 32), что позволяет аппаратно поддержать организацию коротких циклов и значительно увеличить плотность кода.

Команды сопроцессоров имеют следующие отличия от команд управляющего RISC-процессора:

- Нельзя задать команды управления (переходы/переходы к подпрограмме и возвраты из подпрограммы/прерывания).
- Ввод/вывод данных осуществляется только в регистры сопроцессора арифметики с плавающей точкой в соответствии с полем VPMOVE, но адресация в память осуществляется по-прежнему управляющим RISC-процессором.
- Векторные арифметические операции в формате с плавающей точкой задаются полем VPOPER, причём данные арифметические операции не совмещаются с вводом/выводом векторных данных.

В остальном скалярные и векторные команды сопроцессоров и соответствующие команды управляющего RISC-процессора задают аналогичные операции.

Таким образом, процессор использует команды типа VLIW, задающие одновременно операции обмена с памятью, модификацию адресных регистров и арифметическую операцию, причём это относится как к скалярным, так и к векторным командам обработки данных в формате с фиксированной точкой. Объединение в одной команде операций ввода-вывода и арифметической операции позволяет увеличить производительность скалярных команд на реальных задачах до 40%, но для векторных команд это решение, по нашим оценкам, особый выигрыш не даёт. Поэтому для команд сопроцессора арифметики с плавающей точкой такого типа совмещения нет.

В данной реализации, поскольку отсутствует сопроцессор с фиксированной точкой, векторные команды управляющего RISC-процессора не используются. При попытке выполнить такую команду вместо неё реально выполняется команда NOP (нет операции) и в процессорном ядре формируется соответствующее прерывание по неправильной команде.

2.1.3 Многотактовые векторные команды и векторно-конвейерная организация вычислений (динамический VLIW)

Принцип организации выполнения одновременно нескольких операций, заданных разными векторными командами, в разных функциональных узлах одного или нескольких сопроцессоров показан на рисунке (см. Рисунок 2.4). Несмотря на то, что команды поступают на выполнение по одной и в строгой последовательности, за счёт использования многотактовых команд достигается эффект супер-скаляра. Также поддерживается внеочередное выполнение команд (out-of-order execution).

Поток команд	Последовательности выполняемых операций			
	Функциональный узел 1	Функциональный узел 2	Функциональный узел 3	…
Команда 1	1.1			
Команда 2	1.2	2.1		
Команда 3	1.3	2.2	3.1	
	1.4	2.3	3.2	
⋮	⋮	⋮	⋮	
	1.N	2.N-1	3.N-2	
Команда M	M.1	2.N	3.N-1	
Команда M+1	M.2	M+1.1	3.N	
Команда M+2	M.3	M+1.2	M+2.1	
⋮	⋮	⋮	⋮	

Время

Рисунок 2.4 - Векторно-конвейерная организация вычислений (динамический VLIW)

2.1.4 Особенности работы конвейера команд при обмене данными с памятью

Принципы организации конвейера команд процессора цифровой обработки сигналов на базе процессорного ядра NMC4 показан на рисунке 2.5. Его основными особенностями являются:

- наличие общих первой и второй ступеней для всех команд, причем на первой ступени осуществляется вычисление адреса первого данного для команды, последнего адреса и смещения (для векторных команд), модификации адресных регистров, на второй организуется единая очередь команд, ожидающих своих данных перед выполнением;
- несколько параллельных подконвейеров на третьей ступени (стадии выполнения операций), причём ввод и вывод данных осуществляется именно на данной ступени (реализуется Late Write).

Данный конвейер позволяет обеспечить эффективную работу с банками внутренней и внешней памяти, имеющими различную глубину конвейера без потери производительности. Реализация выдачи и приёма данных на одной и той же ступени конвейера (Late Write) резко снижает количество конфликтов по данным.

Рисунок 2.5 - Конвейер команд процессора цифровой обработки сигналов на базе процессорного ядра NeuroMatrix

2.1.5 Единый адресный генератор процессорного ядра

Как было описано ранее, все команды при запуске на выполнение, как и раньше, начинают свою работу на общей части первого уровня конвейера. Скалярная команда на данной ступени конвейера выставляет запрос на доступ в память. В случае отсутствия блокировок по доступу в память и от нижних ступеней конвейера команда попадает на вторую ступень конвейера и освобождает общую часть.

Векторная команда при запуске занимает общую часть первого уровня конвейера на один процессорный такт. За это время она вычисляет адрес первого обращения в память. Одновременно на специальном арифметическом устройстве, содержащем умножитель 5*32 разряда и 32-разрядный сумматор, формируется новое значение адресного регистра, использующегося в качестве базы. Это значение совпадает с тем, что должно получиться после завершения выполнения векторной команды. В следующем такте векторная команда освобождает первый уровень конвейера, уходит на второй уровень и одновременно занимает один из адресных генераторов, находящихся вне процессорного ядра. При этом запоминается адрес первого обращения в память, сохраняется копия значения регистра- смещения и формируется запрос на доступ в память. Далее внешний адресный генератор работает самостоятельно, не занимая ресурсов процессорного ядра.

Наличие единого адресного генератора у процессорного ядра позволяет использовать один и тот же адресный регистр этого ядра в качестве источника адреса для нескольких команд, при этом следующая команда не должна ждать полного окончания предыдущей. При этом блок адресных генераторов вне процессорного ядра обеспечивает генерацию запросов на обмен с памятью для нескольких скалярных и векторных команд одновременно.

2.1.6 Аппаратная вершина системного стека

В процессорном ядре NMC4 имеется аппаратная вершина системного стека. Она копирует содержимое ячейки памяти системного стека, хранящей последний адрес возврата из подпрограммы или прерывания и значение регистра слова состояния процессора PSWR при входе в подпрограмму/прерывание. Поддержка аппаратной вершины системного стека в NMC4 позволяет резко ускорить выполнение команд возврата из подпрограммы или прерывания, особенно если системный стек расположен во внешней по отношению к ядру памяти.

2.1.7 Система прерываний

2.1.7.1 Типы прерываний

Процессорное ядро NMC4 имеет развитую систему прерываний, позволяющую оперативно реагировать на внешние и внутренние события, а также управлять различными периферийными устройствами. Прерывания NMC4 представлены в таблице 2.1 в порядке уменьшения приоритета сверху вниз. Исключение составляет пошаговое прерывание, которое имеет наименьший приоритет.

Таблица 2.1 - Прерывания процессорного ядра NMC4

№	Прерывание	Обозначение	Адрес вектор
1	Немаскируемое прерывание	NMI	0x0000_0000 hex
2	Переполнение	OF	0x0000_0008 hex
3	Неправильная команда	EI	0x0000_0010 hex
4	Пошаговое прерывание	ST	0x0000_0018 hex
5	Прерывание 0 от сопроцессора арифметики с плавающей точкой (некорректные данные)	FPUI0	0x0000_0020 hex
6	Прерывание 1 от сопроцессора арифметики с плавающей точкой (overflow)	FPUI1	0x0000_0028 hex
7	Прерывание 2 от сопроцессора арифметики с плавающей точкой (underflow)	FPUI2	0x0000_0030 hex
8	Прерывание 3 от сопроцессора арифметики с плавающей точкой (потеря значимости)	FPUI3	0x0000_0038 hex
9	Прерывание 4 от сопроцессора арифметики с плавающей точкой (потеря данных)	FPUI4	0x0000_0040 hex
10	Прерывание 5 от сопроцессора арифметики с плавающей точкой (неправильная команда)	FPUI5	0x0000_0048 hex
11	Прерывание от системного интегратора (выборка команд из периферийной области памяти)	IFE	0x0000_0050 hex
12	Прерывание от моста «системный интегратор – AXI» (ошибка обращения во внешнюю память)	DAE	0x0000_0058 hex
13	Прерывание от MPU по защите памяти на запись	PWI	0x0000_0060 hex
14	Прерывание от MPU по защите памяти на чтение	PRI	0x0000_0068 hex
15	Прерывание от блока таймеров (таймер 0)	T0I	0x0000_0070 hex
16	Прерывание от блока таймеров (таймер 1)	T1I	0x0000_0078 hex
17	Межпроцессорное прерывание 0 от системного контроллера	SCI0	0x0000_0080 hex
18	Межпроцессорное прерывание 1 от системного контроллера	SCI1	0x0000_0088 hex
19	Межпроцессорное прерывание 2 от системного контроллера	SCI2	0x0000_0090 hex
20	Межпроцессорное прерывание 3 от системного контроллера	SCI3	0x0000_0098 hex

Продолжение таблицы 2.1

№	Прерывание	Обозначение	Адрес-вектор
21	Прерывание от коммуникационного порта CP0 (передающий канал)	CPO0	0x0000_00A0 hex
22	Прерывание от коммуникационного порта CP0 (принимающий канал)	CPI0	0x0000_00A8 hex
23	Прерывание от коммуникационного порта CP1 (передающий канал)	CPO1	0x0000_00B0 hex
24	Прерывание от коммуникационного порта CP1 (принимающий канал)	CPI1	0x0000_00B8 hex
25	Прерывание от коммуникационного порта CP2 (передающий канал)	CPO2	0x0000_00C0 hex
26	Прерывание от коммуникационного порта CP2 (принимающий канал)	CPI2	0x0000_00C8 hex
27	Прерывание от коммуникационного порта CP3 (передающий канал)	CPO3	0x0000_00D0 hex
28	Прерывание от коммуникационного порта CP3 (принимающий канал)	CPI3	0x0000_00D8 hex
29	Межпроцессорное прерывание 4 от системного контроллера	SCI4	0x0000_00E0 hex
30	Межпроцессорное прерывание 5 от системного контроллера	SCI5	0x0000_00E8 hex
31	Межпроцессорное прерывание 6 от системного контроллера	SCI6	0x0000_00F0 hex
32	Межпроцессорное прерывание 7 от системного контроллера	SCI7	0x0000_00F8 hex

2.1.7.2 Внутренние и внешние прерывания процессорного ядра

Процессор поддерживает 32 прерывания (см. таблицу 2.1): 4 внутренних прерываний процессорного ядра (в таблице с номерами от 1 до 4) и 28 внешних по отношению к процессорному ядру (в таблице с номерами от 5 до 32). Такое деление прерываний на две группы объясняется реализацией двухуровневой системы прерываний, как показано на рисунке 2.6. Запросы от сопроцессора с плавающей точкой и запросы от периферийных узлов (PERIPHERAL UNITS) поступают на контроллер внешних прерываний (INTC) процессорного ядра и фиксируются в специальном программно доступном на чтение и сброс периферийном регистре. Далее запросы перемножаются на программно настраиваемую маску и проходят арбитраж. Затем в соответствии с победившим запросом выставляется адрес-вектор соответствующего прерывания (External Interrupt Vector) и формируется обобщённый запрос на внешнее прерывание (External Interrupt Request). Данный запрос и адрес-вектор фиксируются в процессорном ядре NMC4, после чего сбрасывается победивший запрос на прерывание в контроллере внешних прерываний, и данный контроллер может выставить ещё один обобщённый запрос и соответствующий адрес-вектор прерывания. Следующее внешнее прерывание будет зафиксировано процессорным ядром только после того, как будет

обработано предыдущее внешнее прерывание. Более подробно работа контроллера внешних прерываний будет описана далее.

Рисунок 2.6 - Структурная схема системы прерываний процессорного ядра NMC4

В процессорном ядре имеется блок прерываний (INTERRUPT UNIT), куда поступают немаскируемое прерывание $\overline{\text{NMI}}$, 4 прерывания от ядра (Core Interrupt Requests) и, как описано выше, адрес-вектор внешнего по отношению к ядру прерывания (External Interrupt Vector) и обобщённый запрос на внешнее прерывание (External Interrupt Request). Все эти запросы на прерывание, а также адрес-вектор внешнего прерывания, фиксируются в специальном программно доступном на чтение и сброс регистре INTR. Далее запросы перемножаются на программно настраиваемую маску (кроме немаскируемого прерывания) и проходят арбитраж. Причём обобщённый запрос на внешнее прерывание также имеет свою маску. Затем в соответствии с победившим запросом выставляется адрес перехода на программу обработки соответствующего прерывания (Interrupt Address) и формируется окончательный запрос на прерывание (Interrupt Request). После того, как будет разрешена обработка запроса победившего прерывания, соответствующий запрос на прерывание в ядре будет сброшен. Разрешение на обработку следующего прерывания будет получено лишь после того, как закончатся все контекстные переключения при входе в обработку предыдущего прерывания.

2.2 Управляющее RISC-ядро

2.2.1 Структура RISC-ядра

RISC-ядро является одним из основных узлов процессорного ядра. Оно обеспечивает выборку и дешифрацию команд, хранение и модификацию программного счетчика PC, управление всеми исполнительными конвейерами процессора и их синхронизацию, необходимые предварительные адресные вычисления и передачу адресов к внешним адресным генераторам, а также выполняет арифметические и логические операции над 32-разрядными данными в дополнительном коде, когда использование сопроцессора неэффективно. Структурная схема RISC-ядра представлена на рисунке (см. Рисунок 2.7).

Рисунок 2.7 – Структурная схема RISC- ядра

Ниже приводится описание основных составляющих RISC- ядра:

GR0- GR7 – восемь 32-разрядных регистров общего назначения, которые используются для задания смещения при вычислении адресов команд и данных, а также в качестве источников и приёмников скалярных арифметических операций.

AR0- AR7 - восемь 32-разрядных регистров адреса, которые необходимы при вычислении адресов команд и данных. Адресный регистр AR7 (SP) является также указателем системного стека.

ALU – арифметико-логическое устройство, которое выполняет логические и арифметические операции, пошаговые операции 32-разрядного умножения над регистрами общего назначения. Результат операций также записывается в регистры общего назначения.

SHIFTER – сдвигатель, который выполняет операции сдвига над регистрами общего назначения. Результат операций также записывается в один из регистров общего назначения.

AU1 – первое адресное устройство, которое используется для формирования адреса чтения из памяти/записи в память скалярных данных или первых векторных данных, а также для вычисления адреса перехода в командах условного перехода и перехода к подпрограммам. Оно использует в качестве операндов значение адресного регистра либо счетчика команд PC в качестве адреса и значение регистра общего назначения, либо непосредственную константу из команды в качестве смещения. Результат вычислений может быть выдан на внешнюю шину адреса или смещения адреса, а также быть записан в программном счетчике PC.

AU2 – второе адресное устройство, которое используется для модификации адресных регистров при выполнении как скалярных, так и векторных команд. Оно берёт в качестве операндов значение адресного регистра либо указателя стека как базу и значение регистра общего назначения как смещение. В случае выполнения векторной команды смещение перед адресными вычислениями увеличивается в число раз, соответствующее числу повторений соответствующей векторной команды. Результат операции сохраняется в адресном регистре.

PC – счетчик команд, показывающий на адрес следующей команды, которая поступит на выполнение.

CR – буфер констант. Он хранит константу, задаваемую командой, для её использования при адресных вычислениях или для её записи в программно доступные регистры как процессорного ядра, так и периферийных узлов.

PSWR – регистр слова состояния процессора, который содержит маски на прерывания, флаги (признаки) выполнения арифметических команд, бит разрешения одновременного выполнения нескольких команд, информацию для аппаратной поддержки вершины системного стека.

IR – конвейерные, программно недоступные регистры команд. Они обеспечивают управление всеми конвейерами RISC-ядра и сопроцессоров.

2.2.2 Основные режимы работы RISC-ядра и методы адресации памяти

Структура RISC-процессорного ядра выбрана таким образом, чтобы он мог работать одновременно в двух режимах:

- поддержка операций матрично-векторных сопроцессоров, когда ядро вычисляет адрес следующей команды, а также адрес первых векторных данных и смещение, необходимых сопроцессору;
- поддержка команд управления (организация циклов, ветвлений и т.д.) и скалярных команд обработки данных, когда RISC-ядро вычисляет адрес следующей команды, исполнительный адрес для скалярных операндов, а также выполняет арифметические и логические операции над данными в GR0- GR7.

Адрес следующей команды может быть определен либо с помощью инкрементации PC, либо путем его задания константой в коде команды, либо сложением содержимого одного из адресных регистров - AR0, …, AR7 или PC со смещением, заданным константой в коде команды или содержимым одного из регистров общего назначения - GR0, …, GR7. Тем самым, в RISC- процессорном ядре поддерживаются следующие команды управления (как условные, так и безусловные): переход/переход со смещением (JUMP/SKIP), переход к подпрограмме (CALL), возврат из подпрограммы/прерывания (RETURN/IRETURN), останов (HALT).

Исполнительный адрес операндов может вычисляться с использованием следующих методов адресации: по содержимому адресного регистра без его изменения, с его инкрементацией, декрементацией, а также по сумме содержимого адресного регистра и соответствующего регистра общего назначения. Для скалярных команд кроме этого возможно определение исполнительного адреса с помощью константы, задаваемой в поле команды.

2.2.3 Регистр слова состояния процессора PSWR

32- разрядный регистр слова состояния процессора PSWR содержит всю наиболее важную информацию о работе процессорного ядра. Данный регистр сохраняется в системном стеке при переходе к подпрограмме или прерыванию и восстанавливается при возврате из прерывания. Регистр доступен программно для чтения, записи, а также поразрядной установке или сбросу по маске. Формат регистра PSWR представлен на рисунке (см. Рисунок

2.8), а функциональное назначение его полей приведено в таблице (см. Таблица 2.2). После системного сброса все разряды регистра слова состояния процессора содержат нули, что отражено внизу на рисунке.

31	. . .	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			STE	RESERVED			OVM	EIM	INM	STM	Y	N	Z	V	C
0	. . .	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 2.8 – Формат регистра PSWR

Таблица 2.2 – Функциональное назначение полей регистра PSWR

Разряд	Обозна-чение	Описание
Флаги скалярных АЛУ операций		
PSW<0>	C	Признак переноса арифметической операции
PSW<1>	V	Признак арифметического переполнения
PSW<2>	Z	Признак нулевого результата
PSW<3>	N	Признак отрицательного знака результата
PSW<4>	Y	Признак переноса в соседнюю группу разрядов множителя по Буту
Маски прерываний		
PSW<5>	ST	Маска пошагового прерывания
PSW<6>	INM	Маска внешнего прерывания
PSW<7>	EIM	Маска прерывания по запрещенной команде
PSW<8>	SPM	Маска прерывания по арифметическому переполнению
Прочее управление		
PSW<11:9>	RESERV	Reserved
PSW<12>	STE	Разрешение работы аппаратной вершины стека
PSW<31:13>		Reserved

Разряды с 31 по 13 содержит служебную информацию, которая используется для работы аппаратной вершины системного стека, и программно недоступны.

2.2.4 Регистр запросов на прерывание INTR

32-разрядный регистр запросов на прерывание INTR хранит в себе ещё не обработанные запросы на прерывания, как от процессорного ядра, так и от периферийных устройств. Формат регистра INTR представлен на рисунке 2.9, а функциональное назначение его полей приведено в таблице 2.3. Внизу на рисунке указано состояние разрядов регистра после системного сброса.

Регистр INTR программно доступен только на чтение и побитовый сброс запросов на прерывание. При возникновении запроса на прерывание в соответствующем бите регистра INTR устанавливается единица, которая сбрасывается или когда данный запрос начинает обслуживаться или после соответствующего программного побитового сброса.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						NMI	OVR	EIR	INR	IA					
0	0	0	0	0	1	0	0	0	0						

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Рисунок 2.9 – Формат регистра INTR

Таблица 2.3 – Функциональное назначение полей регистра INTR

Разряд	Обозначение	Описание
INTR<5:0>	IA	Номер внешнего прерывания: 000000 – прерывание 0 от сопроцессора арифметики с плавающей точкой (некорректные данные) 000001 – прерывание 1 от сопроцессора арифметики с плавающей точкой (overflow) 000010 – прерывание 2 от сопроцессора арифметики с плавающей точкой (underflow) 000011 – прерывание 3 от сопроцессора арифметики с плавающей точкой (потеря значимости) 000100 – прерывание 4 от сопроцессора арифметики с плавающей точкой (потеря данных) 000101 – прерывание 5 от сопроцессора арифметики с плавающей точкой (неправильная команда) 000110 – прерывание от системного интегратора (выборка команд из периферийной области памяти) 000111 – прерывание от моста «системный интегратор – AXI» 001000 – прерывание от MPU по защите памяти на запись 001001 – прерывание от MPU по защите памяти на чтение 001010 – прерывание от блока таймеров (таймер 0) 001011 – прерывание от блока таймеров (таймер 1) 001100 – межпроцессорное прерывание 0 от системного контроллера 001101 – межпроцессорное прерывание 1 от системного контроллера 001110 – межпроцессорное прерывание 2 от системного контроллера 001111 – межпроцессорное прерывание 3 от системного контроллера 010000 – прерывание от коммуникационного порта CP0 (передающий канал) 010001 – прерывание от коммуникационного порта CP0 (принимающий канал) 010010 – прерывание от коммуникационного порта CP1 (передающий канал) 010011 – прерывание от коммуникационного порта CP1 (принимающий канал) 010100 – прерывание от коммуникационного порта CP2 (передающий канал) 010101 – прерывание от коммуникационного порта CP2 (принимающий канал) 010110 – прерывание от коммуникационного порта CP3 (передающий канал) 010111 – прерывание от коммуникационного порта CP3 (принимающий канал) 011000 – межпроцессорное прерывание 4 от системного контроллера 011001 – межпроцессорное прерывание 5 от системного контроллера 011010 – межпроцессорное прерывание 6 от системного контроллера 011011 – межпроцессорное прерывание 7 от системного контроллера
INTR<6>	INR	Запрос на внешнее прерывание
INTR<7>	EIR	Запрос на прерывание по запрещенной команде: для векторных команд - использование векторных команд сопроцессора с фиксированной точкой (разряды с 31–го по 29-й равны 0, хоть один разряд с 28-го по 0-й равен единице); для скалярных команд – обмен значениями регистров RISC-ядра и сопроцессора арифметики с плавающей точкой без использования регистра SIR.
INTR<8>	OVR	Запрос на прерывание по переполнению при выполнении скалярной арифметической операции
INTR<9>	NMI	Запрос на немаскируемое внешнее прерывание
INTR<31:10>	RESERVED	Не используется

2.3 Матрично-векторный сопроцессор арифметики с плавающей точкой

2.3.1 Базовые операции матрично-векторного сопроцессора

Базовой операцией сопроцессора арифметики с плавающей точкой является операция умножения 2-элементного вектора на матрицу 2*2, причём элементы вектора и матрицы – 32-разрядные числа с плавающей точкой:

$$Z_m = \sum_{n=0}^{1} X_n \times W_{n,m} + Y_m$$

(m = 0, 1)

Сопроцессор содержит 4 процессорные ячейки, каждое из которых имеет операционное устройство регулярной структуры, состоящее из матрицы умножителей и сумматоров (см. рисунок 2.10).

Рисунок 2.10– Операционное устройство процессорной ячейки

Каждая макроячейка данного устройства выполняет операцию умножения элемента входного вектора Xi на вес Wij, а затем результат прибавляется к выходному значению верхней макроячейки, расположенной в том же столбце. Таким образом, за один такт в каждом столбце независимо вычисляется свой результат.

Поскольку операции производятся над 32-разрядными числами с плавающей точкой, максимальная производительность операционного устройства – 4 умножения и сложения за такт, т.е. 8 FLOPS. Суммарная производительность всех четырёх ячеек сопроцессора – 32 FLOPS. При обработке комплексных чисел за такт также производятся 4 умножения и сложения для одной ячейки и 32 умножения и сложения для всех четырёх процессорных ячеек.

Операционное устройство способно выполнять следующие действия над числами с плавающей точкой, как действительными, так и комплексными, как над данными с одинарной точностью (32 разряда), так и над данными с двойной точностью (64 разряда):

- умножение матрицы на вектор (матрицу);
- поэлементное умножение двух векторов;
- поэлементное сложение/вычитание двух векторов;
- КИХ-фильтр;
- БПФ (только для комплексных чисел).

При работе с числами с плавающей точкой двойной точности достигается максимальная производительность 8 FLOPS.

2.3.2 Структура матрично-векторного сопроцессора

Матрично-векторный сопроцессор является основным вычислительным узлом процессорного ядра при обработке 32-разрядных данных, представленных в формате с плавающей точкой. Структурная схема сопроцессора представлена на рисунке 2.11.

Рисунок 2.11 – Структурная схема матрично-векторного сопроцессора

Основными узлами матрично - векторного сопроцессора являются:

CENTRAL CONTROL UNIT – центральный блок управления сопроцессором. Он принимает команды от управляющего RISC-ядра по внутренней шине команд (INTERNAL INSTRUCTION BUS), дешифрирует и проверяет их на правильность. Если команда правильная, то она запускается на выполнение при условии, что все требуемые ей ресурсы свободны. В случае ошибочности команды она не выполняется, и при этом формируется соответствующее прерывание от сопроцессора. 32-разрядная скалярная шина данных (SCALAR DATA BUS) используется для чтения/записи программно доступных скалярных регистров блока управления.

FP PROCESSING CELL 0, …, FP PROCESSING CELL 3 – 4 одинаковые процессорные ячейки, каждая из которых осуществляет арифметические операции над данными в формате с плавающей точкой как одинарной (32 разряда), так и двойной точности (64 разряда). Каждая ячейка имеет следующие 64-разрядные шины: две входных – IDB0, IDB1 и одну выходную – ODB, что позволяет за один такт осуществить до двух операций чтения и одной операции записи. 32-разрядная скалярная шина данных (SCALAR DATA BUS) используется для чтения/записи программно доступных скалярных регистров процессорных ячеек.

REPACK UNIT – блок упаковки и распаковки данных. Данный блок предназначен для различных преобразований данных в формате с плавающей точкой в целые числа и обратно, а также 64-разрядных данных в 32-разрядные и наоборот.

SWITCH 9->11 – коммутатор 11 в 9. Он позволяет в одном и том же такте осуществить следующие операции:

- до четырёх чтений из памяти по 64-разрядным шинам VBI0 - VBI3 с записью прочитанных данных в процессорные ячейки или в блок упаковки и распаковки данных,
- до двух записей в память по 64-разрядным шинам VBO0 – VBO1 результатов работы процессорных ячеек или блока упаковки и распаковки данных,
- до пяти пересылок данных между процессорными ячейками и блоком упаковки и распаковки данных.

2.3.3 Форматы векторных данных

2.3.3.1 Данные

Матрично-векторный сопроцессор предназначен для обработки 32- и 64-разрядных данных в формате с плавающей точкой. При работе с комплексными 32-разрядными данными они объединяются в 64-разрядные слова (32 разряда – действительная часть и 32 разряда – мнимая часть).

2.3.3.2 64-разрядные слова упакованных данных в памяти

Сопроцессор обменивается с памятью по четырём входным шинам – VBI0 - VBI3– и двум выходным шинам – VBO0 – VBO1. Все обмены осуществляются блоками данных по 64 разряда, причём при работе с действительными 32-разрядными данными в одном 64-разрядном слове упакованы два 32-разрядных числа, при работе с комплексными числами – одно (32 разряда – действительная часть и 32 разряда – мнимая часть). Пример расположения в памяти векторов действительных и комплексных 32-разрядных данных представлен на рисунке (см. Рисунок 2.12). Расположение векторов 64-разрядных данных в памяти определяется программистом, и на это нет аппаратных ограничений.

63 … 32	31 … 0
D_1	D_0
D_3	D_2
. . .	. . .
D_{N+1}	D_N

а) Действительные векторные данные

63 … 32	31 … 0
Re_0	Im_0
Re_1	Im_1
. . .	. . .
Re_N	Im_N

б) Комплексные векторные данные

Рисунок 2.12 – Представление векторов данных в памяти

2.3.4 Центральный блок управления сопроцессором

Центральный блок управления матрично-векторным сопроцессором выполняет следующие функции:

- Отслеживает занятость ресурсов сопроцессора.
- Принимает и дешифрирует команды от управляющего RISC-ядра.
- Проверяет команды на правильность. Если проверка прошла успешно, запускает команды на выполнение при условии, что все ресурсы, необходимые ей, свободны. Если нет, вместо команды вставляется код NOP (нет операции), и формируется запрос на прерывание 5 от сопроцессора арифметики с плавающей точкой (неправильная команда).
- Формирует пять запросов на прерывания по результатам работы сопроцессора:
  - прерывание 0 от сопроцессора арифметики с плавающей точкой (некорректные данные);

- прерывание 1 от сопроцессора арифметики с плавающей точкой (overflow);
- прерывание 2 от сопроцессора арифметики с плавающей точкой (underflow);
- прерывание 3 от сопроцессора арифметики с плавающей точкой (потеря значимости);
- прерывание 4 от сопроцессора арифметики с плавающей точкой (потеря данных).

Команды, пришедшие в блок управления сопроцессором от управляющего RISC-ядра, считаются неправильными в следующих случаях:

- задан не существующий скалярный регистр сопроцессора в командах пересылки типа регистр-регистр или регистр-память;
- задана арифметическая операция над векторным регистром-источником, в который перед этим отсутствовала запись (т.е. он пуст);
- задана арифметическая операция над несколькими векторными регистрами, содержащими разное число элементов данных;
- задана арифметическая операция с маскированием результата, и векторные регистры, определяющие операнды и результат, содержат разное число элементов данных;
- задана команда записи в память содержимого векторного регистра, в который перед этим отсутствовала запись;
- задана команда записи в память содержимого векторного регистра, содержащего число элементов данных, отличное от количества повторений, указанных в коде команды;
- задана любая команда с использованием в качестве источника векторной регистровой пары с разным числом элементов данных в этих регистрах.
- задана команда пересылки векторных регистров, и регистр-источник пуст;
- задана команда чтения данных из памяти в блок упаковки и распаковки, и количество пересылаемых данных с учётом типа переупаковки превышает суммарную глубину входного и выходного FIFO этого блока;
- задана команда записи данных в память из блока упаковки и распаковки, и суммарное число данных во входном и выходном FIFO этого блока меньше количества повторений, указанных в коде команды;

- задана команда пересылки данных из векторного регистра (пары) в блок упаковки и распаковки, и количество пересылаемых данных с учётом типа переупаковки превышает суммарную глубину входного и выходного FIFO этого блока;
- задана команда пересылки данных из блока упаковки и распаковки в векторный регистр (пару), и суммарное число данных во входном и выходном FIFO этого блока меньше количества повторений, указанных в коде команды;
- задана любая команда с использованием в качестве источника векторной регистровой пары с разным числом элементов данных в этих регистрах.

Далее в таблице 2.4 представлен перечень программно доступных скалярных регистров центрального блока управления.

Таблица 2.4 – Перечень программно доступных регистров центрального блока управления

Название	Доступ	Описание
FPCR	ЧТ/ЗП	Регистр управления, содержащий бит параллельной работы и режимы округления
RIER	ЧТ/ЗП	Регистр порядка целого числа при переупаковке
FPIEIR	ЧТ	Регистр, содержащий информацию о прерывании по неправильной команде
FPIOIR	ЧТ	Регистр, содержащий информацию о прерывании по некорректным данным
FPOIR	ЧТ	Регистр, содержащий информацию о прерывании по переполнению
FPUIR	ЧТ	Регистр, содержащий информацию о прерывании по потере значимости
FPIIR	ЧТ	Регистр, содержащий информацию о прерывании по потере точности
FPDLIR	ЧТ	Регистр, содержащий информацию о прерывании по потере данных при сложении трёх операндов

Регистр управления FPCR

Регистр управления FPCR задаёт режим округления результата, а также разрешает или запрещает выполнение нескольких команд в сопроцессоре одновременно. Регистр доступен программно для чтения, поразрядной установке или сбросу по маске 0-го разряда, записи заданного значения в 2-1 разряды. Формат регистра FPCR представлен на рисунке 2.13, а функциональное назначение его полей приведено в таблице 2.5. После системного сброса все разряды регистра управления, кроме 0-го, содержат нули, что отражено внизу на рисунке.

Рисунок 2.13 – Формат регистра FPCR

Таблица 2.5 – Функциональное назначение полей регистра FPCR

Разряд	Обозна-чение	Описание
FPCR <31:3>	RESERVED	Reserved
FPCR <2:1>	RND	Режим округления результата: 00 – к ближайшему; 01 - к -∞; 10 - к +∞; 11 – к нулю.
FPCR <0>	P	Бит параллельной работы: 0 – задаёт строго последовательное выполнение команд сопроцессором; 1 – разрешает одновременное выполнение нескольких команд сопроцессором.

Разряды с 31 по 3 не используются, при их чтении возвращаются все нули, запись в них блокируется.

Регистр порядка целого числа при переупаковке RIER

Регистр порядка целого числа при переупаковке RIER (разряды 10 – 0, причём 10-й разряд - знаковый) задаёт величину предварительного сдвига целого числа, преобразуемого в формат с плавающей точкой. Величина сдвига задаётся в дополнительном коде. Если он равен нулю, при упаковке используются обычные целые числа. Если он – положительное число, производится предварительный сдвиг влево на это число. Если отрицательное – сдвиг вправо на это число. Тем самым поддерживается работа не только с целыми числами, но и с блочной плавающей точкой. Регистр доступен программно для чтения и записи. Формат регистра RIER представлен на рисунке 2.14. После системного сброса все разряды регистра содержат нули, что отражено внизу на рисунке.

Рисунок 2.14 – Формат регистра RIER

Разряды с 31 по 11 не используются, при их чтении возвращаются все нули, запись в них блокируется.

Регистр, содержащий информацию о прерывании по неправильной команде FPIEIR

Регистр FPIEIR хранит информацию о причине запроса на прерывание от сопроцессора по неправильной команде. Эта информация записывается в регистр в момент обнаружения первой неправильной команды, и будет храниться до тех пор, пока прерывание не будет обработано. Регистр программно доступен только на чтение. Формат регистра FPIEIR представлен на рисунке 2.15, а функциональное назначение его полей приведено в таблице 2.6. После системного сброса все разряды регистра с 31-го по 10-й содержат нули, а с 9-го по 0-й не определены, что и отражено внизу на рисунке.

Рисунок 2.15 – Формат регистра FPIEIR

Таблица 2.6 – Функциональное назначение полей регистра FPIEIR

Разряд	Обозна-чение	Описание
FPIEIR <31:10>	RESERVED	Reserved
FPIEIR <9:6>	IET	Тип ошибки в команде, вызвавшей прерывание: 1xxx - векторный регистр-источник пуст; x1xx - несовпадение количества элементов в векторных регистрах-источниках; xx1x – скалярное и векторное чтение из одного и того же векторного регистра-источника; xxx1 – ошибка, относящаяся к блоку упаковки и распаковки данных.
FPIEIR <5:3>	VRD	Номер регистра-приёмника, используемого в команде, не прошедшей проверку.
FPIEIR <2:0>	UNIT	Устройство-приёмник в неправильной команде: 0xx – процессорная ячейка с номером xx; 100 - блок упаковки и распаковки данных; 111 – запись в память.

Разряды с 31 по 10 не используются, при их чтении возвращаются все нули, запись в них блокируется.

Регистры, содержащие информацию о прерываниях по результатам работы сопроцессора FPIOIR, FPOIR, FPUIR, FPIIR и FPDLIR

Регистры FPIOIR, FPOIR, FPUIR, FPIIR и FPDLIR хранят информацию о причине запроса на прерывание от сопроцессора при выполнении им вычислений. Эта информация записывается в регистры в момент обнаружения первого соответствующего события, и будет храниться там до тех пор, пока данное прерывание не будет обработано. Регистры программно доступны только на чтение. Эти регистры имеют один и тот же формат, представленный на рисунке 2.16, а функциональное назначение его полей приведено в таблице 2.7. После системного сброса все разряды регистра с 31-го по 6-й содержат нули, а с 5-го по 0-й не определены, что и отражено внизу на рисунке.

Рисунок 2.16 – Формат регистров FPIOIR, FPOIR, FPUIR, FPIIR и FPDLIR

Таблица 2.7 – Функциональное назначение полей регистров FPIOIR, FPOIR, FPUIR, FPIIR и FPDLIR

Разряд	Обозна-чение	Описание
FPIEIR <31:6>	RESERVED	Reserved
FPIEIR <5:3>	VR	Номер регистра-источника или приёмника, который стал причиной прерывания.
FPIEIR <2:0>	NoCELL	Номер той процессорной ячейки, где произошло прерывание.

Разряды с 31 по 6 не используются, при их чтении возвращаются все нули, запись в них блокируется.

2.3.5 Процессорная ячейка

Структурная схема процессорной ячейки представлена на рисунке 2.17.

Процессорная ячейка включает в себя следующие основные узлы:

VR0 – VR7 - векторные регистры общего назначения, которые используются как в операциях ввода/вывода, так и в арифметических операциях над данными с плавающей точкой. Их максимальная ёмкость 32 64-разрядных слов упакованных данных. В некоторых типах операций векторные регистры образуют регистровые пары, формируя вектора максимальным размером 32 элемента по 128-разрядов.

SWITCH 3->8 - коммутатор 3 в 8. Он позволяет одновременно записать результат арифметической операции и данные, считанные по шинам IDB0, IDB1 из памяти, а также из других ячеек или блока упаковки и распаковки, в три векторных регистра соответственно.

Рисунок 2.17 – Структурная схема процессорной ячейки

SWITCH 8->4 - коммутатор 8 в 4. Он предназначен для выбора до 3-х источников для арифметических операций, а также одного источника для записи в память, в другие ячейки или в блок упаковки и распаковки по выходной шине ODB. Причём, в качестве этих источников может использоваться любой из восьми векторных регистров.

OPERATING UNIT – операционное устройство для выполнения векторных и матричных операций над данными в формате с плавающей точкой. Данное устройство может работать одном и четырёх режимов (см. рисунок 2.18):

- Операции над данными в формате с плавающей точкой двойной точности - см. рисунок 2.18 а). В этом режиме все входные операнды A, B и C и результат D представляют собой 64-разрядные числа в формате с плавающей точкой двойной точности. Основная операция режима – умножение двух чисел и сложение с третьим.
- Операции над комплексными данными в формате с плавающей точкой одинарной точности - см. рисунок 2.18 б). Данный режим характеризуется тем, что все входные

операнды: A1 и A0, B1 и B0, C1 и C0, а также результат D1 и D0 представляют собой комплексные 64-разрядные числа, причём старшие 32 разряда содержат действительную часть (A1, B1, C1 и D1), а младшие 32 разряда (A0, B0, C0 и D0) – мнимую часть. Основная операция режима – умножение двух комплексных чисел и сложение с третьим.

- Операции над векторными данными в формате с плавающей точкой одинарной точности - см. рисунок 2.18 в). В этом режиме все входные операнды A1 и A0, B1 и B0, C1 и C0, а также результат D1 и D0 образуют 64-разрядные вектора из двух 32-разрядных элементов. Основная операция режима – умножение двух векторов по два элемента и сложение с третьим двухэлементным вектором.
- Операции над матричными данными в формате с плавающей точкой одинарной точности - см. рисунок 2.18 г). Данный режим характеризуется тем, что все входные операнды A1 и A0, B1 и B0, B3 и B2, C1 и C0, а также результат D1 и D0 образуют 64-разрядные вектора из двух 32-разрядных элементов, как и в предыдущем случае. Основная операция режима – умножение вектора из двух элементов [C1 C0] на матрицу 2*2 элемента $\begin{bmatrix} B1 & B3 \\ B0 & B2 \end{bmatrix}$ и сложение с третьим двухэлементным вектором [A1 A0]. Особенностью данной операции является то, что матрица считывается сразу из пары векторных регистров: из того, что указан в команде (он обязан иметь чётный номер) считывается столбец $\begin{bmatrix} B1 \\ B0 \end{bmatrix}$, а из регистра с номером на единицу больше – столбец $\begin{bmatrix} B3 \\ B2 \end{bmatrix}$. Имеется также возможность в зависимости от кода команды сформировать матрицу по строкам: из чётного регистра считается строка [B0 B2], из нечётного – строка [B1 B3]. В первом случае это позволяет осуществить умножение матрицы B на вектор-строку C слева, во втором - умножение матрицы B на вектор-столбец C справа.
  Остальные операнды в виде одного 64-разрядного данного или вектора из двух 32-разрядных данных считываются только из одного векторного регистра. Результат также пишется только в один векторный регистр.

D = A + B*C

а) Операции над данными в формате с плавающей точкой двойной точности (64 разряда)

D0 = A0 + B0*C1 + B1*C0
D1 = A1 + B1*C1 – B0*C0

б) Операции над комплексными данными в формате с плавающей точкой одинарной точности (32 разряда)

D0 = A0 + B0*C0
D1 = A1 + B1*C1

в) Операции над векторными данными в формате с плавающей точкой одинарной точности (32 разряда)

D0 = A0 + B2*C0 + B3*C1
D1 = A1 + B0*C0 + B1*C1

г) Операции над матричными данными в формате с плавающей точкой одинарной точности (32 разряда)

Рисунок 2.18 – Режимы работы операционного устройства

SCALAR REGISTERS – программно доступные скалярные регистры, входящие в состав каждой процессорной ячейки. В таблице 2.8 представлен их перечень .

Таблица 2.8 – Перечень программно доступных регистров процессорной ячейки

Название	Доступ	Описание
FPFR	ЧТ/ЗП	Регистр флагов
SPMRL	ЧТ/ЗП	Младшая часть регистра маски одинарной точности
SPMRH	ЧТ/ЗП	Старшая часть регистра маски одинарной точности
DPMR	ЧТ/ЗП	Регистр маски двойной точности

Регистр флагов FPFR

Регистр FPFR хранит флаги результата вычисления последнего элемента результата последней выполненной арифметической команды (для 32-разрядных данных соответствует результату в разрядах 63-32). Регистр программно доступен на чтение и запись. Формат регистра FPFR представлен на рисунке 2.19, а функциональное назначение его полей приведено в таблице 2.9. После системного сброса все разряды регистра содержат нули, что и отражено внизу на рисунке.

Рисунок 2.19 – Формат регистра FPFR

Таблица 2.9 – Функциональное назначение полей регистра FPFR

Разряд	Обозна-чение	Описание
FPFR <31:4>	RESERVED	Reserved
FPFR <3>	N	Признак отрицательного знака результата
FPFR <2>	Z	Признак нулевого результата
FPFR <1:0>	RESERVED	Reserved

Разряды с 31-го по 4-й и с 1-го по 0-й не используются, при их чтении возвращаются все нули, запись в них блокируется.

Регистры масок одинарной точности SPMRL (младшая часть) и SPMRH (старшая часть)

В качестве альтернативы работы с условными командами введён механизм работы с масками. В общем случае, маска формируется по результату арифметической команды. Каждому i-му элементу вектора результата (i = 0, 1,…N, где N≤31) ставится в соответствие i-й разряд регистра маски. Если выполняется определённое условие для i-го элемента вектора результата, единое для всего вектора, то в i-й разряд регистра маски пишется единица, иначе

– ноль. В дальнейшем, при выполнении арифметических операций есть возможность замаскировать результат по содержимому регистра маски: если разряд регистра маски равен 1, то соответствующий элемент вектора результата запишется в регистр-приёмник, если нет - соответствующий элемент в регистре-приёмнике меняться не будет. Таким образом, использование механизма масок позволяет обойтись без команд условного перехода.

Поскольку арифметические операции в формате с плавающей точкой одинарной точности (32 разряда) требуют максимум 64 бита маски (32 элемента вектора по два 32-разрядных слова), используются два 32-разрядных регистра маски одинарной точности – SPMRL (младшая часть) и SPMRH (старшая часть). Их форматы приведены, соответственно, на рисунках 2.20 и 2.21. 0-й разряд регистра SPMRL содержит маску, соответствующую 0-му элементу вектора результата, разрядам с 31-го по 0-й. Соответственно, 0-й разряд регистра SPMRH содержит маску, соответствующую 0-му элементу вектора результата, разрядам с 63-го по 32-й. Таким образом, SPMRL содержит маски для элементов вектора с 0-го по 31-й только для младших 32-х разрядов результата, а SPMRH - маски для элементов вектора с 0-го по 31-й только для старших 32-х разрядов результата. Если число элементов в векторе результата меньше 32-х, то старшие разряды регистров масок не используются.

Регистры SPMRL и SPMRH программно доступны на чтение и запись. После системного сброса все разряды регистров не определены, что и отражено внизу на рисунке.

Рисунок 2.20 – Формат регистра SPMRL

Рисунок 2.21 – Формат регистра SPMRH

При выполнении арифметических операций в формате с плавающей точкой двойной точности (64 разряда) требуется максимум 32 бита маски (32 элемента вектора по одному 64-разрядному слову). В этом случае используется только один регистр маски, а именно SPMRH.

Регистр маски двойной точности DPMR

Регистр DPMR является псевдо-регистром. Запись в него вызовет запись одних и тех же 32-разрядных данных в регистры SPMRL и SPMRH, чтение из него вызовет чтение

SPMRH. Данный псевдо-регистр используется вместе с арифметическими операциями в формате с плавающей точкой двойной точности, чтобы программист не заботился о том, какой из регистров - SPMRL или SPMRH – используется. Вместе с тем, для такого вида операций можно использовать регистр SPMRH в явном виде.

2.3.6 Блок упаковки и распаковки данных

Структурная схема блока упаковки и распаковки данных представлена на рисунке 2.22.

Рисунок 2.22 – Структурная схема блока упаковки и распаковки данных

Блок упаковки и распаковки данных включает в себя следующие основные узлы:

IFIFO – входное FIFO, куда данные, считанные по шине IDB из памяти или из одной из процессорных ячеек, записываются для дальнейшего преобразования.

OFIFO – выходное FIFO, откуда результат переупаковки считывается для записи в память, в другие ячейки или в блок упаковки и распаковки по выходной шине ODB.

REPACK LOGIC – комбинационная схема, реализующая требуемую упаковку и распаковку данных. На её входе и выходе возможно использование следующих восьми форматов данных, причём возможно любое их сочетание:

- в блоке 64-разрядных данных упакованы два 32-разрядных числа в формате с фиксированной точкой;
- в блоке 64-разрядных данных упакованы два 32-разрядных числа в формате с фиксированной точкой, но в дальнейшем используются только старшие 32-разряда (младшие 32-разряда обнуляются);
- в блоке 64-разрядных данных упакованы два 32-разрядных числа в формате с фиксированной точкой, но в дальнейшем используются только младшие 32-разряда (старшие 32-разряда обнуляются);
- в блоке 64-разрядных данных находится одно число в формате с фиксированной точкой;
- в блоке 64-разрядных данных упакованы два 32-разрядных числа в формате с плавающей точкой;

- в блоке 64-разрядных данных упакованы два 32-разрядных числа в формате с плавающей точкой, но в дальнейшем используются только старшие 32-разряда (младшие 32-разряда обнуляются);
- в блоке 64-разрядных данных упакованы два 32-разрядных числа в формате с плавающей точкой, но в дальнейшем используются только младшие 32-разряда (старшие 32-разряда обнуляются);
- в блоке 64-разрядных данных находится одно число в формате с плавающей точкой двойной точности.

Таким образом, возможны 64 варианта упаковки и распаковки данных, имеющих различные форматы и разрядности.

2.4 Система команд процессорного ядра NMC4

Система команд процессорного ядра NMC4 совместима с системой команд других процессорных ядер этого же семейства и является уникальной по отношению к другим процессорам.

Характерными особенностями системы команд NMC4 являются:

- Фиксированная длина команды. Все команды имеют длину 32 разряда, либо 64 разряда, если старшим словом команды является непосредственно передаваемая константа.
- Сверхбольшое слово команды (VLIW). Каждая команда может задавать несколько операций, причем эти операции могут исполняться независимо.
- Аппаратная поддержка циклов. В командах, определяющих работу матрично-векторного сопроцессора, допускается задавать число повторений команды от 1 до 32.
- Принудительное включение последовательного исполнения команд. Процессорное ядро является мультиконвейерным, динамическим суперскалярным вычислительным устройством, которое может исполнять несколько команд одновременно.

Команды процессорного ядра NMC4 делятся на две основные группы: команды управляющего RISC-процессора (см. рисунок 2.23 а) и команды сопроцессора арифметики с плавающей точкой (см. рисунок 2.23 б). Каждая из этих групп, в свою очередь, делится на команды скалярные, т.е. обычные RISC-команды, и векторные, которые задают многократно одни и те же действия над векторами данных, что эквивалентно аппаратной поддержке коротких циклов. В отличие от векторных команд сопроцессора арифметики с плавающей точкой, векторные команды управляющего RISC-процессора с фиксированной точкой в данной реализации не используются.

Форматы команд приведены на рисунке 2.23, краткое их описание дано ниже.

	31	30	29	28	26	25	24	22	21	16	15	0
1.1	0	0	1	MA		R/W	ARi		Rист/пр- к		КОП СК	
1.2	0	1	1	0	MA	R/W	ARi		Rист/пр- к		КОП СК	
	АДРЕС (смещение)											
2.1	0	1	1	1	Rпр- к				Rист		КОП СК	
2.2	0	1	0	0	Rпр- к				0	0	x x x x	КОП СК
	КОНСТАНТА											
3.1	0	1	0	1	КМ	1	ARi		0	1	x ARj	КОП СК
3.2	0	1	0	0	КМ	1	ARi		0	1	x ARj	КОП СК
	КОНСТАНТА- СМЕЩЕНИЕ											
3.3	0	1	0	1	x x	0	x x x		0	1	x x x x	КОП СК
3.4	0	1	0	0	x x	0	x x x		0	1	x x x x	КОП СК
	КОНСТАНТА- СМЕЩЕНИЕ											
4.1	0	0	0	0	КМ	I/C	ARi		1	0	Условие	КОП СК
4.2	0	1	0	0	КМ	I/C	ARi		1	0	Условие	КОП СК
	АДРЕС (смещение)											
4.3	0	0	0	0	0	1	1	1	0	0	1 1 Условие	КОП СК
4.4	0	0	0	0	0	1	1	0	0	0	1 1 Условие	КОП СК
4.5	0	1	1	1	0	1	1	1	0	0	0 1 1 0 0 1	КОП СК

а) Команды управляющего RISC-процессора

Рисунок 2.23 - Система команд процессорного ядра NMC4

	31	30	29	28 26	25	24 22	21 16	15 0
1.1F	1	0	1	MA	R/W	ARi	Rист/пр- к	КОП СК

1.2F	1	1	1	0	MA	R/W	ARi	Rист/пр- к	КОП СК
	АДРЕС (смещение)								

2.1F	1	1	1	1	Rпр- к	Rист	КОП СК

2.2F	1	1	0	0	Rпр- к	0	0	x	x	x	x	КОП СК
	КОНСТАНТА											

3.1F	1	0	0	M	0	0	OP	0 0 0 0	DT	x	VR_A	x x x x x x x					VRD	CELL_D
3.2F	1	0	0	M	0	1	x	OP	0 0 0 0	DT	x	x x	SB	VR_B	VR_C		VRD	CELL_D
3.3F	1	0	0	M	1	0	OP	0 0 0 0	DT	WE	VR_A	SB	VR_B	MGC			VRD	CELL_D
3.4F	1	0	0	M	1	1	OP	0 0 0 0	DT	SA	VR_A	SB	VR_B	VR_C			VRD	CELL_D
4.1F	1	0	0	MA	0		ARi	0	0	1	x	количество	R	x x x x x x			VRD	CELL_D
4.2F	1	0	0	MA	1		ARi	0	0	1	x	количество	R	VRS	CELL_S		x x x x x x	
4.3F	1	0	0	MA	0		ARi	0	1	0	x	количество	R	x x x x x x			RPOP	
4.4F	1	0	0	MA	1		ARi	0	1	0	x	количество	R	x x x x x x x x x x x x				
5.1F	1	0	0	x x x x		0	0	1 0 0 0	x x x x x x x					VRS	CELL_S		VRD	CELL_D
5.2F	1	0	0	x x x x		0	1	1 0 0 0	x x x	VRS0	MS	VRS1		CELL_S			VRD	CELL_D
5.3F	1	0	0	x x x x		1	0	1 0 0 0	x x x x x x		PR	VRS1		CELL_S			VRD	CELL_D
5.4F	1	0	0	x x x x		1	1	1 0 0 0	x	количество	R	x x x x x		PR			VRD	CELL_D

б) Команды сопроцессора арифметики с плавающей точкой

Рисунок 2.23 (окончание) - Система команд процессорного ядра NMC4

2.4.1 Форматы команд, задающих пересылку данных типа «регистр управляющего RISC-процессора – память»

Формат 1.1 Rист/пр- к ↔(fадр.(ARi, GRi)); ARi←fм(ARi, GRi)

31	30	29	28 27 26	25	24 23 22	21 20 19 18 17 16	15 0
0	0	1	МА	R/W	ARi	$R_{ист/пр-к}$	КОП СК

R/W	Направление пересылки данных
0	Чтение из памяти в $R_{ист/пр-к}$
1	Запись в память из $R_{ист/пр-к}$

МА			$f_{адр.}(AR_i, GR_i)$	$f_м(AR_i, GR_i)$
0	0	0	GR_i	AR_i
0	0	1	AR_i+GR_i	AR_i+GR_i
0	1	0	GR_i	GR_i
0	1	1	ARi+GRi	ARi
1	0	0	AR_i	AR_i
1	0	1	AR_i	AR_i+GR_i
1	1	0	AR_i- a	AR_i- a
1	1	1	AR_i	AR_i+a

Примечание: a=1 при адресации 32- разрядных данных;
a=2 при адресации 64- разрядных данных.

Формат 1.2 Rист/пр- к ↔(fадр.(ARi, CM)); ARi←fм(ARi, CM)

31	30	29	28	27 26	25	24 23 22	21 20 19 18 17 16	15 0
0	1	1	0	МА	R/W	ARi	Rист/пр- к	КОП СК
63	АДРЕС (СМЕЩЕНИЕ)							32

R/W	Направление пересылки данных
0	Чтение из памяти в Rист/пр- к
1	Запись в память из Rист/пр- к

МА		fадр.(ARi, CM)	fм(ARi, CM)
0	0	Адрес	ARi
0	1	ARi+CM	ARi+CM
1	0	Адрес	Адрес
1	1	ARi+CM	ARi

2.4.2 Форматы команд пересылки данных типа “регистр-регистр» между регистрами управляющего RISC-процессора

Формат 2.1 Rист → Rпр-к

31	30	29	28	27 26 25 24	23 22 21 20	19 18 17 16 15 … 0
0	1	1	1	Rпр-к	Rист	КОП СК

Формат 2.2 Rпр-к ← КОНСТАНТА

	31	30	29	28	27 26 25 24	23 22	21	20	19	18	17	16	15 … 0	
	0	1	0	0	Rпр-к		0	0	x	x	x	x	КОП СК	
63	КОНСТАНТА													32

2.4.3 Форматы команд модификации адресных регистров управляющего RISC-процессора

Формат 3.1. ARj←fм(ARi,GRi)

31	30	29	28	27	26	25 24 23	22	21	20	19 18 17 16	15 … 0
0	1	0	1	КМ	1	ARi	0	1	x	ARj	КОП СК

КМ		fм(ARi, GRi)
0	0	ARi
0	1	ARi + GRi
1	0	GRi
1	1	PC + GRi

ARпр-к			AR приёмник
0	0	0	AR0
0	0	1	AR1
0	1	0	AR2
0	1	1	AR3
1	0	0	AR4
1	0	1	AR5
1	1	0	AR6
1	1	1	SP(AR7)

Формат 3.2 ARj←fм(ARi, CM)

	31	30	29	28	27 26	25	24 23 22	21	20	19	18 17 16	15 … 0	
	0	1	0	0	КМ	1	ARi	0	1	x	ARj	КОП СК	
63	КОНСТАНТА-СМЕЩЕНИЕ												32

КМ		fм(ARi, CM)
0	0	ARi
0	1	ARi + CM
1	0	CM
1	1	PC + CM

ARпр-к	См. формат 3.1

Формат 3.3. Нет операций ввода/вывода и модификации адресных регистров

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 … 0
0	1	0	1	x	x	0	x	x	x	0	1	x	x	x	x	КОП СК

Формат 3.4 Нет операций ввода/вывода и модификации адресных регистров

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 … 0
0	1	0	0	x	x	0	x	x	x	0	1	x	x	x	x	КОП СК
6 3	КОНСТАНТА-СМЕЩЕНИЕ															32

2.4.4 Форматы команд управления RISC-процессора

Формат 4.1 Переход/переход к подпрограмме (GOTO/CALL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 … 0
0	0	0	0	КМ		J/C	ARi			1	0	Условие				КОП СК

J/C	Тип переход
0	Переход
1	Переход к п/п

КМ		Модификация PC
0	0	PC:=ARi
0	1	PC:=ARi+GRi
1	0	PC:=GRi
1	1	PC:=PC+GRi

Условие				Анализируемое сочетание признаков
0	0	0	0	Переход, если C=0
0	0	0	1	Переход, если V=0
0	0	1	0	Переход, если N+Z=0
0	0	1	1	Переход, если N=0
0	1	0	0	Переход, если (V⊕N)+Z=0
0	1	0	1	Переход, если V⊕N=0
0	1	1	0	Переход, если Z=0
0	1	1	1	Переход безусловный
1	0	0	0	Переход, если C=1
1	0	0	1	Переход, если V=1
1	0	1	0	Переход, если N+Z=1
1	0	1	1	Переход, если N=1
1	1	0	0	Переход, если (V⊕N)+Z=1
1	1	0	1	Переход, если V⊕N=1
1	1	1	0	Переход, если Z=1
1	1	1	1	Перехода нет

Формат 4.2. Переход/переход к подпрограмме со смещением (GOTO/CALL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	0
0	1	0	0	КМ		J/C	ARi		1	0	Условие					КОП СК	
6 3	АДРЕС (СМЕЩЕНИЕ)																32

J/C	Тип переход
0	Переход
1	Переход к п/п

Условие	См. формат 4.1.

КМ		Модификация РС
0	0	PC:=ARi
0	1	PC:=ARi+ CM
1	0	PC:=Адрес
1	1	PC:=PC+CM

Формат 4.3. Возврат из подпрограммы (RETURN)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	0
0	0	0	0	0	1	1	1	0	0	1	1	Условие				КОП СК	

Условие	См. формат 4.1.

Формат 4.4. Возврат из прерывания (IRETURN)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	0
0	0	0	0	0	1	1	0	0	0	1	1	Условие				КОП СК	

Условие	См. формат 4.1.

Формат 4.5. Останов (HALT)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	0
0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	1	КОП СК	

2.4.5 Форматы поля КОП СК (код операции скалярной команды управляющего RISC-процессора)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	ТС		ВС						GRист			GRпр- к		

W	0	КЛОП	$GR_{ист2}$	$GR_{ист1}$	GRпр- к

W	1	КАОП	$GR_{ист2}$	$GR_{ист1}$	GRпр- к

GR			РОН - источник/приемник
0	0	0	GR0
0	0	1	GR1
0	1	0	GR2
0	1	1	GR3
1	0	0	GR4
1	0	1	GR5
1	1	0	GR6
1	1	1	GR7

W		Управление записью в GRпр-к и в регистр признаков
0	1	Есть запись в GRпр-к, нет записи признаков
1	0	Есть запись признаков, нет записи в GRпр- к
1	1	Есть запись в GRпр-к и есть запись признаков

2.4.6 Формат поля КОП СК, задающего операцию сдвига

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	ТС		ВС						GRист			GRпр-к		

ТС		Тип сдвига
0	0	Циклический сдвиг
0	1	Логический сдвиг
1	0	Арифметический сдвиг
1	1	Логический сдвиг через “С”

Примечание: поле ВС задает величину сдвига в дополнительном коде:
положительное значение - сдвиг влево (от 0 до 31 разряда),
отрицательное значение - сдвиг вправо (от 1 до 32 разрядов).
“С” - признак переноса слова состояния процессора.
Логический сдвиг через “С” может выполняться только на один разряд влево или вправо;
сдвиги остальных типов могут выполняться на любое от 1 до 31 число разрядов влево или вправо.
Сдвиг на 0 или 32 разряда любого типа воспринимается как код “нет операции”, и при этом не изменяется ни приёмник результата операции GRпр-к, ни признаки.
Сдвиги от 1 до 31 разряда меняют признаки слова состояния процессора по следующим правилам:
N - признак знака - равен старшему (знаковому) разряду результата;
Z - признак нуля - равен единице, если все разряды регистра приемника GR нулевые, или нулю в противном случае;
C - признак переноса, равен последнему выталкиваемому при сдвиге разряду из GR источника;
V - признак переполнения - устанавливается в единицу только при арифметических сдвигах влево, если получаемый признак C не равен 31-разряду GR приемника, во всех остальных случаях равен нулю.

Схемы различных вариантов сдвига на один разряд изображены на рисунке (см. Рисунок 2.24). Сдвиги на большее число разрядов эквивалентны многократному сдвигу на единицу, хотя и выполняются за один такт работы процессора.

Рисунок 2.24 – Схемы выполнения различных типов операций сдвигов

2.4.7 Формат поля КОП СК, задающего логическую операцию

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W		0	КЛОП				$GR_{ист2}$			$GR_{ист1}$			GRпр-к		

КЛОП				Код логической операции	N	Z	V	C
0	0	0	0	0	0	1	0	0
0	0	0	1	$\overline{GR_{ист2}}$ & $\overline{GR_{ист1}}$	+	+	0	0
0	0	1	0	$GR_{ист2}$ & $\overline{GR_{ист1}}$	+	+	0	0
0	0	1	1	$\overline{GR_{ист1}}$	+	+	0	0
0	1	0	0	$\overline{GR_{ист2}}$ & $GR_{ист1}$	+	+	0	0
0	1	0	1	$\overline{GR_{ист2}}$	+	+	0	0
0	1	1	0	$GR_{ист2} \oplus GR_{ист1}$	+	+	0	0
0	1	1	1	$\overline{GR_{ист2}} + \overline{GR_{ист1}}$	+	+	0	0
1	0	0	0	$GR_{ист2}$ & $GR_{ист1}$	+	+	0	0
1	0	0	1	$GR_{ист2} \oplus \overline{GR_{ист1}}$	+	+	0	0
1	0	1	0	$GR_{ист2}$	+	+	0	0
1	0	1	1	$GR_{ист2} + \overline{GR_{ист1}}$	+	+	0	0
1	1	0	0	$GR_{ист1}$	+	+	0	0
1	1	0	1	$\overline{GR_{ист2}} + GR_{ист1}$	+	+	0	0
1	1	1	0	$GR_{ист2} + GR_{ист1}$	+	+	0	0
1	1	1	1	- 1	1	0	0	0

2.4.8 Формат поля КОП СК, задающего арифметическую операцию

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W		1	КАОП				$GR_{ист2}$			$GR_{ист1}$			GRпр-к		

КАОП				Код арифметической операции	Y	N	Z	V	C
0	0	0	0	$GR_{ист2}$-$GR_{ист1}$	-	+	+	+	+
0	0	0	1	$GR_{ист2}$-$GR_{ист1}$-1+”C”	-	+	+	+	+
0	0	1	0	$GR_{ист2}$+1	-	+	+	+	+
0	0	1	1	$GR_{ист2}$+”C”	-	+	+	+	+
0	1	0	0	$GR_{ист2}$-1	-	+	+	+	+
0	1	0	1	$GR_{ист2}$-1+”C”	-	+	+	+	+
0	1	1	0	$GR_{ист2}$+ $GR_{ист1}$	-	+	+	+	+
0	1	1	1	$GR_{ист2}$+ $GR_{ист1}$+”C”	-	+	+	+	+
1	0	0	0	Первый шаг умножения	+	?	?	0	?
1	0	0	1	Шаг умножения	+	?	?	0	?
1	0	1	X	Резерв					
1	1	0	0	-$GR_{ист2}$	-	+	+	+	+
1	1	X	1	Резерв					
1	1	1	X	Резерв					

Примечание: “C” - признак переноса из слова состояния процессора.

Если в поле W задаёт запись признаков, то они устанавливаются соответственно столбцам N,Z,V,C, иначе они сохраняют своё значение. Признак Y меняется только операциями “Первый шаг умножения” и “Шаг умножения” независимо от поля W. В таблицах используются следующие обозначения:

“+” - признак устанавливается по результату операции;

“0” - признак обнуляется;

“1” - признак устанавливается в единицу;

“?” - значение признака не определено.

Признак Y равен предпоследнему по значимости (1-му) разряду множителя (GR7) при выполнении операций “Первый шаг умножения” и “Шаг умножения”.

Признак N равен 31 (знаковому) разряду результата.

Признак Z равен единице, если все разряды результата нулевые, иначе - нулю.

Признак V равен нулю для операций шага умножения, для остальных операций он формируется как результат операции “исключающее ИЛИ” переноса из 31 разряда и переноса в 31 разряд.

Признак C равен переносу из 31 разряда.

2.4.9 Поле выбора адресного регистра управляющего RISC-процессора

25	24	23	Номер AR или GR
ARi			
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Примечание: AR7 (SP) может использоваться в качестве системного указателя стека адресов возврата при входе/выходе из подпрограммы (прерывания), причём изменяется он в этом случае только на +/- 2.

2.4.10 Поле Rист/пр- к в командах пересылки данных управляющего RISC-процессора

Код регистра	Регистр- источник	Регистр- приемник
0 0 0 0 0 0	GR0,AR0	GR0,AR0
0 0 0 0 0 1	GR1,AR1	GR1,AR1
0 0 0 0 1 0	GR2,AR2	GR2,AR2
0 0 0 0 1 1	GR3,AR3	GR3,AR3
0 0 0 1 0 0	GR4,AR4	GR4,AR4
0 0 0 1 0 1	GR5,AR5	GR5,AR5
0 0 0 1 1 0	GR6,AR6	GR6,AR6
0 0 0 1 1 1	GR7,AR7	GR7,AR7
0 0 1 0 0 0	AR0	AR0
0 0 1 0 0 1	AR1	AR1
0 0 1 0 1 0	AR2	AR2
0 0 1 0 1 1	AR3	AR3
0 0 1 1 0 0	AR4	AR4
0 0 1 1 0 1	AR5	AR5
0 0 1 1 1 0	AR6	AR6
0 0 1 1 1 1	AR7(SP)	AR7(SP)
0 1 0 0 0 0	GR0	GR0
0 1 0 0 0 1	GR1	GR1
0 1 0 0 1 0	GR2	GR2
0 1 0 0 1 1	GR3	GR3
0 1 0 1 0 0	GR4	GR4
0 1 0 1 0 1	GR5	GR5
0 1 0 1 1 0	GR6	GR6
0 1 0 1 1 1	GR7	GR7
0 1 1 0 0 0	PSWR,PC	PSWR,PC
0 1 1 0 0 1	SIR	SIR
0 1 1 0 1 0	INTR	INTRreset
0 1 1 0 1 1	VL	VL
0 1 1 1 0 0	PC	PC
0 1 1 1 0 1	-	PSWRset
0 1 1 1 1 0	-	PSWRreset
0 1 1 1 1 1	PSWR	PSWR

Поле $R_{ист/пр-к}$ в командах пересылки данных управляющего RISC-процессора (продолжение)

Код регистра	Регистр- источник	Регистр- приемник
1 0 0 0 0 0	-	-
1 0 0 0 0 1	-	-
1 0 0 0 1 0	-	-
1 0 0 0 1 1	-	-
1 0 0 1 0 0	-	-
1 0 0 1 0 1	-	-
1 0 0 1 1 0	INTR	-
1 0 0 1 1 1	PC	-
1 0 1 0 0 0	-	-
1 0 1 0 0 1	-	-
1 0 1 0 1 0	-	-
1 0 1 0 1 1	-	-
1 0 1 1 0 0	-	-
1 0 1 1 0 1	-	-
1 0 1 1 1 0	-	-
1 0 1 1 1 1	-	-
1 1 0 0 0 0	-	-
1 1 0 0 0 1	-	-
1 1 0 0 1 0	-	-
1 1 0 0 1 1	-	-
1 1 0 1 0 0	-	-
1 1 0 1 0 1	-	-
1 1 0 1 1 0	-	-
1 1 0 1 1 1	-	-
1 1 1 0 0 0	-	-
1 1 1 0 0 1	-	SIR
1 1 1 0 1 0	-	-
1 1 1 0 1 1	-	-
1 1 1 1 0 0	-	-
1 1 1 1 0 1	-	-
1 1 1 1 1 0	-	-
1 1 1 1 1 1	-	-

Таблица 2.10 – Обозначение регистров управляющего RISC-процессора и их назначение

Обозначение регистров и их назначение	Разрядность
Регистры процессорного ядра	
GRi – регистр общего назначения I (i=0,…,7)	32
ARj – адресный регистр j (j=0,…,6)	32
SP(AR7) – указатель стека адресов возврата	32
PC – программный счетчик	32
PSWR – регистр слова состояния процессора	32
PSWRset – код для побитовой установки PSWR в единицу (псевдорегистр)	-
PSWRreset – код для побитового сброса PSWR в ноль (псевдорегистр)	-
INTR – регистр запросов на прерывание	32
INTRreset – код для побитового сброса INTR в ноль (псевдорегистр)	-
VL – регистр, задающий длину вектора для векторных команд сопроцессора арифметики с плавающей точкой	32
SIR – регистр системного интегратора, через который осуществляются пересылки между регистрами RISC-процессора и сопроцессоров	64

2.4.11 Форматы команд, задающих пересылку данных типа «регистр сопроцессора арифметики с плавающей точкой - память»

Формат 1.1F Rист/пр- к ↔(fадр.(ARi, GRi)); ARi←fм(ARi, GRi)

31	30	29	28 27 26	25	24 23 22	21 20 19 18 17 16	15 0
1	0	1	MA	R/W	ARi	$R_{ист/пр-к}$	КОП СК

R/W	Направление пересылки данных
0	Чтение из памяти в $R_{ист/пр-к}$
1	Запись в память из $R_{ист/пр-к}$

MA	См. формат 1.1

Формат 1.2F Rист/пр- к ↔(fадр.(ARi, CM)); ARi←fм(ARi, CM)

31	30	29	28	27 26	25	24 23 22	21 20 19 18 17 16	15 0
1	1	1	0	MA	R/W	ARi	Rист/пр- к	КОП СК
63	АДРЕС (СМЕЩЕНИЕ)							32

R/W	Направление пересылки данных
0	Чтение из памяти в Rист/пр- к
1	Запись в память из Rист/пр- к

MA	См. формат 1.2

2.4.12 Форматы команд пересылки данных типа “регистр-регистр» между регистрами сопроцессора арифметики с плавающей точкой

Формат 2.1F Rист → Rпр-к

31	30	29	28	27 26 25 24	23 22 21 20 19 18	17 16 15 0
1	1	1	1	Rпр-к	Rист	КОП СК

Формат 2.2F РЕЗЕРВ (Команда запрещена)

31	30	29	28	27 26 25 24	23	22	21	20	19	18	17	16	15 0
1	1	0	0	Rпр-к			0	0	x	x	x	x	КОП СК
63	КОНСТАНТА												32

2.4.13 Арифметические векторные команды сопроцессора арифметики с плавающей точкой

Формат 3.1F (fap.(VR_A)) → VR_D

31	30	29	28 27	26	25	24 23	22 19	18 17	16	15 13	12 6	5 3	2 0
1	0	0	M	0	0	OP	0 0 0 0	DT	x	VR_A	x x x x x x x	VR_D	CELL_D

Поле маски АЛУ операции	M
Нет маски	0 0
Нет маски	0 1
Маскирование по нулевому значению маски	1 0
Маскирование по единичному значению маски	1 1

OP	Поле знака операции
0 0	A
0 1	|A|
1 0	- A
1 1	- |A|

Тип данных	DT
32-разрядные векторные данные	0 0
32-разрядные матричные данные	0 1
64-разрядные комплексные данные	1 0
64-разрядные данные	1 1

VR	Векторный регистр
0 0 0	VR0
0 0 1	VR1
0 1 1	VR2
0 1 1	VR3
1 0 0	VR4
1 0 1	VR5
1 1 0	VR6
1 1 1	VR7

Номер проц. ячейки	CELL_D
0	0 0 0
1	0 0 1
2	
3	0 1 1
Резерв	1 x x

Формат 3.2F (fap.(VR_B * VR_C)) → VR_D

31	30	29	28 27	26	25	24	23	22 19	18 17	16	15 13	12	11 9	8 6	5 3	2 0
1	0	0	M	0	1	x	OP	0 0 0 0	DT	x	x x	SB	VR_B	VR_C	VR_D	CELL_D

Поле маски АЛУ операции	M
Нет маски	0 0
Нет маски	0 1
Маскирование по нулевому значению маски	1 0
Маскирование по единичному значению маски	1 1

OP	Поле знака операции
0	B * C
1	- B * C

Тип данных	DT
32-разрядные векторные данные	0 0
32-разрядные матричные данные	0 1
64-разрядные комплексные данные	1 0
64-разрядные данные	1 1

VR	Векторный регистр
0 0 0	VR0
0 0 1	VR1
0 1 1	VR2
0 1 1	VR3
1 0 0	VR4
1 0 1	VR5
1 1 0	VR6
1 1 1	VR7

Номер проц. ячейки	CELL_D
0	0 0 0
1	0 0 1
2	
3	0 1 1
Резерв	1 x x

SB	Тип чтения из векторного регистра VR_B
0	Читаем векторные данные
1	Читаем скалярные данные

Примечание: При использовании 32-разрядных матричных данных в качестве операнда B 11-10 разряды команды задают чтение регистровой пары (128 разрядов за такт, а не 64, как обычно). Это позволяет за один процессорный такт заполнить матрицу 2 x 2 32-разрядных элементов. 9 разряд команды задаёт, каким образом формируется матрица: если 9 раряд равен нулю, матрица считывается по столбцам (один столбец из векторного регистра с чётным номером, второй – из векторного регистра с нечётным номером), если 9 раряд равен единице – по строкам (одна строка берётся из векторного регистра с чётным номером, вторая – из векторного регистра с нечётным номером). В первом случае это позволяет осуществить умножение матрицы B на вектор-строку C слева, во втором - умножение матрицы B на вектор-столбец C справа (см. рисунок 2.18 и пояснения к нему).

Формат 3.3F (fap.(VR_A + VR_ B)) → VR_D

31	30	29	28 27	26	25	24 23	22 19	18 17	16	15 13	12	11 9	8 6	5 3	2 0
1	0	0	M	1	0	OP	0 0 0 0	DT	WE	VR_A	SB	VR_B	MGC	VR_D	CELL_D

Поле маски АЛУ операции	M
Нет маски	0 0
Нет маски	0 1
Маскирование по нулевому значению маски	1 0
Маскирование по единичному значению маски	1 1

OP	Поле знака операции
0 0	A + B
0 1	A - B
1 0	- A + B
1 1	- A - B

VR	Векторный регистр
0 0 0	VR0
0 0 1	VR1
0 1 0	VR2
0 1 1	VR3
1 0 0	VR4
1 0 1	VR5
1 1 0	VR6
1 1 1	VR7

Номер проц. ячейки	CELL_D
0	0 0 0
1	0 0 1
2	
3	0 1 1
Резерв	1 x x

Тип данных	DT
32-разрядные векторные данные	0 0
32-разрядные матричные данные	0 1
64-разрядные комплексные данные	1 0
64-разрядные данные	1 1

Разрешение записи в VR_D в командах сравнения	WE
Нет записи	0
Есть запись	1

SB	Тип чтения из векторного регистра VR_B
0	Читаем векторные данные
1	Читаем скалярные данные

MGC	Условия генерации маски
0 0 0	Маска и признаки не формируются
0 0 1	Формируются только признаки
0 1 0	Формируются признаки и маска, если больше
0 1 1	Формируются признаки и маска, если меньше
1 0 0	Формируются признаки и маска, если равно
1 0 1	Формируются признаки и маска, если не равно
1 1 0	Формируются признаки и маска, если больше или равно
1 1 1	Формируются признаки и маска, если меньше или равно

Формат 3.4F (fap.(VR_A + VR_ B * VR_C)) → VR_D

Примечание: При использовании 32-разрядных матричных данных в качестве операнда В 11-10 разряды

31	30	29	28 27	26	25	24 23	22 19	18 17	16	15 13	12	11 9	8 6	5 3	2 0
1	0	0	M	1	1	OP	0 0 0 0	DT	SA	VR_A	SB	VR_B	VR_C	VR_D	CELL_D

Поле маски АЛУ операции	M
Нет маски	0 0
Нет маски	0 1
Маскирование по нулевому значению маски	1 0
Маскирование по единичному значению маски	1 1

OP	Поле знака операции
0 0	A + B*C
0 1	A - B*C
1 0	-A + B*C
1 1	-A - B*C

Тип данных	DT
32-разрядные векторные данные	0 0
32-разрядные матричные данные	0 1
64-разрядные комплексные данные	1 0
64-разрядные данные	1 1

VR	Векторный регистр
0 0 0	VR0
0 0 1	VR1
0 1 0	VR2
0 1 1	VR3
1 0 0	VR4
1 0 1	VR5
1 1 0	VR6
1 1 1	VR7

Номер проц. ячейки	CELL_D
0	0 0 0
1	0 0 1
2	
3	0 1 1
Резерв	1 x x

S	Тип чтения из векторного регистра VR_A или VR_B
0	Читаем векторные данные
1	Читаем скалярные данные

Примечание: При использовании 32-разрядных матричных данных в качестве операнда В 11-10 разряды команды задают чтение регистровой пары (128 разрядов за такт, а не 64, как обычно). Это позволяет за один процессорный такт заполнить матрицу 2 х 2 32-разрядных элементов. 9 разряд команды задаёт, каким образом формируется матрица: если 9 раряд равен нулю, матрица считывается по столбцам (один столбец из векторного регистра с чётным номером, второй – из векторного регистра с нечётным номером), если 9 раряд равен единице – по строкам (одна строка берётся из векторного регистра с чётным номером, вторая – из векторного регистра с нечётным номером). В первом случае это позволяет осуществить умножение матрицы В на вектор-строку С слева, во втором - умножение матрицы В на вектор-столбец С справа (см. рисунок 2.18 и пояснения к нему).

2.4.14 Векторные команды ввода-вывода сопроцессора арифметики с плавающей точкой

Формат 4.1F Чтение из памяти векторных данных VR_D <- (fадр.(ARi, GRi)); ARi←fм(ARi, GRi)

31	30	29	28 26	25	24 22	21	20	19	18	17 13	12	11 6	5 3	2 0
1	0	0	MA	0	ARi	0	0	1	x	количество	R	x x x x x x	VR_D	CELL_D

См. формат 1.1

См. формат 3.1F

R	**Источник количества повторов векторной команды**
0	Поле “количество ”в команде
1	Содержимое регистра VL

Формат 4.2F Запись в память векторных данных VR_D -> (fадр.(ARi, GRi)); ARi←fм(ARi, GRi)

31	30	29	28 26	25	24 22	21	20	19	18	17 13	12	11 9	8 6	5 0
1	0	0	MA	1	ARi	0	0	1	x	количество	R	VR_S	CELL_S	x x x x x x

См. формат 1.1

Источник количества повторов векторной команды	**R**
Поле “количество ”в команде	0
Содержимое регистра VL	1

CELL_S	**Номер проц. ячейки - источника**
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 x x	Резерв

Векторный регистр - источник	**VR_S**
VR0	**0 0 0**
VR1	**0 0 1**
VR2	**0 1 0**
VR3	**0 1 1**
VR4	**1 0 0**
VR5	**1 0 1**
VR6	**1 1 0**
VR7	**1 1 1**

Формат 4.3F Чтение из памяти векторных данных с последующим их преобразованием в блоке упаковки и распаковки данных: REPACK UNIT <- (fадр.(ARi, GRi)); ARi←fм(ARi, GRi)

31	30	29	28 26	25	24 22	21	20	19	18	17 13	12	11 6	5 0
1	0	0	MA	0	ARi	0	1	0	x	количество	R	x x x x x x	RPOP

См. формат 1.1

Источник количества повторов векторной команды	R
Поле “количество ”в команде	0
Содержимое регистра VL	1

Операция упаковки/распаковки	RPOP	
Форматы входных (SF) и выходных (преобразованных) данных (DF)	**SF**	**DF**
Два 32-разрядных числа в формате с фиксированной точкой	**0 0 0**	
Два 32-разрядных числа в формате с фиксированной точкой, но в дальнейшем используются только старшие 32-разряда (младшие 32-разряда обнуляются)	**0 0 1**	
Два 32-разрядных числа в формате с фиксированной точкой, но в дальнейшем используются только младшие 32-разряда (старшие 32-разряда обнуляются)	**0 1 0**	
Одно 64-разрядное число в формате с фиксированной точкой	**0 1 1**	
Два 32-разрядных числа в формате с плавающей точкой	**1 0 0**	
Два 32-разрядных числа в формате с плавающей точкой, но в дальнейшем используются только старшие 32-разряда (младшие 32-разряда обнуляются)	**1 0 1**	
Два 32-разрядных числа в формате с плавающей точкой, но в дальнейшем используются только младшие 32-разряда (старшие 32-разряда обнуляются)	**1 1 0**	
Одно 64-разрядное число в формате с плавающей точкой двойной точности	**1 1 1**	

Формат 4.4F Запись в память векторных данных из блока упаковки и распаковки данных: REPACK UNIT -> (fадр.(ARi, GRi)); ARi←fм(ARi, GRi)

	31	30	29	28 26	25	24 22	21	20	19	18	17 13	12	11 0
4.4F	1	0	0	MA	1	ARi	0	1	0	x	количество	R	x x x x x x x x x x x x

См. формат 1.1

Источник количества повторов векторной команды	R
Поле “количество ”в команде	0
Содержимое регистра VL	1

Примечание: поле “Количество” задает количество 64-разрядных слов, участвующих в операциях пересылок данных:

00000 - одно 64-разрядное слово;
00001 - два 64-разрядных слова;
…
11111 – тридцать два 64-разрядных слова.

2.4.15 Векторные команды пересылки данных для сопроцессора арифметики с плавающей точкой

Формат 5.1F Пересылка данных из векторного регистра одной из процессорных ячеек в векторный регистр другой ячейки: VR_D (SEL_D) <- VR_S (SEL_S)

Формат 5.2F Пересылка данных в зависимости от маски из векторного регистра VRS0 или VRS1 одной из процессорных ячеек в векторный регистр другой ячейки: VR_D (SEL_D) <- fмаск. (VRS0 (SEL_S), VRS1 (SEL_S))

Векторный регистр - источник	VRS
VR0	0 0 0
VR1	0 0 1
VR2	0 1 0
VR3	0 1 1
VR4	1 0 0
VR5	1 0 1
VR6	1 1 0
VR7	1 1 1

MS	Выбор типа маски
0	32-разрядная маска
1	64-разрядная маска

Формат 5.3F Пересылка данных из векторного регистра (пары векторных регистров) одной из процессорных ячеек с последующим их преобразованием в блоке упаковки и распаковки данных:

REPACK UNIT <- VRS_i (SEL_S), {VRS_{i+1} (SEL_S)}

Формат 5.4F Пересылка векторных данных из блока упаковки и распаковки в векторный регистр (в пару векторных регистров) одной из процессорных ячеек:

REPACK UNIT - >VRS_i (SEL_S), {VRS_{i+1} (SEL_S)}

2.4.16 Поле Rист/пр- к в командах пересылки данных сопроцессора арифметики с плавающей точкой

Код регистра	Регистр- источник	Регистр- приемник
0 0 0 0 0 0	FPFR0	FPFR0
0 0 0 0 0 1	SPMRL0	SPMRL0
0 0 0 0 1 0	SPMRH0	SPMRH0
0 0 0 0 1 1	DPMR0	SPMRH0
0 0 0 1 0 0	FPFR1	FPFR1
0 0 0 1 0 1	SPMRL1	SPMRL1
0 0 0 1 1 0	SPMRH1	SPMRH1
0 0 0 1 1 1	DPMR1	SPMRH1
0 0 1 0 0 0	FPFR2	FPFR2
0 0 1 0 0 1	SPMRL2	SPMRL2
0 0 1 0 1 0	SPMRH2	SPMRH2
0 0 1 0 1 1	DPMR2	SPMRH2
0 0 1 1 0 0	FPFR3	FPFR3
0 0 1 1 0 1	SPMRL3	SPMRL3
0 0 1 1 1 0	SPMRH3	SPMRH3
0 0 1 1 1 1	DPMR3	SPMRH3
0 1 0 0 0 0	FPIOIR	FPIOIR
0 1 0 0 0 1	FPOIR	FPOIR
0 1 0 0 1 0	FPUIR	FPUIR
0 1 0 0 1 1	FPIIR	FPIIR
0 1 0 1 0 0	FPDLIR	FPDLIR
0 1 0 1 0 1	FPIEIR	FPIEIR
0 1 0 1 1 0	RIER	RIER
0 1 0 1 1 1	IRRreset	-
0 1 1 0 0 0	-	-
0 1 1 0 0 1	SIR	SIR
0 1 1 0 1 0	RND_00	FPCR
0 1 1 0 1 1	RND_01	FPCR
0 1 1 1 0 0	RND_10	FPCR
0 1 1 1 0 1	RND_11	FPCR
0 1 1 1 1 0	Preset	FPCR
0 1 1 1 1 1	Pset	FPCR

Таблица 2.11 – Обозначение регистров сопроцессора арифметики с плавающей точкой и их назначение

Обозначение регистров и их назначение	Разрядность
Регистры процессорных ячеек сопроцессора арифметики с плавающей точкой	
FPFRi– регистр флагов процессорной ячейки i (i=0,…,3)	4
SPMRLi – младшая часть регистра маски одинарной точности процессорной ячейки i (i=0,…,3)	32
SPMRHi – старшая часть регистра маски одинарной точности процессорной ячейки i (i=0,…,3)	32
DPMRi – псевдо-регистр маски двойной точности процессорной ячейки i (i=0,…,3). Запись в него 32-разрядного значения вызывает запись одного и того же в регистры SPMRLi и SPMRHi	32
Регистры контроллера прерываний сопроцессора арифметики с плавающей точкой	
FPIOIR - информация о прерывании по некорректным данным	6
FPOIR - информация о прерывании по переполнению	6
FPUIR - информация о прерывании по потере значимости	6
FPIIR - информация о прерывании по потере точности	6
FPDLIR - информация о прерывании по потере данных при сложении трёх операндов	6
FPIEIR - информация о прерывании по неправильной команде	10
RIER – регистр порядка целого числа при переупаковке	11
IRRreset – сброс всех запросов на прерывание сопроцессора арифметики с плавающей точкой	-
Общие регистры сопроцессора арифметики с плавающей точкой	
SIR – регистр системного интегратора, через который осуществляются пересылки между регистрами RISC-процессора и сопроцессора	32
FPCR – регистр управления, содержащий бит параллельной работы и режимы округления	3
RND_00 – установка режима округления к ближайшему (по умолчанию) в регистре FPCR	-
RND_01 – установка режима округления к -∞ в регистре FPCR	-
RND_10 – установка режима округления к +∞ в регистре FPCR	-
RND_11 – установка режима округления к нулю в регистре FPCR	-
Preset – сброс бита параллельной работы в регистре FPCR	-
Pset – установ бита параллельной работы в регистре FPCR	-

3 Архитектура и структура вычислительного узла на базе процессорного ядра NMC4 (NMPU)

3.1 Структурная схема NMPU

Структурная схема NMPU приведена на рисунке 3.1.

Рисунок 3.1 - Структурная схема процессорного устройства NMPU

В состав вычислительного узла входят следующие блоки:

NMC4 – процессорное ядро, содержащее RISC-процессор и матрично-векторный сопроцессор арифметики с плавающей точкой (FPVCoP). Обмен ядра с внешним миром осуществляется с помощью следующих 64-разрядных шин: команд (IB), скалярных данных (SB), четырёх шин векторных входных данных (VBI0 – VBI3) и двух шин векторных выходных данных (VBO0 – VB1). Кроме того, используются шина адреса перехода по команде (IAGB) и шина адреса данных (AB).

IFU (Instruction Fetch Unit) - блок предвыборки команд, который выстраивает в единую очередь команды, считываемые из внутренней или внешней памяти системы по шине EIB в соответствии с адресом, выставляемым по шине IAB. Также IFU содержит кэш-память команд объёмом 1024*64 бит. При обращении во внешнюю память выбранные команды по

шине EIB попадают в кэш и одновременно перетранслируются по шине IB в процессорное ядро. При повторном обращении по тому же адресу команды будут выбираться из кэш по шине IB, а не из внешней памяти. Программы, расположенные во внутренней памяти, не кэшируются. Кэш команд в IFU позволяет эффективно работать с программами, расположенными во внешней памяти, и освободить программиста от дополнительной работы по трансляции программного кода из внешней во внутреннюю память. Процессор использует как 32-х, так и 64-разрядные команды.

SI (System Integrator) – системный интегратор, который осуществляет доступ во внешнюю и внутреннюю память за данными или к периферийным регистрам по запросу от процессорного ядра NMC4, а также во внешнюю и внутреннюю память за командами по запросу от IFU. Соответственно, для доступа во внешнюю память используется порт M_EMEM, являющийся мастером для шины AXI (M_COR), а к периферии – порт M_PER, являющийся мастером для периферийной шины APB.

IMU0-IMU7 (Internal Memory Unit 0-7) – 8 банков внутренней памяти объёмом по 512 Кбит (8К*64 разряда) каждый. Физически банк внутренней памяти состоит из двух полубанков памяти типа SRAM с организацией 4К х 64 каждый и имеет два порта – один со стороны процессорного ядра, другой – от каналов ПДП. Выбор полубанка памяти, в который производится обращение, осуществляется в зависимости от значения первого разряда адреса выполняемой операции (чтения или записи). Если первые разряды адресов обращений со стороны порта процессорного ядра и со стороны порта ПДП не совпадают, то будут обслужены оба запроса, а если первые разряды совпадают, то обслуживается только один запрос в соответствии с установленным приоритетом.

Банки внутренней памяти поддерживают доступ как за 32-х, так и 64-разрядными данными.

MPU (Memory Protect Unit) – блок защиты памяти NMPU. К нему приходит запрос на доступ во внутреннюю память вычислительного узла извне по порту S_MEM, который является ведомым портом шины AXI. Если запрос попадает в разрешённую область памяти, он обслуживается обычным образом, если нет, то он блокируется с формированием соответствующего прерывания.

CP0 - CP3 (Communication Port 0 - 3) – контроллеры ПДП для коммуникационных портов 0 – 3. Данные извне для процессорной системы поступают по двунаправленным 64-

разрядным каналам связи (LINK0 – LINK3), буферизуются и в режиме ПДП записываются в один из внутренних банков этой системы **(**IMU0-IMU7).

Выходные шины блоков CP0 - CP3, MPU и входные шины внутренних банков памяти IMU0-IMU7 объединяются с помощью шинного коммутатора.

PU (Peripheral Unit) – блок периферийных регистров процессорного ядра, которые содержат:

- регистры системного контроллера;
- регистры запросов и масок внешних прерываний;
- регистры управления и состояния двух интервальных таймеров.

3.2 Карта памяти процессорного узла NMPU

Рисунок 3.2– Карта памяти процессорного узла NMPU

Карты памяти всех узлов NMPU одинаковы и делятся на следующие области памяти:

- NMMB0 - NMMB7 - собственные банки внутренней памяти конкретного NMPU, причём в банке NMMB0 хранятся программы обработки прерываний и начальный загрузчик.
- REG_NMPU - блок собственных периферийных регистров конкретного NMPU. При доступе в эту область памяти идёт переадресация на периферийную шину APB.
- NMPU0 – NMPU3 - области внутренней памяти всех NMPU, входящих в состав своего кластера. Таким образом, любой NMPU может осуществить доступ во внутреннюю память соседа, а также в свою внутреннюю память, но не прямо, как при обращении в NMMB0 - NMMB7, а через внешнюю шину AXI.
- CMB0-CMB3 – банки внутренний памяти своего кластера.
- EMI_CU - банк внешней памяти DDR своего кластера. При этом при использовании DDR-памяти объемом 256M x 32 в кластере со стороны NMPU будут недоступны 1M младших ячеек.
- PCIe - области памяти высокоскоростного линка для связи своего кластера с внешними процессорными системами .
- CCPU - области памяти центрального процессорного узла.
- PC0 – PC3 - адресные пространства всех кластеров, что позволяет NMPU обмениваться данными как с управляющими ПУ своего и чужих кластеров, так и с NMPU своего и чужих кластеров. При этом обращение в эти области памяти будут выполняться заметно медленнее, чем в остальные области.
- Reserved – данная область памяти, где 31 или 30 разряд адреса равен 1, не используется. Обращение в данную область со стороны NMPU означает выполнение эксклюзивной операции, при этом адрес конкретной ячейки памяти определяется разрядами с 29-го по 0-й.

3.3 Системный интегратор и организация доступа в память

Блок системного интегратора (SI) предназначен для обеспечения многотактового параллельного доступа процессорного ядра **NMC4** к внутренней и внешней памяти вычислительного узла NMPU. Блок системного интегратора формирует запросы на чтение или запись, осуществляет арбитраж между ними и коммутирует входные и выходные шины процессорного ядра с входными и выходными шинами данных разных банков памяти.

В данной реализации системный интегратор поддерживает одновременный доступ к восьми банкам внутренней памяти, одному банку внешней памяти по отношению к вычислительному узлу NMPU и к шине периферийных устройств.

Структурная схема подсистемы работы с памятью процессорного ядра NMC4 представлена на рисунке ниже (см. Рисунок 3.3). На данной структурной схеме показано взаимодействие блока системного интегратора SI, и другими узлами вычислительного узла.

Рисунок 3.3 – Структурная схема подсистемы работы с памятью

Особенностью процессорного ядра NMC4, позволяющей вести параллельную выборку данных из разных банков памяти, является наличие в системе команд векторных команд. Векторная команда может выполняться различное количество тактов – от 1 до 32 – в зависимости от кода команды. Если векторная команда однотактовая, то её работа в конвейере ни чем особенным не отличается от работы обычной скалярной команды. В случае, когда векторная команда выполняется несколько тактов, это можно рассматривать как аппаратную организацию цикла, состоящего из одной однотактовой векторной команды.

В каждом такте процессорное ядро может выдать в системный интегратор запрос на чтение или запись данных. При этом на адресной шине ядра формируется адрес первого обращения в память, смещение, позволяющее вычислять последующие адреса и количество обращений. Кроме этого ядром формируется информация, указывающая на входную или выходную шину ядра, по которой будут передаваться данные при выполнении этой операции с памятью. Первый адрес, смещение и количество повторов фиксируется свободным адресным генератором данных, который в каждом такте формирует непосредственный запрос к банкам памяти. Следующий запрос от ядра занимает второй адресный генератор и т.д. Если запросы, формируемые процессорным ядром, обращаются в разные банки памяти, то происходит параллельное обслуживание запросов. Если запросы от ядра обращаются к одному банку памяти, то они обслуживаются последовательно в порядке поступления.

Кроме запросов на чтение или запись данных, на системный интегратор поступают запросы на выборку команд, генерируемые в блоке IFU. В системном интеграторе они рассматривается как выделенный запрос на чтение данных со своей входной шиной.

В каждом такте системный интегратор осуществляет арбитраж доступа к внешней и внутренней памяти и адрес выигравшего запроса подается непосредственно на адресные входы банков внутренней памяти или на блоки интерфейса с внешней памятью или периферийными устройствами. Одновременно системный интегратор формирует управляющие сигналы для коммутатора шин данных, которые будут необходимы для пересылки данных между ядром и памятью.

3.3.1 Интерфейс системного интегратора с внешней памятью

Блок интерфейса с внешней памятью позволяет процессорному ядру осуществлять произвольный доступ к разделяемой памяти, к внутренней памяти остальных процессорных устройств, к периферийным устройствам кластера и к внешней по отношению к кластеру памяти.

Блок интерфейса системного интегратора с внешней памятью выполнен как 64-разрядный master-порт в соответствии со стандартом AMBA AXI 3.0. Данный порт поддерживает однократные операции чтения и записи с разрядностью шин данных в 32- и 64-бита. Аппаратура порта позволяет выполнять как обычные, так и эксклюзивные операции на внешней шине.

3.3.2 Интерфейс системного интегратора с периферийными устройствами

Блок интерфейса системного интегратора с периферийными устройствами предназначен для доступа к программно-доступным регистрам периферийных устройств, принадлежащих данному вычислительному узлу NMPU.

Блок интерфейса с периферийными устройствами выполнен как 32-разрядный master-порт в соответствии со стандартом AMBA AXI 3.0. Данный порт поддерживает однократные операции чтения и записи только с разрядностью шин данных в 32-бита.

3.4 Внутренняя память процессорного устройства NMPU

Объем внутренней памяти вычислительного узла NMPU составляет 4 Мбит. Она разделена на восемь банков памяти по 512 Кбит каждый.

Каждый банк внутренней памяти (NMMB0... NMMB7) имеет двухпортовую организацию. По одному из портов производится чтение и запись данных процессорным ядром, а также выборка команд процессора. Второй порт (порт ПДП) обслуживает чтение и запись данных под управлением контроллеров ПДП, а также чтение и запись данных со стороны других процессорных устройств кластера, управляющего процессорного узла (CPU) или центрального управляющего узла (CCPU). Таким образом, банк внутренней памяти способен отработать до двух запросов в каждом процессорном такте.

Структурная схема банка внутренней памяти представлена на рисунке 3.4.

Рисунок 3.4 – Структурная схема банка внутренней памяти вычислительного узла NMPU

Физически банк внутренней памяти состоит из двух полубанков памяти типа SRAM с организацией 4К х 64бит каждый (на рисунке обозначены как ODD и EVEN). Выбор полубанка памяти, в который производится обращение, осуществляется в зависимости от

значения первого разряда адреса выполняемой операции (чтения или записи). Если первые разряды адресов обращений со стороны порта процессорного ядра и со стороны порта ПДП не совпадают, то будут обслужены оба запроса, а если первые разряды совпадают, то обслуживается только один запрос в соответствии с установленным приоритетом.

Порт процессорного ядра содержит аппаратуру, реализующую заложенный принцип в конвейере процессорного ядра отложенной записи (Late Write). Реализация данного принципа позволяет выравнить конвейеры операций чтения и записи. Аппаратурой, поддерживающей такое выравнивание, в банке внутренней памяти являются буфер адресов операций записи WA_BUF и буфер записываемых данных WD_BUF с соответствующими выходными мультиплексорами.

Реализация принципа отложенной записи в конвейере процессорного ядра предполагает выдачу адресов операций записи и адресов операций чтения на одной ступени конвейера. Операции чтения выполняется в банке памяти естественным образом. Адреса же операций записи фиксируются в буфере WA_BUF и дожидаются прихода записываемых данных. При этом конвейер чтения в банке внутренней памяти не блокируется и последующие операции чтения могут выполняться “в обход” этих операций записи по шине RD_BUS.

Выдаваемые процессорным ядром данные для операций записи по шине WR_BUS фиксируются в соответствующей ячейке буфера WD_BUF и будут записаны в полубанк памяти в моменты отсутствия запросов на чтение данных из этого банка памяти. Глубина конвейера процессорного ядра при работе с внутренней памятью определяет размер буферов WA_BUF и WD_BUF – по 5 ячеек каждый.

“Обход” операций записи последующими операциями чтения возможен только, если адреса операций чтения не совпадают с адресами операций записи, находящимися в буфере WA_BUF. В случае совпадения адресов конвейер чтения из банка памяти аппаратно блокируется до того момента, когда все данные операций записи, чьи адреса находятся в буфере WA_BUF, будут записаны в соответствующие полубанки памяти. Необходимо отметить, что сравнение адресов операций чтения с ячейками буфера WA_BUF происходит только при выполнении операций чтения данных ядром процессора. При выполнении выборки команд такого сравнения не происходит.

Буфер RDR предназначен для фиксации данных, считываемых из полубанков памяти при блокировке конвейера чтения со стороны процессорного ядра. Это позволяет освободить

полубанки памяти от операций чтения данных и осуществлять доступ к ним со стороны порта ПДП.

Порт ПДП банка памяти выполнен как 64-разрядный slave-порт в соответствии со стандартом AMBA AXI 3.0. Данный порт поддерживает однократные операции чтения и записи с разрядностью шин данных в 32- и 64-бита. Порт ПДП банка памяти не поддерживает невыровненные операции на шине, те при 32-битном доступе адрес обращения должен быть выровнен на границу 32-разрядного слова, а при 64-разрядном доступе – на границу 64-разрядного слова.

И операции чтения, и операции записи по порту ПДП выполняются строго последовательно в порядке поступления. Порядок чередования операций чтения и записи между собой может быть произвольным. Сравнение адресов чтения или записи порта ПДП с адресами хранящимися в буфере WA_BUF порта процессорного ядра не производится.

Буфера AWFIFO, ARFIFO, WFIFO и RFIFO позволяют осуществлять непрерывную передачу данных по шине ПДП.

3.5 Блок предвыборки команд IFU и управление кэш-памятью команд

Блок предвыборки команд IFU выстраивает в единую очередь команды, считываемые из внутренней или внешней памяти системы. Также IFU содержит кэш-память команд объёмом 1024*64 бит. При обращении во внешнюю память выбранные команды попадают в кэш и одновременно перетранслируются в процессорное ядро. При повторном обращении по тому же адресу команды будут выбираться из кэш, а не из внешней памяти. Программы, расположенные во внутренней памяти, не кэшируются. Кэш команд в IFU позволяет эффективно работать с программами, расположенными во внешней памяти, и освободить программиста от дополнительной работы по трансляции программного кода из внешней во внутреннюю память. Процессор использует как 32-х, так и 64-разрядные команды, но выборка идёт всегда по 64 разряда (выбирается либо одна 64-разрядная команда, либо сразу две 32-разрядные).

Основные характеристики кэш-памяти команд:

- объём 8 Кбайт (1К*64 разряда);
- наборно-ассоциативная организация (8 way set-associative cache);
- аппаратная поддержка до восьми промахов в кэш без блокировки выборки команд;
- предвыборка до 8 команд из внешней памяти;
- переменная величина страницы кэш (до 128*64 разряда);

- использование LRU алгоритма замещения страниц (Least Recently Used).
- возможность программной заморозки кэш, её очистки и включения/выключения;
- возможность программного распределения кэшируемых и некэшируемых областей внешней памяти.

Управление кэш-памятью команд осуществляется с помощью программного доступа к периферийным регистрам, расположенным в памяти по адресам от 0x0002_0C00h до 0x0002_0C08h, описание которых дано ниже.

3.5.1 Программно доступные регистры блока управления кэш-памятью

Блок содержит два программно доступных регистра:

- регистр управления,
- регистр адреса гиперстраницы.

Имеются 3 способа доступа к регистру управления:

- нормальный доступ - устанавливает в регистре записываемое значение;
- побитовая установка - устанавливает значение 1 в тех разрядах, в которых имеются единицы в записываемом данном, остальные разряды не меняются;
- побитовый сброс - устанавливает значение 0 в тех разрядах, в которых имеются единицы в записываемом данном, остальные разряды не меняются.

Для каждого способа записи выделен отдельный адрес. Чтение регистра может производиться по любому из описанных адресов. Регистр адреса гиперстраницы имеет только нормальный способ доступа.

Таблица 3.1 – Программно доступные регистры блока управления кэш-памятью

Адрес (32-разрядный доступ)	Название	Доступ	Описание
0x0002_0C00h	CHSR	ЧТ/ЗП	Регистр управления кэш-памятью команд
0x0002_0C02h	CHSR_SET	ЧТ/ЗП	Побитовая установка регистра CHSR
0x0002_0C04h	CHSR_CLR	ЧТ/ЗП	Побитовый сброс регистра CHSR
0x0002_0C08h	PDA	ЧТ/ЗП	Регистр адреса гиперстраницы

Регистр управления кэш-памятью команд

Таблица 3.2 – Формат регистра управления кэш-памятью команд

Разряды	Название	Доступ	Описание
31-3	-	-	Зарезервировано.
2	CRST	ЗП	Запись значения 1 в данное поле сбрасывает содержимое кэш-памяти. При чтении возвращается 0.
1	GTWE	ЧТ/ЗП	Разрешение аппаратной записи в регистр гиперстраницы: 0 - аппаратная запись разрешена, 1 - аппаратная запись запрещена.
0	CEN	ЧТ/ЗП	Разрешение работы кэш-памяти: 0 - при выборке команд из внешней памяти не используется кэш-память, 1 - при выборке команд из внешней памяти используется кэш-память.

Бит CEN определяет, будет ли при выборке команд из внешней памяти анализироваться состояние кэш-памяти. Если работа кэш-памяти запрещена, то команды будут выбираться из внешней памяти, минуя кэш-память. Необходимо отметить, что при нулевом значении данного бита состояние кэш-памяти не изменяется, то есть сброс данного поля означает “заморозку” кэш-памяти в состоянии на момент сброса бита CEN.

Бит GTWE разрешает аппаратную запись в регистр адреса гиперстраницы. Если аппаратная запись в регистр GPA разрешена, то при обращении за командой в гиперстраницу, отличную от отраженной в кэш-памяти, произойдет сброс содержимого кэш-памяти и замещение гиперстраницы в регистре GPA. Кэш-память в этом случае будет заполняться с состояния как после системного сброса.

Если аппаратная запись в регистр GPA запрещена, то при обращении за командой в гиперстраницу, отличную от отраженной в кэш-памяти, состояние кэш-памяти не изменится, а команды будут выбираться из внешней памяти так, как будто кэш- память отсутствует.

Данный механизм, в сочетании с программной записью регистра гиперстраницы, позволяет разделить внешнюю память на кэшируемую и не кэшируемую области.

Регистр адреса гиперстраницы

Таблица 3.3 - Формат регистра адреса гиперстраницы

Разряды	Название	Описание
31-20	GyperTag	Поле загрузки адреса гиперстраницы.
19-0	-	Зарезервировано

Регистр предназначен для загрузки адреса гиперстраницы внешней памяти, отражаемой в кэш-памяти. Регистр доступен для записи и чтения. Запись в данный регистр сбрасывает содержимое кэш-памяти и устанавливает записываемое значение как адрес текущей гиперстраницы. Если разрешена аппаратная запись в регистр гиперстраницы, содержимое регистра может меняться, но при чтении возвращается всегда последнее записанное программным способом значение. Если с момента системного сброса записи в регистр не производилось, то возвращается неинициализированное значение.

3.6 Блок защиты памяти MPU

Блок защиты памяти разрешает доступ внешним устройствам в выделенные области памяти NMPU.

Внешние устройства имеют произвольный доступ к внутренней памяти процессорной системы NMPU через интерфейс AXI3 Slave процессорной системы. Блок защиты памяти находится внутри NMPU и непосредственно подключен к интерфейсу AXI3 Slave.

Блок защиты памяти декодирует адрес транзакции и определяет, разрешён или запрещён доступ по этому адресу. Блок пропускает разрешённые транзакции без изменения, а запрещённые транзакции обрабатывает следующим образом:

- с точки зрения внешнего устройства - транзакция завершается с кодом ошибки SLVERR, при этом в транзакции чтения вместо данных во всех передачах транзакции возвращаются нулевые значения,
- с точки зрения процессорной системы NMPU - генерируется (маскируемый по каждой выделенной области) запрос на прерывание к процессору NMC4, который снимается программно.

Интерфейсы чтения и записи AXI3 разделены, поэтому в NMPU присутствуют 2 экземпляра блоков защиты памяти: по одному на чтение и на запись.

Рисунок 3.5 - Структурная схема блока защиты памяти NMPU

Каждый экземпляр блока защиты памяти состоит из двух частей:

- блока управления - который содержит конфигурационные регистры устройства и генерирует сигнал (MASK[31:0]) маски доступных областей памяти,
- блока обработки транзакций - который работает с сигналами шины AXI3, выдаёт сигнал (ERRSET[31:0]) обращения в запрещённую область.

Блок обработки транзакций производит сравнение адресов и самостоятельно завершает транзакции AXI3, адреса которых попадают в запрещённые сегменты. С помощью рисунка 3.6. ниже поясняется работа блока защиты памяти на примере транзакций чтения.

Рисунок 3.6 - Структурная схема блока обработки транзакций

В обычном режиме блок обработки транзакций не вмешивается в обработку проходящих транзакций, только ведёт подсчёт текущего количества незавершённых транзакций (для которых прошла передача адреса и ещё не было передачи последних данных чтения). Как только на входном интерфейсе блока появляется адрес, попадающий в защищённую область, блок блокирует канал адреса, дожидается передачи всех данных от незавершённых транзакций и после этого выдаёт нулевые данные и код ошибки (RRESP=10b, ошибка конечного устройства).

Элементы архитектуры блока (см. рисунок 3.6):

axhs - блок квитирования в канале адреса,
xhs - блок квитирования в канале данных чтения,
xID - регистр идентификатора AXI3 (используется для хранения ID запрещённой транзакции),
Rcnt - счётчик возвращаемых данных в пакете при чтении,

comparision block - комбинационная схема декодирования адреса (детектирует попадание в запрещённый сегмент),
ES - регистр-признак режима обработки запрещённой транзакции (во время обработки запрещённой транзакции блокируется приём адресов от внешнего устройства),
UTC - счётчик незавершённых транзакций (используется для освобождения конвейера от уже выполняющихся в NMCell транзакций).

Блок обработки транзакций сообщает блоку управления об обращении в защищённый сегмент с помощью шины ERRSETx. Блок управления фиксирует сигналы ERRSETx, в зависимости от настройки может генерировать запрос на прерывание к ядру NMC.

Рисунок 3.7 - Структурная схема блока управления

3.6.1 Разделение внутренней памяти NMPU на защищаемые сегменты

Внутренняя память NMPU разделяется на 32 защищаемых сегмента, каждый размером 16 Кбайт. Сегменту с номером i соответствует один бит с номером i в каждом регистре управления блоком. Ниже в таблице 3.4 показана нумерация сегментов.

Таблица 3.4 – Нумерация сегментов внутренней памяти процессорной системы NMPU

Номер сегмента	Адреса в собственном пространстве NMPU (32-разрядный доступ)	Смещение относительно базового адреса в карте памяти внешних NMPU (32-разрядный доступ)	Смещение относительно базового адреса в карте памяти ARM и устройств ввода-вывода (байтовый доступ)
0	0000_0000h-0000_0FFFh	0000_0000h-0000_0FFFh	0000_0000h-0000_3FFFh
1	0000_1000h-0000_1FFFh	0000_1000h-0000_1FFFh	0000_4000h-0000_7FFFh
2	0000_2000h-0000_2FFFh	0000_2000h-0000_2FFFh	0000_8000h-0000_BFFFh
3	0000_3000h-0000_3FFFh	0000_3000h-0000_3FFFh	0000_C000h-0000_FFFFh
4	0000_4000h-0000_4FFFh	0000_4000h-0000_4FFFh	0001_0000h-0001_3FFFh
5	0000_5000h-0000_5FFFh	0000_5000h-0000_5FFFh	0001_4000h-0001_7FFFh
6	0000_6000h-0000_6FFFh	0000_6000h-0000_6FFFh	0001_8000h-0001_BFFFh
7	0000_7000h-0000_7FFFh	0000_7000h-0000_7FFFh	0001_C000h-0001_FFFFh
8	0000_8000h-0000_8FFFh	0000_8000h-0000_8FFFh	0002_0000h-0002_3FFFh
9	0000_9000h-0000_9FFFh	0000_9000h-0000_9FFFh	0002_4000h-0002_7FFFh
10	0000_A000h-0000_AFFFh	0000_A000h-0000_AFFFh	0002_8000h-0002_BFFFh
11	0000_B000h-0000_BFFFh	0000_B000h-0000_BFFFh	0002_C000h-0002_FFFFh
12	0000_C000h-0000_CFFFh	0000_C000h-0000_CFFFh	0003_0000h-0003_3FFFh
13	0000_D000h-0000_DFFFh	0000_D000h-0000_DFFFh	0003_4000h-0003_7FFFh
14	0000_E000h-0000_EFFFh	0000_E000h-0000_EFFFh	0003_8000h-0003_BFFFh
15	0000_F000h-0000_FFFFh	0000_F000h-0000_FFFFh	0003_C000h-0003_FFFFh
16	0001_0000h-0001_0FFFh	0001_0000h-0001_0FFFh	0004_0000h-0004_3FFFh
17	0001_1000h-0001_1FFFh	0001_1000h-0001_1FFFh	0004_4000h-0004_7FFFh
18	0001_2000h-0001_2FFFh	0001_2000h-0001_2FFFh	0004_8000h-0004_BFFFh
29	0001_3000h-0001_3FFFh	0001_3000h-0001_3FFFh	0004_C000h-0004_FFFFh
20	0001_4000h-0001_4FFFh	0001_4000h-0001_4FFFh	0005_0000h-0005_3FFFh
21	0001_5000h-0001_5FFFh	0001_5000h-0001_5FFFh	0005_4000h-0005_7FFFh
22	0001_6000h-0001_6FFFh	0001_6000h-0001_6FFFh	0005_8000h-0005_BFFFh
23	0001_7000h-0001_7FFFh	0001_7000h-0001_7FFFh	0005_C000h-0005_FFFFh
24	0001_8000h-0001_8FFFh	0001_8000h-0001_8FFFh	0006_0000h-0006_3FFFh
25	0001_9000h-0001_9FFFh	0001_9000h-0001_9FFFh	0006_4000h-0006_7FFFh
26	0001_A000h-0001_AFFFh	0001_A000h-0001_AFFFh	0006_8000h-0006_BFFFh
27	0001_B000h-0001_BFFFh	0001_B000h-0001_BFFFh	0006_C000h-0006_FFFFh
28	0001_C000h-0001_CFFFh	0001_C000h-0001_CFFFh	0007_0000h-0007_3FFFh
29	0001_D000h-0001_DFFFh	0001_D000h-0001_DFFFh	0007_4000h-0007_7FFFh
30	0001_E000h-0001_EFFFh	0001_E000h-0001_EFFFh	0007_8000h-0007_BFFFh
31	0001_F000h-0001_FFFFh	0001_F000h-0001_FFFFh	0007_C000h-0007_FFFFh

3.6.2 Программно доступные регистры блока защиты памяти

Регистры блока представлены в карте памяти процессора NMPU. Имеются 4 способа записи в каждый регистр:

нормальный доступ - устанавливает в регистре записываемое значение,

побитовая установка - устанавливает значение 1 в тех разрядах, в которых имеются единицы в записываемом данном, остальные разряды не меняются,

побитовый сброс - устанавливает значение 0 в тех разрядах, в которых имеются единицы в записываемом данном, остальные разряды не меняются,

обнуление - устанавливает значение 0 во всех битах регистра, независимо от записываемого значения (т.е. имеет значение сам факт записи).

Для каждого способа записи выделен отдельный адрес. Некоторые способы для некоторых регистров недоступны. Чтение регистра может производиться по любому из описанных адресов. Ниже показаны адреса регистров блока защиты памяти.

Таблица 3.5 – Перечень программно доступных регистров блока защиты памяти

Адрес (32-разрядный доступ)	Название	Доступ	Описание
0002_1400h	RD_MASK	ЧТ/ЗП	Маска защищаемых сегментов: 0 - обращение в данный сегмент по чтению разрешено, 1 - обращение в данный сегмент по чтению блокируется.
0002_1402h	RD_MASK_SET	ЗП	Побитовая установка регистра RD_MASK
0002_1404h	RD_MASK_CLEAR	ЗП	Побитовый сброс регистра RD_MASK
0002_1406h	RD_MASK_NULL	ЗП	Обнуление регистра RD_MASK
0002_1408h	RD_IRQ_STATUS_RAW	ЧТ	Статус запросов на прерывание без учёта маски: 0 - обращений по чтению в данный сегмент не было, 1 - зафиксировано обращение по чтению в данный сегмент.
0002_140Ah	RD_IRQ_STATUS_RAW_SET	ЗП	Побитовая установка RD_IRQ_STATUS_RAW (только для отладки)
0002_140Ch	RD_IRQ_STATUS_RAW_CLEAR	ЗП	Побитовый сброс регистра RD_IRQ_STATUS_RAW
0002_140Eh	RD_IRQ_STATUS_RAW_NULL	ЗП	Обнуление регистра RD_IRQ_STATUS_RAW
0002_1410h	RD_IRQ_MASK	ЧТ/ЗП	Маска запросов на прерывание от регистра RD_IRQ_STATUS_RAW: 0 - прерывание запрещено, 1 - прерывание разрешено.
0002_1412h	RD_IRQ_MASK_SET	ЗП	Побитовая установка регистра RD_IRQ_MASK
0002_1414h	RD_IRQ_MASK_CLEAR	ЗП	Побитовый сброс регистра RD_IRQ_MASK
0002_1416h	RD_IRQ_MASK_NULL	ЗП	Обнуление регистра RD_IRQ_MASK
0002_1418h	RD_IRQ_STATUS	ЧТ	Статус запросов на прерывание с учётом маски: 0 - нет запроса, 1 - есть запрос.
0002_141Ah	Зарезервировано	-	
0002_141Ch	Зарезервировано	-	

Продолжение таблицы 3.5

Адрес (32-разрядный доступ)	Название	Доступ	Описание
0002_141Eh	RD_IRQ_STATUS_NULL	ЗП	Обнуление незамаскированных запросов на прерывание (сбрасываются все биты регистра RD_IRQ_STATUS и все биты регистра RD_IRQ_STATUS_RAW, для которых установлен бит RD_IRQ_MASK).
0002_1440h	WR_MASK	ЧТ/ЗП	Маска защищаемых сегментов: 0 - обращение в данный сегмент по чтению разрешено, 1 - обращение в данный сегмент по чтению блокируется.
0002_1442h	WR_MASK_SET	ЗП	Побитовая установка регистра WR_MASK
0002_1444h	WR_MASK_CLEAR	ЗП	Побитовый сброс регистра WR_MASK
0002_1446h	WR_MASK_NULL	ЗП	Обнуление регистра WR_MASK
0002_1448h	WR_IRQ_STATUS_RAW	ЧТ	Статус запросов на прерывание без учёта маски: 0 - обращений по чтению в данный сегмент не было, 1 - зафиксировано обращение по чтению в данный сегмент.
0002_144Ah	WR_IRQ_STATUS_RAW_SET	ЗП	Побитовая установка WR_IRQ_STATUS_RAW (только для отладки)
0002_144Ch	WR_IRQ_STATUS_RAW_CLEAR	ЗП	Побитовый сброс регистра WR_IRQ_STATUS_RAW
0002_144Eh	WR_IRQ_STATUS_RAW_NULL	ЗП	Обнуление регистра WR_IRQ_STATUS_RAW
0002_1450h	WR_IRQ_MASK	ЧТ/ЗП	Маска запросов на прерывание от регистра WR_IRQ_STATUS_RAW: 0 - прерывание запрещено, 1 - прерывание разрешено.
0002_1452h	WR_IRQ_MASK_SET	ЗП	Побитовая установка регистра WR_IRQ_MASK
0002_1454h	WR_IRQ_MASK_CLEAR	ЗП	Побитовый сброс регистра WR_IRQ_MASK
0002_1456h	WR_IRQ_MASK_NULL	ЗП	Обнуление регистра WR_IRQ_MASK
0002_1458h	WR_IRQ_STATUS	ЧТ	Статус запросов на прерывание с учётом маски: 0 - нет запроса, 1 - есть запрос.
0002_145Ah	Зарезервировано	-	
0002_145Ch	Зарезервировано	-	
0002_145Eh	WR_IRQ_STATUS_NULL	ЗП	Обнуление незамаскированных запросов на прерывание (сбрасываются все биты регистра WR_IRQ_STATUS и все биты регистра WR_IRQ_STATUS_RAW, для которых установлен бит WR_IRQ_MASK).

3.6.3 Прерывания

Каждый экземпляр (чтение и запись) блока защиты памяти имеет один сигнал запроса на прерывание к NMPU. Сигнал запроса установлен, пока хотя бы для одного из сегментов RD/WR_IRQ_STATUS_RAW[i]=1 и RD/WR_IRQ_MASK[i]=0. Запрос можно снять программно одним из двух способов:

- с помощью регистра RD/WR_IRQ_CLEAR_RAW - побитовый сброс битов регистра RD/WR_IRQ_STATUS_RAW,

- с помощью регистра RD/WR_IRQ_STATUS_CLEAR – сброс всех незамаскированных битов регистра RD/WR_IRQ_STATUS_RAW.

3.7 Блок коммуникационного порта

Блок коммуникационного порта (CPx) служит для обмена данными между 64-разрядным каналом связи (LINK) и оперативной памятью системы. Контроллер обеспечивает устройству на коммуникационной шине прямой доступ в память. Управление блоком CPx осуществляет процессорное ядро NMC4 с помощью периферийной шины.

Блок CPx состоит из двух независимых частей: приёмной и передающей (см. Рисунок 3.8). Обе части блока коммуникационного порта могут работать одновременно, обеспечивая полнодуплексный режим коммуникационной шины.

Рисунок 3.8 - Структурная схема блока коммуникационного порта

Интерфейс с каналом связи соответствует спецификации AMBA AXI4 Stream. Блок коммуникационного порта имеет 2 таких интерфейса, отличающихся только направлением передачи данных: приёмный – через который данные поступают от канала связи к контроллеру, и передающий – через который данные передаются от блока CPx на внешний канал связи.

Выбран простой вариант интерфейса, в который включены сигналы квитирования (handshake) и шина данных.

Интерфейс прямого доступа в память соответствует спецификации шины AMBA AXI3. Блок коммуникационного порта является ведущим устройством шины AMBA AXI3 и имеет полный набор каналов интерфейса: канал адреса записи (AW), канал данных записи (DW), канал подтверждения записи (RW), канал адреса чтения (AR), канал данных чтения (DR).

Обе части блока коммуникационного порта производят одиночные транзакции на шине AXI3. Во всех транзакциях для передачи данных используются все разряды шины данных. Особенностью контроллера является тот факт, что разрядность шины данных приёмного интерфейса с коммуникационной шиной равна разрядности шины данных в канале данных записи интерфейса ПДП, разрядность шины данных передающего интерфейса с коммуникационной шиной равна разрядности шины данных в канале данных чтения интерфейса ПДП. Разрядность шины данных приёмного интерфейса и разрядность шины данных передающего интерфейса равна 64 бита.

Периферийный интерфейс APB блока соответствует спецификации шины AMBA APB2. Этот интерфейс используется для чтения и записи программно доступных регистров контроллера.

Системные сигналы – тактовый сигнал и сигнал сброса. Блок коммуникационного порта использует общие системные сигналы шинной подсистемы AMBA.

Интерфейс запросов на прерывание представляет собой один сигнал IRQ запроса на прерывание. Высокий уровень сигнала IRQ сигнализирует процессорному ядру о запросе на прерывание.

3.7.1 Приёмная часть блока коммуникационного порта

3.7.1.1 Структура приемной части блока коммуникационного порта

Структура приёмной части показана ниже (см. рисунок 3.9).

Рисунок 3.9 - Структура приемной части блока коммуникационного порта

Адресный генератор (АГ), работающий в двух режимах: одномерной и двухмерной адресации. Подробнее работа адресных генераторов контроллера описана ниже.

Буфер данных (БД), размер которого шесть 64-х разрядных слов, служит для временного хранения данных.

Блок управления (БУ), который формирует управляющие сигналы для остальных блоков и шин контроллера. В состав блока управления входят автомат состояния, основной счётчик данных, регистр масок прерывания, бит ошибки/остановки, а также дополнительные счётчики управления: счётчик незавершённых транзакций, счётчик доступных данных.

Конвейерные регистры на шинах данных и адреса используются для улучшения временных характеристик схемы.

3.7.1.2 Принципы работы

Данные поступают в конвейерный регистр через интерфейс AXI4 Stream. Из конвейерного регистра данные записываются в БД. Если буфер данных заполнен, то интерфейс коммуникационной шины блокируется, данные не могут быть приняты.

Из БД данные записываются в память по шине AXI3. Данные записываются одиночными запросами. Запрос в канал адреса не выставляется, пока в БД нет данных.

Регистры приемной части блока коммуникационного порта, доступные через периферийную шину:

регистр текущего адреса,

основной счётчик данных,

регистр смещения адреса,

счётчик последовательных адресов,

регистр режима адресации,

регистр управления,

регистр масок прерывания,

регистр состояния контроллера.

Текущее состояние блока коммуникационного порта определяется автоматом состояний, битом ошибки/остановки и значениями управляющих счётчиков.

Основной счётчик данных работает следующим образом:

переключение -1 при каждой выдаче адреса в канал адреса, ИЛИ переключение -1 при каждом чтении из БД в состоянии очистки буфера.

Счётчик незавершённых транзакций работает следующим образом:

переключение +1 при каждой выдаче адреса в канал адреса,

переключение -1 при каждом получении подтверждения в канале подтверждения записи.

Счётчик доступных данных работает следующим образом:

переключение +1 при каждом приёме слова данных в БД,

переключение -1 при каждой выдаче адреса в канал адреса ИЛИ переключение -1 при каждом чтении из БД в состоянии очистки буфера.

Бит ошибки/остановки устанавливается автоматически при получении в канале подтверждения записи сигнала ошибки записи, этот бит может быть установлен и снят программно.

Автомат состояний приемной части блока коммуникационного порта показан ниже (см. рисунок 3.10).

Рисунок 3.10 - Автомат состояний приемной части ККП

Idle – состояние бездействия, начальное состояние. В этом состоянии не выдаются запросы на запись.

ReceiveWrite – приём и запись данных, основное рабочее состояние. В этом состоянии разрешены запросы на запись. Условия перехода из этого состояния в состояние UncompleteWrite: основной счётчик достиг нуля ИЛИ установка (программная или аппаратная) бита ошибки/остановки.

UncompleteWrite – состояние незавершённой записи, ожидание подтверждения на запись последнего данного. В этом состоянии не выдаются запросы на запись. Условие перехода в состояние Complete: счётчик незавершённых транзакций достиг нуля.

Complete – приём завершён, сигнализация ядру (с помощью запроса на прерывание или статусным битом) о завершении приёма массива данных заданного размера. В этом состоянии не выдаются запросы на запись. Условия перехода в состояние DataMiss: запись в бит Clr значения 1. Условия перехода в состояние Idle: запись в бит Cpl значения 0, И запись в бит Clr значения 0, И запись в бит En значения 0. Условия перехода в состояние ReceiveWrite: запись в бит Cpl значения 0, И запись в бит Clr значения 0, И запись в бит En значения 1.

DataMiss – режим очистки буфера, может быть использован при ошибках и остановках. В этом состоянии не выдаются запросы на запись. Данные из БД читаются по

одному, но не выставляются в канал данных записи. Условие перехода в состояние Complete: основной счётчик достиг нуля.

3.7.2 Передающая часть блока коммуникационного порта

3.7.2.1 Структура передающей части блока коммуникационного порта

Структура передающей части показана ниже (см. рисунок 3.11).

Рисунок 3.11 - Структура передающей части блока коммуникационного порта

Адресный генератор (АГ), работающий в двух режимах: одномерной и двухмерной адресации. Подробнее работа адресных генераторов описана ниже.

Буфер данных (БД), размер которого шесть 64-х разрядных данных. Служит для временного хранения данных.

Блок управления (БУ), который формирует управляющие сигналы для остальных блоков и шин контроллера. В состав блока управления входят автомат состояния, основной счётчик данных, регистр масок прерывания, бит ошибки/остановки, а также дополнительный счётчик управления: счётчик активных запросов.

Конвейерные регистры на шинах данных и адреса используются для улучшения временных характеристик схемы.

3.7.2.2 Принципы работы

Чтение данных происходит по шине AXI3. Данные читаются одиночными запросами. Одиночный запрос в канал адреса выставляется, только если БД может вместить данные от всех не завершённых на шине запросов.

Данные из БД считываются в конвейерный регистр. Через интерфейс AXI4 Stream данные считываются из конвейерного регистра. Интерфейс может быть блокирован коммуникационной шиной в случае её неготовности.

Регистры контроллера, доступные через периферийную шину:

регистр текущего адреса,

основной счётчик данных,

регистр смещения адреса,

счётчик последовательных адресов,

регистр режима адресации,

регистр управления,

регистр масок прерывания,

регистр состояния контроллера.

Текущее состояние контроллера определяется автоматом состояний, битом ошибки/остановки и значением счётчика активных запросов.

Основной счётчик данных работает следующим образом:

переключение -1 при каждой выдаче адреса в канал адреса.

Счётчик активных запросов работает следующим образом:

переключение +1 при каждой выдаче адреса в канал адреса,

переключение -1 при каждом чтении из БД.

Бит ошибки/остановки устанавливается автоматически при получении в канале данных сигнала ошибки чтения, этот бит может быть установлен и снят программно.

Автомат состояний передающей части блока коммуникационного порта показан ниже (см. рисунок 3.12).

Рисунок 3.12 - Автомат состояний передающей части ККП

Idle – состояние бездействия, начальное состояние. В этом состоянии не выдаются запросы на чтение.

ReadSend – чтение и отправка данных, основное рабочее состояние. В этом состоянии разрешены запросы на чтение. Условия перехода из этого состояния в состояние SendOnly: основной счётчик достиг нуля ИЛИ установка (программная или аппаратная) бита ошибки/остановки.

SendOnly – отправка последних данных. В этом состоянии не выдаются запросы на чтение. При установленном бите ошибки/остановки отправка блокируется. Условие перехода в состояние Complete: счётчик активных запросов достиг нуля. Условие перехода в состояние DataMiss: программная запись в бит Clr значения 1.

Complete – отправка завершена, сигнализация ядру (с помощью запроса на прерывание или статусным битом) о завершении отправки массива данных заданного размера. В этом состоянии не выдаются запросы на чтение. Условия перехода в состояние Idle: запись в бит Cpl значения 0, И запись в бит En значения 0. Условия перехода в состояние ReadSend: запись в бит Cpl значения 0, И запись в бит En значения 1.

DataMiss – режим очистки буфера, может быть использован при ошибках и остановках. В этом состоянии не выдаются запросы на чтение. Данные из БД читаются по одному, но не выставляются в интерфейс коммуникационной шины. Условие перехода в состояние Complete: счётчик активных запросов достиг нуля.

3.7.3 Адресный генератор блока коммуникационного порта

Адресный генератор имеется в каждой из двух частей блока коммуникационного порта: приёмной и передающей. Структура адресного генератора показана ниже (см. рисунок 3.13).

Рисунок 3.13 - Структура адресного генератора блока коммуникационного порта

Все регистры адресного генератора доступны для записи и чтения.

Address – регистр текущего адреса, значение этого регистра выставляется в канал адреса при выдаче запроса.

Bias – регистр смещения, значение этого регистра используется для вычисления следующего адреса при двухмерной адресации. Если счётчик RowCounter достиг нуля, то к текущему адресу прибавляется значение регистра Bias.

RowCounter – счётчик последовательных данных, значение этого регистра используется для генерации следующего адреса.

RowCounter_sh - теневой регистр счётчика последовательных данных, хранит записанное значение RowCounter.

БУ формирует сигнал переключения адресного генератора. При переключении адрес из регистра текущего адреса поступает в конвейерный регистр адреса в контроллере, новое значение поступает в регистр текущего адреса из схемы сумматора.

Схема сумматора может прибавлять к текущему адресу смещение и дополнительно единицу.

При двухмерной адресации буфер данных в памяти расположен не непрерывным блоком, а равноотстоящими друг от друга фрагментами (строками равного размера). Внутри строки адреса данных изменяются последовательно, между строками есть расстояние (в адресных единицах) - Bias.

Рисунок 3.14 - Двухмерная адресация в адресном генераторе

Чтобы настроить адресный генератор на адресацию к такому массиву, нужно задать в RowCounter размер строки и задать смещение адреса при переходе от одной строки к другой в регистр Bias.

3.7.4 Программная модель передающей части блока коммуникационного порта

Все регистры передающей части контроллера 32-разрядные, в зарезервированных битах при чтении возвращается 0, запись любого значения в зарезервированные биты никак не влияет на устройство. Набор регистров для каждого коммуникационного порта одинаковый. Адреса отличаются только начальным смещением Base (доступ 32-разрядный):

- Base = 2_1800h для CP0;
- Base = 2_1C00h для CP1;
- Base = 2_2000h для CP2;
- Base = 2_2400h для CP3;

Набор регистров передающей части контроллера коммуникационного порта представлен ниже (см. таблицу 3.6).

Таблица 3.6 - Набор регистров передающей части блока коммуникационного порта

Адрес (32-разрядный доступ)	Название	Доступ	Пояснение
Base+00h	MainCounter	ЧТ/ЗП	Основной счётчик данных
Base+02h	Address	ЧТ/ЗП	Регистр текущего адреса в адресном генераторе
Base+02h	Bias	ЧТ/ЗП	Регистр смещения адреса
Base+06h	RowCounter	ЧТ/ЗП	Счётчик последовательных данных
Base+8h	AddressingMode	ЧТ/ЗП	Регистр режима адресации
Base+0Ah	CSR	ЧТ/ЗП	Регистр управления
Base+0Ch	InterruptMask	ЧТ/ЗП	Регистр масок запросов на прерывание
Base+0Eh	InternalState	ЧТ	Регистр состояния

Примечание: Все адреса и смещения задаются для 32-разрядного доступа (регистры Address и Bias);
Все счётчики считают переданные 64-разрядные слова (счётчики MainCounter и RowCounter).

Регистр текущего адреса Address

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
Address																															

Рисунок 3.16 - Формат регистра Address

Запись в регистр Address (см. рисунок 3.16) устанавливает начальный адрес массива, который следует передать на коммуникационную шину. При чтении возвращается текущий адрес в адресном генераторе.

Основной счётчик данных MainCounter

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																MainCounter															

Рисунок 3.17 - Формат регистра MainCounter

Запись в регистр MainCounter (см. рисунок 3.17) устанавливает количество слов в массиве, который следует предать на коммуникационную шину. При чтении возвращается количество слов, которое осталось прочитать из памяти до завершения отправки.

Регистр смещения адреса Bias

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
Bias																															

Рисунок 3.18 - Формат регистра Bias

Регистр Bias (см. рисунок 3.18) используется только при двухмерной адресации, содержит смещение адреса при переходе от конца одной строки передаваемого массива до начала следующей. Фактически – это размер (в единицах адреса) пропускаемого фрагмента.

Счётчик последовательных данных RowCounter

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																RowCounter															

Рисунок 3.19 - Формат регистра RowCounter

Регистр RowCounter (см. рисунок 3.19) используется только при двухмерной адресации. Запись в регистр RowCounter устанавливает размер строки (в словах) передаваемого массива.

Регистр режима адресации AddressingMode

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																															AM

Рисунок 3.20 - Формат регистра AddressingMode

Регистр AddressingMode (см. рисунок 3.20) определяет режим адресации в адресном генераторе. Бит AM = 0 – одномерная адресация, бит AM = 1 – двухмерная адресация.

Регистр управления CSR

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																												Clr	ES	Cpl	En

Рисунок 3.21 - Формат регистра CSR

Регистр CSR (см. Рисунок 3.21) используется для управления состоянием контроллера:

- Бит En (0-й разряд). Запись в бит En = 1 запускает отправку.
- Бит Cpl (1-й разряд). При чтении Cpl = 1 означает, что текущая отправка завершена. Чтобы произвести следующий запуск, необходимо записать Cpl = 0.
- Бит ES (2-й разряд). При чтении ES = 1 означает, что отправка приостановлена. Бит ES устанавливается аппаратно, если контроллер получил сигнал об ошибке своего обращения в память – по несуществующему или недоступному адресу. Также бит ES можно установить программно. После обработки ошибки бит ES необходимо сбрасывать программно, так как он блокирует отправку.
- Бит Clr (3-й разряд) показывает, что передающая часть контроллера находится в состоянии очистки буфера данных. В этом состоянии контроллер дожидается прихода данных от всех незавершённых запросов в память. Приходящие данные в таком случае теряются. Бит Clr автоматически сбрасывается при достижении ARC = 0.

Регистр масок запросов на прерывание InterruptMask

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																														MIE	MIC

Рисунок 3.22 - Формат регистра InterruptMask

Данный регистр InterruptMask (см. Рисунок 3.22) позволяет запретить выдачу запроса на прерывание от передающей части блока коммуникационного порта по каждой из двух причин - завершение отправки и ошибка при обращении в память:

- Бит MIC (0-й разряд) – маска запроса на прерывание по завершению отправки. MIC = 1 – запрос маскирован, MIC = 0 – запрос разрешён.

- Бит MIE (1-й разряд) – маска запроса на прерывание по ошибке или программной остановке. MIE = 1 – запрос маскирован, MIE = 0 – запрос разрешён.

Регистр состояния InternalState

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
Зарезерви-ровано	FSM	Зарезервировано	ARC

Рисунок 3.23 - Формат регистра InternalState

Регистр состояния InternalState (см. рисунок 3.23) доступен только для чтения, показывает внутреннее состояние передающей части контроллера. Может быть использован для отладки и для обработки ошибочных ситуаций. Регистр содержит следующие поля:

- Поле ARC (разряды 4-0) – счётчик активных запросов. Показывает количество данных, запрос на чтение которых уже отдан, но данные ещё не отправлены на коммуникационную шину. При приостановке передающей части сумма MainCounter + ARC показывает реальное количество "недоотправленных" на коммуникационную шину данных.
- Поле FSM (разряды 28-24) – показывает текущее состояние конечного автомата передающей части:

  01h – Idle – бездействие,

  02h – ReadSend – чтение из памяти и отправка,

  04h – SendOnly – чтение завершено, только отправка,

  08h – Complete – отправка завершена,

  1Ch – DataMiss – режим очистки буфера данных.

3.7.5 Программная модель приёмной части блока коммуникационного порта

Все регистры приёмной части контроллера 32-разрядные, в зарезервированных битах при чтении возвращается 0, запись любого значения в зарезервированные биты никак не влияет на устройство. Набор регистров приемной части блока коммуникационного порта представлен ниже (см. таблицу 3.7).

Таблица 3.7 - Набор регистров приемной части блока коммуникационного порта

Адрес (32-разрядный доступ)	Название	Доступ	Пояснение
Base+10h	MainCounter	ЧТ/ЗП	Основной счётчик данных
Base+12h	Address	ЧТ/ЗП	Регистр текущего адреса в адресном генераторе
Base+14h	Bias	ЧТ/ЗП	Регистр смещения адреса
Base+16h	RowCounter	ЧТ/ЗП	Счётчик последовательных данных
Base+18h	AddressingMode	ЧТ/ЗП	Регистр режима адресации
Base+1Ah	CSR	ЧТ/ЗП	Регистр управления
Base+1Ch	InterruptMask	ЧТ/ЗП	Регистр масок запросов на прерывание
Base+1Eh	InternalState	ЧТ	Регистр состояния

Примечание: Все адреса и смещения задаются для 32-разрядного доступа (регистры Address и Bias);
Все счётчики считают принятые 64-разрядные слова (счётчики MainCounter и RowCounter).

Регистр текущего адреса Address

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
Address																															

Рисунок 3.24 - Формат регистра Address

Запись в регистр Address (см. рисунок 3.24) устанавливает начальный адрес буфера в памяти, в который следует записывать принимаемые с коммуникационной шины данные. При чтении возвращается текущий адрес в адресном генераторе.

Основной счётчик данных MainCounter

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																MainCounter															

Рисунок 3.25 - Формат регистра MainCounter

Запись в регистр MainCounter (см. рисунок 3.25) устанавливает количество слов, которое будет получено с коммуникационной шины. При чтении возвращается количество слов, которое осталось записать в память в текущей передаче.

Регистр смещения адреса Bias

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
Bias																															

Рисунок 3.26 - Формат регистра Bias

Регистр Bias (см. рисунок 3.26) используется только при двухмерной адресации, содержит смещение адреса при переходе от конца одной строки передаваемого массива до начала следующей. Фактически – это размер (в единицах адреса) пропускаемого фрагмента.

Счётчик последовательных данных RowCounter

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																RowCounter															

Рисунок 3.27 - Формат регистра RowCounter

Регистр RowCounter (см. рисунок 3.27) используется только при двухмерной адресации. Запись в регистр RowCounter устанавливает размер строки (в словах) передаваемого массива.

Регистр режима адресации AddressingMode

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																															AM

Рисунок 3.28 - Формат регистра AddressingMode

Регистр AddressingMode (см. рисунок 3.28) определяет режим адресации в адресном генераторе. Бит AM = 0 – одномерная адресация, бит AM = 1 – двухмерная адресация.

Регистр управления CSR

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																												Clr	ES	Cpl	En

Рисунок 3.29 - Формат регистра CSR

Регистр CSR (см. рисунок 3.29) используется для управления состоянием блока коммуникационного порта:

- Бит En (0-й разряд). Запись в бит En = 1 запускает приём.
- Бит Cpl (1-й разряд). При чтении Cpl = 1 означает, что приём и запись заданного количества слов завершены, либо запись приостановлена вследствие ошибки или программной приостановки (бит ES = 1). Чтобы произвести следующий запуск, необходимо записать Cpl = 0.
- Бит ES (2-й разряд). При чтении ES = 1 означает, что запись в память приостановлена. Бит ES устанавливается аппаратно, если блока коммуникационного порта получил сигнал об ошибке своего обращения в память – по несуществующему или недоступному адресу. Также бит ES можно установить программно. После обработки ошибки бит ES необходимо сбрасывать программно, так как он блокирует запись.
- Бит Clr (3-й разряд) показывает, что приёмная часть блока коммуникационного порта находится в состоянии очистки буфера данных. В этом состоянии принимаемые данные теряются – осуществляется только их подсчёт в регистре основного счётчика данных (MainCounter). Бит Clr сбрасывается автоматически при достижении MainCounter = 0.

Регистр масок запросов на прерывание InterruptMask

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																														MIE	MIC

Рисунок 3.30 - Формат регистра InterruptMask

Данный регистр (см. рисунок 3.30) позволяет запретить выдачу запроса на прерывание от приёмной части блока коммуникационного порта по каждой из двух причин - завершение приёма и ошибка при обращении в память:

- Бит MIC (0-й разряд) – маска запроса на прерывание по завершению приёма. MIC = 1 – запрос маскирован, MIC = 0 – запрос разрешён.
- Бит MIE (1-й разряд) – маска запроса на прерывание по ошибке или программной остановке. MIE = 1 – запрос маскирован, MIE = 0 – запрос разрешён.

Регистр состояния InternalState

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	FSM	Зарезервировано	UTC	Зарезервировано	ADC

Рисунок 3.31 - Формат регистра InternalState

Регистр состояния (см. Рисунок 3.31) доступен только для чтения, показывает внутреннее состояние приёмной части блока коммуникационного порта. Может быть использован для отладки и для обработки ошибочных ситуаций. Регистр имеет следующие поля:

- Поле ADC (разряды 4-0) – счётчик доступных данных. Показывает количество данных, имеющихся в приёмном буфере блока коммуникационного порта, но ещё не выданных на запись в память.
- Поле UTC (разряды 20-16) – счётчик незавершённых транзакций. Показывает количество транзакций записи на шине AXI3, для которых ещё не получено подтверждения.
- Поле FSM (разряды 28-24) – показывает текущее состояние конечного автомата передающей части:

  01h – Idle – бездействие,

  02h – ReceiveWrite – приём и запись данных,

  04h – UncompleteWrite – приём завершён, только запись,

  08h – Complete – запись завершена,

  1Ch – DataMiss – режим очистки буфера данных.

3.7.6 Прерывания блока коммуникационного порта

Блок коммуникационного порта может выдавать по 2 запроса на прерывание от каждого коммуникационного порта: от передающей и приёмной части. Эти прерывания вызываются либо при завершении обмена данными, либо при возникновении ошибок при обмене. Описание причин прерываний приведено в таблице 3.8.

Таблица 3.8 - Виды запросов прерывания

Причина прерывания	Бит, вызывающий запрос	Бит, маскирующий запрос
Завершение отправки заданного массива данных	Бит Cpl регистра CSR передающей части	Бит MIC регистра InterruptMask передающей части
Ошибка чтения из памяти при отправке	Бит ES регистра CSR передающей части	Бит MIE регистра InterruptMask передающей части
Завершение приёма заданного количества данных	Бит Cpl регистра CSR приёмной части	Бит MIC регистра InterruptMask приёмной части
Ошибка записи в память при приёме	Бит ES регистра CSR приёмной части	Бит MIE регистра InterruptMask приёмной части

3.8 Мост "системный интегратор - AXI"

Мост "системный интегратор - AXI" обеспечивает процессорному ядру NMC доступ к внешней памяти и периферийным устройствам системы на кристалле.

Рисунок 3.32 – Структурная схема моста "системный интегратор - AXI"

Элементы архитектуры моста:

буфер адресов записи,

буфер адресов чтения,

буфер типа операции записи - хранит признак: 64- или 32-разрядные данные, а также младший бит адреса,

блок управления – обрабатывает сигналы управления всей схемы, отслеживает порядок запросов,

буфер данных записи,

буфер данных чтения,

буфер сравнения для адресов чтения – сравнивает текущий адрес записи со всеми адресами чтения ещё не завершённых транзакций,

буфер сравнения для адресов записи - сравнивает текущий адрес чтения со всеми адресами записи ещё не завершённых транзакций,

блок обработки подтверждений – содержит программно доступные регистры.

Функции и особенности моста:

- Мост преобразует транзакции записи и чтения ядра NMC в транзакции записи и чтения шины AXI.
- Мост отслеживает порядок транзакций чтения и записи, и в случае конфликта по адресам двух транзакций в разных каналах (между чтением и записью), выдача более поздней транзакции блокируется до прихода подтверждения первой транзакции. Это обеспечивает исходный порядок (в котором получены адреса от NMC4) операций доступа в память.
- Обращения в периферийную область (0002_0000h-0003_FFFFh в 32-разрядной адресации) обрабатываются особым образом - исходный порядок обеспечивается для всех транзакций, попадающих в периферийную область.
- Мост различает запросы чтения данных и инструкций процессорного ядра. На шине AXI мост использует разные идентификаторы (ARID и RID): 0 – чтение данных, 1 – выборка инструкций. Адреса инструкций не проходят через буфер сравнения и с ними не сравнивается текущий адрес записи, то есть, если одновременно по одному адресу выполняется транзакция записи и транзакция выборки инструкции, то порядок их исполнения не гарантируется. При чтении данных и выборке инструкций поддерживается интерливинг данных чтения (спецификация AXI, п. 8.3 Read Ordering) неограниченной глубины: данные передаются к системному интегратору в том порядке, в котором получены с шины AXI (системный интегратор различает их по идентификатору, который передаётся вместе с данными).
- Мост корректно обрабатывает транзакции, которые возвращают ошибку в фазе подтверждения (сигналы RRESP, BRESP). Мост сохраняет адрес первой транзакции, закончившейся с ошибкой, и генерирует запрос на прерывание. Адрес ошибочной транзакции и функция управления прерыванием доступны для ядра NMC через периферийный интерфейс. Исключение составляет выборка инструкций – для неё не сохраняется адрес ошибочной транзакции.
- Мост использует только один тип транзакции AXI - одиночная транзакция типа INCR, при этом запись и чтение могут производиться 32- или 64-разрядными словами (все адреса, соответственно, выровнены до четырёх байт).
- Мост не использует идентификаторы AXI (в системе ему назначается постоянный идентификатор).

• Имеются 2 режима записи: режим запаздывающих данных и режим выровненных данных. В режиме выровненных данных адрес выдаётся на шину AXI только при наличии данных для записи. В режиме запаздывающих данных адрес выдаётся до появления данных, в таком случае время, на которое запаздывают данные, зависит от состояния конвейера ядра NMC4 и может достигать десятков тактов.

3.8.1 Особенности работы моста

Основная функция моста - распределение потока обращений в память на 2 независимых конвейера шины AXI: конвейер чтения и конвейер записи. Порядок выполнения операций в разных конвейерах не гарантируется, поэтому в случаях, когда порядок требуется соблюдать, мост обеспечивает порядок путём задержки более позднего запроса.

Порядок, в котором выполняются 2 операции, требуется соблюдать в следующих случаях:

запись в память, затем чтение по тому же адресу - чтение дожидается завершения записи, чтобы вернуть обновлённые данные,

чтение из памяти, затем запись по тому же адресу - запись дожидается завершения чтения, чтобы данные чтения не были обновлены,

запись в область периферийных устройств, затем чтение из области периферийных устройств,

чтение из области периферийных устройств, затем запись в область периферийных устройств - независимо от адреса при обращении к периферийным устройствам требуется соблюдать порядок операций чтения и записи.

Транзакция чтения AXI производится в 2 фазы: фаза адреса и фаза данных (в фазе данных приходит также сигнал подтверждения). В первой фазе адрес из буфера адресов выдаётся на шину AXI и одновременно записывается в буфер сравнения для адресов чтения. Мост поддерживает выдачу нескольких адресов чтения до получения данных, поэтому адреса чтения могут выдаваться до тех пор, пока не заполнится буфер сравнения. Адрес покидает буфер сравнения в момент прихода данных (фаза данных). Мост не поддерживает пакетные передачи, поэтому одному адресу всегда соответствует одно данное.

Транзакция записи AXI производится в 3 фазы: фаза адреса, фаза данных и фаза подтверждения. Мост имеет 2 режима записи: отличие их состоит в том, как соотносятся фаза адреса и фаза данных. В режиме выровненных данных адрес записи не выставляется на шину AXI и остаётся в буфере до прихода соответствующих данных записи. В режиме

запаздывающих данных адрес выдаётся на шину AXI независимо от того, пришли на данный момент данные или нет.

Преимущество режима запаздывающих данных:

меньше блокировок ядра - например, при выполнении трёх операций записи, а затем чтения, если первая запись блокируется, то блокируется и чтение, так как не может обойти третью запись и попасть в буфер адресов чтения.

Преимущество режима выровненных данных:

меньше блокировок в конвейере записи в память - например, когда контроллер памяти принимает адреса записи от одного NMCell, он вынужден дожидаться соответствующих данных и не давать доступа другим устройствам системы: если данные приходят вместе с адресом,

Выдаваемый адрес записи поступает в буфер сравнения для адресов записи. Адрес покидает буфер сравнения в момент прихода подтверждения.

3.8.2 Эксклюзивные операции

В системе команд NMPU нет выделенных команд для эксклюзивного доступа. Мост «системный интегратор – AXI» преобразует обычные операции чтения и записи NMPU в эксклюзивные операции (exclusive access) шины AXI. Для разграничения разных типов доступа используются старшие не используемые на шине AXI биты адреса ядра NMC4:

- бит 31 – признак эксклюзивной транзакции,
- бит 30 – признак фазы статуса.

Соответственно, возможны следующие типы операций чтения/записи, представленные в таблице ниже.

Таблица 3.9 - Типы операций эксклюзивного доступа

Бит 31	Бит 30	ЧТ/ЗП	Описание
0	0	ЧТ	обычное чтение из памяти или периферии
0	0	ЗП	обычная запись в память или периферию
0	1	ЧТ	чтение статуса последней эксклюзивной записи (фактически производится обычное чтение, но данные подменяются кодом статуса)
0	1	ЗП	зарезервированная комбинация (фактически производится обычная запись)
1	0	ЧТ	эксклюзивное чтение
1	0	ЗП	фаза данных эксклюзивной записи
1	1	ЧТ	зарезервированная комбинация (фактически производится эксклюзивное чтение, но данные подменяются кодом статуса)
1	1	ЗП	зарезервированная комбинация (фактически производится эксклюзивная запись)

Никакие транзакции не завершаются самим мостом - происходит просто преобразование адреса путём откидывания разрядов 30 и 31. Эксклюзивные транзакции проводятся на шине AXI с признаком ARLOCK/AWLOCK=b01. Чтение статуса производится как обычная операция чтения (ARLOCK=b00), но адрес следует выбирать так, чтобы данное чтение не изменяло состояния каких-либо устройств. Операция чтения статуса отличается от обычного чтения:

- мост отбрасывает код подтверждения RRESP и данные чтения,
- вместо данных ядру NMC4 отправляется код статуса:

  0 – последняя эксклюзивная запись прошла успешно,

  1 – не успешно.

3.8.3 Ошибки обращения

Каждая транзакция AXI завершается подтверждением, по которому можно определить, нормально завершилась транзакция или с ошибкой. Возможны ошибки следующих видов:

- ошибка адресации – возникает, когда область памяти, в которую направлено обращение, зарезервирована,
- ошибка конечного устройства – возникает, когда slave-устройство, запретило обращение к данному адресу в данный момент,
- ошибка эксклюзивной операции – возникает при попытке эксклюзивного доступа, в slave-устройство, не поддерживающее эксклюзивных операций.

Если в фазе подтверждения транзакции приходит сигнал ошибки, то код ошибки и адрес (читается из буфера сравнения) сохраняются в программно доступных регистрах моста.

Исключение составляет выборка инструкций – для неё не сохраняется адрес ошибочной транзакции.

При возникновении ошибки мост может устанавливать сигнал запроса на прерывание.

3.8.4 Программно доступные регистры моста "системный интегратор - AXI"

Программно доступные регистры моста используются для обработки ошибок обращения в память и для настройки режима записи.

Таблица 3.10 – Перечень регистров моста "системный интегратор - AXI"

Адрес (32-разрядный доступ)	Название	Доступ	Описание
0002_1000h	CSR	ЧТ/ЗП	Регистр управления
0002_1002h	IRQMask	ЧТ/ЗП	Маска прерывания
0002_1004h	RdAddr	ЧТ	Регистр адреса чтения
0002_1006h	WrAddr	ЧТ	Регистр адреса записи

Регистр управления (CSR)

Таблица 3.10 – Описание полей регистра CSR

Разряды	Название	Доступ	Описание
[13]	AWR	ЧТ	Активность канала записи: 0 - мост не содержит команд записи и нет незавершённых транзакций записи, 1 - имеется хотя бы одна незавершённая команда записи. Данный бит не учитывает признак ошибки.
[12]	ARD	ЧТ	Активность канала чтения: 0 - мост не содержит команд чтения и нет незавершённых транзакций чтения, 1 - имеется хотя бы одна незавершённая команда чтения. Данный бит не учитывает признак ошибки.
[11:10]	RRI	ЧТ	Код подтверждения ошибочной операции при выборке команд (RRESP шины AXI). 00 - нормальное завершение (поле имеет значение 00, пока бит ERI не установлен), 01 - зарезервировано, 10 - конечное устройство вернуло ошибку, 11 - обращение в зарезервированную область памяти.
[9:8]	RWR	ЧТ	Код подтверждения ошибочной операции при записи данных (BRESP шины AXI). 00 - нормальное завершение (поле имеет значение 00, пока бит EWR не установлен), 01 - ошибка эксклюзивной записи, 10 - конечное устройство вернуло ошибку, 11 - обращение в зарезервированную область памяти.
[7:6]	RRD	ЧТ	Код подтверждения ошибочной операции при чтении данных (RRESP шины AXI). 00 - нормальное завершение (поле имеет значение 00, пока бит ERD не установлен), 01 – ошибка эксклюзивного чтения, 10 - конечное устройство вернуло ошибку, 11 - обращение в зарезервированную область памяти.
[5]	MCLR	ЧТ/ЗП	Установка режима записи 0. Запись значения 1 в этот бит устанавливает режим записи 0.
[4]	MSET	ЧТ/ЗП	Установка режима записи 1. Запись значения 1 в этот бит устанавливает режим записи 1.
[3]	M	ЧТ	Текущий режим записи: 0 – режим запаздывающих данных, 1 – режим выровненных данных.
[2]	ERI	ЧТ/ЗП	Признак ошибки адресации в запросе на выборку команд. Устанавливается аппаратно в момент прихода подтверждения RRESP на шине AXI. Сброс производится программно путём записи значения 1 в данный бит.
[1]	EWR	ЧТ/ЗП	Признак ошибки адресации в запросе записи. Устанавливается аппаратно в момент прихода подтверждения BRESP на шине AXI. Сброс производится программно путём записи значения 1 в данный бит.
[0]	ERD	ЧТ/ЗП	Признак ошибки адресации в запросе чтения данных. Устанавливается аппаратно в момент прихода подтверждения RRESP на шине AXI. Сброс производится программно путём записи значения 1 в данный бит.

Регистр всегда доступен для чтения, и записи значения 1 в некоторые биты (ERD, EWR, ERI, MSET, MCLR).

Бит M имеет теневой регистр: запись значения 1 в биты MSET и MCLR изменяет только теневой регистр. Перезапись в рабочий регистр осуществляется автоматически, но только во время простоя канала записи (когда бит AWR имеет значение 0). Чтобы изменить значение бита M, необходимо провести запись значения 1 в бит MSET либо MCLR и дождаться, пока в бите M установится требуемое значение.

Признак ошибки (ERD, ERI и EWR) устанавливается аппаратно, если возвращаемый код подтверждения транзакции AXI (RRESP и BRESP, соответственно) свидетельствует об ошибочном завершении транзакции.

Поле кода подтверждения (RRD, RRI и RWR) обновляется с завершением каждой транзакции, пока не установлен соответствующий бит ошибки (ERD, ERI и EWR). Таким образом, в нём остаётся код первой произошедшей ошибки.

Регистр маски прерываний (IRQMask)

Регистр маски прерываний разрешает или запрещает выдачу запроса на прерывание к NMPU.

Таблица 3.11- Описание полей регистра IRQMask

Разряды	Название	Доступ	Описание
[2]	IRI	ЧТ/ЗП	Включение прерывания при ошибках выборки инструкций: 0 - прерывание запрещено, 1 - прерывание разрешено.
[1]	IWR	ЧТ/ЗП	Включение прерывания при ошибках записи: 0 - прерывание запрещено, 1 - прерывание разрешено.
[0]	IRD	ЧТ/ЗП	Включение прерывания при ошибках чтения данных: 0 - прерывание запрещено, 1 - прерывание разрешено.

Регистр адреса чтения и регистр адреса записи (RdAddr и WrAddr)

Регистр адреса (RdAddr и WrAddr) обновляется с завершением каждой транзакции, пока не установлен бит ошибки (ERD и EWR). Таким образом, при возникновении ошибки в нём остаётся адрес первой транзакции, завершившейся с ошибкой. Для выборки инструкций регистр адреса не доступен.

Таблица 3.12 - Описание полей регистра RdAddr и WrAddr

Разряды	Название	Доступ	Описание
[31:30]	-	-	Зарезервировано
[29:0]	ADDR	ЧТ	Адрес завершившейся операции.

3.8.5 Прерывания

Запрос на прерывание от моста установлен всегда, когда установлены биты ERD и IRD, ERI и IRI или EWR и IWR. Запрос снимается программно путём сброса битов ERD, ERI или EWR.

3.9 Мост "системный интегратор - периферийная шина NMPU"

Мост "системный интегратор - периферийная шина NMPU" обеспечивает ядру NMC4 доступ к собственным периферийным устройствам процессорной системы NMPU:

- системный контроллер,
- контроллер внешних прерываний,
- блок таймеров,
- блок предвыборки команд IFU,
- мост «системный интегратор – AXI»,
- блок защиты памяти MPU,
- коммуникационные порты CP0-CP3.

Рисунок 3.33 - Мост "системный интегратор - периферийная шина NMPU"

Мост состоит из буфера адресов, буфера данных чтения, регистра данных записи и схемы управления. Транзакция на периферийной шине занимает всегда 2 такта и не конвейеризована.

Доступ на периферийную шину допускается только скалярными командами загрузки и выгрузки регистров NMPU. При обращении от него к адресам периферийной шины векторными командами поведение системы не специфицировано.

Все регистры на периферийной шине NMPU имеют чётные адреса, а все нечётные адреса зарезервированы. Это означат, что:

- запись по ним никак не влияет устройства системы,
- чтение возвращает значение 0,

- при записи регистровыми парами значение старшего 32-разрядного слова (РОН) отбрасывается, значение младшего (адресный регистр) записывается,
- при чтении регистровыми парами в старшем 32-разрядном слове (РОН) возвращается 0, в младшем (адресный регистр) - возвращается прочитанное значение.

Мост не имеет программно доступных регистров.

3.10 Блок периферийных устройств (PU) процессорного узла NMPU

3.10.1 Общие сведения

Блок периферийных устройств процессора хранит идентификатор своего NMPU, служит для фиксации внешних событий и прерываний, обмена прерываниями между NMPU и управляющим ПУ внутри кластера, а также подсчёта интервалов времени. NMPU управляет этим блоком с помощью реализации программного доступа к регистрам периферийных устройств как на запись, так и на чтение. Причём этот доступ осуществляется по адресам от 0x0002_0000hex до 0x0003_FFFF hex (область периферийных регистров) как к ячейкам памяти. Все периферийные устройства выполнены в виде IP-блоков и имеют стандартные интерфейсы в соответствии со спецификацией шины AMBA APB спецификации 2.0 компании ARM Limited. В состав периферийных устройств NMPU входят следующие блоки:

- системный контроллер;
- контроллер внешних прерываний;
- блок таймеров.

3.10.2 Системный контроллер NMPU

Системный контроллер выполняет следующие функции:

- содержит идентификатор своего NMPU;
- служит для генерации запросов на прерывание к другим ПУ.

Структурная схема системного контроллера представлена ниже (Рисунок 3.34).

Рисунок 3.34 - Системный контроллер NMPU

3.10.2.1 Программно доступные регистры системного контроллера NMPU

Таблица 3.13 – Перечень программно доступных регистров системного контроллера NMPU

Адрес (32-разрядный доступ)	Название	Доступ	Описание
2_0000h	ID	ЧТ	Регистр идентификации процессорной системы.
2_0002h	SYNC	ЧТ/ЗП	Регистр межпроцессорной синхронизации

Регистр идентификации процессорной системы (ID)

Регистр идентификации процессорной системы доступен только для чтения. Регистр содержит 32-разрядный идентификатор NMPU.

Таблица 3.14 – Описание полей регистра ID

Разряды	Название	Доступ	Описание
[31:24]	INDEX	ЧТ	Номер NMPU в составе СнК: 00000000 – NMPU0 кластера 0; 00000001 – NMPU1 кластера 0; 00000010 – NMPU2 кластера 0; 00000011 – NMPU3 кластера 0; 00000100 – NMPU0 кластера 1; 00000101 – NMPU1 кластера 1; 00000110 – NMPU2 кластера 1; 00000111 – NMPU3 кластера 1; 00001000 – NMPU0 кластера 2; 00001001 – NMPU1 кластера 2; 00001010 – NMPU2 кластера 2; 00001011 – NMPU3 кластера 2; 00001100 – NMPU0 кластера 3; 00001101 – NMPU1 кластера 3; 00001110 – NMPU2 кластера 3; 00001111 – NMPU3 кластера 3;
[23:20]	VAR	ЧТ	Особенности архитектуры процессорного ядра: 0010 - ядро NMC4 с сопроцессором арифметики с плавающей точкой,
[19:16]	ARCH	ЧТ	Архитектура процессорного ядра: 0100 - NMC4.
[15:4]	PRTNUM	ЧТ	Идентификатор изделия: 0x0B8 - процессор NM6408.
[3:0]	REV	ЧТ	Версия изделия: 0000.

Регистр межпроцессорной синхронизации (SYNC)

Регистр межпроцессорной синхронизации позволяет NMPU и управляющему ПУ внутри кластера обмениваться сигналами запроса и подтверждения. Поле GENREQ отвечает

за выдачу запросов другим процессорным системам NMPU и управляющему ПУ, поле RCVREQ используется для приёма запросов и выдачи подтверждений.

Таблица 3.15 - Описание полей регистра SYNC

Разряды	Название	Доступ	Описание
31-24	-	-	Зарезервировано
23-16	GENREQ	ЧТ/ЗП	Один бит поля GENREQ соответствует одному биту поля RCVREQ другого ПУ. Запись значения 1 в бит поля GENREQ вызывает аппаратную установку соответствующего бита RCVREQ другого ПУ. Программный сброс бита RCVREQ другого ПУ системы сбрасывает аппаратно бит поля GENREQ.
15-8	-	-	Зарезервировано
7-0	RCVREQ	ЧТ/ЗП	Один бит поля RCVREQ соответствует одному биту GENREQ другого ПУ. Программная установка соответствующего бита GENREQ в другой процессорной системе устанавливает бит RCVREQ. Запись значения 1 в бит поля RCVREQ сбрасывает этот бит и инициирует аппаратный сброс соответствующего бита GENREQ. Установка бита поля RCVREQ вызывает запрос на прерывание.

Таблица 3.16 - Соответствие разрядов полей регистра SYNC и запросов на прерывание

Разряд регистра SYNC запросчика	Разряд регистра SYNC приёмника	Запрос на прерывание в контроллере внешних прерываний приёмника
23 (GENREQ[7])	7 (RCVREQ[7])	Низкоприоритетное межпроцессорное прерывание 7 от соседнего NMPUk своего кластера
22 (GENREQ[6])	6 (RCVREQ[6])	Низкоприоритетное межпроцессорное прерывание 6 от соседнего NMPUj своего кластера
21 (GENREQ[5])	5 (RCVREQ[5])	Низкоприоритетное межпроцессорное прерывание 5 от соседнего NMPUi своего кластера
20 (GENREQ[4])	4 (RCVREQ[4])	Низкоприоритетное межпроцессорное прерывание 4 от управляющего ПУ своего кластера
19 (GENREQ[3])	3 (RCVREQ[3])	Высокоприоритетное межпроцессорное прерывание 3 от соседнего NMPUk своего кластера
18 (GENREQ[2])	2 (RCVREQ[2])	Высокоприоритетное межпроцессорное прерывание 2 от соседнего NMPUj своего кластера
17 (GENREQ[1])	1 (RCVREQ[1])	Высокоприоритетное межпроцессорное прерывание 1 от соседнего NMPUi своего кластера
16 (GENREQ[0])	0 (RCVREQ[0])	Высокоприоритетное межпроцессорное прерывание 0 от управляющего ПУ своего кластера

Примечания: Для NMPU0: NMPUi = NMPU1, NMPUj = NMPU2, NMPUk = NMPU3;
для NMPU1: NMPUi = NMPU0, NMPUj = NMPU3, NMPUk = NMPU2;
для NMPU2: NMPUi = NMPU3, NMPUj = NMPU0, NMPUk = NMPU1;
для NMPU3: NMPUi = NMPU2, NMPUj = NMPU1, NMPUk = NMPU0.

Запросы на прерывание не могут быть маскированы в системном контроллере. Маскирование может быть произведено только в контроллере внешних прерываний.

3.10.3 Контроллер внешних прерываний NMPU

Контроллер внешних прерываний процессорной системы NMPU обрабатывает сигналы запросов на прерывание, направленные от устройств системы (как расположенных внутри NMPU, так и внешних) к процессорному ядру NMC4. Функции контроллера:

- детектирование запросов,
- маскирование неиспользуемых запросов,
- арбитраж активных запросов,
- формирование адреса-вектора перехода,
- подтверждение или разрешение следующих запросов после обработанного.

Контроллер имеет 28 входных сигналов запросов. Поступающий запрос фиксируется в регистре запросов IRR. Далее запрос может быть маскирован регистром IMR. Схема приоритетов (Priority Unit) выбирает запрос с наибольшим приоритетом (схема описана ниже), формирует запрос на прерывание для ядра NMC4 и соответствующий ему адрес-вектор прерывания.

Рисунок 3.35 - Контроллер прерываний (INTC)

Если сигнал INTA имеет активный уровень, то считается, что процессорное ядро NMC4 зафиксировало прерывание. При этом бит в регистре IRR сбрасывается, а также, если данный запрос работает в режиме программного подтверждения (см. таблицу 3.17), аппаратно устанавливается соответствующий бит в регистре IAS. Установка некоторого разряда IAS блокирует обработку новых прерываний, поступающих по той же линии IRQ. Сброс разрядов IAS выполняется программно с помощью команды записи процессорного ядра NMC4.

Если сигнал INTA имеет неактивный уровень, то запрос на прерывание INT будет стоять до прихода сигнала INTA. При этом адрес-вектор INTV на выходе блока INTC не фиксируется, т.е. если на вход IRQx поступит более приоритетное прерывание, то адрес-вектор изменится и будет соответствовать более приоритетному прерыванию.

3.10.3.1 Внешние прерывания процессорной системы NMPU

В таблице ниже (см. таблицу 3.17) приведён список всех прерываний процессорной системы NMPU. Выделены прерывания от блоков, входящих в состав NMPU.

Таблица 3.17 – Перечень всех внешних прерываний NMPU

№	Поз. бита	Описание	Режим подтв.	Адрес-вектор
0	0	Прерывание 0 от сопроцессора арифметики с плавающей точкой (некорректные данные)	Аппаратный	020 hex
1	1	Прерывание 1 от сопроцессора арифметики с плавающей точкой (overflow)	Аппаратный	028 hex
2	2	Прерывание 2 от сопроцессора арифметики с плавающей точкой (underflow)	Аппаратный	030 hex
3	3	Прерывание 3 от сопроцессора арифметики с плавающей точкой (потеря точности)	Аппаратный	038 hex
4	4	Прерывание 4 от сопроцессора арифметики с плавающей точкой (потеря данных)	Аппаратный	040 hex
5	5	Прерывание 5 от сопроцессора арифметики с плавающей точкой (неправильная команда)	Аппаратный	048 hex
6	6	Прерывание от системного интегратора (выборка команды из периферийной области памяти)	Аппаратный	050 hex
7	7	Прерывание от моста «системный интегратор – AXI» (ошибка обращения во внешнюю память)	Программный	058 hex
8	8	Прерывание от MPU по защите памяти на запись	Програм-мный	060 hex
9	9	Прерывание от MPU защиты памяти на чтение	Програм-мный	068 hex
10	10	Прерывание от блока таймеров (таймер 0)	Аппаратный	070 hex
11	11	Прерывание от блока таймеров (таймер 1)	Аппаратный	078 hex
12	12	Межпроцессорное прерывание 0 от системного контроллера	Аппаратный	080 hex
13	13	Межпроцессорное прерывание 1 от системного контроллера	Аппаратный	088 hex
14	14	Межпроцессорное прерывание 2 от системного контроллера	Аппаратный	090 hex
15	15	Межпроцессорное прерывание 3 от системного контроллера	Аппаратный	098 hex
16	16	Прерывание от коммуникационного порта CP0 (передающий канал)	Аппаратный	0A0 hex
17	17	Прерывание от коммуникационного порта CP0 (принимающий канал)	Аппаратный	0A8 hex
18	18	Прерывание от коммуникационного порта CP1 (передающий канал)	Аппаратный	0B0 hex
19	19	Прерывание от коммуникационного порта CP1 (принимающий канал)	Аппаратный	0B8 hex
20	20	Прерывание от коммуникационного порта CP2 (передающий канал)	Аппаратный	0C0 hex
21	21	Прерывание от коммуникационного порта CP2 (принимающий канал)	Аппаратный	0C8 hex
22	22	Прерывание от коммуникационного порта CP3 (передающий канал)	Аппаратный	0D0 hex
23	23	Прерывание от коммуникационного порта CP3 (принимающий канал)	Аппаратный	0D8 hex
24	24	Межпроцессорное прерывание 4 от системного контроллера	Аппаратный	0E0 hex
25	25	Межпроцессорное прерывание 5 от системного контроллера	Аппаратный	0E8 hex
26	26	Межпроцессорное прерывание 6 от системного контроллера	Аппаратный	0F0 hex
27	27	Межпроцессорное прерывание 7 от системного контроллера	Аппаратный	0F8 hex

3.10.3.2 Режимы подтверждения запросов

Контроллер прерываний процессорной системы NMPU поддерживает 2 режима обработки запросов на прерывание от устройств системы:

- режим аппаратного подтверждения,
- режим программного подтверждения.

Режим для каждого из устройств выбран при разработке и не может быть изменён.

В режиме аппаратного подтверждения контроллер фиксирует запрос в регистре IRR в момент положительного фронта входного сигнала запроса. После того как ядро принимает запрос, контроллер готов принять следующий запрос (следующий положительный фронт сигнала запроса) по этому же сигналу.

В режиме программного подтверждения контроллер фиксирует запрос, когда обнаруживает высокий уровень входного сигнала запроса и если не установлен соответствующий бит в регистре статуса IAS. В момент, когда ядро NMC4 принимает адрес-вектор прерывания, бит регистра IRR сбрасывается, а бит регистра IAS устанавливается. Снятие бита в регистре IAS в этом режиме производится программно – команда снятия бита регистра IAS подтверждает обработку текущего запроса и разрешает генерацию следующего.

При работе с устройствами, которые работают в режиме программного подтверждения следует обеспечить, чтобы команда снятия запроса в запрашивающем устройстве выполнялась строго раньше, чем команда снятия бита статуса прерывания (IAS) в контроллере прерываний. Рекомендуется следующий алгоритм программного подтверждения:

- Запись в регистр снятия запроса в запрашивающем устройстве (в этот момент устройство снимет запрос),
- Чтение из любого регистра запрашивающего устройства (чтение выполнится строго после записи),
- Арифметическая или логическая операция с прочитанным значением (выполнится строго после чтения),
- Запись результата операции в любой зарезервированный регистр контроллера прерываний (выполнится строго после операции в АЛУ),
- Запись значения 1 в активный бит регистра IAS (эта запись выполнится последней, данный бит сбросится и контроллер будет готов принять следующий запрос от соответствующего устройства).

3.10.3.3 Приоритеты запросов

Каждому входному сигналу запроса можно присвоить уровень приоритета: высокий или низкий. Арбитраж производится сначала между всеми запросами с высоким приоритетом, затем, если нет запросов с высоким приоритетом, между всеми запросами с низким. Для сигналов с одинаковым уровнем приоритета имеет значение порядковый номер сигнала: наибольший приоритет имеет 0-й запрос, следующий приоритет у 1-го запроса и так далее до 27-го запроса, который имеет наименьший приоритет.

3.10.3.4 Программно доступные регистры контроллера внешних прерываний

Все регистры контроллера внешних прерываний 32-разрядные, каждый бит регистра соответствует одному входному сигналу запроса. Регистры блока представлены в карте памяти NMPU. Имеются 4 способа записи в каждый регистр:

нормальный доступ - устанавливает в регистре записываемое значение,

побитовая установка - устанавливает значение 1 в тех разрядах, в которых имеются единицы в записываемом данном, остальные разряды не меняются,

побитовый сброс - устанавливает значение 0 в тех разрядах, в которых имеются единицы в записываемом данном, остальные разряды не меняются,

обнуление - устанавливает значение 0 во всех битах регистра, независимо от записываемого значения (т.е. эффект производит сам факт записи по выделенному адресу).

Для каждого способа записи выделен отдельный адрес. Некоторые способы для некоторых регистров недоступны. Чтение регистра может производиться по любому из описанных адресов. Ниже показаны адреса регистров контроллера внешних прерываний (см. таблицу 3.18).

Таблица 3.18 - Регистры контроллера внешних прерываний

Адрес (32-разрядный доступ)	Название	Доступ	Описание
0002_0400h	IRR	ЧТ	Регистр запросов на прерывание с номерами от 0 до 31. Значение 1 в бите данного регистра показывает наличие запроса, для которого ещё не выполнена команда перехода по адресу-вектору прерывания. Программный сброс битов данного регистра производится только с помощью регистра IRR_CLR.
0002_0402h	IRR_SET	ЧТ/ЗП	Побитовая установка регистра IRR.
0002_0404h	IRR_CLR	ЧТ/ЗП	Побитовый сброс регистра IRR. Программный сброс бита регистра IRR следует делать, когда данный запрос замаскирован.
0002_0406h	-	-	Зарезервировано.
0002_0408h	IMR	ЧТ/ЗП	Регистр маски прерываний с номерами о 0 до 31. Значение 0 - прерывание запрещено, 1 - прерывание разрешено.
0002_040Ah	IMR_SET	ЧТ/ЗП	Побитовая установка регистра IMR.
0002_040Ch	IMR_CLR	ЧТ/ЗП	Побитовый сброс регистра IMR.
0002_040Eh	IMR_NULL	ЧТ/ЗП	Сброс всех битов регистра IMR.
0002_0410h	IRP	ЧТ/ЗП	Регистров приоритетов прерываний с номерами от 0 до 31. Значение 0 – высокий приоритет, 1 – низкий приоритет.
0002_0412h	IRP_SET	ЧТ/ЗП	Побитовая установка регистра IRP.
0002_0414h	IRP_CLR	ЧТ/ЗП	Побитовый сброс регистра IRP.
0002_0416h	IRP_NULL	ЧТ/ЗП	Сброс всех битов регистра IRP.
0002_0418h	IAS	ЧТ	Регистр подтверждения и статуса запросов с номерами от 0 до 31. Значение 0 - запрос обработан, 1 - запрос обрабатывается (выполнен или выполняется переход по адресу-вектору, но бит регистра IAS ещё не сброшен).
0002_041Ah	-	-	Зарезервировано.
0002_041Ch	IAS_CLR	ЧТ/ЗП	Побитовый сброс регистра IAS.
0002_041Eh	-	-	Зарезервировано.

Регистр запросов на внешние прерывания (IRR)

Таблица 3.19 – Описание полей регистра IRR

Разряды	Доступ	Описание
31-0	ЧТ/ЗП	Битовая строка, в которой i-й разряд соответствует i-му запросу. При чтении каждый бит показывает состояние запроса: 0 – запрос не активен или уже удовлетворён, 1 – запрос активен. Запись значения 1 в бит сбрасывает этот бит.

Бит регистра IRR устанавливается аппаратно в момент приёма запроса, снимается также аппаратно в момент, когда ядро принимает адрес-вектор данного прерывания. Бит может установиться аппаратно, даже если данный запрос замаскирован - в этом случае перед снятием маски следует выполнить программный сброс бита путём записи значения 1.

Функция побитовой установки регистра IRR добавлена для упрощения отладки программного обеспечения.

Биты 31-28 данного регистра не используются, запись в эти биты блокируется, при чтении читаются нули.

Регистр маски прерываний (IMR)

Таблица 3.20 - Описание полей регистра IMR

Разряды	Доступ	Описание
31-0	ЧТ/ЗП	Битовая строка, в которой i-й разряд соответствует i-му запросу. Значения битов: 0 – запрос маскирован, 1 – запрос разрешён.

Регистр маски прерываний разрешает или запрещает прерывание по отдельным сигналам запроса. Если бит регистра IMR имеет значение 0, то соответствующее прерывание запрещено, если значение 1 - то разрешено. Значение регистра допускается изменять в любой момент времени. При снятии маски следует иметь в виду, что разрешаемый запрос может быть активен, то есть установлено значение 1 в бите регистра IRR. Если такой запрос не нужно обрабатывать, то перед снятием маски следует снять активный бит регистра IRR путём записи в этот бит значения 1.

Биты 31-28 данного регистра не используются, запись в эти биты блокируется, при чтении читаются нули.

Регистр приоритетов (IRP)

Таблица 3.21 - Описание полей регистра IRP

Разряды	Доступ	Описание
31-0	ЧТ/ЗП	Битовая строка, в которой i-й разряд соответствует i-му запросу. Значения битов: 0 – высокий уровень приоритета, 1 – низкий уровень приоритета.

Регистр приоритетов устанавливает уровень приоритета для каждого сигнала запроса. Допускается менять значение этого регистра в любой момент времени.

Биты 31-28 данного регистра не используются, запись в эти биты блокируется, при чтении читаются нули.

Регистр подтверждения и статуса запросов (IAS)

Таблица 3.22 - Описание полей регистра IAS

Разряды	Доступ	Описание
31-0	ЧТ/ЗП	Битовая строка, в которой i-й разряд соответствует i-му запросу. При чтении показывает состояние запроса от периферийного устройства: 0 - запрос обработан, 1 - запрос обрабатывается (выполнен или выполняется переход по адресу-вектору, но бит регистра IASL ещё не сброшен).

Регистр подтверждения и статуса используется только для запросов, работающих в режиме программного подтверждения. Бит в регистре IAS устанавливается аппаратно в момент, когда ядро принимает адрес-вектор данного прерывания. Сброс бита производится программно с помощью регистра побитового сброса (IAS_CLR или IAS_CLR). Во время, когда бит регистра IAS установлен, аппаратная установка соответствующего бита в регистре IRR блокируется контроллером. Сброс бита регистра IAS разрешает следующий запрос по той же линии.

Для запросов, работающих в режиме аппаратного подтверждения, биты регистра IAS всегда имеют значение 0.

Биты 31-28 данного регистра не используются, запись в эти биты блокируется, при чтении читаются нули.

3.10.4 Блок таймеров NMPU

В состав периферийных узлов NMPU входят два независимых 32-разрядных таймера, которые пользователь может использовать для отсчета задаваемых интервалов времени.

Таймер может работать в одном из двух режимов: в режиме однократного запуска и в непрерывном режиме. Интервал счета таймера задается программно. В качестве сигнала счета используется тактовый сигнал процессора.

По достижении нулевого значения таймер формирует запрос на прерывание, который может быть обработан стандартным образом.

Основные элементы архитектуры (см. рисунок 3.36) блока таймеров: регистры счётчика, теневые регистры, инкрементеры, схема управления.

Рисунок 3.36 - Структурная схема блока таймеров

3.10.4.1 Программно доступные регистры блока таймеров

Каждый таймер в блоке таймеров имеет 2 регистра: регистр счётчика и регистр настройки.

Таблица 3.23 – Перечень регистров блока таймера

Адрес (32-разрядный доступ)	Название	Доступ	Описание
0002_0800h	TMC0	ЧТ/ЗП	Регистр счётчика таймера 0
0002_0802h	TMM0	ЧТ/ЗП	Регистр настройки таймера 0
0002_0804h	TMC1	ЧТ/ЗП	Регистр счётчика таймера 1
0002_0806h	TMM1	ЧТ/ЗП	Регистр настройки таймера 1

Регистр счётчика таймера (TMCx)

Таблица 3.24– Описание полей регистра TMCx

Разряды	Доступ	Описание
31-0	ЧТ/ЗП	Запись обновляет теневой регистр, чтение возвращает текущее значение рабочего регистра.

Запись в регистр счётчика производится через теневой регистр. Теневой регистр доступен для записи всегда. Копирование значения из теневого регистра в рабочий происходит по любому из условий:

- при запуске таймера (запись значения 1 в бит TA регистра настройки),

- в момент переключения, если текущее значение таймера равно нулю.

При чтении всегда возвращается значение рабочего регистра.

Регистр настройки таймера (TMMx)

Каждый таймер имеет свой регистр настройки. Регистр доступен на запись и чтение.

Таблица 3.25 - Описание полей регистра TMMx

Разряды	Название	Доступ	Описание
[1]	TM	ЧТ/ЗП	Режим работы таймера: 0 – однократный режим работы таймера, 1 - непрерывный режим работы таймера.
[0]	TA	ЧТ/ЗП	Разрешение работы таймера: 0 – таймер остановлен, 1 - таймер осуществляет счет.

3.10.4.2 Принцип работы таймеров

Ниже представлены временные диаграммы работы таймера (см. рисунок 3.37).

Рисунок 3.37 – Временные диаграммы работы таймера

После разрешения работы таймера, по следующему положительному фронту тактового сигнала NMPU, начальное значение, соответствующее отсчитываемому временному интервалу, переписывается из теневого регистра таймера в рабочий. После этого рабочий регистр таймера инкрементируется в каждом такте. При обнулении рабочего счетчика формируется сигнал запроса прерывания (Interrupt Request), который обрабатывается блоком прерываний стандартным образом. Одновременно с этим, если задан непрерывный режим работы таймера, из теневого регистра в рабочий переписывается значение, определяющее отсчитываемый временной интервал. Если задан однократный режим работы таймера, то счет прекращается, и соответствующий бит TA регистра настройки сбрасывается.

Длительность временного интервала, отсчитываемого таймером, задается в дополнительном коде. Таким образом, длительность отсчитываемого временного интервала может изменяться от одного (код FFFFh) до 2^{32} (код 0000h) процессорных такта.

4 Структура процессорного кластера РС

4.1 Структурная схема процессорного кластера РС

Процессорный кластер РС является основной регулярной частью, на базе которой строится высокопроизводительная СнК 1879ВМ8Я. Структурная схема процессорного кластера приведена ниже (см. рисунок 4.1).

Рисунок 4.1 - Структурная схема процессорного кластера РС

Процессорный кластер включает в себя следующие основные блоки:

NMPU0…NMPU3 - четыре вычислительных узла NeuroMatrix на базе процессорного ядра NMC4, которые соединены непосредственно «каждый с каждым» тремя 64-разрядными полнодуплексными коммуникационными портами. Обмен данными через эти порты выполняется с использованием механизма прямого доступа в память (ПДП). Дополнительно четыре узла NMPU соединены между собой через 64-разрядный коммутатор 5 в 5 шины AXI 3.0 (**AXI_NMPU 5 -> 5**), обеспечивающий произвольный доступ узлам NMPU во внутренние банки друг друга. Через этот же коммутатор узлы NMPU подключаются к управляющему процессорному узлу кластера (CPU).

CPU – управляющий процессорный узел кластера, содержащий: управляющий скалярный процессор ARM Cortex-A5, внутреннюю память, а также интерфейсы для интеграции кластера в вычислительную систему. Для связи с внешними блоками используются 4 высокоскоростных коммуникационных канала связи CLLx и два среднескоростных шинных интерфейса стандарта AXI3 (PCu и PCd). Для связи внутри кластера используется 64-разрядный коммутатор 4 в 8 шины AXI 3.0 (**AXI_CPU**).

Cortex-A5 – управляющий скалярный процессор ARM Cortex-A5.

CMB0…CMB3 – четыре банка статической памяти объёмом по 8К*64 разряда, причём банк CMB0 поддерживает как обычный, так и эксклюзивный доступ в память.

CP0…CP3 – четыре контроллера коммуникационных портов, соединенных с узлами NMPU через коммутаторы каналов связи (**LC0...LC3**). Каждый коммутатор позволяет организовать один из трёх вариантов соединения (на примере LC0):

1. Коммуникационный порт CP0 узла NMPU0 соединяется с портом CP0 кластера PC. В таком режиме узел NMPU получает быстрый канал доступа к внутренним банкам памяти кластера CMBx, x=0,…3.
2. Коммуникационный порт CP0 узла NMPU0 соединяется с внешним устройством через канал связи CLL0. В этом режиме узел NMPU получает внешний канал доступа, который может быть использован для соединения нескольких PC-кластеров между собой или для подсоединения внешних устройств.
3. Коммуникационный порт CP0 кластера PC соединяется с внешним устройством через канал связи CLL0. В этом режиме управляющий узел CPU получает внешний канал доступа, который может быть использован для соединения нескольких PC-кластеров между собой или подсоединения внешних устройств.

Для доступа контроллеров коммуникационных портов к банкам памяти CMB0…CMB3 служит 64-разрядный коммутатор 4 в 4 шины AXI 3.0 (AXI_CMB), обеспечивающий доступ любого контроллера коммуникационных портов к любому банку памяти.

EL - внешний 4-проводной дуплексный линк, работающий на частоте 5 ГГц и предназначенный для обмена данными с другими процессорными системами. Каждый такой линк обеспечивает пропускную способность в 2 ГБайта/с в каждую сторону. Он состоит из контроллера линка (**ELC**) и физического интерфейса линка (**EL PHY**). Линк обменивается данными через порт L3 контроллера коммуникационных портов LC0, который обеспечивает соединение либо с коммуникационным портом CP0 узла NMPU0, либо с портом CP0 кластера, либо с внешним устройством через канал связи CLL0.

EMI CPU – контроллер 32-разрядного интерфейса с внешней динамической памятью DDR3-1600 (эффективная частота передачи данных – 1600 МГц). Максимальный объем поддерживаемой памяти – 1Гбайт.

MDMAC – высокопроизводительный контроллер ПДП. Основной задачей этого контроллера является подкачка данных из внешней памяти DDR3 во внутренние банки кластера CMBx, x=0,…3. Контроллер также имеет доступ к внутренним банкам памяти всех NMPU.

MPU – блок защиты памяти. Управляющий узел CPU может запрещать доступ внешних устройств к внутренним банкам CPU и NMPU. В случае обращения в защищаемую область памяти операция не производится, а генерирующему запрос устройству выдается признак ошибки SLVERR. Этот механизм может использоваться для организации межпроцессорного взаимодействия (например, для организации семафоров).

GIC – контроллер прерываний, который поддерживает до 64-х прерываний с программно настраиваемыми адрес-векторами и приоритетами.

DIT – два интервальных 32-разрядных таймера.

GPIO/EXTINT – универсальный блок интерфейсов общего назначения. Он управляет работой 8-ми внешних выводов, каждый из которых может программно настраиваться на функционирование в качестве выводов общего назначения (GPIO), внешних входов прерывания (EXTINT), а также пара выводов может реализовать интерфейс «запрос-подтверждение».

CLSC - системный контроллер кластера. Он выполняет следующие функции: содержит идентификатор своего кластера, а также служит для генерации запросов на прерывание к другим кластерам.

CRGH - Генератор тактовых сигналов и сигналов сброса. Он предназначен управлять генерацией синхросигналов и сигналов сброса для **NMPU0…NMPU3**.

4.1.1 Основные потоки данных в РС-кластере

Ниже приведены основные потоки данных в NM-кластере и способы их обеспечения.

Обмен данными с внешней памятью. РС-кластер осуществляет загрузку/выгрузку данных во внешнюю память через 32-разрядный интерфейс DDR3-1600. Максимальная производительность при обмене с внешней памятью обеспечивается при использовании механизма ПДП (контроллер MDMAC подкачивает/выгружает данные в/из блоки(ов) внутренней памяти CMBx). Узлы NMPU, в свою очередь, используют коммуникационные порты CP0 для обмена данными между внутренними банками кластера CMBx и своими внутренними банками NMBx. Пиковая производительность интерфейса с внешней памятью – 6.4 ГБайт/с.

Процессорные узлы NMPU имеют возможность произвольного доступа во внешнюю память через шинную структуру, но этот способ рассматривается как вспомогательный, т.к. его эффективность гораздо ниже, чем использование каналов ПДП.

Обмен данными между банками внутренней памяти узлов NMPU. Основной способ обмена данными между узлами NMPU – использование коммуникационных портов (CP1..CP3). Блоки коммуникационных портов совмещены с внутренними каналами ПДП и предназначены для двунаправленного полнодуплексного безадресного обмена данными между двумя вычислительными узлами NMPU. Такой способ позволяет быстро передавать данные из внутренней памяти одного NMPU во внутреннюю память другого NMPU без участия процессорного ядра. Наличие 8-х банков памяти позволяет производить операции передачи данных на фоне вычислений. Пиковая односторонняя пропускная способность одного порта 8 ГБайт/с (16 ГБайт/с в обе стороны).

Другой способ обмена между внутренними банками памяти – произвольный доступ процессорных ядер в память соседних узлов NMPU через шинную структуру. Этот способ неэффективен по сравнению с использованием коммуникационных портов, и его рекомендуется рассматриваться как вспомогательный.

Настройка и мониторинг РС-кластера управляющим ARM-процессором. Управляющий процессор Cortex A5 имеет доступ ко всем внутренним банкам узлов NMPU(NMBx), банкам памяти CMBx, а также управляет всеми периферийными устройствами CPU через шину APB. Управляющий процессор не имеет доступа к периферийным регистрам узлов NMPU.

4.1.2 Шинная структура процессорного кластера РС

Соединение основных компонентов NM-кластера выполнено по стандарту AMBA AXI 3.0 с использованием шинных коммутаторов (далее упоминается как «шинная структура»).

Процессорный кластер РС включает 3 основных коммутатора:

- **AXI_NMPU** (5в5) работает на частоте 1ГГц. Обеспечивает произвольный доступ во внутреннюю память NMBx от соседних узлов NMPU, мастер-устройств управляющего узла кластера CPU (Cortex A5, MDMAC), а также мастер-устройств верхнего уровня иерархии СнК (через слэйв-интерфейс S_NMPU).

Используя мастер-интерфейс M_NMPU, узлы NMPU получают произвольный доступ в банки памяти кластера CMBx, к периферийным устройствам кластера на шине APB, а также выход на верхний уровень иерархии СнК.

- **AXI_CMB** (4в4) работает на частоте 800МГц. Обеспечивает доступ во внутренние банки CMBx со стороны коммуникационных портов.

- **AXI_CPU** (4в8) работает на частоте 800МГц. Обеспечивает доступ во внутренние банки CMBx и внешнюю DDR память со стороны мастер-устройств управляющего узла кластера CPU (Cortex A5, MDMAC), а также мастер-устройств верхнего уровня иерархии СнК (через слэйв-интерфейс PCu).

Используя мастер-интерфейс PCd, узлы NMPUx и CPU получают доступ во внутреннюю память соседних кластеров РС, а также во внутреннюю память и периферийные регистры верхнего уровня иерархии СнК.

Этот коммутатор содержит мост AXI-APB к которому подключены периферийные устройства кластера, что позволяет выполнять их программирование узлами NMPU, Cortex’ом узла CPU, а также мастер-устройствами верхнего уровня иерархии.

4.2 Карта памяти CPU

Карта памяти CPU приведена на рисунке ниже (см. рисунок 4.2). Следует отметить, что в данном случае задаётся адресация до байта.

Рисунок 4.2 - Карта памяти CPU

Карты памяти всех управляющих ПУ кластеров (CPU) одинаковы и делятся на следующие области памяти:

- CMB0-CMB3 – банки внутренний памяти своего кластера.
- REG_CPU - блок собственных периферийных регистров конкретного CPU. При доступе в эту область памяти идёт переадресация на периферийную шину APB.
- NMPU0 – NMPU3 - области внутренней памяти всех NMPU, входящих в состав своего кластера. Эти области являются отражением внутренних банков памяти NMMBx всех четырех вычислительных узлов NMPU, позволяя обеспечить произвольный доступ к этим банкам со стороны CPU и соседних NMPU. Область периферийных регистров узлов NMPU не доступна со стороны CPU.
- EMI_CU - банк внешней памяти DDR своего кластера. При этом при использовании DDR-памяти максимального объема (1 ГБайт) со стороны CPU будут недоступны 3 МБайт младших ячеек, т.к. это адресное пространство используется под банки внутренней памяти кластера.
- PCIe - области памяти высокоскоростного линка для связи своего кластера с внешними процессорными системами .
- CCPU - области памяти центрального процессорного узла.
- PC0 – PC3 - адресные пространства всех кластеров, что позволяет CPU обмениваться данными как с управляющими ПУ чужих кластеров, так и с NMPU своего и чужих кластеров. При этом обращение в эти области памяти будут выполняться заметно медленнее, чем в остальные области.

4.3 Управляющий скалярный процессор ARM Cortex-A5

Процессорное ядро ARM Cortex-A5 включает в себя целочисленное ядро, базирующееся на архитектуре ARMv7, сопроцессор с плавающей точкой и кэш-память уровня L1. Ядро поддерживает системы команд ARM и Thumb™, технологию Jazelle для обеспечения возможности прямого выполнения байт-кодов Java. Сопроцессор с плавающей точкой поддерживает архитектуру ARMv7 VFPv4-D16 и соответствует стандарту IEEE754.

Использованное в СБИС 1879ВМ8Я процессорное ядро Cortex A5 основано на его версии r0p1 и имеет конфигурацию, имеющую следующие свойства:

- 64 разрядный AXI интерфейс;
- пятиступенчатый in-order конвейер с динамическим предсказанием переходов;
- интегрированные сопроцессор отладки CP14 и системный сопроцессор CP15;

- устройства управления памятью команд и данных Memory Management Unit (MMU);
- L1 кэш память команд объемом 32 КБайт;
- L1 кэш память данных объемом 32 КБайт;
- сопроцессор для вычислений с плавающей запятой(VFP);
- поддержка трассировки программ;
- система отладки на базе JTAG.

Структурная схема процессорного ядра представлена на рисунке ниже (см. рисунок 4.3).

Рисунок 4.3 – Структурная схема процессорного ядра Cortex-A5

Подробное описание процессорного ядра Cortex A5 можно найти в документе ARM DDI 0433B «Cortex-A5. Revision r0p1. Technical Reference Manual». Описание архитектуры ARMv7 можно найти в документе ARM DDI 0406C «ARM Architecture Reference Manual. ARMv7-A and ARMv7-R edition».

4.4 Внутренняя память кластера РС

Объем внутренней памяти кластера составляет 2 Мбит. Она разделена на четыре банка памяти объёмом по 512 Кбит.

Каждый банк внутренней памяти кластера (CMB0 ... CMB3) имеет двухпортовую организацию. По одному из портов производится чтение и запись данных, а также выборка команд процессорным ядром ARM Cortex-A5. Также этот порт используется для доступа в память со стороны процессорных устройств NMPU этого кластера, центрального управляющего узла (CCPU), а также процессорных устройств NMPU и CPU других кластеров. Второй порт (порт ПДП) обслуживает только чтение и запись данных под управлением контроллеров коммуникационных портов CP0 … CP3. Таким образом, каждый банк внутренней памяти способен отработать до двух запросов в каждом процессорном такте.

Банки памяти CMB0 ... CMB3 структурно и функционально ничем не отличаются от банков внутренней памяти NMMB0…NMMB7 процессорных устройств NMPU, которые были подробно описаны в разделе 3.4.

Исключением является банк CMB0, так как это единственный банк в кластере, который поддерживает эксклюзивные операции. Далее будут описаны особенности выполнения эксклюзивных операций в этом банке как со стороны управляющих ПУ, так и со стороны процессорных устройств NMPU.

4.4.1 Общие сведения о механизме эксклюзивного доступа

Механизм эксклюзивного доступа был введён фирмой ARM Ltd вместо локированных операций при организации семафоров в памяти в многопроцессорных и многопоточных системах. С помощью операций эксклюзивного чтения и эксклюзивной записи строятся примитивы синхронизации между выполняющимися потоками команд.

За основу взят порядок работы инструкций LDREX и STREX ядра ARM, при этом пришлось разработать совместимый механизм для процессорных устройств NMPU, которые в явном виде эксклюзивный доступ не поддерживают.

Спецификацию команд LDREX и STREX можно найти в документе *"ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition" (A3.4)*, спецификацию эксклюзивных транзакций на шинах - в документе *"AMBA AXI Protocol Version: 2.0 Specification" (6.2).*

4.4.2 Общий порядок эксклюзивных операций

Типовой порядок эксклюзивного доступа, независимо от типа процессорного ядра, состоит в следующем:

1. Процессорное ядро производит эксклюзивное чтение из выбранной ячейки памяти, при этом адрес этой ячейки резервируется в контроллере памяти.

2. Процессорное ядро производит попытку эксклюзивной записи в ту же самую ячейку памяти и резервирование снимается.

Эксклюзивная запись может завершиться:

- *успешно* - когда между эксклюзивным чтением и эксклюзивной записью не было другой записи в выбранную ячейку;

- *не успешно* - когда между эксклюзивным чтением и эксклюзивной записью другое процессорное ядро или другое мастер-устройство совершило запись в данную ячейку.

Успешная эксклюзивная запись обновляет значение ячейки, неуспешная - не обновляет.

Контроллеры памяти имеют ограниченные возможности по резервированию ячеек памяти для эксклюзивных операций:

- в банках памяти CMB0 – имеется по 6 позиций (мониторов) для резервирования,
- в контроллерах внешней памяти (EMI_CPU) – по 2 монитора.

Остальные контроллеры памяти в банках CMB1-3 и банках внутренней памяти NMPU не поддерживают эксклюзивного доступа.

Правила занятия мониторов:

1. Одно и то же ядро в одном и том же контроллере памяти не может зарезервировать более одной ячейки – последующее резервирование снимает предыдущее.
2. Когда заняты все мониторы и новое эксклюзивное чтение не может занять монитор в соответствии с правилом 1, тогда резервирование, выполненное раньше других, снимается и новый запрос занимает его место.

Если несколько процессорных ядер пытаются выполнить описанный порядок эксклюзивного доступа в цикле и их число больше доступного числа мониторов в контроллере памяти, то возможна ситуация зацикливания. Следует предотвращать возникновение подобной ситуации программно.

4.4.3 Эксклюзивные операции ядра ARM

Эксклюзивное чтение ядром ARM выполняется с помощью инструкции:

`LDREX Rt, [Rn];` где Rn - адрес в памяти, Rt - регистр-приёмник. Следует отметить, что при выполнении данной инструкции возможно исключение, если контроллер памяти не поддерживает эксклюзивного доступа.

Эксклюзивная запись ядром ARM выполняется с помощью инструкции

`STREX Rd, Rt, [Rn];` где Rn - адрес в памяти, Rt - регистр-источник, Rd - регистр, в который записывается статус эксклюзивной записи: 0 - запись успешная, 1 - запись не успешная.

В архитектуре ARMv7 имеются эксклюзивные операции с 8-, 16-, 32- и 64-разрядным данным.

4.4.4 Эксклюзивные операции NMPU

В системе команд процессорного ядра NMC4 нет выделенных команд для эксклюзивного доступа. Предлагается использовать для этого зарезервированные адреса в адресном пространстве NMPU, отображая их в доступные области, но с признаком эксклюзивного доступа. Для NMPU признаком того, что команда чтения или записи является эксклюзивной, служит старший (31-й) бит адреса обращения. В таблице ниже приведены все адреса, по которым возможен эксклюзивный доступ со стороны NMPU (с учётом карты памяти и набора контроллеров памяти, поддерживающих эксклюзивные транзакции).

Таблица 4.1 - Адреса для эксклюзивного доступа NMPU

Банк памяти	**Адреса для доступа в обычном режиме (32-разрядный доступ)**	**Соответствующие адреса для эксклюзивного доступа (32-разрядный доступ)**
CMB0 своего кластера	000C_0000-000C_3FFF hex	800C_0000-800C_3FFF hex
EMI_CPU своего кластера	000D_0000-0FFF_FFFF hex	800D_0000-8FFF_FFFF hex
CCMB0	1800_0000-1800_3FFF hex	9800_0000-9800_3FFF hex
EMI _CCPU	1810_0000-1FFF_FFFF hex	9810_0000-9FFF_FFFF hex
CMB0 кластера 0	2000_0000-2000_3FFF hex	A000_0000-A000_3FFF hex
EMI_PU кластера 0	2010_0000-27FF_FFFF hex	A030_0000-A7FF_FFFF hex
CMB0 кластера 1	2800_0000-2800_3FFF hex	A800_0000-A800_3FFF hex
EMI_CPU кластера 1	2830_0000-2FFF_FFFF hex	A830_0000-AFFF_FFFF hex
CMB0 кластера 2	3000_0000-3000_3FFF hex	B000_0000-B000_3FFF hex
EMI_CPU кластера 2	3030_0000-37FF_FFFF hex	B030_0000-B7FF_FFFF hex
CMB0 кластера 3	3800_0000-3800_3FFF hex	B800_0000-B800_3FFF hex
EMI_CPU кластера 3	3830_0000-3FFF_FFFF hex	B830_0000-BFFF_FFFF hex

Все эксклюзивные операции проводятся исключительно по адресам, выделенным для эксклюзивного доступа. Обращения по адресам из интервала 4000_0000 - FFFF_FFFF hex, но не попадающие в указанные выше диапазоны, будут восприниматься как обычные обращения по соответствующим адресам из интервала 0000_0000 - 3FFF_FFFF hex.

Эксклюзивное чтение NMPU производится с помощью одной команды чтения любого из трёх типов:

- скалярная команда чтения из памяти в регистр (32-разрядное слово),
- скалярная команда чтения из памяти в регистровую пару (64-разрядное слово),
- одиночная (rep 1) векторная команда чтения (64-разрядное слово).

Эксклюзивная запись производится в две фазы: *фазу данных и фазу статуса*. Фаза данных представляет собой одну команду записи любого из трёх типов:

- скалярная команда записи из памяти в регистр (32-разрядное слово),
- скалярная команда записи из памяти в регистровую пару (64-разрядное слово),
- одиночная (rep 1) векторная команда записи (64-разрядное слово).

При этом фаза данных эксклюзивной записи должна производиться по тому же адресу и передавать слово той же разрядности, что и соответствующая операция эксклюзивного чтения – только в этом случае она может стать успешной.

Фаза статуса эксклюзивной записи представляет собой одну команду чтения любого из двух типов:

- скалярная команда чтения из памяти в регистр (32-разрядное слово),
- скалярная команда чтения из памяти в регистровую пару (64-разрядное слово).

Адрес обращения в фазе статуса может быть любым из специально выделенного интервала: 400C_C000-400C_FFFF hex (попадает в область банка памяти СМВ3, который не поддерживает эксклюзивные операции). Чтение в фазе статуса возвращает в регистр-приёмник 32-разрядное значение: 0 - когда эксклюзивная запись прошла успешно, 1 - когда не успешно. В 64-разрядном слове статуса в старшей половине слова нули (фактически, читаемое из резервной области значение).

Рекомендуется делать так, чтобы между фазами эксклюзивной записи не могло возникнуть прерывание. Наиболее простой способ - помещать соответствующие инструкции в одно 64-разрядное слово, например:

```
ar5 = MyMutex;          // (1) - MyMutex - адрес синхроячейки в
обычном режиме
```

```
ar5 = ar5 + 80000000h;   //  (2)  -  получаем  адрес  синхроячейки  в
режиме эксклюзивного доступа
ar6 = C0050000h;         // (3) - адрес чтения в фазе статуса
[ar5]=ar0,gr0;           //(4) - фаза данных эксклюзивной записи -
пишем длинное слово из пары ar0,gr0
gr7=[ar6];               //(5)   -   фаза   статуса   при   чтении
возвращается статус 0 – успешно, 1 – не успешно
```

В этом примере инструкции (4) и (5) попадают при ассемблировании *в одно 64*-разрядное слово из-за "длинной" (64-разрядной) инструкции (3), поэтому они не могут быть при выполнении разделены обработкой прерывания.

4.5 Контроллеры коммуникационных портов CP0…CP3

Контроллеры коммуникационных портов (CP0…CP3) позволяют осуществлять прямой доступ во внутреннюю память кластера (CMB0…CMB3) за данными, которые будут приняты или переданы коммуникационными портами.

Каждый контроллер CPx содержит два канала ПДП:

- канал память-CPx – чтение данных для последующей передачи их по коммуникационному порту;
- канал CPx-память – запись данных, принятых по коммуникационному порту, в память;

Данные каналы предназначены для обслуживания внешних коммуникационных портов кластера, организации канала обмена данными между кластерами внутри СнК, а также для быстрой связи с NMPU внутри своего кластера. Передающий и приёмный каналы одного коммуникационного порта могут запускаться и работать независимо друг от друга.

Коммуникационные порты структурно и функционально ничем не отличаются от аналогичных портов, входящих в состав процессорных устройств NMPU и подробно описанных в разделе 3.7. Поэтому ниже будет дано только описание программно-доступных регистров этих портов.

Основная разница между портами CPU и аналогичных портов, относящихся к процессорным устройствам NMPU, состоит в следующем:

- Коммуникационные порты CPU при своей работе используют байтовый доступ, поскольку управляющий ими процессор ARM Cortex-A5 использует адресацию до байта. Соответственно, обращение к периферийным регистрам портов осуществляется также с помощью байтового доступа.
- Коммуникационные порты NMPU при своей работе используют 32-разрядный доступ, поскольку управляющее ими процессорное ядро NMC4 использует адресацию до 32-разрядного слова. Соответственно, обращение к периферийным регистрам портов осуществляется также с помощью 32-разрядного доступа.

Только NMPU может управлять своими портами, больше никто не имеет к ним программного доступа. Вместе с тем, это ПУ может также получить доступ к портам CPU через соответствующую область памяти PC0 – PC3. В этом случае оно должно преобразовать байтовые адреса периферийных регистров в 32-разрядные, просто поделив их на 4.

4.5.1 Программная модель передающей части контроллера коммуникационного порта

Все регистры передающей части контроллера коммуникационных портов 32-разрядные, в зарезервированных битах при чтении возвращается 0, запись любого значения в зарезервированные биты никак не влияет на устройство. Набор регистров для каждого коммуникационного порта одинаковый. Адреса отличаются только начальным смещением Base (доступ байтовый):

- Base = 0B_8000h для CP0;
- Base = 0B_9000h для CP1;
- Base = 0B_A000h для CP2;
- Base = 0B_B000h для CP3;

Набор регистров передающей части контроллера коммуникационного порта представлен ниже (см. таблицу 4.2).

Таблица 4.2 - Набор регистров передающей части контроллера коммуникационного порта

Адрес (байтовый доступ)	Название	Доступ	Пояснение
Base+00h	MainCounter	ЧТ/ЗП	Основной счётчик данных
Base+04h	Address	ЧТ/ЗП	Регистр текущего адреса в адресном генераторе
Base+08h	Bias	ЧТ/ЗП	Регистр смещения адреса
Base+0Ch	RowCounter	ЧТ/ЗП	Счётчик последовательных данных
Base+10h	AddressingMode	ЧТ/ЗП	Регистр режима адресации
Base+14h	CSR	ЧТ/ЗП	Регистр управления
Base+18h	InterruptMask	ЧТ/ЗП	Регистр масок запросов на прерывание
Base+1Ch	InternalState	ЧТ	Регистр состояния

Примечание: Все адреса и смещения задаются для байтового доступа (регистры Address и Bias); Все счётчики считают переданные 64-разрядные слова (счётчики MainCounter и RowCounter).

4.5.2 Программная модель приёмной части контроллера коммуникационного порта

Все регистры приёмной части контроллера 32-разрядные, в зарезервированных битах при чтении возвращается 0, запись любого значения в зарезервированные биты никак не влияет на устройство. Набор регистров приемной части блока коммуникационного порта представлен ниже (см. таблицу 4.3).

Таблица 4.3 - Набор регистров приемной контроллера ПДП коммуникационного порта

Адрес (байтовый доступ)	Название	Доступ	Пояснение
Base+20h	MainCounter	ЧТ/ЗП	Основной счётчик данных
Base+24h	Address	ЧТ/ЗП	Регистр текущего адреса в адресном генераторе
Base+28h	Bias	ЧТ/ЗП	Регистр смещения адреса
Base+2Ch	RowCounter	ЧТ/ЗП	Счётчик последовательных данных
Base+30h	AddressingMode	ЧТ/ЗП	Регистр режима адресации
Base+34h	CSR	ЧТ/ЗП	Регистр управления
Base+38h	InterruptMask	ЧТ/ЗП	Регистр масок запросов на прерывание
Base+3Ch	InternalState	ЧТ	Регистр состояния

Примечание: Все адреса и смещения задаются для байтового доступа (регистры Address и Bias); Все счётчики считают переданные 64-разрядные слова (счётчики MainCounter и RowCounter).

4.5.3 Прерывания контроллера коммуникационного порта

Коммуникационные порты могут выдавать по 2 запроса на прерывание: от передающей и приёмной части. Эти прерывания вызываются либо при завершении обмена данными, либо при возникновении ошибок при обмене.

4.6 Коммутаторы каналов связей LC0…LC3

Коммутаторы каналов связей используются для высокоскоростного подсоединения вычислительных узлов NMPUx к управляющему процессорному узлу кластера CPU. Для определенности для этого подключения всегда используется CP0 NMPU.

Каждый коммутатор LCx позволяет организовать один из трех вариантов соединения коммуникационных портов:

1. Коммуникационный порт CP0 узла NMPU соединен с портом CPx кластера PC. В таком режиме узел NMPU получает быстрый канал доступа к внутренним банкам памяти кластера CMBx.
2. Коммуникационный порт CP0 узла NMPU соединен с внешним устройством через канал связи CLL. В этом режиме узел NMPU получает внешний канал доступа, который может быть использован для соединения нескольких PC-кластеров между собой или подсоединения внешних устройств.
3. Коммуникационный порт CP кластера PC соединен с внешним устройством через канал связи CLL. В этом режиме управляющий узел CPU получает внешний канал доступа, который может быть использован для соединения нескольких PC-кластеров между собой или подсоединения внешних устройств.

Коммутатор LC0 имеет дополнительный порт, который позволяет подключиться к внешнему высокоскоростному линку EL. Это даёт возможность обеспечить не один обмен данными, как внутри коммутаторов LC1- LC3, а два в одно и то же время.

4.6.1 Структура коммутатора каналов связи

На рисунке 4.4 приведена структурная схема коммутатора каналов связи.

Рисунок 4.4 - Структурная схема коммутатора каналов связи

Коммутатор содержит программно доступные регистры, настройка которых осуществляется по периферийной шине APB. С помощью них задаётся, какой порт коммутатора с каким будет объединён.

4.6.2 Программно доступные регистры коммутаторов каналов связи

Программно доступные регистры коммутаторов каналов связи расположены в области памяти периферийных устройств REG_CPU и имеют базовые смещения:

Base = 0x000B_4000 hex для LC0;

Base = 0x000B_5000 hex для LC1;

Base = 0x000B_6000 hex для LC2;

Base = 0x000B_7000 hex для LC3.

Перечень регистров коммутаторов каналов связи представлен в таблице 4.4 ниже.

Таблица 4.4 - Перечень регистров коммутаторов каналов связи

Адрес (байтовый доступ)	Тип	Разряд-ность	Начальное значение	Имя	Описание
000B_4000 hex	ЧТ/ЗП	2	0x3	LC0_DST0	Регистр приёмного порта L0 коммутатора LC0
000B_4004 hex	ЧТ/ЗП	2	0x2	LC0_DST1	Регистр приёмного порта L1 коммутатора LC0
000B_4008 hex	ЧТ/ЗП	2	0x1	LC0_DST2	Регистр приёмного порта L2 коммутатора LC0
000B_400C hex	ЧТ/ЗП	2	0x0	LC0_DST3	Регистр приёмного порта L3 коммутатора LC0
000B_4010 hex	ЧТ/ЗП	2	0x3	LC0_SRC0	Регистр передающего порта L0 коммутатора LC0
000B_4014 hex	ЧТ/ЗП	2	0x2	LC0_SRC1	Регистр передающего порта L1 коммутатора LC0
000B_4018 hex	ЧТ/ЗП	2	0x1	LC0_SRC2	Регистр передающего порта L2 коммутатора LC0
000B_401C hex	ЧТ/ЗП	2	0x0	LC0_SRC3	Регистр передающего порта L3 коммутатора LC0
000B_5000 hex	ЧТ/ЗП	2	0x3	LC1_DST0	Регистр приёмного порта L0 коммутатора LC1
000B_5004 hex	ЧТ/ЗП	2	0x2	LC1_DST1	Регистр приёмного порта L1 коммутатора LC1
000B_5008 hex	ЧТ/ЗП	2	0x1	LC1_DST2	Регистр приёмного порта L2 коммутатора LC1
000B_500C hex	ЧТ/ЗП	2	0x3	LC1_SRC0	Регистр передающего порта L0 коммутатора LC1
000B_5010 hex	ЧТ/ЗП	2	0x2	LC1_SRC1	Регистр передающего порта L1 коммутатора LC1
000B_5014 hex	ЧТ/ЗП	2	0x1	LC1_SRC2	Регистр передающего порта L2 коммутатора LC1
000B_6000 hex	ЧТ/ЗП	2	0x0	LC2_DST0	Регистр приёмного порта L0 коммутатора LC2
000B_6004 hex	ЧТ/ЗП	2	0x0	LC2_DST1	Регистр приёмного порта L1 коммутатора LC2
000B_6008 hex	ЧТ/ЗП	2	0x0	LC2_DST2	Регистр приёмного порта L2 коммутатора LC2
000B_600C hex	ЧТ/ЗП	2	0x0	LC2_SRC0	Регистр передающего порта L0 коммутатора LC2
000B_6010 hex	ЧТ/ЗП	2	0x0	LC2_SRC1	Регистр передающего порта L1 коммутатора LC2
000B_6014 hex	ЧТ/ЗП	2	0x0	LC2_SRC2	Регистр передающего порта L2 коммутатора LC2
000B_7000 hex	ЧТ/ЗП	2	0x0	LC3_DST0	Регистр приёмного порта L0 коммутатора LC3
000B_7004 hex	ЧТ/ЗП	2	0x0	LC3_DST1	Регистр приёмного порта L1 коммутатора LC3
000B_7008 hex	ЧТ/ЗП	2	0x0	LC3_DST2	Регистр приёмного порта L2 коммутатора LC3
000B_700C hex	ЧТ/ЗП	2	0x0	LC3_SRC0	Регистр передающего порта L0 коммутатора LC3
000B_7010 hex	ЧТ/ЗП	2	0x0	LC3_SRC1	Регистр передающего порта L1 коммутатора LC3
000B_7014 hex	ЧТ/ЗП	2	0x0	LC3_SRC2	Регистр передающего порта L2 коммутатора LC3

Регистр приёмного порта Lx

Таблица 4.5 - Формат регистра приёмного порта Lx

Биты	Название	Тип	Выполняемая функция
[31:2]	Reserved	ЧТ	Резерв. При чтении выдаются все нули, запись в эти биты игнорируется.
[1:0]	SRC	ЧТ/ЗП	Номер передающего порта, с которым скоммутирован приёмный порт Lx.

Регистр задаёт, в какой передающий порт будут перенаправляться данные из приёмного порта Lx. Допускаются все значения, в том числе можно настроить передачу из приёмного в передающий порт с одинаковыми номерами.

Регистр передающего порта Lx

Таблица 4.6 - Формат регистра передающего порта Lx

Биты	Название	Тип	Выполняемая функция
[31:2]	Reserved	ЧТ	Резерв. При чтении выдаются все нули, запись в эти биты игнорируется.
[1:0]	DST	ЧТ/ЗП	Номер приёмного порта, с которым скоммутирован передающий порт Lx.

Регистр задаёт, из какого приёмного порта должен забирать данные передающий порт Lx. Допускаются все значения, в том числе можно настроить передачу из приёмного в передающий порт с одинаковыми номерами.

4.6.3 Коммутация портов

Если нужно настроить коммутаторов каналов связи LCi на передачу из порта x в порт y, то следует установить:

- LCi_DSTx = y,
- LCi_SRCy = x.

Если LCi_SRCy не равно x, то передача данных будет блокироваться до тех пор, пока не будет совершена настройка LCi_SRCy = x.

4.7 Контроллер ПДП память-память MDMAC

Контроллер обеспечивает обмен данными между разными блоками внутренней памяти (банки NMMBx, CMBx) и банками внешней DDR памяти. Особенностями контроллера ПДП являются:

- Наличие одного универсального канала, который состоит из двух подканалов – передающего и принимающего. Эти каналы настраиваются и запускаются программно, а останавливаются либо аппаратно (по окончанию обмена или по ошибке), либо программно.
- Встроенный буфер на 32 64-разрядных слов (256 байт).
- Аппаратная поддержка пакетов до 4 64-разрядных слов на шине AMBA AXI с целью эффективного использования SDRAM.

Передача данных реализуется контроллером с помощью транзакций чтения и записи на шине AMBA AXI: контроллер производит чтение данных из источника с помощью передающего подканала, помещает данные во встроенный буфер, затем записывает данные в устройство-приёмник с помощью принимающего подканала.

4.7.1 Программно доступные регистры контроллера MDMAC

Каналы ПДП настраиваются с помощью программно доступных регистров передающего и приёмного подканалов. Регистры контроллера расположены в адресном пространстве периферийных устройств REG_CPU.

Набор регистров каждого подканала состоит из:

- регистра текущего адреса (Address);
- регистра смещения адреса (Bias);
- счётчика последовательных данных (RowCounter);
- регистра режима адресации (AddressMode).

Кроме того, есть регистры, общие для обоих подканалов:

- основной счётчик данных (MainCounter);
- регистр управления (Control);
- регистр масок запросов на прерывание (InterruptMask);
- регистр состояния (State).

Перечисленные выше регистры (кроме регистра State), как по разрядности, так и по составу функциональных полей совпадают с регистрами передающего и приёмного каналов коммуникационных портов.

Список регистров приведен в таблице ниже (см. таблицу 4.7). Не указанные в таблице адреса зарезервированы – запись по зарезервированным адресам никак не влияет на устройство, при чтении выдаётся не специфицированное значение. Неиспользованные старшие разряды имеющихся регистров при чтении возвращают 0.

Таблица 4.7 - Список регистров контроллера MDMAC

Название регистра	**Адрес (байтовый доступ)**	**Разрядность**	**Доступ**
Регистры передающего подканала MDMAC			
DMATR_Address	0x000B_C004 hex	32	ЧТ/ЗП
DMATR_Bias	0x000B_C008 hex	32	ЧТ/ЗП
DMATR_RowCounter	0x000B_C00C hex	16	ЧТ/ЗП
DMATR_AddressMode	0x000B_C010 hex	1	ЧТ/ЗП
Регистры приёмного подканала MDMAC			
DMARC_Address	0x000B_C024 hex	32	ЧТ/ЗП
DMARC_Bias	0x000B_C028 hex	32	ЧТ/ЗП
DMARC_RowCounter	0x000B_C02C hex	16	ЧТ/ЗП
DMARC_AddressMode	0x000B_C030 hex	1	ЧТ/ЗП
Общие регистры обоих подканалов			
DMA_MainCounter	0x000B_C000 hex	16	ЧТ/ЗП
DMA_Control	0x000B_C014 hex	4	ЧТ/ЗП
DMA_InterruptMask	0x000B_C038 hex	2	ЧТ/ЗП
DMA_State	0x000B_C03C hex	15	ЧТ/ЗП

Регистр текущего адреса Address

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
Address																															

Рисунок 4.5 - Формат регистра Address

Запись в регистр Address устанавливает начальный адрес буфера в памяти, с которым работает соответствующий подканал. При чтении возвращается текущий адрес в адресном генераторе.

Регистр смещения адреса Bias

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
Bias																															

Рисунок 4.6 - Формат регистра Bias

Регистр Bias используется только при двухмерной адресации, содержит смещение адреса при переходе от конца одной строки передаваемого массива до начала следующей. Фактически – это размер (в единицах адреса) пропускаемого фрагмента.

Счётчик последовательных данных RowCounter

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																RowCounter															

Рисунок 4.7 - Формат регистра RowCounter

Регистр RowCounter используется только при двухмерной адресации. Запись в регистр RowCounter устанавливает размер строки (в словах) передаваемого массива.

Регистр режима адресации AddressingMode

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																															AM

Рисунок 4.8 - Формат регистра AddressingMode

Регистр AddressingMode определяет режим адресации в адресном генераторе. Бит AM=0 – одномерная адресация, бит AM=1 – двухмерная адресация. Подробнее о режимах адресации см. п. 3.7.3.

Основной счётчик данных MainCounter

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																MainCounter															

Рисунок 4.9 - Формат регистра MainCounter

Запись в регистр MainCounter устанавливает количество 64-разрядных слов, которое будет передано. При чтении возвращается количество слов, которое осталось прочитать из памяти в текущей передаче.

Регистр управления CSR

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																												Clr	ES	Cpl	En

Рисунок 4.10 - Формат регистра CSR

Регистр CSR используется для управления состоянием блока коммуникационного порта:

- Бит En (0-й разряд). Запись в бит En = 1 запускает передачу.

- Бит Cpl (1-й разряд). При чтении Cpl = 1 означает, что передача заданного количества слов завершены, либо запись приостановлена вследствие ошибки или программной приостановки (бит ES = 1). Чтобы произвести следующий запуск, необходимо записать Cpl = 0.
- Бит ES (2-й разряд). При чтении ES = 1 означает, что передача приостановлена. Бит ES устанавливается аппаратно, если блока коммуникационного порта получил сигнал об ошибке своего обращения в память – по несуществующему или недоступному адресу. Также бит ES можно установить программно. После обработки ошибки бит ES необходимо сбрасывать программно, так как он блокирует запись.
- Бит Clr (3-й разряд) показывает, что приёмная часть блока коммуникационного порта находится в состоянии очистки буфера данных. В этом состоянии принимаемые данные теряются – осуществляется только их подсчёт в регистре основного счётчика данных (MainCounter). Бит Clr сбрасывается автоматически при достижении MainCounter = 0.

Регистр масок запросов на прерывание InterruptMask

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																														MIE	MIC

Рисунок 4.11 - Формат регистра InterruptMask

Данный регистр позволяет запретить выдачу запроса на прерывание от блока коммуникационного порта по каждой из двух причин: завершение передачи и ошибка при обращении в память. Регистр содержит следующие поля:

- Бит MIC (0-й разряд) – маска запроса на прерывание по завершению передачи. MIC = 1 – запрос маскирован, MIC = 0 – запрос разрешён.
- Бит MIE (1-й разряд) – маска запроса на прерывание по ошибке или программной остановке. MIE = 1 – запрос маскирован, MIE = 0 – запрос разрешён.

Регистр состояния State

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
Зарезерви-ровано				FSM				Зарезерви-ровано			UTC					Зарезерви-ровано			ARC					Зарезерви-ровано			ADC				

Рисунок 4.12 - Формат регистра State

Регистр состояния State доступен только для чтения, показывает внутреннее состояние блока коммуникационного порта. Может быть использован для отладки и для обработки ошибочных ситуаций. Регистр содержит следующие поля:

- Поле ADC (разряды 4-0) – счётчик доступных данных. Показывает количество данных, имеющихся в буфере блока коммуникационного порта, но ещё не выданных на запись в память.
- Поле ARC (разряды 12-8) – счётчик активных запросов. Показывает количество данных, запрос на чтение которых уже отдан, но данные ещё не попали в буфер.
- Поле UTC (разряды 20-16) – счётчик незавершённых транзакций. Показывает количество транзакций записи на шине AXI3, для которых ещё не получено подтверждения.
- Поле FSM (разряды 28-24) – показывает текущее состояние конечного автомата:

  00h – Idle – бездействие,

  01h – ReadWrite – чтение и запись данных,

  03h – WriteOnly – чтение завершено, остаётся только запись,

  07h – UncompleteWrite – ожидание подтверждения последней записи,

  02h – Complete – запись завершена,

  06h – DataMiss – режим очистки буфера данных.

4.7.1.1 Прерывания от контроллера MDMAC

От контроллера MDMAC формируется один обобщённый запроса на прерывание в следующих случаях:

- По нормальному завершению работы контроллера ПДП, когда выполняется условие - установлен бит Cpl регистра DMARC_Control и сброшен бит маски MIC в регистре DMARC_InterruptMask (в т.ч. в конце процесса передачи).
- По ошибке доступа в память контроллера ПДП, когда выполняется одно из условий - либо установлен бит ES регистра DMATR_Control и сброшен бит маски MIE в регистре DMATR_InterruptMask, либо установлен бит ES регистра DMARC_Control и сброшен бит маски MIE в регистре DMARC_InterruptMask.

4.8 Внешний высокоскоростной линк EL

4.8.1 Основные характеристики

Внешний высокоскоростной линк обеспечивает дуплексную передачу типа точка-точка с суммарной пропускной способностью до 32 Гбит/с.

На физическом уровне передача осуществляется при помощи высокоскоростного физического интерфейса EL PHY, аналогичного PHY интерфейсам USB и PCIe. EL PHY имеет в своем составе 4 линии, каждая из которых имеет передающую и принимающую дифференциальные пары, работающие на частоте 2.5 ГГц или 5 ГГц. Интерфейс поддерживает только архитектуру с общим опорным тактовым сигналом (Common Refclk Architecture). Тактовый сигнал должен иметь частоту 100 МГц и скважность 40-60 %.

Линк EL обменивается данными с хостом по шине AXI Stream. Поэтому он способен передавать и принимать только данные, без их количества, адресов и других параметров. Таким образом, внешнее устройство, принимающее данные при помощи линка EL, должна знать их объем и адрес назначения.

Также в линке EL предусмотрены:

- режим пониженного потребления питания;
- исправление ошибок передачи и их подсчет;
- прерывания по некоторым событиям.

Линк EL входит в состав каждого кластера PC. Таким образом, микросхема содержит четыре независимых внешних высокоскоростных линка общей пропускной способностью 128 Гбит/с. Более подробно линк EL будет описан ниже, при этом используются следующие сокращения:

EL – External Link – внешний линк

DMAC (Direct Memory Access Controller) – контроллер прямого доступа в память

MAC – Media Access Controller – контроллер канального уровня связи

PHY – Physical layer – блок физического уровня связи

PIPE – Phy Interface to PCI Express – интерфейс между MAC и Phy

PLL – Phase lock loop – генератор тактового сигнала

4.8.2 Структура внешнего высокоскоростного линка EL

Внешний высокоскоростной линк EL представляет собой последовательно соединенные контроллер ELC (MAC) и физического интерфейса EL PHY (PHY). Каждый из

них требует собственную программную настройку по периферийной шине APB. Схема их соединения и их интерфейсов представлена ниже (см. рисунок 4.13).

Рисунок 4.13 – Структурная схема внешнего высокоскоростного линка EL

4.8.3 Контроллер линка EL

4.8.3.1 Общие сведения

Контроллер ELC, в дальнейшем называемый MAC, выполняет следующие функции:

- Управление сбросом, частотой, питанием Phy;
- Управление электрическими параметрами Phy;
- Формирование передаваемых сообщений – контрольных последовательностей, пакетов данных и подтверждений;
- Обработка входящих сообщений;
- Отслеживание и исправление ошибок передачи.

Рисунок 4.14 – Структура контроллера ELC (MAC)

4.8.3.2 Программно доступные регистры контроллера

Список регистров приведен в таблице ниже (см. таблицу 4.8). Доступ к ним осуществляется по периферийной шине APB. Регистры контроллера расположены в адресном пространстве периферийных устройств REG_CPU, начиная с адреса 000D_8000 hex.

Таблица 4.8 - Список регистров контроллера

Название регистра	Адрес (байтовый доступ)
CTRL_0	000D_8000 hex
CTRL_1	000D_8400 hex
STATE_0	000D_8404 hex
STATE_1	000D_8408 hex
PERIOD	000D_840C hex
L0_L1_PHYS	000D_8410 hex
L2_L3_PHYS	000D_8414 hex
L0_COEF	000D_8418 hex
L1_COEF	000D_841C hex
L2_COEF	000D_8420 hex
L3_COEF	000D_8424 hex
ERROR	000D_8428 hex

Регистр управления CTRL_0

Таблица 4.9 - Формат регистра управления CTRL_0

Биты	Тип доступа	Начальное значение	Описание
31	ЧТ/ЗП	1'h0	l3_swing: размах выходного сигнала линии №3: 1'h0 – полный размах; 1'h1 – половинный размах.
30	ЧТ/ЗП	1'h0	l2_swing: размах выходного сигнала линии №2: 1'h0 – полный размах; 1'h1 – половинный размах.
29	ЧТ/ЗП	1'h0	l1_swing: размах выходного сигнала линии №1: 1'h0 – полный размах; 1'h1 – половинный размах.
28	ЧТ/ЗП	1'h0	l0_swing: размах выходного сигнала линии №0: 1'h0 – полный размах; 1'h1 – половинный размах.
27:25	ЧТ/ЗП	3'h0	l3_margin: точный размах выходного сигнала линии №3*) 3'h0 – максимум.
24:22	ЧТ/ЗП	3'h0	l2_margin: точный размах выходного сигнала линии №2*) 3'h0 – максимум.
21:19	ЧТ/ЗП	3'h0	l1_margin: точный размах выходного сигнала линии №1*) 3'h0 – максимум.
18:16	ЧТ/ЗП	3'h0	l0_margin: точный размах выходного сигнала линии №0*) 3'h0 – максимум.
15:6			reserved:
5	ЧТ	1'h0	phy_status_int_sync: 1'h1 – указывает на окончание процесса смены режима работы. Представляет собой аналог параметра phy_status_int. Сбрасывается установкой параметра phy_status_int.
4	-	-	reserved:
3	ЧТ/ЗП	1'h0	rate: частота передачи данных: 1'h1 – 5 Гбит/с; 1'h0 – 2.5 Гбит/с. Окончание процесса переключения частоты отслеживается по параметру phy_status_int_sync.
2:1	ЧТ/ЗП	2'h2	power: режим питания PHY: 2'h2 – P1 – режим пониженного потребления; 2'h0 – P0 – рабочий режим. Окончание процесса переключения питания отслеживается по параметру phy_status_int_sync.
0	ЧТ/ЗП	1'h1	pipe_reset: сброс PHY и MAC: 1'h1 – PHY и MAC сброшены. PLL выключена. Программный доступ к остальным регистрам запрещен. Режим минимального потребления. 1'h0 – запись нуля снимает сброс и включает PHY и MAC. Окончание процесса включения отслеживается по параметру phy_status_int_sync.

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр управления CTRL_1

Таблица 4.10 - Формат регистра управления CTRL_1

Биты	Тип доступа	Начальное значение	Описание
31:8	-	-	reserved: резерв.
16	ЧТ	1'h0	link_ready: готовность линии к передаче и приему.
15:10	-		reserved: резерв
9:4	ЧТ/ЗП	6'h3F	int_mask: маски прерываний: 1'h0 – соответствующее прерывание не маскировано; 1'h1 – соответствующее прерывание маскировано.
3:1	ЧТ/ЗП	1'h0	error_ctrl: выбор счетчика ошибок, вызывающего прерывание error_int: 3'h0 – error_rx_phy; 3'h1 – error_crc; 3'h2 – error_length; 3'h3 – error_buf_full; 3'h4 – error_unknown_data; 3'h5 – error_receiving.
0	ЧТ/ЗП	1'h0	protocol_rst: сброс протокола работы. Снимается аппаратно на следующий после программной установки такт.

Регистр состояния STATE_0

Таблица 4.11 - Формат регистра состояния STATE_0

Биты	Тип доступа	Начальное значение	Описание
31:28	-	-	reserved: резерв
27:25	ЧТ	3'h0	l3_deskew: задержка данных на линии №3*)
24:22	ЧТ	3'h0	l2_deskew: задержка данных на линии №2*)
21:19	ЧТ	3'h0	l1_deskew: задержка данных на линии №1*)
18:16	ЧТ	3'h0	l0_deskew: задержка данных на линии №0*)
15:6	-	-	reserved: резерв
5	ЧТ/ЗП	1'h0	error_int: достижение счетчиком ошибок значения 5'h1F.
4	ЧТ/ЗП	1'h0	rxstatus_int: отличие RX_STATUS любой линии от нуля*)
3	ЧТ/ЗП	1'h0	rxelecidle_int: электрическое отключение входной линии.
2	ЧТ/ЗП	1'h0	incoming_int: прием пакета данных.
1	ЧТ/ЗП	1'h0	link_ready_int: готовность линии к передаче и приему данных.
0	ЧТ/ЗП	1'h0	phy_status_int: окончание процесса смены режима работы.

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр состояния STATE_1

Таблица 4.12 - Формат регистра состояния STATE_1

Биты	Тип доступа	Начальное значение	Описание
31:20	-	-	reserved: резерв
20:16	ЧТ/ЗП	5'h0	error_cntr: счетчик ошибок 8/10 или CRC; источник ошибок переключается параметром error_ctrl*)
15	ЧТ	1'h1	l3_rxelecidle: текущее значение электрического состояния линии №3
14	ЧТ	1'h1	l2_rxelecidle: текущее значение электрического состояния линии №2
13	ЧТ	1'h1	l1_rxelecidle: текущее значение электрического состояния линии №1
12	ЧТ	1'h1	l0_rxelecidle: текущее значение электрического состояния линии №0
11:9	ЧТ	3'h0	l3_rxstatus: текущее значение статуса приемника линии №3
8:6	ЧТ	3'h0	l2_rxstatus: текущее значение статуса приемника линии №2
5:3	ЧТ	3'h0	l1_rxstatus: текущее значение статуса приемника линии №1
2:0	ЧТ	3'h0	l0_rxstatus: текущее значение статуса приемника линии №0

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр задания периода PERIOD

Таблица 4.13 - Формат регистра состояния PERIOD

Биты	Тип доступа	Начальное значение	Описание
31:19	-	-	reserved: резерв
18:11	ЧТ/ЗП	8'h78	ack_period: период генерации ACK сообщений
10:0	ЧТ/ЗП	11'h4B0	skpos_period: период генерации SKPOS сообщений

Регистр управления линиями 0 и 1 L0_L1_PHYS

Таблица 4.14 - Формат регистра управления линиями 0 и 1 L0_L1_PHYS

Биты	Тип доступа	Начальное значение	Описание
31:25	-	-	reserved: резерв
24	ЧТ/ЗП	1'h0	l1_get_local_preset_coef: запрос расчета коэффициентов эквализации у линии №1*) . Снимается аппаратно на следующий после программной установки такт.
23:20	ЧТ/ЗП	4'h0	l1_tx_local_preset_index: индексы эквализации передатчика линии №1*)
19	ЧТ/ЗП	1'h0	l1_polarity: инвертирует входные данные в приемнике линии №1
18:17	ЧТ/ЗП	2'h1	l1_tx_deemph: коэффициенты преобразования выходного сигнала для передатчика линии №1
16	ЧТ/ЗП	1'h0	l1_tx_compliance: меняет порядок смены полярности выходного сигнала для линии №1
15:9	-	-	reserved: резерв
8	ЧТ/ЗП	1'h0	l0_get_local_preset_coef: запрос расчета коэффициентов эквализации у линии №0*). Снимается аппаратно на следующий после программной установки такт.
7:4	ЧТ/ЗП	4'h0	l0_tx_local_preset_index: индексы эквализации передатчика линии №0*)
3	ЧТ/ЗП	1'h0	l0_polarity: инвертирует входные данные в приемнике линии №0
2:1	ЧТ/ЗП	2'h1	l0_tx_deemph: коэффициенты преобразования выходного сигнала для передатчика линии №0
0	ЧТ/ЗП	1'h0	l0_tx_compliance: меняет порядок смены полярности выходного сигнала для линии №0

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр управления линиями 2 и 3 L2_L3_PHYS

Таблица 4.15 - Формат регистра управления линиями 2 и 3 L2_L3_PHYS

Биты	Тип доступа	Начальное значение	Описание
31:25	-	-	reserved: резерв
24	ЧТ/ЗП	1'h0	l3_get_local_preset_coef: запрос расчета коэффициентов эквализации у линии №3*) . Снимается аппаратно на следующий после программной установки такт.
23:20	ЧТ/ЗП	4'h0	l3_tx_local_preset_index: индексы эквализации передатчика линии №3*)
19	ЧТ/ЗП	1'h0	l3_polarity: инвертирует входные данные в приемнике линии №3
18:17	ЧТ/ЗП	2'h1	l3_tx_deemph: коэффициенты преобразования выходного сигнала для передатчика линии №3
16	ЧТ/ЗП	1'h0	l3_tx_compliance: меняет порядок смены полярности выходного сигнала для линии №3
15:9	-	-	reserved: резерв
8	ЧТ/ЗП	1'h0	l2_get_local_preset_coef: запрос расчета коэффициентов эквализации у линии №2*) . Снимается аппаратно на следующий после программной установки такт.
7:4	ЧТ/ЗП	4'h0	l2_tx_local_preset_index: индексы эквализации передатчика линии №2*)
3	ЧТ/ЗП	1'h0	l2_polarity: инвертирует входные данные в приемнике линии №2
2:1	ЧТ/ЗП	2'h1	l2_tx_deemph: коэффициенты преобразования выходного сигнала для передатчика линии №2
0	ЧТ/ЗП	1'h0	l2_tx_compliance: меняет порядок смены полярности выходного сигнала для линии №2

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр коэффициентов эквализации линии 0 L0_COEF

Таблица 4.16 - Формат регистра коэффициентов эквализации линии 0 L0_COEF

Биты	Тип доступа	Начальное значение	Описание
31:30			reserved: резерв
29:24	ЧТ	6'h0	l0_local_lf: величина lf от линии №0*)
23:18	ЧТ	6'h0	l0_local_fs: величина fs от линии №0*)
17:0	ЧТ	17'h0	l0_tx_local_tx_preset_coef: коэффициенты эквализации линии №0*)

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр коэффициентов эквализации линии 1 L1_COEF

Таблица 4.17 - Формат регистра коэффициентов эквализации линии 1 L1_COEF

Биты	**Тип доступа**	**Начальное значение**	**Описание**
31:30	-	-	reserved: резерв
29:24	ЧТ	6'h0	l1_local_lf: величина lf от линии №1*)
23:18	ЧТ	6'h0	l1_local_fs: величина fs от линии №1*)
17:0	ЧТ	17'h0	l1_tx_local_tx_preset_coef: коэффициенты эквализации линии №1*)

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр коэффициентов эквализации линии 2 L2_COEF

Таблица 4.18 - Формат регистра коэффициентов эквализации линии 2 L2_COEF

Биты	**Тип доступа**	**Начальное значение**	**Описание**
31:30			reserved: резерв
29:24	ЧТ	6'h0	l2_local_lf: величина lf от линии №2*)
23:18	ЧТ	6'h0	l2_local_fs: величина fs от линии №2*)
17:0	ЧТ	17'h0	l2_tx_local_tx_preset_coef: коэффициенты эквализации линии №2*)

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр коэффициентов эквализации линии 3 L3_COEF

Таблица 4.19 - Формат регистра коэффициентов эквализации линии 3 L3_COEF

Биты	Тип доступа	Начальное значение	Описание
31:30	-	-	reserved: резерв
29:24	ЧТ	6'h0	l3_local_lf: величина lf от линии №3*)
23:18	ЧТ	6'h0	l3_local_fs: величина fs от линии №3*)
17:0	ЧТ	17'h0	l3_tx_local_tx_preset_coef: коэффициенты эквализации линии №3*)

Примечание: *) Данный параметр нужен для исследования работы системы связи.

Регистр ошибок ERROR

Таблица 4.20 - Формат регистра ошибок ERROR

Биты	Тип доступа	Начальное значение	Описание
31:30	-	-	reserved: резерв
29:25	ЧТ/ЗП	5'h0	error_receiving: общее количество ошибок приема
24:20	ЧТ/ЗП	5'h0	error_unknown_data: количество ошибок неподходящих данных в принятом пакете
19:15	ЧТ/ЗП	5'h0	error_buf_full: количество ошибок переполнения приемного буфера
14:10	ЧТ/ЗП	5'h0	error_length: количество ошибок размера принятого пакета
9:5	ЧТ/ЗП	5'h0	error_crc: количество ошибок контрольной суммы
4:0	ЧТ/ЗП	5'h0	error_rx_phy: количество ошибок нераспознанных Phy данных

4.8.3.3 Стандартные алгоритмы работы с MAC

Включение контроллера:

После системного сброса тактирование большей части MAC отсутствует, поэтому программный доступ есть только к регистру CTRL_0. Чтение или запись в другие регистры MAC приведет к зависанию программной шины.

Снятие сброса с Phy и включение PLL:

- установить значение параметра pipe_reset в регистре CTRL_0 равным 1'h0;
- дождаться, пока контроллер установит параметр phy_status_int_sync в регистре CTRL_0 равным 1'h1;
- сбросить phy_status_int_sync записью 1'h1 в параметр phy_status_int в регистре STATE_0.

По стандарту PIPE протокола, после снятия pipe_reset приемник и передатчик Phy находятся в отключенном состоянии.

Установка электрических параметров выходного сигнала:

- установить необходимые значения l0_swing, l1_swing, l2_swing, l3_swing в регистре CTRL_0;
- установить необходимые значения l0_tx_deemph, l1_tx_deemph, l2_tx_deemph, l3_tx_deemph в регистрах L0_L1_PHYS и L2_L3_PHYS;
- Включение питания приемника и передатчика Phy:
- установить значение параметра power в регистре CTRL_0 равным 2'h0;
- дождаться, пока контроллер установит параметр phy_status_int_sync в регистре CTRL_0 равным 1'h1;
- сбросить phy_status_int_sync записью 1'h1 в параметр phy_status_int в регистре STATE_0.

По окончании включения питания (или изменения скорости передачи данных), MAC автоматически входит в режим инициализации связи. Режим заканчивается при готовности противоположного устройства. Поскольку это может занять значительный промежуток времени, возможно использование прерывания:

- при необходимости устанавливается маска прерывания int_mask [1] в регистре CTRL_1, равная 1'h0;

- ждём, пока контроллер установит параметр link_ready_int в регистре STATE_0 равным 1'h1;
- при использовании прерывания обратно устанавливается маска int_mask [1] в регистре CTRL_1, равная 1'h1.

По стандарту PIPE протокола, при старте используется минимальная скорость передачи и приема данных 2.5Гбит/с. Изменять ее можно когда угодно, но при этом следует убедиться в отсутствии передачи данных, иначе они могут быть утеряны. Рекомендуется перейти на 5Гбит/с сразу после включения питания.

Изменение скорости передачи данных:

- установить необходимое значение параметра rate в регистре CTRL_0;
- дождаться, пока контроллер установит параметр phy_status_int_sync в регистре CTRL_0 равным 1'h1;
- сбросить phy_status_int_sync записью 1'h1 в параметр phy_status_int в регистре STATE_0.

Выключение контроллера:

В рабочем режиме даже в отсутствии передаваемых данных, выходная линия по-прежнему работает с обычной частотой. Для снижения потребляемой мощности можно отключить питание приемника и передатчика Phy.

- установить значение параметра power в регистре CTRL_0 равным 2'h2;
- дождаться, пока контроллер установит параметр phy_status_int_sync в регистре CTRL_0 равным 1'h1;
- сбросить phy_status_int_sync записью 1'h1 в параметр phy_status_int в регистре STATE_0;

Для более глубокого энергосбережения следует сбросить Phy и находящуюся в нем PLL:

- установить параметр pipe_reset в регистре CTRL_0 равным 1'h1.

При этом будет отключено тактирование части Phy и MAC. Все регистры MAC, кроме CTRL_0 будут сброшены и станут недоступны для программного доступа.

Изменение полярности входных данных:

При разработке печатной платы можно, независимо для каждого линии, подключить положительный выход дифференциальной линии к отрицательному входу, а отрицательный –

к положительному. При этом необходимо установить соответствующие параметры l0_polarity, l1_polarity, l2_polarity, l3_polarity равными 1'h1. Сделать это рекомендуется сразу после снятия сброса и включения PLL.

4.8.3.4 Принципы работы MAC

Тактирование и сбросы

Рисунок 4.15 – Схема тактирования и сбросов EL

Контроллер работает от трех тактовых сигналов:

- AXI Stream CLK (800 МГц)
- APB CLK (200 МГц)
- PIPE CLK (125 МГц или 250 МГц)

Все переходы между тактовыми доменами сделаны асинхронными.

Сигнал сброса RESETN, поступающий от CPU, сразу синхронизируется на APB CLK, поэтому может быть асинхронным. Он асинхронно сбрасывает все тактовые домены контроллера.

Правила сбросов:

1. Установка RESETN не зависит от наличия тактовых сигналов. При этом сразу устанавливаются сбросы PIPE и AXI доменов.

2. Снятие RESETN также не зависит от наличия тактовых сигналов. PIPE сброс должен быть снят программно после настройки конфигурационных регистров Phy и MAC. Сброс с PIPE и AXI доменов MAC снимется автоматически после полного включения Phy.

Рисунок 4.16 – Схема тактирования и сбросов MAC

4.8.4 Режимы работы

Устройство может находиться в следующих состояниях (по убыванию приоритета):

- сброс;
- смена питания;
- смена частоты;
- инициализация связи;
- рабочий режим.

При запросе более высокоприоритетного состояния все, что ниже него, прерываются. Вызывать смену питания во время смены частоты запрещено. При первых трех состояниях выходная линия электрически выключена (PIPE_TX_ELECIDLE = 0), дифференциального сигнала нет.

Сброс

Условия входа:

- сброс RESETN или программная установка параметра pipe_reset.

Условия выхода:

- программное снятие параметра pipe_reset, выдача тактового сигнала от PHY.

Смена питания

Условия входа:

- несоответствие текущего режима питания значению, содержащемуся в программном регистре CTRL_0 в параметре power.

Условия выхода:

- сообщение от PHY о завершении смены режима питания (возможно соответствующее прерывание).

Смена частоты

Условия входа:

- несоответствие текущей частоты значению, содержащемуся в программном регистре CTRL_0 в параметре rate.

Условия выхода:

- сообщение от PHY о завершении смены частоты (возможно соответствующее прерывание).

При электрическом выключении передатчик не работает. При электрическом включении, работа передатчика основана на генерации, выборе и отправке сообщений.

При выполнении соответствующих условий выдается запрос на сообщение. По окончании текущего сообщения, в соответствии с приоритетом выбирается следующее. Ни одно сообщение не может быть прервано другим сообщением.

Инициализации связи

При входе в этот режим происходит сброс протокола.

Условия входа:

- PIPE_TX_ELECIDLE 1 -> 0;

- PIPE_RX_ELECIDLE 0 -> 1;
- получение EIEOS_0.

Условия выхода:

- получение и отправка EIEOS_1 (возможно соответствующее прерывание)

Возможные сообщения и их приоритет (максимальный сверху):

- SkpOS
- EIEOS_x

Рабочий режим

Условия входа. Возможные сообщения и их приоритет (максимальный сверху):

- SkpOS;
- ACK;
- повторный пакет;
- новый пакет.

В случае отсутствия сообщения передаются данные равные нулю.

4.8.5 Сообщения

Новая передача (новый пакет)

Генерируется при условиях:

- рабочий режим;
- наличие входных данных;
- наличие в памяти передатчика места для пакета.

При новой передаче из поступающих от DMAC данных формируется новый пакет. Весь пакет сохраняется в памяти передатчика.

Пакет состоит из:

- контрольный символ старта (SDP);
- идентификатор пакета (N* и N**);
- данные;
- контрольная сумма (C_0, C_1, C_2, C_3);
- контрольный символ окончания (END).

Формат пакета приведен на рисунке ниже.

Рисунок 4.17 – Формат пакета данных

Длина нового пакета переменная, выбирается на основе наличия входных данных, наличия пустого места в памяти передатчика.

Минимальная длина пакета: 1 слово данных (3 слова всего).

Максимальная длина пакета: 32 слова данных (34 слова всего) (ширина одного слова составляет 64 бита).

Приемник на основе идентификатора пакета и его длины рассчитывает идентификатор следующего ожидаемого пакета. Все неподходящие по номеру пакеты игнорируются.

<u>Повторная передача</u>

Генерируется при условии:

- рабочий режим;
- наличие не подтвержденных пакетов;
- получение 3 ACK сообщений с одинаковым идентификатором.

Гарантирует передачу пакетов в условиях их возможного повреждения и неприемом в связи с неготовностью приемника. При повторной передаче требуется точное повторение новых пакетов, поэтому они полностью сохраняются в памяти. Дополнительный бит в памяти указывает на первое/последнее слово и значение PIPE_TX_DATAK = ‘h55.

Поскольку новая передача менее приоритетна, при генерации повторной передаются все сохраненные пакеты, после чего следует пауза (ожидание 3 ACK) в течение которой новые данные могут передаваться.

Чем больше размер памяти в передатчике, тем реже можно посылать ACK сообщения. При возможности хранить 256 слов и периоде ACK в ‘h78 (примерно 1 ACK на 3.5 максимальных сообщения) наблюдается непрерывная передача и прием.

Т.к. одновременные новая и повторная передачи не возможны, память в передатчике используется или для записи или для чтения. Поэтому используется однопортовая память размером 256x65.

<u>ACK сообщение подтверждения</u>

Генерируется при условии:

- рабочий режим;
- достижение таймером запрограммированного значения.

Подтверждает получение пакетов. Несет в себе номер ожидаемого пакета, что сообщает передатчику об успешном приеме всех предыдущих.

Таймер запускается при входе в рабочий режим, останавливается при достижении запрограммированного значения и сбрасывается и запускается заново по отправке ACK.

Сообщение состоит из:

- контрольный символ старта (STP);
- информация о полученных пакетах (N_h, N_l).

Формат сообщения приведен ниже:

Рисунок 4.18 – Формат сообщения ACK

Идентификатор пакета – число, присваиваемое передатчиком каждому новому пакету.

Правила вычисления идентификатора:

$N(0) = 0,$

$N(i) = N(i-1) + L(i-1),$

где

$N(i)$ – идентификатор i-го пакета,

$L(i)$ – длина i-го пакета (количество слов данных).

Сбрасывается значение идентификатора при инициализации связи.

Предполагается использовать идентификатор шириной 9 бит. Тогда младшие 8 бит используются как адрес первого слова пакета в памяти передатчика. 9-ый бит нужен для отличия “старых” данных в памяти от “новых”.

Приемник всегда ожидает пакет с нужным идентификатором и принимает только его, тем самым сохраняя начальную последовательность данных. При получении пакета, приемник вычисляет идентификатор следующего ожидаемого пакета. Также это вычисленное значение посылается с сообщением ACK.

В передаче пакетов идентификатор равен:

N* = N [5:0];

N** = {N [8:6], N [2:0]}.

Такое усложнение введено для устранения одного из типов ошибок.

В передаче ACK сообщения идентификатор равен:

N_l = N [7:0];

N_h = N [8].

Поскольку ACK сообщения посылаются редко, с ними такая ошибка случиться не может.

<u>SkpOS сообщение</u>

SkpOS требуются для компенсации в PHY разности частот тактовых сигналов. Последовательность передается одновременно на каждую линию.

Генерируется при условии:

- достижение таймером запрограммированного значения.

Формат сообщения приведен ниже.

Рисунок 4.19 – Формат сообщения SKP

<u>EIEOS_x сообщение</u>

EIE - контрольный символ - используется для улучшенного распознавания противоположным PHY электрического включения линии и рекомендуется для работы на частоте 5Гбит/с.

EIEOS_0 сообщения требуются для оповещения другого устройства о включении. Последовательность передается одновременно на каждую линию.

Генерируется при условии:

- режим инициализации связи.

Формат сообщения приведен ниже.

Рисунок 4.20 – Формат сообщения EIEOS_0

EIEOS_1 сообщения требуются для сообщения другому устройству о получении EIEOS_0. Последовательность передается одновременно на каждую линию.

Генерируется при условии:

- режим инициализации связи;
- получение EIEOS_0.

При отправке 8 EIEOS_1 и получении хотя бы одного EIEOS_1 режим инициализации заканчивается и начинается рабочий.

Рисунок 4.21 – Формат сообщения EIEOS_1

4.8.6 Ошибка при приеме

<u>Ошибки при приеме пакета</u>

Если пакет не принимается, все ошибки игнорируются.

Пакет начинает приниматься, если на выходе Deskewer найдено стартовое слово с ожидаемым идентификатором. Если сдвиг по линиям превышает максимальный, стартовое слово не будет найдено и пакет будет проигнорирован.

Прием пакета прерывается в следующих случаях:

- ошибка 8/10 в пределах пакета;
- контрольный символ в поле для данных;
- линия перешла в отключенное состояние;

- длина пакета больше максимальной;
- память приемника заполнена.

Прием пакета успешно заканчивается, если найдено конечное слово с требуемой контрольной суммой.

При любой ошибке в приеме пакета, он будет проигнорирован. Все следующие за ним пакеты также не будут приняты, поскольку их идентификатор не совпадает с требуемым. ACK будет передавать один и тот же идентификатор. Через некоторое время передатчик заново пошлет все пакеты, начиная с первого неподтвержденного.

Ошибка при приеме ACK

При обнаружении ошибки в сообщении ACK, оно будет проигнорировано. Место в памяти передатчика не будет освобождено. В случае ее заполнения передатчик будет ожидать следующий ACK, и не будет осуществлять повторную передачу. Следующие принятые ACK освободят место в памяти (или, если они будут нести одинаковый идентификатор, вызовут повторную передачу).

Прием ACK сообщения прерывается в следующих случаях:

- дублированные параметры не совпадают;
- ошибка 8/10 в пределах сообщения;
- контрольный символ в поле для данных;
- неправильный формат (по контрольным символам);
- deskew переполнение.

4.8.7 CRC_8

CRC_8 – блоки в составе передатчика и приемника. CRC код генерируется с использованием следующих параметров: полином 8'h31, начальное значение 8'h00.

Рисунок 4.22 – Временная диаграмма блока генерации CRC

CRC код генерируется в передатчике независимо для каждого линии для нового пакета. CRC код генерируется в приемнике независимо для каждого линии для каждого получаемого пакета. При отличии рассчитанного значения от полученного с последним словом, устанавливается ошибка CRC, пакет игнорируется.

Рисунок 4.23 – Структура блока генерации CRC

4.8.8 Сброс протокола

Сброс протокола – установка всех регистров, отвечающих за поддержание протокола работы в начальное состояние – стирание всех данных, содержащихся в памяти передатчика и приемника, обнуление идентификаторов принятых и отправленных пакетов.

Сброс протокола осуществляется если:

- Программный запрос – установка параметра protocol_rst в управляющем регистре. Предусмотрен для исправления непредвиденных ошибок и зависаний схемы.
- Текущий режим – не рабочий. То есть, если в одной из микросхем был осуществлен сброс, смена режима питания или смена частоты, то находящиеся на этот момент в обоих MAC данные будут утеряны.

Таким образом, при смене режима следует удостовериться в отсутствии передачи данных с обеих микросхем. Потеря передаваемых данных при сбросе противоположной МС неизбежна, но это считается несущественным, поскольку целостность передачи все равно будет нарушена.

4.8.9 Блок Aligner

Aligner – блок в составе приемника.

Ширина принятых данных для одного линии составляет 2 байта – старший и младший. При приеме PHY может поставить младший на место старшего, а старший на место младшего следующего слова. Для дальнейшей работы декодера сообщений байты должны быть выровнены. Этим занимается блок Aligner.

Рисунок 4.24 – Временная диаграмма блока Aligner

Выравнивание основано на поиске контрольных символов COM и STP. Они передаются постоянно и должны быть на позиции младшего байта.

4.8.10 Блок DeSkewer

Deskewer – блок в составе приемника. После выравнивания (align) принятых данных, они могут быть сдвинуты друг относительно друга на разных линиях, как показано на рисунке ниже. По спецификации, этот сдвиг может составлять до 20 нс, то есть 2.5 или 5 тактов контроллера в зависимости от частоты работы.

Рисунок 4.25 – Разбежка данных при передаче

Перед выдачей данных контроллеру ПДП, они должны быть выровнены, сохранены в памяти приемника и проверены на ошибки. Выравнивание предлагается делать первым, поскольку это упростит остальные действия с данными.

Для этого существует отдельный блок DeSkewer.

На каждой линии меткой для выравнивания служит символ начала пакета и ожидаемый идентификатор {SDP, N}. Если по некоторым линиям, но не по всем, пришла метка, то их последующие данные сохраняются в ожидании прихода метки на оставшиеся линии. Работа Deskewer продемонстрирована на рисунке ниже.

Рисунок 4.26 – Временная диаграмма работы блока DeSkewer

Для сохранения данных в ожидании метки используются 4 регистровых файла размером 8x18.

PHY выдает также сигналы, по которым можно отследить некоторые ошибки – PIPE_RXELECIDLE и PIPE_RXSTATUS. Они не должны отставать или обгонять данные, поэтому тоже проходят через Deskewer и регистровые файлы. Поэтому регистровые файлы расширены до 8x20.

Работа Deskewer основана на заголовках входящих пакетов. При их отсутствии выравнивание не будет осуществляться. Поэтому сообщения ACK и EIEOS_x отслеживаются после Aligner, но до Deskewer независимо на каждой линии.

4.8.11 Типовые последовательности

Рисунок 4.27 – Временная диаграмма работы передатчика

На рисунке выше показано формирование выходных данных (pipe_txdata) в соответствии с запрашиваемыми сообщениями.

Штриховые зоны – сообщение запрошено, но не подтверждено.

Сплошные зоны – сообщение запрошено и подтверждено и именно оно посылается на линию передачи.

Рисунок 4.28– Упрощенная временная диаграмма передачи одного пакета

На рисунке выше приведена упрощенная временная диаграмма передачи одного пакета (с опущенными сообщениями SkpOS от обоих микросхем и опущенными ACK от Chip_0).

Рисунок 4.29 – Упрощенная временная диаграмма передачи пакетов

*ACK_Px значит, что все данные до Px успешно приняты.

На рисунке выше приведена упрощенная временная диаграмма передачи новых пакетов от Chip_0 к Chip_1 и соответствующих им ACK сообщений. Она приблизительно соответствует передаче пакетов максимальной длины и рекомендуемому периоду ACK. Точный результат можно получить в ходе моделирования.

Рисунок 4.30 – Упрощенная временная диаграмма передачи пакетов с ошибкой

На рисунке выше приведена временная диаграмма передачи пакетов от Chip_0 к Chip_1. Заштрихованная область – ошибка в передаче пакета P2. Затененные пакеты – игнорируемые приемником Chip_1. После отправки P8 память в передатчике заполнится. После 3го сообщения ACK_P2 начнется повторная передача (подчеркнутая), которая продолжается новыми пакетами.

4.9 Контроллер интерфейса с внешней динамической памятью EMI CPU

Контроллер интерфейса с внешней памятью EMI CPU предназначен для организации и управления обменом данными между внутренней памятью процессорных устройств кластера РС и микросхемами памяти, подключенными к соответствующему внешнему интерфейсу. Контроллер имеет следующие характеристики:

Тип поддерживаемой памяти DDR3-1600 SDRAM;

Частота шины 800 МГц;

Эффективная частота передачи данных 1600 МГц;

Разрядность внешней шины данных, бит – 32;

Количество сигналов выбора микросхемы (chip select) – 1;

Количество банков в микросхеме памяти – 8.

Начальная инициализация, управление контроллера EMI CPU осуществляется по шине AMBA APB.

Обращения к внешним микросхемам памяти DDR осуществляется по шине AMBA AXI по адресам, принадлежащим области памяти EMI_CPU и находящимся в диапазоне 0x0030_0000h – 0x3FFF_FFFF hex для CPU (байтовый доступ) и 0x000D_0000h – 0x0FFF_FFFF hex для NMPU (32-разрядный доступ) данного кластера. Декодирование адреса на шине AMBA AXI контроллером EMI CPU в сигнал выбора микросхем памяти (XCSi), адрес столбца, адрес строки и номер банка внешней памяти происходит автоматически. Параметры декодирования настраиваются в конфигурационных регистрах EMI.

Перед тем, как производить обращения к внешней памяти, необходимо инициализировать контроллер EMI. Процесс инициализации состоит из двух частей: инициализация DLL и программная настройка контроллера. Инициализация DLL осуществляется автоматически после системного сброса. Программная настройка контроллера включает следующие действия:

- настройку конфигурационных регистров контроллера EMI CPU;
- запись управляющего регистра микросхемы внешней памяти;
- перевод контроллера EMI CPU в активное состояние.

4.9.1 Программно доступные регистры интерфейса с внешней динамической памятью EMI CPU

Список регистров интерфейса с внешней динамической памятью EMI CPU приведен в таблице ниже (Таблица 4.21). Доступ к ним осуществляется по периферийной шине APB. Регистры контроллера расположены в адресном пространстве периферийных устройств REG_CPU, начиная с адреса 000C_0000 hex.

Таблица 4.21 – Программно доступные регистры контроллера EMI CPU

Название регистра	Адрес (байтовый доступ)	Внутренний адрес регистра	Тип доступа	Описание
DENALI_CTL_00	0x000C_0000 hex	0x00	ЧТ ЧТ/ЗП ЧТ/ЗП	VERSION DRAM_CLASS START
DENALI_CTL_01	0x000C_0004 hex	0x01	ЧТ ЧТ ЧТ ЧТ	READ_DATA_FIFO_DEPTH MAX_CS_REG MAX_COL_REG MAX_ROW_REG
DENALI_CTL_02	0x000C_0008 hex	0x02	ЧТ ЧТ ЧТ ЧТ	ASYNC_CDC_STAGES WRITE_DATA_FIFO_PTR_WIDTH WRITE_DATA_FIFO_DEPTH READ_DATA_FIFO_PTR_WIDTH
DENALI_CTL_03	0x000C_000c hex	0x03	ЧТ ЧТ ЧТ ЧТ	AXI0_TRANS_WFIFO_LOG2_DEPTH AXI0_WRFIFO_LOG2_DEPTH AXI0_RDFIFO_LOG2_DEPTH AXI0_CMDFIFO_LOG2_DEPTH
DENALI_CTL_04	0x000C_0010 hex	0x04	ЧТ/ЗП ЧТ	TINIT AXI0_WRCMD_PROC_FIFO_LOG2_DEPTH
DENALI_CTL_05	0x000C_0014 hex	0x05	ЧТ/ЗП	TRST_PWRON
DENALI_CTL_06	0x000C_0018 hex	0x06	ЧТ/ЗП	CKE_INACTIVE
DENALI_CTL_07	0x000C_001c hex	0x07	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TBST_INT_INTERVAL ADDITIVE_LAT WRLAT CASLAT_LIN
DENALI_CTL_08	0x000C_0020 hex	0x08	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TRAS_MIN TRC TRRD TCCD
DENALI_CTL_09	0x000C_0024 hex	0x09	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TRTP TFAW TRP TWTR
DENALI_CTL_10	0x000C_0028 hex	0x0a	ЧТ/ЗП ЧТ/ЗП	TMOD TMRD
DENALI_CTL_11	0x000C_002c hex	0x0b	ЧТ/ЗП ЧТ/ЗП	TCKE TRAS_MAX
DENALI_CTL_12	0x000C_0030 hex	0x0c	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TWR TRCD WRITEINTERP TCKESR
DENALI_CTL_13	0x000C_0034 hex	0x0d	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TDAL TRAS_LOCKOUT CONCURRENTAP AP
DENALI_CTL_14	0x000C_0038 hex	0x0e	ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	AREFRESH REG_DIMM_ENABLE TRP_AB BSTLEN[2)]
DENALI_CTL_15	0x000C_003c hex	0x0f	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TRFC TREF_ENABLE RESERVED
DENALI_CTL_16	0x000C_0040 hex	0x10	ЧТ/ЗП	TREF

Продолжение таблицы 4.21

Название регистра	Адрес (байтовый доступ)	Внутренний адрес регистра	Тип доступа	Описание
DENALI_CTL_17	0x000C_0044 hex	0x11	ЧТ/ЗП ЧТ/ЗП	TXPDLL TPDEX
DENALI_CTL_18	0x000C_0048 hex	0x12	ЧТ/ЗП ЧТ/ЗП	TXSNR TXSR
DENALI_CTL_19	0x000C_004c hex	0x13	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	CKE_DELAY ENABLE_QUICK_SREFRESH RESERVED PWRUP_SREFRESH_EXIT
DENALI_CTL_20	0x000C_0050 hex	0x14	ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	LP_CMD CKSRX CKSRE LOWPOWER_REFRESH_ENABLE
DENALI_CTL_21	0x000C_0054 hex	0x15	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	LPI_TIMER_WAKEUP LPI_SR_MCCLK_GATE_WAKEUP LPI_SR_ WAKEUP LPI_PD_ WAKEUP
DENALI_CTL_22	0x000C_0058 hex	0x16	ЧТ/ЗП ЧТ/ЗП	LPI_TIMER_COUNT LPI_ WAKEUP_EN
DENALI_CTL_23	0x000C_005c hex	0x17	ЧТ ЧТ/ЗП ЧТ/ЗП	LP_STATE TDFI_LP_RESP LPI_WAKEUP_TIMEOUT
DENALI_CTL_24	0x000C_0060 hex	0x18	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ	LP_AUTO_MEM_GATE_EN LP_AUTO_EXIT_EN LP_AUTO_ENTRY_EN LP_ARB_STATE
DENALI_CTL_25	0x000C_0064 hex	0x19	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	LP_AUTO_SR_MC_GATE_IDLE LP_AUTO_SR_IDLE LP_AUTO_PD_IDLE
DENALI_CTL_26	0x000C_0068 hex	0x1a	ЧТ/ЗП	RESERVED
DENALI_CTL_27	0x000C_006c hex	0x1b	ЧТ/ЗП	WRITE_MODEREG[1)]
DENALI_CTL_28	0x000C_0070 hex	0x1c	ЧТ/ЗП ЧТ	MR0_DATA_0 MRW_STATUS
DENALI_CTL_29	0x000C_0074 hex	0x1d	ЧТ/ЗП ЧТ/ЗП	MR2_DATA_0 MR1_DATA_0
DENALI_CTL_30	0x000C_0078 hex	0x1e	ЧТ/ЗП ЧТ/ЗП	MR3_DATA_0 MRSINGLE_DATA_0
DENALI_CTL_31	0x000C_007c hex	0x1f	ЧТ/ЗП ЧТ/ЗП	ZQINIT[2)] LONG_COUNT_MASK
DENALI_CTL_32	0x000C_0080 hex	0x20	ЧТ/ЗП ЧТ/ЗП	ZQCS ZQCL
DENALI_CTL_33	0x000C_0084 hex	0x21	ЧТ/ЗП ЗП	ZQ_ON_SREF_EXIT ZQ_REQ
DENALI_CTL_34	0x000C_0088 hex	0x22	ЧТ/ЗП	ZQ_INTERVAL
DENALI_CTL_35	0x000C_008c hex	0x23	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ	COL_DIFF ROW_DIFF RESERVED ZQ_IN_PROGRESS

Продолжение таблицы 4.21

Название регистра	Адрес (байтовый доступ)	Внутренний адрес регистра	Тип доступа	Описание
DENALI_CTL_36	0x000C_0090 hex	0x24	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	AGE_COUNT APREBIT[2)] BANK_ADDR_INTLV_EN BANK_START_BIT
DENALI_CTL_37	0x000C_00904 hex	0x25	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	BANK_SPLIT_EN RESERVED ADDR_CMP_EN COMMAND_AGE_COUNT
DENALI_CTL_38	0x000C_0098 hex	0x26	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	RW_SAME_PAGE_EN RW_SAME_EN PRIORITY_EN PLACEMENT_EN
DENALI_CTL_39	0x000C_009c hex	0x27	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	DISABLE_RD_INTERLEAVE SWAP_EN NUM_Q_ENTRIES_ACT_DISABLE DISABLE_RW_GRUP_W_BNK_CONFLICT
DENALI_CTL_40	0x000C_00a0 hex	0x28	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	Q_FULLNESS REDUC BURST_ON_FLY_BIT INHIBIT_DRAM_CMD
DENALI_CTL_41	0x000C_00a4 hex	0x29	ЧТ/ЗП ЗП ЧТ ЧТ/ЗП	CTRLUPD_REQ_PER_AREF_EN CTRLUPD_REQ CONTROLLER_BUSY IN_ORDER_ACCEPT
DENALI_CTL_42	0x000C_00a8 hex	0x2a	ЧТ ЧТ	DFI_ERROR_INFO DFI_ERROR
DENALI_CTL_43	0x000C_00ac hex	0x2b	ЧТ ЧТ/ЗП	INT_STATUS RESERVED[1)]
DENALI_CTL_44	0x000C_00b0 hex	0x2c	ЗП	INT_ACK
DENALI_CTL_45	0x000C_00b4 hex	0x2d	ЧТ/ЗП	INT_MASK
DENALI_CTL_46	0x000C_00b8 hex	0x2e	ЧТ	OUT_OF_RANGE_ADDR
DENALI_CTL_47	0x000C_00bc hex	0x2f	ЧТ ЧТ ЧТ	OUT_OF_RANGE_SOURCE_ID OUT_OF_RANGE_TYPE OUT_OF_RANGE_LENGTH
DENALI_CTL_48	0x000C_00c0 hex	0x30	ЧТ	PORT_CMD_ERROR_ADDR
DENALI_CTL_49	0x000C_00c4 hex	0x31	ЧТ/ЗП ЧТ ЧТ	TODTL_2CMD PORT_CMD_ERROR_TYPE PORT_CMD_ERROR_ID
DENALI_CTL_50	0x000C_00c8 hex	0x32	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	R2R_SAMECS_DLY RESERVED ODT_EN TODTH_WR
DENALI_CTL_51	0x000C_00cc hex	0x33	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	SW_LEVENING_MODE W2W_SAMECS_DLY W2R_SAMECS_DLY R2W_SAMECS_DLY[2)]

Продолжение таблицы 4.21

Название регистра	Адрес (байтовый доступ)	Внутренний адрес регистра	Тип доступа	Описание
DENALI_CTL_52	0x000C_00d0 hex	0x34	ЧТ ЗП ЗП ЗП	SWLVL_OP_DONE SWLVL_EXIT SWLVL_START SWLVL_LOAD
DENALI_CTL_53	0x000C_00d4 hex	0x35	ЧТ ЧТ ЧТ ЧТ	SWLVL_RESP_3 SWLVL_RESP_2 SWLVL_RESP_1 SWLVL_RESP_0
DENALI_CTL_54	0x000C_00d8 hex	0x36	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЗП	WLMRD WLDQSEN WRLVL_CS WRLVL_REQ
DENALI_CTL_55	0x000C_00dc hex	0x37	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	DFI_PHY_WRLVL_MODE WRLVL_INTERVAL WRLVL_EN
DENALI_CTL_56	0x000C_00e0 hex	0x38	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	WRLVL_AREF_EN WRLVL_RESP_MASK WRLVL_ON_SREF_EXIT WRLVL_PERIODIC
DENALI_CTL_57	0x000C_00e4 hex	0x39	ЧТ/ЗП ЗП ЗП ЧТ	RDLVL_CS RDLVL_GATE_REQ RDLVL_REQ WRLVL_ERROR_STATUS
DENALI_CTL_58	0x000C_00e8 hex	0x3a	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	DFI_PHY_RDLVL_GATE_MODE DFI_PHY_RDLVL_ MODE DFI_GATE_SEQ_EN RDLVL_ SEQ_EN
DENALI_CTL_59	0x000C_00ec hex	0x3b	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	RDLVL_GATE_ON_SREF_EXIT RDLVL_GATE_PERIODIC RDLVL_ ON_SREF_EXIT RDLVL_ PERIODIC
DENALI_CTL_60	0x000C_00f0 hex	0x3c	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	AXI0_W_PRIORITY AXI0_R_PRIORITY AXI0_FIXED_PORT_PRIORITY_ENABLE RDLVL_AREF_EN
DENALI_CTL_61	0x000C_00f4 hex	0x3d	ЧТ ЧТ ЧТ/ЗП	MEM_RST_VALID CKE_STATUS AXI0_FIFO_TYPE_REG
DENALI_CTL_62	0x000C_00f8 hex	0x3e	ЧТ ЧТ/ЗП ЧТ/ЗП	TDFI_PHY_WRLAT DLL_RST_ADJ_DLY DLL_RST_DELAY
DENALI_CTL_63	0x000C_00fc hex	0x3f	ЧТ/ЗП ЧТ ЧТ/ЗП ЧТ	DRAM_CLK_DISABLE TDFI_RDDATA_EN TDFI_PHY_RDLAT[2)] UPDATE_ERROR_STATUS

Продолжение таблицы 4.21

Название регистра	Адрес (байтовый доступ)	Внутренний адрес регистра	Тип доступа	Описание
DENALI_CTL_64	0x000C_0100 hex	0x40	ЧТ/ЗП ЧТ/ЗП	TDFI_CTRLUPD_MAX TDFI_CTRLUPD_MIN
DENALI_CTL_65	0x000C_0104 hex	0x41	ЧТ/ЗП ЧТ/ЗП	TDFI_PHYUPD_TYPE1 TDFI_PHYUPD_TYPE0
DENALI_CTL_66	0x000C_0108 hex	0x42	ЧТ/ЗП ЧТ/ЗП	TDFI_PHYUPD_TYPE3 TDFI_PHYUPD_TYPE2
DENALI_CTL_67	0x000C_010c hex	0x43	ЧТ/ЗП	TDFI_PHYUPD_RESP
DENALI_CTL_68	0x000C_0110 hex	0x44	ЧТ/ЗП	TDFI_CTRLUPD_INTERVAL
DENALI_CTL_69	0x000C_0114 hex	0x45	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TDFI_DRAM_CLK_DISABLE TDFI_CTRL_DELAY[2)] WRLAT_ADJ RDLAT_ADJ
DENALI_CTL_70	0x000C_0118 hex	0x46	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TDFI_WRLVL_WW TDFI_WRLVL_EN TDFI_DRAM_CLK_ENABLE
DENALI_CTL_71	0x000C_011c hex	0x47	ЧТ/ЗП	TDFI_WRLVL_RESP
DENALI_CTL_72	0x000C_0120 hex	0x48	ЧТ/ЗП	TDFI_WRLVL_MAX
DENALI_CTL_73	0x000C_0124 hex	0x49	ЧТ/ЗП ЧТ/ЗП	TDFI_RDLVL_RR TDFI_RDLVL_EN
DENALI_CTL_74	0x000C_0128 hex	0x4a	ЧТ/ЗП	TDFI_RDLVL_RESP
DENALI_CTL_75	0x000C_012c hex	0x4b	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	RDLVL_GATE_EN RDLVL_EN RDLVL_RESP_MASK
DENALI_CTL_76	0x000C_0130 hex	0x4c	ЧТ/ЗП	TDFI_RDLVL_MAX
DENALI_CTL_77	0x000C_0134 hex	0x4d	ЧТ/ЗП ЧТ	RDLVL_INTERVAL RDLVL_ERROR_STATUS
DENALI_CTL_78	0x000C_0138 hex	0x4e	ЧТ/ЗП ЧТ/ЗП ЧТ/ЗП	TDFI_RDCSLAT TDFI_PHY_WRDATA TDFI_GATE_INTERVAL
DENALI_CTL_79	0x000C_013c hex	0x4f	ЧТ/ЗП	TDFI_WRCSLAT

Примечание: 1) Эти параметры могут быть считаны или записаны программно, но также могут быть изменены внутренней логикой.
2) Эти параметры будут установлены в 0x0 по системному сбросу, но будут изменены на не нулевые значения в следующем после системного сброса такте.

Регистр DENALI_CTL_00

Таблица 4.22 – Описание полей регистра DENALI_CTL_00

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
VERSION	31:16	0x2042	0x2042	Содержит номер версии контроллера. Только для чтения.
DRAM_CLASS	11:8	0x0	0x0-0xf	Определяет класс микросхем DRAM, подключённых к контроллеру.
START	0	0x0	0x0-0x1	Инициирует выполнение команд контроллером. Установите в 1 для инициации.

Регистр DENALI_CTL_01

Таблица 4.23 – Описание полей регистра DENALI_CTL_01

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
READ_DATA_ FIFO_DEPTH	31:24	0x00	0x0-0xff	Отражает глубину очереди чтения контроллера. Только для чтения.
MAX_CS_REG	17:16	0x2	0x0-0x2	Содержит максимальное число доступных CS. Только для чтения.
MAX_COL_REG	11:8	0xa	0x0-0xa	Содержит максимальную ширину адреса столбцов в микросхемах DRAM. Только для чтения.
MAX_ROW_REG	3:0	0xf	0x0-0xf	Содержит максимальную ширину адресной шины памяти. Только для чтения.

Регистр DENALI_CTL_02

Таблица 4.24 – Описание полей регистра DENALI_CTL_02

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
ASYNC_CDC_ STAGES	31:24	0x00	0x0-0xff	Показывает число задержек синхронизатора, определенных для асинхронного пересечения границ. Только для чтения.
WRITE_DATA_ FIFO_PTR_WIDTH	23:16	0x00	0x0-0xff	Показывает разрядность указателя глубины очереди записи данных ядра контроллера. Только для чтения.
WRITE_DATA_ FIFO_DEPTH	15:8	0x00	0x0-0xff	Показывает глубину очереди записи данных ядра контроллера. Только для чтения.
READ_DATA_ FIFO_PTR_WIDTH	7:0	0x00	0x0-0xff	Показывает разрядность указателя глубины очереди чтения данных ядра контроллера. Только для чтения.

Регистр DENALI_CTL_03

Таблица 4.25 – Описание полей регистра DENALI_CTL_03

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
AXI0_TRANS_ WFIFO_LOG2_DEPTH	31:24	0x00	0x0-0xff	Показывает глубину FIFO данных для записи порта AXI 0. Значение является log2 глубины. Только для чтения.
AXI0_WRFIFO_ LOG2_DEPTH	23:16	0x00	0x0-0xff	Показывает глубину FIFO записи данных порта AXI 0. Значение является log2 глубины. Только для чтения.
AXI0_RDFIFO_ LOG2_DEPTH	15:8	0x00	0x0-0xff	Показывает глубину FIFO чтения данных порта AXI 0. Значение является log2 глубины. Только для чтения.
AXI0_CMDFIFO_ LOG2_DEPTH	7:0	0x00	0x0-0xff	Показывает глубину FIFO команд порта AXI 0. Значение является log2 глубины. Только для чтения.

Регистр DENALI_CTL_04

Таблица 4.26 – Описание полей регистра DENALI_CTL_04

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TINIT	31:8	0x000000	0x0-0xffffff	Время инициализации DRAM в тактах.
AXI0_WRCMD_ PROC_FIFO_LOG2_ DEPTH	7:0	0x00	0x0-0xff	Показывает глубину FIFO команд записи порта AXI 0. Значение является log2 глубины. Только для чтения.

Регистр DENALI_CTL_05

Таблица 4.26 – Описание полей регистра DENALI_CTL_05

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TRST_PWRON	31:0	0x00000000	0x0-0xffffffff	Продолжительность сброса памяти во время инициализации при включении питания.

Регистр DENALI_CTL_06

Таблица 4.27 – Описание полей регистра DENALI_CTL_06

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
CKE_INACTIVE	31:0	0x00000000	0x0-0xffffffff	Задержка в тактах между снятием сброса и подачей активного сигнала CKE.

Регистр DENALI_CTL_07

Таблица 4.28 – Описание полей регистра DENALI_CTL_07

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:27	-	0x0-0xf	Зарезервировано для дальнейшего использования.
TBST_INT_ INTERVAL	26:24	0x0	0x1-0x7	Максимальное время в тактах, после которого берст прервётся.
RESERVED	23:21	-	0x0-0x7	Зарезервировано для дальнейшего использования.
ADDITIVE_LAT	20:16	0x00	0x0-0x1f	Суммарная задержка в тактах при работе с DRAM
RESERVED	15:11	-	0x0-0xf	Зарезервировано для дальнейшего использования.
WRLAT	12:8	0x00	0x0-0x1f	Значение WRLAT в тактах.
RESERVED	7:6	-	0x0-0x3	Зарезервировано для дальнейшего использования.
CASLAT_LIN	5:0	0x00	0x0-0x3f	Устанавливает задержку в тактах от отправки команды чтения до получения данных из/в контроллер. Бит (0) - это инкремент на полтакта и старшие биты определяют CAS задержку для контроллера.

Регистр DENALI_CTL_08

Таблица 4.29 – Описание полей регистра DENALI_CTL_08

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TRAS_MIN	31:24	0x00	0x0-0xff	Значение TRAS_MIN в тактах.
TRC	23:16	0x00	0x0-0xff	Значение TRC в тактах.
TRRD	15:8	0x00	0x0-0xff	Значение TRRD в тактах.
RESERVED	7:5	-	0x0-0x7	Зарезервировано для дальнейшего использования.
TCCD	4:0	0x00	0x0-0x1f	Задержка в циклах между двумя активными сигналами CAS.

Регистр DENALI_CTL_09

Таблица 4.30 – Описание полей регистра DENALI_CTL_09

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TRTP	31:24	0x00	0x0-0xff	Значение TRTP в тактах.
TFAW	23:16	0x00	0x0-0xff	Значение TFAW в тактах.
TRP	15:8	0x00	0x0-0xff	Значение TRP в тактах.
RESERVED	7:6	-	0x0-0x3	Зарезервировано для дальнейшего использования.
TWTR	5:0	0x00	0x0-0x3f	Значение TWTR в тактах.

Регистр DENALI_CTL_10

Таблица 4.31 – Описание полей регистра DENALI_CTL_10

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:16	-	0x0-0xffff	Зарезервировано для дальнейшего использования.
TMOD	15:8	0x00	0x0-0xff	Число тактов после MRS команды и перед любой другой командой.
TMRD	7:0	0x00	0x0-0xff	Значение TMRD в тактах.

Регистр DENALI_CTL_11

Таблица 4.32 – Описание полей регистра DENALI_CTL_11

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:27	-	0x0-0xf	Зарезервировано для дальнейшего использования.
TCKE	26:24	0x0	0x0-0x7	Минимальная ширина полки CKE.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
TRAS_MAX	16:0	0x00000	0x0-0x1ffff	Значение TRAS_MAX в тактах.

Регистр DENALI_CTL_12

Таблица 4.33 – Описание полей регистра DENALI_CTL_12

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:30	-	0x0-0xf	Зарезервировано для дальнейшего использования.
TWR	29:24	0x00	0x0-0x3f	Значение TWR в тактах.
TRCD	23:16	0x00	0x0-0xff	Значение TRCD в тактах.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
WRITEINTERP	8	0x0	0x0-0x1	Разрешает контроллеру прерывать берстовую запись командой чтения. Установить в 1 для разрешения прерывания.
TCKESR	7:0	0x00	0x0-0xff	Минимальная ширина полки CKE в режиме self-refresh.

Регистр DENALI_CTL_13

Таблица 4.34 – Описание полей регистра DENALI_CTL_13

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TDAL	31:24	0x00	0x0-0xff	Значение TDAL в тактах.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
TRAS_LOCKOUT	16	0x0	0x0-0x1	Разрешает контроллеру выполнять команды auto pre-charge до окончания параметра TRAS_MIN. Установить в 1 для разрешения.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
CONCURRENTAP	8	0x0	0x0-0x1	Разрешает контроллеру выдавать команды другим банкам, пока банк в состоянии auto pre-charge. Установить в 1 для разрешения.
AP	7:0	0x00	0x0-0xff	Разрешение режима auto pre-charge контроллера. Установить в 1 для разрешения.

Регистр DENALI_CTL_14

Таблица 4.35 – Описание полей регистра DENALI_CTL_14

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
AREFRESH	24	0x0	0x0-0x1	Инициирует auto pre-charge в конце границы текущего берста. Установить в 1 для запуска. Только для записи.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
REG_DIMM_ ENABLE	16	0x0	0x0-0x1	Разрешение конвейерных DIMM операций контроллера. Установить в 1 для разрешения.
TRP_AB	15:8	0x00	0x0-0xff	Значение TRP_AB в тактах для всех банков памяти.
RESERVED	7:3	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
BSTLEN	2:0	0x2	0x0-0x7	Закодированная длина берста, отправляемого в микросхемы DRAM во время инициализации. Установить в 1 для BL2, в 2 для BL4, в 3 для BL8.

Регистр DENALI_CTL_15

Таблица 4.36 – Описание полей регистра DENALI_CTL_15

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:26	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
TRFC	25:16	0x000	0x0-0x3ff	Значение TRFC в тактах.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
TREF_ENABLE	8	0x0	0x0-0x1	Выдача команд auto pre-charge в микросхемы DRAM на интервале, определенном TREF параметром. Установить в 1 для разрешения.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RESERVED	0	-	0x0-0x1	Зарезервировано для дальнейшего использования. См. конфигурационные файлы для начального программирования.

Регистр DENALI_CTL_16

Таблица 4.37 – Описание полей регистра DENALI_CTL_16

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:14	-	0x0-0x3ffff	Зарезервировано для дальнейшего использования.
TREF	13:0	0x0000	0x0-0x3fff	Значение TREF в тактах.

Регистр DENALI_CTL_17

Таблица 4.38 – Описание полей регистра DENALI_CTL_17

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TXPDLL	31:16	0x0000	0x0-0xffff	Значение TXPDLL в тактах.
TPDEX	15:0	0x0000	0x0-0xffff	Значение TPDEX в тактах.

Регистр DENALI_CTL_18

Таблица 4.39 – Описание полей регистра DENALI_CTL_18

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TXSNR	31:16	0x0000	0x0-0xffff	Значение TXSNR в тактах.
TXSR	15:0	0x0000	0x0-0xffff	Значение TXSR в тактах.

Регистр DENALI_CTL_19

Таблица 4.40 – Описание полей регистра DENALI_CTL_19

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:27	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
CKE_DELAY	26:24	0x0	0x0-0x7	Дополнительные такты задержки CKE для отчета состояния.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
ENABLE_QUICK_ SREFRESH	16	0x0	0x0-0x1	Разрешение пользователю прерывать инициализацию памяти для входа в режим self-refresh. Установить в 1 для разрешения прерывания.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RESERVED	8	-	0x0-0x1	Зарезервировано для дальнейшего использования. См. конфигурационные файлы для начального программирования
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
PWRUP_ SREFRESH_EXIT	0	0x0	0x0-0x1	Разрешение войти в режим self-refresh вместо полной инициализации памяти при включении питания. Установить в 1 для разрешения.

Регистр DENALI_CTL_20

Таблица 4.41 – Описание полей регистра DENALI_CTL_20

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
LP_CMD	31:24	0x00	0x0-0xff	Интерфейс программных запросов команд управления питанием. Бит (0) контролирует выход, бит (1) – вход, биты (4:2) определяют состояние питания, бит (5) контролирует отключение синхросигнала памяти, бит (6) – отключение синхросигнала контроллера, бит (7) – блокировку. Только запись.
CKSRX	23:16	0x00	0x0-0xff	Задержка стабилизации синхросигнала при выходе из режима self-refresh.
CKSRE	15:8	0x00	0x0-0xff	Удержание тактового сигнала при входе в режим self-refresh.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
LOWPOWER_ REFRESH_ENABLE	0	0x0	0x0-0x1	Разрешение регенерации памяти в режиме пониженного потребления. Бит (0) контролирует cs0, бит (1) - cs1 и т.д. Установите каждый бит в 1 для запрета.

Регистр DENALI_CTL_21

Таблица 4.42 – Описание полей регистра DENALI_CTL_21

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:28	-	0x0-0xf	Зарезервировано для дальнейшего использования.
LPI_TIMER_ WAKEUP	27:24	0x0	0x0-0xf	Определяет значение временного параметра LP_WAKEUP, когда таймер интерфейса управления питанием досчитает до 0.
RESERVED	23:20	-	0x0-0xf	Зарезервировано для дальнейшего использования.
LPI_SR_MCCLK_ GATE_WAKEUP	19:16	0x0	0x0-0xf	Определяет значение временного параметра LP_WAKEUP, когда память в режиме self-refresh и у контроллера отключён тактовый сигнал.
RESERVED	15:12	-	0x0-0xf	Зарезервировано для дальнейшего использования.
LPI_SR_ WAKEUP	11:8	0x0	0x0-0xf	Определяет значение временного параметра LP_WAKEUP, когда память в режиме self-refresh.
RESERVED	7:4	-	0x0-0xf	Зарезервировано для дальнейшего использования.
LPI_PD_ WAKEUP	3:0	0x0	0x0-0xf	Определяет значение временного параметра LP_WAKEUP, когда память в режиме power-down.

Регистр DENALI_CTL_22

Таблица 4.43 – Описание полей регистра DENALI_CTL_22

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	32:20	-	0x0-0x1fff	Зарезервировано для дальнейшего использования.
LPI_TIMER_ COUNT	19:8	0x000	0x0-0xfff	Задаёт интервал счёта для таймера интерфейса управления питанием
RESERVED	7:5	-	0x0-0x7	Зарезервировано для дальнейшего использования.
LPI_ WAKEUP_EN	4:0	0x00	0x0-0x1f	Разрешает выход из состояния малого потребления. Бит (0) разрешает выход из состояния power-down, бит (1) – из состояния self-refresh, бит (2) – из состояния self-refresh с отключённым тактовым сигналом контроллера, бит (3) – резерв, бит (4) – разрешает работу таймеру интерфейса управления питанием. Установите каждый бит в 1 для разрешения.

Регистр DENALI_CTL_23

Таблица 4.44 – Описание полей регистра DENALI_CTL_23

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:30	-	0x0-0x3	Зарезервировано для дальнейшего использования.
LP_ STATE	29:24	0x20	0x0-0x3f	Состояние параметров управления питанием. Биты (4:0) показывают текущее состояние питания. Бит (5) показывает, что биты статуса актуальны. Только чтение.
RESERVED	23:19	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
TDFI_LP_RESP	18:16	0x0	0x0-0x7	Определяет значение временного параметра LP_RESP – максимальное число тактов между выставлением запроса и выдачей подтверждения по интерфейсу управления питанием.
RESERVED	15:12	-	0x0-0xf	Зарезервировано для дальнейшего использования.
LPI_WAKEUP_ TIMEOUT	11:0	0x000	0x0-0xfff	Задаёт максимальное число тактов между снятием запроса и снятием подтверждения по интерфейсу управления питанием.

Регистр DENALI_CTL_24

Таблица 4.45 – Описание полей регистра DENALI_CTL_24

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:26	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
LP_AUTO_MEM_GATE_EN	25:24	0x0	0x0-0x3	Разрешение отключения синхросигнала памяти при входе в энергосберегающий режим через автосчетчики. Бит (0) контролирует отключение питания, бит (1) – режим self-refresh. Установить каждый бит в 1 для разрешения.
RESERVED	23:19	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
LP_AUTO_EXIT_EN	18:16	0x0	0x0-0x7	Разрешение автоматического выхода из любого состояния энергосбережения при поступлении в очередь команд команды записи или чтения. Бит (0) контролирует отключение питания, бит (1) – режим self-refresh, бит (2) – режим self-refresh с отключением синхросигнала памяти и контроллера. Установить каждый бит в 1 для разрешения.
RESERVED	15:11	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
LP_AUTO_ENTRY_EN	10:8	0x0	0x0-0x7	Разрешение автоматического входя в любое состояние энергосбережения по таймеру неактивных действий. Бит (0) контролирует отключение питания, бит (1) – режим self-refresh, бит (2) – режим self-refresh с отключением синхросигнала памяти и контроллера. Установить каждый бит в 1 для разрешения.
RESERVED	7:4	-	0x0-0xf	Зарезервировано для дальнейшего использования.
LP_ARB_STATE	3:0	0x0	0x0-0xf	Отчет о состоянии арбитра. Биты (2:0) показывают какой интерфейс управляет модулем контроля питания. Бит (3) показывает, блокирует ли программный интерфейс арбитра. Для битов (2:0): значение 0 показывает что нет управления, значение 1 показывает управление со стороны программного интерфейса, значение 2 показывает управление со стороны интерфейса с внешней ножки, значение 3 показывает управление со стороны автоматического интерфейса, значение 4 показывает управление динамическим питанием со стороны внешнего интерфейса CS, значение 5 показывает, что контроллер под управлением. Только чтение.

Регистр DENALI_CTL_25

Таблица 4.46 – Описание полей регистра DENALI_CTL_25

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
LP_AUTO_SR_MC_GATE_IDLE	31:24	0x00	0x0-0xff	Число тактов до того, как контроллер переведет память в режим self-refresh с отключением синхросигнала памяти и контроллера.
LP_AUTO_SR_IDLE	23:16	0x00	0x0-0xff	Число тактов до того, как контроллер переведет память в режим self-refresh.
RESERVED	15:12	-	0x0-0xf	Зарезервировано для дальнейшего использования.
LP_AUTO_PD_IDLE	11:0	0x000	0x0-0xfff	Определяет интервал времени отсутствия операций в памяти, после чего контроллер переведет память в режим power-down.

Регистр DENALI_CTL_26

Таблица 4.47 – Описание полей регистра DENALI_CTL_26

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:7	-	0x0-0x1ffffff	Зарезервировано для дальнейшего использования.
RESERVED	6:0	-	0x0-0x7f	Зарезервировано для дальнейшего использования. См. конфигурационные файлы для начального программирования.

Регистр DENALI_CTL_27

Таблица 4.48 – Описание полей регистра DENALI_CTL_27

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:27	-	0x0-0x1ffffff	Зарезервировано для дальнейшего использования.
WRITE_MODEREG	26:0	0x0000000	0x0-0x7ffffff	Управление записью в регистр режима памяти Биты (7:0) определяют номер регистра режима памяти, если бит (23) установлен, биты (15:8) определяют CS, если бит (24) очищен, биты (23:16) определяют, в какие регистры режима памяти писать, бит (24) определяет, будут ли записаны все CS, бит (25) запускает запись.

Регистр DENALI_CTL_28

Таблица 4.49 – Описание полей регистра DENALI_CTL_28

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:23	-	0x0-0x1ff	Зарезервировано для дальнейшего использования.
MR0_DATA_0	22:8	0x0000	0x0-0x7fff	Данные для программирования регистра режима памяти 0 для CS0.
MRW_STATUS	7:0	0x00	0x0-0xff	Статус регистра режима записи памяти. Установка бита (0) показывает ошибку программирования параметра WRITE_MODEREG. Биты (1), (2) и (4) – резерв. Бит (3) показывает ошибку режима self-refresh и режима power-down. Только чтение.

Регистр DENALI_CTL_29

Таблица 4.50 – Описание полей регистра DENALI_CTL_29

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31	-	0x0-0x1	Зарезервировано для дальнейшего использования.
MR2_DATA_0	30:16	0x0000	0x0-0x7fff	Данные для программирования регистра режима памяти 2 для CS0.
RESERVED	15	-	0x0-0x1	Зарезервировано для дальнейшего использования.
MR1_DATA_0	14:0	0x0000	0x0-0x7fff	Данные для программирования регистра режима памяти 1 для CS0.

Регистр DENALI_CTL_30

Таблица 4.51 – Описание полей регистра DENALI_CTL_30

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31	-	0x0-0x1	Зарезервировано для дальнейшего использования.
MR3_DATA_0	30:16	0x0000	0x0-0x7fff	Данные для программирования регистра режима памяти 3 для CS0.
RESERVED	15	-	0x0-0x1	Зарезервировано для дальнейшего использования.
MRSINGLE_ DATA_0	14:0	0x0000	0x0-0x7fff	Данные для программирования в регистр режима памяти одиночной записью для CS0.

Регистр DENALI_CTL_31

Таблица 4.52 – Описание полей регистра DENALI_CTL_31

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:20	-	0x0-0xfff	Зарезервировано для дальнейшего использования.
ZQINIT	19:8	0x000	0x0-0xfff	Число тактов, необходимое для команды ZQINIT.
RESERVED	7:5	-	0x0-0x3	Зарезервировано для дальнейшего использования.
LONG_COUNT_MASK	4:0	0x00	0x0-0x1f	Уменьшение периода работы длинного счетчика (1024 такта). Поддерживаются только значения 0x00 (1024 такта), 0x10 (512 тактов), 0x18 (256 тактов), 0x1C (128 тактов), 0x1E (64 такта) and 0x1F (32 такта).

Регистр DENALI_CTL_32

Таблица 4.53 – Описание полей регистра DENALI_CTL_32

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:28	-	0x0-0xf	Зарезервировано для дальнейшего использования.
ZQCS	27:16	0x000	0x0-0xfff	Число тактов, необходимое для команды ZQCS.
RESERVED	15:12	-	0x0-0xf	Зарезервировано для дальнейшего использования.
ZQCL	11:0	0x000	0x0-0xfff	Число тактов, необходимое для команды ZQCL.

Регистр DENALI_CTL_33

Таблица 4.54 – Описание полей регистра DENALI_CTL_33

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:10	-	0x0-0x3fffff	Зарезервировано для дальнейшего использования.
ZQ_ON_SREF_ EXIT	9:8	0x0	0x0-0x3	Определяет тип ZQ калибровки, выполняемой при выходе из режима self-refresh. Запись значения 0x1 вызовет выполнение ZQCS команды, 0x2 – ZQCL, запись 0x0 отменяет все ZQ команды при выходе из режима self-refresh.
RESERVED	7:2	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
ZQ_REQ	1:0	0x0	0x0-0x3	Запрос пользователя на выполнение ZQ калибровки. Запись значения 0x1 вызовет выполнение ZQCS команды, 0x2 – ZQCL, запись 0x0 отменяет все ZQ команды. В это поле можно писать только после обнуления параметра ZQ_IN_PROGRESS. Только запись.

Регистр DENALI_CTL_34

Таблица 4.55 – Описание полей регистра DENALI_CTL_34

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
ZQ_INTERVAL	31:0	0x00000000	0x0-0xffffffff	Число длинных последовательностей счетчиков, допустимое между автоматическими ZQCS командами.

Регистр DENALI_CTL_35

Таблица 4.56 – Описание полей регистра DENALI_CTL_35

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:28	-	0x0-0xf	Зарезервировано для дальнейшего использования.
COL_DIFF	27:24	0x0	0x0-0xf	Разница между количеством доступных выводов столбцов и количеством используемых.
RESERVED	23:19	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
ROW_DIFF	18:16	0x0	0x0-0x7	Разница между количеством доступных адресных выводов и количеством используемых.
RESERVED	15:10	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
RESERVED	9:8	-	0x0-0x3	Зарезервировано для дальнейшего использования. См. конфигурационные файлы для начального программирования.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
ZQ_IN_PROGRESS	0	0x0	0x0-0x1	Значение 1 показывает, что ZQ команда сейчас выполняется. Только чтение.

Регистр DENALI_CTL_36

Таблица 4.57 – Описание полей регистра DENALI_CTL_36

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
AGE_COUNT	31:24	0x00	0x0-0xff	Начальное значение мастер-счетчика скорости старения команд
RESERVED	23:20	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
APREBIT	19:16	0xa	0x0-0xf	Расположение бита auto pre-charge в адресе DRAM.
RESERVED	15:9	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
BANK_ADDR_ INTLV_EN	8	0x0	0x0-0x1	Разрешение совмещать адрес банка с битами адреса строки. Установить в 1 для разрешения.
RESERVED	7:5	-	0x0-0x7	Зарезервировано для дальнейшего использования.
BANK_START_BIT	4:0	0x00	0x0-0x1f	Определяет младший бит адреса банка внутри адреса страницы или пользователя, когда установлен параметр BANK_ADDR_INTLV_EN.

Регистр DENALI_CTL_37

Таблица 4.58 – Описание полей регистра DENALI_CTL_37

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
BANK_SPLIT_EN	24	0x0	0x0-0x1	Разрешает факт обнаружения адресов в разные банки использовать как правило для постановки команд в очередь. Установить в 1 для разрешения.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RESERVED	16	-	0x0-0x1	Зарезервировано для дальнейшего использования. См. конфигурационные файлы для начального программирования.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
ADDR_CMP_EN	8	-	0x0-0x1	Разрешает факт обнаружения адресных конфликтов использовать как правило для постановки команд в очередь. Установить в 1 для разрешения.
COMMAND_ AGE_COUNT	7:0	0x00	0x0-0xff	Начальное значение индивидуальных счетчиков устаревания команды.

Регистр DENALI_CTL_38

Таблица 4.59 – Описание полей регистра DENALI_CTL_38

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RW_SAME_ PAGE_EN	24	0x0	0x0-0x1	Разрешение группировки запросов в одну страницу во время группировки команд чтения/записи как правила для построения очереди команд. Верно только при установке параметра RW_SAME_EN. Установить в 1 для разрешения.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RW_SAME_ EN	16	0x0	0x0-0x1	Разрешение группировки команд чтения/записи как правила для построения очереди команд. Установить в 1 для разрешения.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
PRIORITY_EN	8	0x0	0x0-0x1	Разрешение использования приоритета как правила построения очереди команд. Установить в 1 для разрешения.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
PLACEMENT_EN	0	0x0	0x0-0x1	Разрешение логики построения очереди команд. Установить в 1 для разрешения.

Регистр DENALI_CTL_39

Таблица 4.60 – Описание полей регистра DENALI_CTL_39

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
DISABLE_RD_ INTERLEAVE	24	0x0	0x0-0x1	Запрещение чередования читаемых данных для команд одного порта независимо от ID запроса.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
SWAP_EN	16	0x0	0x0-0x1	Разрешение логики перестановки команд в модуле выполнения. Установить в 1 для разрешения.
RESERVED	15:11	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
NUM_Q_ENTRIES_ ACT_DISABLE	10:8	0x00	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.
RESERVED	7:2	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
DISABLE_RWGRO UP_W_BNK_ CONFLICT	1:0	0x00	0x0-0x3	Запрещение постановки в группу чтения/записи, когда группировка приводит к конфликту банков. Бит (0) контролирует постановку следующей за конфликтующей командой, бит (1) – постановку через одну от конфликтующей команды. Установить каждый бит в 1 для запрещения.

Регистр DENALI_CTL_40

Таблица 4.61 – Описание полей регистра DENALI_CTL_40

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:27	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
Q_FULLNESS	26:24	0x0	0x0-0x7	Величина, задающая полноту очереди команд.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
REDUC	16	0x0	0x0-0x1	Разрешает режим половинной разрядности данных для контроллера. Установить в 1 для разрешения.
RESERVED	15:12	-	0x0-0xf	Зарезервировано для дальнейшего использования.
BURST_ON_FLY_BIT	11:8	0x0	0x0-0xf	Определяет значение бита «BURST_ON_FLY» в регистре режима памяти.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
INHIBIT_DRAM_CMD	0	0x0	0x0-0x1	Запрещает чередование команд чтения/записи и команд для других банков памяти. Установить в 1 для запрета.

Регистр DENALI_CTL_41

Таблица 4.62 – Описание полей регистра DENALI_CTL_41

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
CTRLUPD_REQ_ PER_AREF_EN	24	0x0	0x0-0x1	Разрешает автоматическое инициирование контроллером обновления (dfi_ctrlupd_req) после каждой регенерации памяти. Установить в 1 для разрешения.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
CTRLUPD_REQ	16	0x0	0x0-0x1	Объявляет, что контроллер инициировал сигнал запроса обновления dfi_ctrlupd_req. Установить в 1 для запуска. Только запись
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
CONTROLLER_ BUSY	8	0x0	0x0-0x1	Показывает, что контроллер обрабатывает команду. Учитывает все порты для внешних транзакций. Значение 1 показывает, что контроллер занят. Только чтение.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
IN_ORDER_ ACCEPT	0	0x0	0x0-0x1	Заставляет контроллер принимать команды в том порядке, в каком они расположены в очереди команд.

Регистр DENALI_CTL_42

Таблица 4.63 – Описание полей регистра DENALI_CTL_42

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:28	-	0x0-0xf	Зарезервировано для дальнейшего использования.
DFI_ERROR_INFO	27:8	0x00000	0x0-0xfffff	Содержит код ошибки интерфейса контроллера к PHY. Только чтение.
RESERVED	7:5	-	0x0-0x7	Зарезервировано для дальнейшего использования.
DFI_ERROR	4:0	0x0	0x0-0x1	Содержит флаг ошибки интерфейса контроллера к PHY. Только чтение.

Регистр DENALI_CTL_43

Таблица 4.64 – Описание полей регистра DENALI_CTL_43

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:29	-	0x0-0x7	Зарезервировано для дальнейшего использования.
INT_STATUS	28:8	0x00000	0x0-0x1fffff	Статус прерываний контроллера. Только чтение.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RESERVED	0	-	0x0-0x1	Зарезервировано для дальнейшего использования. См. конфигурационные файлы для начального программирования.

Регистр DENALI_CTL_44

Таблица 4.65 – Описание полей регистра DENALI_CTL_44

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:20	-	0x0-0xfff	Зарезервировано для дальнейшего использования.
INT_ACK	19:0	0x00000	0x0-0xfffff	Маска очистки битов поля INT_STATUS. Только запись.

Регистр DENALI_CTL_45

Таблица 4.66 – Описание полей регистра DENALI_CTL_45

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:21	-	0x0-0x7ff	Зарезервировано для дальнейшего использования.
INT_MASK	20:0	0x000000	0x0-0x1fffff	Маска формирования прерываний контроллера по содержимому поля INT_STATUS.

Регистр DENALI_CTL_46

Таблица 4.67 – Описание полей регистра DENALI_CTL_46

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:30	-	0x0-0x3	Зарезервировано для дальнейшего использования.
OUT_OF_RANGE_ ADDR	29:0	0x00000000	0x0-0x3fffffff	Адрес команды, которая вызвала прерывание по выходу адреса за границы. Только чтение.

Регистр DENALI_CTL_47

Таблица 4.68 – Описание полей регистра DENALI_CTL_47

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
OUT_OF_RANGE_ SOURCE_ID	31:24	0x00	0x0-0xff	ID источника команды, которая вызвала прерывание по выходу адреса за границы. Только чтение.
RESERVED	23:18	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
OUT_OF_RANGE_ TYPE	17:16	0x0	0x0-0x3	Тип команды, которая вызвала прерывание по выходу адреса за границы. Только чтение.
OUT_OF_RANGE_ LENGTH	15:0	0x0000	0x0-0xffff	Длина команды, которая вызвала прерывание переполнения. Только чтение.

Регистр DENALI_CTL_48

Таблица 4.69 – Описание полей регистра DENALI_CTL_48

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:30	-	0x0-0x3	Зарезервировано для дальнейшего использования.
PORT_CMD_ ERROR_ADDR	29:0	0x00000000	0x0-0x3fffffff	Адрес команды, которая вызвала ошибку команды порта. Только чтение.

Регистр DENALI_CTL_49

Таблица 4.70 – Описание полей регистра DENALI_CTL_49

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TODTL_2CMD	31:24	0x00	0x0-0xff	Определяет задержку от снятия сигнала ODT до следующей команды не записи и не чтения.
RESERVED	23:18	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
PORT_CMD_ ERROR_TYPE	17:16	0x0	0x0-0x3	Тип ошибки и доступа, которые привели к ошибке команды порта. Только чтение.
PORT_CMD_ ERROR_ID	15:0	0x0000	0x0-0xffff	ID источника команды, которая привела к ошибке команды порта. Только чтение.

Регистр DENALI_CTL_50

Таблица 4.71 – Описание полей регистра DENALI_CTL_50

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:29	-	0x0-0x7	Зарезервировано для дальнейшего использования.
R2R_SAMECS_DLY	28:24	0x00	0x0-0x1f	Дополнительная задержка, вставляемая между двумя чтениями для того же CS. Поддерживается любое значение, включая нулевое.
RESERVED	23:21	-	0x0-0x7	Зарезервировано для дальнейшего использования.
RESERVED	20:16	-	0x0-0x1f	Зарезервировано для дальнейшего использования. См. конфигурационные файлы для начального программирования.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
ODT_EN	8	0x0	0x0-0x1	Разрешает использование сигнала ODT. Когда разрешено, контроллер снимает и выставляет этот сигнал для памяти, когда это требуется.
RESERVED	7:4	-	0x0-0xf	Зарезервировано для дальнейшего использования.
TODTH_WR	3:0	0x0	0x0-0xf	Определяет для команды записи минимальное время для сигнала ODT, когда он держит положительную полку.

Регистр DENALI_CTL_51

Таблица 4.72 – Описание полей регистра DENALI_CTL_51

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:27	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
SW_LEVENING_ MODE	26:24	0x0	0x0-0x7	Определяет режим программных операций: 0x0 – нет операций, 0x1 – режим записи, 0x2 - режим тренировки данных, 0x3 - режим тренировки аппаратуры.
RESERVED	23:21	-	0x0-0x7	Зарезервировано для дальнейшего использования.
W2W_SAMECS_ DLY	20:16	0x00	0x0-0x1f	Дополнительная задержка, вставляемая между двумя записями для одного и того же CS. Любое значение поддерживается, включая 0x0.
RESERVED	15:13	-	0x0-0x7	Зарезервировано для дальнейшего использования.
W2R_SAMECS_ DLY	12:8	0x00	0x0-0x1f	Дополнительная задержка, вставляемая между записями и чтениями для одного и того же CS.
RESERVED	7:4	-	0x0-0xf	Зарезервировано для дальнейшего использования.
R2W_SAMECS_ DLY	4:0	0x02	0x0-0x1f	Дополнительная задержка, вставляемая между чтениями и записями для одного и того же CS. Поддерживается любое значение, кроме нулевого.

Регистр DENALI_CTL_52

Таблица 4.73 – Описание полей регистра DENALI_CTL_52

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
SWLVL_OP_DONE	24	0x0	0x0-0x1	Сообщает, что программный режим сейчас выполняется. Значение 0x1говорит, что операция закончилась. Только чтение.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
SWLVL_EXIT	16	0x0	0x0-0x1	Значение 0x1 вызывает выход из программного режима. Только запись.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
SWLVL_START	8	0x0	0x0-0x1	Значение 0x1 вызывает вход в программный режим, тип его определяется полем SW_LEVENING_ MODE. Только запись.
RESERVED	7:4	-	0x0-0xf	Зарезервировано для дальнейшего использования.
SWLVL_LOAD	0	0x0	0x0-0x1	Значение 0x1 означает запрос пользователя на загрузку задержек и выполнение программного режима. Только запись.

Регистр DENALI_CTL_53

Таблица 4.74 – Описание полей регистра DENALI_CTL_53

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
SWLVL_RESP_3	24	0x0	0x0-0x1	Отклик от слайса данных 3 в режиме тренировки. Только чтение.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
SWLVL_RESP_2	16	0x0	0x0-0x1	Отклик от слайса данных 2 в режиме тренировки. Только чтение.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
SWLVL_RESP_1	8	0x0	0x0-0x1	Отклик от слайса данных 1 в режиме тренировки. Только чтение.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
SWLVL_RESP_0	0	0x0	0x0-0x1	Отклик от слайса данных 0 в режиме тренировки. Только чтение.

Регистр DENALI_CTL_54

Таблица 4.75 – Описание полей регистра DENALI_CTL_54

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:30	-	0x0-0x3	Зарезервировано для дальнейшего использования.
WLMRD	29:24	0x00	0x0-0x3f	Задержка между командой MRS и первым стробом в режиме программной записи.
RESERVED	23:22	-	0x0-0x3	Зарезервировано для дальнейшего использования.
WLDQSEN	21:16	0x00	0x0-0x3f	Задержка между командой MRS и первым стробом DQS в режиме программной записи.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
WRLVL_CS	8	0x0	0x0-0x1	Определяет нужный CS для операций в режиме программной записи, инициируемой полем WRLVL_REQ.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
WRLVL_REQ	0	0x0	0x0-0x1	Значение 0x1 вызывает вход в программный режим записи. Только запись.

Регистр DENALI_CTL_55

Таблица 4.76 – Описание полей регистра DENALI_CTL_55

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
DFI_PHY_WRLVL_ MODE	24	0x0	0x0-0x1	Задаёт поддержку PHYв программном режиме записи по интерфейсу контроллер-PHY. Установить в 1 для поддержки.
WRLVL_ INTERVAL	23:8	0x0000	0x0-0xffff	Число длинных последовательностей счётчиков, вычисляемых между автоматическими командами записи в программном режиме.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
WRLVL_EN	0	0x0	0x0-0x1	Разрешение контроллеру перейти к программному режиму записи. Установить в 1 для разрешения.

Регистр DENALI_CTL_56

Таблица 4.77 – Описание полей регистра DENALI_CTL_56

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
WRLVL_AREF_EN	24	0x0	0x0-0x1	Разрешает регенерацию памяти и другие команды без данных в середине выполнения программного режима записи. Установить в 1 для разрешения.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
WRLVL_RESP_ MASK	16	0x0	0x0-0x1	Маскирует dfi_wrlvl_resp сигнал при программном режиме записи.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
WRLVL_ON_SREF_ EXIT	8	0x0	0x0-0x1	При выходе из режима self-refresh разрешает автоматический вход в программный режим записи. Установить в 1 для разрешения.
RESERVED	7:2	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
WRLVL_PERIODIC	1:0	0x0	0x0-0x3	Разрешает dfi_lvl_periodic сигнал при программном режиме записи. Установить в 1 для разрешения.

Регистр DENALI_CTL_57

Таблица 4.78 – Описание полей регистра DENALI_CTL_57

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_CS	24	0x0	0x0-0x1	Определяет нужный CS для операций в режиме тренировки данных, инициируемых полями RDLVL_REQ и RDLVL_GATE_REQ.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_GATE_ REQ	16	0x0	0x0-0x1	Запрос пользователя на инициализацию режима тренировки аппаратуры. Установить в 1 для инициализации. Только запись.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_REQ	8	0x0	0x0-0x1	Запрос пользователя на инициализацию режима тренировки данных. Установить в 1 для инициализации. Только запись.
RESERVED	7:2	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
WRLVL_ERROR_ STATUS	1:0	0x0	0x0-0x3	Хранит код прерываний в программном режиме записи. Бит (0) , равный 1, отражает нарушение параметра TDFI_WRLVL_MAX, а бит (1) , равный 1, отражает нарушение параметра TDFI_WRLVL_RESP. Только чтение.

Регистр DENALI_CTL_58

Таблица 4.79 – Описание полей регистра DENALI_CTL_58

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
DFI_PHY_RDLVL_ GATE_MODE	24	0x0	0x0-0x1	Задаёт поддержку PHY в режиме тренировки аппаратуры по интерфейсу контроллер-PHY. Установить в 1 для поддержки.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
DFI_PHY_RDLVL_ MODE	16	0x0	0x0-0x1	Задаёт поддержку PHY в режиме тренировки данных по интерфейсу контроллер-PHY. Установить в 1 для поддержки.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
DFI_GATE_SEQ_ EN	8	0x0	0x0-0x1	Определяет выполнение команды MPR для режима тренировки аппаратуры.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_ SEQ_EN	0	0x0	0x0-0x1	Определяет выполнение команды MPR для режима тренировки данных.

Регистр DENALI_CTL_59

Таблица 4.80 – Описание полей регистра DENALI_CTL_59

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_GATE_ ON_SREF_EXIT	24	0x0	0x0-0x1	Значение 0x1 разрешает при выходе из режима self-refresh автоматически перейти в режим тренировки аппаратуры.
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_GATE_ PERIODIC	16	0x0	0x0-0x1	Разрешает dfi_lvl_periodic сигнал в режиме тренировки аппаратуры. Установить в 1 для разрешения.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_ON_SREF_ EXIT	8	0x0	0x0-0x1	Значение 0x1 разрешает при выходе из режима self-refresh автоматически перейти в режим тренировки данных.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_ PERIODIC	0	0x0	0x0-0x1	Разрешает dfi_lvl_periodic сигнал в режиме тренировки данных. Установить в 1 для разрешения.

Регистр DENALI_CTL_60

Таблица 4.81 – Описание полей регистра DENALI_CTL_60

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
AXI0_W_PRIORITY	24	0x0	0x0-0x1	Приоритет команд записи 0-го порта AXI. 0x0 - наивысший приоритет
RESERVED	23:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
AXI0_R_PRIORITY	16	0x0	0x0-0x1	Приоритет команд чтения 0-го порта AXI. 0x0 - наивысший приоритет.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
AXI0_FIXED_ PORT_PRIORITY_ ENABLE	8	0x0	0x0-0x1	Определяет управление приоритетом 0-го порта AXI либо по порту, либо по командам. Установить в 1 для приоритета по порту с учётом полей AXI0_W_PRIORITY и AXI0_R_PRIORITY. Установить в 0 для приоритета по командам.
RESERVED	7:1	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
RDLVL_AREF_EN	0	0x0	0x0-0x1	Разрешает регенерацию памяти и другие команды без данных в середине выполнения режима тренировки данных или аппаратуры. Установить в 1 для разрешения.

Регистр DENALI_CTL_61

Таблица 4.82 – Описание полей регистра DENALI_CTL_61

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:17	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
MEM_RST_VALID	16	0x0	0x0-0x1	Регистровый доступ к сигналу mem_rst_valid. Только чтение.
RESERVED	15:9	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
CKE_STATUS	8	0x0	0x0-0x1	Регистровый доступ к сигналу cke_status. Только чтение.
RESERVED	7:2	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
AXI0_FIFO_TYPE_ REG	1:0	0x0	0x0-0x3	Задание тактовых доменов 0-го порта AXI и контроллера: 0x0 – домены асинхронные; 0x1 – 2:1 псевдо-синхронные; 0x2 – 1:2 псевдо-синхронные; 0x3 – синхронные.

Регистр DENALI_CTL_62

Таблица 4.83 – Описание полей регистра DENALI_CTL_62

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:30	-	0x0-0x3	Зарезервировано для дальнейшего использования.
TDFI_PHY_WRLAT	29:24	Вычисленное значение	0x0-0x3f	Хранит рассчитанный временной параметр tPHY_WRLAT (в тактах DFI PHY), максимальное число тактов между командой записи и установкой сигнала dfi_wrdata_en. Только чтение.
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. Если этот сигнал не используется PHY, то параметр можно игнорировать.
DLL_RST_DELAY	15:0	0x0000	0x0-0xffff	Минимальное число тактов, на которое необходимо удерживать сигнал сброса DLL dll_rst_n. Если этот сигнал не используется PHY, то параметр можно игнорировать.

Регистр DENALI_CTL_63

Таблица 4.84 – Описание полей регистра DENALI_CTL_63

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:25	-	0x0-0x7f	Зарезервировано для дальнейшего использования.
DRAM_CLK_ DISABLE	24	0x0	0x0-0x1	Установленное значение сигнала dfi_dram_clk_disable. Бит (0) контролирует cs0, бит (1) – cs1 и т.д. Установить каждый бит в 1 для запрета.
RESERVED	23:22	-	0x0-0x3	Зарезервировано для дальнейшего использования.
TDFI_RDDATA_EN	21:16	Вычисленное значение	0x0-0x3f	Содержит расчетный временной параметр tRDDATA_EN (в тактах DFI PHY), максимальное число тактов между командой чтения и установкой сигнала dfi_rddata_en. Только чтение.
RESERVED	15:14	-	0x0-0x3	Зарезервировано для дальнейшего использования.
TDFI_PHY_RDLAT	13:8	0x06	0x0-0x3f	Определяет tPHY_RDLAT временной параметр (в тактах DFI PHY), максимальное число тактов между установкой dfi_rddata_en и установкой dfi_rddata_valid.
RESERVED	7	-	0x0-0x1	Зарезервировано для дальнейшего использования.
UPDATE_ERROR_ STATUS	6:0	0x00	0x0-0x7f	Идентифицирует источник любой ошибки контроллера или PHY-инициированного обновления. Значение 1 показывает временную ошибку соответствующего временного параметра. Только чтение.

Регистр DENALI_CTL_64

Таблица 4.85 – Описание полей регистра DENALI_CTL_64

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:22	-	0x0-0x3ff	Зарезервировано для дальнейшего использования.
TDFI_CTRLUPD_MAX	21:8	0x0000	0x0-0x3fff	Определяет временной параметр tCTRLUPD_MAX (в тактах DFI), т.е. максимальное количество тактов установки сигнала dfi_ctrlupd_req. Если запрограммирован в ненулевое значение, то временная ошибка вызовет прерывание, и бит (1) установится в параметре UPDATE_ERROR_STATUS.
RESERVED	7:4	-	0x0-0xf	Зарезервировано для дальнейшего использования.
TDFI_CTRLUPD_MIN	3:0	0x0	0x0-0xf	Показывает временной параметр tCTRLUPD_MIN (в тактах DFI), т.е. минимальное количество тактов установки сигнала dfi_ctrlupd_req.

Регистр DENALI_CTL_65

Таблица 4.86 – Описание полей регистра DENALI_CTL_65

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TDFI_PHYUPD_TYPE1	31:16	0x0000	0x0-0xffff	Определяет временной параметр tPHYUPD_TYPE1 (в тактах DFI), т.е. 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) установится в параметре UPDATE_ERROR_STATUS.
TDFI_PHYUPD_TYPE0	15:0	0x0000	0x0-0xffff	Определяет временной параметр tPHYUPD_TYPE0 (в тактах DFI), т.е. 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) установится в параметре UPDATE_ERROR_STATUS.

Регистр DENALI_CTL_66

Таблица 4.87 – Описание полей регистра DENALI_CTL_66

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TDFI_ PHYUPD_ TYPE3	31:16	0x0000	0x0-0xffff	Определяет временной параметр tPHYUPD_TYPE3 (в тактах DFI), т.е. 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) установится в параметре UPDATE_ERROR_STATUS.
TDFI_ PHYUPD_ TYPE2	15:0	0x0000	0x0-0xffff	Определяет временной параметр tPHYUPD_TYPE2 (в тактах DFI), т.е. 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) установится в параметре UPDATE_ERROR_STATUS.

Регистр DENALI_CTL_67

Таблица 4.88 – Описание полей регистра DENALI_CTL_67

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:14	-	0x0-0x3ffff	Зарезервировано для дальнейшего использования.
TDFI_PHYUPD_ RESP	13:0	0x0000	0x0-0x3fff	Определяет временной параметр tPHYUPD_RESP (в тактах DFI), т.е. 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) установится в поле UPDATE_ERROR_STATUS.

Регистр DENALI_CTL_68

Таблица 4.89 – Описание полей регистра DENALI_CTL_68

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
TDFI_CTRLUPD_ INTERVAL	31:0	0x00000000	0x0-0xffffffff	Определяет временной параметр tCTRLUPD_INTERVAL (в тактах DFI), т.е. максимальное число тактов между установками сигнала dfi_ctrlupd_req. Если запрограммирован в не нулевое значение, то временная ошибка вызовет прерывание и бит (0) установится в поле UPDATE_ERROR_STATUS.

Регистр DENALI_CTL_69

Таблица 4.90 – Описание полей регистра DENALI_CTL_69

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:28	-	0x0-0xf	Зарезервировано для дальнейшего использования.
TDFI_DRAM_CLK_ DISABLE	27:24	0x0	0x0-0xf	Определяет временной параметр tDRAM_CLK_DISABLE (в тактах DFI), т.е. задержку между установкой сигнала dfi_dram_clock_disable и отключением тактового сигнала памяти.
RESERVED	23:20	-	0x0-0xf	Зарезервировано для дальнейшего использования.
TDFI_CTRL_ DELAY	19:16	0x2	0x0-0xf	Определяет временной параметр tCTRL_DELAY (в тактах DFI),т.е. задержку между изменением команды DFI и командой памяти.
RESERVED	15:14	-	0x0-0x3	Зарезервировано для дальнейшего использования.
WRLAT_ADJ	13:8	0x00	0x0-0x3f	Поправка к времени записи PHY.
RESERVED	7:6	-	0x0-0x3	Зарезервировано для дальнейшего использования.
RDLAT_ADJ	5:0	0x00	0x0-0x3f	Поправка к времени чтения PHY.

Регистр DENALI_CTL_70

Таблица 4.91 – Описание полей регистра DENALI_CTL_70

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:26	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
TDFI_WRLVL_WW	25:16	0x000	0x0-0x3ff	Определяет временной параметр WRLVL_WW (в тактах DFI), т.е. минимальное число тактов между установкой строба dfi_wrlvl_strobe.
TDFI_WRLVL_EN	15:8	0x00	0x0-0xff	Определяет временной параметр WRLVL_EN (в тактах DFI), т.е. минимальное число тактов между установкой сигнала dfi_wrlvl_en и первой установкой строба dfi_wrlvl_strobe.
RESERVED	7:4	-	0x0-0xf	Зарезервировано для дальнейшего использования.
TDFI_DRAM_ CLK_ENABLE	3:0	0x0	0x0-0xf	Определяет временной параметр tDRAM_CLK_ENABLE (в тактах DFI), т.е. задержку между снятием сигнала dfi_dram_clk_disable и включением синхросигнала памяти.

Регистр DENALI_CTL_71

Таблица 4.92 – Описание полей регистра DENALI_CTL_71

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TDFI_WRLVL_ RESP	31:0	0x00000000	0x0-0xffffffff	Определяет временной параметр tWRLVL_RESP (в тактах DFI), т.е. максимальное число тактов между установкой сигнала dfi_wrlvl_req и установкой сигнала dfi_wrlvl_en.

Регистр DENALI_CTL_72

Таблица 4.93 – Описание полей регистра DENALI_CTL_72

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TDFI_WRLVL_ MAX	31:0	0x00000000	0x0-0xffffffff	Определяет временной параметр tWRLVL_MAX (в тактах DFI), т.е. максимальное число тактов между установкой сигнала dfi_wrlvl_en и установкой сигнала dfi_wrlvl_resp.

Регистр DENALI_CTL_73

Таблица 4.94 – Описание полей регистра DENALI_CTL_73

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:18	-	0x0-0x3fff	Зарезервировано для дальнейшего использования.
TDFI_RDLVL_RR	17:8	0x000	0x0-0x3ff	Определяет временной параметр RDLVL_RR (в тактах DFI), т.е. минимальное число тактов между командами чтения.
TDFI_RDLVL_EN	7:0	0x00	0x0-0xff	Определяет временной параметр RDLVL_EN (в тактах DFI), т.е. минимальное число тактов между установкой сигнала dfi_rdlvl_en или dfi_rdlvl_gate_en и первым чтением или командой MRR.

Регистр DENALI_CTL_74

Таблица 4.95 – Описание полей регистра DENALI_CTL_74

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
TDFI_RDLVL_ RESP	31:0	0x00000000	0x0-0xffffffff	Определяет временной параметр RDLVL_RESP (в тактах DFI), т.е. максимальное число тактов между установкой сигнала dfi_rdlvl_req или dfi_rdlvl_gate_req и установкой сигнала dfi_rdlvl_en или dfi_rdlvl_gate_en.

Регистр DENALI_CTL_75

Таблица 4.96 – Описание полей регистра DENALI_CTL_75

Название	Биты	Значение после системного сброса	Диапазон значений	Описание
RESERVED	31:17	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
RDLVL_GATE_EN	16	0x000	0x0-0x3ff	Разрешает контроллеру режим тренировки аппаратуры. Установить в 1 для разрешения.
RESERVED	15:9	-	0x0-0xf	Зарезервировано для дальнейшего использования.
RDLVL_EN	8	0x00	0x0-0xff	Разрешает контроллеру режим тренировки данных. Установить в 1 для разрешения
RESERVED	7:4	-	0x0-0xf	Зарезервировано для дальнейшего использования.
RDLVL_RESP_ MASK	3:0	0x0	0x0-0xf	Маскирование сигнала dfi_rdrlvl_resp во время режима тренировки данных.

Регистр DENALI_CTL_76

Таблица 4.97 – Описание полей регистра DENALI_CTL_76

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
TDFI_RDLVL_MAX	31:0	0x00000000	0x0-0xffffffff	Определяет временной параметр RDLVL_MAX (в тактах DFI), т.е. максимальное число тактов между установкой сигнала dfi_rdlvl_en или dfi_rdlvl_gate_en и установкой значимого сигнала dfi_rdlvl_resp.

Регистр DENALI_CTL_77

Таблица 4.98 – Описание полей регистра DENALI_CTL_77

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:24	-	0x0-0xff	Зарезервировано для дальнейшего использования.
RDLVL_INTERVAL	23:8	0x0000	0x0-0xffff	Число длинных последовательностей счётчиков, вычисляемых между автоматическими командами режима тренировки данных.
RESERVED	7:2	-	0x0-0x3f	Зарезервировано для дальнейшего использования.
RDLVL_ERROR_ STATUS	1:0	0x0	0x0-0x3	Хранит ошибки, связанные с прерыванием в режимах тренировки данных или аппаратуры. Бит (1) , равный 1, означает нарушение параметра TDFI_RDLVL_RESP. Бит (0) , равный 1, означает нарушение параметра TDFI_RDLVL_MAX.

Регистр DENALI_CTL_78

Таблица 4.99 – Описание полей регистра DENALI_CTL_78

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:30	-	0x0-0x3	Зарезервировано для дальнейшего использования.
TDFI_RDCSLAT	29:24	0x00	0x0-0x3f	Определяет временной параметр PHY_RDCSLAT (в тактах DFI), т.е. максимальное число тактов между командой чтения и установкой сигнала dfi_rddata_cs_n.
RESERVED	23:19	-	0x0-0x1f	Зарезервировано для дальнейшего использования.
TDFI_PHY_ WRDATA	18:16	0x0	0x0-0x7	Определяет временной параметр PHY_ WRDATA (в тактах DFI), т.е. максимальное число тактов между установкой сигнала dfi_wrdata_en и сигналом dfi_wrdata .
TDFI_GATE_ INTERVAL	15:0	0x0000	0x0-0xffff	Число длинных последовательностей счётчиков, вычисляемых между автоматическими командами режима тренировки аппаратуры.

Регистр DENALI_CTL_79

Таблица 4.100 – Описание полей регистра DENALI_CTL_79

Название	**Биты**	**Значение после системного сброса**	**Диапазон значений**	**Описание**
RESERVED	31:6	-	0x0-0x3ffffff	Зарезервировано для дальнейшего использования.
TDFI_WRCSLAT	5:0	0x00	0x0-0x3f	Определяет временной параметр PHY_WRCSLAT (в тактах DFI), т.е. максимальное число тактов между командой записи и установкой сигнала dfi_wrdata_cs_n.

4.10 Блок защиты памяти MPU

Блок защиты памяти разрешает доступ внешним устройствам в выделенные области памяти кластера.

Внешние устройства имеют произвольный доступ к внутренней памяти кластера через слэйв-интерфейс AXI(PCu). Блок защиты памяти находится внутри процессорной системы и непосредственно подключен к интерфейсу PCu.

Блок защиты памяти декодирует адрес транзакции и определяет, разрешён или запрещён доступ по этому адресу. Блок пропускает разрешённые транзакции без изменения, а запрещённые транзакции обрабатывает следующим образом:

- с точки зрения внешнего устройства - транзакция завершается с кодом ошибки SLVERR, при этом в транзакции чтения вместо данных во всех передачах транзакции возвращаются нулевые значения,
- с точки зрения процессорной системы - генерируется (маскируемый по каждой выделенной области) запрос на прерывание к процессору ARM Cortex-A5, который снимается программно.

Интерфейсы чтения и записи шины AXI 3.0 разделены, поэтому в процессорной системе присутствуют 2 экземпляра блоков защиты памяти: по одному на чтение и на запись (см. рисунок 4.31).

Рисунок 4.31 - Структурная схема блока защиты памяти

Каждый экземпляр блока защиты памяти состоит из двух частей:

- блока управления - который содержит конфигурационные регистры устройства и генерирует сигнал (MASK[31:0]) маски доступных областей памяти,
- блока обработки транзакций - который работает с сигналами шины AXI3, выдаёт сигнал (ERRSET[31:0]) обращения в запрещённую область.

Блок обработки транзакций производит сравнение адресов и самостоятельно завершает транзакции шины AXI 3.0, адреса которых попадают в запрещённые сегменты.

В обычном режиме блок обработки транзакций не вмешивается в обработку проходящих транзакций, только ведёт подсчёт текущего количества незавершённых транзакций (для которых прошла передача адреса и ещё не было передачи последних данных чтения). Как только на входном интерфейсе блока появляется адрес, попадающий в защищённую область, блок блокирует канал адреса, дожидается передачи всех данных от незавершённых транзакций и после этого выдаёт нулевые данные и код ошибки (RRESP=10b, т.у. ошибка конечного устройства).

Блок обработки транзакций сообщает блоку управления об обращении в защищённый сегмент с помощью шины ERRSETx. Блок управления фиксирует сигналы ERRSETx и в зависимости от настройки может генерировать запрос на прерывание.

4.10.1 Разделение внутренней памяти на защищаемые сегменты

Внутренняя память кластера разделяется на 32 защищаемых сегмента равного размера. Сегменту с номером i соответствует один бит с номером i в каждом регистре управления блоком. Ниже (см. таблицу 4.101) показана нумерация сегментов.

Таблица 4.101 – Нумерация сегментов внутренней памяти

Номер сегмента	Адреса в собственном пространстве CPU (байтовый доступ)	Смещение относительно базового адреса в карте памяти внешних NMPU (32-разрядный доступ)	Смещение относительно базового адреса в карте памяти ARM и устройств ввода-вывода (байтовый доступ)
0	0000_0000h - 00FF_FFFFh	0000_0000h - 003F_FFFFh	0000_0000h - 00FF_FFFFh
1	0100_0000h - 01FF_FFFFh	0040_0000h - 007F_FFFFh	0100_0000h - 01FF_FFFFh
2	0200_0000h - 02FF_FFFFh	0080_0000h – 00BF_FFFFh	0200_0000h - 02FF_FFFFh
3	0300_0000h - 03FF_FFFFh	00C0_0000h - 00FF_FFFFh	0300_0000h - 03FF_FFFFh
4	0400_0000h - 04FF_FFFFh	0100_0000h - 013F_FFFFh	0400_0000h - 04FF_FFFFh
5	0500_0000h - 05FF_FFFFh	0140_0000h - 017F_FFFFh	0500_0000h - 05FF_FFFFh
6	0600_0000h - 06FF_FFFFh	0180_0000h – 01BF_FFFFh	0600_0000h - 06FF_FFFFh
7	0700_0000h - 07FF_FFFFh	01C0_0000h - 01FF_FFFFh	0700_0000h - 07FF_FFFFh
8	0800_0000h - 08FF_FFFFh	0200_0000h - 023F_FFFFh	0800_0000h - 08FF_FFFFh
9	0900_0000h - 09FF_FFFFh	0240_0000h - 027F_FFFFh	0900_0000h - 09FF_FFFFh
10	0A00_0000h - 0AFF_FFFFh	0280_0000h – 02BF_FFFFh	0A00_0000h - 0AFF_FFFFh
11	0B00_0000h - 0BFF_FFFFh	02C0_0000h - 02FF_FFFFh	0B00_0000h - 0BFF_FFFFh
12	0C00_0000h - 0CFF_FFFFh	0300_0000h - 033F_FFFFh	0C00_0000h - 0CFF_FFFFh
13	0D00_0000h - 0DFF_FFFFh	0340_0000h - 037F_FFFFh	0D00_0000h - 0DFF_FFFFh
14	0E00_0000h - 0EFF_FFFFh	0380_0000h – 03BF_FFFFh	0E00_0000h - 0EFF_FFFFh
15	0F00_0000h - 0FFF_FFFFh	03C0_0000h - 03FF_FFFFh	0F00_0000h - 0FFF_FFFFh
16	1000_0000h - 10FF_FFFFh	0400_0000h - 043F_FFFFh	1000_0000h - 10FF_FFFFh
17	1100_0000h - 11FF_FFFFh	0440_0000h - 047F_FFFFh	1100_0000h - 11FF_FFFFh
18	1200_0000h - 12FF_FFFFh	0480_0000h – 04BF_FFFFh	1200_0000h - 12FF_FFFFh
29	1300_0000h - 13FF_FFFFh	04C0_0000h - 04FF_FFFFh	1300_0000h - 13FF_FFFFh
20	1400_0000h - 14FF_FFFFh	0500_0000h - 053F_FFFFh	1400_0000h - 14FF_FFFFh
21	1500_0000h - 15FF_FFFFh	0540_0000h - 057F_FFFFh	1500_0000h - 15FF_FFFFh
22	1600_0000h - 16FF_FFFFh	0580_0000h – 05BF_FFFFh	1600_0000h - 16FF_FFFFh
23	1700_0000h - 17FF_FFFFh	05C0_0000h - 05FF_FFFFh	1700_0000h - 17FF_FFFFh
24	1800_0000h - 18FF_FFFFh	0600_0000h - 063F_FFFFh	1800_0000h - 18FF_FFFFh
25	1900_0000h - 19FF_FFFFh	0640_0000h - 067F_FFFFh	1900_0000h - 19FF_FFFFh
26	1A00_0000h – 1AFF_FFFFh	0680_0000h – 06BF_FFFFh	1A00_0000h – 1AFF_FFFFh
27	1B00_0000h – 1BFF_FFFFh	06C0_0000h - 06FF_FFFFh	1B00_0000h – 1BFF_FFFFh
28	1C00_0000h – 1CFF_FFFFh	0700_0000h - 073F_FFFFh	1C00_0000h – 1CFF_FFFFh
29	1D00_0000h – 1DFF_FFFFh	0740_0000h - 077F_FFFFh	1D00_0000h – 1DFF_FFFFh
30	1E00_0000h – 1EFF_FFFFh	0780_0000h – 07BF_FFFFh	1E00_0000h – 1EFF_FFFFh
31	1F00_0000h – 1FFF_FFFFh	07C0_0000h - 07FF_FFFFh	1F00_0000h – 1FFF_FFFFh

4.10.2 Программно доступные регистры блока защиты памяти

Регистры блока защиты памяти находятся в области периферийных регистров кластера REG_CPU. Имеются 4 способа записи в каждый регистр:

1. Нормальный доступ - устанавливает в регистре записываемое значение,
2. Побитовая установка - устанавливает значение 1 в тех разрядах, в которых имеются единицы в записываемом данном, остальные разряды не меняются,

3. Побитовый сброс - устанавливает значение 0 в тех разрядах, в которых имеются единицы в записываемом данном, остальные разряды не меняются,
4. Обнуление - устанавливает значение 0 во всех битах регистра, независимо от записываемого значения (т.е. имеет значение сам факт записи).

Для каждого способа записи выделен отдельный адрес. Не все способы доступны для всех регистров. Чтение регистра может производиться по любому из описанных адресов.

Таблица 4.102 – Программно доступные регистры блока защиты памяти

Адрес (байтовый доступ)	Название	Доступ	Описание
0x000B_D000 hex	RD_MASK	ЧТ/ЗП	Маска защищаемых сегментов: 0 – обращение в данный сегмент по чтению разрешено, 1 – обращение в данный сегмент по чтению блокируется.
0x000B_D008 hex	RD_MASK_SET	ЗП	Побитовая установка регистра RD_MASK
0x000B_D010 hex	RD_MASK_CLEAR	ЗП	Побитовый сброс регистра RD_MASK
0x000B_D018 hex	RD_MASK_NULL	ЗП	Обнуление регистра RD_MASK
0x000B_D020 hex	RD_IRQ_STATUS_RAW	ЧТ	Статус запросов на прерывание без учёта маски: 0 – обращений по чтению в данный сегмент не было, 1 – зафиксировано обращение по чтению в данный сегмент.
0x000B_D028 hex	RD_IRQ_STATUS_RAW_SET	ЗП	Побитовая установка RD_IRQ_STATUS_RAW (только для отладки)
0x000B_D030 hex	RD_IRQ_STATUS_RAW_CLEAR	ЗП	Побитовый сброс регистра RD_IRQ_STATUS_RAW
0x000B_D038 hex	RD_IRQ_STATUS_RAW_NULL	ЗП	Обнуление регистра RD_IRQ_STATUS_RAW
0x000B_D040 hex	RD_IRQ_MASK	ЧТ/ЗП	Маска запросов на прерывание от регистра RD_IRQ_STATUS_RAW: 0 – прерывание запрещено, 1 – прерывание разрешено.
0x000B_D048 hex	RD_IRQ_MASK_SET	ЗП	Побитовая установка регистра RD_IRQ_MASK
0x000B_D050 hex	RD_IRQ_MASK_CLEAR	ЗП	Побитовый сброс регистра RD_IRQ_MASK
0x000B_D058 hex	RD_IRQ_MASK_NULL	ЗП	Обнуление регистра RD_IRQ_MASK
0x000B_D060 hex	RD_IRQ_STATUS	ЧТ	Статус запросов на прерывание с учётом маски: 0 – нет запроса, 1 – есть запрос.
0x000B_D068 hex	Зарезервировано	-	-
0x000B_D070 hex	Зарезервировано	-	-
0x000B_D078 hex	RD_IRQ_STATUS_NULL	ЗП	Обнуление незамаскированных запросов на прерывание (сбрасываются все биты регистра RD_IRQ_STATUS и все биты регистра RD_IRQ_STATUS_RAW, для которых установлен бит RD_IRQ_MASK).
0x000B_D100 hex	WR_MASK	ЧТ/ЗП	Маска защищаемых сегментов: 0 – обращение в данный сегмент по чтению разрешено, 1 – обращение в данный сегмент по чтению блокируется.
0x000B_D108 hex	WR_MASK_SET	ЗП	Побитовая установка регистра WR_MASK

Продолжение таблицы 4.102

Адрес (байтовый доступ)	Название	Доступ	Описание
0x000B_D110 hex	WR_MASK_CLEAR	ЗП	Побитовый сброс регистра WR_MASK
0x000B_D118 hex	WR_MASK_NULL	ЗП	Обнуление регистра WR_MASK
0x000B_D120 hex	WR_IRQ_STATUS_RAW	ЧТ	Статус запросов на прерывание без учёта маски: 0 – обращений по чтению в данный сегмент не было, 1 – зафиксировано обращение по чтению в данный сегмент.
0x000B_D128 hex	WR_IRQ_STATUS_RAW_SET	ЗП	Побитовая установка WR_IRQ_STATUS_RAW (только для отладки)
0x000B_D130 hex	WR_IRQ_STATUS_RAW_CLEAR	ЗП	Побитовый сброс регистра WR_IRQ_STATUS_RAW
0x000B_D138 hex	WR_IRQ_STATUS_RAW_NULL	ЗП	Обнуление регистра WR_IRQ_STATUS_RAW
0x000B_D140 hex	WR_IRQ_MASK	ЧТ/ЗП	Маска запросов на прерывание от регистра WR_IRQ_STATUS_RAW: 0 - прерывание запрещено, 1 - прерывание разрешено.
0x000B_D148 hex	WR_IRQ_MASK_SET	ЗП	Побитовая установка регистра WR_IRQ_MASK
0x000B_D150 hex	WR_IRQ_MASK_CLEAR	ЗП	Побитовый сброс регистра WR_IRQ_MASK
0x000B_D158 hex	WR_IRQ_MASK_NULL	ЗП	Обнуление регистра WR_IRQ_MASK
0x000B_D160 hex	WR_IRQ_STATUS	ЧТ	Статус запросов на прерывание с учётом маски: 0 - нет запроса, 1 - есть запрос.
0x000B_D168 hex	Зарезервировано	-	-
0x000B_D170 hex	Зарезервировано	-	-
0x000B_D178 hex	WR_IRQ_STATUS_NULL	ЗП	Обнуление незамаскированных запросов на прерывание (сбрасываются все биты регистра WR_IRQ_STATUS и все биты регистра WR_IRQ_STATUS_RAW, для которых установлен бит WR_IRQ_MASK).

4.10.3 Прерывания

Каждый экземпляр (чтение и запись) блока защиты памяти имеет один сигнал запроса на прерывание. Сигнал запроса установлен, пока хотя бы для одного из сегментов RD/WR_IRQ_STATUS_RAW[i]=1 и RD/WR_IRQ_MASK[i]=0. Запрос можно снять программно одним из двух способов:

- с помощью регистра RD/WR_IRQ_CLEAR_RAW - побитовый сброс битов регистра RD/WR_IRQ_STATUS_RAW;

- с помощью регистра RD/WR_IRQ_STATUS_CLEAR – сброс всех незамаскированных битов регистра RD/WR_IRQ_STATUS_RAW.

4.11 Контроллер прерываний GIC

Контроллер прерываний GIC служит для передачи запросов на прерывание процессорному ядру ARM от периферийных устройств процессорного кластера PC. GIC имеет следующие характеристики:

- предоставляет программный интерфейс для взаимодействия процессорного ядра ARM и системой прерываний,
- поддерживает 64 векторных прерывания с настраиваемыми программно приоритетами (32 уровня приоритетов) и маскированием,
- поддерживает 8 программных прерываний.

4.11.1 Устройство системного контроллера прерываний

Блок не имеет сигналов, соединенных с внешними выводами микросхемы. На входы блока GIC подключаются выходы прерываний всех периферийных устройств системы на кристалле, выходы прерываний подключаются непосредственно к процессорному ядру ARM. GIC также имеет два интерфейса с системной шиной AMBA AXI, по которой процессорное ядро может его программировать.

На рисунке ниже (Рисунок 4.32) приведена структурная схема GIC с описанием основных компонентов (Таблица 4.103).

Рисунок 4.32 – Структураня схема GIC

Таблица 4.103 – Основные компоненты GIC

Название	Назначение
Блок обработки запросов от периферийных устройств (Distributor)	производит фиксацию запроса на прерывание, маскирование и предварительный арбитраж
Блок интерфейса с процессорным ядром (CPU Interface)	завершает арбитраж, передаёт запросы процессорному ядру и контролирует процесс обработки запросов

4.11.2 Запросы на прерывание

К входам линий прерываний GIC подключены выходы запросов на прерывание от периферийных устройств. Вместе с восемью программными прерываниями они образуют единый список. В таблице ниже (см. таблицу 4.104) приведён список прерываний процессорной системы ARM, каждому прерыванию в системе поставлен в соответствие свой порядковый номер (Interrupt ID).

Таблица 4.104 – Прерывания ядра ARM процессорного кластера PC

Interrupt ID	Символическое название	Описание	Приоритет по умолчанию
0	COMMRX	Канал отладки Rx от ARM	Высший
1	COMMTX	Канал отладки Tx от ARM	
2	CTIRQ	Прерывание от CTI	
3	-	Зарезервировано	
4	TIMINT0	Прерывание таймера 0	
5	TIMINT1	Прерывание таймера 1	
6	-	Зарезервировано	
7	-	Зарезервировано	
8	CCPU_HP	Высокоприоритетное прерывание от центральной управляющей процессорной системы (CCPU)	
9	PCi_HP	Высокоприоритетное прерывание от кластера PCi	
10	PCj_HP	Высокоприоритетное прерывание от кластера PCj	
11	PCk_HP	Высокоприоритетное прерывание от кластера PCk	
12	NMPU0_HP	Высокоприоритетное прерывание от NMPU0	
13	NMPU1_HP	Высокоприоритетное прерывание от NMPU1	
14	NMPU2_HP	Высокоприоритетное прерывание от NMPU2	
15	NMPU3_HP	Высокоприоритетное прерывание от NMPU3	
16	CP0_SRC_INT	Обобщенное прерывание от передающего канала CP0	
17	CP0_DST_INT	Обобщенное прерывание от принимающего канала CP0	
18	CP0_SRC_INT	Обобщенное прерывание от передающего канала CP1	
19	CP0_DST_INT	Обобщенное прерывание от принимающего канала CP1	
20	CP0_SRC_INT	Обобщенное прерывание от передающего канала CP2	
21	CP0_DST_INT	Обобщенное прерывание от принимающего канала CP2	
22	CP0_SRC_INT	Обобщенное прерывание от передающего канала CP3	
23	CP0_DST_INT	Обобщенное прерывание от принимающего канала CP3	

Продолжение таблицы 4.104

Interrupt ID	Символическое название	Описание	Приоритет по умолчанию
24	MDMAC_INT	Прерывание от контроллера ПДП память-память MDMAC	
25	MPUW_INT	Прерывание от MPU (канал записи)	
26	MPUR_INT	Прерывание от MPU (канал чтения)	
27	-	Зарезервировано	
28	-	Зарезервировано	
29	-	Зарезервировано	
30	GE_INT0	Внешнее прерывание 0	
31	GE_INT1	Внешнее прерывание 1	
32	EMI_INT	Прерывание от контроллера внешней памяти EMI	
33	ELC_INT	Прерывание от контроллера канала ELC	
34	-	Зарезервировано	
35	-	Зарезервировано	
36	-	Зарезервировано	
37	-	Зарезервировано	
38	GE_INT2	Внешнее прерывание 2	
39	GE_INT3	Внешнее прерывание 3	
40	-	Зарезервировано	
41	-	Зарезервировано	
42	-	Зарезервировано	
43	-	Зарезервировано	
44	-	Зарезервировано	
45	-	Зарезервировано	
46	GE_INT4	Внешнее прерывание 4	
47	GE_INT5	Внешнее прерывание 5	
48	CCPU_LP	Низкоприоритетное прерывание от центральной управляющей процессорной системы (CCPU)	
49	PCi_LP	Низкоприоритетное прерывание от кластера PCi	
50	PCj_LP	Низкоприоритетное прерывание от кластера PCj	
51	PCk_LP	Низкоприоритетное прерывание от кластера PCk	
52	NMPU0_LP	Низкоприоритетное прерывание от NMPU0	
53	NMPU1_LP	Низкоприоритетное прерывание от NMPU1	
54	NMPU2_LP	Низкоприоритетное прерывание от NMPU2	
55	NMPU3_LP	Низкоприоритетное прерывание от NMPU3	

Продолжение таблицы 4.104

Interrupt ID	Символическое название	Описание	Приоритет по умолчанию
56	GE_INT6	Внешнее прерывание 6	
57	GE_INT7	Внешнее прерывание 7	
58	-	Зарезервировано	
59	-	Зарезервировано	
60	-	Зарезервировано	
61	PMU_IRQ	Прерывание от монитора производительности ARM	
62	PLLH_RDYINT	Готовность PLL от CRGH	Низший
63	-	Зарезервировано	

Примечания: Для PC0: PCi = PC1, PCj = PC2, PCk = PC3;
для PC1: PCi = PC0, PCj = PC3, PCk = PC2;
для PC2: PCi = PC3, PCj = PC0, PCk = PC1;
для PC3: PCi = PC2, PCj = PC1, PCk = PC0.

Каждому из запросов на прерывание можно присвоить программный приоритет (0-31), записав нужное значение в соответствующий регистр программного приоритета. Если несколько прерываний имеют одинаковый программный приоритет, то для определения порядка их обработки используется аппаратный приоритет. Аппаратный приоритет тем выше, чем ниже порядковый номер прерывания. То есть, программное прерывание 0 имеет высший аппаратный приоритет, а прерывание 95 – низший.

Запросы на прерывание от периферийных устройств могут быть двух типов: по уровню и по фронту.

Для большинства устройств системы на кристалле используется механизм прерываний по уровню. В этом случае считается, что соответствующее периферийное устройство находится в состоянии запроса на прерывание, и линия прерывания активна, до тех пор пока не будет устранена причина данного запроса в периферийном устройстве.

Запрос на прерывание типа «по фронту» фиксируется контроллером прерываний в момент положительного фронта сигнала на линии прерывания.

4.11.3 Интерфейс GIC с процессорным ядром ARM

Входные запросы на прерывание проходят внутри GIC арбитраж и маскирование. Когда какой-нибудь из запросов выигрывает арбитраж, GIC выдаёт процессорному ядру ARM итоговый запрос и ядро переходит в один из двух режимов:

- IRQ (обычное прерывание) или

- FIQ (быстрое прерывание).

4.11.4 Группы прерываний

GIC позволяет программно настроить режим обработки (IRQ или FIQ) для каждой линии прерывания и для каждого программного запроса. Каждое прерывание GIC считается принадлежащим к одной из двух групп: группе 0 или группе 1. После системного сброса все прерывания принадлежат группе 0 и имеют режим обработки IRQ. Далее можно с помощью записи в программно доступные регистры GIC переносить прерывания в группу 1 и обратно.

Разница между двумя группами заключается в том, что

- для группы 0 можно установить режим обработки прерывания FIQ,
- прерывания группы 1 могут обрабатываться в незащищённом режиме ядра ARM,
- для группы 1 рекомендуется устанавливать значения приоритетов 16-32.

4.11.5 Состояния обработки прерываний

По мере обработки каждое прерывание проходит через состояния, показанные на рисунке ниже (см. Рисунок 4.33).

Рисунок 4.33 – Состояния обработки прерываний

Прерывание переходит в одно из состояний ОЖИДАНИЯ (переходы А1 и А2) по любому из следующих событий:

- периферийное устройство выдаёт запрос на прерывание,
- программная запись в регистр перехода в режим ожидания (GICD_ISPENDR),
- корректная запись в регистр программного запроса (GICD_SGIR).

Отмена ожидания (переходы Б1 и Б2) происходит с прерыванием по любому из следующих событий:

- периферийное устройство по какой-то причине самостоятельно снимает запрос на прерывание – справедливо для прерываний по уровню,
- запись в регистр выхода из режима ожидания (GICD_ICPENDR) – справедливо для прерываний по фронту, программных и для прерываний по уровню, если запрос был инициирован программно.

Прерывание переходит в режим ОБРАБОТКИ (переход В) в момент, когда

- процессорное ядро читает регистр подтверждения прерывания (GICC_IAR) и при этом в поле Interrupt ID выдаётся номер данного прерывания.

Переход Г совершается,

- если одновременно выполнены условия перехода В и Д1.

Завершением обработки прерывания (переходы Д1 и Д2) считается

- запись в регистр завершения обработки прерывания (GICC_EOIR).

4.11.6 Вложенные прерывания

Во время, пока какое-либо прерывание (первое) находится в состоянии ОБРАБОТКИ, может возникнуть второй, более приоритетный, запрос на прерывание. Второй запрос передаётся процессорному ядру, только если удовлетворены оба условия:

- приоритет второго запроса выше, чем пороговый уровень приоритета (GICC_PMR),
- приоритет второго запроса выше, чем текущий приоритет обрабатываемого прерывания (GICC_RPR).

После этого процессорное ядро может прочитать регистр GICC_IAR ещё раз, переведя тем самым в состояние обработки второе прерывание. Второе прерывание считается вложенным и его обработка должна быть завершена (запись в регистр GICC_EOIR) раньше завершения первого прерывания.

4.11.7 Регистровая модель системного контроллера прерываний GIC

Программно доступные регистры системного контроллера прерываний GIC расположены в области памяти периферийных устройств REG_CPU, занимают две области памяти по 4 Кб:

- набор регистров блока обработки запросов от периферийных устройств – GICD, базовый адрес GICD_Base = 0x000B_0000,
- набор регистров блока интерфейса с процессорным ядром – GICC, базовый адрес GICC_Base = 0x000B_1000.

Спецификация регистров представлена в таблице ниже (см. таблицу 4.105).

Таблица 4.105 - Спецификация регистров контроллера GIC

Адрес	Тип	Начальное значение	Имя	Описание
GICD_Base+0x000	ЧТ/ЗП	0x00000000	GICD_CTLR	Регистр включения GIC
GICD_Base+0x004	ЧТ	0x00000402	GICD_TYPER	Регистр конфигурации GIC
GICD_Base+0x008	ЧТ	0x0000043B	GICD_IIDR	Регистр идентификации GIC
GICD_Base+0x080	ЧТ/ЗП	0x00000000	GICD_IGROUPR0	Регистр 0 принадлежности к группе
GICD_Base+0x084	ЧТ/ЗП	0x00000000	GICD_IGROUPR1	Регистр 1 принадлежности к группе
GICD_Base+0x088	ЧТ/ЗП	0x00000000	GICD_IGROUPR2	Регистр 2 принадлежности к группе
GICD_Base+0x100	ЧТ/ЗП	0x00000000	GICD_ISENABLER0	Регистр 0 включения прерывания
GICD_Base+0x104	ЧТ/ЗП	0x00000000	GICD_ISENABLER1	Регистр 1 включения прерывания
GICD_Base+0x108	ЧТ/ЗП	0x00000000	GICD_ISENABLER2	Регистр 2 включения прерывания
GICD_Base+0x180	ЧТ/ЗП	0x00000000	GICD_ICENABLER0	Регистр 0 выключения прерывания
GICD_Base+0x184	ЧТ/ЗП	0x00000000	GICD_ICENABLER1	Регистр 1 выключения прерывания
GICD_Base+0x188	ЧТ/ЗП	0x00000000	GICD_ICENABLER2	Регистр 2 выключения прерывания
GICD_Base+0x200	ЧТ/ЗП	0x00000000	GICD_ISPENDR0	Регистр 0 перехода в режим ожидания
GICD_Base+0x204	ЧТ/ЗП	0x00000000	GICD_ISPENDR1	Регистр 1 перехода в режим ожидания
GICD_Base+0x208	ЧТ/ЗП	0x00000000	GICD_ISPENDR2	Регистр 2 перехода в режим ожидания
GICD_Base+0x280	ЧТ/ЗП	0x00000000	GICD_ICPENDR0	Регистр 0 выхода из режима ожидания
GICD_Base+0x284	ЧТ/ЗП	0x00000000	GICD_ICPENDR1	Регистр 1 выхода из режима ожидания
GICD_Base+0x288	ЧТ/ЗП	0x00000000	GICD_ICPENDR2	Регистр 2 выхода из режима ожидания
GICD_Base+0x300	ЧТ	0x00000000	GICD_ICDABR0	Регистр 0 состояния обработки прерывания
GICD_Base+0x304	ЧТ	0x00000000	GICD_ICDABR1	Регистр 1 состояния обработки прерывания
GICD_Base+0x308	ЧТ	0x00000000	GICD_ICDABR2	Регистр 2 состояния обработки прерывания
GICD_Base+0x400	ЧТ/ЗП	0x00000000	GICD_IPRIORITY0	Регистр 0 приоритетов
GICD_Base+0x404	ЧТ/ЗП	0x00000000	GICD_IPRIORITY1	Регистр 1 приоритетов
GICD_Base+0x420	ЧТ/ЗП	0x00000000	GICD_IPRIORITY8	Регистр 8 приоритетов
GICD_Base+0x424	ЧТ/ЗП	0x00000000	GICD_IPRIORITY9	Регистр 9 приоритетов
GICD_Base+0x428	ЧТ/ЗП	0x00000000	GICD_IPRIORITY10	Регистр 10 приоритетов
GICD_Base+0x42C	ЧТ/ЗП	0x00000000	GICD_IPRIORITY11	Регистр 11 приоритетов
GICD_Base+0x430	ЧТ/ЗП	0x00000000	GICD_IPRIORITY12	Регистр 12 приоритетов
GICD_Base+0x434	ЧТ/ЗП	0x00000000	GICD_IPRIORITY13	Регистр 13 приоритетов
GICD_Base+0x438	ЧТ/ЗП	0x00000000	GICD_IPRIORITY14	Регистр 14 приоритетов
GICD_Base+0x43C	ЧТ/ЗП	0x00000000	GICD_IPRIORITY15	Регистр 15 приоритетов
GICD_Base+0x440	ЧТ/ЗП	0x00000000	GICD_IPRIORITY16	Регистр 16 приоритетов
GICD_Base+0x444	ЧТ/ЗП	0x00000000	GICD_IPRIORITY17	Регистр 17 приоритетов
GICD_Base+0x448	ЧТ/ЗП	0x00000000	GICD_IPRIORITY18	Регистр 18 приоритетов
GICD_Base+0x44C	ЧТ/ЗП	0x00000000	GICD_IPRIORITY19	Регистр 19 приоритетов
GICD_Base+0x450	ЧТ/ЗП	0x00000000	GICD_IPRIORITY20	Регистр 20 приоритетов
GICD_Base+0x454	ЧТ/ЗП	0x00000000	GICD_IPRIORITY21	Регистр 21 приоритетов
GICD_Base+0x458	ЧТ/ЗП	0x00000000	GICD_IPRIORITY22	Регистр 22 приоритетов
GICD_Base+0x45C	ЧТ/ЗП	0x00000000	GICD_IPRIORITY23	Регистр 23 приоритетов

Продолжение таблицы 4.105

Адрес	Тип	Начальное значение	Имя	Описание
GICD_Base+0xC00	ЧТ/ЗП	0x0000AAAA	GICD_ICFGR0	Регистр 0 типа запроса
GICD_Base+0xC08	ЧТ/ЗП	0xAAAAAAAA	GICD_ICFGR2	Регистр 2 типа запроса
GICD_Base+0xC0C	ЧТ/ЗП	0xAAAAAAAA	GICD_ICFGR3	Регистр 3 типа запроса
GICD_Base+0xC10	ЧТ/ЗП	0xAAAAAAAA	GICD_ICFGR4	Регистр 4 типа запроса
GICD_Base+0xC14	ЧТ/ЗП	0xAAAAAAAA	GICD_ICFGR5	Регистр 5 типа запроса
GICD_Base+0xD04	ЧТ	0x00000000	GICD_IRAWST1	Регистр 1 состояния запросов от периферийных устройств
GICD_Base+0xD08	ЧТ	0x00000000	GICD_IRAWST2	Регистр 2 состояния запросов от периферийных устройств
GICD_Base+0xF00	ЗП	0x00000000	GICD_SGIR	Регистр программных запросов на прерывание
GICC_Base+0x000	ЧТ/ЗП	0x00000000	GICC_CTLR	Регистр управления интерфейсом с процессорным ядром
GICC_Base+0x004	ЧТ/ЗП	0x00000000	GICC_PMR	Регистр фильтра приоритетов
GICC_Base+0x008	ЧТ/ЗП	0x00000002	GICC_BPR	Регистр интервалов приоритета
GICC_Base+0x00C	ЧТ	0x0000007F	GICC_IAR	Регистр подтверждения прерывания
GICC_Base+0x010	ЗП	-	GICC_EOIR	Регистр завершения обработки прерывания
GICC_Base+0x014	ЧТ	0x000000FF	GICC_RPR	Регистр текущего приоритета обрабатываемого прерывания
GICC_Base+0x018	ЧТ	0x0000007F	GICC_HPPIR	Регистр наиболее приоритетного прерывания в состоянии ОЖИДАНИЯ
GICC_Base+0x01C	ЧТ/ЗП	0x00000003	GICC_ABPR	Зеркало регистра интервалов приоритета

Особенности доступа к регистрам GIC

Процессорное ядро ARM может находиться в двух состояниях: защищённое (secure) и незащищённое (non-secure). При каждом доступе в программно доступные регистры информация о состоянии ядра передаётся в GIC, соответственно, GIC различает защищённый доступ и незащищённый. Для некоторых регистров поведение при защищённом доступе отличается от поведения при незащищённом.

Доступ в регистры GIC необходимо делать с помощью инструкций чтения и записи 32-разрядного слова (LDR, STR). Исключение составляют регистры GICD_IPRIORITY, в которые разрешён байтовый доступ (LDRB, STRB).

Регистр включения GIC (GICD_CTLR)

Таблица 4.106 - Формат регистра GICD_CTLR

Биты	Название	Тип	Описание
[31:1]	Зарезервировано	-	
[0]	Enable	ЧТ/ЗП	Бит включения блока обработки запросов. Значение 0 – прерывания не передаются в интерфейсный блок, значение 1 разрешает прерываниям из состояния ожидания переходить в состояние обработки.

Физически имеется два экземпляра регистра GICD_CTLR - для защищённого режима и для незащищённого:

- если процессорное ядро обращается в защищённом режиме, то производится доступ к экземпляру, управляющему прерываниями группы 0;
- если процессорное ядро обращается в незащищённом режиме, то производится доступ к экземпляру, управляющему группой 1.

Регистр конфигурации GIC (GICD_TYPER)

Таблица 4.107 - Формат регистра GICD_TYPER

Биты	Название	Тип	Описание
[31:11]	Зарезервировано	-	
[10]	SecurityExtn	ЧТ	Контроллер поддерживает расширения безопасности, поэтому данный бит имеет значение 1
[9:8]	Зарезервировано	-	
[7:5]	CPUNumber	ЧТ	Количество интерфейсов с процессорными ядрами, данное поле имеет значение 000 – один интерфейс
[4:0]	ITLinesNumber	ЧТ	Данный экземпляр GIC поддерживает до 96 прерываний, поэтому данное поле имеет значение 2

Данный регистр нужен для того, чтобы универсальное ПО определяло конфигурацию GIC.

Регистр идентификации GIC (GICD_IIDR)

Таблица 4.108 - Формат регистра GICD_IIDR

Биты	Название	Тип	Описание
[31:16]	Зарезервировано	-	
[15:12]	Revision	ЧТ	Номер ревизии
[11:0]	Implementer	ЧТ	Код разработчика блока по JEP106 (0x0000043B – код фирмы ARM)

Регистры принадлежности к группе (GICD_IGROUPR0 – GICD_IGROUPR2)

Таблица 4.109 - Формат регистров GICD_IGROUPR0 – GICD_IGROUPR2

Биты	Название	Тип	Описание
[31:0]	Group	ЧТ/ЗП	Каждый бит регистра соответствует одной линии прерывания. Значение 0 – прерывание принадлежит к группе 0, значение 1 – прерывание принадлежит к группе 1.

Данные регистры доступны только в защищённом режиме.

Имеются 3 экземпляра регистров:

- в регистре GICD_IGROUPR0 доступны только биты 0-7 (программные прерывания 0-7 соответственно), биты 8-31 зарезервированы,
- в регистре GICD_IGROUPR1 биты 0-31 соответствуют прерываниям 32-63 соответственно,
- в регистре GICD_IGROUPR2 биты 0-31 соответствуют прерываниям 64-95 соответственно.

Регистры включения прерываний (GICD_ISENABLER0 – GICD_ISENABLER2)

Таблица 4.110 - Формат регистров GICD_ISENABLER0 – GICD_ISENABLER2

Биты	Название	Тип	Описание
[31:0]	Set-enable	ЧТ/ЗП	Каждый бит регистра соответствует одной линии прерывания. Запись значения 1 в бит регистра включает соответствующее прерывание, запись значения 0 в бит регистра никак не влияет на состояние соответствующего прерывания. При чтении значение 0 означает, что соответствующее прерывание выключено, значение 1 – включено.

В незащищённом режиме для чтения и записи недоступны биты прерываний, относящихся к группе 0.

Имеются 3 экземпляра регистров:

- в регистре GICD_ISENABLER0 доступны только биты 0-7 (программные прерывания 0-7 соответственно), биты 8-31 зарезервированы,
- в регистре GICD_ISENABLER1 биты 0-31 соответствуют прерываниям 32-63 соответственно,
- в регистре GICD_ISENABLER2 биты 0-31 соответствуют прерываниям 64-95 соответственно.

Регистры выключения прерываний (GICD_ICENABLER0 – GICD_ICENABLER2)

Таблица 4.111 - Формат регистров GICD_ICENABLER0 – GICD_ICENABLER2

Биты	Название	Тип	Описание
[31:0]	Clear-enable	ЧТ/ЗП	Каждый бит регистра соответствует одной линии прерывания. Запись значения 1 в бит регистра выключает соответствующее прерывание, запись значения 0 в бит регистра никак не влияет на состояние соответствующего прерывания. При чтении значение 0 означает, что соответствующее прерывание выключено, значение 1 – включено.

В незащищённом режиме для чтения и записи недоступны биты прерываний, относящихся к группе 0.

Имеются 3 экземпляра регистров:

- в регистре GICD_ICENABLER0 доступны только биты 0-7 (программные прерывания 0-7 соответственно), биты 8-31 зарезервированы,
- в регистре GICD_ICENABLER1 биты 0-31 соответствуют прерываниям 32-63 соответственно,
- в регистре GICD_ICENABLER2 биты 0-31 соответствуют прерываниям 64-95 соответственно.

Регистры перехода в режим ожидания (GICD_ISPENDR0 – GICD_ISPENDR2)

Таблица 4.112 - Формат регистров GICD_ISPENDR0 – GICD_ISPENDR2

Биты	Название	Тип	Описание
[31:0]	Set-pending	ЧТ/ЗП	Каждый бит регистра соответствует одной линии прерывания. Запись значения 1 в бит регистра переводит соответствующее прерывание из неактивного состояния в состояние ОЖИДАНИЯ, запись значения 0 в бит регистра никак не влияет на состояние соответствующего прерывания. При чтении значение 0 означает, что соответствующее прерывание не находится в состоянии ОЖИДАНИЯ, значение 1 – прерывание находится в состоянии ОЖИДАНИЯ.

В незащищённом режиме для чтения и записи недоступны биты прерываний, относящихся к группе 0.

Имеются 3 экземпляра регистров:

- в регистре GICD_ISPENDR0 доступны только биты 0-7 (программные прерывания 0-7 соответственно), биты 8-31 зарезервированы,
- в регистре GICD_ISPENDR1 биты 0-31 соответствуют прерываниям 32-63 соответственно,
- в регистре GICD_ISPENDR2 биты 0-31 соответствуют прерываниям 64-95 соответственно.

Регистры выхода из режима ожидания (GICD_ICPENDR0 – GICD_ICPENDR2)

Таблица 4.113 - Формат регистров GICD_ICPENDR0 – GICD_ICPENDR2

Биты	Название	Тип	Описание
[31:0]	Clear-pending	ЧТ/ЗП	Каждый бит регистра соответствует одной линии прерывания. Запись значения 1 в бит регистра переводит соответствующее прерывание из состояния ОЖИДАНИЯ в неактивное состояние, запись значения 0 в бит регистра никак не влияет на состояние соответствующего прерывания. При чтении значение 0 означает, что соответствующее прерывание не находится в состоянии ОЖИДАНИЯ, значение 1 – прерывание находится в состоянии ОЖИДАНИЯ.

В незащищённом режиме для чтения и записи недоступны биты прерываний, относящихся к группе 0.

Имеются 3 экземпляра регистров:

- в регистре GICD_ICPENDR0 доступны только биты 0-7 (программные прерывания 0-7 соответственно), биты 8-31 зарезервированы,
- в регистре GICD_ICPENDR1 биты 0-31 соответствуют прерываниям 32-63 соответственно,
- в регистре GICD_ICPENDR2 биты 0-31 соответствуют прерываниям 64-95 соответственно.

Регистры состояния обработки прерывания (GICD_ICDABR0 – GICD_ICDABR2)

Таблица 4.114 - Формат регистров GICD_ICDABR0 – GICD_ICDABR2

Биты	Название	Тип	Описание
[31:0]	Active	ЧТ	Каждый бит регистра соответствует одной линии прерывания. При чтении значение 0 означает, что соответствующее прерывание не находится в состоянии ОБРАБОТКИ, значение 1 – прерывание находится в состоянии ОБРАБОТКИ.

В незащищённом режиме для чтения недоступны биты прерываний, относящихся к группе 0.

Имеются 3 экземпляра регистров:

- в регистре GICD_ICDABR0 доступны только биты 0-7 (программные прерывания 0-7 соответственно), биты 8-31 зарезервированы,
- в регистре GICD_ICDABR1 биты 0-31 соответствуют прерываниям 32-63 соответственно,
- в регистре GICD_ICDABR2 биты 0-31 соответствуют прерываниям 64-95 соответственно.

Регистры приоритетов (GICD_IPRIORITY0 – GICD_IPRIORITY23)

Каждый регистр GICD_IPRIORITY содержит четыре восьмиразрядных поля, каждое поле соответствует одной линии прерывания. Возможные значения поля: 0-31. Максимальный приоритет – 0, минимальный – 31. В таблице ниже (см. таблицу 4.115) показано, как поля приоритетов распределены по регистрам.

Таблица 4.115 - Формат регистров GICD_IPRIORITY

Номер преры-вания, Interrupt ID	Экземпляр регистра GICD_IPRIORITY	Поле	Номер преры-вания, Interrupt ID	Экземпляр регистра GICD_IPRIORITY	Поле
0	GICD_IPRIORITY0	Биты 7-3	1	GICD_IPRIORITY0	Биты 15-11
2		Биты 23-19	3		Биты 31-27
4	GICD_IPRIORITY1	Биты 7-3	5	GICD_IPRIORITY1	Биты 15-11
6		Биты 23-19	7		Биты 31-27
-	-	-	-	-	-
32	GICD_IPRIORITY8	Биты 7-3	33	GICD_IPRIORITY8	Биты 15-11
34		Биты 23-19	35		Биты 31-27
36	GICD_IPRIORITY9	Биты 7-3	37	GICD_IPRIORITY9	Биты 15-11
38		Биты 23-19	39		Биты 31-27
40	GICD_IPRIORITY10	Биты 7-3	41	GICD_IPRIORITY10	Биты 15-11
42		Биты 23-19	43		Биты 31-27
44	GICD_IPRIORITY11	Биты 7-3	45	GICD_IPRIORITY11	Биты 15-11
46		Биты 23-19	47		Биты 31-27
48	GICD_IPRIORITY12	Биты 7-3	49	GICD_IPRIORITY12	Биты 15-11
50		Биты 23-19	51		Биты 31-27
52	GICD_IPRIORITY13	Биты 7-3	53	GICD_IPRIORITY13	Биты 15-11
54		Биты 23-19	55		Биты 31-27
56	GICD_IPRIORITY14	Биты 7-3	57	GICD_IPRIORITY14	Биты 15-11
58		Биты 23-19	59		Биты 31-27
60	GICD_IPRIORITY15	Биты 7-3	61	GICD_IPRIORITY15	Биты 15-11
62		Биты 23-19	63		Биты 31-27
64	GICD_IPRIORITY16	Биты 7-3	65	GICD_IPRIORITY16	Биты 15-11
66		Биты 23-19	67		Биты 31-27
68	GICD_IPRIORITY17	Биты 7-3	69	GICD_IPRIORITY17	Биты 15-11
70		Биты 23-19	71		Биты 31-27
72	GICD_IPRIORITY18	Биты 7-3	73	GICD_IPRIORITY18	Биты 15-11
74		Биты 23-19	75		Биты 31-27
76	GICD_IPRIORITY19	Биты 7-3	77	GICD_IPRIORITY19	Биты 15-11
78		Биты 23-19	79		Биты 31-27
80	GICD_IPRIORITY20	Биты 7-3	81	GICD_IPRIORITY20	Биты 15-11
82		Биты 23-19	83		Биты 31-27
84	GICD_IPRIORITY21	Биты 7-3	85	GICD_IPRIORITY21	Биты 15-11
86		Биты 23-19	87		Биты 31-27
88	GICD_IPRIORITY22	Биты 7-3	89	GICD_IPRIORITY22	Биты 15-11
90		Биты 23-19	91		Биты 31-27
92	GICD_IPRIORITY23	Биты 7-3	93	GICD_IPRIORITY23	Биты 15-11
94		Биты 23-19	95		Биты 31-27

В незащищённом режиме для чтения и записи недоступны поля приоритетов прерываний, относящихся к группе 0.

Регистры типа запроса (GICD_ICFGR0 – GICD_ICFGR5)

Каждый регистр GICD_ICFGR содержит 16 двухразрядных полей. Значения поля: b00 – прерывание типа «по уровню», b10 – прерывание типа «по фронту». В таблице ниже (см. таблицу 4.116) показано, как поля распределены по регистрам.

Таблица 4.116 - Формат регистров GICD_ICFGR

Номер прерывания (Interrupt ID)	Экземпляр регистра GICD_ICFGR	Поле	Номер прерывания (Interrupt ID)	Экземпляр регистра GICD_ICFGR	Поле
0	GICD_ICFGR0	Биты 1-0	1	GICD_ICFGR0	Биты 3-2
2		Биты 5-4	3		Биты 7-6
4		Биты 9-8	5		Биты 11-10
6		Биты 13-12	7		Биты 15-14
-	-	-	-	-	-
32	GICD_ICFGR2	Биты 1-0	33	GICD_ICFGR2	Биты 3-2
34		Биты 5-4	35		Биты 7-6
36		Биты 9-8	37		Биты 11-10
38		Биты 13-12	39		Биты 15-14
40		Биты 17-16	41		Биты 19-18
42		Биты 21-20	43		Биты 23-22
44		Биты 25-24	45		Биты 27-26
46		Биты 29-28	47		Биты 31-30
48	GICD_ICFGR3	Биты 1-0	49	GICD_ICFGR3	Биты 3-2
50		Биты 5-4	51		Биты 7-6
52		Биты 9-8	53		Биты 11-10
54		Биты 13-12	55		Биты 15-14
56		Биты 17-16	57		Биты 19-18
58		Биты 21-20	59		Биты 23-22
60		Биты 25-24	61		Биты 27-26
62		Биты 29-28	63		Биты 31-30
64	GICD_ICFGR4	Биты 1-0	65	GICD_ICFGR4	Биты 3-2
66		Биты 5-4	67		Биты 7-6
68		Биты 9-8	69		Биты 11-10
70		Биты 13-12	71		Биты 15-14
72		Биты 17-16	73		Биты 19-18
74		Биты 21-20	75		Биты 23-22
76		Биты 25-24	77		Биты 27-26
78		Биты 29-28	79		Биты 31-30
80	GICD_ICFGR5	Биты 1-0	81	GICD_ICFGR5	Биты 3-2
82		Биты 5-4	83		Биты 7-6
84		Биты 9-8	85		Биты 11-10
86		Биты 13-12	87		Биты 15-14
88		Биты 17-16	89		Биты 19-18
90		Биты 21-20	91		Биты 23-22
92		Биты 25-24	93		Биты 27-26
94		Биты 29-28	95		Биты 31-30

В незащищённом режиме для чтения и записи доступны только поля приоритетов прерываний, относящихся к группе 1. При этом доступны только старшие 4 бита каждого поля и значения 0-15 соответствуют приоритетам 16-32 соответственно.

Регистры состояния запросов от периферийных устройств (GICD_IRAWST1 и GICD_IRAWST2)

Таблица 4.117 - Формат регистров GICD_IRAWST1 и GICD_IRAWST2

Биты	Название	Тип	Описание
[31:0]	RawStatus	ЧТ	Каждый бит регистра соответствует одной линии прерывания. Регистр показывает состояние линии прерывания на входе в GIC: 0 – нет запроса, 1 – есть запрос.

Регистр доступен только в защищённом режиме. Значение битов, соответствующих прерываниям типа «по фронту», не определено.

Имеются 2 экземпляра регистров:

- в регистре GICD_IRAWST1 биты 0-31 соответствуют прерываниям 32-63 соответственно,
- в регистре GICD_IRAWST2 биты 0-31 соответствуют прерываниям 64-94 соответственно.

Регистр программных запросов на прерывание (GICD_SGIR)

Таблица 4.118 - Формат регистра GICD_SGIR

Биты	Название	Тип	Описание
[31:26]	Зарезервировано	-	
[25:24]	TargetListFilter	ЗП	В это поле следует всегда записывать b00
[23:16]	CPUTargetList	ЗП	В это поле следует всегда записывать b00000001
[15]	NSATT	ЗП	Бит незащищённого режима. Запрос на прерывание генерируется, только если справедливо одно из условий: записываемое значение данного бита – 0, запись в данный регистр производится в защищённом режиме и прерывание, выбранное полем SGIINTID, отнесено к группе 0, записываемое значение данного бита – 1, запись в данный регистр производится в защищённом режиме и прерывание, выбранное полем SGIINTID, отнесено к группе 1, записываемое значение данного бита – любое, запись в данный регистр производится в незащищённом режиме и прерывание, выбранное полем SGIINTID, отнесено к группе 1.
[14:4]	Зарезервировано	-	
[3:0]	SGIINTID	ЗП	Номер программного прерывания (Interrupt ID), на которое будет сделан запрос. Допустимые значения: 0-7 (прерывания 0-7 соответственно).

Запись в данный регистр инициирует запрос на программное прерывание.

Регистр управления интерфейсом с процессорным ядром (GICC_CTLR)

Таблица 4.119 - Формат регистра GICC_CTLR

Биты	Название	Тип	Описание
[31:5]	Зарезервировано	-	
[4]	SBPR	ЧТ/ЗП	Бит определяет, каким регистром задаётся размер интервала приоритетов в группе 1. Значение 0 – экземпляром регистра GICC_BPR для группы 1 (или, что то же самое, регистром GICC_ABPR), значение 1 – экземпляром регистра GICC_BPR для группы 0 (то есть одинаковое разбиение для группы 0 и группы 1).
[3]	FIQEn	ЧТ/ЗП	Бит режима прерывания. Значение 0 – прерывания группы 0 обрабатываются ядром ARM в режиме IRQ, значение 1 – в режиме FIQ.
[2]	AckCtl	ЧТ/ЗП	Управляет режимом подтверждения прерываний группы 1. Влияет на поведение регистров GICC_IAR и GICC_EOIR.
[1]	EnableGrp1	ЧТ/ЗП	Включение прерываний группы 1. Значение 0 – выключены, значение 1 – включены.
[0]	EnableGrp0	ЧТ/ЗП	Включение прерываний группы 0. Значение 0 – выключены, значение 1 – включены.

Регистр фильтра приоритетов (GICC_PMR)

Представление данного регистра различается при защищённом и незащищённом доступе. В защищённом режиме можно установить текущий пороговый уровень в значение от 0 до 31.

Таблица 4.120 - Формат регистра GICC_PMR в защищённом режиме

Биты	Название	Тип	Описание
[31:8]	Зарезервировано	-	
[7:3]	Priority	ЧТ/ЗП	Пороговый уровень приоритета. Все запросы на прерывание с приоритетом ниже или равным значению данного поля маскируются.
[2:0]	Зарезервировано	-	

В незащищённом режиме можно установить текущий пороговый уровень в значение от 16 до 31.

Таблица 4.121 - Формат регистра GICC_PMR в незащищённом режиме

Биты	Название	Тип	Описание
[31:8]	Зарезервировано	-	
[7:4]	Priority	ЧТ/ЗП	Пороговый уровень приоритета. При этом считается, что старший бит приоритета равен 1 (то есть значения 0-15 данного поля устанавливают приоритет 16-31 соответственно).
[2:0]	Зарезервировано	-	

Если в защищённом режиме запрограммирован текущий приоритет от 0 до 15, то любая незащищённая запись не проходит, а незащищённое чтение выдаёт 0.

Регистр интервалов приоритета (GICC_BPR)

Таблица 4.122 - Формат регистра GICC_BPR

Биты	Название	Тип	Описание
[31:3]	Зарезервировано	-	
[2:0]	BinaryPoint	ЧТ/ЗП	Длина интервала приоритетов

Регистр GICC_BPR разбивает все возможные значения приоритетов прерываний на интервалы одинаковой длины, которые используются в механизме вложенных прерываний. Длиной интервала считается количество разных значений приоритетов в одном интервале. Физически имеется два экземпляра регистра GICC_BPR: регистр группы 0 (Таблица 4.123) и регистр группы 1 (Таблица 4.124).

Таблица 4.123 - Кодировка значений регистра GICC_BPR группы 0

Значение поля BinaryPoint	Длина интервала	Количество интервалов	Разделение приоритетов по интервалам
0	-	-	Зарезервировано
1	-	-	Зарезервировано
2	1	32	0, 1, ..., 31 – по одному приоритету в интервале
3	2	16	0-1, 2-3, ..., 30-31
4	4	8	0-3, 4-7, ..., 28-31
5	8	4	0-7, 8-15, 16-32, 24-31
6	16	2	0-15, 16-31
7	32	1	0-31 – все приоритеты в одном интервале

Прерывания группы 1 имеют приоритеты с 16 по 31, поэтому кодировка регистра GICC_BPR для группы 1 отличается.

Таблица 4.124 - Кодировка значений регистра GICC_BPR группы 1

Значение поля BinaryPoint	Длина интервала	Количество интервалов	Разделение приоритетов по интервалам
0	-	-	Зарезервировано
1	-	-	Зарезервировано
2	-	-	Зарезервировано
3	1	16	16, 17, ..., 31 – по одному приоритету в интервале
4	2	8	16-17, 18-19, ..., 30-31
5	4	4	16-19, 20-23, 24-27, 28-31
6	8	2	16-23, 24-31
7	16	1	16-31 – все приоритеты группы 1 в одном интервале

Не допускается записывать в регистр GICC_BPR зарезервированные значения.

Экземпляр регистра группы 1 может не использоваться (см. описание бита GICC_CTLR[SBPR]).

Регистр подтверждения прерывания (GICC_IAR)

Таблица 4.125 - Формат регистра GICC_IAR

Биты	Название	Тип	Описание
[31:7]	Зарезервировано	-	
[6:0]	InterruptID	ЧТ	Номер самого приоритетного на данный момент прерывания.

Чтение из данного регистра

- возвращает номер наиболее приоритетного на данный момент прерывания,
- переводит это прерывание в состояние ОБРАБОТКИ.

Описанные действия производятся контроллером, только в ответ на валидное чтение.

Если наиболее приоритетным на данный момент является прерывание из группы 0, то валидным считается чтение только в защищённом режиме процессора. Если из группы 1, то возможны варианты:

- чтение в незащищённом режиме валидно,
- если бит GICC_CTLR[AckCtl] заранее установлен в 1, то валидно чтение в защищённом режиме.

При любых других условиях чтение считается невалидным. Невалидное чтение не влияет на внутреннее состояние контроллера и возвращает

- значение 1022 в случае, когда наиболее приоритетное прерывание принадлежит группе 1, чтение в защищённом режиме и бита GICC_CTLR[AckCtl]=0,
- значение 1023 во всех других случаях.

Регистр завершения обработки прерывания (GICC_EOIR)

Таблица 4.126 - Формат регистра GICC_EOIR

Биты	Название	Тип	Описание
[31:10]	Зарезервировано	-	
[9:0]	EOIINTID	ЗП	Номер прерывания

Валидная запись в данный регистр переводит прерывание с номером EOIINTID из режима ОБРАБОТКИ в неактивное состояние.

Запись считается валидной если справедливо одно из условий:

- прерывание с номером EOIINTID принадлежит группе 0 и запись производится в защищённом режиме,
- прерывание с номером EOIINTID принадлежат группе 1 и запись производится в незащищённом режиме,
- прерывание с номером EOIINTID принадлежат группе 1, запись производится в защищённом режиме и бит GICC_CTLR[AckCtl]=1.

Любая другая запись считается невалидной, кроме того

- запись в защищённом режиме при значении бита GICC_CTLR[AckCtl]=0 недопустима,
- при AckCtl=0 записываемое в защищённом режиме значение EOIINTID должно соответствовать последнему прочитанному в защищённом режиме значению InterruptID регистра GICC_IAR,
- при AckCtl=1 записываемое в защищённом режиме значение EOIINTID должно соответствовать последнему прочитанному в любом режиме значению InterruptID регистра GICC_IAR,
- записываемое в незащищённом режиме значение EOIINTID должно соответствовать последнему прочитанному в незащищённом режиме значению InterruptID регистра GICC_IAR.

Регистр текущего приоритета обрабатываемого прерывания (GICC_RPR)

Представление данного регистра различается при защищённом и незащищённом доступе.

Таблица 4.127 - Формат регистра GICC_RPR в защищённом режиме

Биты	Название	Тип	Описание
[31:8]	Зарезервировано	-	
[7:3]	Priority	ЧТ	Приоритет прерывания, находящегося в состоянии ОБРАБОТКИ. Возвращается наивысший приоритет из интервала, в котором находится наиболее приоритетное на данный момент прерывание.
[2:0]	Зарезервировано	-	

Таблица 4.128 - Формат регистра GICC_RPR в незащищённом режиме

Биты	Название	Тип	Описание
[31:8]	Зарезервировано	-	
[7:4]	Priority	ЧТ	Приоритет прерывания группы 1, находящегося в состоянии ОБРАБОТКИ. При этом считается, что старший бит приоритета равен 1 (то есть значения 0-15 данного поля означает приоритет 16-31 соответственно). Если в состоянии обработки находится прерывание группы 0, то данное поле возвращает 0.
[2:0]	Зарезервировано	-	

Регистр наиболее приоритетного прерывания в состоянии ОЖИДАНИЯ (GICC_HPPIR)

Таблица 4.129 - Формат регистра GICC_HPPIR

Биты	Название	Тип	Описание
[31:10]	Зарезервировано	-	
[9:0]	PENDINTID	ЧТ	Номер наиболее приоритетного прерывания в состоянии ОЖИДАНИЯ.

При защищённом чтении регистр возвращает:

- номер наиболее приоритетного прерывания из всех находящихся в состоянии ОЖИДАНИЯ, если бит GICC_CTLR[AckCtl]=1,
- номер наиболее приоритетного прерывания из группы 0, находящегося в состоянии ОЖИДАНИЯ,
- значение 1022, если в состоянии ожидания находятся только прерывания группы 1,
- значение 1023 во всех остальных случаях.

При незащищённом чтении регистр возвращает:

- номер наиболее приоритетного прерывания из группы 1, находящегося в состоянии ОЖИДАНИЯ,
- значение 1023.

Зеркало регистра интервалов приоритета (GICC_ABPR)

Зеркало регистра интервалов приоритета позволяет в защищённом режиме производить доступ в экземпляр регистра интервала приоритетов для группы прерываний 1 (доступный по адресу регистра GICC_BPR только в незащищённом режиме).

4.11.8 Порядок работы с GIC

В данном разделе описывается пример работы с GIC в простом случае: все прерывания обрабатываются в режиме IRQ, настройки приоритетов оставлены по умолчанию. Представлен порядок действий программы, выполняемой на процессорном ядре ARM непосредственно связанном с данным GIC.

Порядок инициализации после системного сброса:

- запись в регистры GICD_ICFGR информации о типе запросов от каждого периферийного устройства: по фронту или по уровню,
- запись в регистры GICD_IGROUPR значения 0 – все прерывание относятся к группе 0,
- запись в регистры GICD_ISENABLER информации о том, какие прерывания следует включить,
- запись в регистр GICC_PMR значения 0xFF – разрешение прерываний любого приоритета,
- запись в регистр GICC_CTLR значения 1 – включение блока интерфейса с процессорным ядром,

- запись в регистр GICD_CTLR значения 1 – включение блока обработки запросов от периферийных устройств,
- разрешение прерывания IRQ в процессорном ядре ARM (если уже не включено).

Порядок обработки прерывания:

- при получении запроса ядро ARM переходит в режим IRQ и выполняет первую инструкцию общего обработчика, расположенную по адресу 0x00000018 (или в другом месте, если переопределено при инициализации ядра ARM),
- общий обработчик читает регистр GICC_IAR, чтобы определить номер прерывания (InterruptID),
- общий обработчик вызывает обработчик, ассоциированный с полученным номером прерывания,
- ассоциированный обработчик совершает необходимые действия с периферийным устройством, обычно, добиваясь, что периферийное устройство снимает свой запрос, и возвращает управление общему обработчику,
- общий обработчик записывает в регистр GICC_EOIR номер прерывания,
- общий обработчик делает возврат из режима IRQ.

4.12 Блок сдвоенных таймеров DIT

Блок сдвоенных таймеров кластера имеет следующие характеристики:

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

4.12.1 Устройство блока сдвоенных таймеров

Блок не имеет сигналов, соединенных с внешними выводами микросхемы. На рисунке 4.34 приведена структурная схема этого блока.

Рисунок 4.34 – Структурная схема блока сдвоенных таймеров

Блок состоит из двух идентичных блоков Timer 1 и Timer 2, каждый из которых может функционировать в 16 или 32-битном режиме. Внутри каждого блока находится счетчик FRC (Free Running Counter).

Оба таймера могут быть программно настроены на работу в одном из следующих режимов:

- свободный счет (free-running) – счетчик таймера постоянно декрементируется, счет автоматически начинается с максимального значения после достижения нуля;
- периодичный (periodic) – аналогично предыдущему, только после достижения нуля счет начинается со значения, предварительно загруженного в регистр TimerXLoad;

- одиночный (one-shot) – счетчик начинает декрементироваться со значения, загруженного в регистр TimerXLoad, после достижения нуля счет останавливается.

Ниже приведена последовательность работы с таймерами:

1. Необходимо провести начальные настройки используемого таймера, т.е. прописать поля регистра TimerXControl.
2. Загрузить начальное значение счетчика в регистр TimerXLoad.
3. 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.
4. 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.
5. Текущее значение счетчика может быть считано из регистра TimerXValue в любой момент времени.
6. 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.

4.12.2 Регистровая модель блока сдвоенных таймеров

Программно доступные регистры блока сдвоенных таймеров DIT расположены в области памяти периферийных устройств кластера REG_CPU и имеют базовый адрес DIT Base = 0x000C_D000 hex и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже.

Таблица 4.130 - Спецификация регистров блока DIT

Адрес (байтовый доступ)	Тип	Разрядность	Начальное значение	Имя	Описание
0x000C_D000 hex	ЧТ/ЗП	32	0x00000000	Timer1Load	Регистр загрузки. TimerXLoad
0x000C_D004 hex	ЧТ	32	0xFFFFFFFF	Timer1Value	Регистр текущего значения. TimerXValue
0x000C_D008 hex	ЧТ/ЗП	8	0x20	Timer1Control	Регистр управления. TimerXControl
0x000C_D00C hex	ЗП	-	-	Timer1IntClr	Регистр снятия прерываний. TimerXIntClr
0x000C_D010 hex	ЧТ	1	0x0	Timer1RIS	Регистр состояния прерывания до наложения маски. TimerXRIS
0x000C_D014 hex	ЧТ	1	0x0	Timer1MIS	Регистр состояния прерывания после наложения маски. TimerXMIS
0x000C_D018 hex	ЧТ/ЗП	32	0x00000000	Timer1BGLoad	Фоновый регистр загрузки. TimerXBGLoad
0x000C_D020 hex	ЧТ/ЗП	32	0x00000000	Timer2Load	Регистр загрузки. TimerXLoad
0x000C_D024 hex	ЧТ	32	0xFFFFFFFF	Timer2Value	Регистр текущего значения. TimerXValue
0x000C_D028 hex	ЧТ/ЗП	8	0x20	Timer2Control	Регистр управления. TimerXControl
0x000C_D02C hex	ЗП	-	-	Timer2IntClr	Регистр снятия прерываний. TimerXIntClr
0x000C_D030 hex	ЧТ	1	0x0	Timer2RIS	Регистр состояния прерывания до наложения маски. TimerXRIS
0x000C_D034hex	ЧТ	1	0x0	Timer2MIS	Регистр состояния прерывания после наложения маски. TimerXMIS
0x000C_D038 hex	ЧТ/ЗП	32	0x00000000	Timer2BGLoad	Фоновый регистр загрузки. TimerXBGLoad
0x000C_D03C hex - 0x000C_DEFC hex	-	-	-	-	Зарезервировано
0x000C_DF00 hex	ЧТ/ЗП	1	0x0	TimerITCR	Регистр управления тестом интеграции. TimerITCR
0x000C_DF04 hex	ЗП	2	0x0	TimerITOP	Регистр управления тестом интеграции. TimerITOP
0x000C_DF08 hex - 0x000C_DFDC hex	-	-	-	-	Зарезервировано

Продолжение таблицы 4.130

Адрес (байтовый доступ)	Тип	Разрядность	Начальное значение	Имя	Описание
0x000C_DFE0 hex	ЧТ	8	0x04	TimerPeriphID0	Идентификатор периферийного устройства. TimerPeriphID0 биты [7:0]
0x000C_DFE4 hex	ЧТ	8	0x18	TimerPeriphID1	Идентификатор периферийного устройства. TimerPeriphID1 биты [15:8]
0x000C_DFE8 hex	ЧТ	8	0x04	TimerPeriphID2	Идентификатор периферийного устройства. TimerPeriphID2 биты [23:16]
0x000C_DFEC hex	ЧТ	8	0x00	TimerPeriphID3	Идентификатор периферийного устройства. TimerPeriphID3 биты [31:24]
0x000C_DFF0 hex	ЧТ	8	0x0D	TimerPCellID0	PrimeCell идентификатор. TimerPCellID0 биты [7:0]
0x000C_DFF4 hex	ЧТ	8	0xF0	TimerPCellID1	PrimeCell идентификатор. TimerPCellID1 биты [15:8]
0x000C_DFF8 hex	ЧТ	8	0x05	TimerPCellID2	PrimeCell идентификатор. TimerPCellID2 биты [23:16]
0x000C_DFFC hex	ЧТ	8	0xB1	TimerPCellID3	PrimeCell идентификатор. TimerPCellID3 биты [31:24]

Далее приведено подробное описание каждого регистра. Заметим, что использование X в названии регистров означает, что регистр относится к Timer 1, либо к блоку Timer 2.

Регистр загрузки. TimerXLoad

Регистр TimerXLoad - это 32-х битный регистр, в который программно записывается значение, начиная с которого счетчик должен декрементироваться. Если периодический режим активен и текущее значение достигает нуля, то происходит перезагрузка счетчика.

При записи, текущее значение счетчика незамедлительно меняется на значение, записанное по активному фронту синхросигнала TIMCLK.

Отметим, что минимальным значением для TimerXLoad является 1. Если значение TimerXLoad установлено в 0, то прерывания вырабатываются незамедлительно.

Значение данного регистра всегда переписывается, если прописано значение регистра TimerXBGLoad. Но текущее значение счета изменяется не сразу. Если значения записываются сразу в оба регистра: TimerXLoad и TimerXBGLoad, до прихода активного фронта сигнала TIMCLK, тогда по следующему активному фронту сигнала TIMCLK значение, записанное в регистре TimerXLoad, заменяется текущим значением счета. После чего, каждый раз, когда счетчик достигает значения нуль, текущее значение устанавливается в значение, записанное в TimerXBGLoad.

Значение, считанное из TimerXLoad, - это значение, которое будет прописано в счетчике, как только он станет равным нулю в периодическом режиме.

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.

Регистр управления. TimerXControl

Формат регистра управления приведен в таблице ниже.

Таблица 4.131 - 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.

Регистр снятия прерываний. TimerXIntClr

Любая запись в данный регистр снимает прерывание с выхода счетчика.

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 4.132 - Формат регистра TimerXRIS

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

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 4.133 - 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

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».

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 4.134 - 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 4.35 - 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.

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 4.36 – Формат регистра TimerPCellID0-3

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 4.135 - 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.

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 4.136 - 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

4.12.3 Прерывания

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

4.13 Универсальный блок интерфейсов общего назначения GPIO/EXTINT

Универсальный блок интерфейсов общего назначения имеет следующие характеристики:

- 8 универсальных входа/выхода общего назначения, разделенных на четыре группы по два вывода;
- каждая пара выводов может использоваться как два вывода GPIO, или как пара внешних входов прерывания EXTINIT, или как интерфейс «запрос-подтверждение»;
- подключение к системной шине СБИС через AMBA APB интерфейс, используются 8 младших бит шины данных.

4.13.1 Устройство универсального блока интерфейсов общего назначения

На рисунке 4.37 приведена внутренняя структура универсального блока интерфейсов общего назначения с подключенной к нему шиной AMBA APB. В состав модуля входят функционально завершенные блоки интерфейсов GPIO, EXIRC и Request-acknowledge. Блоки коммутации управляются регистрами конфигурации REGIME_CONTR и RA_CONTR. В основе ячейки ввода/вывода лежит буферная ячейка **bufif1**.

Рисунок 4.37 – Структурная схема универсального блока интерфейсов общего назначения

На рисунке 4.38 представлена схема коммутации, управляемая сигналами регистра задания режима, на примере пары выводов. Начиная с нулевого вывода, все универсальные выводы условно разбиты на пары (2*N, 2*N+1), где N = 0,1,2,3. Это обусловлено тем, что

интерфейсу RA, поддерживающемуся выводами блока, необходимо два вывода. В случае же других интерфейсов (GPIO, EXIRC), пара реализует два вывода заданного интерфейса.

Сигналы PO,PC,PI – относятся к интерфейсу GPIO, EXIRC – к одноименному интерфейсу. Подробное описание назначения сигналов приведены в руководствах пользователя соответствующих блоков.

Рисунок 4.38 – Коммутация выводов интерфейсов, входящих в универсальный блок интерфейсов общего назначения

4.13.2 Программно доступные регистры универсального блока интерфейсов общего назначения

Программно доступные регистры универсального блока интерфейсов общего назначения расположены в области памяти периферийных устройств кластера REG_CPU и имеют базовый адрес GERA Base = 0x000C_C000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже.

Таблица 4.137 - Спецификация регистров блока универсального блока интерфейсов общего назначения

	Адрес (байтовый доступ)	Имя регистра	Описание
GPIO	Базовый адрес	PDR0	Регистр входных данных 0
	Базовый адрес + 04 hex	PDR1	Регистр входных данных 1
	Базовый адрес + 08 hex	PDR2	Регистр входных данных 2
	Базовый адрес + 0C hex	PDR3	Регистр входных данных 3
	Базовый адрес + 10 hex	DDR0	Регистр направления 0
	Базовый адрес + 14 hex	DDR1	Регистр направления 1
	Базовый адрес + 18 hex	DDR2	Регистр направления 2
	Базовый адрес + 1C hex	DDR3	Регистр направления 3
EXIRC	Базовый адрес + 20h	EIENB	Регистр разрешения внешних прерываний
	Базовый адрес + 24h	EIREQ	Регистр запроса внешних прерываний
	Базовый адрес + 28h	EILVL	Регистр уровня внешних прерываний
	Базовый адрес + 2Ch	-	**зарезервировано**
RA	Базовый адрес + 30h	RA_CONTR	Регистр назначения пары выводов GEA: 2*N+1, 2*N в режиме request-acknowledge (RA).
	Базовый адрес + 34h	RA_VAL	Регистр задания/снятия запроса в интерфейсе request-acknowledge (RA).
	Базовый адрес + 38h	REGIME	Регистр назначения пары 2*N+1, 2*N универсальных выводов.
	Базовый адрес + 1Ch	-	**зарезервировано**

4.13.3 Конфигурация регистров универсального блока интерфейсов общего назначения

В таблице 4.138 представлены конфигурации регистров, входящих в состав универсального блока интерфейсов общего назначения.

Таблица 4.138 - Конфигурации регистров блока универсального блока интерфейсов общего назначения

	Имя регистра	Разрядность	Доступ	Начальное значение	Описание
GPIO	PDRx	[31:8]	-	-	Зарезервировано
		[7:0]	ЧТ/ЗП*	-	PDR регистры являются регистрами данных для выводов GPIO Входные и выходные данные передаются через этот регистр на соответствующие выводы GPIO. Направления выводов определяются соответствующими битами регистра DDR
	DDRx	[31:8]	-	-	Зарезервировано
		[7:0]	ЧТ/ЗП	DDR_INIT	Регистры DDR являются регистрами управления направлением выводов GPIO DDR = 0 – вывод является входом. DDR = 1 – вывод является выходом
EXIRC	ENBx	[31:8]	-	0x000000h	Зарезервировано
		[7:0]	ЧТ/ЗП	0x0000h	Если бит ENBx ноль, то по соответствующему входу внешний запрос на прерывание любого типа игнорируется. 0 – внешнее прерывание заблокировано (по умолчанию) 1 – внешнее прерывание разрешено
	REQx	[31: 8]	-	0x000000h	Зарезервировано
		[7:0]	ЧТ/ЗП	0x0000h	Если есть запрос на прерывание, бит REQx устанавливается в 1. 0 – Нет запроса на прерывание (по умолчанию) 1 – Есть запрос на прерывание
	LVLx	[31: 16]	-	0x0000h	Зарезервировано
		[15:0]	ЧТ/ЗП	0x5555h	Значение этого регистра определяет, при каком типе внешнего прерывания будет генерироваться внутреннее прерывание. Приведены возможные значения бит LVLx[1:0] для каждого из каналов: 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

Продолжение таблицы 4.138

	Имя регистра	Разрядность	Доступ	Начальное значение	Описание
RA	RA_CONTR	[7:0]	ЧТ/ЗП	RA_INIT	Значение этого регистра определяет конфигурацию выводов в режиме RA. Для GERA[2*N+1, 2*N]: RA_CONTR[2*N+1, 2*N]=00 – SA,SR** RA_CONTR[2*N+1, 2*N]=01 – SR,SA RA_CONTR[2*N+1, 2*N]=10 – DA,DR** RA_CONTR[2*N+1, 2*N]=11 – DR,DA
RA	RA_VAL***	[31:0]	ЧТ/ЗП	-	Значения регистра участвует в пересылке запрос-ответ интерфейса Request-Acknowledge. Старшая половина регистра RA_CONTR[31:16] формирует запрос на определённом выводе посредством записи единицы в соответствующий выводу бит. Младшая половина RA_CONTR[15:0] – формирует ответ через запись единицы в соответствующий выводу бит. Незадействованные биты не влияют на работу интерфейса.
	REGIME	[7:0]	ЧТ/ЗП	REGIME_INIT	Значение регистра определяет назначение универсальных выводов. Для GERA[2*N+1, 2*N]: RA_CONTR[2*N+1, 2*N]=00 – RA,RA RA_CONTR[2*N+1,2*N]=01–**зарезервировано** RA_CONTR[2*N+1,2*N]=10 – EXIRC,EXIRC RA_CONTR[2*N+1, 2*N]=11 – GPIO,GPIO

Примечания: * ЧТ – чтение, ЗП – запись.

** Интерфейс RA поддерживает два режима функционирования: посылка запроса - ожидания ответа или приём запроса - формирование ответа. Эти режимы соответственно обозначены буквами D(режим формирование запроса) и S(режим формирование ответа). Ввиду преследования универсальности, программирование регистра RA_CONTR позволяет менять местами назначения выводов в любом режиме функционирования RA.

*** Регистр RA_VAL на самом деле является псевдорегистром. Поскольку запись по адресу этого регистра представляет собой возведение триггеров отвечающих либо за формирование «запроса», либо «ответа». А чтение по этому адресу – чтение значений сигналов, являющихся сверткой комбинаторной логики сигналов «запроса» и «ответа». Однако для пользователя такая абстракция (как регистра) исключает обособление описания работы этого узла схемы интерфейса RA. Более подробное описание назначения и работы данного псевдорегистра приведено в разделе «Описание работы устройства».

Каждый интерфейс универсальных выводов имеет жёсткую привязку с битами внутренних регистров. Поэтому в зависимости от того, на каких парах реализовывается определённый интерфейс, задействуются привязанные к этим парам внутренние регистры. Для примера, если нулевая пара (2 вывода GEA) используются как EXIRC, далее 1 пара в режиме GPIO (при этом GPIO должен задан как 8-разрядный с любым делением, например, 1 регистр по 2 бита) и оставшиеся 2 пары – два интерфейса RA, где одна ведомая, а последняя - ведущая пара, тогда:

REGIME[7:0] = 8’b0000_1110;

ENB[1:0], REQ[1:0], LVL[1:0] – настраивают нулевую пару EXIRC. При этом остальные выводы прерывания также не перестают функционировать, но они выступают в роли дополнительных;

PDR0[3:2] и DDR0[3:2] – отвечают за задание данных и направления интерфейса GPIO.

RA_CONTR[7:4] – настраивают режим и конфигурацию интерфейсов RA.

RA_VAL[19,18] и RA_VAL[6,5] – участвуют в формировании сигналов запрос и ответ соответственно.

4.13.4 Описание работы блока

4.13.4.1 Настройка выводов

Назначение универсальных выводов после сигнала сброса задано сигналом REGIME_INIT, сигнал RA_INIT задаёт режим и назначение выводов по умолчанию для интерфейса Request-Acknowledge, а вход DDR_INIT - начальное направление универсальных выводов в режиме GPIO.

Изменения назначения выводов производится программно через запись в регистр REGIME. После чего программно настраивается работа выводов уже в определённом режиме. Внутренняя настройка выводов перед их установкой в определённый режим **не желательна**, поскольку следствием такой последовательности действий является появление неизвестных состояний до первого изменения значения на выводах интерфейса. Это не влияет на функционирование, однако данную особенность следует иметь в виду при использовании и тестировании блока.

В режиме GPIO и RA, интерфейс EXIRC не перестаёт функционировать, что позволяет его использовать как следящее за входными сигналами устройство. В случае если это не нужно, значение регистра разрешения прерываний EIENB должно быть равно нулю. Опять же, как говорилось выше, сначала нужно задать REGIME, а уже после настраивать интерфейс EXIRC.

4.13.4.2 Функционирование

Передача данных в режиме GPIO

Регистр PDR(in) обновляется постоянно при изменении PI. Выход PO определяется значением регистра PDR(out).

Когда вывод GPIO используется в качестве входа (DDR=0), чтение возвращает

значение регистра PDR(in), а запись производится в регистр PDR(out).

Когда вывод GPIO используется как выход (DDR=1), запись устанавливает значение регистра PDR(out), а при чтении возвращается записанное значение.

Прерывания EXIRC

Процедура инициализации блока для отслеживания внешнего прерывания следующая:

1. сбросить соответствующий внешнему прерыванию бит регистра EIENB
2. установить соответствующий внешнему прерыванию бит регистра EILVL
3. сбросить соответствующий внешнему прерыванию бит регистра EIREQ
4. 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 U_EXIRC.

При использовании внешнего прерывания для вывода СнК 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.

Интерфейс Request-Acknowledge (RA)

В режиме RA активный уровень универсальных выводов – низкий. Краткий алгоритм работы интерфейса (временные диаграммы представлены на рисунке 4.39 - две верхние диаграммы):

1) ведущее устройство (D - drive) посылает запрос. Соответствует моменту времени **t1**. Формируется программно, на рисунке – жирная вертикальная стрелка.

2) приняв запрос, ведомое устройство (S - slave) формирует ответ (момент времени **t2**), также формируется программно - жирная вертикальная стрелка.

3) получив ответ, ведущее устройство самостоятельно снимает активный уровень запроса (аппаратно) – диагональная стрелка. Снятие – момент времени **t3**.

4)снятие запроса ведущего устройство порождает (диагональная стрелка) аппаратное снятие ответа ведомого устройства – момент времени **t4**.

Начиная с момента времени **t4** интерфейс отработал цикл запрос-ответ и готов к формированию новой пересылки.

Рисунок 4.39 – Временные диаграммы интерфейса Request-Acknowledge

Пара универсальных выводов, определённых как RA интерфейс, может работать как устройство ожидающее запрос и формирующее ответ (S - режим), и как – выдающее запрос и ожидающее ответ (D - режим).

Для идентификации получения запроса/ответа необходимо разрешить прерывание в регистре ENB для бит, соответствующих используемым парам выводов в режиме RA и проследить, чтобы в соответствующих битах регистра LVL стояли значения «00» или «11» (зависит от задачи), являющиеся условиями генерации прерывания по низкому уровню сигнала или по заднему фронту соответственно. Так же есть возможность фиксирования прерывание через чтение регистра RA_VAL.

Регистр RA_VAL как при записи, так и при чтении функционально можно разбить на две части: старшая половина отвечает за режим D, а младшая – S. В случае если выбранная пара RA работает в одном режиме, а запись произведена в биты, отвечающие за другой режим, влияния на работу интерфейса данной пары/пар не произойдет. Эти биты (неправильно записанные) в регистре RA_VAL сбрасываются самостоятельно исходя из логики их формирования.

Рисунок 4.40 – Устройство псевдорегистра RA_VAL

На рисунке 4.40 представлены биты регистра RA_VAL в разных режимах. И в старшей, и в младшей половине один бит отвечает за одну пару выводов GER в режиме интерфейса RA. Т.е. каждой паре в этом регистре отведены два бита. Как видно из Рисунка 4.40, 16-й и 0-й биты принадлежат нулевой паре RA, 17-й и 1-й – первой паре RA и т.д. Биты с 8 по 15 и с 24 по 31 зарезервированы. При этом 0,1,2,…7 отвечают за режим ведомого устройства, а 16,17,18,…23 – ведущего. Таким образом, если пара RA выбрана как ведущее устройство, то для выдачи воздействия производится запись в определённый бит старшей части регистра, а бит младшей части регистра, относящийся к рассмотренной паре RA, не используется, и запись в него, как отмечалось выше, ни на что не влияет.

Чтение из регистра RA_VAL, как это видно из рисунка 4.40 соответствует логическим свёрткам самих внешних сигналов интерфейса. При этом информационную нагрузку несут лишь те биты, которые соответствуют заданным парам в режиме RA. Например, используется седьмая пара GEA в режиме RA – D (ведущая). Тогда чтение из регистра RA_VAL в 23 бите возвратит свертку «по-ИЛИ» сигналов данной пары, а 7-й бит не несёт информационной нагрузки. Если же, эта пара используется в S режиме (ведомое), то тогда в 7-м бите регистра RA_VAL будет находиться значение логического умножения сигналов «запроса» (R- request) и инверсного сигнала «ответа» (A - acknowledge). В последнем случае 23-й бит не представляет интерес при чтении.

4.14 Системный контроллер процессорного кластера CLSC

Системный контроллер процессорного кластера CLSC выполняет следующие функции:

- содержит идентификатор кластера,
- реализует интерфейс программных межпроцессорных прерываний между процессорными системами NM6408, входящими в состав данного кластера.

4.14.1 Программно доступные регистры системного контроллера CLSC

Программно доступные регистры системного контроллера расположены в области памяти периферийных устройств REG_CPU и имеют базовое смещение 000C_B000h.

Таблица 4.139 – Регистры системного контроллера CLSC

Адрес	Название	Доступ	Описание
000C_B000h	CLUSTERID	ЧТ	Идентификатор процессорного кластера
000C_B00Ch	NMPUNMICTRL	ЧТ/ЗП	Регистр немаскируемого прерывания NMPU
000C_B010h	RCVREQ	ЧТ/ЗП	Регистр приёма межпроцессорных прерываний
000C_B014h	DRVREQ	ЧТ/ЗП	Регистр выдачи межпроцессорных прерываний

Идентификатор процессорного кластера (CLUSTERID)

Регистр-идентификатор процессорного кластера доступен только для чтения. Он содержит 32-разрядный идентификатор PC и может принимать одно из значений:

- 0 – процессорный кластер PC0,
- 1 – процессорный кластер PC1,
- 2 – процессорный кластер PC2,
- 3 – процессорный кластер PC3.

Регистр немаскируемого прерывания NMPU (NMPUNMICTRL)

Регистр немаскируемых прерываний NMPU позволяет выдавать запрос на немаскируемое прерывание каждой процессорной системе NMPU кластера. Формат регистра приведён ниже (см. таблицу 4.140).

Таблица 4.140 – Формат регистра немаскируемых прерываний NMPU (NMPUNMICTRL)

Разряды	Название	Доступ	Описание
[31:4]	-	-	Зарезервировано
[3]	NMI3	ЧТ/ЗП	Запрос на немаскируемое прерывание процессорной системе NMPU3
[2]	NMI2	ЧТ/ЗП	Запрос на немаскируемое прерывание процессорной системе NMPU2
[1]	NMI1	ЧТ/ЗП	Запрос на немаскируемое прерывание процессорной системе NMPU1
[0]	NMI0	ЧТ/ЗП	Запрос на немаскируемое прерывание процессорной системе NMPU0

Для того, чтобы вызвать немаскируемое прерывание в процессорной системе NMPU, нужно в соответствующий разряд регистра NMPUNMICTRL записать значение 1, затем записать значение 0. Немаскируемое прерывание происходит в момент изменения соответствующего бита из 0 в 1.

Регистры приёма и выдачи межпроцессорных прерываний (DRVREQ и RCVREQ)

Регистр выдачи межпроцессорных прерываний (см. Таблицу 4.141) используется для выдачи запросов к другим процессорным системам СнК. Регистр приёма межпроцессорных прерываний (см. таблицу 4.142) используется для обработки запросов от других процессорных систем СнК. Разряды регистров DRVREQ и RCVREQ разных процессорных систем соединены между собой (см. Таблицу 4.143).

Таблица 4.141 – Формат регистра выдачи межпроцессорных прерываний (DRVREQ)

Разряды	Название	Доступ	Описание
[31:16]	-	-	Зарезервировано
[15:0]	DRVREQ	ЧТ/ЗП	Один бит поля GENREQ соответствует одному биту поля RCVREQ другого ПУ. Запись значения 1 в бит поля GENREQ вызывает аппаратную установку соответствующего бита RCVREQ другого ПУ. Программный сброс бита RCVREQ другого ПУ системы сбрасывает аппаратно бит поля GENREQ.

Таблица 4.142 – Формат регистра приёма межпроцессорных прерываний (RCVREQ)

Разряды	Название	Доступ	Описание
[31:16]	-	-	Зарезервировано
[15:0]	RCVREQ	ЧТ/ЗП	Один бит поля RCVREQ соответствует одному биту GENREQ другого ПУ. Программная установка соответствующего бита GENREQ в другой процессорной системе устанавливает бит RCVREQ. Запись значения 1 в бит поля RCVREQ сбрасывает этот бит и инициирует аппаратный сброс соответствующего бита GENREQ. Установка бита поля RCVREQ вызывает запрос на прерывание.

Таблица 4.143 – Соответствие разрядов полей регистров GENREQ и RCVREQ и запросов на прерывание

Разряд регистра DRVREQ	Разряд регистра RCVREQ	Запрос на прерывание
DRVREQ[15]	RCVREQ[15]	Низкоприоритетное межпроцессорное прерывание 15 к/от NMPU3
DRVREQ[14]	RCVREQ[14]	Низкоприоритетное межпроцессорное прерывание 14 к/от NMPU2
DRVREQ[13]	RCVREQ[13]	Низкоприоритетное межпроцессорное прерывание 13 к/от NMPU1
DRVREQ[12]	RCVREQ[12]	Низкоприоритетное межпроцессорное прерывание 12 к/от NMPU0
DRVREQ[11]	RCVREQ[11]	Низкоприоритетное межпроцессорное прерывание 11 к/от PCk
DRVREQ[10]	RCVREQ[10]	Низкоприоритетное межпроцессорное прерывание 10 к/от PCj
DRVREQ[9]	RCVREQ[9]	Низкоприоритетное межпроцессорное прерывание 9 к/от PCi
DRVREQ[8]	RCVREQ[8]	Низкоприоритетное межпроцессорное прерывание 8 к/от центральной управляющей процессорной системы (CCPU)
DRVREQ[7]	RCVREQ[7]	Высокоприоритетное межпроцессорное прерывание 7 к/от NMPU3
DRVREQ[6]	RCVREQ[6]	Высокоприоритетное межпроцессорное прерывание 6 к/от NMPU2
DRVREQ[5]	RCVREQ[5]	Высокоприоритетное межпроцессорное прерывание 5 к/от NMPU1
DRVREQ[4]	RCVREQ[4]	Высокоприоритетное межпроцессорное прерывание 4 к/от NMPU0
DRVREQ[3]	RCVREQ[3]	Высокоприоритетное межпроцессорное прерывание 3 к/от PCk
DRVREQ[2]	RCVREQ[2]	Высокоприоритетное межпроцессорное прерывание 2 к/от PCj
DRVREQ[1]	RCVREQ[1]	Высокоприоритетное межпроцессорное прерывание 1 к/от PCi
DRVREQ[0]	RCVREQ[0]	Высокоприоритетное межпроцессорное прерывание 0 к/от центральной управляющей процессорной системы (CCPU)

Примечания: Для PC0: PCi = PC1, PCj = PC2, PCk = PC3;
для PC1: PCi = PC0, PCj = PC3, PCk = PC2;
для PC2: PCi = PC3, PCj = PC0, PCk = PC1;
для PC3: PCi = PC2, PCj = PC1, PCk = PC0.

Запросы на прерывание не могут быть маскированы в системном контроллере. Маскирование может быть произведено только в контроллере прерываний.

4.15 Генератор тактовых сигналов и сигналов сброса CRGH

4.15.1 Особенности CRGH

Данное устройство предназначено для управления со стороны CPU сигналами синхронизации и сброса в для своих NMPU0…NMPU3 в составе кластера PC**.**

Устройство выполняет три основные функции:

- Контроль работы PLL;
- Контроль выходной частоты блока;
- Генерация внутренних сигналов сброса в зависимости от внешних факторов и внутренних управляющих регистров.

CRGH состоит из следующих блоков:

Генератор сигналов сброса со следующими особенностями работы:

- Поддержка нескольких факторов сигнала сброса;
- Внешний сигнал сброса, который инициализирует всю систему;
- Сигнал сброса SRST, который выставляется системой отладки;
- Сигнал сброса от верхнего управляющего блока;
- Программно-управляемый сигнал сброса, доступный через внутренний регистр;
- Генерация внутренних сигналов сброса под воздействием факторов.

Генератор тактовых сигналов со следующими особенностями работы:

- Управление генерацией и остановкой PLL;
- Настройка входных делителей PLL в пределах (1 ÷ 64)*2;
- Настройка умножителей PLL в пределах (1 ÷ 4096)*2;
- Настройка выходных делителей PLL в пределах (1 ÷ 64);
- Выбор тактового сигнала системы между выходом PLL и внешним тактовым сигналом.

Управление тактовыми сигналами со следующими особенностями работы:

- Поддержка трех тактовых доменов;
- Управление коэффициентом деления частоты двух доменов в пределах (1 ÷ 216);
- Управление отключением тактового сигнала двух доменов.

APB-интерфейс для доступа к управляющим регистрам:

- Поддержка протокола AMBA APB.

Рисунок 4.41 - Структурная схема генератора тактовых сигналов и сигналов сброса CRGH

4.15.2 Программно доступные регистры генератора тактовых сигналов и сигналов сброса CRGH

Таблица 4.144 - Регистры CRGH

Адрес (байтовый доступ)	Тип	Начальное значение	Имя	Описание
0x000C_A000 hex	ЧТ/ЗП	CRIPLLCR	PLLCR	Регистр управления PLL
0x000C_A004 hex	ЧТ	0x0	PLLRDYMR	Регистр мониторинга готовности PLL
0x000C_A008 hex	ЧТ/ЗП	0x0	IMACR	Регистр маски прерывания от PLL
0x000C_A00C hex	ЗП	0x0	PICR	Регистр снятия прерывания от PLL
0x000C_A010 hex	ЧТ	CRIDIVMODE	CLKDIV	Регистр управления выходными делителями
0x000C_A014 hex	ЧТ	0x	CRGCR	Регистр управления CRGH
0x000C_AC00 hex	ЧТ/ЗП	0x	PSCR	Регистр управления запуском процессора
0x000C_AF00 hex	ЧТ/ЗП	0x	CRGSR	Регистр статуса CRGH

Регистр управления PLL (PLLCR)

31 26	25 20	19 14	13 1	0
OPDIV1[5:0]	OPDIV0[5:0]	IDIV[6:1]	FBK[13:1]	BYP
R/W	R/W	R/W	R/W	R/W
CRIPLLCR				

Таблица 4.145 – Формат регистра PLLCR

Bit 31:26	**OPDIV1**
	Эти биты используются для задания выходного коэффициента деления PLL OPDIV1<5:0> для тактового сигнала DDRCLK.
Bit 25:20	**OPDIV0**
	Эти биты используются для задания выходного коэффициента деления PLL OPDIV0<5:0> для тактового сигнала CCLK.
Bit 19:14	**IDIV**
	Эти биты используются для задания входного коэффициента деления PLL IDIV<6:1>. IDIV<0> = 1’b1.
Bit 13:1	**FBK**
	Эти биты используются для задания обратного коэффициента деления PLL FBK<13:1>. FBK<16:14> = 3’b000, FBK<0> = 1’b1.
Bit 0	**BYP**
	Этот бит отвечает за управление сигналом Bypass PLL.

1) Данный регистр используется для задания конфигурации PLL. Для уточнения возможных значений рекомендуется воспользоваться документацией на PLL;
2) Параметр TESTCLK_OPDIV блока CRG используется для задания выходного коэффициента деления OPDIV2<5:0> для тактового сигнала TEST_CLOCK;
3) $F_{CLK} = F_{IN} * (FBK+1) / ((IDIV+1) * (OPDIV+1))$. Более подробная информация представлена в документации на PLL;

Регистр мониторинга готовности PLL (PLLRDYMR)

31 … 1	0
Reserved	PLL RDY
R0	R/W
31’d0	1’b0

Таблица 5.146 – Формат регистра PLLRDYMR

Bit 0	**PLLRDY**
	Данный регистр используется для отслеживания сигнала LOCK от PLL.

Регистр маски прерывания от PLL (IMACR)

31 … 1	0
Reserved	RDYINTM
R0	R/W
31’d0	1’b0

Таблица 5.147 – Формат регистра IMACR

Bit 0	**RDYINTM**
	Данный регистр используется для маскирования прерывания, возникающего при установке сигнала LOCK от PLL.

Регистр очистки прерывания от PLL (PICR)

31 … 1	0
Reserved	PLLRDYINT
R0	R/W0
31’d0	1’b0

Таблица 5.148 – Формат регистра PICR

Bit 0	**PLLRDYINT**
	Данный регистр используется для очистки запроса на прерывание. Любая запись в данный регистр интерпретируется как W0.

Регистр управления выходными делителями (CLKDIV)

31 … 8	7 … 4	3 … 0
Reserved	DIVMODE1	DIVMODE0
R0	R/W	R/W
24’d0	CRIDIVMODE	

Таблица 5.149 – Формат регистра CLKDIV[1)]

Bit 7:4	**DIVMODE1[2)]**
	Эти биты используются для задания коэффициента внешнего делителя DIVC1 для тактового сигнала PCLK.
Bit 3:0	**DIVMODE0**
	Эти биты используются для задания коэффициента внешнего делителя DIVC0 для тактового сигнала ACLK.

1) Запись в данный регистр не вызывает отключения PLL и последующего сброса. Она может производиться моментально. После изменения коэффициента деления будет установлен соответствующий сигнал ACLKCHANGED или PCLKCHANGED.
2) Коэффициент деления внешнего делителя может быть представлен в виде выражения $N = 3^m * 2^n$, где m = DIVMODE[3:2], n = DIVMODE[1:0].

Регистр управления CRG (CRGCR)

31 … 3	2	1	0
Reserved	REFLEN	SWRSTREQ	SWLPREQ
R0	R/W	R/W	R/W
29’b0	1’b0	1’b0	1’b0

Таблица 5.150 – Формат регистра CRGCR

Bit 2	**REFLEN**
	Данный регистр используется для разрешения отражений.
Bit 1	**SWRSTREQ**
	Данный регистр используется для выставления запроса на сброс.
Bit 0	**SWLPREQ**
	Данный регистр используется для выставления запроса на отключение тактового сигнала CCLK.

Регистр управления запуском процессора (PSCR)

31 … m	m-1 … 0
Reserved	START[1)]
R0	R/W1
31’d0	1’b0

1)Разрядность сигнала START зависит от внешнего параметра, m = PRSTn_Width

Таблица 5.151 – Формат регистра PSCR

Bit m:0	**START**
	Данный регистр используется для старта процессорного ядра (ядер)

Регистр статуса CRG (CRGSR)

31 … 2	1	0
Reserved	SWRSTACK	SWLPACK
R0	R	R
30’d0	1’b0	1’b0

Таблица 5.152 – Формат регистра CRGSR

Bit 1	**SWRSTACK**
	Данный регистр используется для отслеживания подтверждения запроса на сброс.
Bit 0	**SWLPACK**
	Данный регистр используется для отслеживания подтверждения запроса на отключение тактового сигнала CCLK.

4.15.3 Особенности работы CRGH

4.15.4 APB-интерфейс

Генератор тактовых сигналов и сигналов сброса CRGH поддерживает APB-интерфейс, соответствующий AMBA APB-протоколу.

4.15.4.1 Генерация сигналов сброса

Факторы сброса и реакция факторы сброса

CRGH имеет четыре фактора сигналов сброса: внешний сигнал сброса XRSTn, сигнал сброса от отладчика SRSTinn, сигнал запроса на установление сброса от верхнего управляющего блока UPRST и сигнал сброса, который устанавливается программно SWRST. В таблице 5.83 представлена реакция внутренних сигналов сброса на внешние факторы.

Таблица 5.153 – Факторы сброса и реакция на них

		XRSTn	SRSTinn	UPRST	SWRST	WE_CRPLC
ARSTn		$+_{имп}$	$+_{имп}$	$+_{имп}$	$+_{прог}$	–
PRSTn	START = 0	$+_{уст}$	$+_{уст}$	$+_{уст}$	$+_{уст}$	–
	START = 1	$+_{сбр}$	$+_{сбр}$	$+_{сбр}$	$+_{сбр}$	–
SRSToutn		$+_{имп}$	–	$+_{имп}$	–	–
PLLRST & CRST		$+_{имп}$	$+_{имп}$	$+_{имп}$	–	+
PORST		$+_{вх}$	–	–	–	–
DWNRST		–	–	+	+	–
ISOLREQ		–	–	–	+	–

Внешний сигнал сброса XRSTn

Внешний сигнал сброса происходит путём установки XRSTn в активное положение (активным для XRSTn является низкий уровень). На рисунке 4.42 представлена временная диаграмма входа и выхода из состояния внешнего сброса XRSTn.

Рисунок 4.42 - Временная диаграмма входа и выхода из состояния внешнего сброса XRSTn

1.Установка внешнего сброса XRSTn в активное состояние (активное состояние – низкий уровень).

2. Установка внутренних сигналов сброса CRG (PLLRST, CRST, PORST), сигналов PLL режима by-pass (BYP и PLLRDY), а также сигнала AMBA reset и монитора сброса SRST (ARSTn и SRSToutn) в активное состояние.

3. Переход внешнего сброса XRSTn в неактивное состояние.

4. Переход PORST в неактивное состояние.

5.Переход CRST в неактивное состояние. Стрелкой на диаграмме CRST указано, что переход в неактивное состояние CRST происходит после перехода в неактивное состояние PORST через 16 тактов XCLK.

6. Установка сброса PLL в неактивное состояние. Стрелкой на диаграмме PLLRST указано, что время активного уровня PLLRST должно быть не менее 16 тактов XCLK, но всё время PORST.

7. После снятия сброса PLLRST, PLL выходит из состояния by-pass (BYP устанавливается в неактивное положение) и устанавливается флаг PLLRDY, свидетельствующий о готовности PLL.

8.Установка AMBA reset (ARSTn) в неактивное состояние. Стрелкой на диаграмме ARSTn указано, что переход в неактивное состояние осуществляется после флага готовности PLL (PLLRDY) через 16 тактов XCLK.

9. Установка монитора сброса SRST в неактивное состояние.

Сброс от отладчика SRSTinn

Сброс от отладчика происходит путём установки SRSTinn в активное положение (активным для SRSTinn является низкий уровень). На рисунке 4.43 представлена временная диаграмма входа и выхода из состояния внешнего сброса SRSTinn.

Рисунок 4.43 - Временная диаграмма входа и выхода из сброса от отладчика SRSTinn

1.Установка сброса от отладчика SRSTinn в активное состояние (активное состояние – низкий уровень).

2. Установка внутренних сигналов сброса CRG (PLLRST, CRST), сигналов PLL режима by-pass (BYP и PLLRDY), а также сигнала AMBA reset и монитора сброса SRST (ARSTn и SRSToutn) в активное состояние.

3. Переход сброса от отладчика SRSTinn в неактивное состояние.

4. Установка сброса PLL в неактивное состояние. Стрелкой на диаграмме PLLRST указано, что время активного уровня PLLRST должно быть не менее 16 тактов XCLK, но всё время SRSTinn.

5.Переход CRST в неактивное состояние. Стрелкой на диаграмме CRST указано, что переход в неактивное состояние CRST происходит после перехода в неактивное состояние SRSTinn через 16 тактов XCLK.

6. После снятия сброса PLLRST, PLL выходит из состояния by-pass (BYP устанавливается в неактивное положение) и устанавливается флаг PLLRDY, свидетельствующий о готовности PLL.

7. Установка AMBA reset (ARSTn) в неактивное состояние. Стрелкой на диаграмме ARSTn указано, что переход в неактивное состояние осуществляется после флага готовности PLL (PLLRDY) через 16 тактов XCLK.

Запрос на установление сброса от верхнего управляющего блока UPRST

На рисунке 4.44 представлена временная диаграмма входа и выхода из состояния сброса от верхнего управляющего блока (UPRST).

Рисунок 4.44 - Временная диаграмма входа и выхода из сброса от верхнего управляющего блока UPRST

1. Запрос на сброс поступает на вход верхнего управляющего блока.

2. Запрос на выполнение сброса от верхнего управляющего блока CRG к нижестоящим по иерархии блокам CRG, монитор сброса SRST (SRSToutn) переходит в активное состояние.

3. После запроса на сброс, верхний управляющий блок CRG ожидает подтверждение от ниже стоящих по иерархии CRG(DWNRSTACK).

4. Верхний управляющий блок выдает подтверждение о готовности выполнения сброса.

5. Установка внутренних сигналов сброса CRG (PLLRST, CRST), сигналов PLL режима by-pass (BYP), а также сигнала AMBA reset (ARSTn) в активное состояние.

6. Запрос на сброс на входе верхнего управляющего в неактивном положении.

7.Запрос на выполнение сброса от верхнего управляющего блока CRG к нижестоящим блокам CRG переходит в неактивное положение.

8. После снятия запроса на сброс, верхний управляющий блок CRG ожидает сигнала подтверждения от нижестоящих по иерархии CRG (DWNRSTACK).

9. Установка сброса PLL в неактивное состояние. Стрелкой на диаграмме PLLRST указано, что время активного уровня PLLRST должно быть не менее 16 тактов XCLK, но всё время UPRST.

10. После снятия сброса PLLRST, PLL выходит из состояния by-pass (BYP устанавливается в неактивное положение) и устанавливается флаг PLLRDY, свидетельствующий о готовности PLL.

11. Переход CRST в неактивное состояние. Стрелкой на диаграмме CRST указано, что переход в неактивное состояние CRST происходит после перехода в неактивное состояние DWNRSTREQ через 16 тактов XCLK.

12. Установка AMBA reset (ARSTn) в неактивное состояние. Стрелкой на диаграмме ARSTn указано, что переход в неактивное состояние осуществляется после флага готовности PLL (PLLRDY) через 16 тактов XCLK.

13. Управляющий блок переводит подтверждение о выполнении сброса в неактивное положение.

14. Установка монитора сброса SRST в неактивное состояние.

Программный сброс SWRST

На рисунке 4.45 представлена временная диаграмма входа и выхода из состояния программного сброса (SWRST).

Рисунок 4.45 - Временная диаграмма входа и выхода из программного сброса SWRST

1.Запрос на программный сброс, поступивший с APB-интерфейса.

2. Запрос на отключение шины AXI.

3. Потверждение отключения шины AXI.

4. Запрос на выполнение сброса от верхнего управляющего блока CRG к нижестоящим по иерархии блокам CRG.

5. После запроса на сброс, верхний управляющий блок CRG ожидает подтверждение от ниже стоящих по иерархии CRG(DWNRSTACK).

6. Верхний управляющий блок выдает подтверждение о готовности выполнения сброса.

7. Установка AMBA сброса (ARSTn) в активное состояние.

8. Запрос на программный сброс в неактивном положении.

9. Запрос на выполнение сброса от верхнего управляющего блока CRG к нижестоящим блокам CRG переходит в неактивное положение.

10. После снятия запроса на сброс, верхний управляющий блок CRG ожидает сигнала подтверждения от нижестоящих по иерархии CRG (DWNRSTACK).

11. Установка AMBA reset (ARSTn) в неактивное состояние. Стрелкой на диаграмме ARSTn указано, что время активного уровня ARSTn должно быть не менее 16 тактов XCLK, но всё время DWNRST.

12. Управляющий блок переводит подтверждение о выполнении программного сброса в неактивное положение.

13. Отключение запроса на отключение шины AXI.

14. Включение шины AXI.

4.15.4.2 Генерация тактовых сигналов

Структурная схема генератора тактовых сигналов

Структурная схема делителей тактовых сигналов представлена на рисунке 4.46.

Рисунок 4.46 - Структурная схема делителей тактовых сигналов

В состав делителей тактовых сигналов входят следующие блоки:

- IDIV – входной делитель частоты, настройка делителя может производиться в пределах (1÷64);
- FBK – умножитель частоты в обратной связи, настройка делителя может производиться в пределах (1÷4096);
- ODIV0 – выходной делитель частоты, с которого формируется опорный тактовый сигнал CCLK, настройка делителя может производиться в пределах (1÷64);
- ODIV1 – выходной делитель частоты, с которого формируется тактовый сигнал одного из тактовых доменов CRG, настройка делителя может производиться в пределах (1÷64);

Ограничения для делителей PLL представлены на рисунке 4.47.

Legend:

ODIV1 = CRIPLLCR<31:26> + 1
ODIV0 = CRIPLLCR<25:20> + 1
IDIV = 2*CRIPLLCR<19:14> + 2
FBK = 2*CRIPLLCR<13:1> + 2
DIVMODE1 = 3^CRIDIVMODE<7:6> + 2^CRIDIVMODE<5:4>
DIVMODE0 = 3^CRIDIVMODE<3:2> + 2^CRIDIVMODE<1:0>

PLLCR instead of CRIPLLCR after reset (X or S)
CLKDIV instead of CRIDIVMODE after reset (X or S or U)

DDRCLK = REFCLK*FBK / (IDIV*ODIV1)
PCLK = REFCLK*FBK / (IDIV*ODIV0*DIVMODE1)
ACLK = REFCLK*FBK / (IDIV*ODIV0*DIVMODE0)

Рисунок 4.47 - Ограничения для делителей генератора тактовых сигналов

Управление внешними делителями

Ограничения для внешних делителей представлены в таблице 4.154

Таблица 5.154 – Внешние делители

DIVMODEAPB	DIVMODE	Замечание
8’h0	1/1	
8’h1	1/2	
8’h2	1/4	
8’h3	1/8	
8’h4	1/3	Скважность не равна 50%
8’h5	1/6	
8’h6	1/12	
8’h7	1/24	
8’h8	1/9	Скважность не равна 50%
8’h9	1/18	
8’hA	1/36	
8’hB	1/72	
8’hC	1/27	Скважность не равна 50%
8’hD	1/54	
8’hE	1/108	
8’hF	1/216	

Отключение тактового сигнала запросом от верхнего управляющего блока

Временная диаграмма отключения тактовых сигналов от верхнего управляющего блока представлена на рисунке 4.48.

Рисунок 4.48 - Временная диаграмма отключения тактовых сигналов запросом от верхнего блока

1. Запрос на отключение тактовых сигналов поступает на вход верхнего управляющего блока.
2. Запрос на отключение тактовых сигналов от верхнего управляющего блока CRG к нижестоящим по иерархии блокам CRG.
3. После запроса на отключение тактовых сигналов, верхний управляющий блок CRG ожидает подтверждение от ниже стоящих по иерархии CRG (DWNLPACK).

4. Верхний управляющий блок CRG выдает подтверждение о готовности выполнения отключения тактовых сигналов.

5. Сигнал разрешения подачи тактовых сигналов устанавливается в неактивное положение.

6. Запрос на отключение тактовых сигналов в неактивном положении.

7. Запрос на отключение тактовых сигналов от верхнего управляющего блока CRG к нижестоящим блокам CRG переходит в неактивное положение.

8. После снятия запроса на сброс, верхний управляющий блок CRG ожидает сигнала подтверждения от нижестоящих по иерархии блоков CRG (DWNRSTACK).

9. Сигнал разрешения подачи тактовых сигналов устанавливается в активное положение.

10. Управляющий блок переводит сигнал подтверждения готовности отключения тактовых сигналов в неактивное положение.

Программно-управляемое отключение тактовых сигналов.

Временная диаграмма программного отключения тактовых сигналов представлена на рисунке 4.49.

Рисунок 4.49 - Временная диаграмма программного отключения тактовых сигналов

1.Запрос на программное отключение тактовых сигналов, поступивший с APB-интерфейса.

2. Запрос на отключение шины AXI.

3. Потверждение отключения шины AXI.

4. Запрос на выполнение отключение тактовых сигналов от верхнего управляющего блока CRG к нижестоящим по иерархии блокам CRG.

5. После запроса на отключение тактовых сигналов, верхний управляющий блок CRG ожидает подтверждение от ниже стоящих по иерархии CRG(DWNLPACK).

6. Верхний управляющий блок выдает подтверждение о готовности выполнения отключения тактовых сигналов.

7. Сигнал разрешения подачи тактовых сигналов устанавливается в неактивное положение.

8. Запрос на программное отключение тактовых сигналов в неактивном положении.

9. Запрос на выполнение отключения тактовых сигналов от верхнего управляющего блока CRG к нижестоящим блокам CRG переходит в неактивное положение.

10. После снятия запроса на отключение тактовых сигналов, верхний управляющий блок CRG ожидает сигнала подтверждения от нижестоящих по иерархии CRG (DWNLPACK).

11. Сигнал разрешения подачи тактовых сигналов устанавливается в активное положение.

12. Управляющий блок переводит подтверждение о выполнении программного сброса в неактивное положение, отключение запроса на отключение шины AXI.

14. Включение шины AXI.

5 Структура микросхемы 1879ВМ8Я

5.1 Структурная схема микросхемы 1879ВМ8Я

Микросхема 1879ВМ8Я является высокопроизводительной гетерогенной многопроцессорной системой на кристалле. Структурная схема микросхемы приведена ниже (см. рисунок 5.1).

Рисунок 5.1 - Структурная схема микросхемы 1879ВМ8Я

1879ВМ8Я включает в себя следующие основные блоки:

PC0...PC3 - четыре процессорных кластера, каждый из которых состоит из 4-х вычислительных узлов NeuroMatrix и управляющего ARM процессора (см. гл. 4).

Процессорные кластеры PCx соединены непосредственно «каждый с каждым» тремя 64-разрядными полнодуплексными коммуникационными портами. Обмен данными через эти порты выполняется с использованием механизм прямого доступа в память (ПДП). Дополнительно четыре кластера PCx соединены между собой через 64-разрядный коммутатор стандарта AXI 3.0, обеспечивающий доступ кластерам PCx во внутренние банки друг друга произвольным доступом. Через этот же коммутатор кластера PCx подключаются к центральному управляющему процессорному узлу кластера (CCPU).

CCPU – центральный управляющий процессорный узел кластера, содержащий управляющий скалярный процессор ARM Cortex-A5, внутреннюю память, а также внешние высокоскоростные интерфейсы.

Cortex-A5 – управляющий скалярный процессор ARM Cortex-A5. Он имеет кэш 2-го уровня объёмом 512 Кбайт.

CCMB0…CCMB3 – четыре банка статической памяти объёмом по 16К*64 разряда, причём банк CCMB0 поддерживает эксклюзивный доступ в память.

CP0…CP3 – четыре контроллера коммуникационных портов, соединенных с кластерами PCx через буфера связи LS.

EMI CCPU – контроллер 32-разрядного интерфейса с внешней динамической памятью DDR3-1600 (эффективная частота передачи данных – 1600 МГц). Максимальный объем поддерживаемой памяти – 1Гбайт.

PCIe – высокоскоростной последовательный хост-интерфейс стандарта PCI Express 2.0. Интерфейс состоит из 4 полнодуплексных линий с суммарной пропускной способностью 20GT/c (2ГБайт/с) в каждую сторону (40GT/s или 4ГБайт/с в обе стороны).

MDMAC – высокопроизводительный контроллер ПДП. Основной задачей этого контроллера является подкачка данных из внешней памяти DDR во внутренние банки управляющего узла CCMBx. Контроллер также имеет доступ к внутренним банкам памяти всех кластеров PC.

MPU – блок защиты памяти. Управляющий узел CCPU может запрещать доступ внешних устройств к внутренним банкам CCMBx и внешней памяти DDR. В случае обращения в защищаемую область памяти операция не производится, а выдавшему запрос устройству выдается признак ошибки SLVERR. Этот механизм используется для организации межпроцессорного взаимодействия (семафоры).

ETH –Ethernet MAC контроллер обеспечивает интерфейс между AMBA- AHB шиной и сетью Ethernet. Он поддерживает скорость передачи 10/100 Мбит/с в полудуплексном и дуплексном режимах. Интерфейс Ethernet поддерживает RMII интерфейс, который должен быть соединен с внешним PHY блоком. Блок ETH также аппаратно поддерживает Ethernet Debug Communication Link (EDCL) протокол. Это протокол, базирующийся на UDP/IP протоколе, для удаленной отладки. Блок ETH имеет доступ ко всем областям внутренней памяти.

DIT – два интервальных 32-разрядных таймера.

GIC – контроллер прерываний, который поддерживает до 64-х прерываний с программно настраиваемыми адрес-векторами и приоритетами.

GPIO/EXTINT – универсальный блок интерфейсов общего назначения. Он управляет работой 8-ми внешних выводов, каждый из которых может программно настраиваться на функционирование в качестве выводов общего назначения (GPIO), внешних входов прерывания (EXTINT), а также пара выводов может реализовать интерфейс «запрос-подтверждение».

CSC – системный контроллер, в нем собраны управляющие регистры микросхемы.

SPI - контроллер интерфейса SPI. Он обеспечивает обмен информацией между внутренними банками памяти микросхемы и устройствами, подключенными к внешнему последовательному интерфейсу типа SPI.

WDT – сторожевой таймер.

Temp Sens - контроллер измерения температуры кристалла с температурным датчиком.

Boot ROM – ПЗУ, содержащее начальный загрузчик.

CRGL0…CRGL3 - четыре генератора тактовых сигналов и сигналов сброса для, соответственно, CPU кластера PC0 … CPU кластера PC3.

CCRG - генератор тактовых сигналов и сигналов сброса для CCPU.

DAP – последовательный порт доступа для тестовых и отладочных данных.

5.1.1 Основные потоки данных в микросхеме 1879ВМ8Я

Ниже приведены основные потоки данных в СБИС и способы их обеспечения.

Обмен данными с внешней памятью. PC-кластер осуществляет загрузку/выгрузку данных во внешнюю память через 32-разрядный интерфейс DDR3-1600. Максимальная производительность при обмене с внешней памятью обеспечивается при использовании

механизма ПДП (контроллер MDMAC подкачивает/выгружает данные в/из внутренней памяти CCMBx). Кластеры PCx, в свою очередь, используют коммуникационные порты CP0 (CLL0) для обмена данными между внутренними банками кластера CCMBx и своими внутренними банками. Пиковая производительность интерфейса с внешней памятью – 6.4 ГБайт/с.

Процессорные кластеры PC имеют возможность произвольного доступа во внешнюю память через шинную структуру, но этот способ рассматривается как вспомогательный, т.к. его эффективность гораздо ниже, чем использование каналов ПДП.

Обмен данными между банками внутренней памяти кластеров. Основной способ обмена данными между кластерами PCx – использование коммуникационных портов (CLL0..CLL3). Блоки коммуникационных портов совмещены с внутренними каналами ПДП и предназначены для двунаправленного полнодуплексного безадресного обмена данными между двумя процессорными кластерами PCx. Передающая сторона настраивает коммуникационный порт, задавая адрес локальной памяти, количество передаваемых данных. После этого она ждет готовность приемной стороны. Приемная сторона настраивает коммуникационный порт, задавая адрес своей локальной памяти, то же самое количество передаваемых данных. После этого она ждет готовность передающей стороны. В момент готовности обоих сторон осуществляется передача данных.

Такой способ позволяет быстро передавать данные из внутренней памяти одного кластера во внутреннюю память другого кластера без участия процессорного ядра. Пиковая односторонняя пропускная способность одного порта 6.4Г Байт/с (12.8 ГБайт/с в обе стороны).

Другой способ обмена между внутренними банками памяти кластеров – произвольный доступ в память соседних кластеров через шинную структуру. Этот способ неэффективен по сравнению с использованием коммуникационных портов, и его рекомендуется рассматриваться как вспомогательный.

Настройка и мониторинг системы из четырех PC-кластеров управляющим ARM-процессором. Управляющий процессор Cortex-A5 имеет доступ ко всем внутренним банкам кластеров, банкам памяти CCMBx, а также управляет всеми периферийными устройствами CPU через шину APB.

5.1.2 Шинная структура микросхемы 1879ВМ8Я.

Соединение основных компонентов узла CCPU выполнено по стандарту AMBA AXI 3.0 с использованием шинных коммутаторов (далее упоминается как «шинная структура»).

СБИС 1879ВМ8Я включает 4 основных коммутатора:

- **AXI_PC** (6в6) работает на частоте 800МГц. Обеспечивает произвольный доступ во внутреннюю память кластеров от соседних кластеров, мастер-устройств управляющего узла кластера CCPU (Cortex-A5, MDMAC), а также от внешнего хост-устройства через PCIe интерфейс.

Используя мастер-интерфейс PCtoCCPU, сборка из четырех PC-кластеров получает произвольный доступ в банки памяти центрального управляющего узла CCMBx, к периферийным устройствам кластера на шине APB, а также доступ к внешней памяти микросхемы.

- **AXI_CCMB** (4в4) работает на частоте 800МГц. Обеспечивает доступ во внутренние банки CCMBx со стороны коммуникационных портов CPx.

- **AXI_CCPU** (4в7) работает на частоте 800МГц. Обеспечивает доступ во внутренние банки CCMBx и внешнюю DDR память со стороны мастер-устройств центрального управляющего узла кластера CPU (Cortex A5, MDMAC), а также со стороны хост интерфейса PCIe и отладочного интерфейса DBG .

- **AXI_P32** (1в4) – периферийный 32-разрядный коммутатор, работает на частоте 800МГц. Этот коммутатор содержит мост AXI-APB, к которому подключены периферийные устройства кластера, что позволяет выполнять их программирование. К этому же коммутатору подключены низкоскоростные 32-разрядные устройства микросхемы (BootROM, SPI, ETH), чтобы упростить основной коммутатор AXI_CCPU.

5.1.3 Карта памяти микросхемы 1879ВМ8Я

Карта памяти микросхемы приведена на рисунке 5.2 ниже.

Рисунок 5.2 - Карта памяти микросхемы 1879ВМ8Я

Карты памяти NMPU и CPU были уже описаны, соответственно, в разделах 3.2 и 4.2.

Карта памяти центрального управляющего ПУ (CCPU) состоит из следующих областей памяти:

- CCMB0-CCMB3 – банки внутренний памяти CCPU.
- REG_CCPU - блок собственных периферийных регистров CCPU. При доступе в эту область памяти идёт переадресация на периферийную шину APB.
- EMI_CCPU - банк внешней памяти DDR центрального управляющего ПУ. При этом при использовании DDR-памяти максимального объема (1 ГБайт) со стороны CCPU будут недоступны 1 МБайт младших ячеек, т.к. это адресное пространство используется под банки внутренней памяти кластера и блок периферийных регистров.
- PCIe - области памяти высокоскоростного последовательного хост-интерфейса стандарта PCI Express 2.0.
- CCPU - область памяти центрального процессорного узла.
- PC0 – PC3 - отражение внутреннего адресного пространства соответствующего кластера 0,…3, позволяя обеспечить произвольный доступ к этим банкам со стороны мастер-устройств CCPU и соседних PC-кластеров. Т.к. окно для отражения адресного пространства кластера составляет 512Мб, а максимальное адресное пространство одного PC-кластера – 1Гб, то верхняя половина DDR памяти кластера не доступна мастер-устройствам центрального управляющего узла CCPU.

На рисунке 5.3 изображена карта памяти всех периферийных устройств, относящихся к управляющим ПУ кластеров (CPU) и центральному управляющему ПУ (CCPU).

Рисунок 5.3 - Карта памяти периферийных устройств CPU и CCPU

5.1.4 Система межпроцессорной функциональной синхронизации

Под функциональной синхронизацией понимается синхронизация процессов, запущенных на процессорных устройствах, входящих в состав СнК. Также в функциональную синхронизацию входит синхронизация процессов запущенных на хост-процессоре и процессов СнК. В дальнейшем в данном разделе функциональная синхронизация называется синхронизацией.

Взаимодействие процессов может осуществляться с помощью:

Прерываний.

Ячеек общей памяти (семафоров).

5.1.4.1 Прерывания между вычислительными узлами 1879ВМ8Я

Для синхронизации вычислительных узлов между собой могут быть использованы прерывания. Запрос на прерывание к центральному управляющему процессорному узлу (CCPU) могут формировать (см. рисунок 5.4):

- четыре управляющих процессорных узла кластера (CPU) - по два прерывания от каждого CPU (высокоприоритетное и низкоприоритетное);
- внешние прерывания, поступающие на выводы микросхемы.

В свою очередь центральный управляющий процессор может выдавать запрос на прерывание четырем управляющим процессорным узлам (CPU) - по два прерывания каждому CPU (высокоприоритетное и низкоприоритетное).

Каждый CPU формирует запросы на прерывания каждому NMPU своего кластера: по три прерывания каждому NMPU (высокоприоритетное, низкоприоритетное и немаскируемое прерывание). Каждый узел NMPU может формировать по два прерывания каждому NMPU из своего кластера и управляющему процессорному узлу кластера (CPU).

Рисунок 5.4 - Схема прерываний между вычислительными узлами 1879ВМ8Я

5.1.4.2 Синхронизация на основе ячеек общей памяти (семафоров)

Для реализации семафоров используются блоки защиты памяти (MPU), находящиеся в центральном управляющем узле СБИС - CCPU, а также в управляющих узлах каждого кластера – CPU. Области памяти, защищенные с помощью MPU, позволяют использовать их для хранения семафоров и исключить возможность порчи этих ячеек со стороны других устройств.

Кроме того, банки памяти CMB0 всех кластеров и банк памяти CCMB0 центрального управляющего узла CCPU поддерживают эксклюзивные операции чтения/записи. Механизм эксклюзивного доступа был введён фирмой ARM Ltd вместо локированных операций при организации семафоров в памяти в многопроцессорных и многопоточных системах. С помощью операций эксклюзивного чтения и эксклюзивной записи строятся примитивы синхронизации между выполняющимися потоками команд.

5.2 Управляющий скалярный процессор ARM Cortex-A5

Управляющий скалярный процессор ARM Cortex-A5 центрального управляющего узла полностью идентичен используемому в процессорном кластере (см. раздел 4.3). Единственное его отличие - наличие кэш 2-го уровня объёмом 512 Кбайт.

5.3 Внутренняя память CCPU

Банки CCMB0…CCMB3 внутренней памяти CCPU полностью идентичны банкам CMB0…CMB3 памяти, принадлежащим процессорному кластеру (см. раздел 4.4), за исключением того, что их размер в два раза больше. Причём нулевой банк CCMB0 также поддерживает эксклюзивный доступ в память.

5.4 Контроллер ПДП коммуникационных портов CP

Контроллеры ПДП для коммуникационных портов (CPx) центрального управляющего узла полностью идентичны аналогичным, используемым в процессорном кластере (см. раздел 4.5).

5.5 Контроллер ПДП память-память MDMAC

Контроллер ПДП MDMAC центрального управляющего узла полностью идентичен аналогичному, используемому в процессорном кластере (см. раздел 4.7).

5.6 Контроллер интерфейса с внешней динамической памятью EMI CCPU

Контроллер интерфейса с внешней памятью EMI центрального управляющего узла полностью идентичен аналогичному, используемому в процессорном кластере (см. раздел 4.8).

5.7 Блок защиты памяти MPU

Блок защиты памяти центрального управляющего узла полностью идентичен аналогичному, используемому в процессорном кластере (см. раздел 4.8).

5.8 Системный контроллер прерываний GIC

Системный контроллер прерываний центрального управляющего узла полностью идентичен аналогичному, используемому в процессорном кластере (см. раздел 4.11). Исключением является несколько изменённый список прерываний, приведённый в Таблице 5.1.

Таблица** 5.**1** - **Соответствие прерываний GIC и прерываний о**т периферийных **устройств

Interrupt ID	**Символическое название**	**Описание**	**Приоритет по умолчанию**
0	COMMRX	Канал отладки Rx от ARM	Высший
1	COMMTX	Канал отладки Tx от ARM	
2	CTIRQ	Прерывание от CTI	
3	TS_INT	Прерывание от температурного датчика	
4	TIMINT0	Прерывание таймера 0	
5	TIMINT1	Прерывание таймера 1	
6	WDOGINT	Прерывание сторожевого таймера	
7	-	Зарезервировано	
8	PC0_HP	Высокоприоритетное прерывание от кластера PC0	
9	PC1_HP	Высокоприоритетное прерывание от кластера PC1	
10	PC2_HP	Высокоприоритетное прерывание от кластера PC2	
11	PC3_HP	Высокоприоритетное прерывание от кластера PC3	
12	ELC_INT0	Прерывание от контроллера канала External Link PC0	
13	ELC_INT1	Прерывание от контроллера канала External Link PC1	
14	ELC_INT2	Прерывание от контроллера канала External Link PC2	
15	ELC_INT3	Прерывание от контроллера канала External Link PC3	
16	CP0_SRC_INT	Обобщенное прерывание от передающего канала CP0	
17	CP0_DST_INT	Обобщенное прерывание от принимающего канала CP0	
18	CP0_SRC_INT	Обобщенное прерывание от передающего канала CP1	
19	CP0_DST_INT	Обобщенное прерывание от принимающего канала CP1	
20	CP0_SRC_INT	Обобщенное прерывание от передающего канала CP2	
21	CP0_DST_INT	Обобщенное прерывание от принимающего канала CP2	
22	CP0_SRC_INT	Обобщенное прерывание от передающего канала CP3	
23	CP0_DST_INT	Обобщенное прерывание от принимающего канала CP3	

Продолжение таблицы 5.1

Interrupt ID	Символическое название	Описание	Приоритет по умолчанию
24	MDMAC_INT	Прерывание контроллера ПДП память-память (MDMAC)	
25	MPUW_INT	Прерывание от канала записи MPU	
26	MPUR_INT	Прерывание от канала чтения MPU	
27	SSPDMA_INT	Прерывание контроллера ПДП в блоке SPI	
28	-	Зарезервировано	
29	-	Зарезервировано	
30	GE_INT0	Внешнее прерывание 0	
31	GE_INT1	Внешнее прерывание 1	
32	EMI_INT	Прерывание от контроллера внешней памяти EMI	
33	-	Зарезервировано	
34	SSP_INT	Прерывание от контроллера SPI	
35	ETH_INT	Прерывание от контроллера Ethernet	
36	-	Зарезервировано	
37	-	Зарезервировано	
38	GE_INT2	Внешнее прерывание 2	
39	GE_INT3	Внешнее прерывание 3	
40	PCIE_DMA_INT	Прерывание PCIe от ПДП ('ошибка' либо 'завершение')	
41	PCIE_LCL_INT	Комбинированное прерывание PCIe	
42	PCIE_LEG_INTA	Прерывание PCI legacy INTA	
43	PCIE_LEG_INTB	Прерывание PCI legacy INTB	
44	PCIE_LEG_INTC	Прерывание PCI legacy INTC	
45	PCIE_LEG_INTD	Прерывание PCI legacy INTD	
46	GE_INT4	Внешнее прерывание 4	
47	GE_INT5	Внешнее прерывание 5	
48	PC0_LP	Низкоприоритетное прерывание от кластера PC0	
49	PC1_LP	Низкоприоритетное прерывание от кластера PC1	
50	PC2_LP	Низкоприоритетное прерывание от кластера PC2	
51	PC3_LP	Низкоприоритетное прерывание от кластера PC3	
52	PLLL0_RDYINT	Готовность PLL от CRGL0	
53	PLLL1_RDYINT	Готовность PLL от CRGL1	
54	PLLL2_RDYINT	Готовность PLL от CRGL2	
55	PLLL3_RDYINT	Готовность PLL от CRGL3	

Продолжение таблицы 5.1

Interrupt ID	Символическое название	Описание	Приоритет по умолчанию
56	GE_INT6	Внешнее прерывание 6	
57	GE_INT7	Внешнее прерывание 7	
58	L2C_DECERR	Ошибка декодирования на L2 интерфейсе (AXI DECERR)	
59	L2C_SLVERR	Ошибка слэйва на L2 интерфейсе (AXI SLVERR)	
60	L2C_CINT	Комбинированное прерывание от L2-кэш памяти (включает два выше)	
61	PMUIRQ	Прерывание от ARM Performance Monitor	
62	PLLC_RDYINT	Готовность PLL от CCRG	Низший
63	-	Зарезервировано	

5.9 Блок сдвоенных таймеров DIT

Блок сдвоенных таймеров DIT центрального управляющего узла полностью идентичен аналогичному, используемому в процессорном кластере (см. раздел 4.12).

5.10 Универсальный блок интерфейсов общего назначения GPIO/EXTINT

Универсальный блок интерфейсов общего назначения центрального управляющего узла полностью идентичен аналогичному, используемому в процессорном кластере (см. раздел 4.13).

5.11 Контроллер интерфейса ETHERNET

Микросхема 1879ВМ8Я содержит MAC контроллер (GRETH), который поддерживает сетевой протокол Ethernet, а также поддерживает стандарт шины AMBA AHB. Скорость передачи контроллера составляет 10/100 Мбит/с 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 (MII) и сокращенный формат интерфейса MII (RMII). Данные интерфейсы должны быть соединены с устройством физического уровня (PHY). GRETH также обеспечивает доступ к интерфейсу MII Management, который используется для настройки блока PHY.

Аппаратно в контроллере поддерживается интерфейс Ethernet Debug Communication Link (EDCL). EDCL основан на протоколе UDP/IP, применяемый для удаленной отладки.

Блок-схема внутренней структуры GRETH приведена ниже (см. рисунок 5.5).

Рисунок 5.5 - Блок-схема внутренней структуры GRETH

5.11.1 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 одной или более PHY-микросхем, подключенных к MAC. Управление работой данного интерфейса осуществляется посредством шины APB.

Блок EDCL обеспечивает доступ к внутренней памяти СБИС 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.

Микросхема 1879ВМ8Я 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 и передатчика Ethernet.

5.11.2 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.

5.11.3 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. Минимальная тактовая частота шины AHB для скорости передачи 10 Мбит – 2,5 МГц, для 100 Мбит – не ниже 18 МГц. Использование более низкой тактовой частоты шины AHB приведет к потере пакетов.

5.11.4 Интерфейс передатчика TxDMA

Интерфейс DMA-передатчика используется для передачи данных по сети Ethernet. При передаче данных используются дескрипторы, определенные в памяти.

Установка дескриптора

Дескриптор состоит из двух 32-х разрядных слов и показан на рисунке ниже (см. Рисунок 5.6 и Рисунок 5.7). Количество байт, которые должны быть переданы, устанавливаются в поле длины (LENGHT), поле адреса (ADDRESS) адресует данные. Адрес должен быть выровнен с точностью до слова. 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 5.6 - Дескриптор передатчика GRETH слова 0 (смещение адреса 0x0)

Таблица 5.2 – Описание полей слова 0 дескриптора передатчика GRETH

Биты	Название	Функция
[31:16]	-	Не используется.
[15]	Attempt Limit Error (AL)	Пакет не был передан, потому было достигнуто максимальное количество попыток.
[14]	Under run Error (UE)	Пакет был некорректно передан из-за ошибки переполнения FIFO.
[13]	Interrupt enable (IE)	Разрешение прерываний. Прерывание было сгенерировано, когда пакет от этого дескриптора был послан, 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 байтов, которое должно быть передано.

Рисунок 5.7 - Дескриптор передатчика GRETH слова 1 (смещение адреса 0x4)

Таблица 5.3 - Описание полей слова 1 дескриптора передатчика GRETH

Биты	Название	Функция
[31:2]	Address (ADDRESS)	Указатель на область памяти, откуда данные будут загружены.
[1:0]	-	Не используется

Для активации дескриптора необходимо установить бит EN, после этого бит не должен перезаписываться до момента, пока GRETH сам не сбросит этот бит.

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. 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 от границы 1Кб в начальное состояние (0), в дескрипторах должен быть установлен бит WR.

Поле указателя программно доступно для записи, но необходимо делать это с осторожностью. Запрещено записывать поле указателя при активной передаче данных.

Последний шаг для активизации передачи - это установка бита разрешения передачи (transmit enable) в управляющем регистре. 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) будет установлен.

Обработка дескриптора после передачи

Когда передача пакета закончена, результат записывается в первое слово соответствующего дескриптора. Признак ошибки отсутствия данных (Underrun Error) устанавливается, если передающее FIFO стало пустым до того, как пакет был полностью передан. 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 первого слова дескриптора устанавливаются в 0 после передачи, второе слово остается нетронутым.

Разряд разрешения (enable bit) является признаком возможности повторного использования дескриптора. Дескриптор можно использовать повторно после того, как этот бит был сброшен GRETH-контроллером. В регистре статуса контроллера GRETH 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 регистр разрешения передачи (transmit enable register).

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 максимальное значение, задаваемое при настройке, плюс размер заголовка, то пакет не будет передан.

5.11.5 Интерфейс приемника RxDMA

Интерфейс DMA-приемника используется для приема данных по сетевому протоколу Ethernet. При приеме данных используются дескрипторы, определенные в памяти.

Установка дескрипторов

Дескриптор состоит из двух 32-х разрядных слов и показан на рисунке ниже (см. рисунок 5.8 и рисунок 5.9). 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. Разряд возврата (WR) должен быть установлен прежде, чем будет активирован дескриптор.

31 … 27	26	25 … 19	18	17	16	15	14	13	12	11	10 … 0
RESERVED	MC	RESERVED	LE	OE	CE	FT	AE	IE	WR	EN	LENGTH

Рисунок 5.8 - Дескриптор приемника GRETH слова 0 (смещение адреса 0x0)

Таблица 5.4 - 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)	Установка 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 5.9 - Дескриптор приемника GRETH слова 1 (смещение адреса 0x4)

Таблица 5.5 - Описание полей слова 1 дескриптора приемника GRETH

Биты	Название	Функция
[31:2]	Address (ADDRESS)	Указатель на область памяти, куда данные будут загружены.
[1:0]	-	Не используется.

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. 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 1КБайт (использован дескриптор со смещением адреса 0x3F8). Для того чтобы указатель перешел от границы 1Кб в начальное состояние (0), в дескрипторах должен быть установлен бит WR.

Поле указателя программно доступно для записи, но необходимо делать это с осторожностью. Запрещено записывать поле указателя при активном приеме данных.

Последний шаг для активизации приема - это установка разряда разрешение приема (receive enable) в управляющем регистре. Таким образом, GRETH- контроллеру сообщается о возможности считать первый дескриптор и ожидать входной пакет.

Обработка дескриптора после приема

Сброс разряда активации дескриптора (descriptor enable bit) является признаком окончания приема для GRETH-контроллера, так же сбрасываются другие управляющие разряды (WR, IE). Поле длины (LENGTH) определяет количество принятых байтов. Сохраняются также части формата кадра Ethernet: адрес приемника, адрес источника, тип и поля данных. Разряды 18-14 в слове 0 дескриптора являются статусными разрядами, определяющими различные ошибки при приеме данных (см. Таблицу 5.4). Все пять разрядов сбрасываются после приема данных, если нет ошибок.

Если адрес принятого пакета является недопустимым для MAC, то бит IA регистра статуса будет установлен.

Пакеты размера больше максимального размера вызывают установку признака FT, при этом не гарантируется, что в поле длины (LENGTH) содержится правильное значение количества принятых байтов. Счетчик останавливается после записи последнего байта блока максимального размера.

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 (Receiver AHB Error) и приемник останавливается. Приемник может быть активирован снова, если будет установлен бит разрешения передачи (Receive Enable) в управляющем регистре.

5.11.6 MDIO интерфейс

Интерфейс MDIO обеспечивает доступ к конфигурационным и статусным регистрам PHY-микросхемы, используя двухпроводной интерфейс, являющийся частью интерфейса MII . 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 прописаны: PHY, регистр адресов блока MDIO, также должен быть установлен разряд чтения. После этого устанавливается разряд занятости (Busy bit). Операция чтения заканчивается, когда сбрасывается разряд занятости (Busy bit). Если операция была успешной, разряд Linkfail сбрасывается, в поле данных находятся считанные данные. При ошибочном завершении операции устанавливается бит Linkfail. В этом случае поле данных не определено.

Операция записи начинается записью 16-разрядных данных, адреса PHY и адреса регистра в регистр управления MDIO, а также с установкой бита записи. Операция заканчивается, когда сбрасывается разряд занятости (busy bit), если операция была успешной, бит Linkfail равен 0.

5.11.7 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

Функционирование EDCL

EDCL получает пакеты параллельно с приемным каналом DMA MAC-контроллера. Они используют отдельный MAC адрес, который позволяет различать пакеты для EDCL от пакетов, предназначенных DMA-каналу MAC-контроллера. EDCL также имеет фиксированный IP адрес, который устанавливается в общих настройках. 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 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.

EDCL протоколы

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 5.10) показан IP пакет, требуемый для EDCL интерфейса.

Ethernet Header	IP Header	UDP Header	2 B Offset	4 B Control word	4 B Address	Data 0 - 242 4B Words	Ethernet CRC

Рисунок 5.10 - IP пакет, требуемый для EDCL интерфейса

Для успешной связи по EDCL интерфейсу требуется следующее: корректный MAC адрес приемника, как указано в основных настройках, поле типа протокола Ethernet, содержащее 0x806 (ARP) или 0x800 (IP). После чего IP–адрес сравнивается со значением, определенном в общих настройках, на совпадение. 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 обеспечивает только один доступ в каждых момент времени - 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 5.11) показаны поля протокола прикладного уровня (поля данных исключены) в пакетах, принятых посредством EDCL интерфейса.

16-bit Offset	14-bit Sequence number	1-bit R/W	10-bit Length	7-bit Unused

Рисунок 5.11 - 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 выше (см. Рисунок 5.), содержит данные, которые должны быть записаны. Если R/W равен 0, поле данных пусто в принимаемых пакетах.

На ринке ниже (см. Рисунок 5.12) показаны поля прикладного уровня ответов от EDCL.

16-bit Offset	14-bit sequence number	1-bit ACK/NAK	10-bit Length	7-bit Unused

Рисунок 5.12 - 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.

Установка IP для EDCL и адреса для Ethernet

Значение по умолчанию IP для EDCL и MAC-адресов устанавливается в общих настройках: ipaddrh, ipaddrl, macaddrh и macaddrl. IP адрес может быть позже изменен программно, но MAC адрес зафиксирован. Чтобы активировать EDCL нескольких GRETH-котроллеров в одной и той же подсети, 4 LSB разряда IP и MAC адреса должны быть выборочно установлены входным сигналом. Достигается это установкой в общих настройках edcl =2 и выдачей 4-х разрядного значения LSB на ethi.edcladr.

Размер буфера EDCL

Блок EDCL содержит внутренние блоки памяти, предназначенные для хранения полученных пакетов в процессе передачи данных. Для каждого размера в таблице приведено максимальное количество пакетов, которое можно обрабатывать. Максимальный размер каждого пакета получается сложением размера заголовка и максимального размера данных в пакете. Если отправляется большее количество пакетов данных, чем определено в таблице, до прихода подтверждения, то это приведет к потере пакетов данных.

Таблица 5.6 – Размер буфера EDLC

Total buffer size (kB)	Number of packet buffers	Packet buffer size (B)	Maximum data payload (B)
1	4	256	200
2	4	512	456
4	8	512	456
8	8	1024	968
16	16	1024	968
32	32	1024	968
64	64	1024	968

5.11.8 RMII интерфейс

Интерфейс RMII был разработан в связи с необходимостью интерфейса для Ethernet – контроллеров с меньшим количеством выводов (по сравнению с MII интерфейсом). Он использует 6 (7) сигналов, которые являются подмножеством MII сигналов. Таблица 5.7 показывает соответствие между RMII сигналами и MII интерфейсом.

Таблица 5.7 - Соответствие сигналов между RMII и GRLIB MII интерфейсом

RMII	MII
txd[1:0]	txd[1:0]
tx_en	tx_en
crs_dv	rx_crs
rxd[1:0]	rxd[1:0]
ref_clk	rmii_clk
rx_er	not used

5.11.9 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 Researchs web site (http://gaisler.com).

5.11.10 Регистровая модель контроллера GRETH

Программно доступные регистры контроллера GRETH расположены в области памяти периферийных устройств REG_CCPU, имеют базовое смещение ETH Base = 0x000D_0000 и общий размер 4 КБайт. Регистры имеют разрядность 32 бит. Спецификация регистров представлена в таблице ниже (Таблица 5.8).

Таблица 5.8 – регистры контроллера GRETH

APB address offset	Register
0x0	Control register
0x4	Status/Interrupt-source register
0x8	MAC Address MSB
0xC	MAC Address LSB
0x10	MDIO Control/Status
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

Регистр управления контроллера GRETH

31	30 28	27	26	25	24 15	14	13	12	11	10	9 8	7	6	5	4	3	2	1	0
ED	BS		MA	MC	RESERVED	ED	RD	DD	ME	PI	RES	SP	RS	PM	FD	RI	TI	RE	TE

Рисунок 5.13 – Управляющий регистр контроллера GRETH

Таблица 5.9 - Структура управляющего регистра контроллера GRETH

Биты	Название	Функция
[31]	EDCL available (ED)	Устанавливается, если EDCL режим 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)	Устанавливается в 1, когда ядро контроллера поддерживает прием начального 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 значению сигнала ethi.edcldisable.
[13]	RAM debug enable (RD)	1 – разрешает режим отладки памяти. Значение при сбросе – 0. Бит доступен только если в общих настройках ramdebug не равен нулю.
[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 остановлено, и как только сбросится бит занятости MDIO, интерфейс становится доступным. FSM не может быть заново запущен.
[11]	Multicast enable (ME)	Разрешение приема начального адреса для пакета данных (multicast).
[10]	PHY status change interrupt enable (PI)	Разрешить прерывания для определенных изменений статуса 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. Сам себя сбрасывает.
[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 приемника. Прерывание будет сгенерировано каждый раз, когда передан пакет, бит устанавливается 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 передатчика. Прерывание будет сгенерировано каждый раз, когда принят пакет, бит устанавливается в 1. Прерывание генерируется независимо от того, был ли пакет передан успешно, или завершен с ошибкой Значение при сбросе “0”.

Продолжение таблицы 5.9

Биты	Название	Функция
[1]	Receive enable (RE)	Будет установлен всегда, когда активированы новые дескрипторы. До тех пор, пока этот разряд 1, GRETH-контроллер будет считывать новые дескрипторы и как только он встретит неактивный дескриптор, он остановится до тех пор, пока RE не установится снова. В этот разряд будет записана 1, после того, как новый дескриптор активирован. Значение при сбросе “0”.
[0]	Transmit enable (TE)	Будет установлен всегда, когда активированы новые дескрипторы. До тех пор, пока этот разряд 1, GRETH-контроллер будет считывать новые дескрипторы и как только он встретит неактивный дескриптор, он остановится до тех пор, пока TE не установится снова. В этот разряд будет записана 1, после того, как новый дескриптор разрешен. Значение при сбросе “0”.

Регистр статуса контроллера GRETH

Рисунок 5.14 –регистр статуса контроллера GRETH

Таблица 5.10 - 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 сбросе “0”.
[6]	Too small (TS)	Был принят пакет меньше минимального размера. Сбрасывается, когда записывается 1. Значение при сбросе “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.

Регистр, определяющий старшие биты (MSB) MAC-адреса

Рисунок 5.15 - Регистр, определяющий старшие биты (MSB) MAC-адреса

Таблица 5.11 - Структура регистра, определяющего MSB MAC-адрес

Биты	**Название**	**Функция**
[31:16]	-	Не используется.
[15:0]	Bit 47-32	Два старших значащих байта MAC – адреса. Не обнуляются

Регистр, определяющий младшие биты (LSB) MAC-адреса

31	0
Bit 31 downto 0 of the MAC Address	

Рисунок 5.16 - Регистр, определяющий младшие биты (LSB) MAC-адреса

Таблица 5.12 - Структура регистра, определяющего LSBs MAC-адрес

Биты	Название	Функция
[31:0]	Bit 31-0	Четыре младших значащих байта MAC – адреса. Не обнуляются.

Регистр управления/статуса интерфейса MDIO контроллера GRETH

31 16	15 11	10 6	4	3	2	1	0
DATA	PHY ADDRESS	REGISTER ADDRESS		NV	BU	LF	RD	WR

Рисунок 5.17 - Регистр управления/статуса интерфейса MDIO контроллера GRETH

Таблица 5.13 - Структура регистра управления/статуса интерфейса MDIO контроллера GRETH

Биты	Название	Функция
[31:16]	Data (DATA)	При операции чтения определяет считанные данные. При операции записи определяет данные, которые будут переданы. Значение при сбросе - 0x0000.
[15:11]	PHY address (PHYADDR)	Поле содержит адрес PHY-блока, который будет выбираться во время операции чтения или записи. Значение при сбросе - 0x0000.
[10:6]	Register address (REGADDR)	Это поле содержит адрес регистра, который будет выбираться 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 действительные данные, что означает, что поле данных содержит 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 не обнаружено связь с функциональным управлением. Значение при сбросе “1”.
[1]	Read (RD)	Запущена операция чтения. Данные хранятся в поле данных. Значение при сбросе “0”.
[0]	Write (WR)	Запущена операция записи. Данные берутся из поля данных. Значение при сбросе “0”.

Регистр базового адреса таблицы дескрипторов передатчика *GRETH- контроллера*

Рисунок 5.18 - Регистр базового адреса таблицы дескрипторов передатчика

Таблица 5.14 - 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.
[2:0]	-	Не используется

Регистр базового адреса таблицы дескрипторов приемника GRETH-контроллера

Рисунок 5.19 - Регистр базового адреса таблицы дескрипторов приемника

Таблица 5.15 - 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.
[2:0]	-	Не используется

Регистр IP блока EDCL GRETH-контроллера

Рисунок 5.20 - Регистр IP блока EDCL GRETH- контроллера

Таблица 5.16 - Структура регистра IP блока EDCL

Биты	**Название**	**Функция**
[31:0]	EDCL IP адрес	Значение при обнулении берется из общих настроек: *ipaddressh* и *ipaddressl*.

Регистр старших битов hash таблицы GRETH-контроллера

Рисунок 5.21 - Регистр старших битов hash таблицы GRETH- контроллера

Таблица 5.17 - Структура регистра старших битов hash таблицы

Биты	Название	Функция
[31:0]	Hash table (64:32)	Значение бит с 64 по 32 hash таблицы.

Регистр младших битов hash таблицы GRETH-контроллера

Рисунок 5.22 - Регистр младших битов hash таблицы GRETH- контроллера

Таблица 5.18 - Структура регистра младших битов hash таблицы

Биты	Название	Функция
[31:0]	Hash table (31:0)	Значение бит с 31 по 0 hash таблицы.

Регистр старших бит MAC-адреса для EDCL GRETH-контроллера

Рисунок 5.23 - Регистр старших бит MAC-адреса для EDCL GRETH-контроллера

Таблица 5.19 - Структура регистра старших бит MAC-адреса для EDCL

Биты	Название	Функция
[31:16]	-	Зарезервировано
[15:0]	Bit 47 … 32 of the EDCL MAC Address	Определены два старших байта MAC-адреса EDCL. При аппаратном сбросе принимает значения, заданные в общих настройках VHDL: macaddrh.

Регистр младших бит MAC-адреса для EDCL GRETH-контроллера

Рисунок 5.24 - Регистр младших бит MAC-адреса для EDCL GRETH-контроллера

Таблица 5.20 - Структура регистра младших бит MAC-адреса для EDCL

Биты	Название	Функция
[31:0]	Bit 31 ... 0 of the EDCL MAC Address	Определены четыре младших байта MAC-адреса EDCL. При аппаратном сбросе принимает значения, заданные в общих настройках VHDL: macaddrh и macaddrh1.

5.12 Высокоскоростной последовательный интерфейс PCIe

Высокоскоростной последовательный интерфейс используется для интеграции микросхемы 1879ВМ8Я в вычислительную систему. Контроллер интерфейса PCIe обеспечивает отображение внешних устройств шины PCI Express в карте памяти процессора и наоборот, внутренней адресуемой памяти процессорного чипа в адресном пространстве PCI Express.

Интерфейс соответствует стандарту PCI Express 2.0 и содержит 4 линии. Работа на частоте 5 ГГц позволяет достичь пиковой двусторонней пропускной способности 40 GT/c. Интерфейс поддерживает только архитектуру с общим опорным тактовым сигналом (Common Refclk Architecture). Тактовый сигнал должен иметь частоту 100 МГц и скважность 40-60 %.

5.12.1 Устройство интерфейса PCIe

Основные особенности установленного в микросхеме интерфейса PCIe:

- тип устройства – Dual Mode (может быть как конечным устройством шины PCI Express – End Point, так и хост-контроллером – Root Port);
- количество линий передачи/приема – 4;
- количество виртуальных каналов – 1;
- количество функций PCI – 1.

Интерфейс поддерживает:

- прерывания MSI;
- Legacy прерывания;
- D1 состояние PCI функции;
- генерация и проверка ECRC;
- обращение порядковых номеров линий PCI Express;
- встроенный контроллер прямого доступа в память;
- внутренняя и внешняя трансляция адресов;
- 6 входных регионов в режиме End Point, 2 входных региона в режиме Root Port;
- 32 выходных региона не зависимо от режима;
- гибкая настройка карты памяти (ECN Resizable BAR).

Интерфейс не поддерживает:

- горячее подключение (Hot Plug);
- независимые источники опорного сигнала (SRIS);
- прерывания MSI-X;
- D2 состояние PCI функции.

На показанной ниже структурной схеме (см. Рисунок 5.25) можно выделить следующие составные блоки контроллера PCIe:

- блок физического интерфейса (PCI Express PHY);
- интерфейсный блок контроллера (PIPE Bridge);
- контроллер физического уровня (Physical Layer);
- контроллер канального уровня (Data Link Layer);
- контроллер уровня транзакций (Transaction Layer);
- интерфейсный блок шины AXI (AXI Bridge);
- контроллер прямого доступа в память (DMAC);
- блок конфигурационных регистров (Configuration Register Set).

Системный контроллер CSC управляет сбросами, базовой конфигурацией и прерываниями PCIe.

Рисунок 5.25 - Структурная схема контроллера интерфейса PCIe

5.12.2 Программная модель

Контроллер интерфейса PCIe имеет множество регистров, доступных управляющему скалярному процессору ARM. К некоторым регистрам также имеют доступ внешние устройства PCI Express. Программно доступная область имеет базовое смещение 0x000D_8000 hex и общий размер 32 КБайт. Все регистры можно условно разделить по назначению на 4 типа:

- Стандартные конфигурационные регистры PCI Express (Physical Function Configuration Register Set – см. таблицу 5.21, Root Port Configuration Register Set - см. таблицу 5.22). Выбор режима Root Port или End Point изменяет их набор,
- Регистры локального управления (Local Management Registers - см. таблицу 5.23),
- Регистры обработки AXI сообщений (AXI Configuration Registers - см. таблицу 5.24),
- Регистры управления ПДП (PCIe DMA Registers - см. таблицу 5.25).

Программный доступ к некоторым стандартным конфигурационным регистрам в RP режиме может быть расширен с R или R/WOCLR до R/W. Для этого при обращении к ним надо также установить [13] бит адреса.

Адреса регистров, приведенные в таблицах ниже – адреса DWORD. Поддерживается байтовая запись в регистры. Неопределенные адреса зарезервированы. Конфигурационный запрос по шине PCIe по зарезервированным адресам вызовет завершение с UR (Unsupported Request) кодом. Чтение по шине APB по зарезервированным адресам вернет 0. Запись по шине APB по зарезервированным адресам не возымеет действия.

Выбор режимов работы, управление сбросами, прерывания и статус интерфейса контролируются через программные регистры системного контроллера CSC и описаны в соответствующем разделе.

Блок физического интерфейса имеет собственные программируемые регистры, содержащие физические параметры. При сбросе они принимают стандартные значения и не требуют программной установки. Ввиду этого, а так же большого их количества, эти регистры не описываются в данной документации. Спецификация на программируемую часть PCIe Phy может быть предоставлена по запросу.

Таблица 5.21 – Стандартные конфигурационные регистры (режим End Point)

Адрес (байтовый доступ)	Название регистра
0x000D_8000 hex	PCIe_EP_i_vendor_id_device_id
0x000D_8004 hex	PCIe_EP_i_command_status
0x000D_8008 hex	PCIe_EP_i_revision_id_class_code
0x000D_800C hex	PCIe_EP_i_bist_header_latency_cache_line
0x000D_8010 hex	PCIe_EP_i_base_addr_0
0x000D_8014 hex	PCIe_EP_i_base_addr_1
0x000D_8018 hex	PCIe_EP_i_base_addr_2
0x000D_801C hex	PCIe_EP_i_base_addr_3
0x000D_8020 hex	PCIe_EP_i_base_addr_4
0x000D_8024 hex	PCIe_EP_i_base_addr_5
0x000D_802C hex	PCIe_EP_i_subsystem_vendor_id_subsystem_i
0x000D_8034 hex	PCIe_EP_i_capabilities_pointer
0x000D_803C hex	PCIe_EP_i_intrpt_line_intrpt_pin
0x000D_8080 hex	PCIe_EP_i_pwr_mgmt_cap
0x000D_8084 hex	PCIe_EP_i_pwr_mgmt_ctrl_stat_rep
0x000D_8090 hex	PCIe_EP_i_msi_ctrl_reg
0x000D_8094 hex	PCIe_EP_i_msi_msg_low_addr
0x000D_8098 hex	PCIe_EP_i_msi_msg_hi_addr
0x000D_809C hex	PCIe_EP_i_msi_msg_data
0x000D_80A0 hex	PCIe_EP_i_msi_mask
0x000D_80A4 hex	PCIe_EP_i_msi_pending_bits
0x000D_80C0 hex	PCIe_EP_i_pcie_cap_list
0x000D_80C4 hex	PCIe_EP_i_pcie_dev_cap
0x000D_80C8 hex	PCIe_EP_i_pcie_dev_ctrl_status
0x000D_80CC hex	PCIe_EP_i_link_cap
0x000D_80D0 hex	PCIe_EP_i_link_ctrl_status
0x000D_80E4 hex	PCIe_EP_i_pcie_dev_cap_2
0x000D_80E8 hex	PCIe_EP_i_pcie_dev_ctrl_status_2
0x000D_80EC hex	PCIe_EP_i_link_cap_2_reg
0x000D_80F0 hex	PCIe_EP_i_link_ctrl_status_2
0x000D_8100 hex	PCIe_EP_i_AER_enhanced_cap_hdr
0x000D_8104 hex	PCIe_EP_i_uncorr_err_status
0x000D_8108 hex	PCIe_EP_i_uncorr_err_mask
0x000D_810C hex	PCIe_EP_i_uncorr_err_severity
0x000D_8110 hex	PCIe_EP_i_corr_err_status
0x000D_8114 hex	PCIe_EP_i_corr_err_mask
0x000D_8118 hex	PCIe_EP_i_advcd_err_cap_ctrl
0x000D_811C hex	PCIe_EP_i_hdr_log_0
0x000D_8120 hex	PCIe_EP_i_hdr_log_1
0x000D_8124 hex	PCIe_EP_i_hdr_log_2
0x000D_8128 hex	PCIe_EP_i_hdr_log_3

Таблица 5.22 – Стандартные конфигурационные регистры (режим Root Port)

Адрес (байтовый доступ)	Название регистра
0x000D_8000 hex	PCIe_RP_i_vendor_id_device_id
0x000D_8004 hex	PCIe_RP_i_command_status
0x000D_8008 hex	PCIe_RP_i_revision_id_class_code
0x000D_800C hex	PCIe_RP_i_bist_header_latency_cache_line
0x000D_8010 hex	PCIe_RP_i_RC_BAR_0
0x000D_8014 hex	PCIe_RP_i_RC_BAR_1
0x000D_8018 hex	PCIe_RP_i_pcie_bus_numbers
0x000D_801C hex	PCIe_RP_i_pcie_io_base_limit
0x000D_8020 hex	PCIe_RP_i_pcie_mem_base_limit
0x000D_8024 hex	PCIe_RP_i_pcie_prefetch_base_limit
0x000D_8028 hex	PCIe_RP_i_pcie_prefetch_base_upper
0x000D_802C hex	PCIe_RP_i_pcie_prefetch_limit_upper
0x000D_8030 hex	PCIe_RP_i_pcie_io_base_limit_upper
0x000D_8034 hex	PCIe_RP_i_capabilities_pointer
0x000D_803C hex	PCIe_RP_i_intrpt_line_intrpt_pin
0x000D_8080 hex	PCIe_RP_i_pwr_mgmt_cap
0x000D_8084 hex	PCIe_RP_i_pwr_mgmt_ctrl_stat_rep
0x000D_8090 hex	PCIe_RP_i_msi_ctrl_reg
0x000D_8094 hex	PCIe_RP_i_msi_msg_low_addr
0x000D_8098 hex	PCIe_RP_i_msi_msg_hi_addr
0x000D_809C hex	PCIe_RP_i_msi_msg_data
0x000D_80A0 hex	PCIe_RP_i_msi_mask
0x000D_80A4 hex	PCIe_RP_i_msi_pending_bits
0x000D_80C0 hex	PCIe_RP_i_pcie_cap_list
0x000D_80C4 hex	PCIe_RP_i_pcie_cap
0x000D_80C8 hex	PCIe_RP_i_pcie_dev_ctrl_status
0x000D_80CC hex	PCIe_RP_i_link_cap
0x000D_80D0 hex	PCIe_RP_i_link_ctrl_status
0x000D_80D8 hex	PCIe_RP_i_slot_ctrl_status
0x000D_80DC hex	PCIe_RP_i_root_ctrl_cap
0x000D_80E0 hex	PCIe_RP_i_root_status
0x000D_80E4 hex	PCIe_RP_i_pcie_cap_2
0x000D_80E8 hex	PCIe_RP_i_pcie_dev_ctrl_status_2
0x000D_80EC hex	PCIe_RP_i_link_cap_2
0x000D_80F0 hex	PCIe_RP_i_link_ctrl_status_2
0x000D_8100 hex	PCIe_RP_i_AER_enhncd_cap
0x000D_8104 hex	PCIe_RP_i_uncorr_err_status
0x000D_8108 hex	PCIe_RP_i_uncorr_err_mask
0x000D_810C hex	PCIe_RP_i_uncorr_err_severity
0x000D_8110 hex	PCIe_RP_i_corr_err_status
0x000D_8114 hex	PCIe_RP_i_corr_err_mask
0x000D_8118 hex	PCIe_RP_i_adv_err_cap_ctl
0x000D_811C hex	PCIe_RP_i_hdr_log_0
0x000D_8120 hex	PCIe_RP_i_hdr_log_1
0x000D_8124 hex	PCIe_RP_i_hdr_log_2
0x000D_8128 hex	PCIe_RP_i_hdr_log_3
0x000D_812C hex	PCIe_RP_i_root_err_cmd
0x000D_8130 hex	PCIe_RP_i_root_err_stat
0x000D_8134 hex	PCIe_RP_i_err_src_id

Таблица 5.23 – Регистры локального управления.

Адрес (байтовый доступ)	Название регистра
0x000B_9000 hex	PCIe_LocMgmt_i_pl_config_0_reg
0x000B_9004 hex	PCIe_LocMgmt_i_pl_config_1_reg
0x000B_9008 hex	PCIe_LocMgmt_i_dll_tmr_config_reg
0x000B_900C hex	PCIe_LocMgmt_i_rcv_cred_lim_0_reg
0x000B_9010 hex	PCIe_LocMgmt_i_rcv_cred_lim_1_reg
0x000B_9014 hex	PCIe_LocMgmt_i_transm_cred_lim_0_reg
0x000B_9018 hex	PCIe_LocMgmt_i_transm_cred_lim_1_reg
0x000B_901C hex	PCIe_LocMgmt_i_transm_cred_update_int_config_0_reg
0x000B_9020 hex	PCIe_LocMgmt_i_transm_cred_update_int_config_1_reg
0x000B_9024 hex	PCIe_LocMgmt_i_L0S_timeout_limit_reg
0x000B_9028 hex	PCIe_LocMgmt_i_transmit_tlp_count_reg
0x000B_902C hex	PCIe_LocMgmt_i_transmit_tlp_payload_dword_count_reg
0x000B_9030 hex	PCIe_LocMgmt_i_receive_tlp_count_reg
0x000B_9034 hex	PCIe_LocMgmt_i_receive_tlp_payload_dword_count_reg
0x000B_9038 hex	PCIe_LocMgmt_i_compln_tmout_lim_0_reg
0x000B_903C hex	PCIe_LocMgmt_i_compln_tmout_lim_1_reg
0x000B_9040 hex	PCIe_LocMgmt_i_L1_st_reentry_delay_reg
0x000B_9044 hex	PCIe_LocMgmt_i_vendor_id_reg
0x000B_9048 hex	PCIe_LocMgmt_i_aspm_L1_entry_tmout_delay_reg
0x000B_904C hex	PCIe_LocMgmt_i_pme_turnoff_ack_delay_reg
0x000B_9050 hex	PCIe_LocMgmt_i_linkwidth_control_reg
0x000B_9074 hex	PCIe_LocMgmt_i_sris_control_reg
0x000B_9100 hex	PCIe_LocMgmt_i_shdw_hdr_log_0_reg
0x000B_9104 hex	PCIe_LocMgmt_i_shdw_hdr_log_1_reg
0x000B_9108 hex	PCIe_LocMgmt_i_shdw_hdr_log_2_reg
0x000B_910C hex	PCIe_LocMgmt_i_shdw_hdr_log_3_reg
0x000B_9110 hex	PCIe_LocMgmt_i_shdw_func_num_reg
0x000B_9114 hex	PCIe_LocMgmt_i_shdw_ur_err_reg
0x000B_9144 hex	PCIe_LocMgmt_i_debug_dllp_count_gen1_reg
0x000B_9148 hex	PCIe_LocMgmt_i_debug_dllp_count_gen2_reg
0x000B_914C hex	PCIe_LocMgmt_i_debug_dllp_count_gen3_reg
0x000B_9200 hex	PCIe_LocMgmt_i_negotiated_lane_map_reg
0x000B_9204 hex	PCIe_LocMgmt_i_receive_fts_count_reg
0x000B_9208 hex	PCIe_LocMgmt_i_debug_mux_control_reg
0x000B_920C hex	PCIe_LocMgmt_i_local_error_status_register
0x000B_9210 hex	PCIe_LocMgmt_i_local_intrpt_mask_reg
0x000B_9214 hex	PCIe_LocMgmt_i_lcrc_err_count_reg
0x000B_9218 hex	PCIe_LocMgmt_i_ecc_corr_err_count_reg
0x000B_9224 hex	PCIe_LocMgmt_i_pme_service_timeout_delay_reg
0x000B_9228 hex	PCIe_LocMgmt_i_root_port_requestor_id_reg
0x000B_922C hex	PCIe_LocMgmt_i_ep_bus_device_number_reg
0x000B_9240 hex	PCIe_LocMgmt_i_pf_0_BAR_config_0_reg
0x000B_9244 hex	PCIe_LocMgmt_i_pf_0_BAR_config_1_reg
0x000B_92C0 hex	PCIe_LocMgmt_i_pf_config_reg
0x000B_9300 hex	PCIe_LocMgmt_i_rc_BAR_config_reg
0x000B_9C80 hex	PCIe_LocMgmt_i_ecc_corr_err_count_reg_axi
0x000B_9C88 hex	PCIe_LocMgmt_low_power_debug_and_control0
0x000B_9C8C hex	PCIe_LocMgmt_low_power_debug_and_control1
0x000B_9C94 hex	PCIe_LocMgmt_tl_internal_control

Таблица 5.24 – Регистры обработки AXI сообщений.

Базовый адрес (байтовый доступ)	Адрес (байтовый доступ)	Название регистра	Название набора регистров
0x000B_C000 hex	0x000B_C000 hex	PCIe_AXI_addr0	Outbound Region 0 Register Set (наборы регистров для других регионов аналогичные)
	0x000B_C004 hex	PCIe_AXI_addr1	
	0x000B_C008 hex	PCIe_AXI_desc0	
	0x000B_C00C hex	PCIe_AXI_desc1	
	0x000B_C018 hex	PCIe_AXI_axi_addr0	
0x000B_C020 hex			Outbound Region 1 Register Set
0x000B_C040 hex			Outbound Region 2 Register Set
0x000B_C060 hex			Outbound Region 3 Register Set
0x000B_C080 hex			Outbound Region 4 Register Set
0x000B_C0A0 hex			Outbound Region 5 Register Set
0x000B_C0C0 hex			Outbound Region 6 Register Set
0x000B_C0E0 hex			Outbound Region 7 Register Set
0x000B_C100 hex			Outbound Region 8 Register Set
0x000B_C120 hex			Outbound Region 9 Register Set
0x000B_C140 hex			Outbound Region 10 Register Set
0x000B_C160 hex			Outbound Region 11 Register Set
0x000B_C180 hex			Outbound Region 12 Register Set
0x000B_C1A0 hex			Outbound Region 13 Register Set
0x000B_C1C0 hex			Outbound Region 14 Register Set
0x000B_C1E0 hex			Outbound Region 15 Register Set
0x000B_C200 hex			Outbound Region 16 Register Set
0x000B_C220 hex			Outbound Region 17 Register Set
0x000B_C240 hex			Outbound Region 18 Register Set
0x000B_C260 hex			Outbound Region 19 Register Set
0x000B_C280 hex			Outbound Region 20 Register Set

Продолжение таблицы 5.24

Базовый адрес (байтовый доступ)	Адрес (байтовый доступ)	Название регистра	Название набора регистров
0x000B_C2A0 hex			Outbound Region 21 Register Set
0x000B_C2C0 hex			Outbound Region 22 Register Set
0x000B_C2E0 hex			Outbound Region 23 Register Set
0x000B_C300 hex			Outbound Region 24 Register Set
0x000B_C320 hex			Outbound Region 25 Register Set
0x000B_C340 hex			Outbound Region 26 Register Set
0x000B_C360 hex			Outbound Region 27 Register Set
0x000B_C380 hex			Outbound Region 28 Register Set
0x000B_C3A0 hex			Outbound Region 29 Register Set
0x000B_C3C0 hex			Outbound Region 30 Register Set
0x000B_C3E0 hex			Outbound Region 31 Register Set
	0x000B_C800 hex	PCIe_AXI_RP_Inb_BAR_0_Addr	
	0x000B_C808 hex	PCIe_AXI_RP_Inb_BAR_1_Addr	
	0x000B_C810 hex	PCIe_AXI_RP_Inb_BAR_2_Addr	
	0x000B_C824 hex	PCIe_AXI_L0	
	0x000B_C840 hex	PCIe_AXI_EP_Inb_BAR_0_Addr	
	0x000B_C848 hex	PCIe_AXI_EP_Inb_BAR_1_Addr	
	0x000B_C850 hex	PCIe_AXI_EP_Inb_BAR_2_Addr	
	0x000B_C858 hex	PCIe_AXI_EP_Inb_BAR_3_Addr	
	0x000B_C860 hex	PCIe_AXI_EP_Inb_BAR_4_Addr	
	0x000B_C868 hex	PCIe_AXI_EP_Inb_BAR_5_Addr	
	0x000B_C870 hex	PCIe_AXI_EP_Inb_BAR_6_Addr	

Таблица 5.25 – Регистры управления ПДП

Адрес (байтовый доступ)	Название регистра
0x000B_D000 hex	PCIe_DMA_channel_0_ctrl
0x000B_D004 hex	PCIe_DMA_channel_0_sp_l
0x000B_D008 hex	PCIe_DMA_channel_0_sp_u
0x000B_D00C hex	PCIe_DMA_channel_0_attr_l
0x000B_D010 hex	PCIe_DMA_channel_0_attr_u
0x000B_D014 hex	PCIe_DMA_channel_1_ctrl
0x000B_D018 hex	PCIe_DMA_channel_1_sp_l
0x000B_D01C hex	PCIe_DMA_channel_1_sp_u
0x000B_D020 hex	PCIe_DMA_channel_1_attr_l
0x000B_D024 hex	PCIe_DMA_channel_1_attr_u
0x000B_D0A0 hex	PCIe_DMA_common_udma_int
0x000B_D0A4 hex	PCIe_DMA_common_udma_int_ena
0x000B_D0A8 hex	PCIe_DMA_common_udma_int_dis
0x000B_D0AC hex	PCIe_DMA_common_udma_ib_ecc_uncorrectable_errors
0x000B_D0B0 hex	PCIe_DMA_common_udma_ib_ecc_correctable_errors
0x000B_D0B4 hex	PCIe_DMA_common_udma_ob_ecc_uncorrectable_errors
0x000B_D0B8 hex	PCIe_DMA_common_udma_ob_ecc_correctable_errors
0x000B_D0F8 hex	PCIe_DMA_common_udma_cap_ver
0x000B_D0FC hex	PCIe_DMA_common_udma_config

5.12.2.1 Описание программно доступных стандартных конфигурационных регистров (режим End Point)

В этом разделе при описании программно доступных регистров поле «Тип доступа» в таблицах относится к конфигурационным обращениям по шине PCIe. Возможности программного доступа по шине APB приведены в поле «Описание».

Регистр идентификации изготовителя и устройства (PCIe_EP_i_vendor_id_device_id)

Таблица 5.26 - Описание полей регистра PCIe_EP_i_vendor_id_device_id

Биты	Тип доступа	Начальное значение	Описание
15:0	R	16'h17cd	Vendor ID : ID разработчика микросхемы, присвоенное ему PCI SIG. Значение настраивается программно в регистре PCIe_LocMgmt_i_vendor_id_reg.
31:16	R	16'h100	Device ID : ID устройства, присвоенное ему разработчиком микросхемы. Доступен по APB шине на запись.

Регистр управления и состояния (PCIe_EP_i_command_status)

Таблица 5.27 - Описание полей регистра PCIe_EP_i_command_status

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	IO-Space Enable : разрешение передачи сообщений типа IO через контроллер. Доступен по APB шине на запись.
1	R/W	0x0	Mem-Space Enable : разрешение передачи сообщений типа Memory через контроллер. Доступен по APB шине на запись.
2	R/W	0x0	Bus-Master Enable : разрешение приема сообщений типа IO и Memory от контроллера. Доступен по APB шине на запись.
5:3	R	0x0	Reserved
6	R/W	0x0	Parity Error Response Enable : 1 – установка бита Master Data Parity Error разрешена; 0 – установка Master Data Parity Error запрещена. Доступен по APB шине на запись.
7	R	0x0	Reserved
8	R/W	0x0	SERR Enable : 1 – разрешает отправку сообщений о фатальных и не фатальных ошибках хост-контроллеру; 0 – отправка запрещена. Доступен по APB шине на запись.
9	R	0x0	Reserved
10	R/W	0x0	INTx Message Disabled : 1 – INTx (Legacy) прерывания запрещены; 0 – INTx (Legacy) прерывания разрешены. Доступен по APB шине на запись.
15:11	R	0x0	Reserved
18:16	R	0x0	Reserved
19	R	0x0	Interrupt Status : устанавливается, только если прерывания INTx (Legacy) разрешены. Устанавливается, если сообщение Assert_INTx было оправлено, а сообщение Deassert_INTx не было отправлено.
20	R	0x1	Capabilities List : указывает на наличие расширенного списка регистров (PCI Extended Capabilities registers). Всегда равно 1.
23:21	R	0x0	Reserved
24	R/WOCLR	0x0	Master Data Parity Error: устанавливается при получении завершения с ошибкой (poisoned completion) или отправлении записи с ошибкой (poisoned write request). Этот бит может быть сброшен записью 1 по шине APB.
26:25	R	0x0	Reserved
27	R/WOCLR	0x0	Signaled Target Abort : устанавливается, если контроллер послал сообщение завершения со статусом Completer Abort. Этот бит может быть сброшен записью 1 по шине APB.
28	R/WOCLR	0x0	Received Target Abort : устанавливается, если контроллер принял сообщение завершения со статусом Completer Abort. Этот бит может быть сброшен записью 1 по шине APB.
29	R/WOCLR	0x0	Received Master Abort : устанавливается, если контроллер принял сообщение завершения со статусом Unsupported Request. Этот бит может быть сброшен записью 1 по шине APB.
30	R/WOCLR	0x0	Signaled System Error : устанавливается при отправлении контроллером хост-контроллеру сообщения о фатальной или не фатальной ошибке. Этот бит может быть сброшен записью 1 по шине APB.
31	R/WOCLR	0x0	Detected Parity Error : устанавливается при получении контроллером сообщения TLP с ошибкой. Этот бит может быть сброшен записью 1 по шине APB.

Регистр номера версии и текущей функции (PCIe_EP_i_revision_id_class_code)

Таблица 5.28 - Описание полей регистра PCIe_EP_i_revision_id_class_code

Биты	Тип доступа	Начальное значение	Описание
7:0	R	8'h0	Revision ID : указывает версию микросхемы, устанавливается разработчиком микросхемы. Доступен по APB шине на запись.
15:8	R	8'h0	Programming Interface Byte : показывает набор регистров устройства. Доступен по APB шине на запись.
23:16	R	8'h0	Sub-Class Code : содержит подкатегорию текущей функции. Доступен по APB шине на запись.
31:24	R	8'h0	Class Code : содержит номер текущей функции. Доступен по APB шине на запись.

Регистр управления встроенным тестированием, формата заголовка, таймера задержек и размера страниц кэш (PCIe_EP_i_bist_header_latency_cache_line)

Таблица 5.29 - Описание полей регистра PCIe_EP_i_bist_header_latency_cache_line

Биты	Тип доступа	Начальное значение	Описание
7:0	R/W	8'h0	Cache Line Size: см. PCI Specifications 3.0. Регистр доступен по PCIe и APB шине на чтение и запись, но его значение не используется.
15:8	R	8'h0	Latency Timer: не используется. Всегда равно 0.
22:16	R	7'h0	Header Type: формат заголовка. Всегда равно 0.
23	R	1'h0	Device Type: всегда равно 0.
31:24	R	8'h0	BIST Register: регистр управления BIST. Доступен по APB шине на запись.

Регистр базового адреса 0 (PCIe_EP_i_base_addr_0)

Это один из шести BAR, определенных согласно PCI Specifications 3.0. Эти регистры используются для определения диапазонов адресов для Memory и IO запросов к EP устройствам. В ходе начальной инициализации устройства, конфигурирующая программа определяет размер диапазона адресов путем записи всех 1 в BAR, чтения из BAR и определения положения первого MSB, равного 1. Если считаны все нули, этот BAR выключен. BAR 0 может быть настроен как BAR 32-битных транзакций или спарен с BAR 1 для 64-битных транзакций.

Таблица 5.30 - Описание полей регистра PCIe_EP_i_base_addr_0

Биты	Тип доступа	Начальное значение	Описание
0	R	0x0	Memory Space Indicator : тип области, соответствующей BAR 0: 0 – memory; 1 – IO. Значение настраивается программно в регистре PCIe_LocMgmt_i_pf_0_BAR_config_0_reg.
1	R	0x0	Reserved
2	R	0x0	Size: ширина адреса (только для memory области): 0 – 32 бита; 1 – 64 бита, в этом случае BAR 1 дополняет BAR 0 (только для BAR с четными номерами). Значение настраивается программно в регистре PCIe_LocMgmt_i_pf_0_BAR_config_0_reg.
3	R	0x0	Prefetchability: возможность предвыборки данных (только для memory области). Значение настраивается программно в регистре PCIe_LocMgmt_i_pf_0_BAR_config_0_reg.
7:4	R	0x0	Reserved
17:8	R	0x0	Base Address - RO part: определяет базовый адрес региона. Записываемое значение маскируется значением размера региона из регистра PCIe_LocMgmt_i_pf_0_BAR_config_0_reg . Доступен по APB шине на запись.
31:18	R/W	0x0	Base Address - RW part: определяет базовый адрес региона. Записываемое значение маскируется значением размера региона из регистра PCIe_LocMgmt_i_pf_0_BAR_config_0_reg. Доступен по APB шине на запись.

Регистр базового адреса 1 (PCIe_EP_i_base_addr_1)

Это один из шести BAR, определенных согласно PCI Specifications 3.0. BAR 1 может быть настроен как BAR 32-битных транзакций или спарен с BAR 0 для 64-битных транзакций. Во втором случае хранит 32 старших бита базового адреса.

Таблица 5.31 - Описание полей регистра PCIe_EP_i_base_addr_1

Биты	Тип доступа	Начальное значение	Описание
0	R	0x0	Memory Space Indicator : тип области, соответствующей BAR 1 0 – memory 1 – IO Значение настраивается программно в регистре PCIe_LocMgmt_i_pf_0_BAR_config_0_reg.
1	R	0x0	Reserved
2	R	0x0	Size: только 32 битный адрес (для всех BAR с нечетными номерами) Значение настраивается программно в регистре PCIe_LocMgmt_i_pf_0_BAR_config_0_reg.
3	R	0x0	Prefetchability: возможность предвыборки данных (только для memory области) Значение настраивается программно в регистре PCIe_LocMgmt_i_pf_0_BAR_config_0_reg.
7:4	R	0x0	Reserved
19:8	R	0x0	Base Address - RO part: определяет базовый адрес региона. Записываемое значение маскируется значением размера региона из регистра PCIe_LocMgmt_i_pf_0_BAR_config_0_reg. Доступен по APB шине на запись.
31:20	R/W	0x0	Base Address - RW part: определяет базовый адрес региона. Записываемое значение маскируется значением размера региона из регистра PCIe_LocMgmt_i_pf_0_BAR_config_0_reg. Доступен по APB шине на запись.

Регистр базового адреса 2 (PCIe_EP_i_base_addr_2)

Это один из шести BAR, определенных согласно PCI Specifications 3.0. Эти регистры используются для определения диапазонов адресов для Memory и IO запросов к EP устройствам. В ходе начальной инициализации устройства, конфигурирующая программа определяет размер диапазона адресов путем записи всех 1 в BAR, чтения из BAR и определения положения первого MSB, равного 1. Если считаны все нули, этот BAR выключен. BAR 2 может быть настроен как BAR 32-битных транзакций или спарен с BAR 3 для 64-битных транзакций.

Таблица 5.32 - Описание полей регистра PCIe_EP_i_base_addr_2

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	После снятия сброса этот регистр зарезервирован, область 2 не активна. Назначения полей такие же, как у остальных четных BAR.

Регистр базового адреса 3 (PCIe_EP_i_base_addr_3)

Это один из шести BAR, определенных согласно PCI Specifications 3.0. BAR 3 может быть настроен как BAR 32-битных транзакций или спарен с BAR 2 для 64-битных транзакций. Во втором случае хранит 32 старших бита базового адреса.

Таблица 5.33 - Описание полей регистра PCIe_EP_i_base_addr_3

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	После снятия сброса этот регистр зарезервирован, область 3 не активна. Назначения полей такие же, как у остальных нечетных BAR.

Регистр базового адреса 4 (PCIe_EP_i_base_addr_4)

Это один из шести BAR, определенных согласно PCI Specifications 3.0. Эти регистры используются для определения диапазонов адресов для Memory и IO запросов к EP устройствам. В ходе начальной инициализации устройства, конфигурирующая программа определяет размер диапазона адресов путем записи всех 1 в BAR, чтения из BAR и определения положения первого MSB, равного 1. Если считаны все нули, этот BAR выключен. BAR 4 может быть настроен как BAR 32-битных транзакций или спарен с BAR 5 для 64-битных транзакций.

Таблица 5.34 - Описание полей регистра PCIe_EP_i_base_addr_4

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	После снятия сброса этот регистр зарезервирован, область 4 не активна. Назначения полей такие же, как у остальных четных BAR.

Регистр базового адреса 5 (PCIe_EP_i_base_addr_5)

Это один из шести BAR, определенных согласно PCI Specifications 3.0. BAR 5 может быть настроен как BAR 32-битных транзакций или спарен с BAR 4 для 64-битных транзакций. Во втором случае хранит 32 старших бита базового адреса.

Таблица 5.35 - Описание полей регистра PCIe_EP_i_base_addr_5

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	После снятия сброса этот регистр зарезервирован, область 5 не активна. Назначения полей такие же, как у остальных нечетных BAR.

Регистр номера подверсии и подсистемы (PCIe_EP_i_subsystem_vendor_id_subsystem_i)

Таблица 5.36 - Описание полей регистра PCIe_EP_i_subsystem_vendor_id_subsystem_i

Биты	Тип доступа	Начальное значение	Описание
15:0	R	16'h17cd	Subsystem Vendor ID : ID разработчика подсистемы, присвоенное ему PCI SIG. Значение настраивается программно в регистре PCIe_LocMgmt_i_vendor_id_reg.
31:16	R	16'h0	Subsystem ID : ID подсистемы, присвоенное ему разработчиком микросхемы. Доступен по APB шине на запись.

Регистр указателя характеристик (PCIe_EP_i_capabilities_pointer)

Таблица 5.37 - Описание полей регистра PCIe_EP_i_capabilities_pointer

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x80	Capabilities Pointer: содержит указатель на первую структуру характеристик (PCI Capability Structure). По умолчанию это структура управления питанием (Power Management Capability Structure). Доступен по APB шине на запись.
31:8	R	0x0	Reserved

Регистр линий и выводов прерываний (PCIe_EP_i_intrpt_line_intrpt_pin)

Этот регистр используется, только если контроллер поддерживает Legacy прерывания.

Таблица 5.38 - Описание полей регистра PCIe_EP_i_intrpt_line_intrpt_pin

Биты	Тип доступа	Начальное значение	Описание
7:0	R/W	8'hff	Interrupt Line Register : параметр не поддерживается.
10:8	R	0x1	Interrupt Pin Register : указывает вывод, на который выдается Legacy прерывание, полученное по PCIe интерфейсу: 0x1 – INTA; 0x2 – INTB; 0x3 – INTC; 0x4 – INTD.
31:11	R	0x0	Reserved

Регистр характеристик управления питанием (PCIe_EP_i_pwr_mgmt_cap)

Таблица 5.39 - Описание полей регистра PCIe_EP_i_pwr_mgmt_cap

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x01	Capability ID: показывает, что это структура управления питанием. Доступен по APB шине на запись.
15:8	R	8'h90	Capabilities Pointer: содержит указатель на следующую структуру характеристик (PCI Capability Structure). Доступен по APB шине на запись.
18:16	R	0x3	Version ID: показывает версию системы управления питанием, поддерживаемую текущей функцией. Доступен по APB шине на запись.
19	R	0x0	PME Clock: всегда равен 0
20	R	0x0	Reserved
21	R	0x0	Device Specific Initialization Bit: всегда равен 0
24:22	R	0x0	Max Current Required from Aux Power Suppl : всегда равен 0
25	R	0x1	D1 Support: показывает возможность поддержки PCI функцией состояния D1. Доступен по APB шине на запись.
26	R	0x0	D2 Support: показывает возможность поддержки PCI функцией состояния D2, всегда равен 0.
27	R	0x01	PME Support for D0 State: показывает, может ли эта PCI функция посылать сообщения о событиях управления питанием (PME messages) находясь в состоянии D0. Доступен по APB шине на запись.
28	R	0x1	PME Support for D1 State: показывает, может ли эта PCI функция посылать сообщения о событиях управления питанием (PME messages) находясь в состоянии D1. Доступен по APB шине на запись.
29	R	0x0	PME Support for D2 State: всегда равен 0 (D2 состояние не поддерживается)
30	R	0x01	PME Support for D3 (hot) Statue: показывает, может ли эта PCI функция посылать сообщения о событиях управления питанием (PME messages) находясь в состоянии D3 (hot). Доступен по APB шине на запись.
31	R	0x0	PME Support for D3(cold) State : всегда равен 0

Регистр управления питанием и его состояния (PCIe_EP_i_pwr_mgmt_ctrl_stat_rep)

Таблица 5.40 - Описание полей регистра PCIe_EP_i_pwr_mgmt_ctrl_stat_rep

Биты	Тип доступа	Начальное значение	Описание
1:0	R/W	0x0	Power State: используется для отслеживания текущего состояния активности функции и для переходов в новое состояние: 00 – D0; 01 – D1; 10 – reserved; 11 – D3 (hot). Доступен по APB шине на запись.
2	R	0x0	Reserved
3	R	0x01	No Soft Reset: сохранение настроек PCI функции при переводе ее в D3(hot) состояние. Доступен по APB шине на запись.
7:4	R	0x0	Reserved
8	R/W	0x0	PME Enable: разрешение сообщений о событиях управления питанием (PME messages). Доступен по APB шине на запись.
14:9	R	0x0	Reserved
15	R/WOCLR	0x0	PME Status: При установленном бите PME Enable запись 1 по шине APB вызовет передачу PME сообщения (только для режима End Point). RC в процессе обработки сообщения должен сбросить этот бит посредством конфигурационной записи 1 (Config Write). Доступен по APB шине на запись.
23:16	R	0x0	Reserved
31:24	R	0x0	Data Register: всегда равен 0

Регистр управления сообщениями (PCIe_EP_i_msi_ctrl_reg)

Таблица 5.41 - Описание полей регистра PCIe_EP_i_msi_ctrl_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x05	Capability ID: показывает, что это структура управления MSI прерываниями. Всегда равно 0x05.
15:8	R	8'hc0	Capabilities Pointer: содержит указатель на следующую структуру характеристик (PCI Capability Structure). Доступен по APB шине на запись.
16	R/W	0x0	MSI Enable: разрешение MSI прерываний. Доступен по APB шине на запись.
19:17	R	0x0	Multiple Message Capable: количество различных MSI сообщений, которые может генерировать контроллер для этой функции (т.е. количество разных векторов прерываний): 000 = 1; 001 = 2; 010 = 4; 011 = 8; 100 = 16; 101 = 32. Контроллер поддерживает всего до 32 различных прерываний, но установка этого поля должна быть основана на количестве прерываний, непосредственно используемых клиентом. Доступен по APB шине на запись.
22:20	R/W	0x0	Multiple Message Enable: количество различных MSI сообщений, которые генерирует контроллер для этой функции (т.е. количество разных векторов прерываний) 000 = 1; 001 = 2; 010 = 4; 011 = 8; 100 = 16; 101 = 32. Установка этого поля должна быть основана на количестве прерываний, непосредственно используемых функцией. Доступен по APB шине на запись.
23	R	0x1	64-Bit Address Capable: установка в 1 показывает, что устройство поддерживает 64-разрядную адресацию для MSI сообщений. Доступен по APB шине на запись.
24	R	0x1	MSI masking capable: Доступен по APB шине на запись.
31:25	R	0x0	Reserved

Регистр младшей части адреса MSI сообщений (PCIe_EP_i_msi_msg_low_addr)

Таблица 5.42 - Описание полей регистра PCIe_EP_i_msi_msg_low_addr

Биты	Тип доступа	Начальное значение	Описание
1:0	R	0x0	Reserved
31:2	R/W	0x0	Message Address Low : младшая часть адреса MSI сообщений. Доступен по APB шине на запись.

Регистр старшей части адреса MSI сообщений (PCIe_EP_i_msi_msg_hi_addr)

Таблица 5.43 - Описание полей регистра PCIe_EP_i_msi_msg_hi_addr

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	0x0	Message Address High : старшая часть адреса MSI сообщений. Если равна нулю, то адрес 32-разрядный. Доступен по APB шине на запись.

Регистр данных MSI сообщений (PCIe_EP_i_msi_msg_data)

Таблица 5.44 - Описание полей регистра PCIe_EP_i_msi_msg_data

Биты	Тип доступа	Начальное значение	Описание
15:0	R/W	0x0	Message Data : данные MSI сообщения. Доступен по APB шине на запись.
31:16	R	0x0	Reserved

Регистр масок пришедшего вектора MSI прерывания (PCIe_EP_i_msi_mask)

Таблица 5.45 - Описание полей регистра PCIe_EP_i_msi_mask

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	MSI Mask : маска битов пришедшего вектора MSI прерывания. Ширина этого параметра меняется в зависимости от Multiple Message Enable и, соответственно, количества векторов прерываний. Доступен по APB шине на запись.
31:1	R	0x0	Reserved Ширина этого параметра меняется в зависимости от Multiple Message Enable и, соответственно, количества векторов прерываний.

Регистр масок отправленного вектора MSI прерывания (PCIe_EP_i_msi_pending_bits)

Таблица 5.46 - Описание полей регистра PCIe_EP_i_msi_pending_bits

Биты	Тип доступа	Начальное значение	Описание
0	R	0x0	MSI Pending bits : должен быть программно установлен при попытке отправки замаскированного прерывания. Ширина этого параметра меняется в зависимости от Multiple Message Enable. Доступен по APB шине на запись.
31:1	R	0x0	Reserved Ширина этого параметра меняется в зависимости от Multiple Message Enable и, соответственно, количества векторов прерываний.

Регистр списка структур характеристик (PCIe_EP_i_pcie_cap_list)

Таблица 5.47 - Описание полей регистра PCIe_EP_i_pcie_cap_list

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x010	Capability ID : показывает, что это структура характеристик Express расширения. Всегда равно 0x010.
15:8	R	0x0	Next Capability Pointer : содержит указатель на следующую структуру характеристик. Равенство нулю показывает, что эта структура последняя. Всегда равно 0.
19:16	R	0x2	Capability Version : показывает версию этой структуры характеристик. В данном случае она совместима с PCI Express Base Specification Revision 3.0. Доступен по APB шине на запись.
23:20	R	0x0	Device Type : показывает тип устройства 0x0 – конечное устройство – End Point (EP) 0x4 – хост-контроллер – Root Port (RP) Всегда равно 0 в текущем режиме.
24	R	0x0	Slot Status : установлен в 1, если линк подключен к слоту. Всегда равно 0.
29:25	R	0x0	Interrupt Message Number : показывает вектор MSI или MSI-X прерывания в случае, если оно вызывается в соответствии с битами статуса в Slot Status Register, Root Status Register или текущей структуре характеристик. Значение устанавливается в зависимости от выбранного режима прерываний – MSI или MSI-X. Всегда равно 0.
30	R	0x0	TCS Routing Supported : 0 – устройство не поддерживает маршрутизацию доверенных конфигурационных запросов (Trusted Configuration Requests). 1 – опция поддерживается. Не используется в режиме конечного устройства. Всегда равно 0.
31	R	0x0	Reserved

Регистр характеристик устройства (PCIe_EP_i_pcie_dev_cap)

Таблица 5.48 - Описание полей регистра PCIe_EP_i_pcie_dev_cap

Биты	Тип доступа	Начальное значение	Описание
2:0	R	3'b011	Max Payload Size : показывает максимальное количество данных в сообщении. Доступен по APB шине на запись.
4:3	R	0x0	Phantom Functions Supported : расширение поля тэга за счет неиспользуемых битов. Не используется в этой микросхеме. Всегда равно 0.
5	R	0x0	Extended Tag Field Supported : расширенное поле tag не поддерживается. Всегда равно 0.
8:6	R	0x4	Acceptable L0S Latency : допустимая контроллером задержка перехода из L0s в L0 состояния. Доступен по APB шине на запись.
11:9	R	0x0	Acceptable L1 Latency : допустимая контроллером задержка перехода из L1 в L0 состояния. Доступен по APB шине на запись.
14:12	R	0x0	Reserved
15	R	0x01	Role-Based Error Reporting : разрешение ролевых (role-based) отчетов об ошибках. Доступен по APB шине на запись.
17:16	R	0x0	Reserved
25:18	R	0x0	Captured Slot Power Limit Value : максимальное значение потребления, поддерживаемое слотом. Доступен по APB шине на запись.
27:26	R	0x0	Captured Power Limit Scale : масштаб значения поля Captured Slot Power Limit Value текущего регистра. Доступен по APB шине на запись.
28	R	0x0	Function Level Reset Capability : Всегда равно 0.
31:29	R	0x0	Reserved

Регистр управления и состояния устройства (PCIe_EP_i_pcie_dev_ctrl_status)

Таблица 5.49 – Описание полей регистра PCIe_EP_i_pcie_dev_ctrl_status

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	Enable Correctable Error Reporting : разрешение посылки ERR_COR сообщений при обнаружении исправимой ошибки. Доступен по APB шине на запись.
1	R/W	0x0	Enable Non- Fatal Error Reporting : разрешение посылки ERR_NONFATAL сообщений при обнаружении не фатальной ошибки. Доступен по APB шине на запись.
2	R/W	0x0	Enable Fatal Error Reporting : разрешение посылки ERR_FATAL сообщений при обнаружении фатальной ошибки. Доступен по APB шине на запись.
3	R/W	0x0	Enable Unsupported Request Reporting : разрешение посылки сообщений об обнаружении неподдерживаемого запроса. Доступен по APB шине на запись.
4	R/W	0x01	Enable Relaxed Ordering : разрешает установку бита Relaxed Ordering в передаваемых транзакциях. Доступен по APB шине на запись.
7:5	R/W	0x0	Max Payload Size : показывает максимальное количество данных в сообщении. Устройство должно быть в состоянии принимать пакеты с таким количеством данных и не должно посылать пакеты большего размера. Устанавливается программно на основании данных из Device Capabilities Register этого и ответного устройств. Доступен по APB шине на запись.
8	R	0x0	Extended Tag Field Enable : расширение поля tag с 5 до 8 бит. Всегда равно 0.
9	R	0x0	Enable Phantom Functions : Всегда равно 0.
10	R	0x0	Enable Aux Power : Всегда равно 0.
11	R/W	0x1	Enable No Snoop : установка в 1 разрешает установку бита No Snoop в передаваемых транзакциях, не требующих когерентности кэш-памяти. Доступен по APB шине на запись.
14:12	R/W	0x02	Max Read Request Size : показывает максимальное количество данных в запросах чтения, сгенерированных этим устройством. Доступен по APB шине на запись.
15	R	0x0	Function-Level Reset : Всегда равно 0.

Продолжение таблицы 5.49

Биты	Тип доступа	Начальное значение	Описание
16	R/WOCLR	0x0	Correctable Error Detected : устанавливается контроллером, когда он обнаруживает исправимую ошибку. Не зависит от разрешения соответствующего сообщения и маскирования. Сброс при записи 1 по шине APB.
17	R/WOCLR	0x0	Non-Fatal Error Detected : устанавливается контроллером, когда он обнаруживает не фатальную ошибку. Не зависит от разрешения соответствующего сообщения и маскирования. Сброс при записи 1 по шине APB.
18	R/WOCLR	0x0	Fatal Error Detected : устанавливается контроллером, когда он обнаруживает фатальную ошибку. Не зависит от разрешения соответствующего сообщения и маскирования. Сброс при записи 1 по шине APB.
19	R/WOCLR	0x0	Unsupported Request Detected : устанавливается контроллером, когда он обнаруживает неподдерживаемый запрос. Не зависит от разрешения соответствующего сообщения. Сброс при записи 1 по шине APB.
20	R	0x0	Aux Power Detected : не используется. Всегда равно 0.
21	R	0x0	Transaction Pending : устанавливается, если есть ожидающие запросы, требующие подтверждения (Non-Posted). Только для чтения по шине APB.
31:22	R	0x0	Reserved

Регистр характеристик линка (PCIe_EP_i_link_cap)

Таблица 5.50 – Описание полей регистра PCIe_EP_i_link_cap

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x2	Maximum Link Speed : показывает максимальную скорость линка. Соответствует значению, установленному в системном контроллере. Только для чтения по шине APB.
9:4	R	0x4	Maximum Link Width : показывает максимальное количество линий. Соответствует значению, установленному в системном контроллере. Только для чтения по шине APB.
11:10	R	0x3	Active State Power Management : показывает уровень поддерживаемого ASPM. Доступен по APB шине на запись.
14:12	R	0x2	L0S Exit Latency : показывает время, требуемое устройству для перехода из состояния L0S в L0. Значение зависит от реализации Phy. Доступен по APB шине на запись.
17:15	R	0x3	L1 Exit Latency : показывает время, требуемое устройству для выхода из состояния L1. Значение зависит от реализации Phy. Доступен по APB шине на запись.
18	R	0x0	Clock Power Management : показывает, что устройство поддерживает отключение опорного тактового сигнала. Доступен по APB шине на запись.
19	R	0x0	Surprise Down Error Reporting Capability : показывает способность устройства сообщать о ошибке внезапного выключения (Surprize Down error). Эта функция не поддерживается. Всегда равно 0.
20	R	0x0	Data Link Layer Active Reporting Capability : показывает способность устройства сообщать о достижении автоматом управления пакетным уровнем состояния DL_Active. Эта функция не поддерживается. Всегда равно 0.
21	R	0x0	Link Bandwidth Notification Capability : поддержка уведомлений о ширине линка и механизме прерываний. Не используется в режиме Endpoint. Всегда равно 0.
22	R	0x1	ASPM Optionality Compliance : показывает поддержку устройством дополнительных ASPM возможностей. Доступен по APB шине на запись.
23	R	0x0	Reserved
31:24	R	8'h0	Port Number : показывает номер порта, присвоенный PCIe линку, подключенному к устройству. Доступен по APB шине на запись.

Регистр управления и состояния линка (PCIe_EP_i_link_ctrl_status)

Таблица 5.51– Описание полей регистра PCIe_EP_i_link_ctrl_status

Биты	Тип доступа	Начальное значение	Описание
1:0	R/W	0x0	Active State Power Management Control : управление ASPM возможностями: 00 – ASPM запрещен; 01 – L0s состояние разрешено, L1 запрещено; 10 – L0s состояние запрещено, L1 разрешено; 11 – L0s и L1 состояния разрешены. Эти биты могут быть установлены только при установленных соответствующих битах Active State Power Management регистра PCIe_EP_i_link_cap. Доступен по APB шине на запись.
2	R	0x0	Reserved
3	R/W	0x0	Read Completion Boundary : показывает границы завершения чтения для хост-контроллера, подключенного к этому конечному устройству: 0 – 64 байта; 1 – 128 байт. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
4	R	0x0	Link Disable : не используется в Endpoint режиме.
5	R	0x0	Retrain Link : не используется в Endpoint режиме.
6	R/W	0x0	Common Clock Configuration : показывает отношения опорного тактового сигнала этого EP и ответного устройства: 0 – асинхронные; 1 – синхронные. Доступен по APB шине на запись.
7	R/W	0x0	Extended Synch : установка этого бита увеличивает частоту передачи управляющих пакетов (Ordered Sets) при переходе из L0s в L0 состояние. Доступен по APB шине на запись.
8	R	0x0	Enable Clock Power Management : Когда этот бит установлен в 1, устройство может использовать вывод CLKREQ# PCIe слота для управления тактовым сигналом. Бит доступен на запись только при установленном Clock Power Management бите в Link Capability Register. Доступен по APB шине на запись.
9	R	0x0	Hardware Autonomous Width Disable : установленный, этот бит показывает программе, что она должна запрашивать изменение ширины линка только в случае некорректной работы текущего набора линий. Доступен по APB шине на запись.
10	R	0x0	Link Bandwidth Management Interrupt Enable : не используется в Endpoint режиме.
11	R	0x0	Link Autonomous Bandwidth Interrupt Enable : не используется в Endpoint режиме.
15:12	R	0x0	Reserved

Продолжение таблицы 5.51

Биты	Тип доступа	Начальное значение	Описание
19:16	R	0x2	Negotiated Link Speed : согласованная скорость передачи данных: 0001 – 2.5 Гб/с на линию; 0010 – 5 Гб/с на линию. Доступен по APB шине только на чтение.
25:20	R	0x4	Negotiated Link Width : согласованное количество линий. Устанавливается по окончании обучения линка. Доступен по APB шине только на чтение.
26	R	0x0	Reserved
27	R	0x0	Link Training Status : устанавливается аппаратно, если LTSSM в режиме Configuration или Recovery или в начале переобучения линка. Всегда равно 0 в режиме Endpoint. Доступен по APB шине только на чтение.
28	R	0x0	Slot Clock Configuration : показывает, что используется тактовый сигнал от разъёма. Если устройство использует независимый тактовый сигнал, несмотря на наличие такового на разъёме, этот бит должен быть сброшен в 0. Доступен по APB шине на запись.
29	R	0x0	Data Link Layer Active : показывает статус пакетного уровня. Всегда равно 0.
30	R/WOCLR	0x0	Link Bandwidth Management Status : устанавливается при аппаратном изменении ширины линка или скорости передачи, связанном с некорректной работой. Всегда равно 0, если Link Bandwidth Notification Capability равен 0. Всегда равно 0 в режиме Endpoint. Сброс при записи 1 по шине APB.
31	R/WOCLR	0x0	Link Autonomous Bandwidth Status : устанавливается при аппаратном изменении ширины линка или скорости передачи, не связанном с некорректной работой. Всегда равно 0, если Link Bandwidth Notification Capability равен 0. Всегда равно 0 в режиме Endpoint. Сброс при записи 1 по шине APB.

Регистр характеристик устройства 2 (PCIe_EP_i_pcie_dev_cap_2)

Таблица 5.52 - Описание полей регистра PCIe_EP_i_pcie_dev_cap_2

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x02	Completion Timeout Ranges : показывает поддерживаемый устройством таймаут подтверждения сообщения. По умолчанию 10-250 мс. Доступен по APB шине на запись.
4	R	0x01	Completion Timeout Disable Supported : показывает возможность функции выключить таймаут подтверждения сообщения. Доступен по APB шине на запись.
5	R	0x0	ARI forwarding support : всегда равно 0.
6	R	0x0	Atomic OP routing supported : маршрутизация атомарных операций (Atomic OP) поддерживается. Не используется в режиме Endpoint.
7	R	0x0	32-Bit Atomic Op Completer Supported : всегда равно 0.
8	R	0x0	64-Bit Atomic Op Completer Supported : всегда равно 0.
9	R	0x0	128-Bit CAS Atomic Op Completer Supported : всегда равно 0.
10	R	0x0	Reserved
11	R	0x0	LTR Mechanism Supported : всегда равно 0.
13:12	R	0x0	TPH Completer Supported : всегда равно 0.
17:14	R	0x0	Reserved
19:18	R	0x0	OBFF Supported : показывает, что функция поддерживает OBFF. Всегда равно 0.
20	R	0x1	Extended Format Field Supported : показывает, что функция поддерживает 3-битный формат поля Fmt заголовка TLP. Всегда равно 1.
21	R	0x0	End-End TLP Prefix Supported : показывает, что функция поддерживает End-End TLP Prefix. Всегда равно 0.
23:22	R	0x0	Max End-End TLP Prefixes : максимальное количество End-End TLP Prefixes 01 – 1 End-End TLP Prefix 10 – 2 End-End TLP Prefix Всегда равно 0.
31:24	R	0x0	Reserved

Регистр управления и состояния устройства 2 (PCIe_EP_i_pcie_dev_ctrl_status_2)

Таблица 5.53 – Описание полей регистра PCIe_EP_i_pcie_dev_ctrl_status_2

Биты	Тип доступа	Начальное значение	Описание
3:0	R/W	0x0	Completion Timeout Value : поддерживаемый устройством таймаут подтверждения сообщения: 0101 – диапазон 1; 0110 – диапазон 2. Соответствующие значения устанавливаются в регистрах Completion Timeout Interval Registers 0 и 1. Доступен по APB шине на запись.
4	R/W	0x0	Completion Timeout Disable : установка этого бита выключает таймаут подтверждения сообщений. Доступен по APB шине на запись.
5	R	0x0	ARI forwarding enable : всегда равно 0.
6	R	0x0	Atomic Op Requester Enable : разрешение генерации атомарных операций. Всегда равно 0.
7	R	0x0	Reserved
8	R	0x0	IDO Request Enable : когда этот бит установлен, функции разрешено устанавливать бит ID-based Ordering (IDO) Attribute в посылаемых запросах. Всегда равно 0.
9	R	0x0	IDO Completion Enable : когда этот бит установлен, функции разрешено устанавливать бит ID-based Ordering (IDO) Attribute в посылаемых подтверждениях. Всегда равно 0.
10	R	0x0	LTR Mechanism Enable : всегда равно 0.
12:11	R	0x0	Reserved
14:13	R	0x0	OBFF Enable : разрешение OBFF (Optimized Buffer Flush/Fill): 00 – запрещено; 01 – версия A; 10 – версия B. Всегда равно 0.
31:15	R	0x0	Reserved

Регистр характеристик линка 2 (PCIe_EP_i_link_cap_2)

Таблица 5.54 – Описание полей регистра PCIe_EP_i_link_cap_2

Биты	Тип доступа	Начальное значение	Описание
0	R	1'h0	Reserved
2:1	R	0x3	Supported Link Speeds Vector : показывает возможные скорости работы линка: 0001 – 2.5 Гбит/с; 0011 – 5 Гбит/с. Соответствует значению, установленному в системном контроллере. Доступен по APB шине только на чтение.
4:3	R	2'h0	Reserved
8:5	R	0x0	Reserved
10:9	R	0x0	Lower SKP OS Generation Supported Speeds Vector : если не равно 0, показывает, что на указанных частотах поддерживается SRIS и также поддерживается программное управление частотой генерации SKP OS. Доступен по APB шине только на чтение.
15:11	R	0x0	Reserved
17:16	R	0x0	Lower SKP OS Reception Supported Speeds Vector : если не равно 0, показывает, что на указанных частотах поддерживается SRIS и также поддерживается программное управление частотой генерации SKP OS. Доступен по APB шине только на чтение.
19:18	R	2'h0	Reserved
22:20	R	0x0	Reserved
23	R	0x0	Reserved
31:24	R	0x0	Reserved

Регистр управления и состояния линка 2 (PCIe_EP_i_link_ctrl_status_2)

Таблица 5.55– Описание полей регистра PCIe_EP_i_link_ctrl_status_2

Биты	Тип доступа	Начальное значение	Описание
3:0	R/W	4'd2	Target Link Speed : для Endpoint показывает максимальную скорость передачи при переконфигурации. Также показывает скорость передачи при переходе режим Compliance, вызванный установкой бита Enter Compliance. Значение после сброса зависит от настройки в системном контроллере: 0001 – 2.5 Гбит/с; 0010 – 5 Гбит/с. Доступен по APB шине на запись.
4	R/W	0x0	Enter Compliance : в Endpoint режиме установка в 1 включает Compliance режим. При этом также должен быть инициирован горячий сброс (Hot Reset). Доступен по APB шине на запись.
5	R/W	0x0	Hardware Autonomous Speed Disable : установка это бита запрещает аппаратное повышение скорости передачи. Доступен по APB шине на запись.
6	R	0x0	Selectable Deemphasis : не используется в режиме Endpoint.
9:7	R/W	0x0	Transmit Margin : используется только при отладке и в Compliance режиме. Управляет уровнем напряжения на выходе передатчика. 000 – нормальный уровень 001 – 800-1200 мВ для полного размаха и 400-700 мВ для половины размаха. 010-111 – описание значений в спецификации PCI Express Base Specification 2.0. Значение сбрасывается в 0 при переходе автомата обучения линка (LTSSM) в состояние Polling.Configuration. Доступен по APB шине на запись.
10	R/W	0x0	Enter Modified Compliance : используется только при отладке и в Compliance режиме. При установке этого бита в режиме Compliance будут передаваться Modified Compliance Pattern. При этом также должен быть установлен бит Enter Compliance и хост должен инициировать горячий сброс (Hot Reset). Доступен по APB шине на запись.
11	R/W	0x0	Compliance SOS : при установке это бита между Compliance Patterns будут передаваться SKP контрольные последовательности. Доступен по APB шине на запись.
15:12	R/W	0x0	Compliance De-Emphasis : устанавливает значение De-Emphasis в режиме Compliance 0 – -6 дБ 1 – -3.5 дБ остальные значения зарезервированы Используется только при работе на скорости 5 Гбит/с (Gen 2). Доступен по APB шине на запись.
16	R	0x0	Current De- Emphasis Level : показывает текущий уровень De-Emphasis передатчика. 0 – -6 дБ 1 – -3.5 дБ Используется только при работе на скорости 5 Гбит/с (Gen 2). Доступен по APB шине только на чтение.
21:17	R	0x0	Reserved
31:22	R	0x0	Reserved

Регистр основных характеристик расширенного обработчика ошибок (PCIe_EP_i_AER_enhanced_cap_hdr)

Таблица 5.56 – Описание полей регистра PCIe_EP_i_AER_enhanced_cap_hdr

Биты	Тип доступа	Начальное значение	Описание
15:0	R	0x01	PCI Express Extended Capability ID : показывает, что это структура характеристик расширенного обработчика ошибок (AER). Доступен по APB шине только на чтение.
19:16	R	4'h2	Capability Version : показывает версию структуры характеристик. Доступен по APB шине на запись.
31:20	R	12'h0	Next Capability Offset : содержит смещение адреса следующей структуры характеристик относительно текущей. Доступен по APB шине на запись.

Регистр статуса неисправимых ошибок (PCIe_EP_i_uncorr_err_status)

Таблица 5.57– Описание полей регистра PCIe_EP_i_uncorr_err_status

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x0	Reserved
4	R/WOCLR	0x0	Data Link Protocol Error Status : устанавливается, если идентификатор подтверждения (DLLP Ack/Nak) не равен идентификатору соответствующего сообщения (TLP). Сброс при записи 1 по шине APB.
11:5	R	0x0	Reserved
12	R/WOCLR	0x0	Poisoned TLP Status : устанавливается при получении TLP с установленным битом ошибки (poisoned bit). Ошибка считается не фатальной. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
13	R/WOCLR	0x0	Flow Control Protocol Error Status : устанавливается при ошибке последовательности управления. Сброс при записи 1 по шине APB.
14	R/WOCLR	0x0	Completion Timeout Status : устанавливается, если ответ на внешний запрос не был получен за необходимый период времени. Ошибка считается не фатальной. Сброс при записи 1 по шине APB.
15	R/WOCLR	0x0	Completer Abort Status : устанавливается, если контроллер вернул сообщение Completer Abort в ответ на запрос, пришедший по линку. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
16	R/WOCLR	0x0	Unexpected Completion Status : устанавливается при получении не ожидаемого сообщения завершения (unexpected Completion). Сброс при записи 1 по шине APB.
17	R/WOCLR	0x0	Receiver Overflow Status : устанавливается при получении пакета уровня транзакций (TLP) при заполненном приемном буфере. Сброс при записи 1 по шине APB.
18	R/WOCLR	0x0	Malformed TLP Status : устанавливается при получении TLP с неправильной структурой. Ошибка считается фатальной. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
19	R/WOCLR	0x0	ECRC Error Status : устанавливается при обнаружении ошибки ECRC в принятом TLP. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
20	R/WOCLR	0x0	Unsupported Request Error Status : устанавливается при получении неподдерживаемого контроллером сообщения. Ошибка считается не фатальной. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
21	R	0x0	Reserved
22	R/WOCLR	0x0	Uncorrectable Internal Error Status : устанавливается при обнаружении внутренних неисправимых ошибок контроллера (ошибок честности/ECC внутренних шин или памятей). Ошибка считается фатальной. Сброс при записи 1 по шине APB.
31:23	R	0x0	Reserved

Регистр масок неисправимых ошибок (PCIe_EP_i_uncorr_err_mask)

Биты масок в этом регистре управляют отчетами о неисправимых ошибках. Для каждого типа ошибки из регистра PCIe_EP_i_uncorr_err_status есть своя маска. При установленной маске в случае обнаружения ошибки:

1. По шине PCIe не будет послано соответствующее сообщение хост-контроллеру (RP).
2. Заголовок TLP, содержащего ошибку, не будет сохранен в Header Log Register.
3. First Error Pointer в структуре характеристик AER и Control Register не будут обновлены.

Таблица 5.58– Описание полей регистра PCIe_EP_i_uncorr_err_ mask

Биты	**Тип доступа**	**Начальное значение**	**Описание**
3:0	R	0x0	Reserved
4	R/W	0x0	Data Link Protocol Error Mask : маска ошибок Data Link Protocol. Доступен по APB шине на запись.
11:5	R	0x0	Reserved
12	R/W	0x0	Poisoned TLP Mask : маска ошибок Poisoned TLP. Доступен по APB шине на запись.
13	R/W	0x0	Flow Control Protocol Error Mask : маска ошибок последовательности управления. Доступен по APB шине на запись.
14	R/W	0x0	Completion Timeout Mask : маска ошибок времени ожидания завершение транзакции. Доступен по APB шине на запись.
15	R/W	0x0	Completer Abort Mask : маска ошибок прерывания завершения полученной транзакции. Доступен по APB шине на запись.
16	R/W	0x0	Unexpected Completion Mask : маска ошибок получения не ожидаемых завершений. Доступен по APB шине на запись.
17	R/W	0x0	Receiver Overflow Mask : маска ошибок переполнения приемника. Доступен по APB шине на запись.
18	R/W	0x0	Malformed TLP Mask : маска ошибок структурно неправильных TLP. Доступен по APB шине на запись.
19	R/W	0x0	ECRC Error Mask : маска ошибок ECRC. Доступен по APB шине на запись.
20	R/W	0x0	Unsupported Request Error Mask : маска ошибок неподдерживаемых запросов. Доступен по APB шине на запись.
21	R	0x0	Reserved
22	R/W	0x1	Uncorrectable Internal Error Mask : маска внутренних ошибок контроллера. Доступен по APB шине на запись.
31:23	R	0x0	Reserved

Регистр степени тяжести неисправимых ошибок (PCIe_EP_i_uncorr_err_severity)

Таблица 5.59– Описание полей регистра PCIe_EP_i_uncorr_err_severity

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x0	Reserved
4	R/W	0x01	Data Link Protocol Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
5	R	0x1	Surprise Down Error Severity : всегда равно 1
11:6	R	0x0	Reserved
12	R/W	0x0	Poisoned TLP Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
13	R/W	0x01	Flow Control Protocol Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
14	R/W	0x0	Completion Timeout Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
15	R/W	0x0	Completer Abort Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
16	R/W	0x0	Unexpected Completion Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
17	R/W	0x01	Receiver Overflow Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
18	R/W	0x01	Malformed TLP Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
19	R/W	0x0	ECRC Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
20	R/W	0x0	Unsupported Requeset Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
21	R	0x0	Reserved
22	R/W	0x01	Uncorrectable Internal Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
31:23	R	0x0	Reserved

Регистр статуса исправимых ошибок (PCIe_EP_i_corr_err_status)

Таблица 5.60– Описание полей регистра PCIe_EP_i_corr_err_status

Биты	Тип доступа	Начальное значение	Описание
0	R/WOCLR	0x0	Receiver Error Status : устанавливается при обнаружении на физическом уровне одной из следующих ошибок: - 8B10B кодировки; - четности; - переполнении эластичного буфера. Сброс при записи 1 по шине APB.
5:1	R	0x0	Reserved
6	R/WOCLR	0x0	Bad TP Status : устанавливается при обнаружении пакетным уровнем в TLP одной из следующих ошибок: - LCRC; - окончание пакета символом EDB и инверсия расчетного LCRC не равна полученному LCRC. Сброс при записи 1 по шине APB.
7	R/WOCLR	0x0	Bad DLLP Status : устанавливается при обнаружении пакетным уровнем в DLLP ошибки LCRC при отсутствии ошибок физического уровня. Сброс при записи 1 по шине APB.
8	R/WOCLR	0x0	Replay Number Rollover Status : устанавливается после трех повторных посылок одного пакета. Сброс при записи 1 по шине APB.
11:9	R	0x0	Reserved
12	R/WOCLR	0x0	Replay Timer Timeout Status : устанавливается при переполнении таймера пакетного уровня, что вызывает повторную посылку пакета TLP. Сброс при записи 1 по шине APB.
13	R/WOCLR	0x0	Advisory Non-Fatal Error Status : устанавливается при появлении неисправимой ошибки (Ucorrectable error). Подробное описание в главе 6.2.3.2.4 спецификации PCI Express 2.0. Сброс при записи 1 по шине APB.
14	R/WOCLR	0x0	Corrected Internal Error Status : устанавливается при обнаружении исправимой внутренней ошибки (исправимой ECC ошибки при чтении из памятей). Сброс при записи 1 по шине APB.
15	R/WOCLR	0x0	Header Log Overflow Status : устанавливается при переполнении Header Log Register. Сброс при записи 1 по шине APB.
31:16	R	0x0	Reserved

Регистр масок исправимых ошибок (PCIe_EP_i_corr_err_mask)

Биты масок в этом регистре управляют отчетами об исправимых ошибках. Для каждого типа ошибки из регистра PCIe_EP_i_corr_err_status есть своя маска. При установленной маске в случае обнаружения ошибки по шине PCIe не будет послано соответствующее сообщение хост-контроллеру (RP).

Таблица 5.61– Описание полей регистра PCIe_EP_i_corr_err_ mask

Биты	**Тип доступа**	**Начальное значение**	**Описание**
0	R/WOCLR	0x0	Receiver Error Mask : маска ошибок физического уровня. Сброс при записи 1 по шине APB.
5:1	R	0x0	Reserved
6	R/WOCLR	0x0	Bad TP Mask : маска ошибок Bad TP. Сброс при записи 1 по шине APB.
7	R/WOCLR	0x0	Bad DLLP Mask : маска ошибок Bad DLLP. Сброс при записи 1 по шине APB.
8	R/WOCLR	0x0	Replay Number Rollover Mask : маска ошибок количества повторных отправлений TLP. Сброс при записи 1 по шине APB.
11:9	R	0x0	Reserved
12	R/WOCLR	0x0	Replay Timer Timeout Mask : маска ошибок переполнения таймера повторной отправки сообщений. Сброс при записи 1 по шине APB.
13	R/WOCLR	0x0	Advisory Non-Fatal Error Mask : маска ошибок Advisory Non-Fatal. Сброс при записи 1 по шине APB.
14	R/WOCLR	0x0	Corrected Internal Error Mask : маска внутренних исправимых ошибок. Сброс при записи 1 по шине APB.
15	R/WOCLR	0x0	Header Log Overflow Mask : маска ошибки переполнения Header Log Register. Сброс при записи 1 по шине APB.
31:16	R	0x0	Reserved

Регистр управления и характеристик расширенного обработчика ошибок (PCIe_EP_i_advcd_err_cap_ctrl)

Таблица 5.62 – Описание полей регистра PCIe_EP_i_advcd_err_cap_ctrl

Биты	Тип доступа	Начальное значение	Описание
4:0	R	0x0	First Error Pointer : указывает на ошибку из Uncorrectable Error Status Register, которая была обнаружена первой. Значение удерживается до тех пор, пока не будет сброшено соответствующее поле регистра Uncorrectable Error Status Register. После сброса бита статуса текущей ошибки, следующий из установленных битов ошибок обновит указатель до соответствующего значения. Доступен по APB шине только на чтение.
5	R	0x1	ECRC Generation Capability : показывает, что контроллер может генерировать ECRC для посылаемых пакетов. Доступен по APB шине на запись.
6	R/W	0x0	Enable ECRC Generation : разрешает контроллеру генерацию ECRC для посылаемых пакетов. Доступен по APB шине на запись.
7	R	0x1	ECRC Check Capability : показывает, что контроллер может проверять ECRC в принимаемых пакетах. Доступен по APB шине на запись.
8	R/W	0x0	Enable ECRC Check : разрешает контроллеру проверку ERCR в принимаемых пакетах. Доступен по APB шине на запись.
9	R	0x0	Multiple Header Recording Capable : устанавливается, если функция может сохранить более одного заголовка в регистрах Header Log Registers. Всегда равно 0.
10	R	0x0	Multiple Header Recording Enable : установка разрешает сохранять множество заголовков в регистрах Header Log Registers. Всегда равно 0.
11	R	0x0	Reserved
31:12	R	0x0	Reserved

Регистр заголовка пакета 0 (PCIe_EP_i_hdr_log_0)

Это первый из четырех регистров, использующихся для захвата заголовка полученного контроллером TLP пакета в случае возникновения неисправимой ошибки. Если в регистре Uncorrectable Error Status Register установлено несколько бит, то здесь отражается заголовок только первой полученной транзакции. Какая ошибка ей соответствует можно определить на основании поля First Error Pointer. Регистр не может быть перезаписан, пока статусный бит соответствующей ошибки не будет сброшен.

Первое двойное слово заголовка TLP сохраняется в этом регистре с обратным порядком байт. Байт, содержащий поля Type/Format заголовка, сохраняется в [31:24] битах регистра Header Log Register 0.

Таблица 5.63 – Описание полей регистра PCIe_EP_i_hdr_log_0

Биты	**Тип доступа**	**Начальное значение**	**Описание**
31:0	R	0x0	Header DWORD 0 : содержит первое слово заголовка принятого TLP. Доступен по APB шине только на чтение.

Регистр заголовка пакета 1 (PCIe_EP_i_hdr_log_1)

Второе двойное слово заголовка TLP сохраняется в этом регистре с обратным порядком байт.

Таблица 5.64 – Описание полей регистра PCIe_EP_i_hdr_log_1

Биты	**Тип доступа**	**Начальное значение**	**Описание**
31:0	R	0x0	Header DWORD 1 : содержит второе слово заголовка принятого TLP. Доступен по APB шине только на чтение.

Регистр заголовка пакета 2 (PCIe_EP_i_hdr_log_2)

Третье двойное слово заголовка TLP сохраняется в этом регистре с обратным порядком байт.

Таблица 5.65 – Описание полей регистра PCIe_EP_i_hdr_log_2

Биты	**Тип доступа**	**Начальное значение**	**Описание**
31:0	R	0x0	Header DWORD 2 : содержит третье слово заголовка принятого TLP. Доступен по APB шине только на чтение.

Регистр заголовка пакета 3 (PCIe_EP_i_hdr_log_3)

Если принятый TLP имеет заголовок длиной 4 DWORDs, то этот регистр содержит четвертое двойное слово с обратным порядком байт.

Таблица 5.66 – Описание полей регистра PCIe_EP_i_hdr_log_3

Биты	**Тип доступа**	**Начальное значение**	**Описание**
31:0	R	0x0	Header DWORD 3 : содержит четвертое слово заголовка принятого TLP. Доступен по APB шине только на чтение.

5.12.2.2 Описание программно доступных стандартных конфигурационных регистров (режим Root Port)

В этом разделе при описании программно доступных регистров поле «Тип доступа» в таблицах относится к конфигурационным обращениям по шине PCIe. Возможности программного доступа по шине APB приведены в поле «Описание».

Регистр идентификации изготовителя и устройства (PCIe_RP_i_vendor_id_device_id)

Таблица 5.67 - Описание полей регистра PCIe_RP_i_vendor_id_device_id

Биты	Тип доступа	Начальное значение	Описание
15:0	R	16'h17cd	Vendor ID : ID разработчика устройства, присвоенное ему PCI SIG. Значение настраивается программно в регистре PCIe_LocMgmt_i_vendor_id_reg.
31:16	R	16'h100	Device ID : ID устройства, присвоенное ему разработчиком микросхемы. Доступен по APB шине на запись.

Регистр управления и состояния (PCIe_RP_i_command_status)

Таблица 5.68 - Описание полей регистра PCIe_EP_i_command_status

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	IO-Space Enable : разрешение передачи сообщений типа IO через контроллер. Доступен по APB шине на запись.
1	R/W	0x0	Mem-Space Enable : разрешение передачи сообщений типа Memory через контроллер. Доступен по APB шине на запись.
2	R/W	0x0	Bus-Master Enable : разрешение приема сообщений типа IO и Memory от контроллера. Доступен по APB шине на запись.
5:3	R	0x0	Reserved
6	R/W	0x0	Parity Error Response Enable : 1 – установка бита Master Data Parity Error разрешена; 0 – установка Master Data Parity Error запрещена. Доступен по APB шине на запись.
7	R	0x0	Reserved
8	R/W	0x0	SERR Enable : 1 – разрешает отправку сообщений о фатальных и не фатальных ошибках хост-контроллеру; 0 – отправка запрещена. Доступен по APB шине на запись.
9	R	0x0	Reserved
10	R/W	0x0	INTx Message Disabled : 1 – INTx (Legacy) прерывания запрещены; 0 – INTx (Legacy) прерывания разрешены. Доступен по APB шине на запись.
15:11	R	0x0	Reserved
18:16	R	0x0	Reserved
19	R	0x0	Interrupt Status : устанавливается, только если прерывания INTx (Legacy) разрешены. Устанавливается, если сообщение Assert_INTx было оправлено, а сообщение Deassert_INTx не было отправлено.
20	R	0x1	Capabilities List : указывает на наличие расширенного списка регистров (PCI Extended Capabilities registers). Всегда равно 1.
23:21	R	0x0	Reserved
24	R/WOCLR	0x0	Master Data Parity Error : устанавливается при получении завершения с ошибкой (poisoned completion) или отправлении записи с ошибкой (poisoned write request). Этот бит может быть сброшен записью 1 по шине APB.
26:25	R	0x0	Reserved
27	R/WOCLR	0x0	Signaled Target Abort : устанавливается, если контроллер послал сообщение завершения со статусом Completer Abort. Этот бит может быть сброшен записью 1 по шине APB.
28	R/WOCLR	0x0	Received Target Abort : устанавливается, если контроллер принял сообщение завершения со статусом Completer Abort. Этот бит может быть сброшен записью 1 по шине APB.
29	R/WOCLR	0x0	Received Master Abort : устанавливается, если контроллер принял сообщение завершения со статусом Unsupported Request. Этот бит может быть сброшен записью 1 по шине APB.
30	R/WOCLR	0x0	Signaled System Error : устанавливается при отправлении контроллером хост-контроллеру сообщения о фатальной или не фатальной ошибке. Этот бит может быть сброшен записью 1 по шине APB.
31	R/WOCLR	0x0	Detected Parity Error : устанавливается при получении контроллером сообщения TLP с ошибкой. Этот бит может быть сброшен записью 1 по шине APB.

Регистр номера версии и текущей функции (PCIe_RP_i_revision_id_class_code)

Таблица 5.69 - Описание полей регистра PCIe_RP_i_revision_id_class_code

Биты	Тип доступа	Начальное значение	Описание
7:0	R	8'h0	Revision ID : указывает версию микросхемы, устанавливается разработчиком микросхемы. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
15:8	R	8'h0	Programming Interface Byte : показывает набор регистров устройства. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
23:16	R	8'h0	Sub-Class Code : содержит подкатегорию текущей функции. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
31:24	R	8'h0	Class Code : содержит номер текущей функции. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.

Регистр управления встроенным тестированием, формата заголовка, таймера задержек и размера страниц кэш (PCIe_RP_i_bist_header_latency_cache_line)

Таблица 5.70 - Описание полей регистра PCIe_RP_i_bist_header_latency_cache_line

Биты	Тип доступа	Начальное значение	Описание
7:0	R/W	8'h0	Cache Line Size : см. PCI Specifications 3.0. Регистр доступен по PCIe и APB шине на чтение и запись, но его значение не используется.
15:8	R	8'h0	Latency Timer : не используется, всегда равно 0.
22:16	R	7'h1	Header Type : формат заголовка, всегда равно 1.
23	R	1'h0	Device Type : всегда равно 0.
31:24	R	8'h0	BIST Register : регистр управления BIST. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.

Регистр базового адреса 0 в конфигурационном пространстве типа 1 (PCIe_RP_i_RC_BAR_0)

Данный регистр может быть настроен как базовый адрес для 32-битных транзакций или спарен с PCIe_RP_i_RC_BAR_1 для 64-битных транзакций.

Таблица 5.71 - Описание полей регистра PCIe_RP_i_RC_BAR_0

Биты	**Тип доступа**	**Начальное значение**	**Описание**
0	R	0x0	BAR Type : тип доступа к региону BAR 0 0 – memory 1 – IO Значение настраивается программно в регистре PCIe_LocMgmt_i_rc_BAR_config_reg .
1	R	0x0	Reserved
2	R	0x0	Size : для региона типа memory: 0 – 32 битная адресация, 1 – 64 битная адресация. Для региона типа IO это Base Address [2]. Значение настраивается программно в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.
3	R	0x0	Prefetchability : для региона типа memory указывает на включение предвыборки данных. Для региона типа IO это Base Address [3]. Значение настраивается программно в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.
21:4	R	0x0	Base Address - RO part : определяет базовый адрес региона. Записываемое значение маскируется значением размера региона из регистра PCIe_LocMgmt_i_rc_BAR_config_reg. Доступен по APB шине на запись.
31:22	R/W	0x0	Base Address - RW part : определяет базовый адрес региона. Записываемое значение маскируется значением размера региона из регистра PCIe_LocMgmt_i_rc_BAR_config_reg. Доступен по APB шине на запись.

Регистр базового адреса 1 (PCIe_RP_i_RC_BAR_1)

Данный регистр может быть настроен как базовый адрес для 32-битных транзакций или спарен с PCIe_RP_i_RC_BAR_0 для 64-битных транзакций. Во втором случае он будет хранить 32 старших бита базового адреса. Выключен по умолчанию, но может быть включен настройкой регистра PCIe_LocMgmt_i_rc_BAR_config_reg.

Таблица 5.72 - Описание полей регистра PCIe_RP_i_RC_BAR_1

Биты	Тип доступа	Начальное значение	Описание
0	R	0x0	BAR Type : тип доступа к региону BAR 0: 0 – memory; 1 – IO. Значение может быть изменено в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.
1	R	0x0	Reserved
2	R	0x0	Size : для региона типа memory: 0 – 32 битная адресация; 1 – 64 битная адресация. Для региона типа IO это Base Address [2].
3	R	0x0	Prefetchability : для региона типа memory указывает на включение предвыборки данных. Для региона типа IO это Base Address [3].
21:4	R	0x0	Base Address - RO part : определяет базовый адрес региона. Записываемое значение маскируется значением размера региона из регистра PCIe_LocMgmt_i_rc_BAR_config_reg. Доступен по APB шине на запись.
31:22	R/W	0x0	Base Address - RW part : определяет базовый адрес региона. Записываемое значение маскируется значением размера региона из регистра PCIe_LocMgmt_i_rc_BAR_config_reg. Доступен по APB шине на запись.

Регистр номеров шин (PCIe_RP_i_pcie_bus_numbers)

Таблица 5.73 - Описание полей регистра PCIe_RP_i_pcie_bus_numbers

Биты	Тип доступа	Начальное значение	Описание
7:0	R/W	0x0	Primary Bus Number : это значение может быть программно записано и считано по шине APB, но контроллером оно не используется
15:8	R/W	0x0	Secondary Bus Number : это значение может быть программно записано и считано по шине APB, но контроллером оно не используется
23:16	R/W	0x0	Subordinate Bus Number : это значение может быть программно записано и считано по шине APB, но контроллером оно не используется
31:24	R	0x0	Secondary Latency Timer : Reserved

Регистр базового адреса ввода/вывода, ограничения для базового адреса ввода/вывода и дополнительного состояния (PCIe_RP_i_pcie_io_base_limit)

Таблица 5.74 - Описание полей регистра PCIe_RP_i_pcie_io_base_limit

Биты	Тип доступа	Начальное значение	Описание
0	R	0x0	Type1 cfg IO bar size : ширина конфигурационных регистров. Значение может быть программно установлено в регистре PCIe_LocMgmt_i_rc_BAR_config_reg в 20 бите. Всегда равно 0 при снятом 19 бите регистра PCIe_LocMgmt_i_rc_BAR_config_reg.
3:1	R	0x0	Reserved
7:4	R	0x0	IO Base Register : не используется контроллером. Доступен по APB шине на запись. Всегда равно 0, если IO BAR запрещены.
8	R	0x0	Type1 cfg IO bar size : всегда равно значению бита [0].
11:9	R	0x0	Reserved
15:12	R	0x0	IO Limit Register : не используется контроллером. Доступен по APB шине на запись. Всегда равно 0, если IO BAR запрещены.
23:16	R	0x0	Reserved
24	R/WOCLR	0x0	Master Data Parity Error : контроллер не устанавливает этот бит. Сброс при записи 1 по шине APB. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса. Всегда равно 0, если бит Parity Error Response Enable регистра PCIe_RP_i_intrpt_line_intrpt_pin равен 0.
26:25	R	0x0	Reserved
27	R/WOCLR	0x0	Signaled Target Abort : контроллер не устанавливает этот бит. Сброс при записи 1 по шине APB. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
28	R/WOCLR	0x0	Received Target Abort : контроллер не устанавливает этот бит. Сброс при записи 1 по шине APB. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
29	R/WOCLR	0x0	Received Master Abort : контроллер не устанавливает этот бит. Сброс при записи 1 по шине APB. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
30	R/WOCLR	0x0	Received System Error : контроллер не устанавливает этот бит. Сброс при записи 1 по шине APB. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
31	R/WOCLR	0x0	Detected Parity Error : контроллер не устанавливает этот бит. Сброс при записи 1 по шине APB. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.

Регистр базового и конечного адреса памяти (PCIe_RP_i_pcie_mem_base_limit)

Таблица 5.75 - Описание полей регистра PCIe_RP_i_pcie_mem_base_limit

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x0	Reserved
15:4	R/W	12'h0	Memory Base Register : не используется контроллером. Доступен по APB шине на запись.
19:16	R	0x0	Reserved
31:20	R/W	12'h0	Memory Limit Register : : не используется контроллером. Доступен по APB шине на запись.

Регистр предвыборки базового и конечного адреса памяти (PCIe_RP_i_pcie_prefetch_base_limit)

Таблица 5.76 - Описание полей регистра PCIe_RP_i_pcie_prefetch_base_limit

Биты	Тип доступа	Начальное значение	Описание
15:0	R	16'h0	Prefetchable Memory Base Register : : не используется контроллером. Доступен по APB шине на запись. Всегда равно 0, если не установлен бит Type1 cfg prefetchable mem bar enable в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.
31:16	R	16'h0	Prefetchable Memory Limit Register : не используется контроллером. Доступен по APB шине на запись. Всегда равно 0, если не установлен бит Type1 cfg prefetchable mem bar enable в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.

Регистр крайнего значения для предвыборки базового адреса памяти (PCIe_RP_i_pcie_prefetch_base_upper)

Таблица 5.77 - Описание полей регистра PCIe_RP_i_pcie_prefetch_base_upper

Биты	Тип доступа	Начальное значение	Описание
31:0	R	32'h0	Prefetchable Base Register Upper : не используется контроллером. Доступен по APB шине на запись. Всегда равно 0, если 64bit prefetchable memory транзакции не разрешены в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.

Регистр крайнего значения для предвыборки конечного адреса памяти (PCIe_RP_i_pcie_prefetch_limit_upper)

Таблица 5.78 - Описание полей регистра PCIe_RP_i_pcie_prefetch_limit_upper

Биты	Тип доступа	Начальное значение	Описание
31:0	R	32'h0	Prefetchable Base Register Upper : не используется контроллером. Доступен по APB шине на запись. Всегда равно 0, если 64bit prefetchable memory транзакции не разрешены в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.

Регистр крайнего значения для базового адреса ввода/вывода, крайнего значения для ограничения адреса ввода/вывода (PCIe_RP_i_pcie_io_base_limit_upper)

Таблица 5.79 - Описание полей регистра PCIe_RP_i_pcie_io_base_limit_upper

Биты	Тип доступа	Начальное значение	Описание
15:0	R	0x0	IO Base Register Upper : не используется контроллером. Доступен по APB шине на запись. Всегда равно 0, если 32bit IO транзакции не разрешены в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.
31:16	R	0x0	IO Limit Register Upper : не используется контроллером. Доступен по APB шине на запись. Всегда равно 0, если 32bit IO транзакции не разрешены в регистре PCIe_LocMgmt_i_rc_BAR_config_reg.

Регистр указателя характеристик (PCIe_RP_i_capabilities_pointer)

Таблица 5.80 - Описание полей регистра PCIe_RP_i_capabilities_pointer

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x80	Capabilities Pointer : содержит указатель на первую структуру характеристик (PCI Capability Structure). По умолчанию это структура управления питанием (Power Management Capability Structure). Доступен по APB шине на запись.
31:8	R	0x0	Reserved

Регистр линий и выводов прерываний (PCIe_RP_i_intrpt_line_intrpt_pin)

Таблица 5.81 - Описание полей регистра PCIe_RP_i_intrpt_line_intrpt_pin

Биты	Тип доступа	Начальное значение	Описание
7:0	R/W	8'hff	Interrupt Line Register : не используется контроллером в режиме RP. Доступен по APB шине на запись.
10:8	R	0x01	Interrupt Pin Register : показывает, каким будет передаваться Legacy прерывание, полученное данной функцией: 0x1 – INTA; 0x2 – INTB; 0x3 – INTC; 0x4 – INTD. Доступен по APB шине на запись.
15:11	R	0x0	Reserved
16	R/W	0x0	Parity Error Response Enable : разрешение установки бита Master Data Parity Error. Доступен по APB шине на запись.
17	R/W	0x0	Bridge Control SERR Enable : не используется в этом контроллере. Доступен по APB шине на запись.
18	R/W	0x0	ISA Enable : не используется в этом контроллере. Доступен по APB шине на запись.
19	R/W	0x0	VGA Enable : не используется в этом контроллере. Доступен по APB шине на запись.
20	R/W	0x0	VGA 16 DEcode : не используется в этом контроллере. Доступен по APB шине на запись.
21	R	0x0	Reserved
22	R/W	0x0	Bridge Control Register Secondary Bus Reset : запускает горячий сброс линка. Доступен по APB шине на запись.
31:23	R	0x0	Reserved

Регистр характеристик управления питанием (PCIe_RP_i_pwr_mgmt_cap)

В режиме RP содержание этого регистра не влияет на функционирование контроллера.

Таблица 5.82 - Описание полей регистра PCIe_RP_i_pwr_mgmt_cap

Биты	**Тип доступа**	**Начальное значение**	**Описание**
7:0	R	0x01	Capability ID : показывает, что это структура управления питанием. Доступен по APB шине на запись.
15:8	R	8'h90	Capabilities Pointer : содержит указатель на следующую структуру характеристик (PCI Capability Structure). Доступен по APB шине на запись.
18:16	R	0x3	Version ID : показывает версию системы управления питанием, поддерживаемую текущей функцией. Доступен по APB шине на запись.
19	R	0x0	PME Clock : всегда равен 0
20	R	0x0	Reserved
21	R	0x0	Device Specific Initialization Bit : всегда равен 0
24:22	R	0x0	Max Current Required from Aux Power Supply : всегда равен 0
25	R	0x1	D1 Support : показывает возможность поддержки PCI функцией состояния D1. Доступен по APB шине на запись.
26	R	0x0	D2 Support : показывает возможность поддержки PCI функцией состояния D2, всегда равен 0
27	R	0x01	PME Support for D0 State : показывает, может ли эта PCI функция посылать сообщения о событиях управления питанием (PME messages) находясь в состоянии D0. Доступен по APB шине на запись.
28	R	0x1	PME Support for D1 State : показывает, может ли эта PCI функция посылать сообщения о событиях управления питанием (PME messages) находясь в состоянии D1. Доступен по APB шине на запись.
29	R	0x0	PME Support for D2 State : всегда равен 0 (D2 состояние не поддерживается)
30	R	0x01	PME Support for D3(hot) Statue : показывает, может ли эта PCI функция посылать сообщения о событиях управления питанием (PME messages) находясь в состоянии D3(hot). Доступен по APB шине на запись.
31	R	0x1	PME Support for D3(cold) State : показывает, может ли эта PCI функция посылать сообщения о событиях управления питанием (PME messages) находясь в состоянии D3 (cold). Доступен по APB шине на запись.

Регистр управления питанием и его состояния (PCIe_RP_i_pwr_mgmt_ctrl_stat_rep)

Таблица 5.83 - Описание полей регистра PCIe_RP_i_pwr_mgmt_ctrl_stat_rep

Биты	Тип доступа	Начальное значение	Описание
1:0	R/W	0x0	Power State : доступен по APB шине на запись.
2	R	0x0	Reserved
3	R	0x01	No Soft Reset : доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
7:4	R	0x0	Reserved
8	R/W	0x0	PME Enable : доступен по APB шине на запись.
14:9	R	0x0	Reserved
15	R/WOCLR	0x0	PME Status : доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
23:16	R	0x0	Reserved
31:24	R	0x0	Data Register : всегда равен 0

Регистр управления сообщениями (PCIe_RP_i_msi_ctrl_reg)

Таблица 5.84 - Описание полей регистра PCIe_RP_i_msi_ctrl_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x05	Capability ID : показывает, что это структура управления MSI прерываниями. Всегда равно 0x05.
15:8	R	8'hc0	Capabilities Pointer : содержит указатель на следующую структуру характеристик (PCI Capability Structure). Доступен по APB шине на запись.
16	R/W	0x0	MSI Enable : разрешение MSI прерываний. Доступен по APB шине на запись.
19:17	R	0x0	Multiple Message Capable : количество различных MSI сообщений, которые может генерировать контроллер для этой функции (т.е. количество разных векторов прерываний) 000 = 1 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 Контроллер поддерживает всего до 32 различных прерываний, но установка этого поля должна быть основана на количестве прерываний, непосредственно используемых клиентом. Доступен по APB шине на запись.
22:20	R/W	0x0	Multiple Message Enable : количество различных MSI сообщений, которые генерирует контроллер для этой функции (т.е. количество разных векторов прерываний) 000 = 1 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 Установка этого поля должна быть основана на количестве прерываний, непосредственно используемых функцией. Доступен по APB шине на запись.
23	R	0x1	64-Bit Address Capable : установка в 1 показывает, что устройство поддерживает 64-разрядную адресацию для MSI сообщений. Доступен по APB шине на запись.
24	R	0x1	MSI masking capable : Доступен по APB шине на запись.
31:25	R	0x0	Reserved

Регистр младшей части адреса MSI сообщений (PCIe_RP_i_msi_msg_low_addr)

Таблица 5.85 - Описание полей регистра PCIe_RP_i_msi_msg_low_addr

Биты	Тип доступа	Начальное значение	Описание
1:0	R	0x0	Reserved
31:2	R/W	0x0	Message Address Low : младшая часть адреса MSI сообщений. Доступен по APB шине на запись.

Регистр старшей части адреса MSI сообщений (PCIe_RP_i_msi_msg_hi_addr)

Таблица 5.86 - Описание полей регистра PCIe_RP_i_msi_msg_hi_addr

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	0x0	Message Address High : старшая часть адреса MSI сообщений. Если равна нулю, то адрес 32-разрядный. Доступен по APB шине на запись.

Регистр данных MSI сообщений (PCIe_RP_i_msi_msg_data)

Этот регистр содержит данные, используемые в MSI сообщениях, генерируемых для соответствующей функции. Если количество различных векторов прерываний равно 1, то в качестве данных будет использоваться непосредственно указанное здесь значение. Если количество различных векторов больше 1, то LSB указанного значения заменяется на кодированный номер вектора.

Таблица 5.87 - Описание полей регистра PCIe_RP_i_msi_msg_data

Биты	Тип доступа	Начальное значение	Описание
15:0	R/W	0x0	Message Data : данные MSI сообщения. Доступен по APB шине на запись.
31:16	R	0x0	Reserved

Регистр масок пришедшего вектора MSI прерывания (PCIe_RP_i_msi_mask)

Таблица 5.88 - Описание полей регистра PCIe_RP_i_msi_mask

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	MSI Mask : маска битов пришедшего вектора MSI прерывания. Ширина этого параметра меняется в зависимости от Multiple Message Enable и, соответственно, количества векторов прерываний. Доступен по APB шине на запись.
31:1	R	0x0	Reserved Ширина этого параметра меняется в зависимости от Multiple Message Enable и, соответственно, количества векторов прерываний.

Регистр масок отправленного вектора MSI прерывания (PCIe_RP_i_msi_pending_bits)

Таблица 5.89 - Описание полей регистра PCIe_RP_i_msi_pending_bits

Биты	Тип доступа	Начальное значение	Описание
0	R	0x0	MSI Pending Bits : должен быть программно установлен при попытке отправки замаскированного прерывания. Ширина этого параметра меняется в зависимости от Multiple Message Enable. Доступен по APB шине на запись.
31:1	R	0x0	Reserved Ширина этого параметра меняется в зависимости от Multiple Message Enable и, соответственно, количества векторов прерываний.

Регистр списка структур характеристик (PCIe_RP_i_pcie_cap_list)

Таблица 5.90 - Описание полей регистра PCIe_RP_i_pcie_cap_list

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x010	Capability ID : показывает, что это структура характеристик Express расширения. Всегда равно 0x010.
15:8	R	0x0	Next Capability Pointer : содержит указатель на следующую структуру характеристик. Равенство нулю показывает, что эта структура последняя. Всегда равно 0.
19:16	R	0x2	Capability Version : показывает версию этой структуры характеристик. В данном случае она совместима с PCI Express Base Specification Revision 3.0. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
23:20	R	0x4	Device Type : показывает тип устройства: 0x0 – конечное устройство – End Point (EP); 0x4 – хост-контроллер – Root Port (RP). Всегда равно 4 в текущем режиме.
24	R	0x0	Slot Status : установлен в 1, если линк подключен к слоту. Всегда равно 0.
29:25	R	0x0	Interrupt Message Number : показывает вектор MSI или MSI-X прерывания в случае, если оно вызывается в соответствии с битами статуса в Slot Status Register, Root Status Register или текущей структуре характеристик. Значение устанавливается в зависимости от выбранного режима прерываний – MSI или MSI-X. Всегда равно 0.
30	R	0x0	TCS Routing Supported : 0 – устройство не поддерживает маршрутизацию доверенных конфигурационных запросов (Trusted Configuration Requests); 1 – опция поддерживается. Не используется в режиме конечного устройства. Всегда равно 0.
31	R	0x0	Reserved

Регистр характеристик устройства (PCIe_RP_i_pcie_cap)

Таблица 5.91 - Описание полей регистра PCIe_RP_i_pcie_cap

Биты	Тип доступа	Начальное значение	Описание
2:0	R	3'b011	Max Payload Size : показывает максимальное количество данных в сообщении. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
4:3	R	0x0	Phantom Functions Supported : расширение поля тэга за счет неиспользуемых битов. Не используется в этой микросхеме. Всегда равно 0.
5	R	0x0	Extended Tag Field Supported : расширенное поле tag не поддерживается. Всегда равно 0.
8:6	R	0x0	Acceptable L0S Latency : допустимая контроллером задержка перехода из L0s в L0 состояния. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
11:9	R	0x0	Acceptable L1 Latency : допустимая контроллером задержка перехода из L1 в L0 состояния. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
14:12	R	0x0	Reserved
15	R	1'b1	Role-Based Error Reporting : разрешение ролевых (role-based) отчетов об ошибках. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
17:16	R	0x0	Reserved
25:18	R	0x0	Captured Slot Power Limit Value : максимальное значение потребления, поддерживаемое слотом. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
27:26	R	0x0	Captured Power Limit Scale : масштаб значения поля Captured Slot Power Limit Value текущего регистра. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
28	R	0x0	Function Level Reset Capability : Всегда равно 0.
31:29	R	0x0	Reserved

Регистр управления и состояния устройства (PCIe_RP_i_pcie_dev_ctrl_status)

Таблица 5.91 – Описание полей регистра PCIe_RP_i_pcie_dev_ctrl_status

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	Enable Correctable Error Reporting : не используется в режиме Root Port.
1	R/W	0x0	Enable Non- Fatal Error Reporting : не используется в режиме Root Port.
2	R/W	0x0	Enable Fatal Error Reporting : не используется в режиме Root Port.
3	R/W	0x0	Enable Unsupported Request Reporting : разрешение посылки сообщений об обнаружении неподдерживаемого запроса. Доступен по APB шине на запись.
4	R/W	0x01	Enable Relaxed Ordering : разрешает установку бита Relaxed Ordering в передаваемых транзакциях. Доступен по APB шине на запись.
7:5	R/W	0x0	Max Payload Size : показывает максимальное количество данных в сообщении. Устройство должно быть в состоянии принимать пакеты с таким количеством данных и не должно посылать пакеты большего размера. Устанавливается программно на основании данных из Device Capabilities Register этого и ответного устройств. Доступен по APB шине на запись.
8	R	0x0	Extended Tag Field Enable : не используется в режиме Root Port. Всегда равно 0.
9	R	0x0	Enable Phantom Functions : всегда равно 0.
10	R	0x0	Enable Aux Power : всегда равно 0.
11	R/W	0x1	Enable No Snoop : установка в 1 разрешает установку бита No Snoop в передаваемых транзакциях, не требующих когерентности кэша. Доступен по APB шине на запись.
14:12	R/W	0x02	Max Read Request Size : показывает максимальное количество данных в запросах чтения, сгенерированных этим устройством. Доступен по APB шине на запись.
15	R	0x0	Reserved

Продолжение таблицы 5.91

Биты	Тип доступа	Начальное значение	Описание
16	R/WOCLR	0x0	Correctable Error Detected : устанавливается контроллером, когда он обнаруживает исправимую ошибку. Не зависит от разрешения соответствующего сообщения и маскирования. Сброс при записи 1 по шине APB.
17	R/WOCLR	0x0	Non-Fatal Error Detected : устанавливается контроллером, когда он обнаруживает не фатальную ошибку. Не зависит от разрешения соответствующего сообщения и маскирования. Сброс при записи 1 по шине APB.
18	R/WOCLR	0x0	Fatal Error Detected : устанавливается контроллером, когда он обнаруживает фатальную ошибку. Не зависит от разрешения соответствующего сообщения и маскирования. Сброс при записи 1 по шине APB.
19	R/WOCLR	0x0	Unsupported Request Detected : устанавливается контроллером, когда он обнаруживает неподдерживаемый запрос. Не зависит от разрешения соответствующего сообщения. Сброс при записи 1 по шине APB.
20	R	0x0	Aux Power Detected : не используется. Всегда равно 0.
21	R	0x0	Transaction Pending : устанавливается, если есть ожидающие запросы, требующие подтверждения (Non-Posted). Только для чтения по шине APB.
31:22	R	0x0	Reserved

Регистр характеристик линка (PCIe_RP_i_link_cap)

Таблица 5.93 – Описание полей регистра PCIe_RP_i_link_cap

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x2	Maximum Link Speed : показывает максимальную скорость линка. Соответствует значению, установленному в системном контроллере. Только для чтения по шине APB.
9:4	R	0x4	Maximum Link Width : показывает максимальное количество линий. Соответствует значению, установленному в системном контроллере. Только для чтения по шине APB.
11:10	R	0x3	Active State Power Management : показывает уровень поддерживаемого ASPM. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
14:12	R	0x2	L0S Exit Latency : показывает время, требуемое устройству для перехода из состояния L0S в L0. Значение зависит от реализации Phy. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
17:15	R	0x3	L1 Exit Latency : показывает время, требуемое устройству для выхода из состояния L1. Значение зависит от реализации Phy. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
18	R	0x0	Clock Power Management : показывает, что устройство поддерживает отключение опорного тактового сигнала. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
19	R	0x0	Surprise Down Error Reporting Capability : показывает способность устройства сообщать о ошибке внезапного выключения (Surprize Down error). Эта функция не поддерживается. Всегда равно 0.
20	R	0x0	Data Link Layer Active Reporting Capability : показывает способность устройства сообщать о достижении автоматом управления пакетным уровнем состояния DL_Active. Эта функция не поддерживается. Всегда равно 0.
21	R	0x1	Link Bandwidth Notification Capability : поддержка уведомлений о ширине линка и механизме прерываний. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
22	R	0x1	ASPM Optionality Compliance : показывает поддержку устройством дополнительных ASPM возможностей. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
23	R	0x0	Reserved
31:24	R	8'h0	Port Number : показывает номер порта, присвоенный PCIe линку, подключенному к устройству. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.

Регистр управления и состояния линка (PCIe_RP_i_link_ctrl_status)

Таблица 5.94 – Описание полей регистра PCIe_RP_i_link_ctrl_status

Биты	Тип доступа	Начальное значение	Описание
1:0	R/W	0x0	Active State Power Management Control : управление ASPM возможностями: 00 – ASPM запрещен; 01 – L0s состояние разрешено, L1 запрещено; 10 – L0s состояние запрещено, L1 разрешено; 11 – L0s и L1 состояния разрешены. Эти биты могут быть установлены только при установленных соответствующих битах Active State Power Management регистра PCIe_RP_i_link_cap. Доступен по APB шине на запись.
2	R	0x0	Reserved
3	R	0x0	Read Completion Boundary : показывает границы чтения хост-контроллера: 0 – 64 байта; 1 – 128 байт. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
4	R/W	0x0	Link Disable : установка этого бита переводит автомат обучения (LTSSM) в отключенный режим (Disable). LTSSM остается в таком состоянии, пока бит не будет снят. Доступен по APB шине на запись.
5	W	0x0	Retrain Link : установка этого бита вызывает переобучение линка. При чтении всегда возвращает 0. Доступен по APB шине на запись.
6	R/W	0x0	Common Clock Configuration : показывает отношения опорного тактового сигнала этого EP и ответного устройства: 0 – асинхронные; 1 – синхронные. Доступен по APB шине на запись.
7	R/W	0x0	Extended Synch : установка этого бита увеличивает частоту передачи управляющих пакетов (Ordered Sets) при переходе из L0s в L0 состояние. Доступен по APB шине на запись.
8	R	0x0	Enable Clock Power Management : не используется в режиме RP. Всегда равно 0.
9	R	0x0	Hardware Autonomous Width Disable : установленный, этот бит показывает программе, что она должна запрашивать изменение ширины линка только в случае некорректной работы текущего набора линий. Доступен по APB шине на запись.
10	R/W	0x0	Link Bandwidth Management Interrupt Enable : разрешает выдачу прерывания при установке бита Link Bandwidth Management Status. Всегда равно нулю, если Link Bandwidth Notification Capability равен нулю. Доступен по APB шине на запись.
11	R/W	0x0	Link Autonomous Bandwidth Interrupt Enable : разрешает выдачу прерывания при установке бита Link Autonomous Bandwidth Status. Всегда равно нулю, если Link Bandwidth Notification Capability равен нулю. Доступен по APB шине на запись.
15:12	R	0x0	Reserved

Продолжение таблицы 5.94

Биты	Тип доступа	Начальное значение	Описание
19:16	R	0x1	Negotiated Link Speed : согласованная скорость передачи данных: 0001 – 2.5 Гб/с на линию; 0010 – 5 Гб/с на линию. Доступен по APB шине только на чтение.
25:20	R	0x4	Negotiated Link Width : согласованное количество линий. Устанавливается по окончании обучения линка. Доступен по APB шине только на чтение.
26	R	0x0	Reserved
27	R	0x0	Link Training Status : устанавливается аппаратно, если LTSSM в режиме Configuration или Recovery или при начале переобучения линка. Доступен по APB шине только на чтение.
28	R	0x0	Slot Clock Configuration : показывает, что используется тактовый сигнал от разъёма. Если устройство использует независимый тактовый сигнал, несмотря на наличие такового на разъёме, этот бит должен быть сброшен в 0. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
29	R	0x0	Data Link Layer Active : показывает статус пакетного уровня. Всегда равно 0.
30	R/WOCLR	0x0	Link Bandwidth Management Status : устанавливается при аппаратном изменении ширины линка или скорости передачи, связанном с некорректной работой. Всегда равно 0, если Link Bandwidth Notification Capability равен 0. Сброс при записи 1 по шине APB.
31	R/WOCLR	0x0	Link Autonomous Bandwidth Status : устанавливается при аппаратном изменении ширины линка или скорости передачи, не связанном с некорректной работой. Всегда равно 0, если Link Bandwidth Notification Capability равен 0. Сброс при записи 1 по шине APB.

Регистр управления и состояния слота (PCIe_RP_i_slot_ctrl_status)

Таблица 5.95 – Описание полей регистра PCIe_RP_i_slot_ctrl_status

Биты	Тип доступа	Начальное значение	Описание
21:0	R	0x0	Reserved
22	R	1'b1	Presence Detect State : 0 – слот пуст 1 – PCIe карта присутствует Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
31:23	R	0x0	Reserved

Регистр управления и характеристик режима ROOT (PCIe_RP_i_root_ctrl_cap)

Таблица 5.96 – Описание полей регистра PCIe_RP_i_root_ctrl_cap

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	System Error on Correctable Error Enable : значение может быть записано и считано по шине APB, но оно не используется при работе контроллера.
1	R/W	0x0	System Error on Non-Fatal Error Enable : значение может быть записано и считано по шине APB, но оно не используется при работе контроллера.
2	R/W	0x0	PME Interrupt Enable : значение может быть записано и считано по шине APB, но оно не используется при работе контроллера.
3	R/W	0x0	CRS Software Visibility Enable : значение может быть записано и считано по шине APB, но оно не используется при работе контроллера.
31:4	R	0x0	Reserved

Регистр состояния режима ROOT (PCIe_RP_i_root_status)

Таблица 5.97 – Описание полей регистра PCIe_RP_i_root_status

Биты	Тип доступа	Начальное значение	Описание
15:0	R	0x0	PME Requester ID : показывает PCI Requester ID последнего полученного PME сообщения. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
16	R/WOCLR	0x0	PME Status : показывает, что PME был подтвержден устройством с Requester ID, указанным в PME Requester ID. Сброс при записи 1 по шине APB. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
17	R	0x0	PME Pending : устанавливается, если другой PME ожидает обработки при установленном PME Status. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
31:18	R	0x0	Reserved

Регистр характеристик устройства 2 (PCIe_RP_i_pcie_cap_2)

Таблица 5.98 - Описание полей регистра PCIe_RP_i_pcie_cap_2

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x02	Completion Timeout Ranges : показывает поддерживаемый устройством таймаут подтверждения сообщения. Непосредственно используемые значения таймаута расположены в регистрах локального управления. По умолчанию 10-250 мс. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
4	R	0x01	Completion Timeout Disable Supported : показывает возможность функции выключить таймаут подтверждения сообщения. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
5	R	0x0	ARI forwarding support : всегда равно 0.
6	R	0x1	Atomic OP routing supported : маршрутизация атомарных операций (Atomic OP) поддерживается. Доступен по APB шине на запись при дополнительно установленном [13] бите адреса.
7	R	0x0	32-Bit Atomic Op Completer Supported : всегда равно 0.
8	R	0x0	64-Bit Atomic Op Completer Supported : всегда равно 0.
9	R	0x0	128-Bit CAS Atomic Op Completer Supported : всегда равно 0.
10	R	0x0	Reserved
11	R	0x0	LTR Mechanism Supported : всегда равно 0.
13:12	R	0x0	TPH Completer Supported : всегда равно 0.
17:14	R	0x0	Reserved
19:18	R	0x0	OBFF Supported : показывает, что функция поддерживает OBFF. Всегда равно 0.
20	R	0x1	Extended Format Field Supported : показывает, что функция поддерживает 3-битный формат поля Fmt заголовка TLP. Всегда равно 1.
21	R	0x0	End-End TLP Prefix Supported : показывает, что функция поддерживает End-End TLP Prefix. Всегда равно 0.
23:22	R	0x0	Max End-End TLP Prefixes : максимальное количество End-End TLP Prefixes: 01 – 1 End-End TLP Prefix; 10 – 2 End-End TLP Prefix. Всегда равно 0.
31:24	R	0x0	Reserved

Регистр управления и состояния устройства 2 (PCIe_RP_i_pcie_dev_ctrl_status_2)

Таблица 5.99 – Описание полей регистра PCIe_RP_i_pcie_dev_ctrl_status_2

Биты	Тип доступа	Начальное значение	Описание
3:0	R/W	0x0	Completion Timeout Value : поддерживаемый устройством таймаут подтверждения сообщения: 0101 – диапазон 1; 0110 – диапазон 2. Соответствующие значения устанавливаются в регистрах Completion Timeout Interval Registers 0 и 1. Доступен по APB шине на запись.
4	R/W	0x0	Completion Timeout Disable : установка этого бита выключает таймаут подтверждения сообщений. Доступен по APB шине на запись.
5	R	0x0	ARI forwarding enable : показывает, что [7:0] биты ID используются как номер функции при преобразовании конфигурационных пакетов типа 1 в тип 0. Всегда равно 0.
6	R	0x0	Atomic Op Requester Enable : разрешение генерации атомарных операций. Всегда равно 0.
7	R	0x0	Reserved
8	R	0x0	IDO Request Enable : когда этот бит установлен, функции разрешено устанавливать бит ID-based Ordering (IDO) Attribute в посылаемых запросах. Всегда равно 0.
9	R	0x0	IDO Completion Enable : когда этот бит установлен, функции разрешено устанавливать бит ID-based Ordering (IDO) Attribute в посылаемых подтверждениях. Всегда равно 0.
10	R	0x0	LTR Mechanism Enable : всегда равно 0.
12:11	R	0x0	Reserved
14:13	R	0x0	OBFF Enable : разрешение OBFF (Optimized Buffer Flush/Fill): 00 – запрещено; 01 – версия A; 10 – версия B; Всегда равно 0.
31:15	R	0x0	Reserved

Регистр характеристик линка 2 (PCIe_RP_i_link_cap_2)

Таблица 5.100 – Описание полей регистра PCIe_RP_i_link_cap_2

Биты	Тип доступа	Начальное значение	Описание
0	R	1'h0	Reserved
2:1	R	0x3	Supported Link Speeds Vector : показывает возможные скорости работы линка: 0001 – 2.5 Гбит/с; 0011 – 5 Гбит/с. Соответствует значению, установленному в системном контроллере. Доступен по APB шине только на чтение.
4:3	R	2'h0	Reserved
8:5	R	0x0	Reserved
10:9	R	0x0	Lower SKP OS Generation Supported Speeds Vector : если не равно 0, показывает, что на указанных частотах поддерживается SRIS и также поддерживается программное управление частотой генерации SKP OS. Доступен по APB шине только на чтение.
15:11	R	0x0	Reserved
17:16	R	0x0	Lower SKP OS Reception Supported Speeds Vector : если не равно 0, показывает, что на указанных частотах поддерживается SRIS и также поддерживается программное управление частотой генерации SKP OS. Доступен по APB шине только на чтение.
19:18	R	2'h0	Reserved
22:20	R	0x0	Reserved
23	R	0x0	Reserved
31:24	R	0x0	Reserved

Регистр управления и состояния линка 2 (PCIe_RP_i_link_ctrl_status_2)

Таблица 5.101 – Описание полей регистра PCIe_RP_i_link_ctrl_status_2

Биты	Тип доступа	Начальное значение	Описание
3:0	R/W	4'd2	Target Link Speed : для Endpoint показывает максимальную скорость передачи при переконфигурации. Также показывает скорость передачи при переходе режим Compliance, вызванный установкой бита Enter Compliance. Значение после сброса зависит от настройки в системном контроллере: 0001 – 2.5 Гбит/с; 0010 – 5 Гбит/с. Доступен по APB шине на запись.
4	R/W	0x0	Enter Compliance : в Endpoint режиме установка в 1 включает Compliance режим. При этом также должен быть инициирован горячий сброс (Hot Reset). Доступен по APB шине на запись.
5	R/W	0x0	Hardware Autonomous Speed Disable : установка это бита запрещает аппаратное повышение скорости передачи. Доступен по APB шине на запись.
6	R/W	0x0	Selectable Deemphasis : не используется в режиме Endpoint.
9:7	R/W	0x0	Transmit Margin : используется только при отладке и в Compliance режиме. Управляет уровнем напряжения на выходе передатчика: 000 – нормальный уровень; 001 – 800-1200 мВ для полного размаха и 400-700 мВ для половины размаха; 010-111 – описание значений в спецификации PCI Express Base Specification 2.0. Значение сбрасывается в 0 при переходе автомата обучения линка (LTSSM) в состояние Polling.Configuration. Доступен по APB шине на запись.
10	R/W	0x0	Enter Modified Compliance : используется только при отладке и в Compliance режиме. При установке этого бита в режиме Compliance будут передаваться Modified Compliance Pattern. При этом также должен быть установлен бит Enter Compliance и хост должен инициировать горячий сброс (Hot Reset). Доступен по APB шине на запись.
11	R/W	0x0	Compliance SOS : при установке это бита между Compliance Patterns будут передаваться SKP контрольные последовательности. Доступен по APB шине на запись.
15:12	R/W	0x0	Compliance De-Emphasis : устанавливает значение De-Emphasis в режиме Compliance: 0 – -6 дБ; 1 – -3.5 дБ. остальные значения зарезервированы Используется только при работе на скорости 5 Гбит/с (Gen 2). Доступен по APB шине на запись.
16	R	0x1	Current De- Emphasis Level : показывает текущий уровень De-Emphasis передатчика: 0 – -6 дБ; 1 – -3.5 дБ. Используется только при работе на скорости 5 Гбит/с (Gen 2). Доступен по APB шине только на чтение.
21:17	R	0x0	Reserved
31:22	R	0x0	Reserved

Регистр основных характеристик расширенного обработчика ошибок (PCIe_RP_i_AER_enhncd_cap)

Таблица 5.102 – Описание полей регистра PCIe_RP_i_AER_enhncd_cap

Биты	Тип доступа	Начальное значение	Описание
15:0	R	0x01	PCI Express Extended Capability ID : показывает, что это структура характеристик расширенного обработчика ошибок (AER). Доступен по APB шине только на чтение.
19:16	R	4'h2	Capability Version : показывает версию структуры характеристик. Доступен по APB шине на запись.
31:20	R	12'h0	Next Capability Offset : содержит смещение адреса следующей структуры характеристик относительно текущей. Доступен по APB шине на запись.

Регистр статуса неисправимых ошибок (PCIe_RP_i_uncorr_err_status)

Таблица 5.103 – Описание полей регистра PCIe_RP_i_uncorr_err_status

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x0	Reserved
4	R/WOCLR	0x0	Data Link Protocol Error Status : устанавливается, если идентификатор подтверждения (DLLP Ack/Nak) не равен идентификатору соответствующего сообщения (TLP). Сброс при записи 1 по шине APB.
11:5	R	0x0	Reserved
12	R/WOCLR	0x0	Poisoned TLP Status : устанавливается при получении TLP с установленным битом ошибки (poisoned bit). Ошибка считается не фатальной. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
13	R/WOCLR	0x0	Flow Control Protocol Error Status : устанавливается при ошибке последовательности управления. Сброс при записи 1 по шине APB.
14	R/WOCLR	0x0	Completion Timeout Status : устанавливается, если ответ на внешний запрос не был получен за необходимый период времени. Ошибка считается не фатальной. Сброс при записи 1 по шине APB.
15	R/WOCLR	0x0	Completer Abort Status : устанавливается, если контроллер вернул сообщение Completer Abort в ответ на запрос, пришедший по линку. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
16	R/WOCLR	0x0	Unexpected Completion Status : устанавливается при получении не ожидаемого сообщения завершения (unexpected Completion). Сброс при записи 1 по шине APB.
17	R/WOCLR	0x0	Receiver Overflow Status : устанавливается при получении пакета уровня транзакций (TLP) при заполненном приемном буфере. Сброс при записи 1 по шине APB.
18	R/WOCLR	0x0	Malformed TLP Status : устанавливается при получении TLP с неправильной структурой. Ошибка считается фатальной. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
19	R/WOCLR	0x0	ECRC Error Status : устанавливается при обнаружении ошибки ECRC в принятом TLP. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
20	R/WOCLR	0x0	Unsupported Request Error Status : устанавливается при получении неподдерживаемого контроллером сообщения. Ошибка считается не фатальной. Заголовок полученного сообщения сохраняется в Header Log Register. Сброс при записи 1 по шине APB.
21	R	0x0	Reserved
22	R/WOCLR	0x0	Uncorrectable Internal Error Status : устанавливается при обнаружении внутренних неисправимых ошибок контроллера (ошибок честности/ECC внутренних шин или памятей). Ошибка считается фатальной. Сброс при записи 1 по шине APB.
31:23	R	0x0	Reserved

Регистр масок неисправимых ошибок (PCIe_RP_i_uncorr_err_mask)

Биты масок в этом регистре управляют отчетами о неисправимых ошибках. Для каждого типа ошибки из регистра PCIe_RP_i_uncorr_err_status есть своя маска. При установленной маске в случае обнаружения ошибки:

- Выдача ошибок системе TODO.
- Заголовок TLP, содержащего ошибку, не будет сохранен в Header Log Register.
- First Error Pointer в структуре характеристик AER и Control Register не будут обновлены.

Таблица 5.104 – Описание полей регистра PCIe_RP_i_uncorr_err_mask

Биты	**Тип доступа**	**Начальное значение**	**Описание**
3:0	R	0x0	Reserved
4	R/W	0x0	Data Link Protocol Error Mask : маска ошибок Data Link Protocol. Доступен по APB шине на запись.
11:5	R	0x0	Reserved
12	R/W	0x0	Poisoned TLP Mask : маска ошибок Poisoned TLP. Доступен по APB шине на запись.
13	R/W	0x0	Flow Control Protocol Error Mask : маска ошибок последовательности управления. Доступен по APB шине на запись.
14	R/W	0x0	Completion Timeout Mask : маска ошибок времени ожидания завершение транзакции. Доступен по APB шине на запись.
15	R/W	0x0	Completer Abort Mask : маска ошибок прерывания завершения полученной транзакции. Доступен по APB шине на запись.
16	R/W	0x0	Unexpected Completion Mask : маска ошибок получения не ожидаемых завершений. Доступен по APB шине на запись.
17	R/W	0x0	Receiver Overflow Mask : маска ошибок переполнения приемника. Доступен по APB шине на запись.
18	R/W	0x0	Malformed TLP Mask : маска ошибок структурно неправильных TLP. Доступен по APB шине на запись.
19	R/W	0x0	ECRC Error Mask : маска ошибок ECRC. Доступен по APB шине на запись.
20	R/W	0x0	Unsupported Request Error Mask : маска ошибок неподдерживаемых запросов. Доступен по APB шине на запись.
21	R	0x0	Reserved
22	R/W	0x1	Uncorrectable Internal Error Mask : маска внутренних ошибок контроллера. Доступен по APB шине на запись.
31:23	R	0x0	Reserved

Регистр степени тяжести неисправимых ошибок (PCIe_RP_i_uncorr_err_severity)

Таблица 5.105 – Описание полей регистра PCIe_RP_i_uncorr_err_severity

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x0	Reserved
4	R/W	0x01	Data Link Protocol Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
5	R	0x1	Surprise Down Error Severity : всегда равно 1
11:6	R	0x0	Reserved
12	R/W	0x0	Poisoned TLP Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
13	R/W	0x01	Flow Control Protocol Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
14	R/W	0x0	Completion Timeout Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
15	R/W	0x0	Completer Abort Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
16	R/W	0x0	Unexpected Completion Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
17	R/W	0x01	Receiver Overflow Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
18	R/W	0x01	Malformed TLP Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
19	R/W	0x0	ECRC Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
20	R/W	0x0	Unsupported Requeset Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
21	R	0x0	Reserved
22	R/W	0x01	Uncorrectable Internal Error Severity : ошибка обрабатывается как: 0 – не фатальная; 1 – фатальная. Доступен по APB шине на запись.
31:23	R	0x0	Reserved

Регистр статуса исправимых ошибок (PCIe_RP_i_corr_err_status)

Таблица 5.106 – Описание полей регистра PCIe_RP_i_corr_err_status

Биты	Тип доступа	Начальное значение	Описание
0	R/WOCLR	0x0	Receiver Error Status : устанавливается при обнаружении на физическом уровне одной из следующих ошибок: - 8B10B кодировки; - четности; - переполнении эластичного буфера. Сброс при записи 1 по шине APB.
5:1	R	0x0	Reserved
6	R/WOCLR	0x0	Bad TP Status : устанавливается при обнаружении пакетным уровнем в TLP одной из следующих ошибок: - LCRC; - окончание пакета символом EDB и инверсия расчетного LCRC не равна полученному LCRC. Сброс при записи 1 по шине APB.
7	R/WOCLR	0x0	Bad DLLP Status : устанавливается при обнаружении пакетным уровнем в DLLP ошибки LCRC при отсутствии ошибок физического уровня. Сброс при записи 1 по шине APB.
8	R/WOCLR	0x0	Replay Number Rollover Status : устанавливается после трех повторных посылок одного пакета. Сброс при записи 1 по шине APB.
11:9	R	0x0	Reserved
12	R/WOCLR	0x0	Replay Timer Timeout Status : устанавливается при переполнении таймера пакетного уровня, что вызывает повторную посылку пакета TLP. Сброс при записи 1 по шине APB.
13	R/WOCLR	0x0	Advisory Non-Fatal Error Status : устанавливается при появлении неисправимой ошибки (Ucorrectable error). Подробное описание в главе 6.2.3.2.4 спецификации PCI Express 2.0. Сброс при записи 1 по шине APB.
14	R/WOCLR	0x0	Corrected Internal Error Status : устанавливается при обнаружении исправимой внутренней ошибки (исправимой ECC ошибки при чтении из памятей). Сброс при записи 1 по шине APB.
15	R/WOCLR	0x0	Header Log Overflow Status : устанавливается при переполнении Header Log Register. Сброс при записи 1 по шине APB.
31:16	R	0x0	Reserved

Регистр масок исправимых ошибок (PCIe_RP_i_corr_err_mask)

Биты масок в этом регистре управляют отчетами об исправимых ошибках. Для каждого типа ошибки из регистра PCIe_RP_i_corr_err_status есть своя маска. При установленной маске в случае обнаружения ошибки выдача ошибок системе TODO.

Таблица 5.107 – Описание полей регистра PCIe_RP_i_corr_err_mask

Биты	**Тип доступа**	**Начальное значение**	**Описание**
0	R/WOCLR	0x0	Receiver Error Mask : маска ошибок физического уровня. Сброс при записи 1 по шине APB.
5:1	R	0x0	Reserved
6	R/WOCLR	0x0	Bad TP Mask : маска ошибок Bad TP. Сброс при записи 1 по шине APB.
7	R/WOCLR	0x0	Bad DLLP Mask : маска ошибок Bad DLLP. Сброс при записи 1 по шине APB.
8	R/WOCLR	0x0	Replay Number Rollover Mask : маска ошибок количества повторных отправлений TLP. Сброс при записи 1 по шине APB.
11:9	R	0x0	Reserved
12	R/WOCLR	0x0	Replay Timer Timeout Mask : маска ошибок переполнения таймера повторной отправки сообщений. Сброс при записи 1 по шине APB.
13	R/WOCLR	0x0	Advisory Non-Fatal Error Mask : маска ошибок Advisory Non-Fatal. Сброс при записи 1 по шине APB.
14	R/WOCLR	0x0	Corrected Internal Error Mask : маска внутренних исправимых ошибок. Сброс при записи 1 по шине APB.
15	R/WOCLR	0x0	Header Log Overflow Mask : маска ошибки переполнения Header Log Register. Сброс при записи 1 по шине APB.
31:16	R	0x0	Reserved

Регистр управления и характеристик расширенного обработчика ошибок (PCIe_RP_i_adv_err_cap_ctl)

Таблица 5.108 – Описание полей регистра PCIe_RP_i_adv_err_cap_ctl

Биты	Тип доступа	Начальное значение	Описание
4:0	R	0x0	First Error Pointer : указывает на ошибку из PCIe_RP_i_uncorr_err_status , которая была обнаружена первой. Значение удерживается до тех пор, пока не будет сброшено соответствующее поле регистра PCIe_RP_i_uncorr_err_status. После сброса бита статуса текущей ошибки, следующий из установленных битов ошибок обновит указатель до соответствующего значения. Доступен по APB шине только на чтение.
5	R	0x1	ECRC Generation Capability : показывает, что контроллер может генерировать ECRC для посылаемых пакетов. Доступен по APB шине на запись.
6	R/W	0x0	Enable ECRC Generation : разрешает контроллеру генерацию ECRC для посылаемых пакетов. Доступен по APB шине на запись.
7	R	0x1	ECRC Check Capability : показывает, что контроллер может проверять ECRC в принимаемых пакетах. Доступен по APB шине на запись.
8	R/W	0x0	Enable ECRC Check : разрешает контроллеру проверку ERCR в принимаемых пакетах. Доступен по APB шине на запись.
9	R	0x0	Multiple Header Recording Capable : устанавливается, если функция может сохранить более одного заголовка в регистрах Header Log Registers. Всегда равно 0.
10	R	0x0	Multiple Header Recording Enable : установка разрешает сохранять множество заголовков в регистрах Header Log Registers. Всегда равно 0.
11	R	0x0	Reserved
31:12	R	0x0	Reserved

Регистр заголовка пакета 0 (PCIe_RP_i_hdr_log_0)

Это первый из четырех регистров, использующихся для захвата заголовка полученного контроллером TLP пакета в случае возникновения неисправимой ошибки. Если в регистре PCIe_RP_i_uncorr_err_status установлено несколько бит, то здесь отражается заголовок только первой полученной транзакции. Какая ошибка ей соответствует можно определить на основании поля First Error Pointer. Регистр не может быть перезаписан, пока статусный бит соответствующей ошибки не будет сброшен.

Первое двойное слово заголовка TLP сохраняется в этом регистре с обратным порядком байт. Байт, содержащий поля Type/Format заголовка, сохраняется в [31:24] битах регистра Header Log Register 0.

Таблица 5.109 – Описание полей регистра PCIe_RP_i_hdr_log_0

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	Header DWORD 0 : содержит первое слово заголовка принятого TLP. Доступен по APB шине только на чтение.

Регистр заголовка пакета 1 (PCIe_RP_i_hdr_log_1)

Второе двойное слово заголовка TLP сохраняется в этом регистре с обратным порядком байт.

Таблица 5.110 – Описание полей регистра PCIe_RP_i_hdr_log_1

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	Header DWORD 1 : содержит второе слово заголовка принятого TLP. Доступен по APB шине только на чтение.

Регистр заголовка пакета 2 (PCIe_RP_i_hdr_log_2)

Третье двойное слово заголовка TLP сохраняется в этом регистре с обратным порядком байт.

Таблица 5.111 – Описание полей регистра PCIe_RP_i_hdr_log_2

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	Header DWORD 2 : содержит третье слово заголовка принятого TLP. Доступен по APB шине только на чтение.

Регистр заголовка пакета 3 (PCIe_RP_i_hdr_log_3)

Если принятый TLP имеет заголовок длиной 4 DWORDs, то этот регистр содержит четвертое двойное слово с обратным порядком байт.

Таблица 5.112 – Описание полей регистра PCIe_RP_i_hdr_log_3

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	Header DWORD 3 : содержит четвертое слово заголовка принятого TLP. Доступен по APB шине только на чтение.

Регистр управления ошибками в режиме ROOT (PCIe_RP_i_root_err_cmd)

Этот регистр содержит биты, управляющие обработкой хост-контроллером отчетов об ошибках, полученных от ответных устройств. Регистр не доступен по шине PCIe.

Таблица 5.113 – Описание полей регистра PCIe_RP_i_root_err_cmd

Биты	Тип доступа	Начальное значение	Описание
0	NA	0x0	Correctable Error Reporting Enable : разрешение генерации прерывания при получении исправимой ошибки. Доступен по APB шине на запись.
1	NA	0x0	Non-Fatal Error Reporting Enable : разрешение генерации прерывания при получении не фатальной ошибки. Доступен по APB шине на запись.
2	NA	0x0	Fatal Error Reporting Enable : разрешение генерации прерывания при получении фатальной ошибки. Доступен по APB шине на запись.
31:3	NA	0x0	Reserved

Регистр состояния по ошибкам в режиме ROOT (PCIe_RP_i_root_err_stat)

Регистр не доступен по шине PCIe.

Таблица 5.114 – Описание полей регистра PCIe_RP_i_root_err_stat

Биты	Тип доступа	Начальное значение	Описание
0	NA	0x0	Correctable Error Message Received : устанавливается при получении хост-контроллером оповещения от внешнего устройства об исправимой ошибке. Сброс при записи 1 по шине APB.
1	NA	0x0	Multiple Correctable Error Messages Received : устанавливается при получении хост-контроллером оповещения от внешнего устройства об исправимой ошибке и бит Correctable Error Message Received уже установлен. Сброс при записи 1 по шине APB.
2	NA	0x0	Fatal/Non-Fatal Error Message Received : устанавливается при получении хост-контроллером оповещения от внешнего устройства о фатальной/не фатальной ошибке. Сброс при записи 1 по шине APB.
3	NA	0x0	Multiple Fatal/Non-Fatal Error Messages Received : устанавливается при получении хост-контроллером оповещения от внешнего устройства о фатальной/не фатальной ошибке и бит Fatal/Non-Fatal Error Message Received уже установлен. Сброс при записи 1 по шине APB.
4	NA	0x0	First Uncorrectable Fatal : в установленном состоянии показывает, что первая неисправимая ошибка была фатальной. Сброс при записи 1 по шине APB.
5	NA	0x0	Non-Fatal Error Messages Received : устанавливается при получении хост-контроллером оповещения от внешнего устройства об одной или нескольких не фатальных ошибках. Сброс при записи 1 по шине APB.
6	NA	0x0	Fatal Error Messages Received : устанавливается при получении хост-контроллером оповещения от внешнего устройства об одной или нескольких фатальных ошибках. Сброс при записи 1 по шине APB.
31:7	NA	0x0	Reserved

Регистр идентификации источника ошибок (PCIe_RP_i_err_src_id)

Регистр не доступен по шине PCIe.

Таблица 5.115 – Описание полей регистра PCIe_RP_i_err_src_id

Биты	Тип доступа	Начальное значение	Описание
15:0	NA	0x0	Correctable Error Message Source ID : содержит ID устройства, полученное из принятого ERR_COR сообщения. Доступен по APB шине только на чтение.
31:16	NA	0x0	Fatal/Non-Fatal Error Message Source ID : содержит ID устройства, полученное из принятого ERR_FATAL или ERR_NONFATAL сообщения. Доступен по APB шине только на чтение.

5.12.2.3 Регистры локального управления

В этом разделе при описании программно доступных регистров поле «Тип доступа» в таблицах относится к шине APB. Все регистры локального управления недоступны для конфигурационных запросов по шине PCIe. Тем не менее, поскольку они являются частью AXI пространства, к ним можно обращаться с помощью Memory или IO запросов (как к внутренним адресам микросхемы).

Регистр конфигурации физического уровня 0 (PCIe_LocMgmt_i_pl_config_0_reg)

Таблица 5.116 - Описание полей регистра PCIe_LocMgmt_i_pl_config_0_reg

Биты	Тип доступа	Начальное значение	Описание
0	R	0x0	Link Status : текущий статус линка 0 – обучение не завершено; 1 – обучение завершено и линк готов к работе.
2:1	R	0x2	Negotiated Lane Count : согласованное с ответным устройством количество линий (00 = x1, 01 = x2, 10 = x4, 11 = x8)
4:3	R	0x0	Negotiated Speed : текущая скорость работы (00 = 2.5G, 01 = 5G)
5	R	0x1	Link Training Direction : показывает, что линк завершил обучение как: 0 – хост-контроллер; 1 – физическая функция.
6	R	0x0	Phy Error Reporting : режим отслеживания ошибок Phy 0 – совпадение ошибки от Phy с приемом DLLP или TLP; 1 – все ошибки от Phy.
7	R	0x0	Tx Swing Setting : управляет размахом выходного сигнала: 0 – стандартное значение; 1 – пониженное значение для экономии питания в мобильных приложениях.
15:8	R/W	0x0	Received FTS Count for 2.5 GT/s speed : количество быстрых обучающих последовательностей (FTS), которые контроллер должен послать при выходе из L0s состояния.
23:16	R/W	0x0	Received Link ID: ID, полученный от ответного устройства в процессе обучения линка.
29:24	R/W	0x0	LTSSM State : текущее состояние автомата обучения линка.
30	R	0x0	Remote Linkwidth Upconfigure Capability Status : показывает, что в ходе обучения противоположное устройство установило бит Link Upconfigure.
31	R	0x0	Master Loopback Enable : в режиме хост-контроллера включает проверку loopback. Не используется в режиме функции.

Регистр конфигурации физического уровня 1 (PCIe_LocMgmt_i_pl_config_1_reg)

Таблица 5.117 - Описание полей регистра PCIe_LocMgmt_i_pl_config_1_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R/W	0x0	Transmitted Link ID : ID, посылаемое устройством при обучении в режиме хост-контроллера.
15:8	R/W	0x80	Transmitted FTS Count at 2.5 GT/s Speed : количество FTS, посылаемое в ходе обучения и требуемое этому контроллеру для синхронизации после выхода из L0s состояния.
23:16	R/W	0x80	Transmitted FTS Count at 5 GT/s Speed : количество FTS, посылаемое в ходе обучения и требуемое этому контроллеру для синхронизации после выхода из L0s состояния.
31:24	R/W	0x40	Transmitted FTS Count at 8 GT/s Speed : количество FTS, посылаемое в ходе обучения и требуемое этому контроллеру для синхронизации после выхода из L0s состояния. Не должен использоваться в этой микросхеме.

Регистр конфигурации таймера уровня передачи данных по линку (PCIe_LocMgmt_i_dll_tmr_config_reg)

Таблица 5.118- Описание полей регистра PCIe_LocMgmt_i_dll_tmr_config_reg

Биты	Тип доступа	Начальное значение	Описание
8:0	R/W	0x0	Transmit-Side Replay Timeout Adjustment : дополнительная задержка к таймеру повторной посылки сообщения. [8] – знаковый бит [7:0] * 16 нс – длительность интервала
15:9	R	0x0	Reserved
24:16	R/W	0x0	Receive-Side ACK-NAK Replay Timeout Adjustment : дополнительная задержка к таймеру повторной посылки подтверждений (ACK/NAK) [8] – знаковый бит [7:0] * 16 нс – длительность интервала
31:25	R	0x0	Reserved

Регистр количества принимаемых данных 0 (PCIe_LocMgmt_i_rcv_cred_lim_0_reg)

Этот регистр содержит глубины буферов приемника PCIe шины, объявляемые контроллером в ходе инициализации пакетного уровня. При изменении значений регистра необходимо провести переобучение линка, иначе изменения не вступят в силу. После сброса регистр содержит максимально возможные значения, покрывающие буфера полностью. Записывать можно только значения меньше или равные исходным. В противном случае возможна ситуация переполнения приемника, если AXI не будет успевать пропускать транзакции. Значение 0 соответствует бесконечной глубине буферов приемника.

Таблица 5.119- Описание полей регистра PCIe_LocMgmt_i_rcv_cred_lim_0_reg

Биты	**Тип доступа**	**Начальное значение**	**Описание**
11:0	R/W	0x3e0	Posted Payload Credit VC0 : количество 64 битных слов данных, которые может принять контроллер (сообщений, не требующих подтверждения)
19:12	R/W	0x20	Posted Header Credit VC0 : количество заголовков, которые может принять контроллер (сообщений, не требующих подтверждения)
31:20	R/W	0x20	Non-Posted Payload Credit VC0 : количество 64 битных слов данных, которые может принять контроллер (сообщений, требующих подтверждения)

Регистр количества принимаемых данных 1 (PCIe_LocMgmt_i_rcv_cred_lim_1_reg)

Этот регистр содержит глубины буферов приемника PCIe шины, объявляемые контроллером в ходе инициализации пакетного уровня. При изменении значений регистра необходимо провести переобучение линка, иначе изменения не вступят в силу. После сброса регистр содержит максимально возможные значения, покрывающие буфера полностью. Записывать можно только значения меньше или равные исходным. В противном случае возможна ситуация переполнения приемника, если AXI не будет успевать пропускать транзакции. Значение 0 соответствует бесконечной глубине буферов приемника.

Таблица 5.120- Описание полей регистра PCIe_LocMgmt_i_rcv_cred_lim_1_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R/W	0x20	Non-Posted Header Credit Limit VC0 : количество заголовков, которые может принять контроллер (сообщений, требующих подтверждения)
19:8	R/W	0x0	Completion Payload Credit VC0 : количество 64 битных слов данных, которые может принять контроллер (сообщений подтверждений)
23:20	R	0x0	Reserved
31:24	R/W	0x0	Completion Header Credit VC0 : количество заголовков, которые может принять контроллер (сообщений подтверждений)

Регистр количества передаваемых данных 0 (PCIe_LocMgmt_i_transm_cred_lim_0_reg)

Этот регистр содержит глубины буферов приемника ответного устройства, полученные контроллером в ходе инициализации пакетного уровня.

Таблица 5.121 - Описание полей регистра PCIe_LocMgmt_i_transm_cred_lim_0_reg

Биты	Тип доступа	Начальное значение	Описание
11:0	R	0x0	Posted Payload Credit VC0 : количество 64 битных слов данных, которые может принять ответный контроллер (сообщений, не требующих подтверждения)
19:12	R	0x0	Posted Header Credit VC0 : количество заголовков, которые может принять ответный контроллер (сообщений, не требующих подтверждения)
31:20	R	0x0	Non-Posted Payload Credit VC0 : количество 64 битных слов данных, которые может принять ответный контроллер (сообщений, требующих подтверждения)

Регистр количества передаваемых данных 1 (PCIe_LocMgmt_i_transm_cred_lim_1_reg)

Этот регистр содержит глубины буферов приемника ответного устройства, полученные контроллером в ходе инициализации пакетного уровня.

Таблица 5.122 - Описание полей регистра PCIe_LocMgmt_i_transm_cred_lim_1_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x0	Non-Posted Header Credit Limit VC0 : количество заголовков, которые может принять ответный контроллер (сообщений, требующих подтверждения)
19:8	R	0x0	Completion Payload Credit VC0 : количество 64 битных слов данных, которые может принять ответный контроллер (сообщений подтверждений)
23:20	R	0x0	Reserved
31:24	R	0x0	Completion Header Credit VC0 : количество заголовков, которые может принять ответный контроллер (сообщений подтверждений)

Регистр конфигурации при передаче 0

(PCIe_LocMgmt_i_transm_cred_update_int_config_0_reg)

Таблица 5.123 - Описание полей регистра PCIe_LocMgmt_i_transm_cred_update_int_config_0_reg

Биты	Тип доступа	Начальное значение	Описание
15:0	R/W	16'd4	Minimum Posted Update Interval : минимальный период обновления информации об освобождении места в принимающем FIFO. Касается только сообщений, не требующих подтверждений. Значение периода равно [15:0]*16ns. При изменении пропускной способности будет перезаписано значение по умолчанию.
31:16	R/W	16'd4	Minimum Non-Posted Update Interval : минимальный период обновления информации об освобождении места в принимающем FIFO. Касается только сообщений, требующих подтверждений. Значение периода равно [31:16]*16ns. При изменении пропускной способности будет перезаписано значение по умолчанию.

Регистр конфигурации при передаче 1

(PCIe_LocMgmt_i_transm_cred_update_int_config_1_reg)

Таблица 5.124 - Описание полей регистра PCIe_LocMgmt_i_transm_cred_update_int_config_1_reg

Биты	Тип доступа	Начальное значение	Описание
15:0	R/W	16'd4	Minimum Completion Update Interval : минимальный период обновления информации об освобождении места в принимающем FIFO. Касается только сообщений, не требующих подтверждений. Значение периода равно [15:0]*16ns.
31:16	R/W	16'd938	Maximum Update Interval : максимальный период обновления информации об освобождении места в принимающем FIFO. Касается всех сообщений. Если место не появляется, то каждые [31:16]*16ns контроллер посылает повторное сообщение, описывающее доступный объем буфера приемника.

Регистр интервала времени перевода в режим пониженного потребления (PCIe_LocMgmt_i_L0S_timeout_limit_reg)

Таблица 5.125 - Описание полей регистра PCIe_LocMgmt_i_L0S_timeout_limit_reg

Биты	Тип доступа	Начальное значение	Описание
15:0	R/W	0x0177	L0S Timeout : интервал времени, необходимый контроллеру для автоматического перехода в режим пониженного потребления L0s.
31:16	R	0x0	Reserved :

Регистр количества переданных транзакций (PCIe_LocMgmt_i_transmit_tlp_count_reg)

Таблица 5.126 - Описание полей регистра PCIe_LocMgmt_i_transmit_tlp_count_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/WOCLR	0x0	Transmit TLP Count : количество переданных пакетов уровня транзакций (TLP).

Регистр количества переданных двойных слов (PCIe_LocMgmt_i_transmit_tlp_payload_dword_count_reg)

Таблица 5.127 - Описание полей регистра PCIe_LocMgmt_i_transmit_tlp_payload_dword_count_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/WOCLR	0x0	Transmit TLP Payload Byte Count : количество переданных двойных слов (64 бит) в пакетах уровня транзакций (TLP).

Регистр количества принятых транзакций (PCIe_LocMgmt_i_receive_tlp_count_reg)

Таблица 5.128 - Описание полей регистра PCIe_LocMgmt_i_receive_tlp_count_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/WOCLR	0x0	Receive TLP Count : количество принятых пакетов уровня транзакций (TLP).

Регистр количества принятых двойных слов

(PCIe_LocMgmt_i_receive_tlp_payload_dword_count_reg)

Таблица 5.129 - Описание полей регистра PCIe_LocMgmt_i_receive_tlp_payload_dword_count_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/WOCLR	0x0	Receive TLP Payload Byte Count : количество принятых двойных слов (64 бит) в пакетах уровня транзакций (TLP).

Регистр лимита времени на передачу транзакции 0

(PCIe_LocMgmt_i_compln_tmout_lim_0_reg)

Таблица 5.130 - Описание полей регистра PCIe_LocMgmt_i_compln_tmout_lim_0_reg

Биты	Тип доступа	Начальное значение	Описание
23:0	R/W	24'd12_500_000	Completion Timeout Limit : если за время [23:0]*4ns не будет получено подтверждение (completion), контроллер пошлет сообщение повторно. По умолчанию это 50 мс.
31:24	R	0x0	Reserved :

Регистр лимита времени на передачу транзакции 1

(PCIe_LocMgmt_i_compln_tmout_lim_1_reg)

Таблица 5.131 - Описание полей регистра PCIe_LocMgmt_i_compln_tmout_lim_1_reg

Биты	Тип доступа	Начальное значение	Описание
27:0	R/W	28'd50_000_000	Completion Timeout Limit : если в за время [23:0]*4ns не будет получено подтверждение (completion), контроллер пошлет сообщение повторно. По умолчанию это 200 мс.
31:28	R	0x0	Reserved :

Регистр задержки времени до повторного входа в L1 состояние

(PCIe_LocMgmt_i_L1_st_reentry_delay_reg)

Таблица 5.132 - Описание полей регистра PCIe_LocMgmt_i_L1_st_reentry_delay_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	0x0	L1 Re-Entry Delay : Задержка до повторного входа в L1 состояние равна [31:0]*16ns

Регистр идентификации изготовителя и устройства (PCIe_LocMgmt_i_vendor_id_reg)

Таблица 5.133 - Описание полей регистра PCIe_LocMgmt_i_vendor_id_reg

Биты	Тип доступа	Начальное значение	Описание
15:0	R/W	16'h17cd	Vendor ID : ID разработчика микросхемы.
31:16	R/W	16'h17cd	Subsystem Vendor ID : ID разработчика подсистемы.

Регистр задержки времени до входа в L1 состояние (PCIe_LocMgmt_i_pme_turnoff_ack_delay_reg)

Таблица 5.134 - Описание полей регистра PCIe_LocMgmt_i_pme_turnoff_ack_delay_reg

Биты	Тип доступа	Начальное значение	Описание
19:0	R/W	20'd750	L1 Timeout : задержка до входа в L1 состояние равна [19:0]*16ns. Установка параметра в 0 запрещает переход в L1 состояние.
30:20	R	0x0	Reserved
31	R/W	0x0	Disable Check for Link RX IDLE : 0 – отсчет времени для перехода в L1 ведется при отсутствии активности на приемной и передающей линиях; 1 – отсчет времени для перехода в L1 ведется при отсутствии активности только на приемной линии.

Регистр задержки времени от приема контроллером сообщения входа в режим пониженного потребления до отправки подтверждения (PCIe_LocMgmt_i_pme_turnoff_ack_delay_reg)

Таблица 5.135 - Описание полей регистра PCIe_LocMgmt_i_pme_turnoff_ack_delay_reg

Биты	Тип доступа	Начальное значение	Описание
15:0	R/W	0x64	PME Turnoff Ack Delay : задержка в микросекундах от приема контроллером сообщения входа в режим пониженного потребления (PME_Turn_Off message) до отправки подтверждения (PME_TO_Ack). Установка параметра в 0 запрещает подтверждения таких сообщений.
31:16	R	0x0	Reserved

Регистр управления шириной линка (PCIe_LocMgmt_i_linkwidth_control_reg)

Таблица 5.136 - Описание полей регистра PCIe_LocMgmt_i_linkwidth_control_reg

Биты	Тип доступа	Начальное значение	Описание
3:0	R/W	4'b1111	Target Lane Map: карта линий (lane), которые будут включены в интерфейс (link) в ходе переобучения: 4'b0001 – переобучение к линку шириной 1; 4'b0011 – переобучение к линку шириной 2; 4'b1111 – переобучение к линку шириной 4.
15:4	R	0x0	Reserved
16	R/W	0x0	Link Upconfigure Retrain Link : установка 1 запускает переобучение интерфейса в режиме изменения ширины. Аппаратно сбрасывается по окончании процесса.
23:17	R	0x0	Reserved
25:24	R/W	0x0	EP Target Link Speed : содержит значение частоты передачи, на которую должно перейти EP устройство. Этот параметр не должен быть больше параметра Target Link Speed регистра Link Control And Status Register 2. Также он не должен быть больше параметра TODO: 2'b00 – GEN1 – 2.5 ГГц; 2'b01 – GEN2 – 5.0 ГГц.
30:26	R	0x0	Reserved
31	R/W	0x0	EP Link Speed Change Retrain Link : установка 1 запускает переобучение интерфейса в режиме изменения частоты. Аппаратно сбрасывается по окончании процесса.

Регистр разрешения режима SRIS (PCIe_LocMgmt_i_sris_control_reg)

Таблица 5.137 - Описание полей регистра PCIe_LocMgmt_i_sris_control_reg

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	SRIS Enable : разрешение режима SRIS для Phy. Не поддерживается в этой микросхеме.
31:1	R	0x0	Reserved

Теневой регистр заголовка пакета 0 (PCIe_LocMgmt_i_shdw_hdr_log_0_reg)

Таблица 5.138 – Описание полей регистра PCIe_LocMgmt_i_shdw_hdr_log_0_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	0x0	Shadow header log 0 : содержит [31:0] биты заголовка TLP сообщения, полученного с ошибкой. Не требует установки, если уже есть в AER регистрах.

Теневой регистр заголовка пакета 1 (PCIe_LocMgmt_i_shdw_hdr_log_1_reg)

Таблица 5.139 – Описание полей регистра PCIe_LocMgmt_i_shdw_hdr_log_1_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	0x0	Shadow header log 0 : содержит [31:0] биты заголовка TLP сообщения, полученного с ошибкой. Не требует установки, если уже есть в AER регистрах.

Теневой регистр заголовка пакета 2 (PCIe_LocMgmt_i_shdw_hdr_log_2_reg)

Таблица 5.140 – Описание полей регистра PCIe_LocMgmt_i_shdw_hdr_log_2_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	0x0	Shadow header log 2 : содержит [31:0] биты заголовка TLP сообщения, полученного с ошибкой. Не требует установки, если уже есть в AER регистрах.

Теневой регистр заголовка пакета 3 (PCIe_LocMgmt_i_shdw_hdr_log_3_reg)

Таблица 5.141 – Описание полей регистра PCIe_LocMgmt_i_shdw_hdr_log_3_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	0x0	Shadow header log 3 : содержит [31:0] биты заголовка TLP сообщения, полученного с ошибкой. Не требует установки, если уже есть в AER регистрах.

Теневой регистр номера функции TODO (PCIe_LocMgmt_i_shdw_func_num_reg)

Таблица 5.142 – Описание полей регистра PCIe_LocMgmt_i_shdw_func_num_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R/W	0x0	Shadow register target function number : содержит номер функции.
31:8	R	0x0	Reserved

Теневой регистр ошибки функции TODO (PCIe_LocMgmt_i_shdw_ur_err_reg)

Таблица 5.143 – Описание полей регистра PCIe_LocMgmt_i_shdw_ur_err_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x0	Shadow Register UR Error TODO : содержит номер ошибки функции TODO.
31:8	R	0x0	Reserved

Регистр количества принятых сообщений 1 (PCIe_LocMgmt_i_debug_dllp_count_gen1_reg)

Таблица 5.144 - Описание полей регистра PCIe_LocMgmt_i_debug_dllp_count_gen1_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	GEN1 DLLP Count : содержит количество сообщений пакетного уровня (DLLP), полученных при работе на частоте 2.5 ГГц.

Регистр количества принятых сообщений 2 (PCIe_LocMgmt_i_debug_dllp_count_gen2_reg)

Таблица 5.145 - Описание полей регистра PCIe_LocMgmt_i_debug_dllp_count_gen2_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	GEN2 DLLP Count : содержит количество сообщений пакетного уровня (DLLP), полученных при работе на частоте 5 ГГц.

Регистр количества принятых сообщений 3 (PCIe_LocMgmt_i_debug_dllp_count_gen3_reg)

Таблица 5.146 - Описание полей регистра PCIe_LocMgmt_i_debug_dllp_count_gen3_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R	0x0	GEN3 DLLP Count : содержит количество сообщений пакетного уровня (DLLP), полученных при работе на частоте 8 ГГц. Не должен использоваться в этой микросхеме.

Регистр индикации работающих линий (PCIe_LocMgmt_i_negotiated_lane_map_reg)

Таблица 5.147 - Описание полей регистра PCIe_LocMgmt_i_negotiated_lane_map_reg

Биты	Тип доступа	Начальное значение	Описание
3:0	R	0x0	Negotiated Lane Map : по окончании обучения установленные биты соответствуют работающим линиям. Справедливо только в режимах L0 и L0s.
15:4	R	0x0	Reserved
16	R	0x0	Lane Reversal Status : аппаратно устанавливается, если при обучении была инвертирована последовательность линий.
31:17	R	0x0	Reserved

Регистр количества последовательностей ускоренного обучения (PCIe_LocMgmt_i_receive_fts_count_reg)

Таблица 5.148 - Описание полей регистра PCIe_LocMgmt_i_receive_fts_count_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x0	Received FTS Count for 5GT/s Speed : количество последовательностей ускоренного обучения (FTS), которые должны быть посланы этим контроллером при выходе из L0s состояния при работе на частоте 5 ГГц.
31:8	R	0x0	Reserved

Регистр управления отладкой работы интерфейса (PCIe_LocMgmt_i_debug_mux_control_reg)

Таблица 5.149 - Описание полей регистра PCIe_LocMgmt_i_debug_mux_control_reg

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	0x0	Этот регистр используется для отладки работы интерфейса.

Регистр ошибок и состояния (PCIe_LocMgmt_i_local_error_status_register)

Таблица 5.150 - Описание полей регистра PCIe_LocMgmt_i_local_error_status_register

Биты	Тип доступа	Начальное значение	Описание
0	R/WOCLR	0x0	PNP RX FIFO Parity Error : ошибка четности при чтении из PNP FIFO RAM
1	R/WOCLR	0x0	Completion RX FIFO Parity : ошибка четности при чтении из Completion FIFO RAM
2	R/WOCLR	0x0	Replay RAM Parity Error : ошибка четности при чтении из Replay Buffer RAM
3	R/WOCLR	0x0	PNP RX FIFO Overflow : переполнение PNP FIFO RAM
4	R/WOCLR	0x0	Completion RX FIFO Overflow : переполнение Completion FIFO RAM
5	R/WOCLR	0x0	Replay Timeout : достижение таймером повтора максимального значения.
6	R/WOCLR	0x0	Replay Timer Rollover : четыре подряд Replay Timeout с одним и тем же пакетом.
7	R/WOCLR	0x0	Phy Error : ошибка в приемнике на физическом уровне.
8	R/WOCLR	0x0	Malformed TLP Received : прием поврежденного пакета уровня транзакций.
9	R/WOCLR	0x0	Unexpected Completion Received : получение не ожидаемого подтверждения.
10	R/WOCLR	0x0	Flow Control Error : ошибка управления потоком.
11	R/WOCLR	0x0	Completion Timeout : отсутствие подтверждения (completion) в течение требуемого времени.
12	R	0x0	Reserved
16:13	R	0x0	Reserved
17	R	0x0	Reserved
18	R/WOCLR	0x0	Unmapped TC : неправильные класс трафика.
19	R/WOCLR	0x0	MSI Mask Value Change : устанавливается при изменении маски MSI прерываний.
20	R	0x0	Reserved
21	R/WOCLR	0x0	Hardware Autonomous Width Change Disable Toggle : устанавливается при изменении Hardware Autonomous Width Change бита
27:22	R	0x0	Reserved
28	R/WOCLR	0x0	axi master deinterleave buffer ram ecc uncorrectable error : ошибка четности при чтении из AXI Interleaving RAM.
30:29	R	0x0	Reserved
31	R/WOCLR	0x0	AXI slave reorder RAM ECC uncorrectable error : ошибка четности при чтении из AXI Slave Read Reorder RAM.

Регистр масок прерываний (PCIe_LocMgmt_i_local_intrpt_mask_reg)

Таблица 5.151 - Описание полей регистра PCIe_LocMgmt_i_local_intrpt_mask_reg

Биты	Тип доступа	Начальное значение	Описание
0	R/W	1'b1	PNP RX FIFO Parity Error : маска соответствующего прерывания.
1	R/W	1'b1	Completion RX FIFO Parity Error : маска соответствующего прерывания.
2	R/W	1'b1	Replay RAM Parity Error : маска соответствующего прерывания.
3	R/W	1'b1	PNP RX FIFO Overflow : маска соответствующего прерывания.
4	R/W	1'b1	Completion RX FIFO Overflow : маска соответствующего прерывания.
5	R/W	1'b1	Replay Timeout : маска соответствующего прерывания.
6	R/W	1'b1	Replay Timer Rollover : маска соответствующего прерывания.
7	R/W	1'b1	Phy Error : маска соответствующего прерывания.
8	R/W	1'b1	Malformed TLP Received : маска соответствующего прерывания.
9	R/W	1'b1	Unexpected Completion Received : маска соответствующего прерывания.
10	R/W	1'b1	Flow Control Error : маска соответствующего прерывания.
11	R/W	1'b1	Completion Timeout : маска соответствующего прерывания.
12	R	1'b0	Reserved
16:13	R	0x0	Reserved
17	R	0x0	Reserved
18	R/W	0x1	Unmapped TC : маска соответствующего прерывания.
19	R/W	0x1	MSI Mask Value Change : маска соответствующего прерывания.
20	R	1'b0	Reserved
21	R/W	1'b1	Hardware Autonomous Width Change Disable Toggle : маска соответствующего прерывания.
25:22	R	0x0	Reserved
26	R/W	0x0	DMA outbound buffer uncorrectable error mask : маска ошибки четности при чтении из выходного буфера ПДП контроллера.
27	R/W	0x0	DMA inbound buffer uncorrectable error mask : маска ошибки четности при чтении из входного буфера ПДП контроллера.
28	R/W	0x0	axi master deinterleave buffer ram ecc uncorrectable error mask : маска соответствующего прерывания.
30:29	R	0x0	Reserved
31	R/W	0x0	axi slave reorder ram ecc uncorrectable error mask : маска соответствующего прерывания.

Регистр количества транзакций с LCRC ошибкой (PCIe_LocMgmt_i_lcrc_err_count_reg)

Таблица 5.152 - Описание полей регистра PCIe_LocMgmt_i_lcrc_err_count_reg

Биты	Тип доступа	Начальное значение	Описание
15:0	R/WOCLR	0x0	LCRC Error Count : количество пакетов уровня транзакций (TLP), полученных с LCRC ошибкой
31:16	R	0x0	Reserved

Регистр количества исправляемых ошибок (PCIe_LocMgmt_i_ecc_corr_err_count_reg)

Таблица 5.153 - Описание полей регистра PCIe_LocMgmt_i_ecc_corr_err_count_reg

Биты	Тип доступа	Начальное значение	Описание
7:0	R/WOCLR	0x0	PNP FIFO RAM Correctable Error Count : количество исправимых ошибок, обнаруженных при чтении из PNP FIFO RAM. Является счетчиком с насыщением, сбрасывается записью 1 во все биты.
15:8	R/WOCLR	0x0	SC FIFO RAM Correctable Error Count : количество исправимых ошибок, обнаруженных при чтении из SC FIFO RAM Correctable Error Count. Является счетчиком с насыщением, сбрасывается записью 1 во все биты.
23:16	R/WOCLR	0x0	Replay RAM Correctable Error Count : количество исправимых ошибок, обнаруженных при чтении из Replay RAM. Является счетчиком с насыщением, сбрасывается записью 1 во все биты.
31:24	R/WOCLR	0x0	TPH ST RAM Correctable Error Count : параметр не поддерживается в этой микросхеме.

Регистр задержки повторного послания сообщений изменения режима питания (PCIe_LocMgmt_i_pme_service_timeout_delay_reg)

Таблица 5.154 - Описание полей регистра PCIe_LocMgmt_i_pme_service_timeout_delay_reg

Биты	Тип доступа	Начальное значение	Описание
19:0	R/W	0x186A0	PME Service Timeout Delay : задержка повторного послания сообщений PM_PME. Значение задержки равно ([19:0] + 1)*1мкс. Эти сообщения относятся к управлению питанием.
20	R/W	0x0	Disable PME message on PM Status : установка этого бита запрещает автоматическое послание сообщений изменения режима питания (PME).
31:21	R	0x0	Reserved

Регистр идентификации транзакций (PCIe_LocMgmt_i_root_port_requestor_id_reg)

Таблица 5.155 - Описание полей регистра PCIe_LocMgmt_i_root_port_requestor_id_reg

Биты	Тип доступа	Начальное значение	Описание
15:0	R/W	0x0	Root Port Requestor ID : номер шины, устройства и функции (RID) присваиваемые хост-контроллером всем пакетам уровня транзакций (TLP)
31:16	R	0x0	Reserved

Регистр идентификации устройства и шины (PCIe_LocMgmt_i_ep_bus_device_number_reg)

Таблица 5.156 - Описание полей регистра PCIe_LocMgmt_i_ep_bus_device_number_reg

Биты	Тип доступа	Начальное значение	Описание
4:0	R	0x0	Device Number : номер устройства, полученный в режиме EP
7:5	R	0x0	Reserved
15:8	R	0x0	Bus Number : номер шины, полученный в режиме EP
31:16	R	0x0	Reserved

Регистр управления отладкой работы интерфейса 2 (PCIe_LocMgmt_i_debug_mux_control_2_reg)

Таблица 5.157 - Описание полей регистра PCIe_LocMgmt_i_debug_mux_control_2_reg

Биты	Тип доступа	Начальное значение	Описание
0	R	0x0	Reserved
1	R	0x0	Reserved
2	R	0x0	Reserved
3	R/W	0x0	Enable Link Lane Number Check for Loopback and Link Disable : обычно (согласно спецификации PCIe), если устройство во время обучения получает две последовательных TS1 последовательности с установленными битами loopback или disable, то оно переходит в соответствующий режим (независимо от источника TS1). Установка этого параметра включает проверку источника TS1 сообщения, что предотвратит такую ситуацию. Рекомендуемое значение параметра – 0.
4	R	0x0	Reserved
7:5	R	0x0	Reserved
8	R	0x0	Reserved
9	R	0x0	Reserved
31:10	R	0x0	Reserved

Регистр физического базового адреса 0 (PCIe_LocMgmt_i_pf_0_BAR_config_0_reg)

Таблица 5.158 - Описание полей регистра PCIe_LocMgmt_i_pf_0_BAR_config_0_reg

Биты	Тип доступа	Начальное значение	Описание
4:0	R/W	5'hb	BAR 0 Aperture : определяет размер региона №0 в режиме EP Для 32-битной адресации: 5'b00000 = 128 B; 5'b00001 = 256 B; 5'b00010 = 512 B; 5'b00011 = 1 KB; 5'b00100 = 2 KB; 5'b00101 = 4 KB; 5'00110 = 8 KB; 5'00111 = 16 KB; 5'01000 = 32 KB; 5'01001 = 64 KB; 5'01010 = 128 KB; 5'01011 = 256 KB; 5'01100 = 512 KB; 5'01101 = 1 MB; 5'01110 = 2 MB; 5'01111 = 4 MB; 5'10000 = 8 MB; 5'10001 = 16 MB; 5'10010 = 32 MB; 5'10011 = 64 MB; 5'10100 = 128 MB; 5'10101 = 256 MB; 5'10110 = 512 MB; 5'10111 = 1 GB; 5'11000 = 2 GB. Дя 64-битной адресации так же разрешены значения: 5'11001 = 4 GB; 5'11010 = 8 GB; 5'11011 = 16 GB; 5'11100 = 32 GB; 5'11101 = 64 GB; 5'11110 = 128 GB; 5'11111 = 256 GB; остальные значения зарезервированы.
7:5	R/W	3'h4	BAR 0 Control : тип доступа к региону №0: 000 – отсутствие доступа к региону; 001 – доступ типа IO; 100 – доступ типа Memory 32 non prefetchable; 101 – доступ типа Memory 32 prefetchable; 110 – доступ типа Memory 64 non prefetchable; 111 – доступ типа Memory 64 prefetchable; остальные значения зарезервированы.
12:8	R/W	5'hd	BAR 1 Aperture : аналогичный параметр для региона №1
15:13	R/W	3'h4	BAR 1 Control : аналогичный параметр для региона №1
20:16	R/W	5'h5	BAR 2 Aperture : аналогичный параметр для региона №2
23:21	R/W	3'h0	BAR 2 Control : аналогичный параметр для региона №2
28:24	R/W	5'h5	BAR 3 Aperture : аналогичный параметр для региона №3
31:29	R/W	3'h0	BAR 3 Control : аналогичный параметр для региона №3

Регистр физического базового адреса 1 (PCIe_LocMgmt_i_pf_0_BAR_config_1_reg)

Таблица 5.159 - Описание полей регистра PCIe_LocMgmt_i_pf_0_BAR_config_1_reg

Биты	Тип доступа	Начальное значение	Описание
4:0	R/W	5'h5	BAR 4 Aperture : аналогичный параметр для региона №4.
7:5	R/W	3'h0	BAR 4 Control : аналогичный параметр для региона №4.
12:8	R/W	5'h5	BAR 5 Aperture : аналогичный параметр для региона №5.
15:13	R/W	3'h0	BAR 5 Control : аналогичный параметр для региона №5.
31:16	R	0x0	Reserved

Регистр конфигурации функции (PCIe_LocMgmt_i_pf_config_reg)

Таблица 5.160 - Описание полей регистра PCIe_LocMgmt_i_pf_config_reg

Биты	Тип доступа	Начальное значение	Описание
0	R	0x01	Function 0 Enable : включение функции №0. Всегда равно 1.
31:1	R	0x0	Reserved

Регистр конфигурации базовых адресов (PCIe_LocMgmt_i_rc_BAR_config_reg)

Таблица 5.161 - Описание полей регистра PCIe_LocMgmt_i_rc_BAR_config_reg

Биты	Тип доступа	Начальное значение	Описание
5:0	R/W	0x14	RC BAR 0 Aperture : определяет размер региона №0 в режиме хост-контроллера (RP): 0000 = 4B, 00001 =8B,.....01_1111 = 8G, ...10_0100 = 256G остальные значения зарезервированы.
8:6	R/W	0x4	RC BAR 0 control : тип доступа к региону №0 в режиме хост-контроллера: 000 – отсутствие доступа к региону; 001 – доступ типа IO; 100 – доступ типа Memory non prefetchable; 101 – доступ типа Memory prefetchable; остальные значения зарезервированы.
13:9	R/W	0x14	RC BAR 1 Aperture : аналогичный параметр для региона №1
16:14	R/W	0x0	RC BAR 1 control : аналогичный параметр для региона №1
17	R/W	0x0	Type1 cfg prefetchable mem bar enable : разрешение доступа типа Memory prefetchable в стандартном конфигурационном пространстве типа 1 (type1 config space).
18	R/W	0x0	Type1 cfg prefetchable mem bar size : ширина адреса для доступа типа Memory prefetchable в стандартном конфигурационном пространстве типа 1: 0 – 32 бита; 1 – зарезервировано.
19	R/W	0x0	Type1 cfg IO bar enable : разрешение доступа типа IO в стандартном конфигурационном пространстве типа 1.
20	R/W	0x0	Type1 cfg IO bar size : ширина адреса для доступа типа IO в стандартном конфигурационном пространстве типа 1: 0 – 32 бита; 1 – зарезервировано.
30:21	R	0x0	Reserved
31	R/W	0x0	RC BAR Check Enable : 1 – включение проверки адресов принимаемых сообщений на соответствие границам регионов; 0 – настройки BAR игнорируются, все сообщения пропускаются.

Регистр количества исправляемых ошибок для вспомогательных памятей на AXI шине (PCIe_LocMgmt_i_ecc_corr_err_count_reg_axi)

Таблица 5.162 - Описание полей регистра PCIe_LocMgmt_i_ecc_corr_err_count_reg_axi

Биты	Тип доступа	Начальное значение	Описание
7:0	R/WOCLR	0x0	AXI slave reorder RAM ECC correctable error count : количество исправимых ошибок при чтении из AXI Slave Read Reorder RAM. Запись 0xFF сбросит счетчик к начальному значению.
23:8	R	0x0	Reserved
31:24	R/WOCLR	0x0	AXI master deinterleave buffer RAM ECC Correctable Error Count : количество исправимых ошибок при чтении из AXI Interleaving SRAM. Запись 0xFF сбросит счетчик к начальному значению.

Регистр 0 управления отладкой перехода в режим L1
(PCIe_LocMgmt_low_power_debug_and_control0)

Таблица 5.163 - Описание полей регистра PCIe_LocMgmt_low_power_debug_and_control0

Биты	Тип доступа	Начальное значение	Описание
23:0	R	24'h000000	Reserved
24	R/W	0x0	Do Not block Request interface : 0 – нормальная работа; 1 – контроллер не блокирует TLP в процессе входа в L1 состояние. Этот параметр используется только для отладки.
26:25	R	0x0	L1 entry mode : показывает каким образом контроллер в последний раз вошел в L1 состояние: [0] – автоматически (ASPM); [1] – программно (PM). Этот параметр используется только для отладки.
27	R/W	0x0	Disable L1 exit upon Pending Tlps : запрещает автоматический выход из L1 состояния при получении TLP. Этот параметр используется только для отладки.
31:28	R	4'h0	Reserved

Регистр 1 управления отладкой перехода из режима L1
(PCIe_LocMgmt_low_power_debug_and_control1)

Таблица 5.164 - Описание полей регистра PCIe_LocMgmt_low_power_debug_and_control1

Биты	Тип доступа	Начальное значение	Описание
7:0	R	0x0	L1 or L1.x Exit Trigger conditions : показывает событие, вызвавшее выход из L1 (и его подтипов) состояния: [0] – установка сигнала CLIENT_REQ_EXIT_L1 (доступен из системного контроллера CSC в этой микросхеме); [1] – обнаружен дифференциальный сигнал на входной линии; [2] – получено новое сообщение уровня транзакций (TLP); [3] – внутренний запрос на послание сообщения уровня транзакций; [4] – транзакции от встроенного ПДП контроллера или ранее заблокированные транзакции; [5] – установка сигнала #CLKREQ; [6] – установка сигнала CLIENT_REQ_EXIT_L1_SUBSTATE; [7] – обращение к регистрам. События 5, 6, 7 только для подтипов состояния L1 Этот параметр используется только для отладки.
31:8	R	24'h000000	Reserved

Регистр внутреннего управления уровнем транзакций (PCIe_LocMgmt_tl_internal_control)

Таблица 5.165 - Описание полей регистра PCIe_LocMgmt_tl_internal_control

Биты	Тип доступа	Начальное значение	Описание
0	R/W	0x0	Enable CRS under FLR : разрешает принятие внешних конфигурационных сообщений при высокоуровневом сбросе (FLR – function level reset).
31:1	R	31'h00000000	Reserved

5.12.2.4 Регистры управления AXI портом

В этом разделе при описании программно доступных регистров поле «Тип доступа» в таблицах относится к шине APB. Все регистры локального управления недоступны для конфигурационных запросов по шине PCIe. Тем не менее, поскольку они являются частью AXI пространства, к ним можно обращаться с помощью Memory или IO запросов (как к внутренним адресам микросхемы).

Регистр трансляции адреса 0 региона 0 (PCIe_AXI_outregion_0_addr_translation_0)

Этот регистр содержит параметры трансляции адресов AXI slave шины. Подробное описание трансляции адресов описано в разделе 5.12.8.

Таблица 5.166 - Описание полей регистра PCIe_AXI_outregion_0_addr_translation_0

Биты	Тип доступа	Начальное значение	Описание
5:0	R/W	6'h00	Number_bits : ([5:0] + 1) бит адреса, начиная с нулевого, не будут изменены в ходе трансляции
7:6	R	2'b00	Reserved
31:8	R/W	24'h00000000	Address bits [31:8] : эти биты заменяют соответствующие биты адреса AXI шины. Замене подлежат все биты, не попадающие в интервал, определяемый параметром Number_bits

Регистр трансляции адреса 1 региона 0 (PCIe_AXI_outregion_0_addr_translation_1)

Этот регистр содержит параметры трансляции адресов AXI slave шины.

Таблица 5.167 - Описание полей регистра PCIe_AXI_outregion_0_addr_translation_1

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Address bits [63:32] : эти биты заменяют [63:32] биты адреса AXI шины. Замене подлежат все биты, не попадающие в интервал, определяемый параметром Number_bits. (Не смотря на то, что адрес AXI шины 32 битный, адрес PCI шины может быть 64 битным).

Регистр заголовка-дескриптора 0 региона 0 (PCIe_AXI_outregion_0_header_descriptor_0)

Таблица 5.168 - Описание полей регистра PCIe_AXI_outregion_0_header_descriptor_0

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Descriptor bits [31:0] : младшая часть дескриптора региона №0: [3:0] – тип транзакции (0010 = memory IO, 0110 = IO, остальные значения зарезервированы). [4] – отмена режима подсматривания (No Snoop). [5] – произвольный порядок сообщений (Relaxed Ordering). [6] – порядок сообщений в зависимости от ID (ID-Based Ordering). [16:7] – Reserved [19:17] – класс трафика (PCIe Traffic Class). [20] – при транзакции типа write memory IO, делает его ошибочным (устанавливает бит poisoned). [21] – добавление ECRC к сообщениям независимо от остальных настроек. [22] – Reserved [23] – разрешение использования битов [31:27] текущего регистра в качестве номера устройства и битов [7:0] следующего регистра в качестве номера шины. В противном случае используются значения, полученные в ходе обучения PCIe. Должен быть установлен в режиме RP. [26:24] – номер функции. [31:27] – номер устройства при установленном бите [23], иначе не используется.

Регистр заголовка-дескриптора 1 региона 0 (PCIe_AXI_outregion_0_header_descriptor_1)

Таблица 5.169 - Описание полей регистра PCIe_AXI_outregion_0_header_descriptor_1

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Descriptor bits [31:0] : старшая часть дескриптора региона №0 [7:0] – номер шины при установленном бите [23] предыдущего регистра, иначе не используется. [31:8] – Reserved

Регистр базового адреса 0 региона 0 (PCIe_AXI_outregion_0_region_base_addr)

Таблица 5.170 - Описание полей регистра PCIe_AXI_outregion_0_region_base_addr

Биты	Тип доступа	Начальное значение	Описание
5:0	R/W	6'h0	Lower Mask bits for SOC address : ([5:0] + 1) битов адреса, начиная с нулевого не будут участвовать в выборе региона
7:6	R/W	2'h0	Reserved
31:8	R/W	24'h00000000	Address bits [31:8] : биты этого параметра сравниваются с адресом, полученным по AXI шине. При равенстве выбирается регион №0.

Регистры набора региона x (AXI Outbound Region x Register Sets)

Контроллер PCIe поддерживает 32 внешних региона. Для каждого из них есть пять регистров, аналогичных описанным выше для 0-го региона (x = 0,1, … ,31):

PCIe_AXI_outregion_x_addr_translation_0,

PCIe_AXI_outregion_x_addr_translation_1,

PCIe_AXI_outregion_x_header_descriptor_0,

PCIe_AXI_outregion_x_header_descriptor_1,

PCIe_AXI_outregion_ x _region_base_addr.

Регистр трансляции базового адреса 0 (PCIe_AXI_inregion_rc_bar_0_addr_translation)

Таблица 5.171 - Описание полей регистра PCIe_AXI_inregion_rc_bar_0_addr_translation

Биты	Тип доступа	Начальное значение	Описание
5:0	R/W	6'h00	Number_bits : ([5:0] + 1) бит адреса, начиная с нулевого, не будут изменены в ходе трансляции.
7:6	R	2'b00	Reserved
31:8	R/W	24'h000000 00	Address bits [31:8] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, не попадающие в интервал, определяемый параметром Number_bits.

Регистр трансляции базового адреса 1 (PCIe_AXI_inregion_rc_bar_1_addr_translation)

Таблица 5.172 - Описание полей регистра PCIe_AXI_inregion_rc_bar_1_addr_translation

Биты	Тип доступа	Начальное значение	Описание
5:0	R/W	6'h00	Number_bits : ([5:0] + 1) бит адреса, начиная с нулевого, не будут изменены в ходе трансляции.
7:6	R	2'b00	Reserved
31:8	R/W	24'h000000 00	Address bits [31:8] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, не попадающие в интервал, определяемый параметром Number_bits.

Регистр трансляции базового адреса 2 (PCIe_AXI_inregion_rc_bar_2_addr_translation)

Таблица 5.173 - Описание полей регистра PCIe_AXI_inregion_rc_bar_2_addr_translation

Биты	Тип доступа	Начальное значение	Описание
5:0	R/W	6'h00	Number_bits : ([5:0] + 1) бит адреса, начиная с нулевого, не будут изменены в ходе трансляции.
7:6	R	2'b00	Reserved
31:8	R/W	24'h00000000	Address bits [31:8] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, не попадающие в интервал, определяемый параметром Number_bits.

Регистр индикации программного сброса интерфейса (PCIe_AXI_l0)

Таблица 5.174 - Описание полей регистра PCIe_AXI_l0

Биты	Тип доступа	Начальное значение	Описание
0	R/W	1'h0	Link down indication bit : этот бит устанавливается автоматически при программном сбросе интерфейса (link down reset). Потом необходимо записать 0 для работы.
31:1	R	31'h00000000	Reserved

Регистр 0 трансляции базового адреса 0 (PCIe_AXI_inregion_ep_bar_0_addr_translation)

Таблица 5.175 - Описание полей регистра PCIe_AXI_inregion_ep_bar_0_addr_translation

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Address bits [31:0] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, указанные в регистре 0.

Регистр 0 трансляции базового адреса 1 (PCIe_AXI_inregion_ep_bar_1_addr_translation)

Таблица 5.176 - Описание полей регистра PCIe_AXI_inregion_ep_bar_1_addr_translation

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Address bits [31:0] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, указанные в регистре 0.

Регистр 0 трансляции базового адреса 2 (PCIe_AXI_inregion_ep_bar_2_addr_translation)

Таблица 5.177 - Описание полей регистра PCIe_AXI_inregion_ep_bar_2_addr_translation

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Address bits [31:0] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, указанные в регистре 0.

Регистр 0 трансляции базового адреса 3 (PCIe_AXI_inregion_ep_bar_3_addr_translation)

Таблица 5.178 - Описание полей регистра PCIe_AXI_inregion_ep_bar_3_addr_translation

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Address bits [31:0] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, указанные в регистре 0.

Регистр 0 трансляции базового адреса 4 (PCIe_AXI_inregion_ep_bar_4_addr_translation)

Таблица 5.179 - Описание полей регистра PCIe_AXI_inregion_ep_bar_4_addr_translation

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Address bits [31:0] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, указанные в регистре 0.

Регистр 0 трансляции базового адреса 5 (PCIe_AXI_inregion_ep_bar_5_addr_translation)

Таблица 5.180 - Описание полей регистра PCIe_AXI_inregion_ep_bar_5_addr_translation

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Address bits [31:0] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, указанные в регистре 0.

Регистр 0 трансляции базового адреса 6 (PCIe_AXI_inregion_ep_bar_6_addr_translation)

Таблица 5.181 - Описание полей регистра PCIe_AXI_inregion_ep_bar_6_addr_translation

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Address bits [31:0] : эти биты заменяют соответствующие биты адреса PCIe шины. Замене подлежат все биты, указанные в регистре 0.

5.12.2.5 Регистры управления контроллером ПДП

В этом разделе при описании программно доступных регистров поле «Тип доступа» в таблицах относится к шине APB. Все регистры локального управления недоступны для конфигурационных запросов по шине PCIe. Тем не менее, поскольку они являются частью AXI пространства, к ним можно обращаться с помощью Memory или IO запросов (как к внутренним адресам микросхемы).

Регистр управления каналом ПДП 0 (PCIe_DMA_channel_0_ctrl)

Таблица 5.182 - Описание полей регистра PCIe_DMA_channel_0_ctrl

Биты	Тип доступа	Начальное значение	Описание
0	R/W	1'b0	Go command bit : запуск работы канала №0 контроллера ПДП
1	R/W	1'b0	Inbound or outbound select : направление передачи данных: 0 – от PCIe шины в микросхему; 1 – от микросхемы в PCIe шину.
31:2	R	30'd0	Reserved

Регистр младшей части дескриптора передачи каналом ПДП 0 (PCIe_DMA_channel_0_sp_l)

Таблица 5.183 - Описание полей регистра PCIe_DMA_channel_0_sp_l

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Start pointer Lower DWORD : младшие 32 бита адреса первого дескриптора передачи.

Регистр старшей части дескриптора передачи каналом ПДП 0 (PCIe_DMA_channel_0_sp_u)

Таблица 5.184 - Описание полей регистра PCIe_DMA_channel_0_sp_u

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Start Pointer Upper DWORD : старшие 32 бита адреса первого дескриптора передачи. Не используются в этой микросхеме.

Регистр младшей части дескриптора атрибутов канала ПДП 0 (PCIe_DMA_channel_0_attr_l)

Таблица 5.185 - Описание полей регистра PCIe_DMA_channel_0_attr_l

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Descriptor Attributes Lower DWORD : младшие 32 бита параметров шины AXI, используемые для получения дескриптора передачи. {1'b0, ar_attr[22]} – arbar [21:20] – ardomain [18:16] – arsnoop [15:12] – arregion [11:8] – arqos [7] – arlock [6:3] – arcache [2:0] – arprot

Регистр старшей части дескриптора атрибутов канала ПДП 0 (PCIe_DMA_channel_0_attr_u)

Таблица 5.186 - Описание полей регистра PCIe_DMA_channel_0_attr_u

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Descriptor Attributes Upper DWORD : старшие 32 бита параметров шины AXI, используемые для получения дескриптора передачи. Не используются в этой микросхеме.

Регистр управления каналом ПДП 1 (PCIe_DMA_channel_1_ctrl)

Таблица 5.187 - Описание полей регистра PCIe_DMA_channel_1_ctrl

Биты	Тип доступа	Начальное значение	Описание
0	R/W	1'b0	Go command bit : запуск работы канала №1 контроллера ПДП
1	R/W	1'b0	Inbound or outbound select : направление передачи данных: 0 – от PCIe шины в микросхему; 1 – от микросхемы в PCIe шину.
31:2	R	30'd0	Reserved

Регистр младшей части дескриптора передачи каналом ПДП 1 (PCIe_DMA_channel_1_sp_l)

Таблица 5.188 - Описание полей регистра PCIe_DMA_channel_1_sp_l

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Start pointer Lower DWORD : младшие 32 бита адреса первого дескриптора передачи.

Регистр старшей части дескриптора передачи каналом ПДП 1 (PCIe_DMA_channel_1_sp_u)

Таблица 5.189 - Описание полей регистра PCIe_DMA_channel_1_sp_u

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Start Pointer Upper DWORD : старшие 32 бита адреса первого дескриптора передачи. Не используются в этой микросхеме.

Регистр младшей части дескриптора атрибутов канала ПДП 1 (PCIe_DMA_channel_1_attr_l)

Таблица 5.190 - Описание полей регистра PCIe_DMA_channel_1_attr_l

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Descriptor Attributes Lower DWORD : младшие 32 бита параметров шины AXI, используемые для получения дескриптора передачи. {1'b0, ar_attr[22]} – arbar [21:20] – ardomain [18:16] – arsnoop [15:12] – arregion [11:8] – arqos [7] – arlock [6:3] – arcache [2:0] – arprot

Регистр старшей части дескриптора атрибутов канала ПДП 1 (PCIe_DMA_channel_1_attr_u)

Таблица 5.191 - Описание полей регистра PCIe_DMA_channel_1_attr_u

Биты	Тип доступа	Начальное значение	Описание
31:0	R/W	32'h00000000	Descriptor Attributes Upper DWORD : старшие 32 бита параметров шины AXI, используемые для получения дескриптора передачи. Не используются в этой микросхеме.

Регистр запросов на прерывания от ПДП (PCIe_DMA_common_udma_int)

Таблица 5.192 - Описание полей регистра PCIe_DMA_common_udma_int

Биты	Тип доступа	Начальное значение	Описание
0	R/WOCLR	1'b0	Channel 0 Done Interrupt : окончание каналом №0 передачи, соответствующей текущему дескриптору.
1	R/WOCLR	1'b0	Channel 1 Done Interrupt : окончание каналом №1 передачи, соответствующей текущему дескриптору.
2	R/WOCLR	1'b0	Channel 2 Done Interrupt : не используется в этой микросхеме.
3	R/WOCLR	1'b0	Channel 3 Done Interrupt : не используется в этой микросхеме.
4	R/WOCLR	1'b0	Channel 4 Done Interrupt : не используется в этой микросхеме.
5	R/WOCLR	1'b0	Channel 5 Done Interrupt : не используется в этой микросхеме.
6	R/WOCLR	1'b0	Channel 6 Done Interrupt : не используется в этой микросхеме.
7	R/WOCLR	1'b0	Channel 7 Done Interrupt : не используется в этой микросхеме.
8	R/WOCLR	1'b0	Channel 0 Error Interrupt : ошибка в работе канала №0.
9	R/WOCLR	1'b0	Channel 1 Error Interrupt : ошибка в работе канала №1.
10	R/WOCLR	1'b0	Channel 2 Error Interrupt : не используется в этой микросхеме.
11	R/WOCLR	1'b0	Channel 3 Error Interrupt : не используется в этой микросхеме.
12	R/WOCLR	1'b0	Channel 4 Error Interrupt : не используется в этой микросхеме.
13	R/WOCLR	1'b0	Channel 5 Error Interrupt : не используется в этой микросхеме.
14	R/WOCLR	1'b0	Channel 6 Error Interrupt : не используется в этой микросхеме.
15	R/WOCLR	1'b0	Channel 7 Error Interrupt : не используется в этой микросхеме.
31:16	R	16'd0	Reserved

Регистр разрешения прерываний от ПДП (PCIe_DMA_common_udma_int_ena)

Таблица 5.193 - Описание полей регистра PCIe_DMA_common_udma_int_ena

Биты	Тип доступа	Начальное значение	Описание
0	R/WOCLR	1'b1	Channel 0 Done Enable Interrupt : запись этого бита разрешает прерывание по окончанию передачи канала №0.
1	R/WOCLR	1'b1	Channel 1 Done Enable Interrupt : запись этого бита разрешает прерывание по окончанию передачи канала №1.
2	R/WOCLR	1'b1	Channel 2 Done Enable Interrupt : не используется в этой микросхеме.
3	R/WOCLR	1'b1	Channel 3 Done Enable Interrupt : не используется в этой микросхеме.
4	R/WOCLR	1'b1	Channel 4 Done Enable Interrupt : не используется в этой микросхеме.
5	R/WOCLR	1'b1	Channel 5 Done Enable Interrupt : не используется в этой микросхеме.
6	R/WOCLR	1'b1	Channel 6 Done Enable Interrupt : не используется в этой микросхеме.
7	R/WOCLR	1'b1	Channel 7 Done Enable Interrupt : не используется в этой микросхеме.
8	R/WOCLR	1'b1	Channel 0 Error Enable Interrupt : запись этого бита разрешает прерывание при ошибке работы канала №0.
9	R/WOCLR	1'b1	Channel 1 Error Enable Interrupt : запись этого бита разрешает прерывание при ошибке работы канала №1.
10	R/WOCLR	1'b1	Channel 2 Error Enable Interrupt : не используется в этой микросхеме.
11	R/WOCLR	1'b1	Channel 3 Error Enable Interrupt : не используется в этой микросхеме.
12	R/WOCLR	1'b1	Channel 4 Error Enable Interrupt : не используется в этой микросхеме.
13	R/WOCLR	1'b1	Channel 5 Error Enable Interrupt : не используется в этой микросхеме.
14	R/WOCLR	1'b1	Channel 6 Error Enable Interrupt : не используется в этой микросхеме.
15	R/WOCLR	1'b1	Channel 7 Error Enable Interrupt : не используется в этой микросхеме.
31:16	R	16'd0	Reserved

Регистр запрещения прерываний от ПДП (PCIe_DMA_common_udma_int_dis)

Таблица 5.194 - Описание полей регистра PCIe_DMA_common_udma_int_dis

Биты	Тип доступа	Начальное значение	Описание
0	R/WOSET	1'b1	Channel 0 Done Disable Interrupt : запись этого бита запрещает прерывание по окончанию передачи канала №0.
1	R/WOSET	1'b1	Channel 1 Done Disable Interrupt : запись этого бита запрещает прерывание по окончанию передачи канала №1.
2	R/WOSET	1'b1	Channel 2 Done Disable Interrupt : не используется в этой микросхеме.
3	R/WOSET	1'b1	Channel 3 Done Disable Interrupt : не используется в этой микросхеме.
4	R/WOSET	1'b1	Channel 4 Done Disable Interrupt : не используется в этой микросхеме.
5	R/WOSET	1'b1	Channel 5 Done Disable Interrupt : не используется в этой микросхеме.
6	R/WOSET	1'b1	Channel 6 Done Disable Interrupt : не используется в этой микросхеме.
7	R/WOSET	1'b1	Channel 7 Done Disable Interrupt : не используется в этой микросхеме.
8	R/WOSET	1'b1	Channel 0 Error Disable Interrupt : запись этого бита запрещает прерывание при ошибке работы канала №0.
9	R/WOSET	1'b1	Channel 1 Error Disable Interrupt : запись этого бита запрещает прерывание при ошибке работы канала №1.
10	R/WOSET	1'b1	Channel 2 Error Disable Interrupt : не используется в этой микросхеме.
11	R/WOSET	1'b1	Channel 3 Error Disable Interrupt : не используется в этой микросхеме.
12	R/WOSET	1'b1	Channel 4 Error Disable Interrupt : не используется в этой микросхеме.
13	R/WOSET	1'b1	Channel 5 Error Disable Interrupt : не используется в этой микросхеме.
14	R/WOSET	1'b1	Channel 6 Error Disable Interrupt : не используется в этой микросхеме.
15	R/WOSET	1'b1	Channel 7 Error Disable Interrupt : не используется в этой микросхеме.
31:16	R	16'd0	Reserved

Регистр количества неисправимых ошибок ECC при приёме (PCIe_DMA_common_udma_ib_ecc_uncorrectable_errors)

Таблица 5.195 - Описание полей регистра PCIe_DMA_common_udma_ib_ecc_uncorrectable_errors

Биты	Тип доступа	Начальное значение	Описание
15:0	R	16'd0	ECC Error Reg : количество неисправимых ошибок, возникших при чтении из Inbound DMA RAM
31:16	R	16'd0	Reserved

Регистр количества исправимых ошибок ECC при приёме (PCIe_DMA_common_udma_ib_ecc_correctable_errors)

Таблица 5.196 - Описание полей регистра PCIe_DMA_common_udma_ib_ecc_correctable_errors

Биты	Тип доступа	Начальное значение	Описание
15:0	R	16'd0	ECC Error Reg : количество исправимых ошибок, возникших при чтении из Inbound DMA RAM.
31:16	R	16'd0	Reserved

Регистр количества неисправимых ошибок ECC при передаче (PCIe_DMA_common_udma_ob_ecc_uncorrectable_errors)

Таблица 5.197 - Описание полей регистра PCIe_DMA_common_udma_ob_ecc_uncorrectable_errors

Биты	Тип доступа	Начальное значение	Описание
15:0	R	16'd0	ECC Error Reg : количество неисправимых ошибок, возникших при чтении из Outbound DMA RAM.
31:16	R	16'd0	Reserved

Регистр количества исправимых ошибок ECC при передаче (PCIe_DMA_common_udma_ob_ecc_correctable_errors)

Таблица 5.198 - Описание полей регистра PCIe_DMA_common_udma_ob_ecc_correctable_errors

Биты	Тип доступа	Начальное значение	Описание
15:0	R	16'd0	ECC Error Reg : количество исправимых ошибок, возникших при чтении из Outbound DMA RAM.
31:16	R	16'd0	Reserved

Регистр версии ПДП (PCIe_DMA_common_udma_cap_ver)

Таблица 5.199 - Описание полей регистра PCIe_DMA_common_udma_cap_ver

Биты	Тип доступа	Начальное значение	Описание
7:0	R	8'd1	min_ver : минимальная версия.
15:8	R	8'd0	maj_ver : максимальная версия.
31:16	R	16'd0	Reserved

Регистр конфигурации ПДП (PCIe_DMA_common_udma_config)

Таблица 5.200 - Описание полей регистра PCIe_DMA_common_udma_config

Биты	Тип доступа	Начальное значение	Описание
3:0	R	4'd2	num_channels : количество каналов ПДП.
7:4	R	4'd2	num_partitions : количество разделов.
11:8	R	4'd6	partition_size : размер раздела.
12	R	1'd0	sys_aw_gt_32 : ширина внутреннего адреса больше 32 бит.
13	R	1'd0	sys_tw_gt_32 : ширина атрибута для получения внутреннего дескриптора больше 32 бит.
14	R	1'd1	ext_aw_gt_32 : ширина внешнего адреса больше 32 бит.
15	R	1'd0	ext_tw_gt_32 : ширина атрибута для получения внешнего дескриптора больше 32 бит.
31:16	R	16'd0	Reserved

5.12.3 Базовое преобразование адресов

Контроллер интерфейса PCIe может изменять (транслировать) адреса, получаемые им изнутри по AXI шине или снаружи по PCIe шине.

5.12.3.1 Изменение адресов AXI шины

Трансляция адреса одинакова для режимов RP и EP и происходит по следующему алгоритму.

Сначала полученному по AXI шине адресу ставится в соответствие один из 32 регионов (или никакой из них). Несколько младших бит адреса (количество определяется полем Lower Mask bits for SOC address регистра PCIe_AXI_outregion_0_region_base_addr или аналогичного регистра для другого региона) не участвуют в выборе. Оставшиеся старшие биты адреса сравниваются с полем Address bits того же регистра. Это сравнение производится для всех 32 регионов. При совпадении считается, что AXI транзакция относится к этому региону. При нескольких совпадениях больший приоритет у региона с меньшим номером.

Далее происходит трансляция адреса в соответствии с правилом, установленным для выбранного региона. Несколько младших бит адреса не изменяются (количество определяется полем Number_bits регистра PCIe_AXI_outregion_0_addr_translation_0 или аналогичного регистра для другого региона). Вместо оставшихся старших бит подставляется значение поля Address bits того же регистра. При 64 разрядной адресации на PCIe шине, адрес дополняется значениями из регистра PCIe_AXI_outregion_0_addr_translation_1 или аналогичного регистра для другого региона. Таким образом, исходное пространство в 1 Гбайт можно отображать на 32 отдельных пространства.

Каждому региону присуще не только свое правило изменения адреса, но и тип транзакции (заголовок TLP). Он описывается в регистрах PCIe_AXI_outregion_0_header_descriptor_0 и PCIe_AXI_outregion_0_header_descriptor_1.

5.12.3.2 Изменение адресов PCIe шины

Трансляция адреса различается для режимов RP и EP.

Для режима EP она происходит по следующему алгоритму. Сначала полученному по PCIe шине адресу ставится в соответствие один из 6 регионов, или никакой из них. Каждый регион имеет базовый адрес и размер (поля Base Address и BAR 0 Aperture в регистрах PCIe_EP_i_base_addr_0 и PCIe_LocMgmt_i_pf_0_BAR_config_0_reg или аналогичных регистров для другого региона). При попадании адреса PCIe шины в описанный диапазон,

считается, что PCIe транзакция относится к этому региону. При нескольких совпадениях больший приоритет у региона с меньшим номером.

Далее происходит трансляция адреса в соответствии с правилом, установленным для выбранного региона. Несколько младших бит адреса не изменяются (количество определяется размером региона). Вместо оставшихся старших бит подставляется значение поля Address bits регистра PCIe_AXI_inregion_ep_bar_0_addr_translation или аналогичного регистра для другого региона.

В режиме RP количество регионов равно 2. При трансляции адреса количество младших неизменяемых бит определяется не размером региона, а полем Number_bits регистра PCIe_AXI_inregion_rc_bar_0_addr_translation или аналогичного регистра для другого региона.

5.12.3.3 Встроенный контроллер прямого доступа

Контроллер интерфейса PCIe имеет встроенный контроллер прямого доступа (uDMA). Параметры ПДП передач располагаются в связанном списке дескрипторов (descriptor linked list), который, в свою очередь, должен быть расположен в памяти микросхемы.

Встроенный контроллер ПДП поддерживает три типа передач: блочную, распределенную запись и распределенное чтение.

При блочной передаче (Bulk mode) данные передаются одним целым массивом (см. рисунок 5.26). При этом используется один дескриптор. Размер передаваемых данных ограничен 16 Мбайт. В блочном режиме можно динамически добавлять новые дескрипторы к связанному списку.

Рисунок 5.26 – Блочная передача данных контроллером ПДП

При распределенной записи (Scattering mode) данные читаются из одной области и пишутся в несколько разных областей (см. рисунок 5.27). При этом используется несколько дескрипторов. Первый описывает параметры чтения. Следующие дескрипторы описывают параметры для записей данных. Общий размер данных в этом случае ограничен 16 Кбайт (размером буфера ПДП).

Рисунок 5.27 – Распределенная запись данных контроллером ПДП

При распределенном чтении (Gathering mode) данные читаются из нескольких разных областей и пишутся в одну область (см. Рисунок 5.28). При этом используется несколько дескрипторов. Первые несколько дескрипторов описывают параметры для чтения данных.

Последний дескриптор описывает параметры записи. Общий размер данных в этом случае ограничен 16 Кбайт (размером буфера ПДП).

Рисунок 5.28 – Распределенное чтение данных контроллером ПДП

Дескриптор ПДП транзакции

Таблица 5.201 – Содержание дескриптора ПДП транзакции

Смещение от базового адреса дескриптора	Размер (байт)	Описание	
0	4	31:0	Начальный адрес считываемых/записываемых по AXI шине данных. Базовое преобразование адресов не применяется к этому адресу. Трансляция входящих транзакций применяется к этому адресу.
4	4		Управляющие параметры AXI шины
		31:8	Reserved
		7	AxLOCK[0]
		6:3	AxCACHE[3:0]
		2:0	AxPROT[2:0]
8	8	63:0	Начальный адрес считываемых/записываемых по PCIe шине данных
16	4		Параметры заголовков TLP сообщений
		31:26	Reserved
		25:10	Используемый ID
		9	Установка разрешает использование поля [25:10]
		8:6	Reserved
		5:3	PCIe Transfer Class
		2	ID-Based Ordering
		1	Relaxed Ordering
		0	No Snoop
20	3	23:0	Количество пересылаемых данных в байтах. При равенстве нулю пересылается 2^24 байт.
23	1		Общие настройки дескриптора
		7:6	Reserved
		5	Разрешение перехода к следующему дескриптору по окончанию текущего
		4:3	Reserved
		2:1	Назначение текущего дескриптора: 00 – блочное чтение и запись 01 – чтение данных во внутренний буфер при распределенных транзакциях (по AXI или PCIe шине) 10 – выдача данных из внутреннего буфера при распределенных транзакциях (по AXI или PCIe шине)
		0	Разрешение прерывания после выполнения операции.
24	1		Статус AXI шины
		7:3	Reserved
		2	Обнаружена внутренняя ошибка четности данных (при обращении к AXI шине)
		1:0	BRESP[1:0] или RRESP[1:0]

Продолжение таблицы 5.201

Смещение от базового адреса дескриптора	Размер (байт)	Описание	
25	1		Статус PCIe шины
		7:4	Reserved
		3:0	Статус завершения TLP сообщения: 0000 – нормальное завершение (normal completion); 0001 – ошибочное завершение (TLP poisoned); 0010 – ошибочное завершение (UR, CA или CRS); 0011 – ошибочное завершение (отсутствие данных); 0100 – сообщение завершения несет неправильные ID, класс трафика или атрибуты; 0101 – сообщение завершения несет неправильную младшую часть адреса; 0110 – сообщение завершения несет неправильный тэг; 0111 – время ожидания завершения истекло; 1000 – сообщение завершения несет неправильное количество данных; 1001 .. 1011 – Reserved; 1100 – внутренняя ошибка четности данных (при обращении к PCIe шине); 1101 .. 1110 – Reserved; 1111 – ошибки нескольких типов обнаружены.
26	1		Статус канала ПДП
		7	Buffer Not Empty : Показывает наличие данных в буфере ПДП после завершения исполнения связанного списка дескрипторов.
		6	Buffer Underflow : устанавливается в режиме распределенной записи (Scattering data), если не хватает данных в буфере ПДП.
		5	Buffer Overflow : устанавливается в режиме распределенного чтения (Gathering data), если данных в буфере ПДП слишком много.
		4	Descriptor Error : устанавливается при неправильном дескрипторе.
		3	Data Integrity Error : устанавливается при ошибке четности при обращении к буферам ПДП контроллера.
		2	Устанавливается, если по шине AXI были переданы не все данные.
		1	Устанавливается, если по шине PCIe были переданы не все данные.
		0	Устанавливается по окончании работы контроллера ПДП с данным дескриптором.
27	1	7:0	Reserved
28	4	31:0	Указатель на следующий дескриптор в связанном списке. Базовое преобразование адресов не применяется к этому адресу. Трансляция входящих транзакций применяется к этому адресу.

5.12.4 Управление прерываниями

5.12.4.1 Исходящие прерывания

Текущий контроллер PCIe находясь в режиме конечного устройства (EndPoint) позволяет генерировать два типа прерываний – Legacy и MSI.

Legacy прерывания: в этом случае контроллер в режиме EP имитирует четыре вывода шины PCI: INTA, INTB, INTC и INTD. Вызов прерывания осуществляется программно через системный контроллер CSC. Прерывание от CSC может быть аппаратно преобразовано в разные Legacy прерывания. Параметры настраиваются в регистре PCIe_EP_i_intrpt_line_intrpt_pin контроллера PCIe.

Для установки прерывания необходимо:

- установить бит int_pending_status_reg в регистре ARMSC_PCIE_REG_1;
- установить бит assert200 в регистре ARMSC_PCIE_REG_3;
- дождаться установки бита int_rdy_200 в регистре ARMSC_PCIE_REG_3.

Для снятия прерывания необходимо:

- установить бит deassert200 в регистре ARMSC_PCIE_REG_3;
- дождаться установки бита int_rdy_200 в регистре ARMSC_PCIE_REG_3;
- сбросить бит int_pending_status_reg в регистре ARMSC_PCIE_REG_1.

MSI прерывания: в этом случае прерывания передаются от конечного устройства (EP) к хост-контроллеру (RP) при помощи обычных Memory или IO транзакций. Транзакция генерируется процессором на основе предварительно считанных и проанализированных данных из контроллера PCIe (из MSI Capability Structure). Транзакция передается по AXI и далее по PCIe шинам как обычная запись данных.

5.12.4.2 Входящие прерывания

Legacy прерывания: контроллер PCIe в режиме хост-контроллера выдает отдельно все четыре (INTA, INTB, INTC и INTD) возможных Legacy прерывания на контроллер прерываний GIC. Они попадают туда, соответственно, под номерами 42, 43, 44 и 45.

MSI прерывания: контроллер PCIe в режиме хост-контроллера никак не отличает входящие MSI прерывания от пакетов записи данных и выдает их на мастер AXI. Дальнейшее прохождение прерывания не зависит от интерфейса PCIe, но можно предположить следующие варианты. Первый – данные записываются в память, которая периодически проверяется программно. В случае обнаружения нужного значения процессор распознает

прерывание. Второй – данные записываются в GIC в регистр программных прерываний. Прерывание вызывается и обрабатывается немедленно.

5.13 Системный контроллер CSC

Системный контроллер CSC выполняет следующие функции:

- содержит идентификатор центральной управляющей процессорной системы CCPU,
- содержит регистр режима начальной загрузки,
- содержит регистр управления характеристиками буферов ввода-вывода микросхемы,
- управляет инициализацией контроллера PCI Express,
- реализует интерфейс программных межпроцессорных прерываний между процессорными системами.

5.13.1 Внешние выводы системного контроллера CSC

В таблице ниже приведен список внешних выводов, относящихся к системному контроллеру. Их функциональное назначение описано в разделе 5.17.

Таблица 5.202 – Выводы микросхемы, относящиеся к системному контроллеру

Вывод	Тип буфера	Примечание
BOOTM2, BOOTM1, BOOTM0	I	Режим начальной загрузки процессора

Программно доступные регистры системного контроллера (см. Таблицу 5.203) расположены в области памяти периферийных устройств REG_CCPU и имеют базовое смещение 000C_B000h.

Таблица 5.203 – Регистры системного контроллера CSC

Адрес	Название	Доступ	Описание
000C_B000h	CLUSTERID	ЧТ	Идентификатор центральной управляющей процессорной системы.
000C_B004h	BOOTMODE	ЧТ	Регистр режима начальной загрузки.
000C_B008h	IOBCTRL	ЧТ/ЗП	Регистр управления характеристиками буферов ввода-вывода.
000C_B00Ch	PCIE_RST	ЧТ/ЗП	Регистр сброса PCIe.
000C_B010h	RCVREQ	ЧТ/ЗП	Регистр приёма межпроцессорных прерываний.
000C_B014h	DRVREQ	ЧТ/ЗП	Регистр выдачи межпроцессорных прерываний.
000C_B018h	PCIE_REG_1	ЧТ/ЗП	Регистр управления базовыми настройками PCIe.
000C_B01Ch	PCIE_REG_2	ЧТ/ЗП	Регистр управления потреблением функции.
000C_B020h	PCIE_REG_3	ЧТ/ЗП	Регистр управления Legacy прерываниями.

Идентификатор центральной управляющей процессорной системы (CLUSTERID)

Регистр идентификации центральной управляющей процессорной системы доступен только для чтения. Он содержит 32-разрядный идентификатор, равный 4, что и означает идентификатор CCPU.

Регистр режима начальной загрузки (BOOTMODE)

Регистр режима начальной загрузки (см. Таблицу 5.204) отображает состояние внешних входов BOOTM0, BOOTM1, BOOTM2. Данное значение используется начальным загрузчиком микросхемы для определения способа загрузки.

Таблица 5.204 – Формат регистра BOOTMODE

Разряды	Название	Доступ	Описание
[31:3]	-	-	Зарезервировано
[2]	BOOTM2	ЧТ	Значение входа BOOTM2
[1]	BOOTM1	ЧТ	Значение входа BOOTM1
[0]	BOOTM0	ЧТ	Значение входа BOOTM0

Регистр управления характеристиками буферов ввода-вывода (IOBCTRL)

Регистр управления характеристиками буферов ввода-вывода позволяет программно управлять характеристиками буферов ввода-вывода микросхемы. Для некоторых выводов можно устанавливать следующие характеристики:

- ST0, ST1 - ширина гистерезиса триггеров Шмита на входах (значения: ST0=0, ST1=0 - без гистерезиса; ST0=1, ST1=0 - 0,11 - 0,17 В; ST1=1 - 0,21 - 0,28 В),
- SL (slew rate enable) - сглаживание выбросов выходного фронта за счет времени переключения (1 - есть сглаживание, 0 - нет),
- PE (pullup enable) - включение подтягивающего резистора (1 - включен, 0 - выключен).

Таблица 5.205 – Формат регистра IOBCTRL

Разряды	Характеристи-ка	Начальное значение	Выводы
[31:22]	-	-	Зарезервировано
[21]	PE	1	SRSTn
[20]	PE	1	GPIOC0,GPIOC1,GPIOC2,GPIOC3,GPIOC4,GPIOC5,GPIOC6, GPIOC7
[19]	PE	1	GPIO30,GPIO31,GPIO32,GPIO33,GPIO34,GPIO35,GPIO36, GPIO37
[18]	PE	1	GPIO20,GPIO21,GPIO22,GPIO23,GPIO24,GPIO25,GPIO26, GPIO27
[17]	PE	1	GPIO10,GPIO11,GPIO12,GPIO13,GPIO14,GPIO15,GPIO16, GPIO17
[16]	PE	1	GPIO00,GPIO01,GPIO02,GPIO03,GPIO04,GPIO05,GPIO06, GPIO07
[15]	SL	0	GPIOC0,GPIOC1,GPIOC2,GPIOC3,GPIOC4,GPIOC5,GPIOC6, GPIOC7
[14]	SL	0	GPIO30,GPIO31,GPIO32,GPIO33,GPIO34,GPIO35,GPIO36, GPIO37
[13]	SL	0	GPIO20,GPIO21,GPIO22,GPIO23,GPIO24,GPIO25,GPIO26, GPIO27
[12]	SL	0	GPIO10,GPIO11,GPIO12,GPIO13,GPIO14,GPIO15,GPIO16, GPIO17
[11]	SL	0	GPIO00,GPIO01,GPIO02,GPIO03,GPIO04,GPIO05,GPIO06, GPIO07
[10]	SL	0	WKUPRQ,WKUPAK,TDO_SWO,TMS_SWDIO,SRSTn,WDT
[9]	ST1	0	XRSTn,WKUPRQ
[8]	ST0	0	XRSTn,WKUPRQ
[7]	ST1	0	BS_TDI,BS_TCK,BS_TMS,BS_TRSTn,TDI,TCK_SWCLK, TMS_SWDIO,TRSTn,SRSTn
[6]	ST0	0	BS_TDI,BS_TCK,BS_TMS,BS_TRSTn,TDI,TCK_SWCLK, TMS_SWDIO,TRSTn,SRSTn
[5]	ST1	0	SPI_RXD
[4]	ST0	0	SPI_RXD
[3]	ST1	0	ETH_MDIO
[2]	ST0	0	ETH_MDIO
[1]	ST1	0	GPIOC0,GPIOC1,GPIOC2,GPIOC3,GPIOC4,GPIOC5,GPIOC6, GPIOC7, GPIO30,GPIO31,GPIO32,GPIO33,GPIO34,GPIO35,GPIO36, GPIO37, GPIO20,GPIO21,GPIO22,GPIO23,GPIO24,GPIO25,GPIO26, GPIO27, GPIO10,GPIO11,GPIO12,GPIO13,GPIO14,GPIO15,GPIO16, GPIO17, GPIO00,GPIO01,GPIO02,GPIO03,GPIO04,GPIO05,GPIO06, GPIO07
[0]	ST0	0	GPIOC0,GPIOC1,GPIOC2,GPIOC3,GPIOC4,GPIOC5,GPIOC6, GPIOC7, GPIO30,GPIO31,GPIO32,GPIO33,GPIO34,GPIO35,GPIO36, GPIO37, GPIO20,GPIO21,GPIO22,GPIO23,GPIO24,GPIO25,GPIO26, GPIO27, GPIO10,GPIO11,GPIO12,GPIO13,GPIO14,GPIO15,GPIO16, GPIO17, GPIO00,GPIO01,GPIO02,GPIO03,GPIO04,GPIO05,GPIO06, GPIO07

Регистры приёма и выдачи межпроцессорных прерываний (DRVREQ и RCVREQ)

Регистр выдачи межпроцессорных прерываний (см. таблицу 5.206) используется для выдачи запросов к другим процессорным системам СнК. Регистр приёма межпроцессорных прерываний (см. таблицу 5.207) используется для обработки запросов от других процессорных систем СнК. Разряды регистров DRVREQ и RCVREQ разных процессорных систем соединены между собой (см. таблицу 5.208).

Таблица 5.206 – Формат регистра выдачи межпроцессорных прерываний (DRVREQ)

Разряды	Название	Доступ	Описание
[31:8]	-	-	Зарезервировано
[7:0]	DRVREQ	ЧТ/ЗП	Один бит поля GENREQ соответствует одному биту поля RCVREQ другого ПУ. Запись значения 1 в бит поля GENREQ вызывает аппаратную установку соответствующего бита RCVREQ другого ПУ. Программный сброс бита RCVREQ другого ПУ системы сбрасывает аппаратно бит поля GENREQ.

Таблица 5.207 – Формат регистра приёма межпроцессорных прерываний (RCVREQ)

Разряды	Название	Доступ	Описание
[31:8]	-	-	Зарезервировано
[7:0]	RCVREQ	ЧТ/ЗП	Один бит поля RCVREQ соответствует одному биту GENREQ другого ПУ. Программная установка соответствующего бита GENREQ в другой процессорной системе устанавливает бит RCVREQ. Запись значения 1 в бит поля RCVREQ сбрасывает этот бит и инициирует аппаратный сброс соответствующего бита GENREQ. Установка бита поля RCVREQ вызывает запрос на прерывание.

Таблица 5.208 – Соответствие разрядов полей регистров GENREQ и RCVREQ и запросов на прерывание

Разряд регистра DRVREQ	Разряд регистра RCVREQ	Запрос на прерывание
DRVREQ[7]	RCVREQ[7]	Низкоприоритетное межпроцессорное прерывание 7 к/от PC3
DRVREQ[6]	RCVREQ[6]	Низкоприоритетное межпроцессорное прерывание 6 к/от PC2
DRVREQ[5]	RCVREQ[5]	Низкоприоритетное межпроцессорное прерывание 5 к/от PC1
DRVREQ[4]	RCVREQ[4]	Низкоприоритетное межпроцессорное прерывание 4 к/от PC0
DRVREQ[3]	RCVREQ[3]	Высокоприоритетное межпроцессорное прерывание 3 к/от PC3
DRVREQ[2]	RCVREQ[2]	Высокоприоритетное межпроцессорное прерывание 2 к/от PC2
DRVREQ[1]	RCVREQ[1]	Высокоприоритетное межпроцессорное прерывание 1 к/от PC1
DRVREQ[0]	RCVREQ[0]	Высокоприоритетное межпроцессорное прерывание 0 к/от PC0

Системный контроллер также содержит регистры управления контроллером PCIe:

- регистр сброса PCIe - PCIE_RST (см. Таблицу 5.209),
- регистр управления базовыми настройками PCIe - PCIE_REG_1 (см. таблицу 5.210),
- регистр управления потреблением функции - PCIE_REG_2 (см. таблицу 5.211),
- регистр управления Legacy прерываниями - PCIE_REG_3 (см. таблицу 5.212).

Таблица 5.209 – Формат регистра PCIE_RST

Биты	Тип доступа	Начальное значение	Описание
0	RW	0x0	Resetn : запись 1 – запуск последовательности включения интерфейса; запись 0 – ничего не произойдет; чтение 0 – последовательность включения не выполнена; чтение 1 – последовательность включения выполнена. Сразу после записи 1 чтение должно выдавать 0. Включение здесь показывает готовность блока (стабилизацию PLL) и не имеет отношения к установлению связи с ответным устройством ни на каком уровне.
31:1	R	0x0	Reserved

Таблица 5.210 – Формат регистра PCIE_REG_1

Биты	Тип доступа	Начальное значение	Описание
0	R/W	1'h1	pcie_generation_sel : версия PCIe: 0 - Gen1 (2.5Gbs per lane); 1 - Gen1 (2.5Gbs per lane) или Gen2 (5Gbs per lane) после переобучения, инициированного RC. Изменение этого значения допустимо только до снятия сброса.
1	R/W	1'h0	mode_select : 0 – Endpoint; 1 - Root Complex. Изменение этого значения допустимо только до снятия сброса.
3:2	R/W	2'h2	lane_count_in : количество используемых линий: 00 - линия №0; 01 - линии №0 и №1; 10 - линии №0, №1, №2, и №3; 11 – reserved. Этот параметр должен быть одинаковым у обоих соединенных устройств. Изменение этого значения допустимо только до снятия сброса.
4	R/W	1'h1	link_training_enable_reg : разрешение инициации последовательности обучения: 0 - запрещено; 1 - разрешено.
5	R/W	1'h1	config_enable_reg : 0 - входящие конфигурационные запросы запрещены; возвращается ошибка (CRS Completion) по шине PCIe; 1 - входящие конфигурационные запросы разрешены.
6	R/W	1'h0	req_pm_transition_l23_ready_reg : только для Endpoint. Установка этого бита переводит контроллер PCIe в L23_READY состояние и, впоследствии, в L2 состояние. Это следует делать после того, как PCI функции перешли в состояние D3. После снятия этого бита контроллер должен быть сброшен.
7	R/W	1'h0	client_req_exit_l2_reg : может быть программно установлено после входа линка в состояние L2 и до отключения питания системы. Пока питание и тактовый сигнал еще присутствуют, можно программно установить этот бит, чтобы инициировать переход из L2_IDLE в DETECT состояние.

Продолжение таблицы 5.210

Биты	Тип доступа	Начальное значение	Описание
8	R/W	1'h0	int_pending_status_reg : должно быть установлено, если требуется передача Legacy прерывания, но оно замаскировано.
9	R/W	1'h0	down_cfg_susp_ln_00 : питание линии №0: 0 - включено; 1 - выключено. Изменение этого значения допустимо только до снятия сброса.
10	R/W	1'h0	down_cfg_susp_ln_01 : питание линии №1: 0 - включено; 1 - выключено. Изменение этого значения допустимо только до снятия сброса.
11	R/W	1'h0	down_cfg_susp_ln_02 : питание линии №2: 0 - включено; 1 - выключено. Изменение этого значения допустимо только до снятия сброса.
12	R/W	1'h0	down_cfg_susp_ln_03 : питание линии №3: 0 – включено; 1 – выключено. Изменение этого значения допустимо только до снятия сброса.
13	R/W	1'h0	hot_reset_in_reg : установка этого бита в режиме RC сбрасывает PCIe линк. Не работает в режиме EP.
14	R	1'h0	hot_reset_out_sync : устанавливается в режимах RC и EP по осуществлении горячего сброса.
15	R	1'h0	link_down_reset_out_sync : устанавливается контроллером PCIe при обнаружении события отключения линка (link-down event – LINK_UP переключается в 0)
17:16	R	2'h0	link_status_sync : показывает статус линка PCIe: 0 – ответное устройство не обнаружено; 1 – происходит обучение линка; 2 – линк готов, происходит инициализация на пакетном уровне; 3 – линк готов, инициализация пакетного уровня завершена.
21:18	R	4'h0	link_power_state_sync : текущий режим потребления линка: 1 – L0; 2 – L0s; 4 – L1; 8 – L2.
31:22			Reserved

Таблица 5.211 – Формат регистра PCIE_REG_2

Биты	**Тип доступа**	**Начальное значение**	**Описание**
0	R	1'h0	int200 : устанавливается контроллером при изменении режима потребления функции на D1 или D3. Сохраняется в установленном состоянии, пока не будет подтвержден. При этом контроллер PCIe не будет возвращать завершения транзакций конфигурационных чтений и записей.
1	RWOS	1'h0	ack200 : запись 1 вызовет подтверждение изменения режима потребления функции. Будет сброшен аппаратно вслед за аппаратным сбросом int200.
2	R/W	1'h0	autoack200 : 1 – разрешение автоматического подтверждения изменения режима потребления функции 0 – автоматическое подтверждение отключено
31:3			Reserved

Таблица 5.212 – Формат PCIE_REG_3

Биты	Тип доступа	Начальное значение	Описание
0	RWOS	1'h0	assert200 : запись 1 – вызовет установку Legacy прерывания. Будет аппаратно сброшен после подтверждения установки прерывания.
1	RWOS	1'h0	deassert200 : запись 1 – вызовет снятие Legacy прерывания. Будет аппаратно сброшен после подтверждения снятия прерывания.
2	R	1'h0	int_rdy_200 : устанавливается при получении подтверждения установки/снятия прерывания.
31:3			Reserved

5.14 Контроллер синхронного последовательного интерфейса SPI

Контроллер интерфейса SPI обеспечивает обмен информацией между процессором и устройствами, подключенными к внешнему последовательному интерфейсу типа SPI. Контроллер обеспечивает подключение стандартных устройств со следующими разновидностями интерфейса типа SPI:

- интерфейс SPI в спецификации фирмы Motorola (Motorola SPI);
- интерфейс SPI в спецификации фирмы Texas Instruments (Texas Instruments SPI);
- интерфейс SPI в спецификации фирмы National Semiconductors (National Semiconductors Microwire).

Контроллер обеспечивает подключение до 4 внешних устройств в режиме разделения времени.

Обмен данными между внешним устройством и памятью процессора возможен как в программном режиме, так и в режиме ПДП.

Управление контроллером, программный режим работы с внешними устройствами, а также конфигурация встроенного контроллера ПДП осуществляется по 32-разрядной шине AMBA APB спецификации 3.0. Обмен данными в режиме ПДП осуществляется по 32-разрядной шине AMBA AXI спецификации 3.0, причем порт ПДП подключается к этой шине в качестве “мастера”.

Работа блока контроллера тактируется тактовым сигналом F_{SSPCLK} равным *** МГц. Скорость передачи данных задается программистом путем деления тактовой частоты работы контроллера на программно задаваемый коэффициент. Максимальная скорость передачи данных не должна превышать 50 Мбит/сек.

5.14.1 Внешние выводы SPI интерфейса

В таблице 5.213 приведен список внешних выводов SPI порта микросхемы.

Таблица 5.213 - Выводы порта SPI микросхемы

Вывод	Тип буфера	Функциональное назначение вывода
SPICLK	O	Тактовый сигнал интерфейса SPI
SPITXD	O	Выход данных порта SPI
SPIRXD	I	Вход данных порта SPI
SPI_CS0…SPI_CS3	O	Выбор ведомого SPI устройства 0…3 (активный уровень сигнала – низкий)

5.14.2 Программно доступные регистры контроллера SPI

Программно доступные регистры контроллера порта SPI расположены в области памяти периферийных устройств REG_CCPU, имеют базовое смещение SSP Base = 0x000C_F000 hex и общий размер 4 КБайта. Спецификация регистров представлена в таблице 5.214.

Таблица 5.214 - Спецификация регистров контроллера интерфейса SPI

Адрес (байтовый доступ)	Тип	Разряд-ность	Начальное значение	Имя	Описание
0x000C_F000 hex	ЧТ/ЗП	16	0x0000	SSPCR0	Регистр 0 .
0x000C_F004 hex	ЧТ/ЗП	4	0x0	SSPCR1	Регистр 1 управления режимами работы порта.
0x000C_F008 hex	ЧТ/ЗП	16	0x????	SSPDR	Регистр данных порта.
0x000C_F00C hex	ЧТ	5	0x03	SSPSR	Регистр состояния порта.
0x000C_F010 hex	ЧТ/ЗП	8	0x00	SSPCPSR	Регистр множителя синхросигнала.
0x000C_F014 hex	ЧТ/ЗП	4	0x0	SSPIMSC	Регистр масок прерываний порта.
0x000C_F018 hex	ЧТ	4	0x8	SSPRIS	Регистр состояния прерываний от порта до наложения маски.
0x000C_F01C hex	ЧТ	4	0x0	SSPMIS	Регистр состояния прерываний от порта после наложения маски.
0x000C_F020 hex	ЗП	4	0x0	SSPICR	Регистр сброса прерываний порта.
0x000C_F024 hex	ЧТ/ЗП	2	0x0	SSPDMACR	Регистр управления режимом ПДП порта.
0x000C_F028 hex – 0x000C_F0FF hex	-	-	-	-	Зарезервировано.
0x000C_F100 hex	ЧТ/ЗП	32	0x0000	SSPDMATXADR	Регистр адреса передатчика контроллера ПДП порта.
0x000C_F104 hex	ЧТ/ЗП	32	0x0000	SSPDMARXADR	Регистр адреса приемника контроллера ПДП порта.
0x000C_F108 hex	ЧТ/ЗП	32	0x0000	SSPDMACNT	Счетчик данных контроллера ПДП порта.
0x000C_F10C hex	ЧТ/ЗП	1	0x0	SSPDMARXDIS	Регистр конфигурации режима приема контроллера ПДП порта.
0x000C_F110 hex	ЧТ/ЗП	1	0x0	SSPDMACTR	Регистр управления контроллером ПДП порта.
0x000C_F114 hex – 0x000C_F11B hex	-	-	-	-	Зарезервировано.

Продолжение таблицы 5.214

Адрес (байтовый доступ)	Тип	Разряд-ность	Начальное значение	Имя	Описание
0x000C_F11C hex	ЧТ	2	0x0	SSPDMASTS	Регистр состояния контроллера ПДП порта.
0x000C_F120 hex	ЧТ/ЗП	1	0x0	SSPDMAIM	Регистр маски прерывание контроллера ПДП порта.
0x000C_F124 hex	ЧТ/ЗП	1	0x0	SSPDMAIR	Регистр запроса на прерывание от контроллера ПДП порта.
0x000C_F128 - 0x000C_F12F hex	-	-	-	-	Зарезервировано.
0x000C_F130 hex	ЧТ/ЗП	1	0x1	SSPDMAPT	Регистр упаковки данных контроллера ПДП порта.
0x000C_F134 hex - 0x000C_F13F hex	-	-	-	-	Зарезервировано.
0x000C_F140 hex	ЧТ/ЗП	4	0x1	SSPDMACS	Регистр выбора ведомого устройства порта.
0x000C_F144 hex – 0x000C_F3FF hex	-	-	-	-	Зарезервировано.

Регистр управления 0 контроллера порта ***SSPCR0***

Таблица 5.215 - Формат регистра SSPCR0

Биты	Название	Тип	Описание
[15:8]	SCR	ЧТ/ЗП	Serial Clock Rate. Множитель скорости передачи данных.
[7]	SPH	ЧТ/ЗП	SSPCLKOUT Phase. Фаза сигнала SSPCLKOUT.
[6]	SPO	ЧТ/ЗП	SSPCLKOUT Polarity. Полярность сигнала SSPCLKOUT.
[5:4]	FRF	ЧТ/ЗП	Frame Format. Формат данных: 00 - Motorola SPI формат передачи; 01 - Texas Instruments SPI формат передачи; 10 - National Semiconductors Microwire формат передачи; 11 - не используется.
[3:0]	DSS	ЧТ/ЗП	Data Size Select. Выбор размера данных: 0000…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.

Поле **SCR** используется для определения скорости приема и передачи данных на внешних выводах контроллера. Скорость передачи данных вычисляется по следующей формуле:

$$\frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$$

, где F_{SSPCLK} – тактовая частота работы контроллера порта, CPSDVSR – значение от 2 до 254, задаваемое в регистре множителя синхросигнала SSPCPSR.

Поле SCR может принимать значения от 0 до 255.

Скорость передачи данных не может превышать 50 Мбит/сек.

Поля **SPO, SPH** и **FRF** определяют формат передачи данных. Форматы передачи данных описаны в далее.

Поле **DSS** определяет количество бит, в 16-разрядном слове данных, выдаваемых на внешние выводы интерфейса и, соответственно, принимаемых с внешних выводов. Как при выдаче, так и при приеме данные должны быть выровнены по правому краю 16-разрядного слова.

Регистр управления 1 контроллера порта SSPCR1

Таблица 5.216 - Формат регистра SSPCR1

Биты	Название	Тип	Описание
[15:4]	-	-	Зарезервировано, при записи должен записываться 0.
[3]	-	-	Зарезервировано, при записи должен записываться 1.
[2]	-	-	Зарезервировано, при записи должен записываться 0.
[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.

Регистр данных SSPDR

Таблица 5.217 - Формат регистра SSPDR

Биты	Название	Тип	Описание
[15:0]	DATA	ЧТ/ЗП	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.

Запись в SSPDR приводит к записи данных в буфер передаваемых данных. Буфер передаваемых данных состоит из 8 16-разрядных регистров организованных в очередь по принципу FIFO. Данные из передающего буфера автоматически записываются в передающий сдвиговый регистр и последовательно выдвигаются на внешний вывод SPITXD. Количество выдвигаемых бит программируется полем **DSS** регистра **SSPCR0.**

Одновременно с выдачей данных производится защелкивание принимаемых данных с внешнего вывода SPIRXD. После накопления полного SPI фрейма, определяемого полем **DSS** регистра **SSPCR0,** данные помещаются в приемный буфер данных. Приемный буфер данных также состоит из 8 16-разрядных регистров организованных в очередь по принципу FIFO.

Чтение данных из регистра SSPDR и является чтением данных из приемного буфера. Если разрядность принимаемых данных меньше, чем 16 бит, то данные автоматически выравниваются по правому краю слова.

Состояние приемного и передающего буферов отражается в регистре состояния контроллера **SSPSR.**

Регистр состояния контроллера порта SSPSR

Таблица 5.218 - Формат регистра SSPSR

Биты	Название	Тип	Описание
[15:5]	-	-	Зарезервировано
[4]	BSY	ЧТ	Флаг занятости порта SPI: 0 - порт SPI простаивает; 1 - порт SPI передает и/или принимает пакет данных.
[3]	RFF	ЧТ	Признак полноты принимающего буфера данных: 0 - принимающий буфер не полон; 1 - принимающий буфер полон.
[2]	RNE	ЧТ	Признак пустоты принимающего буфера данных: 0 - принимающий буфер пуст; 1 - принимающий буфер не пуст.
[1]	TNF	ЧТ	Признак полноты передающего буфера данных: 0 - передающий буфер данных полон; 1 - передающий буфер данных не полон.
[0]	TFE	ЧТ	Признак пустоты передающего буфера данных: 0 - передающий буфер данных не пуст; 1 - передающий буфер данных пуст.

Регистр множителя синхросигнала SSPCPSR

Таблица 5.51 - Формат регистра SSPCPSR

Биты	Название	Тип	Описание
[15:8]	-	-	Зарезервирован, должен быть прописан 0
[7:0]	CPSDVSR	ЧТ/ЗП	Clock Prescale Devisor. Значение делителя синхросигнала. Должен принимать четные значения от 2 до 254, в зависимости от частоты тактового сигнала работы контроллера F_{SSPCLK}. Младшие значащие биты всегда принимают значение 0 при чтении.

Регистр маски прерываний SSPIMSC

Таблица 5.219 - Формат регистра SSPIMSC

Биты	Название	Тип	Описание
[15:4]	-	-	Зарезервировано, должен быть прописан 0
[3]	TXIM	ЧТ/ЗП	Маска прерывания по заполнению передающего буфера данных более чем на половину: 0 - прерывание замаскировано; 1 - прерывание не маскировано.
[2]	RXIM	ЧТ/ЗП	Маска прерывания по заполнению приемного буфера данных менее чем на половину: 0 - прерывание замаскировано; 1 - прерывание не маскировано.
[1]	RTIM	ЧТ/ЗП	Маска прерывания по наличию данных в приемном буфере после завершения приема данных с внешних выводов: 0 - прерывание замаскировано; 1 - прерывание не маскировано.
[0]	RORIM	ЧТ/ЗП	Маска прерывания по переполнению приемного буфера данных: 0 - прерывание замаскировано; 1 - прерывание не маскировано.

Запись 1 в соответствующий бит регистра SSPIMSC устанавливает маску, а запись 0 – снимает её. Чтение регистра возвращает текущее значение маски.

Регистр запросов на прерывание до наложения маски SSPRIS

Таблица 5.220 - Формат регистра SSPRIS

Биты	Название	Тип	Описание
[15:4]	-	-	Зарезервировано.
[3]	TXRIS	ЧТ	Запрос на прерывание по заполнению передающего буфера данных более чем на половину.
[2]	RXRIS	ЧТ	Запрос на прерывание по заполнению приемного буфера данных менее чем на половину.
[1]	RTRIS	ЧТ	Запрос на прерывание по наличию данных в приемном буфере после завершения приема данных с внешних выводов.
[0]	RORRIS	ЧТ	Запрос на прерывание по переполнению приемного буфера данных.

Регистр состояния прерывания после наложения маски SSPMIS

Таблица 5.221 - Формат регистра SSPMIS

Биты	Название	Тип	Описание
[15:4]	-	-	Зарезервировано.
[3]	TXMIS	ЧТ	Запрос на прерывание по заполнению передающего буфера данных более чем на половину.
[2]	RXMIS	ЧТ	Запрос на прерывание по заполнению приемного буфера данных менее чем на половину.
[1]	RTMIS	ЧТ	Запрос на прерывание по наличию данных в приемном буфере после завершения приема данных с внешних выводов.
[0]	RORMIS	ЧТ	Запрос на прерывание по переполнению приемного буфера данных.

Регистр сброса запросов на прерывания SSPICR

Таблица 5.222 - Формат регистра SSPICR

Биты	Название	Тип	Описание
[15:2]	-	-	Зарезервирован, должен быть прописан 0
[1]	RTIC	ЗП	Сброс запроса на прерывание по наличию данных в приемном буфере после завершения приема данных с внешних выводов.
[0]	RORIC	ЗП	Сброс запроса на прерывание по переполнению приемного буфера данных.

Регистр управления ПДП SSPDMACR

Таблица 5.223 - Формат регистра SSPDMACR

Биты	Название	Тип	Описание
[15:2]	-	-	Зарезервирован, должен быть прописан 0
[1]	TXDMAE	ЧТ/ЗП	Разрешение доступа ПДП контроллера к передающему буферу данных: 0 - доступ запрещен; 1 - доступ разрешен.
[0]	RXDMAE	ЧТ/ЗП	Разрешение доступа ПДП контроллера к приемному буферу данных: 0 - доступ запрещен; 1 - доступ разрешен.

Регистр начального адреса передаваемых данных контроллера ПДП SSPDMATXADR

Таблица 5.224 - Формат регистра SSPDMATXADR

Биты	Название	Тип	Описание
[31:2]	SSDMA_TXADR	ЧТ/ЗП	Начальный адрес передаваемых данных.
[1:0]	-	-	Зарезервирован, должен быть прописан 0.

Значение адреса должно быть выровнено на границу 32-разрядного слова. В процессе выполнения процедуры ПДП значение регистра не изменяется.

Регистр начального адреса принимаемых данных контроллера ПДП SSPDMARXADR

Таблица 5.225 - Формат регистра SSPDMARXADR

Биты	Название	Тип	Описание
[31:2]	SSDMA_RXADR	ЧТ/ЗП	Начальный адрес принимаемых данных.
[1:0]	-	-	Зарезервирован, должен быть прописан 0.

Значение адреса должно быть выровнено на границу 32-разрядного слова. В процессе выполнения процедуры ПДП значение регистра не изменяется.

Регистр количества передаваемых и принимаемых данных контроллера ПДП SSPDMACNT

Таблица 5.226 - Формат регистра SSPDMACNT

Биты	Название	Тип	Описание
[31:0]	SSDMA_CNT	ЧТ/ЗП	Количество передаваемых и принимаемых данных.

Количество передаваемых и принимаемых данных задается в байтах. В процессе выполнения процедуры ПДП значение регистра не изменяется.

Регистр конфигурации режима приема контроллера ПДП SSPDMARXDIS

Таблица 5.227 - Формат регистра SSPDMARXDIS

Биты	Название	Тип	Описание
[31:1]	-	-	Зарезервирован, должен быть прописан 0.
[0]	SSDMA_RXDIS	ЧТ/ЗП	Управление записью в память в режиме ПДП: 0 - запись производится; 1 - запись не производится.

Регистр управления контроллером ПДП порта SSPDMACTR

Таблица 5.228 - Формат регистра SSPDMACTR

Биты	Название	Тип	Описание
[31:1]	-	-	Зарезервирован, должен быть прописан 0
[0]	SSDMA_CTR	ЧТ/ЗП	Управление процедурой ПДП: 0 - процедура ПДП не выполняется; 1 - процедура ПДП выполняется.

Установка регистра управления SSPDMACTR в 1 запускает выполнение процедуры ПДП. После завершения процедуры ПДП данный разряд сбрасывается аппаратно. Пользователь может остановить процедуру ПДП, сбросив данный разряд программно.

Регистр состояния контроллера ПДП порта SSPDMASTS

Таблица 5.229 - Формат регистра SSPDMASTS

Биты	Название	Тип	Описание
[31:2]	-	-	Зарезервировано.
[1]	SSDMA_TXSTS	ЧТ	0 - передающая часть контроллера ПДП остановлена; 1 - передающая часть контроллера ПДП работает.
[0]	SSDMA_RXSTS	ЧТ	0 - приемная часть контроллера ПДП остановлена; 1 - приемная часть контроллера ПДП работает.

Регистр маски прерывания от контроллера ПДП порта SSPDMAIM

Таблица 5.230 - Формат регистра SSPDMAIM

Биты	Название	Тип	Описание
[31:1]	-	-	Зарезервировано, должны быть прописаны все нули.
[0]	SSDMA_IM	ЧТ/ЗП	Маска прерывания от контроллера ПДП: 0 = прерывание замаскировано; 1 = прерывание не маскировано.

Регистр запроса на прерывания от контроллера ПДП порта SSPDMAIR

Таблица 5.231 - Формат регистра SSPDMAIR

Биты	Название	Тип	Описание
[31:1]	-	-	Зарезервировано, должны быть прописаны все нули.
[0]	SSDMA_IR	ЧТ/ЗП	Запрос на прерывание от контроллера ПДП.

Запрос на прерывание устанавливается в регистре SSPDMAIR при аппаратном окончании процедуры ПДП. Если процедура ПДП остановлена программистом запрос на прерывание не возникает.

Регистр упаковки данных контроллера ПДП порта SSPDMAPT

Таблица 5.65 - Формат регистра SSPDMAPT

Биты	Название	Тип	Описание
[31:1]	-	-	Зарезервировано, должны быть прописаны все нули.
[0]	SSDMA_PT	ЧТ/ЗП	Режим упаковки/распаковки данных: 0 = 8 разрядные слова; 1 = 16 разрядные слова.

Регистр выбора ведомого устройства порта SSPDMACS

Таблица 5.232 - Формат регистра SSPDMACS

Биты	Название	Тип	Описание
[31:4]	-	-	Зарезервировано, должны быть прописаны все нули.
[3]	SSDMA_CS3	ЧТ/ЗП	Активное устройство подключено к SPI_CS3.
[2]	SSDMA_CS2	ЧТ/ЗП	Активное устройство подключено к SPI_CS2.
[1]	SSDMA_CS1	ЧТ/ЗП	Активное устройство подключено к SPI_CS1.
[0]	SSDMA_CS0	ЧТ/ЗП	Активное устройство подключено к SPI_CS0 (данное состояние устанавливается после системного сброса).

Путем установки в 1 соответствующего разряда выбирается какой из выводов SPI_CS0 - SPI_CS3 будет активным при передаче данных по интерфейсу SPI .

5.14.3 Протокол передачи данных в различных режимах работы SPI интерфейса

Контроллер последовательного порта поддерживает следующие форматы обмена данными с внешними устройствами:

- спецификации фирмы Motorola (Motorola SPI);
- спецификации фирмы Texas Instruments (Texas Instruments SPI);
- спецификации фирмы National Semiconductors (National Semiconductors Microwire).

Формат передачи данных задается программистом путем определения полей SPO, SPH и FRF регистра управления 0 контроллера (SSPCR0).

5.14.3.1 Формат передачи данных спецификации Motorola SPI

Работа SPI интерфейса при данном формате передачи данных возможна в 4 режимах (mode 0,1,2 и 3), отличающихся друг от друга состоянием сигнала SPICLK в неактивном состоянии и активным фронтом данного сигнала. Режим работы задается путем записи соответствующих значений в биты SPO и SPH регистра SSPCR0 контроллера.

Бит SPO управляет состоянием выхода SPICLK в неактивном состоянии. Если SPO = 0, то вывод SPICLK имеет низкий уровень в отсутствии передачи данных. Если SPO = 1, то вывод SPICLK имеет высокий уровень в отсутствии передачи данных.

Бит SPH управляет активным фронтом 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.

При работе в формате передачи данных Motorola SPI в любом режиме (0,1,2 или 3) сигнал SPI_CSi выполняет функции выборки устройства, с которым работает интерфейс. Активный уровень сигнала SPI_CSi – низкий. Активный уровень данного сигнала выставляется на весь период обмена данными

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

На рисунках 5.29 – 5.30 представлены временные диаграммы обмена по порту SPI в режиме Motorola SPI mode 0 при передаче одного слова данных и состоящей из нескольких пакетов данных.

Выдача данных происходит по срезу сигнала SPICLK, защелкивание входных данных – по фронту сигнала SPICLK. В промежутках между пакетами (словами) вывод SPI_CSx принимает значение 1 на протяжении полутакта синхросигнала SPICLK.

Рисунок 5.29 - Motorola SPI mode 0 (SPO=0, SPH=0) одиночная передача

Рисунок 5.30- Motorola SPI mode 0 (SPO=0, SPH=0) передача из нескольких пакетов

Motorola SPI mode 1 (SPO=0, SPH=1)

На рисунке 5.31 представлены временные диаграммы обмена по порту SPI в режиме Motorola SPI mode 1.

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.

Рисунок 5.31 - Motorola SPI mode 1 (SPO=0, SPH=1) одиночная передача

Motorola SPI mode 2 (SPO=1, SPH=0)

На рисунках 5.32 – 5.33 представлены временные диаграммы обмена по порту SPI в режиме Motorola SPI mode 2 при передаче одного слова данных и состоящей из нескольких пакетов данных.

Выдача данных происходит по фронту сигнала SPICLK, защелкивание входных данных – по срезу сигнала SPICLK. Между пакетами (словами) вывод SPI_CSx принимает значение 1 на протяжении полутакта синхросигнала SPICLK.

Рисунок 5.32 - Motorola SPI mode 2 (SPO=1, SPH=0) одиночная передача

Рисунок 5.33 - Motorola SPI mode 2 (SPO=1, SPH=0) передача из нескольких пакетов

Motorola SPI mode 3 (SPO=1, SPH=1)

На рисунке 5.34 представлены временные диаграммы обмена по порту SPI в режиме Motorola SPI mode 3.

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.

Рисунок 5.34 - Motorola SPI mode 3 (SPO=1, SPH=1) одиночная передача

5.14.3.2 Формат передачи данных спецификации Texas Instruments SPI

На рисунках 5.35-5.36 представлены временные диаграммы обмена по порту SPI в режиме Texas Instruments SPI при передаче одного слова данных и состоящей из нескольких пакетов данных.

При работе в формате передачи данных Texas Instruments SPI сигнал выбор устройства, с которым работает интерфейс, осуществляется путем выдачи на вывод SPI_CSi сигнала высокого уровня длительностью один такт. При последовательном обмене

несколькими словами выдача на вывод SPI_CSi сигнала высокого уровня происходит в момент выдачи последнего бита предыдущего слова данных.

Работа интерфейса в данном режиме не зависит от состояния полей SPO и SPH регистра SSPCR0 контроллера.

Рисунок 5.35 - Texas Instruments SPI одиночная передача

Рисунок 5.36 - Texas Instruments SPI передача из нескольких пакетов

5.14.3.3 Формат передачи данных спецификации National Semiconductors Microwire

На рисунках 5.37-5.38 представлены временные диаграммы обмена по порту SPI в режиме National Semiconductors Microwire при передаче одного слова данных и состоящей из нескольких пакетов данных.

Рисунок 5.37 - National Semiconductors Microwire одиночная передача

Формат передачи данных National Semiconductors Microwire отличатся от формата передачи SPI тем, что вместо полнодуплексной передачи данных происходит полудуплексная, т.е. передатчик сначала выдает управляющую информацию – 8 бит, а затем принимает данные в количестве, определяемом состоянием поля DSS регистра управления 0 контроллера SSPCR0. В качестве управляющей информации всегда берется младший байт 16-разрядного слова, записываемого в буфер передаваемых данных (регистр SSPDR) при инициализации передачи.

Рисунок 5.38 - National Semiconductors Microwire передача из нескольких пакетов

5.14.4 Работы контроллера ПДП интерфейса

Контроллер ПДП предназначен для обмена данными между внешними устройствами, подключенными к внешним выводам интерфейса и памятью без участия процессора. Контроллер ПДП программируется по 32-разрядной шине AMBA APB спецификации 3.0. Обмен данными в процессе выполнения процедуры ПДП производится по 32-разрядной шине AMBA AXI спецификации 3.0, к которой контроллер ПДП подключен в качестве мастера.

Так как обмен по интерфейсу при любом формате обмена является дуплексным (полудуплексным), т.е. одновременно идет и передача и прием данных, в контроллере ПДП нет разделения на процедуры чтения данных из памяти и записи данных в память. Эти оба процесса выполняются в рамках одной процедуры ПДП.

Перед запуском процедуры ПДП необходимо убедиться, что при инициализации контроллера интерфейса было разрешено использование режима ПДП на прием и на передачу, т.е. поля TXDMAE и RXDMAE регистра SSPDMACR установлены в лог. 1. Кроме этого необходимо убедиться, что в приемном буфере не осталось данных от предыдущих обращений к интерфейсу. Если предыдущее обращение к интерфейсу осуществлялось в режиме ПДП, то очистку приемного буфера выполняет сам контроллер ПДП.

При запуске процедуры ПДП пользователь определяет адреса массива передаваемых и принимаемых данных и размер передачи в регистрах SSPDMATXADR, SSPDMARXADR и SSPDMACNT соответственно. Адреса массивов данных в памяти должны быть выровнены на границу 32-разрядного слова. (Биты 1 и 0 адресов должны быть равны лог. 0). Размер передачи задается в байтах и может быть произвольным.

Старт процедуры ПДП осуществляется установкой в лог 1 поля SSDMA_CTR регистра управления контроллером ПДП SSPDMACTR. После старта значения адресов и размер массива данных переписываются в адресные счетчики и счетчики данных и при дальнейшей работе не изменяются. Контроллер ПДП считывает данные из памяти, распаковывает их и записывает в буфер передаваемых данных контроллера интерфейса. По мере готовности данных в буфере принимаемых данных контроллера ПДП считывает их из буфера и после упаковки в 32-разрядное слова записывает в память процессора.

Упаковка и распаковка данных зависит от состояния поля SSDMA_PT регистра упаковки данных SSPDMAPT. При нулевом значении данного поля считанное из памяти процессора 32-разрядное слово данных преобразуется в 4 16-разрядных слова, старший байт которых равен нулю, а младший байт имеет значение соответствующего байта считанного слова. Эти 4 слова последовательно записываются в буфер передаваемых данных контроллера интерфейса. Соответственно, у 4 последовательно считанных из буфера передаваемых данных 16-разрядных слов отбрасываются старшие байты, а младшие байты последовательно упаковываются в 32-разрядное слово, которое записывается в память

процессора. Такая упаковка не приводит к потере данных, только если значение поля DSS регистра конфигурации 0 контроллера SSPCR0 не превышает 0111b.

При единичном значении поля SSDMA_PT считанное 32-разрядное слово данных преобразуется в 2 16-разрядных слова, которые последовательно записываются в буфер передаваемых данных. Соответственно, два последовательно считанных из буфера передаваемых данных 16-разрядных слова упаковываются в одно 32-разрядное слово, записываемое в память процессора.

Если данные, считанные из внешнего устройства, пользователю не нужны, т.е. происходит только запись во внешнее устройство, то запись считанных данных в память процессора можно запретить, установив в лог. 1 поле SSDMA_RXDIS регистра конфигурации режима приема SSPDMARXDIS. В этом случае чтение данных из буфера принимаемых данных осуществляется, но запись в память процессора блокируется.

Состояние контроллера ПДП отражается в регистре состояния контроллера ПДП SSPDMASTS. Поля регистра устанавливаются и сбрасываются аппаратно. Процедура ПДП окончена, когда оба поля регистра сброшены в состояние лог. 0.

После окончания процедуры ПДП вырабатывается запрос на прерывание, который фиксируется в регистре запросов на прерывание SSPDMAIR. Данное прерывание может быть замаскировано установкой маски прерывания в регистре SSPDMAIM. Запрос на прерывание в регистре SSPDMAIR должен быть сброшен программно.

Если процедура ПДП останавливается путем сброса поля SSDMA_CTR регистра управления контроллером ПДП SSPDMACTR, то сразу прекращается считывание из памяти процессора записываемых во внешнее устройство данных. После записи в память процессора последнего считанного из внешнего устройства слова данных аппаратно сбрасываются соответствующие поля регистра состояния SSPDMASTS. Однако запрос на прерывание в данном случае не формируется.

5.14.5 Прерывания контроллера SPI

Контроллер формирует 2 запроса на прерывание.

DMAINTR - запрос прерывание по окончанию процедура ПДП. Прерывание маскируемое. Запрос устанавливается по окончанию процедуры ПДП по обнулению счетчика размера передачи. Сброс запроса на прерывание должен производиться программно.

SSPINTR - обобщенный запрос от контроллера интерфейса SPI. Данный запрос является комбинацию нижеперечисленных запросов на прерывание:

- **SSPRXINTR**. Запрос на прерывание по готовности принимаемых данных. Данный запрос устанавливается, когда буфер принимаемых данных содержит четыре или более слов.
- **SSPTXINTR**. Запрос на прерывание по готовности буфера передаваемых данных к приему данных. Данный запрос устанавливается, когда буфера передаваемых данных содержит менее четырех слов.
- **SSPRORINTR.** Запрос на прерывание по переполнению буфера принимаемых данных. Данный запрос устанавливается, когда буфера принимаемых данных заполнен. Прием данных продолжается, приводя к потере данных в приемном сдвиговом регистре.
- **SSPRTINTR.** Запрос на прерывание по истечении времени ожидания приема данных. Этот запрос устанавливается, когда принятые данные находятся в буфере принимаемых данных дольше 32 тактов. Этот механизм дополнительно оповещает программиста о том, что необходимо забрать данные из буфера. Прерывание SSPRTINTR снимается, когда буфер принимаемых данных становится пустым, или если новые данные получены с SPI шины. Прерывание также может быть снято путем записи бита RTIC=1 в регистре SSPICR.

Каждый из четырех запросов на прерывание может быть замаскирован путем установления соответствующих масок в регистре SSPIMSC. Состояние запросов на прерывания до и после наложения маски можно узнать, прочитав SSPRIS и SSPMIS регистры.

5.15 Сторожевой таймер WDT

Сторожевой таймер служит индикатором попадания процессора в состояние “зависания” и сброса процессора в случае обнаружения данной ситуации. Управление блоком сторожевого таймера осуществляется по шине AMBA APB версии 2.0. Блок состоит из программируемого 32-разрядного таймера, работающего на тактовой частоте шины APB, схемы управления режимами работы таймера и схемы формирования контрольных и управляющих сигналов.

5.15.1 Организация работы сторожевого таймера

После снятия сигнала системного сброса, блок находится в выключенном состоянии. Если возникла необходимость воспользоваться сторожевым таймером, то программным образом надо задать интервал счета таймера, формируя тем самым квант времени работы сторожевого таймера. После записи интервала счета счетчик начинает декрементироваться.

Когда счетчик достигает нулевого значения, генерируется сигнал прерывания WDOGINT. Данный сигнал прерывания заводится на контроллеры прерываний обоих процессорных систем. Одновременно в счетчик заносится начальное значение интервала счета и декремент счетчика возобновляется. Если до достижения счетчиком нулевого значения, прерывание WDOGINT не будет обработано (сброшено), то будет сформирован сигнал WDOGRST, на основе которого внешним образом можно сгенерировать сигнал системного сброса процессора.

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

5.15.2 Программно доступные регистры блока сторожевого таймера

Программно доступные регистры блока сторожевого таймера WDT расположены в области памяти периферийных устройств REG_CCPU, и имеют начальный адрес WDT Base = 0x000C_E000 hex и общий размер 4 КБайт. Спецификация регистров блока представлена в таблице 5.233.

Таблица 5.233 - Спецификация регистров блока WDT

Адрес (байтовый доступ)	Тип	Раз-ряд-ность	Начальное значение	Имя	Описание
0x000C_E000 hex	ЧТ/ЗП	32	0xFFFFFFFF	WdogLoad	Регистр загрузки.
0x000C_E004 hex	ЧТ	32	0xFFFFFFFF	WdogValue	Регистр значения.
0x000C_E008 hex	ЧТ/ЗП	2	0x0	WdogControl	Регистр управления.
0x000C_E00C hex	ЗП	-	-	WdogIntClr	Регистр сброса прерываний.
0x000C_E010 hex	ЧТ	1	0x0	WdogRIS	Регистр состояния прерывания до наложения маски.
0x000C_E014 hex	ЧТ	1	0x0	WdogMIS	Регистр состояния прерывания после наложения маски.
0x000C_E018 hex - 0x000C_EBFF hex	-	-	-	-	Зарезервировано
0x000C_EC00 hex	ЧТ/ЗП	32	0x0	WdogLock	Регистр блокировки.
0x000C_EC04 hex - 0x000C_EEFF hex	-	-	-	-	Зарезервировано
0x000C_EF00 hex	ЧТ/ЗП	1	0x0	WdogITCR	Регистр управления тестом интеграции.
0x000C_EF04 hex	ЗП	2	0x0	WdogITOP	Регистр управления тестом интеграции.
0x000C_EF08 hex - 0x000C_EFDC hex	-	-	-	-	Зарезервировано.
0x000C_EFE0 hex	ЧТ	8	0x05	WdogPeriphID0	Идентификатор периферийного устройства.
0x000C_EFE4 hex 0x3F9	ЧТ	8	0x18	WdogPeriphID1	Идентификатор периферийного устройства.
0x000C_EFE8 hex	ЧТ	8	0x14	WdogPeriphID2	Идентификатор периферийного устройства.
0x000C_EFEC hex	ЧТ	8	0x00	WdogPeriphID3	Идентификатор периферийного устройства.
0x000C_EFF0 hex	ЧТ	8	0x0D	WdogPCellID0	Идентификатор PrimeCell.
0x000C_EFF4 hex	ЧТ	8	0xF0	WdogPCellID1	Идентификатор PrimeCell.
0x000C_EFF8 hex	ЧТ	8	0x05	WdogPCellID2	Идентификатор PrimeCell.
0x000C_EFFC hex	ЧТ	8	0xB1	WdogPCellID3	Идентификатор PrimeCell.

WdogLoad - регистр загрузки -32-битный регистр, доступный по записи и по чтению. Данный регистр хранит значение интервала счета, с которого счетчик начинает декрементироваться. Если значение регистра WdogLoad установлено в нуль, то прерывания сторожевым таймером вырабатываются незамедлительно. Если содержимое регистра отлично от нуля, то это значение загружается в счетчик незамедлительно после записи регистра.

WdogValue – регистр значения счетчика - 32-битный регистр, доступный только на чтение,. При чтении этого регистра выдается текущее значение декрементирующего счетчика.

WdogControl - регистр управления - 2-битный регистр, доступный на запись и чтение, который разрешает программному обеспечению (ПО) управлять сторожевым таймером. В таблице 5.234 приведен формат регистра WdogControl.

Таблица 5.234 - Формат регистра WdogControl

Биты	Название	Тип	Выполняемая функция
[31:2]	-	-	Зарезервировано.
[1]	RESEN	ЧТ/ЗП	Разрешающий сигнал блока WDOG для сброса выхода, WDOGRST. Работает как маска для сброса выхода. Если установлен высокий уровень, то разрешен сброс, если низкий, то сброс неактивен.
[0]	INTEN	ЧТ/ЗП	Сигнал, разрешающий прерывание события, WDOGINT. Если установлен высокий уровень, то счетчик находится в активном состоянии и разрешены прерывания, если установлен низкий уровень, то счетчик и прерывания неактивны. Нужно перезагрузить счетчик значением, которое хранится в WdogLoad, если прерывание раннее было неактивным, но затем стало разрешенным.

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

WdogRIS - регистр состояния прерывания до наложения маски Данный регистр определяет состояние прерываний, формируемых счетчиком, до наложения маски,. Регистр WdogRIS возводится в единицу в момент, когда счетчик сторожевого таймера достигает нуля. В таблице 5.235 приведен формат регистра WdogRIS.

Таблица 5.235 - Формат регистра WdogRIS

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

WdogMIS - регистр состояния прерывания после наложения маски Данный регистр определяет состояние прерывания счетчика после наложения маски. Это состояние формируется логическим функцией «И» между битом WDOGRIS и битом INTEN регистра управления. В таблице 5.236 приведен формат регистра WdogMIS.

Таблица 5.236 - Формат регистра WdogMIS

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

WdogLock - регистр блокировки записи. Запись любого значения кроме 0x1ACCE551 в данный регистр блокирует программную запись во все другие регистры блока. Запись значения 0x1ACCE551 разблокирует программную запись во все регистры блока. Таким

образом, можно защитить регистры WDOG от некорректно работающего программного обеспечения. Чтение из этого регистра возвращает состояние блокировки:

0 – доступ по записи разрешен (нет блокировки);

1 – доступ по записи запрещен (заблокирован).

В таблице 5.237 приведен формат регистра WdogLock.

Таблица 5.237 - Формат регистра WdogLock

Биты	Название	Тип	Выполняемая функция
[31:0]	WDOGLOCK	ЧТ/ЗП	Запись значения 0x1ACCE551 в этот регистр разрешает запись во все регистры. Запись любого другого значения делает невозможным доступ по записи ко всем регистрам. Чтение возвращает состояние блокировки: 0x00000000 – запись во все регистры разрешена; 0x00000001 – запись во все регистры запрещена.

WdogPeriphID0-3 – Идентификатор периферийного устройства Регистр TimerPeriphID0-3 представляет четыре 8-битных регистра, предназначенных только для чтения, которые охватывают адресное пространство от 0xFE0 до0xFEC. Регистры могут быть концептуально представлены как 32-битный регистр. В таблице 5.238 приведен формат этого регистра.

Таблица 5.238 - Формат регистра WdogPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для DIT 0x805
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока. Номер ревизии начинается с нуля
Configuration[31:24]	Является вариантом конфигурации периферии. Для DIT 0x0

Рисунок 5.39 - Формат регистра WdogPeriphID0-3

Отметим, что при программировании важно помнить, что периферийные блоки занимают 4 Кбайт адресного пространства. Обращение к периферийным регистрам должно быть 32-битным, независимо от их реальной разрядности.

WdogPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, программно доступные только по чтению, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. Регистр WdogPCellID установлен в 0xB105F00D. На рисунке 5.40 приведен формат этого регистра.

Рисунок 5.40 - Формат регистра WdogPCellID0-3

WdogITCR - регистр управления тестом интеграции. Данный однобитный регистр используется для активации тестового режима проверки правильности интеграции периферийного блока в систему – далее «теста интеграции». Когда используется этот режим, прерывание WDOGINT и сигнал сброса WDOGRST контролируются регистром WdogITOP. Формат регистра приведен в таблице 5.239.

Таблица 5.239 - Формат регистра WdogITCR

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервированы.
[0]	ITEN	ЧТ/ЗП	Разрешение режима «теста интеграции». Когда этот бит установлен в 1, WDOG переходит в режим теста интеграции, иначе находится в нормальном режиме.

WdogITOP - регистр управления выходными сигналами В режиме теста интеграции, на выходные сигналы прерывания подаются соответствующие разряды данного регистра. Формат регистра приведен в таблице 5.240.

Таблица 5.240 - 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.

5.16 Контроллер измерения температуры кристалла с температурным датчиком Temp Sens

Контроллер измерения температуры кристалла предназначен для формирования тактового сигнала датчика измерения температуры, управления датчиком и оценки результатов измерения в режиме реального времени. Измерение температуры производится в диапазоне от -40°С до +125°С с точностью ±5°С.

Температурный датчик реализован на температурно-чувствительной ячейке, аналоговый выход которой непрерывно оцифровывается АЦП последовательного приближения с частотой оцифровки не более 50 кГц.

После фиксации значения температуры на выходе АЦП контроллер производит сравнение полученного значения с порогом, задаваемым программно. В случае превышения порога формируется прерывание по превышению температуры, которое может быть обработано стандартным образом.

Управление контроллером осуществляется по 32-разрядной шине AMBA APB спецификации 3.0. Работа блока контроллера тактируется тактовым сигналом шины APB.

5.16.1 Программно доступные регистры контроллера измерения температуры

Программно доступные регистры контроллера измерения температуры расположены в области памяти периферийных устройств, имеют базовый адрес TS Base = 0x000D_1000 и общий размер 4 Кб. Спецификация регистров представлена в таблице 5.241.

Таблица 5.241 - Спецификация регистров контроллера измерения температуры

Адрес (байтовый доступ)	Тип	Разряд-ность	Началь-ное значение	Имя	Описание
0x000D_1000h	ЧТ/ЗП	1	0x1	TS_PWDN	Регистр управления режимом пониженного потребления
000D_1004h	ЧТ/ЗП	1	0x0	TS_START	Регистр управления режимом измерения температуры
000D_1008h	ЧТ/ЗП	1	0x0	TS_IR	Регистр запроса на прерывание по превышению температуры.
000D_100Ch	ЧТ/ЗП	1	0x0	TS_IM	Регистр маски прерываний температурного датчика.
000D_1010h	ЧТ/ЗП	12	0xFFF	TS_CLK_DIV	Регистр делителя частоты АЦП температурного датчика.
000D_1014h	ЧТ/ЗП	12	0xFFF	TS_LEVEL	Регистр порога температуры.
000D_1018h	ЧТ/ЗП	12	0x000	TS_DATA	Регистр текущего значения температуры.
000D_101Ch	ЧТ/ЗП	1	0x0	TS_ENZC	Регистр управления режимом калибровки датчика.
000D_1020h - 000D_1FFFh	-	-	-	-	Зарезервировано

Регистр **TS_PWDN** –регистр управления режимом пониженного потребления. Регистр доступен по чтению и записи и управляет режимом пониженного потребления датчика измерения температуры. В таблице 5.242 приведен формат регистра TS_PWDN.

Таблица 5.242 - Формат регистра TS_PWDN

Биты	Название	Тип	Описание
[0]	PWDN	ЧТ/ЗП	0 – Датчик температуры находится в рабочем режиме; 1 - Датчик температуры находится в режиме пониженного потребления.

Регистр **TS_START** –регистр управления режимом измерения температуры. Регистр доступен по чтению и записи и управляет режимом измерения температуры контроллера. В таблице 5.243 приведен формат регистра TS_START.

Таблица 5.243 - Формат регистра TS_START

Биты	Название	Тип	Описание
[0]	START	ЧТ/ЗП	0 – контроллер находится в пассивном режиме; 1 – контроллер находится в режиме измерения температуры.

Регистр **TS_IR** - регистр запроса на прерывание по превышению температуры. Регистр доступен по чтению и по записи. В таблице 5.244 приведен формат регистра TS_IR.

Таблица 5.244 - Формат регистра TS_IR

Биты	Название	Тип	Описание
[0]	IR	ЧТ/ЗП	Запрос на прерывание по превышению температуры

Регистр **TS_IM** - регистр маски запроса на прерывание по превышению температуры. Регистр доступен по чтению и по записи. В таблице 5.245 приведен формат регистра TS_IM.

Таблица 5.245 - Формат регистра TS_IM

Биты	Название	Тип	Описание
[0]	IM	ЧТ/ЗП	Маска прерывания по превышению температуры: 0 – маска установлена; 1 – маска сброшена.

Регистр **TS_CLK_DIV** – 12-битный регистр делителя частоты АЦП температурного датчика. Регистр доступен по чтению и по записи. Формат регистра представлен в таблице 5.246.

Таблица 5.246 - Формат регистра TS_CLK_DIV

Биты	Название	Тип	Описание
[11:0]	CLK_DIV	ЧТ/ЗП	Значение делителя частоты.

Тактовый сигнал, подаваемый на АЦП, формируется из тактового сигнала работы контроллера (т.е. тактового сигнала шины APB) путем деления на коэффициент, задаваемый в регистре TS_CLK_DIV. Частота тактового сигнала, подаваемого на АЦП, вычисляется по формуле:

$$F_{АЦП} = \frac{Fapb}{2*(N+1)}$$, где

$F_{АЦП}$ - тактовая частота АЦП датчика температуры,

Fapb - тактовая частота работы контроллера (тактовая частота шины APB),

N - значение, задаваемое в регистре TS_CLK_DIV.

Частота тактового сигнала, подаваемого на АЦП температурного датчика не должна превышать 50 кГц.

Регистр **TS_LEVEL** – регистр порога температуры. Регистр доступен по чтению и по записи. В Таблица 5.247 приведен формат регистра TS_LEVEL.

Таблица 5.247 - Формат регистра TS_LEVEL

Биты	Название	Тип	Описание
[11:0]	LEVEL	ЧТ/ЗП	Значение порога температуры, с которым будет сравниваться выход АЦП температурного датчика.

Регистр **TS_DATA** - регистр текущего значения температуры. Регистр доступен по чтению и по записи. В Таблица 5.248 приведен формат регистра TS_DATA.

Таблица 5.248 - Формат регистра TS_DATA

Биты	Название	Тип	Описание
[11:0]	DATA	ЧТ/ЗП	Последний отсчет значения температуры с выхода АЦП.

Регистр **TS_ENZC** –регистр управления режимом калибровки. Регистр доступен по чтению и записи и управляет режимом калибровки датчика. В Таблица 5.249**Ошибка! Источник ссылки не найден.** приведен формат регистра TS_ENZC.

Таблица 5.249 - Формат регистра TS_ENZC

Биты	Название	Тип	Описание
[0]	ENZC	ЧТ/ЗП	0 – контроллер находится в рабочем режиме; 1 – контроллер находится в режиме калибровки.

5.16.2 Работа с контроллером измерения температуры кристалла

Процедура активизации датчика измерения температуры на кристалле должны происходить следующим образом:

Сначала необходимо задать тактовую частоту работы АЦП, записав значение делителя частоты в регистр TS_CLK_DIV. Значение делителя частоты должно быть таким, чтобы частота тактового сигнала АЦП не превышала 50 кГц.

Далее в регистре TS_LEVEL задается порог температуры, с которым будет сравниваться измеряемое значение температуры на выходе АЦП. Соответствие значения температуры на кристалле коду на выходе АЦП представлено в Таблица 5.250.

Таблица 5.250 - Соответствие измеренной температуры кристалла коду на выходе АЦП

Температура °С	Код на выходе АЦП	Температура °С	Код на выходе АЦП	Температура °С	Код на выходе АЦП
- 40	3795	+ 20	3683	+ 80	3543
- 35	3787	+ 25	3673	+ 85	3529
- 30	3779	+ 30	3663	+ 90	3515
- 25	3769	+ 35	3651	+ 95	3501
- 20	3761	+ 40	3641	+ 100	3487
- 15	3751	+ 45	3629	+ 105	3471
- 10	3743	+ 50	3617	+ 110	3457
- 5	3733	+ 55	3605	+ 115	3441
0	3723	+ 60	3593	+ 120	3425
+ 5	3713	+ 65	3581	+ 125	3409
+ 10	3703	+ 70	3569	-	-
+ 15	3693	+ 75	3555	-	-

После установки порога АЦП датчика температуры выводится из режима пониженного потребления, путем записи лог. 0 в регистр TS_PWDN. Измерение температуры начинается при установке в лог. 1 регистра TS_START.

Текущее значение температуры можно посмотреть, программно считав значение кода из регистра TS_DATA. Первое измеренное значение записывается в регистр TS_DATA через семь тактов частоты работы АЦП датчика температуры после перевода датчика из состояния пониженного потребления в рабочий режим.

5.16.3 Прерывания контроллера измерения температуры на кристалле

Контроллер формирует следующий запрос на прерывание:

- TSINT - запрос прерывание по превышению температуры на кристалле порога, установленного в регистре TS_LEVEL. Прерывание маскируемое. Запрос на прерывание фиксируется в регистре TS_IR независимо от состояния маски. Сброс запроса на прерывание должен производиться программно.

5.17 Загрузочное ПЗУ BROM и начальная загрузка микросхемы 1879ВМ8Я

Микросхема 1879ВМ8Я имеет банк памяти ПЗУ объёмом 8 КБайт, который содержит программу начального загрузчика.

Начальный загрузчик – это программа, которая начинает выполняться после снятия внешнего сигнала сброса и инициализации схемы ФАПЧХ внутри СБИС 1879ВМ8Я. Задача этой программы загрузить во внутреннюю память процессора ARM Cortex-A5 центрального управляющего узла следующую исполняемую программу и передать ей управление. Этой следующей программой может быть, например, загрузчик операционной системы.

Начальная загрузка микросхемы возможна через интерфейсы:

- SPI – в этом случае начальный загрузчик самостоятельно извлекает образ из флэш-памяти, доступной по интерфейсу SPI,
- PCIe или Ethernet (Ethernet Debug Communication Link, EDCL) – в этом случае образ программы должен быть загружен внешним хост-контроллером через соответствующий интерфейс.

Выбор режима загрузки осуществляется начальным загрузчиком в зависимости от состояния внешних входов BOOTM2, BOOTM1, BOOTM0 в соответствии с таблицей 5.251.

Таблица 5.251 – Варианты начальной загрузки микросхемы 1879ВМ8Я

BOOTM2	BOOTM1	BOOTM0	Режим загрузки
0	0	0	Загрузка по SPI.
0	0	1	Загрузка только по Ethernet.
0	1	0	Загрузка только по PCIe.
0	1	1	Загрузка по Ethernet или PCIe (ожидается из любого источника).
1	0	0	Загрузка по SPI с индикацией на GPIOC.
1	0	1	Загрузка только по Ethernet с индикацией на GPIOC.
1	1	0	Загрузка только по PCIe с индикацией на GPIOC.
1	1	1	Зарезервировано (тестовый режим).

Код начального загрузчика всегда доступен для чтения по адресу 0x000C4000 управляющего процессорного узла CCPU. Однако, после начального сброса микросхемы содержимое ПЗУ копируется во внутренний банк памяти CCMB1 по адресу 0x00030000, исполнение кода начального загрузчика производится далее из этого банка.

Блок-схема алгоритма начального загрузчика показана ниже (см. рисунок 5.41).

Рисунок 5.41 – Блок-схема алгоритма начального загрузчика

Образ загружаемой программы в случае загрузки по SPI помещается в банк CCMB0 по адресу 0x00000000. При загрузке хост-контроллером программа может быть расположена в любом месте кроме адресов 0x00030000 - 0x0003FFFF банка CCMB1 (где работает начальный загрузчик).

Независимо от способа загрузки

- выполнение загруженной программы начинается с адреса 0x00000000,
- в момент старта загружаемой программы процессор находится в режиме супервизора (CPSR[4:0]=b10011).

Следует обратить внимание, что внутренний банк памяти CCMB0 виден хост-контроллеру по адресам с базовым смещением 0x60000000.

5.17.1 Загрузка хост-контроллером

В режиме загрузки хост-контроллером начальный загрузчик настраивает интерфейсы Ethernet и/или PCIe таким образом, что обеспечивает хост-контроллеру произвольный доступ в адресное пространство микросхемы

- по Ethernet – специальные пакеты протокола EDCL (п. 5.11.7) преобразуются аппаратно в транзакции чтения и записи в адресном пространстве микросхемы,
- по PCIe – загрузчик настраивает 2 окна по 2 ГБайта BAR0 и BAR1 таким образом, что всё адресное пространство микросхемы (4 ГБайта) становится доступным хост-контроллеру.

Предполагаемый сценарий работы хост-контроллера:

- загрузить программу во внутреннюю память микросхемы с помощью произвольного доступа через PCIe или EDCL,
- записать ненулевое значение по адресу 0x00030000 (или 0x60030000 с точки зрения хост-контроллера).

Поскольку хост-контроллер имеет произвольный доступ к ресурсам микросхемы, то он не обязан следовать приведённому выше сценарию. Допускается, например, напрямую загружать другие процессорные ядра микросхемы или инициализировать внешнюю память DDR и только потом загружать программу.

5.17.2 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 FHPMP/8 = 10 МГц, т.к. начальный делитель опорного синхросигнала, выставляемый загрузчиком – 20,
- поддержка команды FAST_READ (код – 0x0B),
- загрузочная flash-память должна быть подключена к выводу SPI_CS0,
- максимальный объем flash-памяти 16Мбайт.

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.

Последовательность загрузки через порт SPI

Двоичный образ загружаемой программы в процессе загрузки помещается во внутреннюю память микросхемы. При загрузке по SPI двоичный образ загружаемой программы должен содержать

- в последнем слове CRC32-свертку всех остальных слов,
- в шестом слове служебное слово загрузчика.

Схематическое представление двоичного образа загружаемой программы приведено на рисунке ниже (см. рисунок 5.42).

Рисунок 5.42 – Образ загружаемой программы

Загрузка через SPI выполняется в два этапа. На первом этапе SPI инициализируется с делителем частоты равным 20, 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 (см. рисунок 5.43). Проверяется содержимое служебного слова, и если оно правильное, выполняется второе чтение. В противном случае происходит аварийное прекращение загрузки.

31 … 24	23 … 16	15 … 0
Делитель частоты SPI (четное число от 4 до 254)	Магический код (0xA5)	Размер загружаемой программы (до 0xFFFF 32-х разрядных слов)

Рисунок 5.43 – 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.

5.17.3 Использование GPIOC для индикации состояния начального загрузчика

Использование выводов порта общего назначения центрального управляющего ПУ (GPIOC) для индикации состояния начального загрузчика зависит от состояния внешнего входа микросхемы BOOTM2. Если BOOTM2 = 0, то загрузчик конфигурирует все разряды GPIO как входы и индикации состояния начального загрузчика не происходит.

Если BOOTM2 = 1, то

- в момент старта загрузчик конфигурирует разряды GPIOC [7:0] как выходы.

Остальные разряды GPIOC конфигурируются как входы.

- значение 0x00 на GPIOC[7:0] говорит о том, что процесс загрузки стартовал.
- значение 0xC0 говорит об успешном завершении загрузки
- значение 0xCe говорит об ошибке загрузки, где e – код ошибки.

Возможные коды ошибки загрузки:

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 при загрузке по SPI.

5.17.4 Генераторы тактовых сигналов и сигналов сброса CRGL0…CRGL3

Данные генераторы предназначены для управления со стороны CCPU сигналами синхронизации и сброса в для управляющих ПУ в, соответственно, кластере PC0, …., PC3**.** CRGL0…CRGL3 по составу и функциональным особенностям работы полностью идентичны генератору GRGH, который был описан ранее (см. раздел 4.15). Исключением являются только адреса программно доступных регистров этих устройств, для каждого устройства они свои. Поэтому в данном разделе приведен только перечень регистров для CRGL0…CRGL3 с указанием их адресов (см. таблицы 5.252- 5.255).

Таблица 5.252 - Спецификация регистров генератора тактовых сигналов и сигналов сброса CRGL0

Адрес (байтовый доступ)	Тип	Имя регистра	Описание
0x000C_6000 hex	ЧТ/ЗП	PLLCR	Регистр управления PLL.
0x000C_6004 hex	ЧТ	PLLRDYMR	Регистр мониторинга готовности PLL.
0x000C_6008 hex	ЧТ/ЗП	IMACR	Регистр маски прерывания от PLL.
0x000C_600C hex	ЗП	PICR	Регистр снятия прерывания от PLL.
0x000C_6010 hex	ЧТ	CLKDIV	Регистр управления выходными делителями.
0x000C_6014 hex	ЧТ	CRGCR	Регистр управления CRGL0.
0x000C_6C00 hex	ЧТ/ЗП	PSCR	Регистр управления запуском процессора.
0x000C_6F00 hex	ЧТ/ЗП	CRGSR	Регистр статуса CRGL0.

Таблица 5.253 - Спецификация регистров генератора тактовых сигналов и сигналов сброса CRGL1

Адрес (байтовый доступ)	Тип	Имя регистра	Описание
0x000C_7000 hex	ЧТ/ЗП	PLLCR	Регистр управления PLL.
0x000C_7004 hex	ЧТ	PLLRDYMR	Регистр мониторинга готовности PLL.
0x000C_7008 hex	ЧТ/ЗП	IMACR	Регистр маски прерывания от PLL.
0x000C_700C hex	ЗП	PICR	Регистр снятия прерывания от PLL.
0x000C_7010 hex	ЧТ	CLKDIV	Регистр управления выходными делителями.
0x000C_7014 hex	ЧТ	CRGCR	Регистр управления CRGL1.
0x000C_7C00 hex	ЧТ/ЗП	PSCR	Регистр управления запуском процессора.
0x000C_7F00 hex	ЧТ/ЗП	CRGSR	Регистр статуса CRGL1.

Таблица 5.254 - Спецификация регистров генератора тактовых сигналов и сигналов сброса CRGL2

Адрес (байтовый доступ)	Тип	Имя регистра	Описание
0x000C_8000 hex	ЧТ/ЗП	PLLCR	Регистр управления PLL.
0x000C_8004 hex	ЧТ	PLLRDYMR	Регистр мониторинга готовности PLL.
0x000C_8008 hex	ЧТ/ЗП	IMACR	Регистр маски прерывания от PLL.
0x000C_800C hex	ЗП	PICR	Регистр снятия прерывания от PLL.
0x000C_8010 hex	ЧТ	CLKDIV	Регистр управления выходными делителями.
0x000C_8014 hex	ЧТ	CRGCR	Регистр управления CRGL2.
0x000C_8C00 hex	ЧТ/ЗП	PSCR	Регистр управления запуском процессора.
0x000C_8F00 hex	ЧТ/ЗП	CRGSR	Регистр статуса CRGL2.

Таблица 5.255 - Спецификация регистров генератора тактовых сигналов и сигналов сброса CRGL3

Адрес (байтовый доступ)	Тип	Имя регистра	Описание
0x000C_9000 hex	ЧТ/ЗП	PLLCR	Регистр управления PLL.
0x000C_9004 hex	ЧТ	PLLRDYMR	Регистр мониторинга готовности PLL.
0x000C_9008 hex	ЧТ/ЗП	IMACR	Регистр маски прерывания от PLL.
0x000C_900C hex	ЗП	PICR	Регистр снятия прерывания от PLL.
0x000C_9010 hex	ЧТ	CLKDIV	Регистр управления выходными делителями.
0x000C_9014 hex	ЧТ	CRGCR	Регистр управления CRGL3.
0x000C_9C00 hex	ЧТ/ЗП	PSCR	Регистр управления запуском процессора.
0x000C_9F00 hex	ЧТ/ЗП	CRGSR	Регистр статуса CRGL3.

5.17.5 Генератор тактовых сигналов и сигналов сброса CCRG

Данный генератор предназначен для управления со стороны внешнего хоста сигналами синхронизации и сброса в для центрального управляющего ПУ (CCPU). CCRG по составу и функциональным особенностям работы полностью идентичны генератору GRGH, который был описан ранее (см. раздел 4.15). Исключением являются только адреса программно доступных регистров этих устройств, для каждого устройства они свои. Поэтому в данном разделе приведен только перечень регистров для CCRG с указанием его адресов (см. таблицу 5.256).

Таблица 5.256 - Спецификация регистров генератора тактовых сигналов и сигналов сброса CCRG

Адрес (байтовый доступ)	Тип	Имя регистра	Описание
0x000C_A000 hex	ЧТ/ЗП	PLLCR	Регистр управления PLL
0x000C_A004 hex	ЧТ	PLLRDYMR	Регистр мониторинга готовности PLL
0x000C_A008 hex	ЧТ/ЗП	IMACR	Регистр маски прерывания от PLL
0x000C_A00C hex	ЗП	PICR	Регистр снятия прерывания от PLL
0x000C_A010 hex	ЧТ	CLKDIV	Регистр управления выходными делителями
0x000C_A014 hex	ЧТ	CRGCR	Регистр управления CCRG
0x000C_AC00 hex	ЧТ/ЗП	PSCR	Регистр управления запуском процессора
0x000C_AF00 hex	ЧТ/ЗП	CRGSR	Регистр статуса CCRG

5.17.6 Последовательный порт доступа для тестовых и отладочных данных DAP

Данный порт реализован на основе JTAG-порта, но помимо чисто тестовых функций может выполнять также задачи мониторинга и отладки на основе технологии CoreSight фирмы ARM ltd с помощью штатных аппаратных и программных средств указанной фирмы. С помощью порта можно реализовать следующие действия:

1. Системный сброс всех процессорных ядер или отдельного ядра.
2. Изменение частоты работы центрального управляющего ПУ (CCPU), а также управляющего ПУ (CPU) и четырех ПУ на базе NMC4 (NMPU) в каждом кластере.
3. Запись начальных данных и программ во внутреннюю память микросхемы 1879ВМ8Я.
4. Запуск всех или конкретных процессорных ядер для выполнения своей программы.
5. Проверка корректности выполнения программ с помощью чтения результата из внутренней памяти микросхемы 1879ВМ8Я.
6. Пошаговая отладка программ и их трассировка (только для управляющих ПУ на основе ядер ARM Cortex-A5).

5.17.7 Внешние выводы последовательного порта доступа для тестовых и отладочных данных DAP

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

Таблица 5.257 – Выводы микросхемы, относящиеся к последовательному порту доступа для тестовых и отладочных данных DAP

Вывод	Тип буфера	Примечание
TDO_SWO	O(Z)	Выход данных порта JTAG отладочного модуля
TDI	I	Вход данных тестового порта JTAG
TCK_SWCLK	I	Тактовый сигнал тестового порта отладочного модуля
TMS_SWDIO	I/O	Выбор режима работы отладочного модуля
TRSTn	I	Сброс порта JTAG отладочного модуля
SRSTn	I/O	Сброс отладочного модуля (pullup 80 КОм)

5.17.8 Рекомендуемая схема подключения DAP

Аппаратный отладчик подключается к плате с микросхемой 1879ВМ8Я через 14-выводный IDC14M разъем с шагом выводов 2,54 мм (ARM JTAG 14). На рисунке ниже приведена типовая схема подключения JTAG-разъема к микросхеме.

Рисунок 5.44 – Схема типового подключения JTAG-разъема к микросхеме 1879ВМ8Я

Дополнительно рекомендуется устанавливать микросхемы монитора питания с выходом типа «открытый коллектор» (например, STM1001) на входы микросхемы SRSTn и TRSTn для обеспечения стабильной работы.

Микросхема поддерживает отладку по стандарту JTAG, а также по стандарту SWD. Описанная выше схема подключения позволяет использовать оба стандарта для отладки ПО на микросхеме 1879ВМ8Я.

Частота обмена по отладочному интерфейсу зависит от длины соединительного кабеля. Приведенная схема разводки применима для коротких кабелей (около 30 см). В этом случае теоретически достижимы следующие максимальные частоты:

- 10МГц при использовании JTAG интерфейса;
- 50МГц при использовании SWD интерфейса.

Для кабелей средней длины мощности выходных буферов микросхемы (сигналов TDO_SWO, TMS_SWDIO и SRSTn) 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.

В таблице ниже приведено описание сигналов разъема ARM JTAG 14 и особенностей работы интерфейса при подсоединении микросхемы 1879ВМ8Я к аппаратному отладчику DSTREAM фирмы ARM.

Таблица 5.258 – Описание контактов JTAG-разъема ARM JTAG 14

Контакт	I/O	Описание
TDI	O	Вход данных тестового порта от отладчика. Этот сигнал должен иметь pullup резистор.
TDO	I	Выход данных тестового порта к отладчику. Этот сигнал должен иметь pullup резистор. Рекомендуется установка последовательного согласующего резистора вблизи СБИС на данный сигнал.
TMS	O	Выбор режима тестирования от отладчика. Этот сигнал должен иметь pullup резистор.
TCK	O	Тактовый сигнал тестового порта от отладчика. TDI и TMS выдаются синхронно с этим синхросигналом. Этот сигнал должен иметь pullup резистор.
nTRST	O	Сброс тестового порта в микросхеме от отладчика. Сигнал с активным низким уровнем. Этот сигнал должен иметь pullup резистор.
nSRST	I/O	Внешний программный сброс микросхемы от отладчика. Сигнал с активным низким уровнем. Сигнал может быть выставлен в активное состояние как отладчиком DSTREAM так и самой СБИС. DSTREAM способен отслеживать выставление этого сигнала со стороны СБИС. СБИС содержит встроенный pullup резистор с типовым значением 80кОм на данном выводе для исключения случайного сброса при отсутствии подключения к DSTREAM. Но т.к. этот pullup резистор может быть программно отключен, то рекомендуется дублировать его на печатной плате.
SWDIO (SWD режим)	I/O	Двунаправленные данные отладочного порта в режиме Serial Wire Debug (SWD). Этот сигнал должен иметь pullup резистор. Рекомендуется установка последовательного согласующего резистора вблизи СБИС на данный сигнал.
SWCLK (SWD режим)	O	Синхросигнал отладочного порта в режиме Serial Wire Debug (SWD). Этот сигнал должен иметь pullup резистор.
SWO (SWD режим)	I	Выходной сигнал трасс отладочного порта в режиме Serial Wire Debug (SWD). Этот сигнал должен иметь pullup резистор. Рекомендуется установка последовательного согласующего резистора вблизи микросхемы 1879ВМ8Я на данный сигнал.
VTREF	I	Напряжение смещения. DSTREAM определяет наличие питания в отлаживаемой системе, питание должно быть как минимум 0,6В. Используется для сопряжения логических уровней DSTREAM с уровнями буферов ввода/вывода отлаживаемой микросхемы 1879ВМ8Я. Этот сигнал может иметь pullup резистор для защиты от КЗ, но не более 100 Ом.
GND	-	Общий

6 Электрические, динамические и конструктивные характеристики микросхемы 1879ВМ8Я

6.1 Состав и расположение внешних выводов микросхемы 1879ВМ8Я

Микросхема 1879ВМ8Я изготавливается в 1444-выводном корпусе HFCBGA (High Performance Flip Chip Ball Grid Array). Расположение внешних выводов микросхемы представлено на рисунке 6.1.

Рисунок 6.1 - Расположение внешних выводов микросхемы 1879ВМ8Я (вид со стороны выводов)

В таблице 6.1 приводятся выводы микросхемы, сгруппированные по их функциональному назначению.

Таблица 6.1 - Выводы микросхемы 1879ВМ8Я в соответствии с их функциональным назначением

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Общее управление (8 выводов)				
CLK_REFI	1	I	Вход опорного тактового сигнала	AU1
CLK_REFO	1	O	Выход опорного тактового сигнала	AT1
XRSTn	1	I	Системный сброс	AV2
BOOTM0, BOOTM1, BOOTM2	3	I	Режим начальной загрузки процессора	AR3, AR1, AT3
TM	1	I	Режим тестирования процессора, при нормальной работе на вход подаётся лог. ноль	AT2
VPD	1	I	Управление статическим током, при нормальной работе на вход подаётся лог. ноль	AV3
Управление спящим режимом процессора(2 вывода)				
WKUPRQ	1	I	Запрос на вывод процессора из спящего режима	AP1
WKUPAK	1	O	Разрешение на снятие запроса на вывод процессора из спящего режима	AP2
JTAG интерфейс (5 выводов)				
BS_TDO	1	O(Z)	Выход данных тестового порта JTAG	D37
BS_TDI	1	I	Вход данных тестового порта JTAG	A38
BS_TCK	1	I	Тактовый сигнал тестового порта JTAG	B37
BS_TMS	1	I	Выбор режима тестирования JTAG	C37
BS_TRSTn	1	I	Сброс тестового порта JTAG	A37
Debug интерфейс (6 выводов)				
TDO_SWO	1	O(Z)	Выход данных порта JTAG отладочного модуля	F1
TDI	1	I	Вход данных тестового порта JTAG	A1
TCK_SWCLK	1	I	Тактовый сигнал тестового порта отладочного модуля	B2
TMS_SWDIO	1	I/O	Выбор режима работы отладочного модуля	C1
TRSTn	1	I	Сброс порта JTAG отладочного модуля	A2
SRSTn	1	I/O	Сброс отладочного модуля	C2
Выводы SPI интерфейса (7 выводов)				
SPI_CLK	1	O	Синхросигнал SPI интерфейса	AV38
SPI_TXD	1	O	Передаваемые данные	AT37
SPI_RXD	1	I	Принимаемые данные	AV37
SPI_CS0n	1	O	Выбор приемника 0	AT38
SPI_CS1n	1	O	Выбор приемника 1	AU38
SPI_CS2n	1	O	Выбор приемника 2	AR38
SPI_CS3n	1	O	Выбор приемника 3	AU37

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы Ethernet интерфейса (14 выводов)				
EDCLA0, EDCLA1, EDCLA2, EDCLA3	4	I	Выводы конфигурации MAC адреса	AV35, AU35, AV36, AR35
ETH_ENABLE	1	I	Конфигурационный вход наличия микросхемы физического уровня	AM29
ETH_TXD0, ETH_TXD1	2	O	Выходные данные интерфейса	AL27, AM28
ETH_TXEN	1	O	Разрешение передачи данных	AM27
ETH_RXD0, ETH_RXD1	2	I	Входные данные интерфейса	AK27, AN29
ETH_CRSDV	1	I	Признак обнаружения несущей	AK29
ETH_REFCLK	1	1	Входной опорный тактовый сигнал 50МГц	AL29
ETH_MDC	1	O	Выходной тактовый сигнал для микросхемы физического уровня	AN27
ETH_MDIO	1	I/O	Шина управления микросхемой физического уровня	AN28
Выводы Watch DOG таймера (1 вывод)				
WDT	1	O	Выход сторожевого таймера	AN20
Выводы порта общего назначения управляющей части (8 выводов)				
GPIOC0, …, GPIOC7	8	I/O	Программируемы входы/выходы	AK23, AM23, AL23, AN25, AN23, AL25, AM25, AK25

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы DDR контроллера управляющей части (76 выводов)				
DDRC_CK, DDRC_CKn	2	O	Тактовый сигнал DDR памяти	B22, C22
DDRC_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления	G18
DDRC_A0, …, DDRC_A14	15	O	Шина адреса памяти	J20, G22, H23, J22, J23, G23, J17, H17, H16, H20, J16, F15, H15, G19, G15
DDRC_BA0, …, DDRC_BA2	3	O	Выбор адресуемых банков в микросхемах памяти	H19, J21, H22
DDRC_CSn	1	O	Выбор внешней памяти	G17
DDRC_RASn	1	O	Строб адреса строки внешней памяти	H18
DDRC_CASn	1	O	Строб адреса столбца внешней памяти	J19
DDRC_WEn	1	O	Признак операции записи	J18
DDRC_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти	H21
DDRC_DM0, …, DDRC_DM3	4	O	Сигналы маскирования байтов данных при операциях записи	B20, E24, D18, D15
DDRC_DQ0, …, DDRC_DQ31	32	I/O	Шина данных	E22,C21,B21,D21,E21,C20,D20,E20, C24,B24,E25, D24,B23,E23,D23,C23, B17,C17,E17,E18,D17,C18,B18,C19, E14,C14,D14,B14,E15,C15,B15,C16
DDRC_DQS0, DDRC_DQS0n,…, DDRC_DQS3, DDRC_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных	A20, A21, A24, A23, A17, A18, A15, A14
DDRC_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти	F17
DDRC_ATB0, DDRC_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	G14, J15
DDRC_CAL0	1	I	Вывод для подключения калибровочного резистора	G16
DDRC_PLL_TESTO_P, DDRC_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса	F20, F21
Выводы контроллера PCI Express (21 вывод)				
PCI_CLKREF_P, PCI_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера PCI Express	AT19, AR19
PCI_TXP0, PCI_TXN0, …, PCI_TXP3, PCI_TXN3	8	O	Парафазные выходы данных	AT21, AR21, AT23, AR23, AT17, AR17, AT15, AR15
PCI_RXP0, PCI_RXN0, …, PCI_RXP3, PCI_RXN3	8	I	Парафазные входы данных	AV20, AU20, AV22, AU22, AV18, AU18, AV16, AU16
PCI_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)	AR20
PCI_ATB0, PCI_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	AU21, AV19

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы кластера 0				
Выводы порта общего назначения (8 выводов)				
GPIO00, …, GPIO7	8	I/O	Программируемы входы/выходы	F37, F36, E38, F35, F38, D38, E37, C38
Выводы DDR контроллера (76 выводов)				
DDR0_CK, DDR0_CKn	2	O	Тактовый сигнал DDR памяти	P33, N33
DDR0_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления	L33
DDR0_A0, …, DDR0_A14	15	O	Шина адреса памяти	T32,V31,R31,R32,P31,U31,K31,J33, J32, J31, H32, H33, H31, L31, L32
DDR0_BA0, …, DDR0_BA2	3	O	Выбор адресуемых банков в микросхемах памяти	T31, V33, V32
DDR0_CSn	1	O	Выбор внешней памяти	K32
DDR0_RASn	1	O	Строб адреса строки внешней памяти	W31
DDR0_CASn	1	O	Строб адреса столбца внешней памяти	W32
DDR0_WEn	1	O	Признак операции записи	U32
DDR0_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти	P32
DDR0_DM0, …, DDR0_DM3	4	O	Сигналы маскирования байтов данных при операциях записи	V34, P34, M36, J34
DDR0_DQ0, …, DDR0_DQ31	32	I/O	Шина данных	V36,W36,V37,V35,U34,U35,U36,U37, R36,T36,R34,R35,R37,P36,P35,P37, M34,N34,M35,M37,L34,L35,L37,L36, J36,K36,J37,J35,H34,H35,H36,H37
DDR0_DQS0, DDR0_DQS0n, …, DDR0_DQS3, DDR0_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных	V38, U38, P38, R38, M38, L38, J38, H38
DDR0_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти	N32
DDR0_ATB0, DDR0_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	M32, M31
DDR0_CAL0	1	I	Вывод для подключения калибровочного резистора	N31
DDR0_PLL_TESTO_P, DDR0_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса	N36, N37

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы кластера 0				
Выводы последовательного порта кластера (21 вывод)				
CP0_CLKREF_P, CP0_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера последовательного порта	A31, B31
CP0_TXP0, CP0_TXN0, …, CP0_TXP3, CP0_TXN3	8	O	Парафазные выходы данных	A33, B33, A35, B35, A29,B29, A27, B27
CP0_RXP0, CP0_RXN0, …, CP0_RXP3, CP0_RXN3	8	I	Парафазные входы данных	C32, D32, C34, D34, C30, D30, C28, D28
CP0_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)	A34
CP0_ATB0, CP0_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	A32, D31

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы кластера 1				
Выводы порта общего назначения (8 выводов)				
GPIO10, …, GPIO17	8	I/O	Программируемы входы/выходы	E1, F3, E2, D1, F2, F4, D2, E3
Выводы DDR контроллера (76 выводов)				
DDR1_CK, DDR1_CKn	2	O	Тактовый сигнал DDR памяти	P6, N6
DDR1_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления	N8
DDR1_A0, …, DDR1_A14	15	O	Шина адреса памяти	T7, V6, T8, R7, R8, N7, J6, H7, J7, K7, J8, H8, K8, L8, L7
DDR1_BA0, …, DDR1_BA2	3	O	Выбор адресуемых банков в микросхемах памяти	U7, V7, V8
DDR1_CSn	1	O	Выбор внешней памяти	H6
DDR1_RASn	1	O	Строб адреса строки внешней памяти	W8
DDR1_CASn	1	O	Строб адреса столбца внешней памяти	W7
DDR1_WEn	1	O	Признак операции записи	U8
DDR1_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти	P8
DDR1_DM0, …, DDR1_DM3	4	O	Сигналы маскирования байтов данных при операциях записи	V2, P5, M3, J2
DDR1_DQ0, …, DDR1_DQ31	32	I/O	Шина данных	V4, W3, V3, V5, U5, U4, U3, U2, R5, T3, R4, R3, R2, P3, P4, P2, M4, N5, M5, M2, L4, L5, L2, L3, J4, K3, J3, J5, H5, H4, H3, H2
DDR1_DQS0, DDR1_DQS0n, …, DDR1_DQS3, DDR1_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных	V1, U1, P1, R1, M1, L1, J1, H1
DDR1_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти	P7
DDR1_ATB0, DDR1_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	M7, M8
DDR1_CAL0	1	I	Вывод для подключения калибровочного резистора	L6
DDR1_PLL_TESTO_P, DDR1_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса	N3, N2

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы кластера 1				
Выводы последовательного порта кластера (21 вывод)				
CP1_CLKREF_P, CP1_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера последовательного порта	A8, B8
CP1_TXP0, CP1_TXN0, …, CP1_TXP3, CP1_TXN3	8	O	Парафазные выходы данных	A6, B6, A4, B4, A10, B10 A12, B12
CP1_RXP0, CP1_RXN0, …, CP1_RXP3, CP1_RXN	8	I	Парафазные входы данных	C7, D7, C5, D5, C9, D9, C11, D11
CP1_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)	A5
CP1_ATB0, CP1_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	A7, D8

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы кластера 2				
Выводы порта общего назначения (8 выводов)				
GPIO20, …, GPIO27	8	I/O	Программируемые входы/выходы	AP38, AP37, AP36, AT36, AP35, AR36, AT35, AR37
Выводы DDR контроллера (76 выводов)				
DDR2_CK, DDR2_CKn	2	O	Тактовый сигнал DDR памяти	AE33, AF33
DDR2_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления	AH33
DDR2_A0, …, DDR2_A14	15	O	Шина адреса памяти	AD31, AC32, AE32, AE31, AF32, AD32, AK33, AK32, AJ31, AM33, AK31, AL32, AL31, AM31, AM32
DDR2_BA0, …, DDR2_BA2	3	O	Выбор адресуемых банков в микросхемах памяти	AA33, AB32, AC31
DDR2_CSn	1	O	Выбор внешней памяти	AJ32
DDR2_RASn	1	O	Строб адреса строки внешней памяти	AA31
DDR2_CASn	1	O	Строб адреса столбца внешней памяти	AA32
DDR2_WEn	1	O	Признак операции записи	AB31
DDR2_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти	AF31
DDR2_DM0, …, DDR2_DM3	4	O	Сигналы маскирования байтов данных при операциях записи	AB37, AE37, AG37, AL34
DDR2_DQ0, …, DDR2_DQ31	32	I/O	Шина данных	AA35, AA37, AA36, AA34, AB35, AB34, AC36, AB36, AD35, AD37, AD36, AD34, AE36, AE35, AE34, AF34, AG35, AG36, AG34, AH34, AH36, AH35, AJ36, AH37, AK35, AK37, AK36, AK34, AL35, AL36, AL37, AM34
DDR2_DQS0, DDR2_DQS0n,…, DDR2_DQS3, DDR2_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных	AA38, AB38, AD38, AE38, AG38, AH38, AK38, AL38
DDR2_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти	AG32
DDR2_ATB0, DDR2_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	AG31, AH31
DDR2_CAL0	1	I	Вывод для подключения калибровочного резистора	AH32
DDR2_PLL_TESTO_P, DDR2_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса	AF36, AF37

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы кластера 2				
Выводы последовательного порта кластера (21 вывод)				
CP2_CLKREF_P, CP2_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера последовательного порта	AR29, AT29
CP2_TXP0, CP2_TXN0, …, CP2_TXP3, CP2_TXN3	8	O	Парафазные выходы данных	AT31, AR31, AT33, AR33, AT27, AR27, AT25, AR25
CP2_RXP0, CP2_RXN0, …, CP2_RXP3, CP2_RXN3	8	I	Парафазные входы данных	AV30, AU30, AV32, AU32, AV28, AU28, AV26, AU26
CP2_REXT	1	I	Вход калибровочного резистора (3.1 КОм 1%)	AR30
CP2_ATB0, CP2_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	AR32, AV29

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы кластера 3				
Выводы порта общего назначения (8 выводов)				
GPIO30, …, GPIO37	8	I/O	Программируемы входы/выходы	AK10, AL10, AK11, AL11, AL12, AM10, AN11, AM12
Выводы DDR контроллера (76 выводов)				
DDR3_CK, DDR3_CKn	2	O	Тактовый сигнал DDR памяти	AE6, AF6
DDR3_CKE	1	O	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления	AH6
DDR3_A0, …, DDR3_A14	15	O	Шина адреса памяти	AD8,AC8,AE7,AE8,AF7,AD7,AK6,AJ8, AK7,AM6,AK8,AL7,AM7,AM8,AL8
DDR3_BA0, …, DDR3_BA2	3	O	Выбор адресуемых банков в микросхемах памяти	AA6, AB8, AC7
DDR3_CSn	1	O	Выбор внешней памяти	AJ7
DDR3_RASn	1	O	Строб адреса строки внешней памяти	AA8
DDR3_CASn	1	O	Строб адреса столбца внешней памяти	AA7
DDR3_WEn	1	O	Признак операции записи	AB7
DDR3_RESETn	1	O	Асинхронный сброс микросхем внешней DDR памяти	AF8
DDR3_DM0, …, DDR3_DM3	4	O	Сигналы маскирования байтов данных при операциях записи	AB5, AE2, AG2, AL5
DDR3_DQ0, …, DDR3_DQ31	32	I/O	Шина данных	AA4, AA2, AA3, AA5, AB4, AB3, AC3, AB2, AD4, AD2, AD3, AD5, AE3, AE4, AE5, AF5, AG4, AG3, AG5, AH5, AH3, AH4, AJ3, AH2, AK4, AK2, AK3, AK5, AL4, AL3, AL2, AM5
DDR3_DQS0, DDR3_DQS0n, …, DDR3_DQS3, DDR3_DQS3n	8	I/O	Дифференциальный строб байтов на шине данных	AA1, AB1, AD1, AE1, AG1, AH1, AK1, AL1
DDR3_ODT	1	O	Управление согласующими резисторами, встроенными в микросхемы DDR памяти	AG7
DDR2_ATB0, DDR1_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	AG8, AH8
DDR2_CAL0	1	I	Вывод для подключения калибровочного резистора	AH7
DDR3_PLL_TESTO_P, DDR3_PLL_TESTO_N	2	I/O	Дифференциальные тестовые выводы PLL интерфейса	AF3, AF2

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Выводы кластера 3				
Выводы последовательного порта кластера (21 вывод)				
CP3_CLKREF_P, CP3_CLKREF_M	2	I	Опорный парафазный тактовый сигнал физического интерфейса контроллера последовательного порта	AR9, AT9
CP3_TXP0, CP3_TXN0, …, CP3_TXP3, CP3_TXN3	8	O	Парафазные выходы данных	AT7, AR7, AT5, AR5, AT11, AR11, AT13, AR13
CP3_RXP0, CP3_RXN0, …, CP3_RXP3, CP3_RXN3	8	I	Парафазные входы данных	AV8, AU8, AV6, AU6, AV10, AU10, AV12, AU12
CP3_REXT	1	I	Вход калибровочного резистора (3.1 Ком 1%)	AR8
CP3_ATB0, CP3_ATB1	2	I/O	Выводы тестовой аналоговой шины интерфейса	AR6, AU5

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
			Цепи питания	
VCC3	297	S	Напряжение питания ядра (0,9 В)	M11, M12, M13, M14, M15, M16, M17, M18, M19, M20, M21, M22, M23, M24, M25, M26, M27, M28, N11, N12, N13, N14, N15, N16, N17, N18, N19, N20, N21, N22, N23, N24, N25, N26, N27, N28, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20, P21, P22, P23, P24, P25, P26, P27, P28, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21,R22, R23, R24, R25, R26, R27, R28, T11, T12, T13, T14, T15, T16, T17, T18, T19, T20, T21, T22, T23, T24, T25, T26, T27, T28, U11, U12, U13, U14, U15, U16, U17, U18, U19, U20, U21, U22, U23, U24, U25, U26, U27, U28, V11, V12, V13, V14, V15, V16, V17, V18, V19, V20, V21, V22, V23, V24, V25, V26, V27, V28, W11, W12, W13, W14, W15, W16, W17, W18, W19, W20, W21, W22, W23, W24, W25, W26, W27, W28, Y11, Y12, Y13, Y14, Y15, Y16, Y17, Y18, Y19, Y20, Y21, Y22, Y23, Y24, Y25, Y26, Y27, Y28, AA11, AA12, AA13, AA14, AA15, AA16, AA17, AA18, AA19, AA20, AA21, AA22, AA23, AA24, AA25, AA26, AA27, AA28, AB11, AB12, AB13, AB14, AB15, AB16, AB17, AB18, AB19, AB20, AB21, AB22, AB23, AB24, AB25, AB26, AB27, AB28, AC11, AC12, AC13, AC14, AC15, AC16, AC17, AC18, AC19, AC20, AC21, AC22, AC23, AC24, AC25, AC26, AC27, AC28, AD11, AD12, AD13, AD14, AD15, AD16, AD17, AD18, AD19, AD20, AD21, AD22, AD23, AD24, AD25, AD26, AD27, AD28, AE11, AE12, AE13, AE14, AE15, AE16, AE17, AE18, AE19, AE20, AE21, AE22, AE23, AE24, AE25, AE26, AE27, AE28, AF11, AF12, AF13, AF14, AF15, AF16, AF17, AF18, AF19, AF20, AF21, AF22, AF23, AF24, AF25, AF26, AF27, AF28, AG11, AG12, AG13, AG14, AG15, AG16, AG17, AG18, AG19, AG20, AG21, AG22, AG23, AG24, AG25, AG26, AG27, AG28, AL17, G28, F11, AL20, AN15, J28, K12, AL21, AK16
VCC2	25	S	Напряжение питания интерфейса DDR3 +1,5 В	L17, L18, L19, L20, L21, N29, P29, R29, T29, U29, N10, P10, R10, T10, U10, AB29, AC29, AD29, AE29, AF29, AB10, AC10, AD10, AE10, AF10
VCC1	7	S	Напряжение питания КМОП-буферов ввода/вывода +1,8 В	B1, B38, AL28, AM11, AN21, AR2, AU36

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
GND	474	S	Общий	A3,A11,A13,A16,A19,A22,A25,A26,A28,A36, B3,B5,B7,B9,B11,B13,B16,B19,B25,B26,B28, B30,B32,B34,B36,C3,C13,C25,C26,C36,D3, D4,D12,D13,D16,D19,D22,D25,D26,D27,D35, D36,E4,E5,E6,E7,E8,E9,E10,E11,E12,E13, E16,E19,E26,E27,E28,E29,E30,E31,E32,E33, E34,E35,E36,F5,F6,F7,F8,F9,F10,F13,F14,F16, F18,F19,F22,F23,F24,F25,F26,F29,F30,F31, F32,F33,F34,G1,G2,G3,G4,G5,G6,G7,G8,G9, G10,G13,G20,G21,G24,G25,G26,G29,G30, G31,G32,G33,G34,G35,G36,G37,G38,H9,H10, H11,H12,H13,H14,H24,H25,H26,H27,H28, H29,H30,J9,J10,J13,J14,J24,J25,J26,J29,J30, K1,K2,K4,K5,K6,K9,K10,K13,K14,K15,K16, K17,K18,K19,K20,K21,K22,K23,K24,K25, K26,K29,K30,K33,K34,K35,K37,K38,L9,L10, L11,L12,L13,L14,L15,L16,L22,L23,L24,L25, L26,L27,L28,L29,L30,M6,M9,M10,M29,M30, M33,N1,N4,N9,N30,N35,N38,P9,P30,R6,R9, R30,R33,T1,T2,T4,T5,T6,T9,T30,T33,T34, T35,T37,T38,U6,U9,U30,U33,V9,V10,V29, V30,W1,W2,W4,W5,W6,W9,W10,W29,W30, W33,W34,W35,W37,W38,Y1,Y2,Y3,Y4,Y5, Y6,Y7,Y8,Y9,Y10,Y29,Y30,Y31,Y32,Y33, Y34,Y35,Y36,Y37,Y38,AA9,AA10,AA29, AA30,AB6,AB9,AB30,AB33,AC1,AC2,AC4, AC5,AC6,AC9,AC30,AC33,AC34,AC35, AC37,AC38,AD6,AD9,AD30,AD33,AE9, AE30,AF1,AF4,AF9,AF30,AF35,AF38,AG6, AG9,AG10,AG29,AG30,AG33,AH9,AH10, AH11,AH12,AH13,AH14,AH15, AH16, AH17, AH18,AH19,AH20,AH21,AH22,AH23,AH24, AH25,AH26,AH27,AH28,AH29,AH30,AJ1, AJ2,AJ4,AJ5,AJ6,AJ9,AJ10,AJ11,AJ12,AJ13, AJ14,AJ15,AJ16,AJ17,AJ18,AJ19,AJ20,AJ21, AJ22,AJ23,AJ24,AJ25,AJ26,AJ27,AJ28,AJ29, AJ30,AJ33,AJ34,AJ35,AJ37,AJ38,AK9, AK12, AK13,AK14,AK24,AK26,AK28,AK30,AL6, AL9,AL13,AL14,AL24,AL26,AL30,AL33, AM1,AM2,AM3,AM4,AM9,AM13,AM14, AM18,AM19,AM20,AM21,AM22,AM24, AM26,AM30,AM35,AM36,AM37,AM38,AN1, AN2,AN3,AN4,AN5,AN6,AN7,AN8,AN9, AN10,AN12,AN13,AN14,AN19,AN22,AN24, AN26,AN30,AN31,AN32,AN33,AN34,AN35, AN36,AN37,AN38,AP3,AP4,A5P,AP6,AP7, AP8,AP9,AP10,AP11,AP12,AP13,AP14,AP15, AP16,AP17,AP18,AP19,AP20,AP21,AP22, AP23,AP24,AP25,AP26,AP27,AP28,AP29, AP30,AP31,AP32,AP33,AP34,AR4,AR12, AR14,AR16,AR24,AR26,AR34,AT4,AT6, AT8,AT10,AT12,AT14,AT16,AT18,AT20, AT22,AT24,AT26,AT28,AT30,AT32,AT34, AU2,AU3,AU4,AU14,AU24,AU34,AV1,AV4, AV5,AV13,AV14,AV15,AV23,AV24,AV25, AV33,AV34

Продолжение таблицы 6.1

Обозначение	Кол-во	Тип	Функциональное назначение	Номер вывода
Аналоговые цепи питания				
VCC1A	6	S	Аналоговое напряжение питания +1,8 В	AR22, C35, C4, AU33, AV9, AM17
VCC3A	44	S	Аналоговое напряжение питания +0,9 В	AU19, C31, C8, AU27, AU11, AR18, C27, C12, AV31, AV7, AV21, D33, D6, AR28, AR10, AU15, A30, A9, AV27, AV11, AU23, D29, D10, AU31, AU7, AU17, C33, C6, AU25, AU13, AV17, C29, C10, AU29, AU9, AL18, G27, F12, AL19, AM15, K28, K11, AK21, AL15
GNDA	11	S	Аналоговая земля	AK18, F27, G11, AK19, AM16 K27, J11, AK22, AK15, AN17, AN18
GND_PLL	9	S	Земля PLL	AK17, F28, G12, AK20, AN16 J27, J12, AL22, AL16

Примечания: 1) Для выводов с префиксом «n» на конце имени сигнала активным является низкий уровень сигнала.
2) Используемые обозначения типов выводов:
I –вход;
O - выход;
O(Z) - выход с высокоимпедансным состоянием;
I/O – двунаправленный вывод;
S –питание.

6.2 Конструктивные характеристики

Микросхема 1879ВМ8Я изготавливается в 1444-выводном корпусе типа High-Performance Flip Chip Ball Grid Array (HFCBGA). Внешний вид корпуса микросхемы представлен на рисунке 6.2 с указанием сведений о габаритных и установочных размерах. Для повышения отводимого тепла корпуса HFCBGA дополняют металлической пластиной, которая контактирует с обратной стороной кристалла и уменьшает термическое сопротивление кристалл-среда.

Основные характеристики корпуса:

Механические характеристики:

1. размер - 40×40 мм^2;
2. шаг выводов- 1,00 мм;
3. размер вывода – 0,60 мм.

Электрические характеристики:

4. индуктивность вывода – не более 6,5 нГн;
5. сопротивление вывода – не более 0,270 Ом;
6. ёмкость вывода – не более 1,5 пФ.

Тепловые характеристики:

7. Тепловое сопротивление при скорости обдува 0 м/с – 6,84 °С/Вт;
8. Тепловое сопротивление при скорости обдува 1 м/с – 5,99 °С/Вт;
9. Тепловое сопротивление при скорости обдува 2 м/с – 5,49 °С/Вт.

Рисунок 6.2 - Корпус микросхемы 1879ВМ8Я

6.3 Электрические характеристики

Электрические параметры микросхем должны соответствовать нормам, приведенным в таблице 6.2.

Таблица 6.2 – Электрические параметры микросхемы

Наименование параметра, единица измерения	Обозна-чение	Норма параметра,		Режим измерения	Темпера-тура среды, ^{0}C
		не менее	не более		
Выходное напряжение низкого уровня буферов КМОП, В	U_{OL1}	0,0	0,2	U_{CC1} = 1,98 В, втекающий выходной ток I_{OL1}= 100 мкА	+25 минус 60 + 85
Выходное напряжение высокого уровня буферов КМОП, В	U_{OH1}	1,4	1,62	U_{CC1} = 1,62 В, вытекающий выходной ток I_{OH1}= -100 мкА	+25 минус 60 + 85
Входной ток утечки низкого уровня, мкА	I_{ILL}	-	10	U_{IL} = 0 В	+25 минус 60 + 85
Входной ток утечки высокого уровня, мкА	I_{ILH}	-	10	$U_{IH} = U_{CC1}$	+25 минус 60 + 85
Ток утечки низкого уровня в состоянии «Выключено», мкА	I_{OZL}	-	10	U_{IL} = 0 В	+25 минус 60 + 85
Ток утечки высокого уровня в состоянии «Выключено», мкА	I_{OZH}	-	10	$U_{IH} = U_{CC1}$	+25 минус 60 + 85
Динамический ток потребления буферов КМОП, А	I_{CC1}	-	3	U_{CC1} = 1,98 В	+25 минус 60 + 85
Динамический ток потребления буферов DDR3, А	I_{CC2}	-	3,8	U_{CC2} = 1,60 В	+25 минус 60 + 85
Динамический ток потребления ядра, А	I_{CC3}	-	26	U_{CC3} = 0,95 В	+25 минус 60 + 85

Номинальное значение напряжений питания микросхемы должно быть:

- буфер ввода/вывода КМОП: (U_{CC1}) 1,8 ± 0,18 В;
- буфер ввода/вывода DDR3: (U_{CC2}) 1,5 ± 0,10 В;
- ядро: (U_{CC3}) 0,9 ± 0,05 В;

Значения предельно-допустимых и предельных электрических режимов эксплуатации микросхем в диапазоне рабочих температур среды должны соответствовать нормам, приведенным в таблице 6.3 .

Таблица 6.3 – Нормы предельно-допустимых и предельных электрических режимов эксплуатации микросхем

Наименование параметра режима, единица измерения	**Обозна-чение**	**Норма параметра**			
		предельно-допустимого режима,		**предельного режима,**	
		не менее	**не более**	**не менее**	**не более**
Напряжение питания буферов ввода/вывода, В	U_{CC1}	1,62	1,98	минус 0,2	2,5
Напряжение питания интерфейса с памятью DDR3/DDR3L, В	U_{CC2}	1,4	1,6	минус 0,2	2,0
Напряжение питания ядра, В	U_{CC3}	0,85	0,95	минус 0,2	1,2
Напряжение на выводах «вход-выход» в состоянии «выключено», В	U_{IOZ}	0	1,98	минус 0,2	2,5
Входное напряжение высокого уровня, В	U_{IH}	1,45	1,98	-	2,5
Входное напряжение низкого уровня, В	U_{IL}	0	0,4	минус 0,2	-
Выходной ток буферов, мА	I_O	минус 4	4	минус 13	13

6.4 Временные характеристики

В данном разделе приведены временные диаграммы и временные параметры сигналов микросхемы 1879ВМ8Я. Временные параметры микросхемы определялись в полном диапазоне внешних воздействий (см. таблицу 6.4).

6.4.1 Временные диаграммы и временные параметры тактовых сигналов и сигналов общего назначения

Рисунок 6.3 - Временная диаграмма тактового сигнала и сигнала сброса процессора

Рисунок 6.4 - Временная диаграмма входов прерываний

Рисунок 6.5 - Временная диаграмма тактового сигнала Ethernet интерфейса

Рисунок 6.6 - Временная диаграмма тактового парафазного сигнала PCI Express и последовательных портов кластеров

Таблица 6.4 Временные параметры тактовых сигналов и входных сигналов общего назначения

Обозначение	Функциональное описание	Временной параметр, нс	
		Не менее	Не более
T_{CP}	Период сигнала тактового сигнала CLK_REFI		62,50
t_{CH}	Длительность сигнала высокого уровня на входе CLK_REFI	40% периода тактового сигнала	60% периода тактового сигнала
t_{CL}	Длительность сигнала низкого уровня на входе CLK_REFI	40% периода тактового сигнала	60% периода тактового сигнала
t_{FCUP}	Длительность фронта сигнала CLK_REFI		0,2
t_{FCDN}	Длительность среза сигнала CLK_REFI		0,2
T_{RESET}	Длительность сигнала системного сброса процессора	50∗P	
T_{INT}	Длительность сигнала низкого уровня на входе маскируемых прерываний	1,5∗P	
T_{ETH}	Период сигнала тактового сигнала ETH_REFCLK		20
T_{ETH}	Длительность сигнала высокого уровня на входе ETH_REFCLK	40% периода тактового сигнала	60% периода тактового сигнала
T_{ETHL}	Длительность сигнала низкого уровня на входе ETH_REFCLK	40% периода тактового сигнала	60% периода тактового сигнала
t_{FETHUP}	Длительность фронта сигнала ETH_REFCLK		0,2
t_{FETHDN}	Длительность среза сигнала ETH_REFCLK		0,2

Примечание: Параметр P равен периоду выбранного тактового сигнала CLK_REFI.

Продолжение таблицы 6.4

Обозначение	Функциональное описание	Временной параметр, нс	
T_{ELH}	Период сигнала тактового сигнала ELC_CLKREF		10
T_{ELCH}	Длительность сигнала высокого уровня на входе ELC_CLKREF	40% периода тактового сигнала	60% периода тактового сигнала
T_{ELCL}	Длительность сигнала низкого уровня на входе ELC_CLKREF	40% периода тактового сигнала	60% периода тактового сигнала
t_{FECLUP}	Длительность фронта сигнала ELC_CLKREF		0,2
t_{FECLDN}	Длительность среза сигнала ELC_CLKREF		0,2

6.5 Порядок подачи питания при включении

Последовательность подачи напряжений питания и сигналов при включении:

1) Напряжение питания для ядра U_{CC3} (0,9 В);
2) Напряжение питание для аналоговых блоков (0,9 В и 1,8 В);
3) Напряжение питания для буферов ввода/вывода: U_{CC1} (1,8 В) и U_{CC2} (1,5 В);
4) Тактовый сигнал CLK_REFI;
5) системный сброс XRSTn.

Последовательность подачи остальных сигналов не регламентируется. Выключение напряжений питания производится в порядке, обратном включению.

Лист регистрации изменений

Изм.	Номера страниц				Всего страниц в документе	№ документа	Входящий № сопроводительного докум. и дата	Подпись	Дата
	измененных	заменнных	новых	Аннулированных					
3		все					ЮФКВ.016-2018	Харламова	28.02.2018