

# **МИКРОСХЕМА ИНТЕГРАЛЬНАЯ 1892ВМ7Я**

## **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

## ОГЛАВЛЕНИЕ

<b>1. ВВЕДЕНИЕ</b> .....	<b>9</b>
1.1 НАЗНАЧЕНИЕ .....	9
1.2 ФУНКЦИОНАЛЬНЫЕ ПАРАМЕТРЫ И ВОЗМОЖНОСТИ .....	10
1.3 СТРУКТУРНАЯ СХЕМА .....	13
1.4 ИНСТРУМЕНТАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ .....	15
1.5 ОПЕРАЦИОННАЯ СИСТЕМА ДЛЯ МИКРОСХЕМЫ 1892ВМ7Я .....	15
<b>2. ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР</b> .....	<b>16</b>
2.1 ОСНОВНЫЕ ХАРАКТЕРИСТИКИ CPU .....	16
2.2 БЛОК СХЕМА .....	16
2.3 СОСТАВЛЯЮЩИЕ ЛОГИЧЕСКИЕ БЛОКИ .....	17
2.3.1 Устройство исполнения.....	17
2.3.2 Устройство умножения/деления (MDU) .....	17
2.3.3 Системный управляющий сопроцессор .....	18
2.3.4 Сопроцессор арифметики в формате с плавающей точкой (FPU) .....	18
2.3.5 Устройство управления памятью (MMU).....	18
2.3.6 Контроллер кэш.....	18
2.3.7 Устройство шинного интерфейса (BIU – Bus Interface Unit).....	18
2.3.8 ОпCD контроллер.....	18
2.4 КОНВЕЙЕР .....	19
2.4.1 Стадии конвейера.....	19
2.4.2 Операции умножения и деления .....	20
2.4.3 Задержка выполнения команд перехода (Jump, Branch).....	20
2.4.4 Обходные пути передачи данных (Data bypass) .....	21
2.4.5 Задержка загрузки данных .....	22
2.5 СОПРОЦЕССОР АРИФМЕТИКИ В ФОРМАТЕ С ПЛАВАЮЩЕЙ ТОЧКОЙ (FPU).....	23
2.5.1 Введение.....	23
2.5.2 Регистры FPU .....	23
2.5.3 Исключения FPU.....	30
2.5.4 Время выполнения команд FPU.....	33
2.6 УСТРОЙСТВО УПРАВЛЕНИЯ ПАМЯТЬЮ (MMU).....	34
2.6.1 Введение.....	34
2.6.2 Режимы работы.....	35
2.6.3 Буфер быстрого преобразования адреса (TLB) .....	41
2.6.4 Преобразование виртуального адреса в физический в режиме TLB. ....	43
2.7 ИСКЛЮЧЕНИЯ.....	47
2.7.1 Условия исключений.....	48
2.7.2 Приоритеты исключений .....	48
2.7.3 Расположение векторов исключений .....	49
2.7.4 Обработка общих исключений .....	49
2.7.5 Исключения.....	51
2.7.6 Алгоритмы обработки исключений.....	56
2.8 РЕГИСТРЫ CPU.....	59
2.8.1 Назначение .....	59
2.8.2 Обзор регистров CPU .....	59
2.8.3 Регистры CPU.....	60
2.9 КЭШ .....	76
2.10 КАРТА ПАМЯТИ CPU .....	77
<b>3. ЦИФРОВОЙ СИГНАЛЬНЫЙ ПРОЦЕССОР</b> .....	<b>93</b>
3.1 ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ DSP-КЛАСТЕРА QELCORE-28.....	93
3.2 СТРУКТУРНАЯ СХЕМА .....	93
3.2.1 Интерфейсы DSP-кластера QELcore-28.....	94
3.2.2 Организация работы DSP-кластера QELcore-28.....	95
3.3 ОРГАНИЗАЦИЯ ОБЩЕГО ПОЛЯ ПАМЯТИ ДАННЫХ.....	96
3.3.1 Карта памяти .....	97

3.3.2	Дисциплина обработки одновременных обращений к общему полю памяти данных со стороны DSP-ядер (арбитраж) .....	98
3.4	РЕГИСТРЫ УПРАВЛЕНИЯ И СОСТОЯНИЯ QELCORE-28 .....	99
3.4.1	Регистр маски прерываний ( <i>MASKR_DSP</i> ) .....	99
3.4.2	Регистр запросов прерываний ( <i>QSTR_DSP</i> ) .....	99
3.4.3	Регистр управления и состояния ( <i>CSR_DSP</i> ) .....	100
3.4.4	Счетчик тактов в состоянии RUN ( <i>TOTAL_RUN_CNTR</i> ) .....	101
3.4.5	Счетчик тактов ( <i>TOTAL_CLK_CNTR</i> ) .....	101
3.5	БУФЕР ОБМЕНА XBUF .....	101
3.5.1	Регистр флагов обмена ( <i>EFR</i> ) .....	101
3.5.2	Режимы обменов с XBUF .....	102
3.6	ПРОГРАММНАЯ МОДЕЛЬ DSP-ЯДРА ELCORE-28 .....	102
3.7	РЕГИСТРЫ ALU .....	102
3.7.1	Регистровый файл .....	104
3.7.2	Регистры-аккумуляторы .....	105
3.7.3	Регистр PDNR .....	106
3.7.4	Регистр CCR .....	106
3.8	РЕГИСТРЫ АДРЕСНЫХ ГЕНЕРАТОРОВ И ВИДЫ АДРЕСНОЙ АРИФМЕТИКИ .....	108
3.8.1	Регистры AGU .....	108
3.8.2	Регистры AGU-Y .....	108
3.8.3	Назначение регистров адресных генераторов .....	109
3.8.4	Особенности X- и Y- указателей .....	111
3.8.5	Начальное состояние регистров адреса A0-A7, AT .....	112
3.8.6	Регистр адреса вектора прерывания IVAR .....	112
3.9	РЕГИСТРЫ УСТРОЙСТВА ПРОГРАММНОГО УПРАВЛЕНИЯ PCU .....	112
3.9.1	Регистр управления и состояния DCSR .....	113
3.9.2	Программный счетчик PC .....	113
3.9.3	Регистр состояния SR .....	113
3.9.4	Регистр-идентификатор IDR .....	115
3.9.5	Регистр адреса окончания цикла LA .....	115
3.9.6	Регистр счетчика циклов LC .....	115
3.9.7	Стеки SS, CSL, CSH .....	116
3.9.8	Регистр указателей стека SP .....	116
3.9.9	Регистры адреса останова SAR, SAR1-SAR7 .....	116
3.9.10	Счетчик команд CNTR .....	116
3.9.11	Регистры управления прерываниями и DMA-обменами .....	117
3.9.12	Механизм обработки прерываний .....	117
3.9.13	Регистр запросов на прерывание DSP ( <i>IRQR</i> ) .....	117
3.9.14	Регистры масок запросов на прерывание DSP ( <i>IMASKR, QMASKR0, QMASKR1, QMASKR2</i> ) .....	118
3.9.15	Регистр запуска DMA со стороны DSP ( <i>DSTART</i> ) .....	118
3.9.16	Регистр таймера ( <i>TMR</i> ) .....	119
3.9.17	Регистр управления локальным арбитражем ( <i>ARBR</i> ) .....	119
3.9.18	Регистр специфических функций ( <i>SFR</i> ) .....	121
3.10	ПРОГРАММНЫЙ КОНВЕЙЕР DSP-ЯДРА ELCORE-28 .....	122
3.11	ПЕРЕЧЕНЬ АДРЕСУЕМЫХ РЕГИСТРОВ DSP-КЛАСТЕРА .....	124
<b>4.</b>	<b>СИСТЕМНОЕ УПРАВЛЕНИЕ .....</b>	<b>129</b>
4.1	СИСТЕМА СИНХРОНИЗАЦИИ .....	129
4.1.1	Входы синхронизации и умножители частоты .....	129
4.1.2	Управление работой PLL .....	130
4.1.3	Отключение и включение тактовой частоты .....	131
4.2	КОНТРОЛЛЕР ПРЕРЫВАНИЙ .....	133
4.3	СИСТЕМНЫЕ РЕГИСТРЫ .....	138
4.4	ПРОЦЕДУРА НАЧАЛЬНОЙ ЗАГРУЗКИ .....	139
4.5	ЛОГИКА ВЗАИМОДЕЙСТВИЯ CPU И DSP .....	139
4.5.1	Функции CPU .....	139
4.5.2	Функции DSP .....	140
<b>5.</b>	<b>ИНТЕРВАЛЬНЫЙ ТАЙМЕР .....</b>	<b>141</b>
5.1	НАЗНАЧЕНИЕ .....	141

5.2	СТРУКТУРНАЯ СХЕМА ИТ .....	141
5.3	ОПИСАНИЕ РЕГИСТРОВ ИНТЕРВАЛЬНОГО ТАЙМЕРА .....	142
5.4	ПРОГРАММИРОВАНИЕ ИТ .....	143
<b>6.</b>	<b>СТОРОЖЕВОЙ ТАЙМЕР .....</b>	<b>144</b>
6.1	НАЗНАЧЕНИЕ .....	144
6.2	СТРУКТУРНАЯ СХЕМА .....	144
6.3	ОПИСАНИЕ РЕГИСТРОВ WDT .....	145
6.4	ПРОГРАММИРОВАНИЕ WDT .....	146
<b>7.</b>	<b>КОНТРОЛЛЕР ПРЯМОГО ДОСТУПА В ПАМЯТЬ (DMA) .....</b>	<b>149</b>
7.1	ПЕРЕЧЕНЬ КАНАЛОВ DMA .....	149
7.2	ОРГАНИЗАЦИЯ ОБМЕНА ДАННЫМИ В МИКРОСХЕМЕ .....	149
7.3	КАНАЛЫ DMA ТИПА ПАМЯТЬ - ПАМЯТЬ .....	150
7.4	КАНАЛЫ DMA ПЕРИФЕРИЙНЫХ ПОРТОВ .....	156
7.5	ПРОЦЕДУРА САМОИНИЦИАЛИЗАЦИИ .....	158
7.6	ПРЕРЫВАНИЯ DMA .....	159
<b>8.</b>	<b>ПОРТ ВНЕШНЕЙ ПАМЯТИ ОБЩЕГО НАЗНАЧЕНИЯ .....</b>	<b>160</b>
8.1	ВВЕДЕНИЕ .....	160
8.2	СОСТАВ И НАЗНАЧЕНИЕ РЕГИСТРОВ ПОРТА ВНЕШНЕЙ ПАМЯТИ .....	160
8.2.1	<i>Регистр конфигурации CSCON0</i> .....	161
8.2.2	<i>Регистр конфигурации CSCON1</i> .....	162
8.2.3	<i>Регистр конфигурации CSCON2</i> .....	162
8.2.4	<i>Регистр конфигурации CSCON3</i> .....	163
8.2.5	<i>Регистр конфигурации CSCON4</i> .....	164
8.2.6	<i>Регистр конфигурации SDRCON</i> .....	164
8.2.7	<i>Регистр параметров SDRTMR</i> .....	167
8.2.8	<i>Регистр состояний и управления SDRCSR</i> .....	168
8.2.9	<i>Регистр FLY_WS</i> .....	170
8.3	ВРЕМЕННЫЕ ДИАГРАММЫ ОБМЕНА ДАННЫМИ .....	171
8.3.1	<i>Общие положения</i> .....	171
8.3.2	<i>Обмен данными с асинхронной памятью</i> .....	172
8.3.3	<i>Обмен данными в режиме Flyby</i> .....	180
8.3.4	<i>Обмен данными с синхронной статической памятью</i> .....	184
8.4	РЕКОМЕНДАЦИИ ПО ПОДКЛЮЧЕНИЮ ВНЕШНЕЙ ПАМЯТИ .....	185
8.4.1	<i>Память типа SDRAM</i> .....	185
8.4.2	<i>Память типа NOR FLASH</i> .....	185
<b>9.</b>	<b>ПОРТ ВНЕШНЕЙ ПАМЯТИ ТИПА DDR SDRAM .....</b>	<b>186</b>
9.1	ОБЩИЕ ПОЛОЖЕНИЯ .....	186
9.2	РЕГИСТРЫ DDR_PORT .....	186
9.2.1	<i>Регистр конфигурации DDRAM</i> .....	187
9.2.2	<i>Регистр базового адреса DDR_BAR</i> .....	188
9.2.3	<i>Регистр параметров DDRAM DDR_TMR</i> .....	188
9.2.4	<i>Регистр состояний и управления DDR_CSR</i> .....	189
9.2.5	<i>Регистр режимов DDR_MOD</i> .....	191
<b>10.</b>	<b>УНИВЕРСАЛЬНЫЙ АСИНХРОННЫЙ ПОРТ (UART) .....</b>	<b>192</b>
10.1	ОБЩИЕ ПОЛОЖЕНИЯ .....	192
10.2	РЕГИСТРЫ UART .....	193
10.2.1	<i>Общие положения</i> .....	193
10.2.2	<i>Регистр LCR</i> .....	194
10.2.3	<i>Регистр FCR</i> .....	195
10.2.4	<i>Регистр LSR</i> .....	195
10.2.5	<i>Регистр IER</i> .....	197
10.2.6	<i>Регистр IIR</i> .....	197
10.2.7	<i>Регистр MCR</i> .....	199
10.2.8	<i>Программируемый генератор скорости обмена</i> .....	199
10.3	РАБОТА С FIFO ПО ПРЕРЫВАНИЮ .....	199

10.4	РАБОТА С FIFO ПО ОПРОСУ .....	200
<b>11.</b>	<b>КОНТРОЛЛЕР ИНТЕРФЕЙСА SERIAL RAPIDIO (SRIO).....</b>	<b>201</b>
11.1	ОБЩИЕ ПОЛОЖЕНИЯ.....	201
11.2	СТРУКТУРНАЯ СХЕМА .....	202
11.3	РЕГИСТРЫ SRIO .....	203
11.3.1	Общие положения.....	203
11.3.2	Регистры системные .....	207
11.3.3	Регистры устройства выполнения операций ввода-вывода (LSU).....	214
11.3.4	Регистры устройства MPU.....	226
11.3.5	Архитектурные регистры логического и транспортного уровней RapidIO .....	234
11.3.6	Архитектурные регистры физического уровня RapidIO.....	239
11.3.7	Дополнительные регистры физического уровня .....	243
11.4	УСТРОЙСТВО ВЫПОЛНЕНИЯ ОПЕРАЦИЙ ВВОДА-ВЫВОДА (LSU).....	263
11.4.1	Общие положения.....	263
11.4.2	Описание операций ввода-вывода .....	263
11.4.3	Выполнение операций ввода-вывода .....	272
11.5	УСТРОЙСТВО ВЫПОЛНЕНИЯ ОПЕРАЦИЙ ПЕРЕДАЧИ СООБЩЕНИЙ (MPU).....	280
11.5.1	Общие положения.....	280
11.5.2	Описание операций передачи сообщений .....	280
11.5.3	Прием сообщений.....	282
11.5.4	Передача сообщений .....	288
11.6	ФОРМИРОВАНИЕ И ОБРАБОТКА ПРЕРЫВАНИЙ.....	293
11.7	ИНИЦИАЛИЗАЦИЯ SRIO .....	294
11.7.1	Запуск инициализации SRIO.....	294
11.7.2	Потеря инициализации SRIO.....	295
<b>12.</b>	<b>КОНТРОЛЛЕР ИНТЕРФЕЙСА SPACEWIRE (SWIC).....</b>	<b>298</b>
12.1	ОБЩИЕ ПОЛОЖЕНИЯ.....	298
12.2	БЛОК-СХЕМА .....	298
12.3	ПРЕРЫВАНИЯ.....	300
12.4	ПЕРЕЧЕНЬ РЕГИСТРОВ SWIC.....	301
12.4.1	Общие положения.....	301
12.5	ОПИСАНИЕ РЕГИСТРОВ SWIC .....	301
12.5.1	Регистр HW_VER .....	301
12.5.2	Регистр STATUS.....	302
12.5.3	Регистр RX_CODE.....	304
12.5.4	Регистр MODE_CR.....	304
12.5.5	Регистр TX_SPEED.....	305
12.5.6	Регистр TX_CODE .....	306
12.5.7	Регистр RX_SPEED.....	307
12.5.8	Регистр CNT_RX_PACK .....	307
12.5.9	Регистр CNT_RX0_PACK .....	307
12.5.10	Регистр ISR_L.....	307
12.5.11	Регистр ISR_H.....	308
12.5.12	Регистр TRUE_TIME.....	308
12.5.13	Регистр TOUT_CODE.....	308
12.5.14	Регистр ISR_tout_L.....	309
12.5.15	Регистр ISR_tout_H.....	309
12.5.16	Регистр LOG_ADDR .....	309
12.6	РАБОТА СО SWIC. ПАКЕТЫ ДАННЫХ, ДЕСКРИПТОРЫ ПАКЕТОВ.....	309
12.6.1	Расположение данных в памяти.....	310
12.6.2	Схема обработки данных процессором.....	310
12.6.3	Прием данных из канала SpaceWire.....	310
12.6.4	Передача данных в канал SpaceWire .....	312
12.6.5	Выравнивание границ пакетов по границам слов .....	314
12.6.6	Формат дескриптора пакета.....	315
12.6.7	Возможность передачи коммуникационного пакета.....	315
12.6.8	Использование симплексного режима .....	316
12.6.9	Маркеры времени.....	317

12.6.10	Коды распределенных прерываний.....	318
12.6.11	Коды подтверждения распределенных прерываний.....	318
12.6.12	Установка скорости передачи данных.....	318
12.6.13	Установление соединения.....	319
12.6.14	Определение скорости приема данных.....	319
<b>13.</b>	<b>КОНТРОЛЛЕР ETHERNET MAC 10/100.....</b>	<b>320</b>
13.1	ВВЕДЕНИЕ.....	320
13.1.1	Назначение.....	320
13.1.2	Основные характеристики.....	320
13.1.3	Особенности использования.....	320
13.2	ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ.....	321
13.2.1	Структурная схема.....	321
13.3	ПРОГРАММНАЯ МОДЕЛЬ.....	323
13.3.1	Программирование контроллера Ethernet MAC 10/100.....	323
13.3.2	Регистры контроллера Ethernet MAC 10/100.....	349
<b>14.</b>	<b>КОНТРОЛЛЕР ШИНЫ PCI.....</b>	<b>360</b>
14.1	ФУНКЦИОНАЛЬНЫЕ ПАРАМЕТРЫ И ВОЗМОЖНОСТИ.....	360
14.2	СТРУКТУРНАЯ СХЕМА.....	360
14.3	РЕГИСТРЫ.....	361
14.3.1	Конфигурационные регистры.....	363
14.3.2	Регистры управления обменом данных.....	365
14.4	ОБМЕН ДАННЫМИ ПО ШИНЕ PCI В РЕЖИМЕ TARGET (SLAVE).....	371
14.5	ОБМЕН ДАННЫМИ С ШИНОЙ PCI В РЕЖИМЕ MASTER.....	372
14.6	ОБМЕН ДАННЫМИ С ШИНОЙ PCI ЧЕРЕЗ АДРЕСНОЕ ОКНО.....	373
14.7	ПЕРЕДАЧА ПРЕРЫВАНИЙ.....	374
14.7.1	Передача вектора прерывания из шины PCI.....	374
14.7.2	Передача прерываний в шину PCI.....	375
14.7.3	Регистры управления прерываниями.....	375
14.8	ПЕРЕДАЧА СТАРТОВОГО АДРЕСА.....	376
14.9	АРБИТР.....	376
<b>15.</b>	<b>КОНТРОЛЛЕР ШИНЫ I2C.....</b>	<b>377</b>
15.1	НАЗНАЧЕНИЕ.....	377
15.2	ОСНОВНЫЕ ХАРАКТЕРИСТИКИ.....	377
15.3	СТРУКТУРНАЯ СХЕМА.....	377
15.4	РЕГИСТРЫ КОНТРОЛЛЕРА I2C.....	378
15.4.1	Регистр PRER.....	378
15.4.2	Регистр CTR.....	379
15.4.3	Регистр TXR.....	379
15.4.4	Регистр RXR.....	379
15.4.5	Регистр CR.....	379
15.4.6	Регистр SR.....	380
15.4.7	Регистр PR_CNT.....	380
15.4.8	Регистр SYNC.....	381
15.5	ФУНКЦИОНИРОВАНИЕ КОНТРОЛЛЕРА I2C.....	381
15.6	ПРОГРАММИРОВАНИЕ КОНТРОЛЛЕРА I2C.....	382
<b>16.</b>	<b>ПОРТ ВВОДА ВИДЕОДАНЫХ VPIN.....</b>	<b>385</b>
16.1	НАЗНАЧЕНИЕ.....	385
16.2	АРХИТЕКТУРА И ФУНКЦИОНИРОВАНИЕ ПОРТА VPIN.....	385
16.3	ПРОГРАММНО-ДОСТУПНЫЕ РЕГИСТРЫ.....	387
16.3.1	Регистр управления и состояния (CSR).....	387
16.3.2	Регистр - счетчик строк/ счетчик пикселей (Line_cnt/Pix_cnt).....	388
16.3.3	Регистр - счетчик кадров (Frame_cnt).....	388
16.4	РЕЖИМЫ РАБОТЫ ПОРТА VPIN.....	388
16.4.1	Способы интерпретации входных видеоданных.....	388
16.4.2	Упаковка цветовых компонент.....	390
16.4.3	Режим съемки одного кадра (Snapshot).....	391



16.4.4	Режим декодирования маркеров VT.656 .....	391
<b>17.</b>	<b>ПОРТ ВЫВОДА ВИДЕОДАНЫХ VPOUT .....</b>	<b>393</b>
17.1	НАЗНАЧЕНИЕ .....	393
17.2	АРХИТЕКТУРА И ФУНКЦИОНИРОВАНИЕ ПОРТА VPOUT .....	393
17.3	ПРОГРАММНО-ДОСТУПНЫЕ РЕГИСТРЫ .....	395
17.3.1	Регистр управления и состояния (CSR) .....	395
17.3.2	Регистр периода сигнала VCLKO_out (DIV) .....	396
17.3.3	Регистр начала/конца активной части строки (Hstart/Hend) .....	396
17.3.4	Регистр начала/конца активной части кадра (Vstart/Vend) .....	397
17.3.5	Регистр - счетчик строк/ счетчик пикселей (Line_cnt/Pix_cnt) .....	397
17.3.6	Регистр - счетчик кадров (Frame_cnt) .....	397
17.3.7	Регистр начала/конца сигнала VDEN в строке (EHstart/EHend) .....	397
17.3.8	Регистр начала/конца активной части кадра (Vstart/Vend) .....	398
17.4	РЕЖИМЫ РАБОТЫ ПОРТА .....	398
17.4.1	Режим RGB .....	398
17.4.2	Режим i80 .....	401
17.4.3	Режим GPIO .....	401
<b>18.</b>	<b>МНОГОФУНКЦИОНАЛЬНЫЙ БУФЕРИЗИРОВАННЫЙ ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP).....</b>	<b>402</b>
18.1	ОСОБЕННОСТИ MFBSP .....	402
18.1.1	Основные характеристики MFBSP в режиме I2S .....	403
18.1.2	Основные характеристики MFBSP в режиме SPI .....	404
18.1.3	Основные характеристики MFBSP в режиме LPORT .....	405
18.1.4	Основные характеристики MFBSP в режиме порта ввода-вывода общего назначения .....	406
18.2	ОБЩИЕ СВЕДЕНИЯ ОБ MFBSP .....	406
18.2.1	Режимы работы MFBSP .....	406
18.2.2	Структурная схема многофункционального буферизированного последовательного порта .....	407
18.2.3	Назначение выводов порта в различных режимах .....	408
18.2.4	Перечень регистров MFBSP .....	410
18.2.5	Каналы DMA многофункциональных портов MFBSP .....	410
18.2.6	Прерывания от каналов DMA MFBSP .....	411
18.2.7	Прерывания от MFBSP .....	411
18.3	РАБОТА MFBSP В РЕЖИМЕ I2S .....	413
18.3.1	Назначение MFBSP в режиме I2S .....	413
18.3.2	Регистр управления и состояния CSR_MFBSP (режим I2S) .....	413
18.3.3	Регистр управления направлением выводов DIR_MFBSP (режим I2S) .....	414
18.3.4	Регистр управления приёмником RCTR (режим I2S) .....	415
18.3.5	Регистр управления передатчиком TCTR (режим I2S) .....	417
18.3.6	Регистр состояния приёмника RSR (режим I2S) .....	419
18.3.7	Регистр состояния передатчика TSR (режим I2S) .....	420
18.3.8	Регистр управления темпом приёма RCTR_RATE (режим I2S) .....	421
18.3.9	Регистр управления темпом передачи TCTR_RATE (режим I2S) .....	421
18.3.10	Псевдорегистр TSTART (режим I2S) .....	422
18.3.11	Псевдорегистр RSTART (режим I2S) .....	422
18.3.12	Регистр аварийного управления портом EMERG_MFBSP (режим I2S) .....	422
18.3.13	Регистр маски прерываний от порта IMASK (режим I2S) .....	423
18.3.14	Структурная схема MFBSP для режима I2S .....	424
18.3.15	Варианты соединения порта с внешними устройствами .....	425
18.3.16	Передача данных в режиме I2S .....	426
18.3.17	Формирование тактовых сигналов приёмника (RCLK) и передатчика (TCLK) .....	429
18.3.18	Формирование управляющих сигналов приёмника и передатчика в режиме I2S .....	430
18.3.19	Тракт передачи данных .....	431
18.3.20	Тракт приёма данных .....	432
18.3.21	Прерывания от последовательного порта .....	433
18.4	РАБОТА MFBSP В РЕЖИМЕ SPI .....	433
18.4.1	Назначение последовательного порта в режиме SPI .....	433
18.4.2	Регистр управления и состояния CSR_MFBSP (режим SPI) .....	435
18.4.3	Регистр управления направлением выводов DIR_MFBSP (режим SPI) .....	435

18.4.4	Регистр управления приёмником RCTR (режим SPI) .....	436
18.4.5	Регистр управления передатчиком TCTR (режим SPI).....	438
18.4.6	Регистр состояния приёмника RSR (режим SPI).....	439
18.4.7	Регистр состояния передатчика TSR (режим SPI).....	440
18.4.8	Регистр управления темпом приёма RCTR_RATE (режим SPI).....	441
18.4.9	Регистр управления темпом передачи TCTR_RATE (режим SPI).....	441
18.4.10	Псевдорегистр TSTART (режим SPI) .....	442
18.4.11	Псевдорегистр RSTART (режим SPI) .....	442
18.4.12	Регистр аварийного управления портом EMERG_MFBSP (режим SPI) .....	442
18.4.13	Регистр маски прерываний от порта IMASK (режим SPI).....	443
18.4.14	Структурная схема MFBSP для режима SPI.....	444
18.4.15	Варианты соединения порта с внешними устройствами.....	445
18.4.16	Передача данных в режиме SPI .....	446
18.4.17	Пример чтения 8 разрядного слова по заданному адресу из ведомого устройства с интерфейсом C-BUS.....	448
18.4.18	Формирование тактовых сигналов приёмника (RSCCK) и передатчика (TSCCK).....	449
18.4.19	Формирование управляющих сигналов приёмника и передатчика в режиме SPI.....	450
18.4.20	Тракт передачи данных.....	451
18.4.21	Тракт приёма данных.....	453
18.4.22	Прерывания от последовательного порта .....	454
18.5	РАБОТА MFBSP В РЕЖИМЕ ЛИНКОВОГО ПОРТА (LPORT) .....	454
18.5.1	Назначение линкового порта .....	454
18.5.2	Регистр управления и состояния CSR_MFBSP (режим LPORT).....	454
18.5.3	Регистр состояния приёмника RSR (режим LPORT) .....	455
18.5.4	Регистр состояния передатчика TSR (режим LPORT) .....	456
18.5.5	Регистр аварийного управления портом EMERG_MFBSP (режим LPORT).....	457
18.5.6	Регистр маски прерываний от порта IMASK (режим LPORT).....	458
18.5.7	Структурная схема MFBSP для режима линкового порта .....	459
18.5.8	Соединение с внешними устройствами .....	459
18.5.9	Передача данных по линковому порту .....	460
18.5.10	Прерывания от линковых портов .....	462
18.6	РАБОТА MFBSP В РЕЖИМЕ ПОРТА ВВОДА-ВЫВОДА ОБЩЕГО НАЗНАЧЕНИЯ.....	462
18.6.1	Регистр данных порта ввода вывода GPIO_DR .....	462
18.6.2	Регистр управления направлением выводов DIR_MFBSP.....	463
18.7	РЕКОМЕНДАЦИИ ПО АВАРИЙНОМУ ВЫКЛЮЧЕНИЮ ПЕРЕДАТЧИКА .....	463
<b>19.</b>	<b>ПОРТ JTAG И ВСТРОЕННЫЕ СРЕДСТВА ОТЛАДКИ ПРОГРАММ .....</b>	<b>464</b>
<b>20.</b>	<b>ЭЛЕКТРИЧЕСКИЕ И ВРЕМЕННЫЕ ПАРАМЕТРЫ .....</b>	<b>465</b>
20.1	ЭЛЕКТРОПИТАНИЕ.....	465
20.2	ЭЛЕКТРИЧЕСКИЕ ПАРАМЕТРЫ .....	465
20.3	ДИНАМИЧЕСКАЯ ПОТРЕБЛЯЕМАЯ МОЩНОСТЬ.....	466
20.4	ПРЕДЕЛЬНО-ДОПУСТИМЫЕ И ПРЕДЕЛЬНЫЕ ЭЛЕКТРИЧЕСКИЕ РЕЖИМЫ ЭКСПЛУАТАЦИИ.....	467
20.5	ВРЕМЕННЫЕ ПАРАМЕТРЫ.....	468
20.6	РЕКОМЕНДАЦИИ ПО ПОДКЛЮЧЕНИЮ КВАРЦЕВОГО РЕЗОНАТОРА.....	469
<b>21.</b>	<b>ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ.....</b>	<b>470</b>
<b>22.</b>	<b>ИСТОРИЯ ИЗМЕНЕНИЙ .....</b>	<b>483</b>
22.1	20 ДЕКАБРЯ 2012 Г.....	483
22.2	31 ИЮЛЯ 2013 Г .....	483
22.3	08 АПРЕЛЯ 2015 Г .....	483



# 1. ВВЕДЕНИЕ

## 1.1 Назначение

Микросхема интегральная сигнального микропроцессора 1892ВМ7Я спроектирована как однокристалльная пятипроцессорная «система на кристалле» на базе IP-ядерной (IP-intellectual property) платформы «МУЛЬТИКОР», разработанной в ОАО НПЦ «ЭЛВИС».

В качестве пяти процессоров микросхема 1892ВМ7Я содержит 32-разрядный центральный процессор (CPU – Central Processing Unit) и четыре высокопроизводительных процессоров-акселераторов для цифровой обработки сигналов (DSP – Digital Signal Processing) с плавающей/фиксированной точкой, обеспечивающий обработку информации с переменными форматами данных от битовых форматов, до стандартных форматов данных с плавающей точкой в формате IEEE754.

Все пять процессоров работают независимо друг от друга (каждый по своей собственной программе) и вследствие этого представляют систему на кристалле MIMD – архитектуры (MIMD – Multiple Instructions Multiple Data).

Микросхема 1892ВМ7Я сочетает в себе лучшие качества двух классов приборов: микроконтроллеров и цифровых процессоров обработки сигналов, что особенно важно для микроминиатюрных встраиваемых применений, когда приходится решать в рамках ограниченных габаритов одновременно обе задачи: управления и высокоточной обработки информации, включая сигналы и изображение.

Для разработчика системы обеспечивается уникальная возможность применения новых алгоритмов принятия решений в CPU на основе параллельно выполняемых процедур адаптивного анализа и обработки сигналов в DSP, что реализуется в пределах одной и той же микросхемы.

Для этих целей разработаны методы применения RLS/LNS алгоритмов на базе микросхем серий «МУЛЬТИКОР», в частности для адаптивных антенных решеток.

Микросхема 1892ВМ7Я обеспечивает работу под операционной системой Linux.

Микросхема 1892ВМ7Я предназначена для применения в следующих приложениях:

- Радиолокационные и гидроакустические системы;
- Графические ускорители;
- Телекоммуникации и мультимедиа: базовые станции, DVB – приемники и т.д.
- Сигнальная обработка: БПФ, фильтрация, корреляция, быстрая свертка.
- Управление объектами с использованием высокоточных адаптивных методов;
- Системы промышленного контроля;
- Высокоточная обработка сигналов и данных.

## 1.2 Функциональные параметры и возможности

Микросхема 1892ВМ7Я имеет следующие функциональные параметры и возможности:

- Центральный процессор (CPU):
  - Архитектура – MIPS32;
  - 32-х битные шины передачи адреса и данных;
  - Кэш команд объемом 16 Кбайт;
  - Кэш данных объемом 16 Кбайт;
  - Архитектура привилегированных ресурсов в стиле ядра R4000:
    - Регистры Count/Compare для прерываний реального времени;
    - Отдельный вектор обработки исключений по прерываниям;
  - Программируемое устройство управления памятью:
    - Два режима работы – с TLB (Translation Lookaside Buffer) и FM (Fixed Mapped);
    - 16 строк в режиме TLB.
  - Устройство умножения и деления;
  - Сопроцессор арифметики в формате с плавающей точкой;
  - JTAG IEEE 1149.1, встроенные средства отладки программ
  - Оперативная память центрального процессора (CRAM) объемом 128 Кбайт;
  - 5 внешних запросов прерывания, в том числе немаскируемое прерывание (NMI).
- Цифровой сигнальный процессор (DSP):
  - MIMD (Multiple Instruction Multiple Data) организация потоков команд и данных;
  - Кластер (симметричный мультипроцессор) QELcore-28, состоящий из 4-х DSP-ядер ELcore-28 (DSP0 – DSP3), работает на общем поле памяти данных, содержит набор общих для всего кластера регистров управления и состояния, а также буфер обмена данными XBUF;
  - Набор инструкций, совмещающий процедуры обработки и пересылки;
  - Расширенные возможности по динамическому диапазону обрабатываемых данных, позволяющие обрабатывать данные в 8/16/32-разрядных форматах с фиксированной точкой, либо в одном из форматов с плавающей точкой – 24E8 (стандарт IEEE 754) или 32E16 (расширенный формат). Обеспечение при этом компромиссного выбора между точностью и производительностью. Аппаратные меры повышения точности и динамического диапазона (блочная плавающая точка; режим насыщения; инструкции преобразования форматов);
  - Аппаратная поддержка программных циклов;
  - Память программ PRAM объемом 32 Кбайт на каждое ядро DSP;
  - Двухпортовая оперативная память данных объемом 512 Кбайт (128 Кбайт на каждое ядро DSP);

- DSP за 1 такт может выполнять (пиковая производительность): 32 32-разрядных операции с плавающей точкой (IEEE 754), или 32 32-разрядных операций с фиксированной точкой, или 128 16-разрядных операций с фиксированной точкой, или 192 8-разрядных операций с фиксированной точкой.
- Порт внешней памяти (MPORT):
  - Шина данных – 64 разряда, шина адреса – 32 разряда;
  - Встроенный контроллер управления статической асинхронной памятью типа SRAM, FLASH, ROM, синхронной статической памятью типа SBSRAM, синхронной динамической памятью типа SDRAM;
  - Программное конфигурирование типа блоков памяти и их объема;
  - Программное задание циклов ожидания при обмене со статической асинхронной памятью;
  - Формирование сигналов выборки 5 блоков внешней памяти;
  - Перевод SDRAM в режим энергосбережения.
- Два порта внешней памяти типа DDR SDRAM (DDR\_PORT):
  - Шина данных – 32 разряда;
  - Пиковая пропускная способность – 1600 Мбайт/с;
  - Программное конфигурирование типа блоков памяти и их объема;
  - Перевод DDR SDRAM в режим энергосбережения.
- Контроллер шины PCI (PMSC – PCI Master-Slave controller):
  - Соответствует спецификации Local Bus Specification. Rev. 2.3;
  - Тактовая частота – от 33 до 66 МГц;
  - Разрядность – 32 разряда;
  - Режимы Master и Slave (Target);
  - 2 канала DMA;
  - Обеспечивается обмен данными между шиной PCI и любой областью памяти микропроцессора;
  - Встроенный арбитр на 5 запросов шины с циклически изменяемыми приоритетами запросов;
  - Встроенные средства для организации мультипроцессорных систем.
- Периферийные устройства:
  - два дуплексных порта по стандарту Serial RapidIO с пропускной способностью 1,25 Гбит/с каждый (SRIO0, SRIO1);
  - два дуплексных канала по стандарту SpaceWire с пропускной способностью от 2 до 300 Мбит/с каждый (SWIC0, SWIC1);
  - порт ввода видеоданных (VPIN);
  - порт вывода видеоданных (VPOUT);
  - контроллер шины I2C;
  - два многофункциональных буферизированных последовательных порта MFBSPP (Multifunctional Buffered Serial Port). Режимы работы - SPI, I2S, LPORT, GPIO;
  - контроллер Ethernet 10/100 МГц;

- 16-канальный контроллер прямого доступа (DMA) типа память-память. Поддержка 2-мерной и разрядно-инверсной адресации. Восемь внешних запросов прямого доступа. Возможность передачи данных в режиме Flyby (подобный режиму, реализованному в ADSP-TS201) между внешними устройствами и внешней памятью;
  - Контроллер прерываний;
  - универсальный асинхронный порт (UART) типа 16550;
  - два универсальных 32-разрядных таймера (IT0, IT1), интервальные/реального времени с тремя источниками входной частоты: CLK, XTI, RTCXTI;
  - 32-разрядный сторожевой таймер (WDT).
- Дополнительные возможности и особенности:
- Умножители/делители входной частоты на основе узлов фазовой автоподстройки частоты (PLL);
  - Встроенные средства отладки программ (OnCD) с портом JTAG в соответствии со стандартом IEEE 1149.1;
  - Режимы энергосбережения;
  - Поддержка операционной системы Linux;
  - Пластиковый корпус типа HSBGA-765.

### 1.3 Структурная схема

Структурная схема микросхемы 1892ВМ7Я приведена на рис. 1.1.

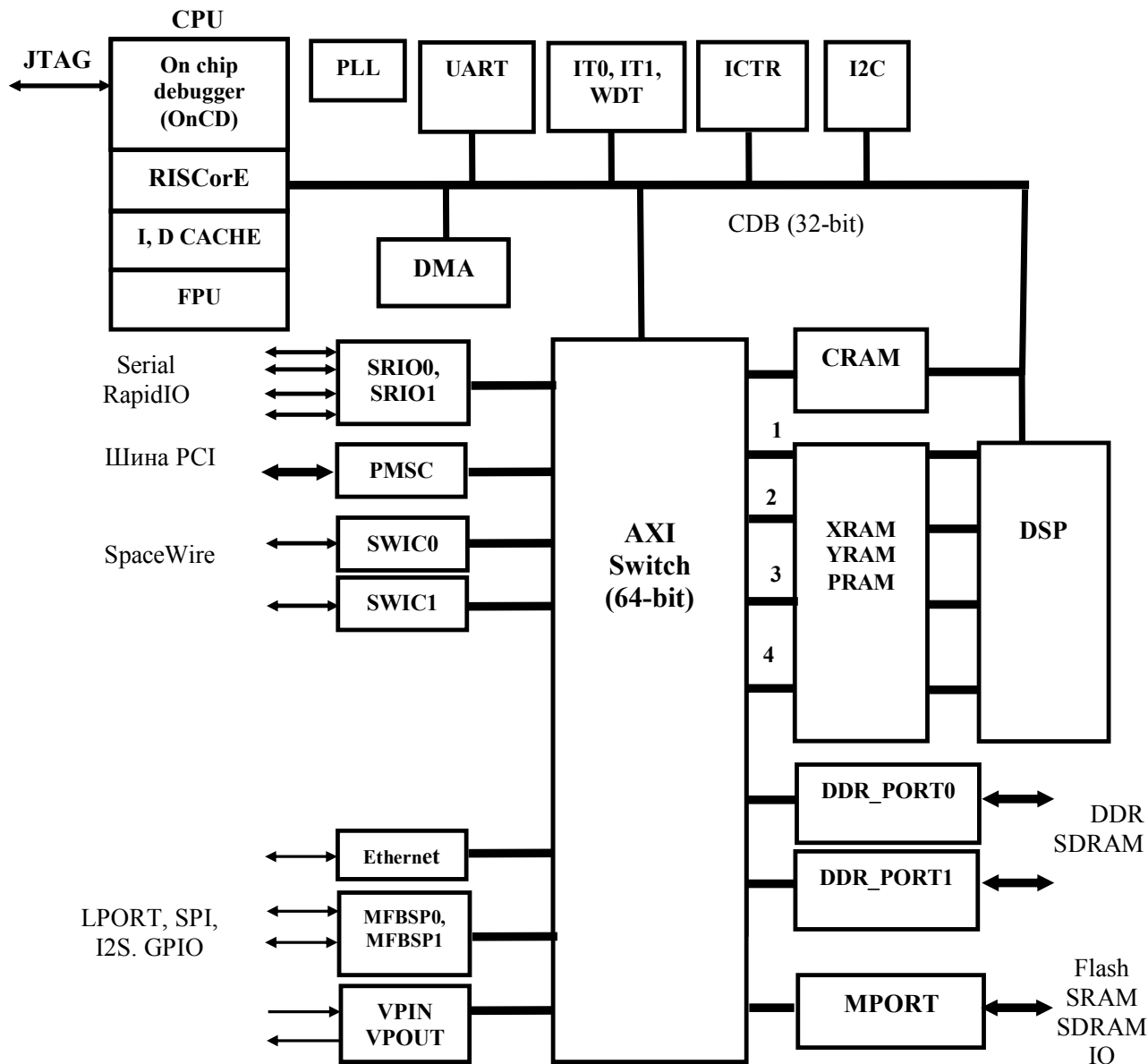


Рисунок 1.1. Структурная схема сигнального микропроцессора 1892ВМ7Я

В состав микросхемы 1892ВМ7Я входят следующие основные узлы:

- CPU – центральный процессор на основе RISC-ядра и сопроцессора с плавающей точкой (FPU);
- I, D CACHE – кэш команд и кэш данных CPU;
- DSP – цифровой сигнальный процессор;
- XRAM, YRAM, PRAM – память DSP;
- CRAM – оперативная память центрального процессора;

- CDB – шина данных CPU;
- MPORT – порт внешней памяти общего назначения;
- DDR\_PORT0, DDR\_PORT1 – порты памяти типа DDR;
- DMA – контроллер прямого доступа в память;
- OnCD – встроенные средства отладки программ;
- UART – асинхронный последовательный порт;
- AXI Switch - коммутатор;
- PLL – умножители частоты на основе PLL;
- Ethernet – контроллер Ethernet MAC 10/100 МГц;
- SWIC0, SWIC1 – контроллеры интерфейса SpaceWire;
- SRIO0, SRIO1 – контроллеры интерфейсов Serial RapidIO;
- PMSC - контроллер шины PCI;
- VPIN – порт ввода видеоданных;
- VPOUT – порт вывода видеоданных;
- I2C – контроллер шины I2C;
- MFBSP – (Multifunctional Buffered Serial Port) многофункциональный буферизированный последовательный порт (SPI, I2S, LPORT, GPIO);
- ICTR – контроллер прерываний;
- UART – универсальный асинхронный порт;
- IT0, IT1 – универсальные таймеры, интервальные/реального времени;
- WDT – сторожевой таймер;
- JTAG – отладочный порт.

Коммутатор обеспечивает передачу данных между любым исполнительным устройством (Slave) и любым задатчиком (Master). При этом процесс передачи данных между любыми парами Slave ↔ Master выполняется параллельно и без конфликтов.

Исполнительными устройствами являются блоки внутренней памяти, (CRAM, память DSP0-DSP3) или любая внешняя память, доступная через MPORT. Задатчиками могут быть CPU, каналы DMA SWIC, SRIO, MFBSP, Ethernet, PMSC, VPIN, VPOUT, каналы DMA типа память-память.



## 1.4 Инструментальное программное обеспечение

Для данной микросхемы разработана интегрированная среда проектирования программного обеспечения MCStudio, которая обеспечивает полный цикл разработки и отладки программ. Эта среда функционирует на инструментальной машине IBM PC.

Интегрированная среда проектирования включает в себя:

- среду разработки программ для CPU- и DSP-ядер;
- среду отладки программ в исходных текстах, исполняемых на программном симуляторе, и отладчик для работы с платой отладочного модуля для данной микросхемы или целевым устройством. Целевое устройство подключается к персональному компьютеру через эмулятор JTAG для процессоров серии «Мультикор».
- средства программного моделирования;
- возможность доступа пользователю ко всем инструментам через один интерфейс.

## 1.5 Операционная система для микросхемы 1892ВМ7Я

Linux - свободно распространяемое ядро Unix-подобной операционной системы. Linux обладает всеми свойствами современной Unix-системы, включая полноценную многозадачность, развитую подсистему управления памятью и сетевую подсистему.

Ядро Linux, поставляемое вместе со свободно распространяемыми прикладными и системными программами образует полнофункциональную универсальную операционную систему. Большую часть базовых системных компонент Linux унаследовал от проекта GNU, целью которого является создание свободной микроядерной операционной системы с лицом Unix.

## 2. ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР

### 2.1 Основные характеристики CPU

- Архитектура – MIPS32;
- 32-х битные пути передачи адреса и данных;
- Кэш команд объемом 16 Кбайт;
- Кэш данных объемом 16 Кбайт;
- Архитектура привилегированных ресурсов в стиле ядра R4000:
  - Регистры Count/Compare для прерываний реального времени;
  - Отдельный вектор обработки исключений по прерываниям;
- Программируемое устройство управления памятью:
  - Два режима работы – с TLB и Fixed Mapped (FM);
  - 16 строк в режиме TLB;
  - В режиме FM адресные пространства отображаются с использованием битов регистров;
- Устройство умножения и деления;
- Сопроцессором арифметики в формате с плавающей точкой;
- Поддержка отладки JTAG.

### 2.2 Блок схема

Блок схема процессорного ядра RISCORE32 приведена на Рисунок 2.1.

Ядро содержит следующие узлы:

- Устройство исполнения (Execution Core);
- Устройство целочисленного умножения и деления (MDU);
- Системный управляющий сопроцессор (CP0);
- Сопроцессор арифметики в формате с плавающей точкой (FPU);
- Устройство управления памятью (MMU – Memory Management Unit);
- Контроллер кэш (Cache Controller);
- Устройство шинного интерфейса (BIU);
- Кэш команд (Instruction Cache);
- Кэш данных (Data Cache);
- Преобразователь виртуального адреса в физический адрес (TLB/FM);
- Средства отладки программ (OnCD – On Chip Debugger) с JTAG портом.

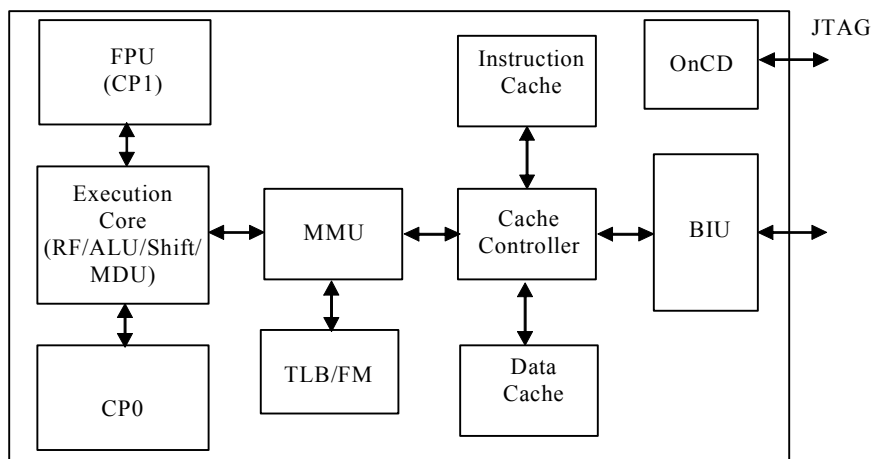


Рисунок 2.1. Блок схема процессорного ядра RISCore32

## 2.3 Составляющие логические блоки

В следующих подразделах описываются устройства, входящие в состав процессорного ядра.

### 2.3.1 Устройство исполнения

Входящее в ядро устройство исполнения реализует архитектуру load-store (загрузка-сохранение) с одноктактными операциями арифметического логического устройства (АЛУ) (логические операции, операции сдвига, сложение и вычитание). В ядре имеется тридцать два 32-х битных регистра общего назначения, используемых для скалярных целочисленных операций и вычисления адреса. В регистровом файле есть два порта чтения и один порт записи. Также используются обходные пути передачи данных для минимизации количества остановок конвейера.

В состав устройства исполнения входят:

- 32-х битный сумматор, используемый для вычисления адреса данных;
- Адресное устройство для вычисления адреса следующей команды;
- Логика определения перехода и вычисления адреса перехода;
- Блок выравнивания при загрузке данных;
- Мультиплексоры обходных путей передачи данных для исключения остановок конвейера в тех случаях, когда команды, производящие данные и команды, использующие эти данные, расположены в программе достаточно близко;
- Блок обнаружения Нуля/Единицы для реализации команд CLZ и CLO;
- АЛУ для выполнения побитных операций;
- Сдвигающее устройство и устройство выравнивания при сохранении данных.

### 2.3.2 Устройство умножения/деления (MDU)

Устройство умножения/деления выполняет соответствующие операции. MDU выполняет операции умножения за 17 тактов, операции умножения с накоплением за 18 тактов, операции деления за 33 такта и операции деления с накоплением за 34 такта. Попытка активизировать следующую команду умножения/деления до завершения выполнения предыдущей, так же как и использование результата этой операции до того, как она закончена, вызывает остановку конвейера. В MDU имеется вывод, определяющий формат операции – знаковый или беззнаковый.

### **2.3.3 Системный управляющий сопроцессор**

Сопроцессор отвечает за преобразование виртуального адреса в физический, протоколы кэш, систему управления исключениями, выбор режима функционирования (Kernel/User) и за разрешение/запрещение прерываний. Конфигурационная информация доступна посредством чтения регистров CP0 (см. раздел 2.7 “Регистры CP0”).

### **2.3.4 Сопроцессор арифметики в формате с плавающей точкой (FPU)**

Сопроцессор арифметики в формате с плавающей точкой выполняет операции в соответствии со стандартом ANSI/IEEE Standard 754-1985, “IEEE Standard for Binary Floating-Point Arithmetic.” Поддерживаются операции, как с одинарной, так и с двойной точностью. Сопроцессор выполняет дополнительные операции, не определенные стандартом. Сопроцессор содержит 16 64-разрядных регистра для хранения операндов с одинарной и двойной точностью. Сопроцессор также содержит регистры управления и состояния, которые обеспечивают обработку исключений в соответствии с требованиями стандарта.

### **2.3.5 Устройство управления памятью (MMU)**

Процессорное ядро содержит устройство управления памятью (MMU), реализующее интерфейс между исполнительным блоком и контроллером кэш. Ядро может работать как в режиме TLB – с 16-строчной, полностью ассоциативной матрицей TLB, так и в режиме FM (Fixed Mapped), когда используются простые преобразования виртуального адреса в физический адрес.

### **2.3.6 Контроллер кэш**

В данной версии процессора реализованы кэш команд и кэш данных, виртуально индексируемые и контролируемые по физическому тэгу типа direct mapped, что позволяет осуществлять доступ к кэш параллельно с преобразованием виртуального адреса в физический. Объем каждой кэш составляет 16 Кбайт.

### **2.3.7 Устройство шинного интерфейса (BIU – Bus Interface Unit)**

Устройство шинного интерфейса управляет внешними интерфейсными сигналами в соответствии со спецификацией шины АНВ (Advanced High-performance Bus) архитектуры АМВА (Advanced Microcontroller Bus Architecture).

### **2.3.8 OnCD контроллер**

В ядре имеется устройство для отладки программ OnCD с портом JTAG.

## 2.4 Конвейер

В RISC-ядре процессора реализован конвейер, состоящий из пяти стадий и аналогичный конвейеру ядра R3000. Конвейер дает возможность процессору работать на высокой частоте, при этом минимизируется сложность устройства, а также уменьшается стоимость и потребление энергии.

В этой главе содержатся следующие разделы:

- Раздел 2.1, “Стадии работы конвейера”
- Раздел 2.2, “Операции умножения и деления”
- Раздел 2.3, “Задержка выполнения команд перехода”
- Раздел 2.4, “Обходные пути передачи данных (Data bypass)”
- Раздел 2.5, “Задержка загрузки данных”
- Раздел 2.6, “Особые случаи при выполнении команд (Instruction Hazards)”

### 2.4.1 Стадии конвейера

Конвейер содержит пять стадий:

- Выборка команды (стадия I- Instruction)
- Дешифрация команды (стадия D - Data)
- Исполнение команды (стадия E - Execution)
- Выборка из памяти (стадия M - Memory)
- Обратная запись (стадия W – Write Back)

На Рисунок 2.2 показаны операции, выполняемые RISC-ядром на каждом этапе конвейера.

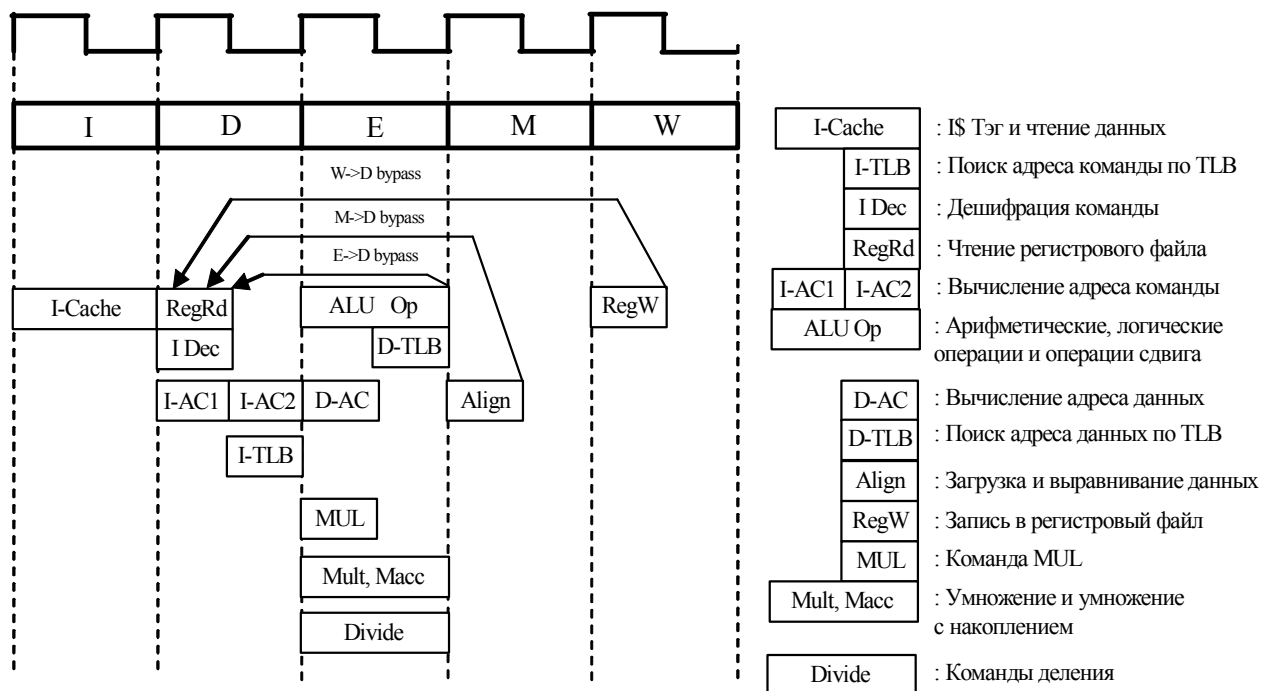


Рисунок 2.2

#### **2.4.1.1 Стадия I: выборка команды**

На этой стадии команда выбирается из командного кэш.

#### **2.4.1.2 Стадия D: дешифрация команды**

На этой стадии:

- Операнды выбираются из регистрового файла.
- Операнды передаются на эту стадию со стадий E, M и W.
- ALU определяет, выполняется ли условие перехода и вычисляет виртуальный адрес перехода для команд перехода.
- Осуществляется преобразование виртуального адреса в физический адрес.
- Производится поиск адреса команды по TLB и вырабатывается признак hit/miss.
- Командная логика выбирает адрес команды.

#### **2.4.1.3 Стадия E: исполнение**

На этой стадии:

- ALU выполняет арифметические или логические операции для команд типа регистр-регистр.
- Производится преобразование виртуального адреса в физический адрес для данных, используемых командами загрузки и сохранения.
- Производится поиск данных по TLB и вырабатывается признак hit/miss.
- Все операции умножения и деления выполняются на этой стадии.

#### **2.4.1.4 Стадия M: выборка из памяти**

На этой стадии осуществляется загрузка и выравнивание загруженных данных в границах слова.

#### **2.4.1.5 Стадия W: обратная запись**

На этой стадии для команд типа регистр-регистр или для команд загрузки результат записывается обратно в регистровый файл.

### **2.4.2 Операции умножения и деления**

Время выполнения этих операций соответствует 17 тактам для команд умножения и 18 тактам для команд умножения с накоплением, а также 33 тактам для команд деления и 34 тактам для команд деления с накоплением.

### **2.4.3 Задержка выполнения команд перехода (Jump, Branch)**

Конвейер осуществляет выполнение команд перехода с задержкой в один такт. Однотактная задержка является результатом функционирования логики, ответственной за принятие решения о переходе на стадии D конвейера. Эта задержка позволяет использовать адрес перехода, вычисленный на предыдущей стадии, для доступа к команде на следующей D-стадии. Слот задержки перехода (branch delay slot) позволяет отказаться от остановок конвейера при переходе. Вычисление адреса и проверка условия перехода выполняются одновременно на стадии D. Итоговое значение PC (счетчика команд) используется для выборки очередной команды на стадии I, которая является второй командой после перехода. На Рисунке 2.3 показан слот задержки перехода.



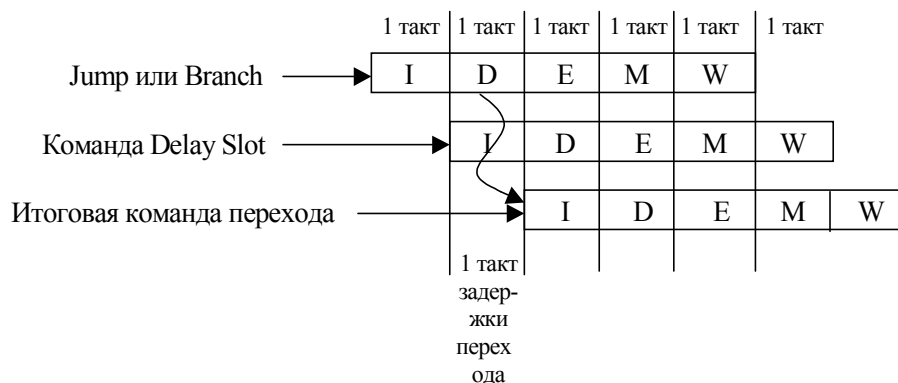


Рисунок 2.3. Слот задержки перехода

### 2.4.4 Обходные пути передачи данных (Data bypass)

Для большинства команд MIPS32 исходными операндами являются значения, хранящиеся в регистрах общего назначения. Эти операнды выбираются из регистрового файла в первой половине D-стадии. После исполнения на ALU результат, в принципе, готов для использования другими командами. Но запись результата в регистровый файл осуществляется только на стадии W. Это лишает следующую команду возможности использовать результат в течение 3-х циклов, если ее операндом является результат выполнения последней операции, сохраненный в регистровом файле. Для преодоления этой проблемы используются обходные пути передачи данных.

Мультиплексоры обходных путей передачи данных для обоих операндов располагаются между регистровым файлом и ALU (Рисунок 2.4). Они позволяют передавать данные с выхода стадий E, M и W конвейера прямо на стадию D, если один из регистров источника (source) декодируемой команды совпадает с регистром назначения (target) одной из предшествующих команд. Входы мультиплексоров подключены к обходным путям M→D и E→D, а также W→D.

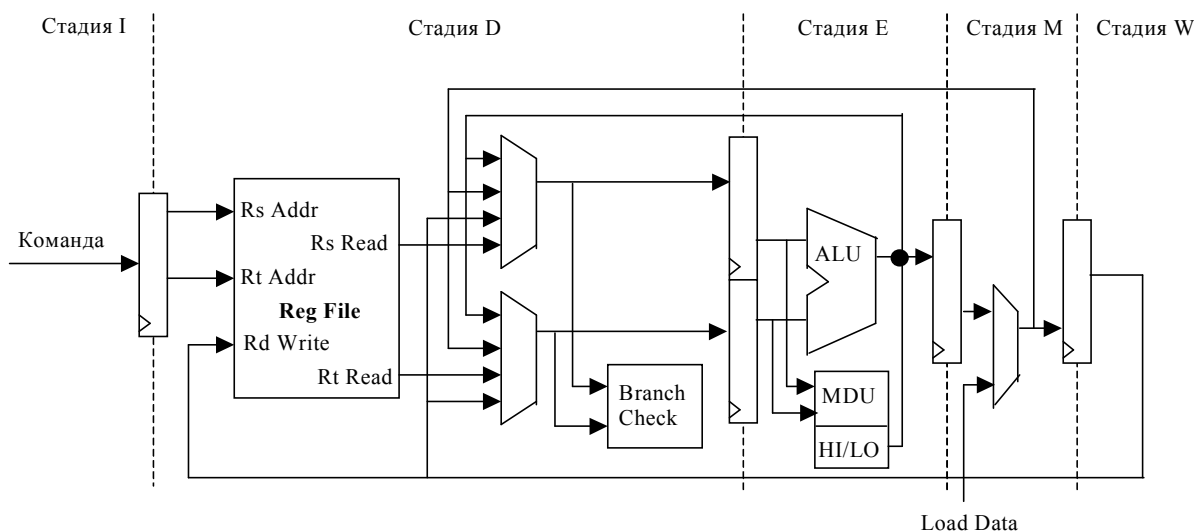


Рисунок 2.4

На Рисунок 2.5 показаны обходные пути передачи данных для команды Add1, за которой следует команда Sub2 и затем снова Add3. Поскольку команда Sub2 в качестве одного из операндов использует результат операции Add1, используется обходной путь E→D. Следующая команда Add3 использует результаты обеих предшествующих операций: Add1 и Sub2. Так как данные команды Add1 в это время находятся на стадии M,

используется обходной путь  $M \rightarrow D$ . Кроме того, вновь используется обходной путь  $E \rightarrow D$  для передачи результата операции  $Sub_2$  команде  $Add_3$ .

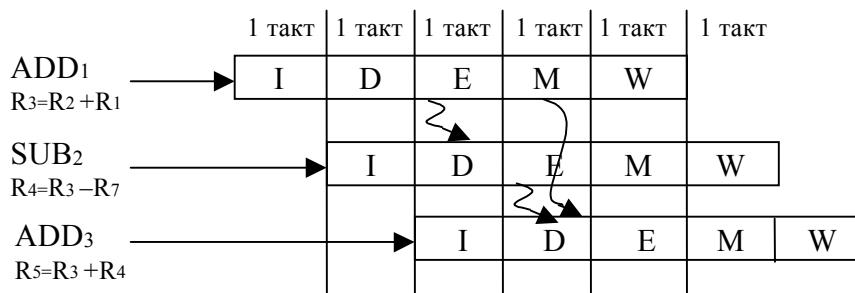


Рисунок 2.5

### 2.4.5 Задержка загрузки данных

Данные, выбираемые командами загрузки (Load), становятся доступными на конвейере только после выравнивания на стадии M. При этом данные, являющиеся исходными операндами, должны предоставляться командам для обработки уже на стадии D. Поэтому, если сразу за командой загрузки следует команда, для которой один из регистров исходных операндов совпадает с регистром, в который производится загрузка данных, это вызывает приостановку в работе конвейера на стадии D. Эта приостановка осуществляется аппаратной вставкой команды NOP. Во время этой задержки часть конвейера, которая находится дальше стадии D, продолжает продвигаться. Если же команда, использующая загружаемые данные, следует за командой загрузки не сразу, а через одну или через две, то для обеспечения бесперебойной работы конвейера используется один из обходных путей передачи данных:  $M \rightarrow D$  или  $W \rightarrow D$  (Рисунок 2.6).

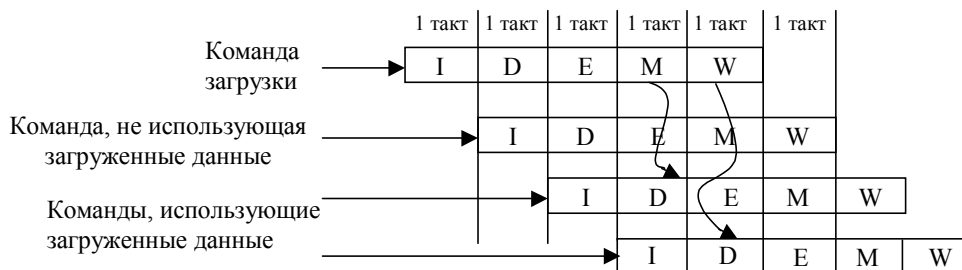


Рисунок 2.6

## 2.5 Сопроцессор арифметики в формате с плавающей точкой (FPU)

### 2.5.1 Введение

Сопроцессор арифметики в формате с плавающей точкой выполняет операции в соответствии со стандартом ANSI/IEEE Standard 754-1985, “IEEE Standard for Binary Floating-Point Arithmetic.” Поддерживаются операции, как с одинарной, так и с двойной точностью (single- or double-precision). Сопроцессор выполняет дополнительные операции, не определенные стандартом. Сопроцессор содержит 16 64-разрядных регистра для хранения операндов с одинарной и двойной точностью. Сопроцессор также содержит регистры управления и состояния, которые обеспечивают обработку исключений в соответствии с требованиями стандарта.

FPU реализован как сопроцессор CP1.

### 2.5.2 Регистры FPU

#### 2.5.2.1 Типы регистров

В FPU имеется три типа регистров:

- регистры общего назначения (FGR);
- регистры в формате с плавающей точкой (FPR);
- регистры управления (FCR).

32-разрядные регистры FGR являются прямо адресуемыми. FPU содержит 32 таких регистра.

64-разрядные регистры в формате с плавающей точкой FPR являются логическими и используются для хранения данных в процессе выполнения операций в формате с плавающей точкой. Эти регистры образованы конкатенацией двух соседних регистров FGR. В зависимости от операции, FPR содержит величину с одинарной или двойной точностью.

Регистры управления регистры FCR используются для выбора режима округления, обработки исключений и сохранения состояния.

В Таблица 2.1. приведены регистры управления FPU в порядке возрастания нумерации.

**Таблица 2.1. Управляющие регистры FPU**

Номер регистра	Название Регистра	Функция
0	FIR	Регистр версии и реализации (Implementation and Revision register)
25	FCCR	Регистр кодов условий (Condition Codes register)
26	FEXR	Регистр исключений (Exceptions register)
28	FENR	Регистр разрешения исключений (Enables register)
31	FCSR	Регистр управления и состояния (Control/Status register)

В командах STC1 и CFC1 регистры FCCR, FEXR и FENR получают доступ к соответствующим частям регистра FCSR, т.е. эти регистры являются отражением соответствующих частей регистра FCSR.

Доступ к регистрам управления FPU не является привилегированным. Любая программа, которая выполняет инструкции с плавающей точкой, имеет доступ к регистрам управления FPU. Доступ к ним осуществляется посредством STC1 и CFC1 команд.

### 2.5.2.2 Регистры общего назначения и регистры в формате с плавающей точкой

32 регистра общего назначения (FGR) являются 32-разрядными и могут непосредственно адресоваться. Они используются в операциях в формате с плавающей точкой и индивидуально доступны по командам move, load и store. Перечень регистров FGR приведен в Таблица 2.2.

Таблица 2.2. Регистры FGR и FPR

Номер регистра FGR	Название регистра FGR	Название регистра FPR
0	FGR0	FPR0 (least)
1	FGR1	FPR0 (most)
2	FGR2	FPR2 (least)
3	FGR3	FPR2 (most)
.	.	.
.	.	.
28	FGR28	FPR28 (least)
29	FGR29	FPR28 (most)
30	FGR30	FPR30 (least)
31	FGR31	FPR30 (most)

Регистры в формате с плавающей точкой (FPR) формируются из регистров FGR, посредством их конкатенации. Для адресации этих регистров используется только четный номер. Нечетный номер является недопустимым. В процессе операций с одинарной точностью используется только младшая часть (least) регистра FPR используется.

### 2.5.2.3 Форматы величин, хранящихся в регистрах FPR

В отличие от процессора целочисленной арифметики, FPU не интерпретирует двоичную кодировку входных операндов и не производит двоичное кодирование результатов каждой операции. Значение, хранящееся в регистре FPR, имеет определенный формат или тип. Этот формат могут использовать только те команды, которые оперируют с ним (этим форматом). Формат может быть неизвестным (не интерпретируемым) либо одним из существующих числовых форматов: формат с плавающей точкой одинарной или двойной точностью, слово или двойное слово с фиксированной точкой.

Числовая величина в регистре FPR всегда установлена, когда она записана в этот регистр:

- при загрузке регистра FPR по команде load в регистр записываются двоичные данные, формат которых не интерпретируется.
- команды вычисления в формате с плавающей точкой или команды move, формируют в регистре FPR результат формата fmt.

Когда регистр FPR с не интерпретируемым значением используется как входной операнд для команды, которая требует значение в формате fmt и рассматривает двоичное содержимое как значение в формате fmt, значение в регистре FPR изменяется к значению в формате fmt. То есть, двоичное содержимое этого регистра не может рассматриваться в другом формате.

Если регистр FPR содержит значение в формате `fmt`, то вычислительные команды не должны использовать этот регистр как входной операнд другого формата. Если такое происходит, то значение в регистре становится неизвестным и результат команды также является неизвестным значением. Использование FPR регистра с неизвестным значением в качестве входного операнда команды приводит к результату, значение которого также неизвестно.

Формат величины, находящейся в регистре FPR, не изменяется, когда происходит чтение этого регистра командой `store`. Команда `store` выводит двоичную кодировку в соответствии со значением, содержащимся в регистре FPR. Если значение в регистре FPR неизвестно, то закодированное двоичное значение, выведенное операцией, неопределенно.

#### 2.5.2.4 Управляющие регистры

##### 2.5.2.4.1 Регистр реализации (FIR, CP1 Control Register 0)

Регистр реализации (Floating Point Implementation Register - FIR) - это 32-битный регистр доступный только на чтение. Он содержит информацию, которая определяет возможности FPU, идентификацию FPU и номер версии FPU. На Рисунок 2.7 показан формат регистра FIR, а в Таблица 2.3 описаны поля этого регистра.

31	18	17	16	15	8	7	0
0		D	S	Processor ID		Revision	

Рисунок 2.7. Формат FIR регистра

Таблица 2.3. Описание полей регистра FIR

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
-	31:18	Не используется	0	0
D	17	Указывает, реализованы ли тип данных двойной точности (D) и соответствующие инструкции: 0 - не реализованы 1 – реализованы	R	1
S	16	Указывает, реализованы ли тип данных одинарной точности (S) и соответствующие инструкции: 0 – не реализованы 1 - реализованы	R	1
Processor ID	15:8	Идентификация типа процессора вычислений с плавающей точкой (FPU)	R	0000 0000
Revision	7:0	Номер версии FPU. Это поле позволяет программам различать разные версии одного типа FPU.	R	0000 0000

##### 2.5.2.4.2 Регистр управления и состояния (FCSR, CP1 Control Register 31)

Регистр управления и состояния (Floating Point Control and Status Register - FCSR) – это 32-битный регистр, который управляет работой FPU и содержит информацию о состоянии FPU:

- выбор режима округления для арифметических операций;
- выборочное разрешение исключений при возникновении соответствующих условий исключений;

- управление некоторыми опциями обработки денормализованных чисел;
- сообщает о любых IEEE исключениях произошедших во время последней выполненной команды;
- сообщает о IEEE исключениях произошедших в совокупности выполненных команд;
- показывает код условия, который является результатом команд сравнения.

Доступ к регистру *FCSR* не является привилегированным. Любая программа, которая имеет доступ к FPU (если он разрешён в регистре *Status* ), может читать из или записывать в регистр *FCSR*. На Рисунок 2.8 представлен формат *FCSR* регистра, в Таблица 2.8 описаны поля этого регистра.

31	25	24	23	22-18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
FCC			F S	FC C	0	Cause					Enables					Flags					RM					
7	6	5	4	3	2	1	0				E	V	Z	O	U	I	V	Z	O	U	I	V	Z	O	U	I

Рисунок 2.8. Формат регистра *FCSR*

Таблица 2.4. Описание полей регистра *FCSR*

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
FCC	31:25, 23	Коды условий. Эти биты содержат результат выполнения FPU команд сравнения и используются в командах условных переходов и в командах условных перемещений данных. Какой FCC бит используется точно определено в команде перехода или перемещения.	R/W	Не определено
FS	24	Сброс в ноль. Когда FS=1, денормализованный результат операции сбрасывается в ноль вместо появления исключения “Нереализованная операция” (Unimplemented Operation).	R/W	Не определено
-	22:18	Не используются	0	0
Cause	17:12	Биты причины. Эти биты показывают условия исключений, которые возникают во время выполнения арифметических команд. Бит устанавливается в 1, если соответствующая исключительная ситуация появилась во время выполнения команды и устанавливается в 0 в противоположном случае. По значениям этих бит можно определить какая исключительная ситуация вызвана выполнением предыдущей арифметической команды. Значение каждого бита данного поля представлено в Таблица 2.5.	R/W	Не определено
Enables	11:7	Биты разрешения соответствующего исключения при возникновении любой из пяти IEEE исключительных ситуаций. Исключение происходит в случае, когда соответствующие бит Cause и бит Enables одновременно установлены либо во время выполнения арифметической операции, либо при перемещении нового значения в регистр <i>FCSR</i> или <i>FEXR</i> и <i>FENR</i> по команде <i>move</i> . Заметьте, что бит E в поле Cause не имеет соответствующего бита в поле Enables, так как исключение “Нереализованная Операция” всегда разрешено. Значение каждого бита данного поля представлено в Таблица 2.5.	R/W	Не определено



Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
Flags	6:2	<p>Флаговые биты. Это поле показывает любые исключительные ситуации, вызванные завершившимися командами со времени последнего программного сброса данного поля.</p> <p>Когда при арифметической операции возникает исключительная ситуация, которая не приводит к FPU исключению (соответствующий бит в Enables сброшен), то соответствующий бит (биты) устанавливается в поле Flags. В других ситуациях поле Flags остаётся без изменений. Арифметические операции, которые приводят к возникновению FPU исключения (бит в Enables установлен), не изменяют состояния бит в поле Flags.</p> <p>У этого поля нет аппаратного сброса, оно должно явно сбрасываться программой.</p> <p>Значение каждого бита данного поля представлено в Таблица 2.5.</p>	R/W	Не определено
RM	1:0	<p>Режим округления. Обозначает режим округления, который используется большинством операций в формате с плавающей точкой (некоторые операции используют специфический режим округления). Возможные кодировки этого поля представлены в Таблица 2.6.</p>	R/W	Не определено

Поля FCC, FS, Cause, Enables, Flags и RM в регистрах FCSR, FCCR, FEXR и FENR всегда обозначают правильные состояния. Это означает что, если новое значение поля записывается в FCSR регистр, то это новое значение можно прочитать в соответствующем альтернативном регистре FCCR, FEXR или FENR. И наоборот, записав новое значение поля в альтернативный регистр, его можно прочитать в FCSR регистре.

**Таблица 2.5. Описание бит в полях Cause, Enables и Flags**

Имя бита	Значение бита
E	Нереализованная операция (Unimplemented Operation) Этот бит существует только в поле Cause
V	Недействительная операция (Invalid Operation)
Z	Деление на ноль (Divide by Zero)
O	Переполнение (Overflow)
U	Потеря значимости (Underflow)
I	Неточность (Inexact)

**Таблица 2.6. Описание режимов округления**

Кодировка поля RM	Описание
0	RN – округление к ближайшему (round to nearest) Округление результата к ближайшему представимому значению. Когда два представимых значения одинаково близки, результат округляется к значению, чей наименее значащий бит равен 0 (чётный)
1	RTZ – округление к нулю (round towards zero) Округление результата к ближайшему значению, величина (модуль) которого не больше величины результата
2	RP – округление к плюс бесконечности (round towards plus infinity) Округление результата к ближайшему значению не меньшему чем сам результат
3	RM – округление к минус бесконечности (round towards minus infinity) Округление результата к ближайшему значению не большему чем сам результат.

#### 2.5.2.4.3 Регистр кодов условий (FCCR, CP1 Control Register 25)

Регистр кодов условий (Floating Point Condition Codes Register - FCCR) является альтернативным регистром для чтения и записи поля кодов условий FCC, которое также хранятся в регистре FCSR. В отличие от FCSR регистра, в регистре FCCR восемь бит поля FCC являются смежными. На Рисунок 2.9 представлен формат *FCSR* регистра, в Таблица 2.7 описаны поля этого регистра.

31	8	7	0						
0000 0000 0000 0000 0000 0000		FCC							
		7	6	5	4	3	2	1	0

**Рисунок 2.9. Формат регистра FCCR**
**Таблица 2.7. Описание полей регистра FCCR**

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
-	31:8	Не используются	0	0
FCC	7:0	Коды условий. Эти биты содержат результат выполнения FPU команд сравнения и используются в командах условных переходов и в командах условных перемещений данных. Какой FCC бит используется точно определено в команде перехода или перемещения. См. описание поля FCC в регистре <i>FCSR</i> в Таблица 2.4.	R/W	Не определено

#### 2.5.2.4.4 Регистр исключений (FEXR, CP1 Control Register 26)

Регистр исключений (Floating Point Exceptions Register - FEXR регистр) является альтернативным регистром для чтения и записи полей Cause и Flags, которые также хранятся в регистре FCSR. На Рисунок 2.10 представлен формат *FEXR* регистра, в Таблица 2.8 описаны поля этого регистра.

31	18	17	16	15	14	13	12	11	7	6	5	4	3	2	1	0	
0		Cause						0			Flags					0	
		E	V	Z	O	U	I				V	Z	O	U	I		

Рисунок 2.10. Формат регистра FEXR

Таблица 2.8. Описание полей регистра FEXR

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
-	31:18, 11:7, 1:0	Не используются	0	0
Cause	17:12	Биты причины. Эти биты показывают исключительные ситуации, которые возникают во время выполнения FPU арифметических команд. См. описание поля Cause в регистре FCSR в Таблица 2.4.	R/W	Не определено
Flags	6:2	Флаговые биты. Это поле показывает любые исключительные ситуации вызванные завершившимися командами со времени последнего программного сброса данного поля. См. описание поля Flags в регистре FCSR в .	R/W	Не определено

#### 2.5.2.4.5 Регистр разрешения исключений (FENR, CP1 Control Register 28)

Регистр разрешения исключений (Floating Point Enable Register - *FENR регистр*) является альтернативным регистром для чтения и записи полей Enables, FS и RM, которые также хранятся в регистре FCSR. На Рисунок 2.11 представлен формат FENR регистра, в Таблица 2.9 описаны поля этого регистра.

31	12	11	10	9	8	7	6	3	2	1	0	
0000 0000 0000 0000 0000						Enables			0000	FS	RM	
						V	Z	O	U	I		

Рисунок 2.11. Формат регистра FENR

Таблица 2.9. Описание полей регистра FENR

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
0	31:12, 6:3	Не используется	0	0
Enables	11:7	Биты разрешения соответствующего исключения при возникновении любой из пяти IEEE исключительных ситуаций. См. описание поля Enables в регистре FCSR в Таблица 2.4.	R/W	Не определено
FS	2	Сброс в ноль. Когда FS=1, денормализованный результат операции сбрасывается в ноль вместо появления исключения “Нереализованная операция” (Unimplemented Operation). См. описание поля FS в регистре FCSR в .	R/W	Не определено
RM	1:0	Режим округления. Обозначает режим округления, который используется большинством операций с плавающей точкой. См. описание поля RM в регистре FCSR в .	R/W	Не определено

### 2.5.3 Исключения FPU

#### 2.5.3.1 Формирование исключения

При возникновении исключения команда, вызвавшая его, а также все последующие команды не выполняются и не изменяют содержимого регистров FGR. При необходимости, после обработки исключения выполнение прерванного потока команд может быть возобновлено.

В поле *Cause* содержатся признаки исключений. Оно обновляется при выполнении каждой арифметической операции в формате с плавающей точкой. Признак устанавливается в 1, если возникает соответствующее условие исключения, иначе он устанавливается в 0.

Исключение возникает каждый раз, если одновременно признак поля *Cause* и соответствующий ему бит *Enable* установлены в 1. Это происходит или во время выполнения операции в формате с плавающей точкой или, при передаче данных в регистр FCSR по команде *move*. Бит *Enable* для Unimplemented Operation не существует, то есть исключение по этому условию возникает всегда.

Содержимое поля *Cause* используется в обработчике исключения. Перед выходом из обработчика исключения по операции в формате с плавающей точкой, или перед установкой бит поля *Cause* по команде *move*, необходимо сначала обнулить соответствующие биты *Enable*, для того, чтобы предотвратить повторное возникновение исключения.

Пользовательским программам не доступны биты поля *Cause*. Если эта информация необходима этим программам, то она должна быть доступна им другими путями, а не через регистр *Status*.

Если операция в формате с плавающей точкой устанавливает только неразрешенные биты поля *Cause*, то исключения не происходит, и записывается результат, определяемый стандартом IEEE (см. Таблица 2.10). Когда операция в формате с плавающей точкой не вызывает исключения, программа может контролировать условия исключения, считывая содержимое поля *Cause*.

Поле *Flag* – совокупная накопленная информация по условиям исключений. Команды, которые вызывают исключения, не обновляют биты поля *Flag*. Биты поля *Flag* устанавливаются в 1, если соответствующее условие исключения возникает, иначе биты остаются без изменения. Бит для условия исключения типа Unimplemented Operation в этом поле не предусмотрено. В результате выполнения операции в формате с плавающей точкой биты поля *Flag* никогда не сбрасываются, но могут быть установлены или сброшены (обнулены) при записи данных в регистр FCSR по команде *move*.

#### 2.5.3.2 Условие исключений

В этом пункте описаны следующие пять условий исключения, определенных стандартом ANSI/IEEE Standard 754-1985:

- исключение по недопустимой операции (Invalid Operation Exception);
- исключение при делении на ноль (Division By Zero Exception);
- исключение по ложному переполнению (Underflow Exception);
- исключение по переполнению (Overflow Exception);
- неточное исключение (Inexact Exception).

Этот пункт также содержит описание исключения по нереализованной операции (unimplemented operation). Оно используется для сообщения о необходимости программной эмуляции команды. Обычно арифметическая операция IEEE может вызывать только одно условие исключения. Единственный случай, когда два исключения могут происходить в то же самое время, это Inexact With Overflow и Inexact With Underflow.

Под управлением программы, условие исключения IEEE может вызывать прерывание (trap) процессора или не вызывать его. Стандарт IEEE определяет результат операции при возникновении условия исключения для случая, когда прерывание процессора по этому исключению не разрешено. Для этого случая результаты операций приведены в Таблица 2.10. При переполнении результат операции зависит от режима округления.

**Таблица 2.10. Результаты операций при исключениях**

Бит	Описание	Результат операции
V	Invalid Operation	Quiet NaN
Z	Divide by Zero	Properly signed infinity
U	Underflow	Округленный результат (Rounded result)
I	Inexact	Округленный результат. Если это исключение вызвано переполнением (Overflow) при неразрешенном прерывании, то формируется результат с переполнением.
O	Overflow	Зависит от режима округления: 0 (RN) – infinity со знаком промежуточного результата; 1 (RZ) – format’s infinity со знаком промежуточного результата; 2 (RP) – при положительном переполнении – positive infinity. При отрицательном переполнении - format’s most negative infinity; 3 (RM) - при положительном переполнении – format’s largest finite number. При отрицательном переполнении – minus infinity.

### 2.5.3.3 Исключение по недопустимой операции

Это исключение возникает, если один или оба операнда недопустим для выполняемой операции.

Недопустимые операции:

- Один или оба операнда являются NaN (за исключением не арифметических команд MOV.fmt, MOVT.fmt, MOVF.fmt, MOVN.fmt, и MOVZ.fmt);
- Сложение или вычитание: вычитание бесконечных величин, таких как  $(+\infty) + (-\infty)$  или  $(-\infty) - (-\infty)$ ;
- Умножение:  $0 * \infty$ , с любыми знаками;
- Деление:  $0/0$  или  $\infty / \infty$ , с любыми знаками;
- Квадратный корень: операнд меньше чем 0 (-0 является допустимым значением);
- Преобразование числа в формате с плавающей запятой к формату с фиксированной запятой, если возникает переполнение, или значение операнда равно infinity или NaN препятствуют точному представлению данных в необходимом формате;
- Некоторые операции сравнения, в которых один или оба операнда имеют значение QNaN.

#### 2.5.3.4 *Исключение при делении на ноль*

Это исключение возникает, если делитель равен нулю, а делимое является конечным числом, отличным от нуля. Результат, когда не возникает прерывания, равен бесконечности. Деление  $(0/0)$  и  $(\infty/0)$  не приводят к исключению. При делении  $(0/0)$  возникает исключение по недопустимой операции. Результат  $(\infty/0)$  – бесконечность со знаком.

#### 2.5.3.5 *Исключение по ложному переполнению(потеря значимости)*

Два связанных события могут повлиять на возникновение ложного переполнения:

- близость результата к нулю (tininess): создание бесконечно малого результата отличного от нуля находящегося в промежутке между  $\pm 2^{E_{\min}}$ , который из-за своей малой величины может вызывать впоследствии какое либо другое исключение, например как переполнение при делении;
- потеря точности: экстраординарная потеря точности во время аппроксимации таких малых чисел ненормированными числами.

Стандарт IEEE определяет, что «близость результата к нулю» может быть обнаружена в любой из следующих моментов времени:

- после округления, когда не нулевой результат получен из предположения неограниченности диапазона экспоненты и находится строго между  $\pm 2^{E_{\min}}$ ;
- пред округлением, когда не нулевой результат получен из предположения неограниченности, как диапазона экспоненты, так и точности, и находится строго между  $\pm 2^{E_{\min}}$ ;

В FPU близость результата к нулю обнаруживается после округления.

Стандарт IEEE определяет, что потеря точности может быть получена в результате любого из следующих условий:

- нарушение нормализации (denormalization), когда полученный результат отличается от вычисленного без ограничений диапазона экспоненты;
- неточный результат (inexact result), когда полученный результат отличается от вычисленного без ограничений диапазона экспоненты и точности.

В FPU потеря точности формируется, если получен неточный результат.

Если прерывание процессора при ложном переполнении не разрешено, признак U вырабатывается, когда обнаруживается одновременно и близость к нулю и потеря точности. При этом, результат может быть нулевым, ненормализованным или  $2^{E_{\min}}$ .

Если прерывание процессора при ложном переполнении разрешено, признак U вырабатывается, когда обнаруживается только близость к нулю, в не зависимости от потери точности.

#### 2.5.3.6 *Исключение при переполнении*

Это исключение возникает, когда величина округленного результата в формате с плавающей запятой (где диапазон экспоненты не ограничен) больше, чем наибольшее конечное число результирующего формата (destination format's largest finite number).

Если прерывание процессора при переполнении не разрешено, результат определяется режимом округления и знаком промежуточного результата.

### 2.5.3.7 Неточное исключение

Неточное исключение возникает, если:

- округленный результат операции не является точным;
- округленный результат операции вызывает переполнение, а прерывание по переполнению не разрешено.

### 2.5.3.8 Исключение по нереализованной операции

Это исключения не регламентировано стандартом IEEE. Операции, которые не полностью поддерживаются аппаратурой, вызывают исключение, для того, чтобы программное обеспечение могло выполнить соответствующую операцию.

Для этого условия исключения не предусмотрено разрешающего бита, то есть прерывание процессора возникает всегда. После того, как соответствующее эмулирование будет выполнено, прерванная программа возобновляется.

## 2.5.4 Время выполнения команд FPU

Время выполнения команд в формате с плавающей точкой приведено в Таблица 2.11.

**Таблица 2.11. Время выполнения команд FPU**

Команда	Время выполнения, такты
BC1F, BC1T, FLOOR, ROUND, TRUNC	1
CFC1, CTC1, MFC1, MOVF	1
CVT.S, CVT.D, CEIL	2
ABS, ADD, SUB, MULL, NEG	3
SQRT.S/SQRT.D	6/15
DIV.S/DIV.D	11/16



## 2.6 Устройство управления памятью (MMU)

### 2.6.1 Введение

Процессорное ядро содержит устройство управления памятью (MMU), реализующее интерфейс между устройством исполнения и контроллером кэш. MMU преобразует виртуальный адрес в физический прежде, чем посылает запрос контроллеру кэш для сравнения тэга или блоку шинного интерфейса для доступа к внешнему запоминающему устройству. Это преобразование является очень полезным свойством функционирования операционных систем при управлении физической памятью таким образом, чтобы в ней размещались несколько процессов, активных в одной и той же области памяти, и может быть даже на одном виртуальном адресе, но обязательно в различных областях физической памяти. Другие свойства MMU - защита зон памяти и определение протокола кэш.

MMU может выполнять преобразование адресов в двух режимах: в режиме TLB и в режиме FM. Режим преобразования определяется битом FM регистра CSR.

В режиме TLB используется полностью ассоциативная таблица преобразования адресов (TLB), имеющая 16 парных строк (entries). Во время преобразования осуществляется поиск соответствия по TLB. Если искомая строка отсутствует, генерируется прерывание.

В режиме FM (Fixed Mapped) работа MMU основана на простом алгоритме, обеспечивающем преобразование виртуального адреса в физический посредством механизма фиксированного отображения. Правила преобразования отличаются для различных областей виртуального адресного пространства (useg/kuseg, kseg0, kseg1, kseg2, kseg3).

На Рисунок 2.12 показано, взаимодействие MMU с процедурой доступа к кэш в режиме TLB, а на Рисунок 2.13 – в режиме FM.

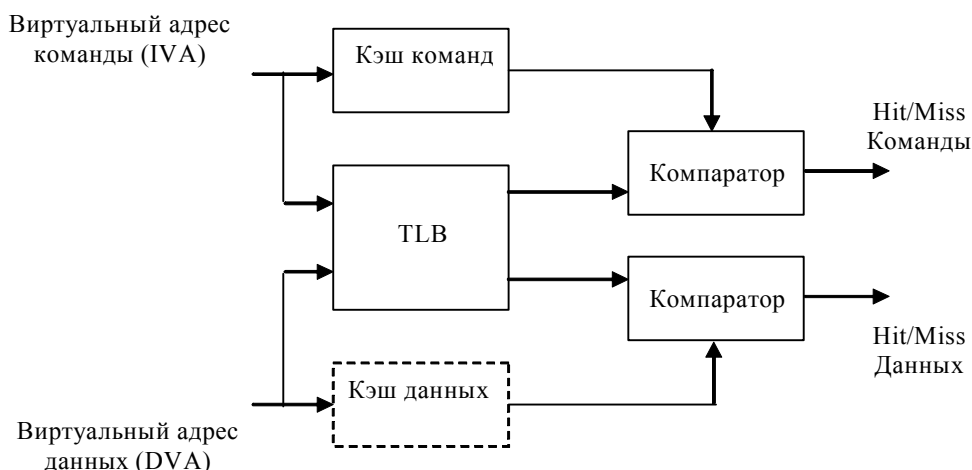


Рисунок 2.12

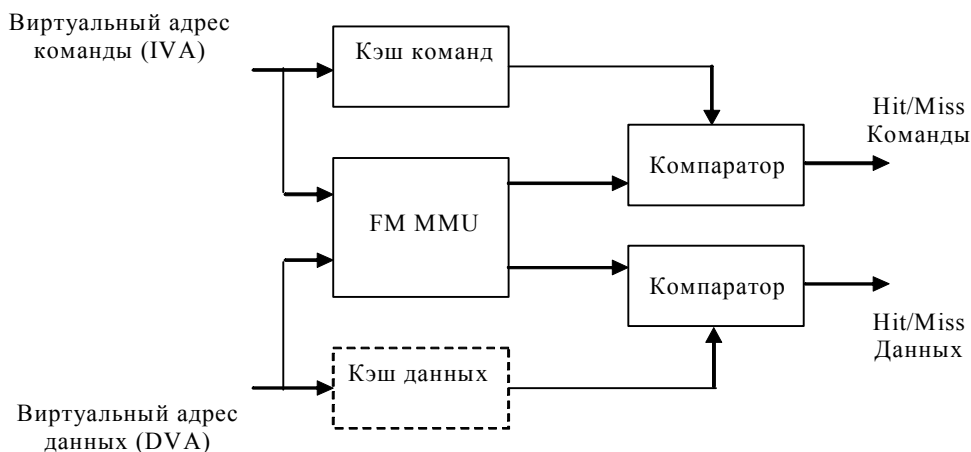


Рисунок 2.13

## 2.6.2 Режимы работы.

Процессорное ядро поддерживает два режима работы:

- Режим User (непривилегированный режим)
- Режим Kernel (привилегированный режим)

Режим User в основном используется для прикладных программ. Режим Kernel обычно используется для обработки исключительных ситуаций и привилегированных функций операционной системы, включая управление сопроцессором CP0 и доступ к устройствам ввода-вывода.

Преобразования, выполняемые MMU, зависят от режима работы процессора.

### 2.6.2.1 Виртуальные сегменты памяти

Виртуальные сегменты памяти, на которые делится адресное пространство, различаются в зависимости от режима работы процессора. На Рисунок 2.14 показана сегментация для 4 Гбайт ( $2^{32}$  байт) виртуального адресного пространства, адресуемого 32-разрядным виртуальным адресом для обоих режимов работы.

Ядро входит в режим Kernel после аппаратного сброса или когда происходит исключение. В режиме Kernel программное обеспечение имеет доступ к полному адресному пространству и ко всем регистрам CP0. В режиме User доступ ограничен подмножеством виртуального адресного пространства (0x0000\_0000 - 0x7FFF\_FFFF) и запрещен доступ к функциям CP0. В режиме User недоступны виртуальные адреса 0x8000\_0000 - 0xFFFF\_FFFF и обращение к ним вызывает исключение.

0xFFFF_FFFF			kseg3
0xE000_0000			
0xDFFF_FFFF			kseg2
0xC000_0000			
0xBFFF_FFFF			kseg1
0xA000_0000			
0x9FFF_FFFF			kseg0
0x8000_0000			
0x7FFF_FFFF	useg		kuseg
0x0000_0000			

Рисунок 2.14. Карта виртуальной памяти для режимов User и Kernel

Каждый из сегментов, показанных на Рисунок 2.14, является либо отображаемым (mapped), либо неотображаемым (unmapped). Различие объясняется в следующих двух разделах.

#### 2.6.2.1.1 Неотображаемые сегменты

В неотображаемом сегменте механизмы TLB или FM для преобразования виртуального адреса в физический адрес не используются. Особенно важно иметь неотображаемые сегменты памяти после аппаратного сброса, потому что TLB еще не запрограммировано и не может осуществлять преобразования.

Для неотображаемых сегментов преобразование виртуального адреса в физический является фиксированным.

Все неотображаемые сегменты, за исключением kseg0, никогда не кэшируемы. Кэшируемость kseg0 определяется полем K0 регистра Config CP0.

#### 2.6.2.1.2 Отображаемые сегменты

В отображаемом сегменте для преобразования виртуального адреса в физический адрес используются TLB или FM.

В режиме TLB преобразование отображаемых сегментов имеет постраничную основу. При преобразовании выявляется информация о кэшируемости страницы, а также атрибуты защиты, относящиеся к странице.

Для режима FM отображаемые сегменты имеют закрепленное преобразование виртуального адреса в физический. Кэшируемость сегмента определяется значениями полей K23 и KU регистра Config CP0. При FM-преобразовании невозможна защита сегментов от записи.

### 2.6.2.2 Режим User

В режиме User доступно однородное виртуальное адресное пространство размером 2 Гбайт ( $2^{31}$  байт), называемое сегментом пользователя.

На Рисунок 2.15 показано размещение виртуального адресного пространства режима User.

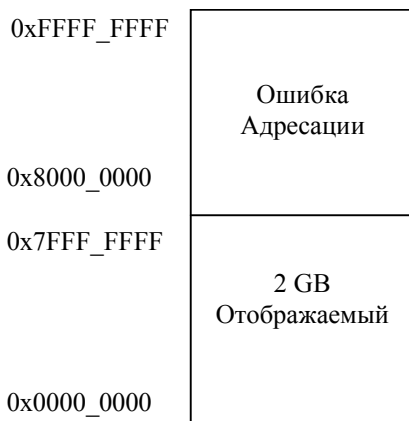


Рисунок 2.15

Сегмент потребителя начинается с адреса 0x0000\_0000 и заканчивается адресом 0x7FFF\_FFFF. Обращения по всем остальным адресам вызывают прерывания по ошибке адресации.

Процессор находится в режиме User, если в регистре Status CP0 установлены следующие значения разрядов:

- UM = 1
- EXL = 0
- ERL = 0

В Таблица 2.12 приводятся характеристики сегмента useg режима User.

Таблица 2.12

Адрес	Регистр Состояния			Имя сегмента	Диапазон адресов	Размер сегмента
	EXL	ERL	UM			
A(31)=0	0	0	1	useg	0x0000_0000 → 0x7FFF_FFFF	2GB ( $2^{31}$ байт)

Для всех допустимых виртуальных адресов режима User старший значащий бит адреса равен нулю, поскольку в режиме User допустимо обращение только к нижней половине карты виртуальной памяти. Любая попытка обращения по адресу со старшим битом, равным 1, в режиме User вызывает прерывание по ошибке адресации.

В режиме TLB виртуальный адрес перед преобразованием расширяется содержимым 8-разрядного поля ASID, образуя уникальный виртуальный адрес. Кэшируемость ссылки для страницы в этом режиме определяется установкой определенных бит строки TLB.

В режиме FM, область виртуальных адресов 0x0000\_0000-0x7FFF\_FFFF преобразуется в область физических адресов 0x4000\_0000-0xBFFF\_FFFF. Кэшируемость задается полем KU регистра Config CP0.

### 2.6.2.3 Режим Kernel

Процессор находится в режиме Kernel, когда регистр Status CP0 содержит хотя бы одно из следующих значений:

- UM = 0
- ERL = 1
- EXL = 1

Когда обнаруживается исключение, биты EXL или ERL устанавливаются, и процессор входит в режим Kernel. При завершении процедуры обработки исключения обычно выполняется команда возвращения из исключения (ERET). Команда ERET осуществляет переход по PC исключения, очищает ERL и EXL (если ERL=0). В результате возможен возврат процессора в режим User.

Виртуальное адресное пространство режима Kernel разделено на области в соответствии со значением старших битов виртуального адреса, как показано на Рисунок 2.16. Кроме того, в Таблица 2.13 содержатся характеристики сегментов режима Kernel.

0xFFFF_FFFF	Kernel virtual address space Mapped, 512 MB	kseg3
0xE000_0000		
0xDFFF_FFFF	Kernel virtual address space Mapped, 512 MB	kseg2
0xC000_0000		
0xBFFF_FFFF	Kernel virtual address space Unmapped, Uncached, 512 MB	kseg1
0xA000_0000		
0x9FFF_FFFF	Kernel virtual address space Unmapped, 512 MB	kseg0
0x8000_0000		
0x7FFF_FFFF		
	Mapped, 2048 MB	kuseg
0x0000_0000		

Рисунок 2.16

Таблица 2.13

Адрес	Регистр Состояния			Имя сегмента	Диапазон адресов	Размер сегмента
	EXL	ERL	UM			
A(31)=0	UM = 0			kuseg	0x0000_0000 → 0x7FFF_FFFF	2 GB (2 <sup>31</sup> )
A(31:29)=100 <sub>2</sub>	или			kseg0	0x8000_0000 → 0x9FFF_FFFF	512 MB (2 <sup>29</sup> )
A(31:29)=101 <sub>2</sub>	EXL=1			kseg1	0xA000_0000 → 0xBFFF_FFFF	512 MB (2 <sup>29</sup> )
A(31:29)=110 <sub>2</sub>	или			kseg2	0xC000_0000 → 0xDFFF_FFFF	512 MB (2 <sup>29</sup> )
A(31:29)=111 <sub>2</sub>	ERL=1			kseg3	0xE000_0000 → 0xFFFF_FFFF	512 MB (2 <sup>29</sup> )

#### 2.6.2.3.1 Режим Kernel, Пространство пользователя (kuseg)

Если старший значащий бит виртуального адреса A[31]=0, то выбирается виртуальное адресное пространство kuseg объемом 2 Гбайт, отображенное на адреса 0x0000\_0000 - 0x7FFF\_FFFF.

При ERL=0 в режиме TLB виртуальный адрес расширяется 8-битным значением поля ASID для образования уникального виртуального адреса. Кэшируемость определяется полем C строки TLB.

При ERL=0 в режиме FM, область виртуальных адресов 0x0000\_0000-0x7FFF\_FFFF преобразуется в область физических адресов 0x4000\_0000-0xBFFF\_FFFF. Кэшируемость задается полем KU регистра Config CP0.

При ERL = 1 в режимах TLB и FM, область адресов пользователя становится неотображаемым и некэшируемым адресным пространством. Виртуальный адрес kuseg соответствует тому же физическому адресу и не включает поле ASID. То есть, область виртуальных адресов kuseg соответствует области физических адресов 0x0000\_0000-0x7FFF\_FFFF.

#### 2.6.2.3.2 Режим Kernel, пространство 0 режима Kernel (kseg0).

Если в режиме Kernel три старших бита виртуального адреса равны 100<sub>2</sub>, выбирается виртуальное адресное пространство kseg0. Это область размером 2<sup>29</sup> байт (512 MB), которая расположена внутри границ, определяемых адресами 0x8000\_0000 и 0x9FFF\_FFFF.

Вне зависимости от состояния бита ERL и режима работы ссылки к kseg0 не отображаются, а физический адрес получается вычитанием 0x8000\_0000 из виртуального адреса. Кэшируемость сегмента kseg0 определяется значением поля K0 регистра Config CP0.

#### 2.6.2.3.3 Режим Kernel, пространство 1 режима Kernel (kseg1)

Если в режиме Kernel три старших бита виртуального адреса равны  $101_2$ , выбирается виртуальное адресное пространство kseg1. Это область размером  $2^{29}$  байт (512 МВ), которая расположена внутри границ, определяемых адресами 0xA000\_0000 и 0xBFFF\_FFFF.

Вне зависимости от состояния бита ERL и режима работы ссылки к kseg1 не отображаются, а физический адрес получается вычитанием 0xA000\_0000 из виртуального адреса.

#### 2.6.2.3.4 Режим Kernel, пространство 2 режима Kernel (kseg2)

Если в режиме Kernel три старших бита виртуального адреса равны  $110_2$ , выбирается виртуальное адресное пространство kseg2.

В режиме TLB вне зависимости от состояния бита ERL это виртуальное пространство отображается через TLB и его кэшируемость определяется полем C строки TLB.

В режиме FM вне зависимости от состояния бита ERL это виртуальное пространство зафиксировано в физических адресах 0xC000\_0000 - 0xDFFF\_FFFF и его кэшируемость определяется полем K23 Регистра Config CP0.

#### 2.6.2.3.5 Режим Kernel, пространство 3 режима Kernel (kseg3)

Если в режиме Kernel три старших бита виртуального адреса равны  $111_2$ , выбирается 32-разрядное виртуальное адресное пространство kseg3.

В режиме TLB вне зависимости от состояния бита ERL это пространство отображается через TLB и его кэшируемость определяется полем C строки TLB.

В режиме FM вне зависимости от состояния бита ERL это виртуальное пространство зафиксировано в физических адресах 0xE000\_0000 - 0xFFFF\_FFFF и его кэшируемость определяется полем K23 регистра Config.



### 2.6.3 Буфер быстрого преобразования адреса (TLB)

В этой главе описывается управление памятью с помощью буфера быстрого преобразования адреса (TLB), которое осуществляется в режиме TLB.

В режиме TLB реализуется полностью ассоциативный буфер быстрого преобразования адреса (TLB), содержащий 16 двойных строк, позволяющих отображать 32 виртуальных страницы в соответствующие физические адреса. TLB организовано в виде 16 парных строк – четных и нечетных, содержащих адреса страниц размером от 4 Кбайт до 16 Мбайт, которые хранятся в 4 Гбайтном физическом адресном пространстве. Задача TLB состоит в преобразовании виртуальных адресов и их соответствующего идентификатора адресного пространства (ASID) в физический адрес памяти. Преобразование выполняется путем сравнения старших разрядов виртуального адреса (вместе с битами поля ASID) с каждой из строк тэговой порции TLB и иначе называется поиском соответствия по TLB (поиском соответствия тэга одной из строк виртуальному адресу на входе TLB).

Буфер TLB организован в виде страничных пар для минимизации общего количества хранящейся информации. Каждая строка тэговой порции соответствует двум физическим строкам данных – строке четных страниц и строке нечетных страниц. Самый старший разряд виртуального адреса, не участвующий в сравнении тэгов, определяет какая строка из двух строк данных используется. Поскольку размер страницы может варьироваться для каждой пары страниц, определение адресных разрядов, участвующих в сравнении и разряда, задающего четность страницы, должно осуществляться динамически при поиске по TLB.

На Рисунок 2.17 показано содержание одной из 16 двойных строк TLB.

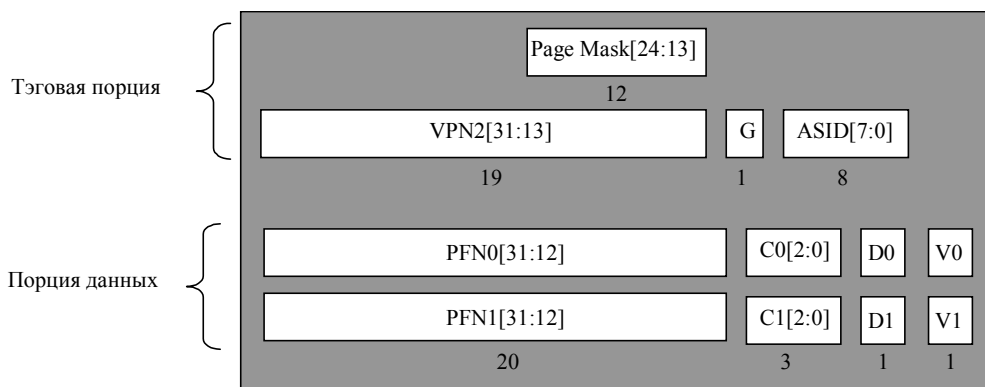


Рисунок 2.17

Описание полей строки TLB приведены в Таблица 2.14.

Таблица 2.14

Название поля	Описание																								
Page Mask[24:13]	Значение маски размера страницы. Определяет размер страницы маскировкой соответствующих разрядов VPN2, и тем самым исключением их из рассмотрения. Также используется для задания адресного разряда, определяющего четность страницы (PFN0-PFN1). См. следующую таблицу:																								
	<table border="1"> <thead> <tr> <th>Page Mask [11:0]</th> <th>Размер страницы</th> <th>Бит определения четности</th> </tr> </thead> <tbody> <tr> <td>0000 0000 0000</td> <td>4 Кб</td> <td>VAddr[12]</td> </tr> <tr> <td>0000 0000 0011</td> <td>16 Кб</td> <td>VAddr[14]</td> </tr> <tr> <td>0000 0000 1111</td> <td>64 Кб</td> <td>VAddr[16]</td> </tr> <tr> <td>0000 0011 1111</td> <td>256 Кб</td> <td>VAddr[18]</td> </tr> <tr> <td>0000 1111 1111</td> <td>1 Мб</td> <td>VAddr[20]</td> </tr> <tr> <td>0011 1111 1111</td> <td>4 Мб</td> <td>VAddr[22]</td> </tr> <tr> <td>1111 1111 1111</td> <td>16 Мб</td> <td>VAddr[24]</td> </tr> </tbody> </table>	Page Mask [11:0]	Размер страницы	Бит определения четности	0000 0000 0000	4 Кб	VAddr[12]	0000 0000 0011	16 Кб	VAddr[14]	0000 0000 1111	64 Кб	VAddr[16]	0000 0011 1111	256 Кб	VAddr[18]	0000 1111 1111	1 Мб	VAddr[20]	0011 1111 1111	4 Мб	VAddr[22]	1111 1111 1111	16 Мб	VAddr[24]
	Page Mask [11:0]	Размер страницы	Бит определения четности																						
	0000 0000 0000	4 Кб	VAddr[12]																						
	0000 0000 0011	16 Кб	VAddr[14]																						
	0000 0000 1111	64 Кб	VAddr[16]																						
	0000 0011 1111	256 Кб	VAddr[18]																						
	0000 1111 1111	1 Мб	VAddr[20]																						
	0011 1111 1111	4 Мб	VAddr[22]																						
1111 1111 1111	16 Мб	VAddr[24]																							
В столбце Page Mask приведены все возможные значения Page Mask. Поскольку каждая пара битов этого поля всегда имеет одинаковое значение, физическая строка в TLB содержит сокращенную версию Page Mask, содержащую только 6 бит. Однако для программы это значение всегда преобразуется в 12-битное.																									
Следует иметь в виду, что при кэшируемых ссылках, страницы размером 4 Кбайт использовать нельзя.																									
VPN2[31:13]	Виртуальный номер страницы, поделенный на 2. Данное поле содержит старшие разряды виртуального номера страницы. Виртуальный номер разделен на 2 потому, что он соответствует паре страниц TLB. Разряды 31:25 всегда участвуют в сравнении. Участие в сравнении разрядов 24:13 зависит от размера страницы, задаваемого полем Page Mask.																								
G	Бит глобальности. Если он установлен, данная строка является глобальной для всех процессов и подпроцессов, и таким образом, поле ASID исключается из рассмотрения.																								
ASID[7:0]	Идентификатор адресного пространства. Определяет процесс или подпроцесс, с которым ассоциируется данная строка TLB.																								
PFN0[31:12], PFN1[31:12]	Физический номер кадра. Задаёт старшие разряды физического адреса. Для страниц размером более 4 Кбайт используется подмножество этого поля.																								
C0[2:0], C1[2:0]	Кэшируемость. Содержит закодированное значение атрибута кэшируемости и определяет должна ли страница помещаться в кэш или нет. Поле кодируется следующим образом:																								
	<table border="1"> <thead> <tr> <th>C[2:0]</th> <th>Атрибуты когерентности</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>При записи преобразуется в код 011</td> </tr> <tr> <td>001</td> <td>При записи преобразуется в код 011</td> </tr> <tr> <td>010</td> <td>Некэшируемая страница</td> </tr> <tr> <td>011</td> <td>Кэшируемая страница</td> </tr> <tr> <td>100</td> <td>При записи преобразуется в код 011</td> </tr> <tr> <td>101</td> <td>При записи преобразуется в код 011</td> </tr> <tr> <td>110</td> <td>При записи преобразуется в код 011</td> </tr> <tr> <td>111</td> <td>При записи преобразуется в код 010</td> </tr> </tbody> </table>	C[2:0]	Атрибуты когерентности	000	При записи преобразуется в код 011	001	При записи преобразуется в код 011	010	Некэшируемая страница	011	Кэшируемая страница	100	При записи преобразуется в код 011	101	При записи преобразуется в код 011	110	При записи преобразуется в код 011	111	При записи преобразуется в код 010						
	C[2:0]	Атрибуты когерентности																							
	000	При записи преобразуется в код 011																							
	001	При записи преобразуется в код 011																							
	010	Некэшируемая страница																							
	011	Кэшируемая страница																							
	100	При записи преобразуется в код 011																							
	101	При записи преобразуется в код 011																							
110	При записи преобразуется в код 011																								
111	При записи преобразуется в код 010																								
D0, D1	“Dirty” (Грязная страница) – бит разрешения записи. Показывает, что в страницу была сделана запись и/или разрешена запись в данную страницу. Если этот бит установлен, разрешены операции сохранения в данной странице. Если не установлен, сохранения в данной странице будут вызывать исключения модификации.																								
V0, V1	Бит валидности. Показывает, что данная строка TLB и, соответственно, отображение виртуальной страницы, действительны. Если этот бит установлен, то обращения к данной странице разрешены. Если не установлен, то обращения к странице будут вызывать исключения TLB (TLB invalid).																								

Для заполнения строки TLB используются команды TLBWI и TLBWR (см. документ “Процессорное ядро RISCore32. Система команд”). Перед запуском этих команд нужно обновить некоторые регистры CP0, записав в них значения, которые будут затем помещены в строку TLB.

- Значение Page Mask задается в регистре Page Mask CP0.
- Значения VPN2 и ASID задаются в регистре EntryHi CP0.
- Значения PFN0, C0, D0, V0 и G задаются в регистре EntryLo0 CP0.
- Значения PFN1, C1, D1, V1 и G задаются в регистре EntryLo1 CP0.

Биты глобальности G входят в оба регистра EntryLo0 и EntryLo1. Бит G строки TLB является результатом логической операции "И", проведенной над битами глобальности из EntryLo0 и EntryLo1. Более подробно эти регистры описаны в разделе 2.7 “Регистры CP0”.

Наличие идентификатора адресного пространства (ASID) дает возможность уменьшить частоту попаданий при поисках по TLB на контекстной основе. Это определяет возможность одновременного существования нескольких процессов как в TLB, так и в кэш команд. Значение ASID хранится в регистре EntryHi и сравнивается со значением ASID каждой строки.

#### **2.6.4 Преобразование виртуального адреса в физический в режиме TLB.**

Преобразование виртуального адреса в физический начинается со сравнения полученного виртуального адреса с виртуальными адресами, хранящимися в TLB. Соответствие имеет место, если виртуальный номер страницы (VPN) адреса совпадает с полем VPN строки TLB с учетом маски, хранящейся в этой строке, а также выполняется одно из двух условий:

- Установлен бит глобальности (G) для четных и нечетных страниц в строке TLB;
- Поле ASID виртуального адреса совпадает с полем ASID строки TLB.

Это соответствие называется попаданием TLB. Если не имеется ни одного соответствия, возникает исключение промаха TLB и программному обеспечению дается возможность пополнить TLB из расположенной в памяти таблицы страниц виртуальных /физических адресов. На Рисунок 2.18 показана логика преобразования виртуального адреса в физический.

На этом рисунке виртуальный адрес расширяется 8-разрядным идентификатором адресного пространства (ASID), который уменьшает частоту попаданий при просмотрах TLB на контекстной основе. Это 8-разрядное поле ASID содержит номер, присвоенный процессу, и хранится в регистре EntryHi CP0.

1. Виртуальный адрес (VA), представленный виртуальным номером страницы (VPN), сравнивается с тэгом из строки TLB (VPN2) с учетом маски (PageMask).
2. Если имеется соответствие, номер страничного кадра (PFN0 или PFN1, в зависимости от значения бита четности – самого старшего бита, не участвующего в сравнении) извлекается и помещается в старшие разряды физического адреса (PA)
3. В младшие разряды физического адреса помещается смещение (Offset), не участвующее в сравнении.

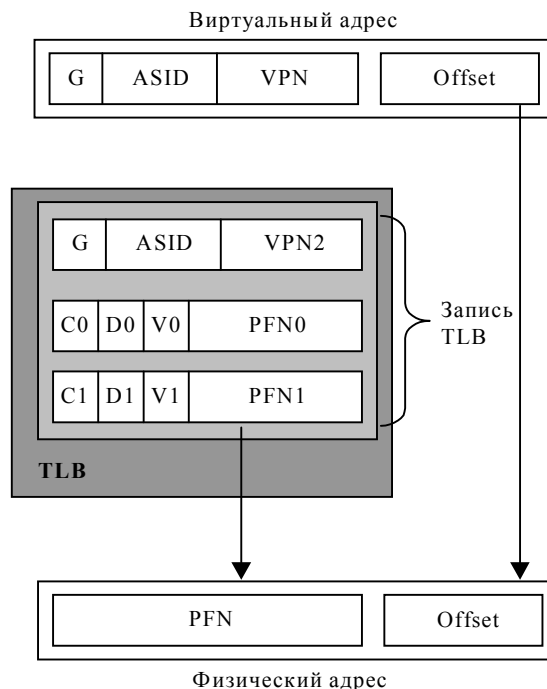


Рисунок 2.18

Когда происходит совпадение виртуальных адресов при поиске по TLB, физический номер кадра (PFN) извлекается из соответствующей физической порции строки TLB и дополняется смещением, взятым из виртуального адреса, формируя, таким образом, физический адрес. Смещение представляет собой адрес в пределах пространства страничного кадра. Как показано на рисунке, смещение не пропускается через TLB.

На Рисунок 2.19 показана блок-схема процесса преобразования адреса. В верхней части рисунка показан виртуальный адрес для страницы размером 4 Кбайт. Ширина поля смещения определяется размером страницы.

В нижней части рисунка показан виртуальный адрес для страницы размером 16 Мбайт.

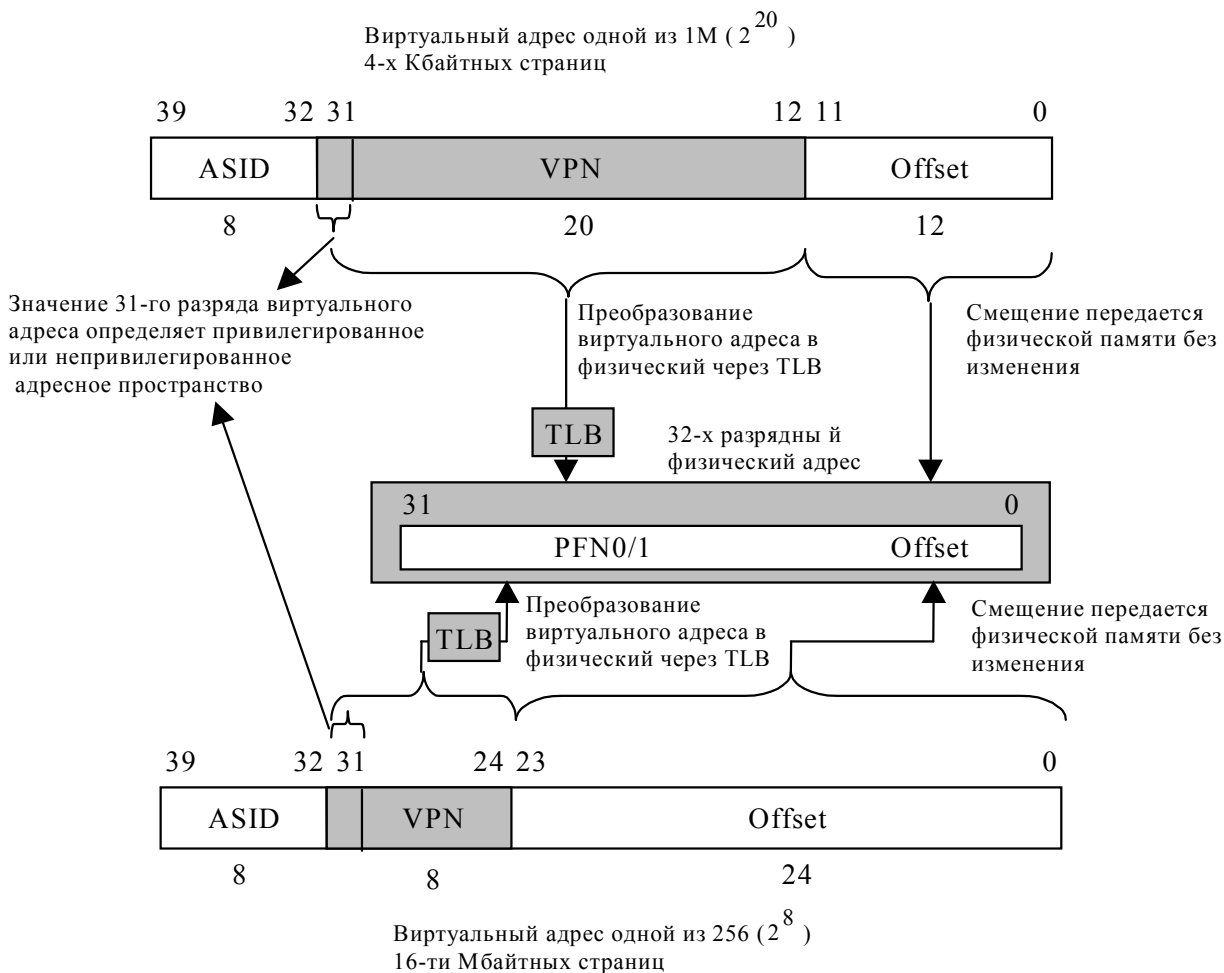


Рисунок 2.19

### 2.6.4.1 Попадания (hits), промахи (misses), и множественные попадания (multiple matches)

Каждая строка TLB содержит тэг и два поля данных. Если найдено соответствие, старшие разряды виртуального адреса заменяются физическим номером кадра (PFN), хранящимся в соответствующей строке массива данных TLB. Способ разбиения памяти при отображении определяется в терминах TLB-страниц. TLB поддерживает страницы различных размеров в пределах от 4 КБ до 16 МБ с шагом по степеням 4. Если соответствие найдено, но строка является запрещенной (т.е., бит V в поле данных равен 0), выработывается исключение TLB Invalid.

Если соответствие не найдено, возникает исключение TLB Refill, и программное обеспечение пополняет TLB из таблицы страниц, находящейся в памяти. На Рисунок 2.20 показан алгоритм преобразования и условия возникновения исключений TLB.

Программное обеспечение может делать записи в конкретные строки TLB или использовать аппаратный механизм записи в случайно выбранные строки. Регистр Random определяет, в какую строку будет сделана запись командой TLBWR. Этот регистр декрементируется на каждом такте продвижения конвейера, возвращаясь к максимальному значению после достижения величины, равной значению регистра Wired. Таким образом, строки TLB, чей номер меньше значения регистра Wired, не затрагиваются ко-

мандой TLBWR, что позволяет зарезервировать TLB-отображения первостепенной важности.

В режиме TLB также реализован механизм сравнения при записи с целью предотвращения возникновения нескольких соответствий (множественных попаданий). Работает он следующим образом. При выполнении операции записи в TLB, поле VPN2 сравнивается с одноименными полями всех строк TLB. Если будет найдено соответствие, возникнет аппаратно обрабатываемое исключение, которое установит бит TS регистра Status CP0 и прервет эту операцию. Подробно исключения описаны в п. 2.7. В каждой строке TLB имеется скрытый бит, обнуляемый при аппаратном сбросе. Устанавливается этот бит при записи в данную строку, разрешая просмотр этой строки при поисках соответствий. Поэтому непроинициализированные строки не вызывают неадекватные преобразования адресов.

Замечание: этот скрытый бит инициализации приводит все строки TLB к запрещенному состоянию после аппаратного сброса, что делает ненужной процедуру очистки (flush) TLB. Но для совместимости с другими MIPS – процессорами рекомендуется заполнять значения тэгов уникальными величинами и обнулять бит валидности (V).

Очистить строку TLB (вывести ее из рассмотрения при поиске) можно, записав в нее значение с неотображаемым через TLB адресом.

Смена размера маски или других переменных строки TLB не приводит к исключению, если она не вводит в противоречие данной строки с другими строками. Например, увеличение размера страницы расширением маски в одной строке TLB может привести к перекрытию данной страницы с другими страницами TLB.

#### **2.6.4.2 Размеры страниц и алгоритм замещения**

Для управления общим количеством отображаемого адресного пространства и характеристиками замещения в различных областях памяти ядро обеспечивает два механизма. Первый заключается в том, что размер страницы может быть задан относительно каждой строки TLB, что позволяет отображать страницы размером от 4 Кбайт до 16 Мбайт (по степеням 4). В регистр Page Mask CP0 загружается требуемый размер страницы, который при выполнении операции записи попадает в очередную строку TLB. Таким образом, операционная система может задавать отображения особых назначений. Например, характерный кадровый буфер (frame buffer) может быть отображен на память всего одной строкой TLB.

Второй механизм управляет замещением, когда возникает промах при просмотре TLB. Для выбора строки TLB, в которую будет записано новое отображение, в процессорном ядре предусмотрен алгоритм случайного замещения. Но существует также способ программно предотвратить случайное замещение зарезервированных отображений, количество которых определяется значением регистра Wired CP0. (см. также п. 2.8.3.6).

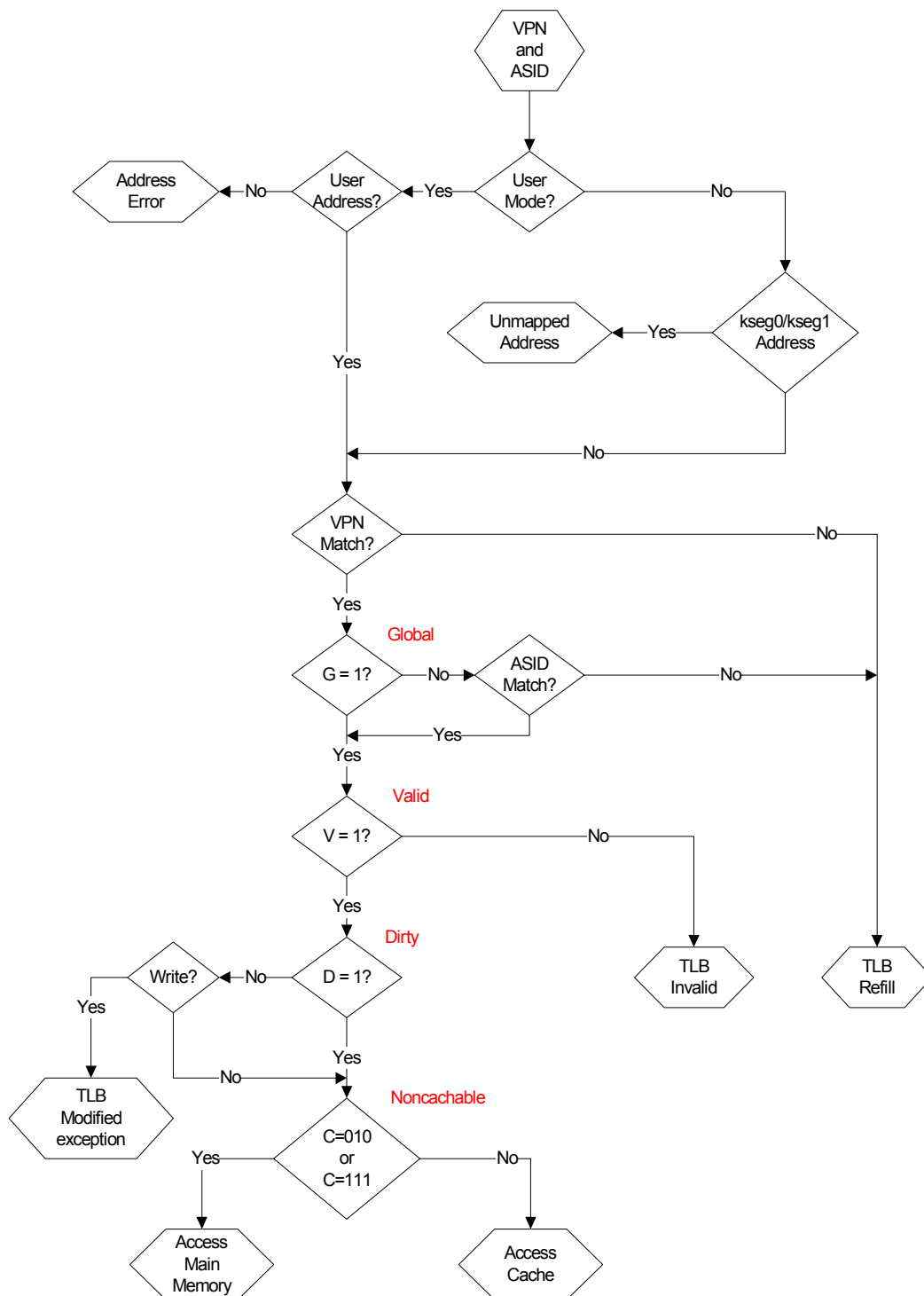


Рисунок 2.20. Алгоритм преобразования адреса через TLB.

## 2.7 Исключения

Процессорное ядро способно принимать исключения от ряда источников, в том числе промах буфера преобразования адресов (TLB), арифметическое переполнение, прерывание ввода-вывода, и системные вызовы. Обнаружив одно из этих исключений, CPU приостанавливает нормальную последовательность исполнения команд и процессор входит в режим Kernel.



В режиме Kernel ядро отключает прерывания и вынуждает процессор запустить программу обработчика исключений, расположенную в фиксированных адресах памяти. Обработчик сохраняет контекст процессора – содержимое счетчика команд, текущий режим процессора и статус разрешения прерываний. Таким образом, контекст может быть восстановлен по завершению обработки исключения.

При возникновении исключения в регистр Exception Program Counter (EPC) загружается адрес, начиная с которого исполнение команд может возобновиться после завершения обработки исключения. В регистр EPC помещается адрес команды, вызвавшей исключение или, если команда находилась в слоте задержки перехода, адрес команды перехода, предшествующей слоту задержки. Чтобы различить эти ситуации, программное обеспечение должно проанализировать бит BD (branch delay) в регистре Cause CP0.

### 2.7.1 Условия исключений

Исключения обрабатываются на стадии M конвейера. Когда исключительная ситуация обнаруживается, команда, находящаяся на стадии M, и все команды, следующие за ней на конвейере, отменяются. Соответственно, все условия остановки конвейера, относящиеся к этой команде, а также условия последующих исключений, которые также могут относиться к ней, игнорируются, поскольку обслуживание приостановок для отмененной команды не приносит выигрыша.

Когда условие исключения обнаруживается на стадии M, процессор заполняет необходимые регистры CP0 значениями, относящимися к состоянию исключения, изменяет счетчик команд (PC) на адрес соответствующего вектора обработки исключения и очищает признаки исключения, относящиеся к более ранним стадиям конвейера.

Такая реализация позволяет завершить исполнение команды, находящейся на стадии W, и запретить завершение последующих команд. Таким образом, значения, сохраненного в регистре EPC (в случае ошибок – в Error PC), достаточно для возобновления исполнения. Это также обеспечивает поступление исключений в соответствии с порядком исполнения команд – команда, вызывающая исключение, может быть уничтожена командой с более поздней стадии конвейера, также вызвавшей исключение.

### 2.7.2 Приоритеты исключений

В Таблица 2.15. перечислены все возможные исключения со своими относительными приоритетами от высшего к низшему. Некоторые из этих исключений могут случаться одновременно, в этом случае вызывается исключение с наивысшим приоритетом.

Таблица 2.15

Исключение	Описание
Reset	Аппаратный сброс
NMI	Внешнее немаскируемое прерывание и прерывание от таймера WDT (см. табл. 7.2).
TLB_Ri, TLB_Ii	Промах TLB при выборке команды, Попадание в запрещенную страницу TLB (V=0) при выборке команды
AdELi	Ошибка выравнивания адреса при выборке команды; Ссылка на адрес режима Kernel при работе в режиме User при выборке команды
MCheck	Запись в TLB, создающая конфликт с существующей строкой TLB
Sys	Выполнение команды SYSCALL
Bp	Выполнение команды BREAK
CpU	Выполнение команды сопроцессора в режиме User
RI	Выполнение зарезервированной команды
Ov	Переполнение в арифметической команде
Tr	Выполнение trap (когда условие trap истинно)

Исключение	Описание
AdELd	Ошибка выравнивания адреса при загрузке данных;
AdES	Ссылка на адрес режима Kernel при работе в режиме User при загрузке данных Ошибка выравнивания адреса при сохранении данных; Попытка сохранения по адресу Kernel в режиме User
TLB_Rd, TLB_Id	Промех TLB при загрузке данных; Попадание в запрещенную страницу TLB (V=0) при загрузке данных
TLB_M	Сохранение в TLB-странице с D=0
Interrupt	Установка немаскируемых HW или SW - прерываний

### 2.7.3 Расположение векторов исключений

Векторы исключений аппаратного сброса и NMI всегда находятся по адресу 0xBFC\_0000. Адреса всех других исключений являются комбинациями векторных смещений и базового адреса. В Таблица 2.16 приведены базовые адреса как функции исключения и состояния бита BEV Регистра Status. В Таблица 2.17. приведены смещения от базового адреса как функции исключения. В Таблица 2.18 эти две таблицы сведены в одну таблицу, содержащую все возможные адреса векторов исключений как функции состояний, влияющих на выбор этих векторов.

Таблица 2.16.

Исключение	Status <sub>BEV</sub>	
	0	1
Reset, NMI	0xBFC0_0000	
Остальные исключения	0x8000_0000	0xBFC0_0200

Таблица 2.17. Базовые адреса векторов исключений

Исключение	Смещение вектора
TLB Refill, EXL = 0	0x000
Reset, NMI	0x000
Исключения общего характера (General Exeptions)	0x180
Interrupt, Cause <sub>IV</sub> = 1	0x200

Таблица 2.18. Векторы исключений

Исключение	BEV	EXL	IV	Вектор
Reset, NMI	-	-	-	0xBFC0_0000
TLB Refill	0	0	-	0x8000_0000
TLB Refill	0	1	-	0x8000_0180
TLB Refill	1	0	-	0xBFC0_0200
TLB Refill	1	1	-	0xBFC0_0380
Interrupt	0	0	0	0x8000_0180
Interrupt	0	0	1	0x8000_0200
Interrupt	1	0	0	0xBFC0_0380
Interrupt	1	0	1	0xBFC0_0400
Остальные	0	-	-	0x8000_0180
Остальные	1	-	-	0xBFC0_0380

### 2.7.4 Обработка общих исключений

Кроме исключений аппаратного сброса и NMI, которые обслуживаются особым образом, обработка всех остальных исключений происходит в соответствии со следующим основным маршрутом:

- Если бит EXL Регистра Состояния (Status) очищен, в регистр EPC загружается значение PC, по которому выполнение программы будет перезапущено, и при необходимости устанавливается бит BD в Регистре Причины (Cause). Если команда не находится в слоте задержки перехода, бит BD в Регистре Причины будет очищен, а в регистр EPC загружается значение, соответствующее текущему PC. Если же команда находится в слоте задержки перехода, бит BD в Регистре Причины устанавливается в “1”, и в EPC загружается значение, равное PC - 4. Если бит EXL в Регистре Состояния установлен, в регистр EPC ничего не загружается, и бит BD в Регистре Причины не модифицируется.
- В поля CE и ExcCode Регистра Причины загружаются значения, соответствующие исключению.
- Устанавливается бит EXL в Регистре Состояния (Status).
- Процессор стартует с вектора исключения.

Значение, загруженное в EPC, представляет собой адрес возврата из исключения и в обычной ситуации программе обработки исключения не требуется его модифицировать. Программе также не нужно просматривать бит BD в Регистре Причины, если не возникает потребность определить действительный адрес команды, вызвавшей исключение.

#### Operation:

```

if StatusEXL == 0 then
  if InstructionInBranchDelaySlot then
    EPC <= PC - 4
    CauseBD <= 1
  else
    EPC <= PC
    CauseBD <= 0
  endif
  if (ExceptionType == TLBRefill) then
    vectorOffset <= 0x000
  elseif (ExceptionType == Interrupt) and
    (CauseIV == 1) then
    vectorOffset <= 0x200
  else
    vectorOffset <= 0x180
  endif
  else
    vectorOffset <= 0x180
  endif
  CauseCE <= FaultingCoprocesorNumber
  CauseExcCode <= ExceptionType
  StatusEXL <= 1
  if (StatusBEV == 1) then
    PC <= 0xBFC0_0200 + vectorOffset
  else
    PC <= 0x8000_0000 + vectorOffset
  endif

```

## 2.7.5 Исключения

В следующих разделах описаны все исключения в порядке, соответствующем табл.2.4.

### 2.7.5.1 Исключение по аппаратному сбросу (*Reset Exception*)

Это немаскируемое исключение, которое происходит при установке сигнала аппаратного сброса. Когда возникает исключение аппаратного сброса, процессор выполняет полную начальную инициализацию, то есть приводит автоматы к начальному состоянию и переводит процессор в состояние, из которого он может начать запуск команд, находящихся в неэкэшируемой и неотображаемой области. После возникновения исключения аппаратного сброса состояние процессора не определено, за исключением следующего:

- Регистр Random устанавливается в значение, равное количеству строк TLB - 1.
- Регистр Wired устанавливается в 0.
- Регистр Config устанавливается в свое начальное состояние (boot state).
- Поля BEV, TS, NMI и ERL Регистра Status устанавливаются в заданные значения.
- В PC загружается значение 0xBFC0\_0000 (виртуальный адрес).

#### **Вектор исключения:**

Reset (0xBFC0\_0000)

#### **Operation:**

Random  $\leq$  TLBEntries - 1  
Wired  $\leq$  0  
Config  $\leq$  ConfigurationState  
Status<sub>BEV</sub>  $\leq$  1  
Status<sub>TS</sub>  $\leq$  0  
Status<sub>NMI</sub>  $\leq$  0  
Status<sub>ERL</sub>  $\leq$  1  
PC  $\leq$  0xBFC0\_0000

### 2.7.5.2 Исключение по немаскируемому прерыванию (*Non Maskable Interrupt – NMI Exception*)

Немаскируемое прерывание возникает по положительному фронту входного сигнала NMI или при срабатывании сторожевого таймера WDT. Исключение NMI происходит только в пределах границ команды, поэтому оно не вызывает сброса или другую переинициализацию аппаратных средств. Состояние кэш, памяти, а также другие состояния процессора остаются неизменными. Значения регистров также сохраняются за исключением следующего:

- Поля BEV, TS, NMI и ERL регистра Status принимают заданные значения.
- В регистр ErrorPC загружается значение PC - 4, если прерывание произошло на фоне команды в слоте задержки перехода. В противном случае в регистр ErrorPC загружается значение PC.
- В PC загружается значение 0xBFC0\_0000.

#### **Вектор исключения:**

Reset (0xBFC0\_0000)

**Operation:**

```
StatusBEV <= 1
StatusTS <= 0
StatusNMI <= 1
```

```
StatusERL <= 1
if InstructionInBranchDelaySlot then
    ErrorEPC <= PC - 4
else
    ErrorEPC <= PC
endif
```

```
PC <= 0xBFC0_0000
```

**2.7.5.3 Исключение по обновлению TLB — выборка команды или доступ к данным (TLB Refill Exception – Instruction Fetch or Data Access)**

Исключение TLB Refill происходит во время выборки команды или доступа к данным, если в TLB нет ни одной строки, соответствующей ссылке к отображенному адресному пространству, и бит EXL в регистре Status равен 0.

**Значение поля ExcCode регистра Cause:**

TLBL: Произошла ссылка по загрузке данных или выборке команды

TLBS: Произошла ссылка по сохранению данных

**Дополнительно сохраняемые состояния:**

Таблица 2.19

Состояние регистра	Значение
BadVAddr	ошибочный адрес
Context	поле BadVPN2 содержит VA <sub>31:13</sub> ошибочного адреса
EntryHi	поле VPN2 содержит VA <sub>31:13</sub> ошибочного адреса; поле ASID содержит ASID отсутствующей ссылки

**Вектор исключения:**

Вектор TLB Refill (смещение 0x000)

**2.7.5.4 Исключение TLB Invalid — выборка команды или доступ к данным (TLB Invalid Exception – Instruction Fetch or Data Access)**

Исключение TLB Invalid происходит во время выборки команды или доступа к данным в одном из следующих случаев:

- В TLB нет ни одной строки, соответствующей ссылке к отображенному адресному пространству, и бит EXL в регистре Status равен 1.
- Строка TLB соответствует ссылке к отображенному адресу, но ее бит валидности выключен.

**Значение поля ExcCode регистра Cause:**

TLBL: Произошла ссылка по загрузке данных или выборке команды

TLBS: Произошла ссылка по сохранению данных

**Дополнительно сохраняемые состояния:**

Таблица 2.20

Состояние регистра	Значение
BadVAddr	ошибочный адрес
Context	поле BadVPN2 содержит VA <sub>31:13</sub> ошибочного адреса
EntryHi	поле VPN2 содержит VA <sub>31:13</sub> ошибочного адреса; поле ASID содержит ASID отсутствующей ссылки

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

**2.7.5.5 Исключение по ошибке адресации — выборка команды / доступ к данным (Address Error Exception – Instruction Fetch / Data Access)**

Исключение по ошибке адресации во время доступа к команде или данным возникает при попытке выполнить одно из следующих действий:

- Выбрать команду, загрузить или сохранить слово данных, если они не выровнены в границах слова
- Загрузить или сохранить половину слова, если оно не выровнено в границах половины слова
- Обратиться по адресу пространства Kernel при работе в режиме User

**Значение поля ExcCode регистра Cause:**

ADEL: Произошла ссылка по загрузке данных или выборке команды

ADES: Произошла ссылка по сохранению данных

**Дополнительно сохраняемые состояния:**

Таблица 2.21

Состояние регистра	Значение
BadVAddr	ошибочный адрес

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

**2.7.5.6 Исключение по аппаратному контролю (Mcheck – Machine Check Exception)**

Данное исключение возникает, если при выполнении команды записи в TLB (TLBWI или TLBWR) обнаруживается, что поле виртуального адреса записываемой строки соответствует такому же полю одной из строк, уже хранящихся в TLB.

При возникновении данной ситуации запись в TLB не выполняется и устанавливается бит TS в регистре Status. Этот бит является статусным и не влияет на функционирование процессорного ядра. Сбрасывается он программно после разрешения данной ситуации, осуществляемого очисткой конфликтных строк в TLB.

**Значение поля ExcCode регистра Cause:**

Mcheck

**Дополнительно сохраняемые состояния:**

Нет

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

### 2.7.5.7 *Исключение исполнения – системный вызов (System Call Exception)*

Исключение System Call является одним из шести исключений исполнения. Все такие исключения имеют одинаковый приоритет. Исключение System Call возникает при исполнении команды SYSCALL.

**Значение поля EcxCode регистра Cause:**

Sys

**Дополнительно сохраняемые состояния:**

Нет

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

### 2.7.5.8 *Исключение исполнения — Breakpoint (Execution Exception – Breakpoint)*

Исключение Breakpoint является одним из шести исключений исполнения. Все такие исключения имеют одинаковый приоритет. Исключение Breakpoint возникает при исполнении команды BREAK.

**Значение поля EcxCode регистра Cause:**

Bp

**Дополнительно сохраняемые состояния:**

Нет

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

### 2.7.5.9 *Исключение исполнения — зарезервированная команда (Execution Exception – Reserved Instruction)*

Исключение зарезервированной команды является одним из шести исключений исполнения. Все такие исключения имеют одинаковый приоритет. Исключение зарезервированной команды вызывается при исполнении команды с неопределенным кодом операции или полем функции.

**Значение поля EcxCode регистра Cause:**

RI

**Дополнительно сохраняемые состояния:**

Нет

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)



#### **2.7.5.10 Исключение исполнения — недоступен сопроцессор (*Execution Exception – Coprocessor Unusable*)**

Исключение недоступности сопроцессора является одним из шести исключений исполнения. Все такие исключения имеют одинаковый приоритет. Исключение недоступности сопроцессора вызывается при попытке исполнения команды сопроцессора CP0 в режиме User.

**Значение поля ExhCode регистра Cause:**

CpU

**Дополнительно сохраняемые состояния:**

Нет

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

#### **2.7.5.11 Исключение исполнения — целочисленное переполнение (*Execution Exception – Integer Overflow*)**

Исключение целочисленного переполнения является одним из шести исключений исполнения. Все такие исключения имеют одинаковый приоритет. Исключение целочисленного переполнения вызывается, когда выбранные целочисленные команды приводят к переполнению в двоичном коде.

**Значение поля ExhCode регистра Cause:**

Ov

**Дополнительно сохраняемые состояния:**

Нет

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

#### **2.7.5.12 Исключение исполнения — Trap (*Execution Exception – Trap*)**

Исключение Trap является одним из шести исключений исполнения. Все такие исключения имеют одинаковый приоритет. Исключение Trap вызывается, если условие команды trap истинно (TRUE).

**Значение поля ExhCode регистра Cause:**

Tr

**Дополнительно сохраняемые состояния:**

Нет

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

### 2.7.5.13 Исключение сохранения в запрещенной области (TLB Modified Exception)

Это исключение возникает при обращении по записи данных к отображенному адресу, если выполняется следующее условие:

- Найденная строка TLB действительна, но страница запрещена для записи.

**Значение поля ExcCode регистра Cause:**

Mod

**Дополнительно сохраняемые состояния:**

Таблица 2.22

Состояние регистра	Значение
BadVAddr	Ошибочный адрес
Context	Поля BadVPN2 содержат VA <sub>31:13</sub> ошибочного адреса
EntryHi	Поле VPN2 содержит VA <sub>31:13</sub> ошибочного адреса; поле ASID содержит ASID отсутствующей ссылки

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180)

### 2.7.5.14 Исключение прерывания (Interrupt Exception)

Исключение прерывания возникает, когда сигнал одного или более разрешенных регистром Status прерываний устанавливается на входе процессора.

**Значение поля ExcCode регистра Cause:**

Int

**Дополнительно сохраняемые состояния:**

Таблица 2.23

Состояние регистра	Значение
Cause <sub>IP</sub>	Указывает код прерывания

**Вектор исключения:**

Общий Вектор исключения (смещение 0x180), если бит IV регистра Cause равен 0;

Вектор прерывания (смещение 0x200), если бит IV регистра Cause равен 1.

## 2.7.6 Алгоритмы обработки исключений

В этом разделе приведены алгоритмы обработки следующих исключений:

- Общие исключения;
- Исключения пропуска при поиске по TLB;
- Исключения Reset и NMI;

Исключения аппаратно обрабатываются, а затем программно обслуживаются.

Алгоритмы обработки исключений приведены на Рисунок 2.21, Рисунок 2.22, Рисунок 2.23.

Все исключения кроме Reset, NMI и TLB-miss первого уровня. Прерывания могут быть замаскированы битами IE и IM

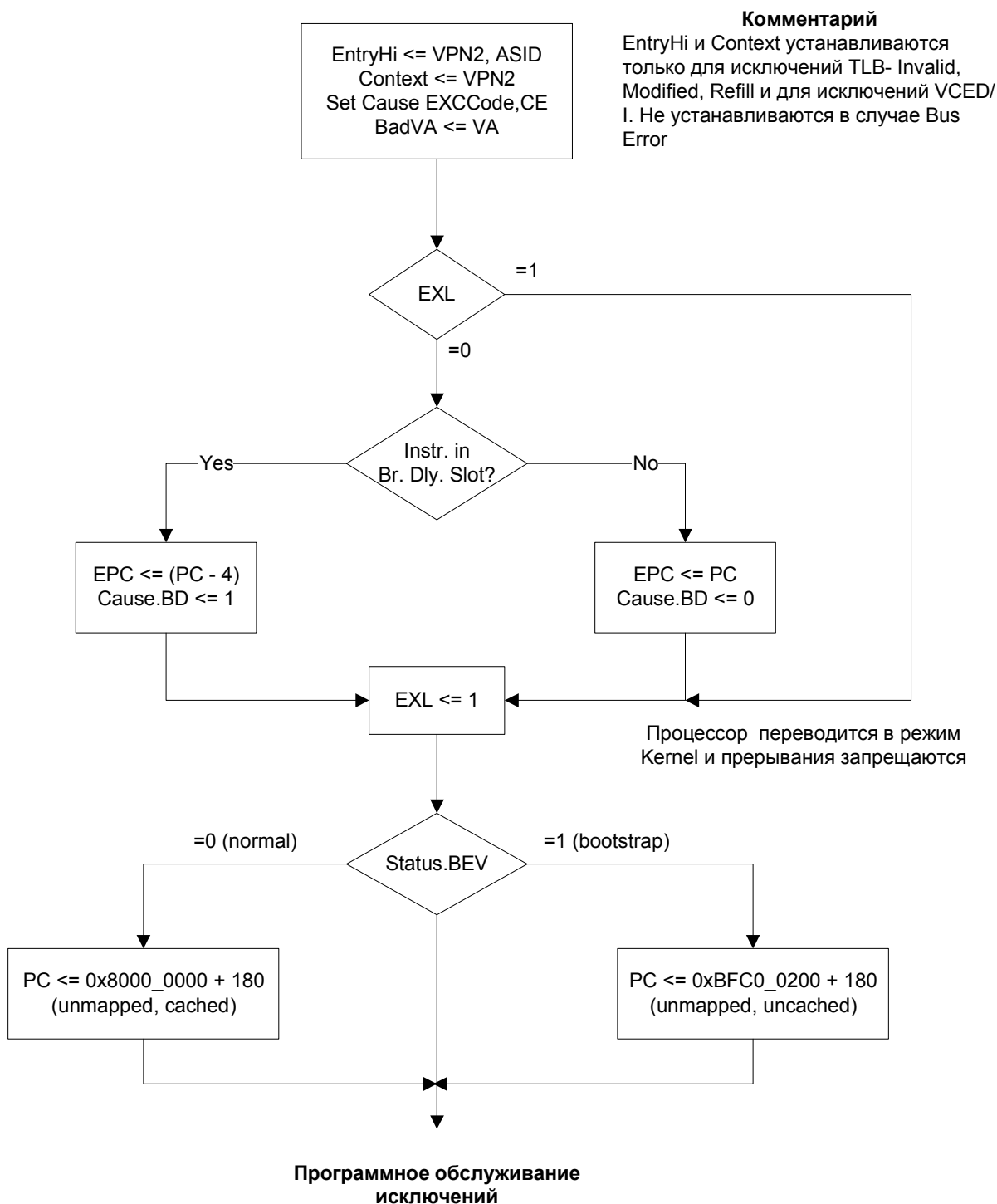


Рисунок 2.21 Обработка общих исключений

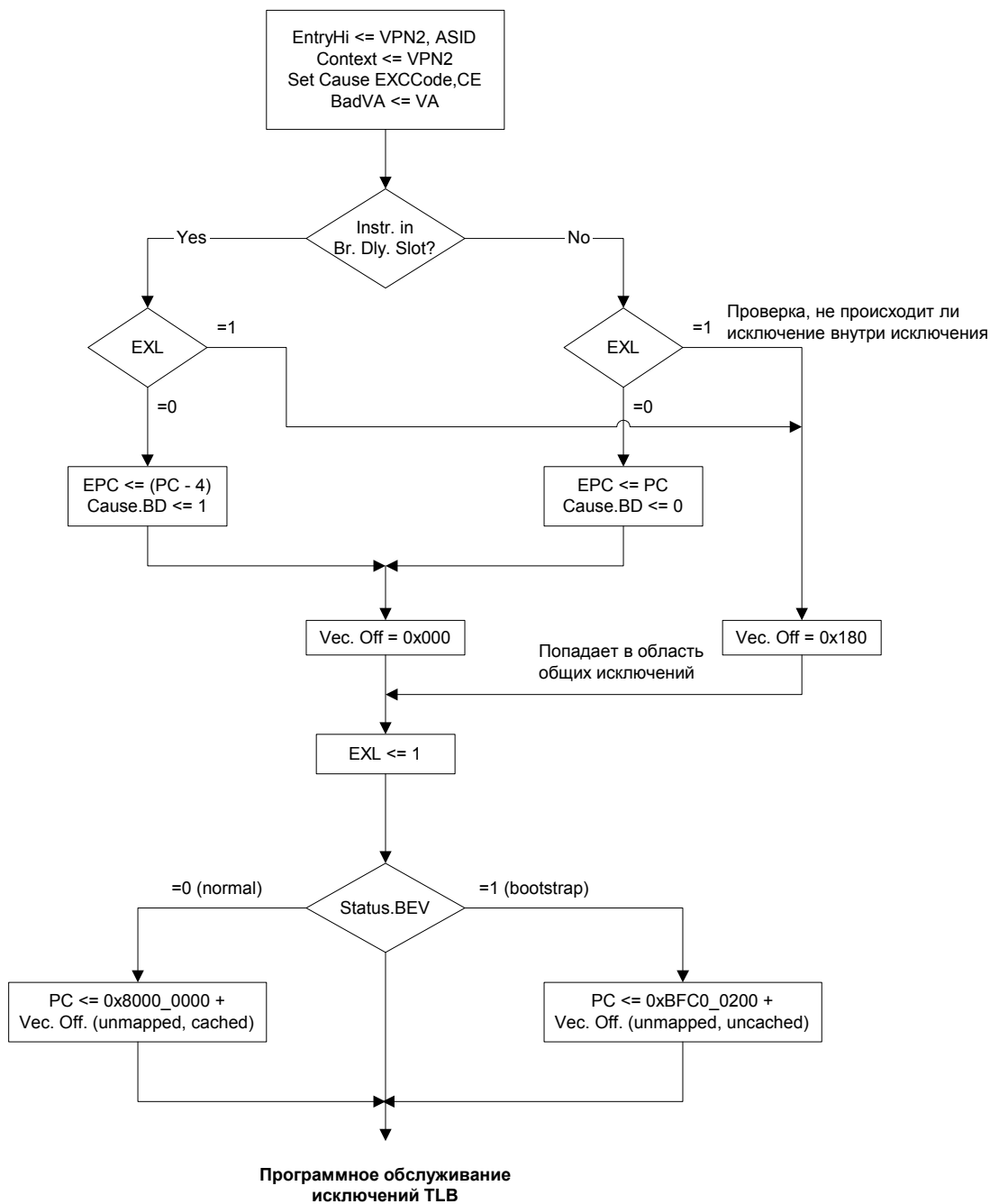


Рисунок 2.22 Обработка исключений TLB Refill и TLB Invalid

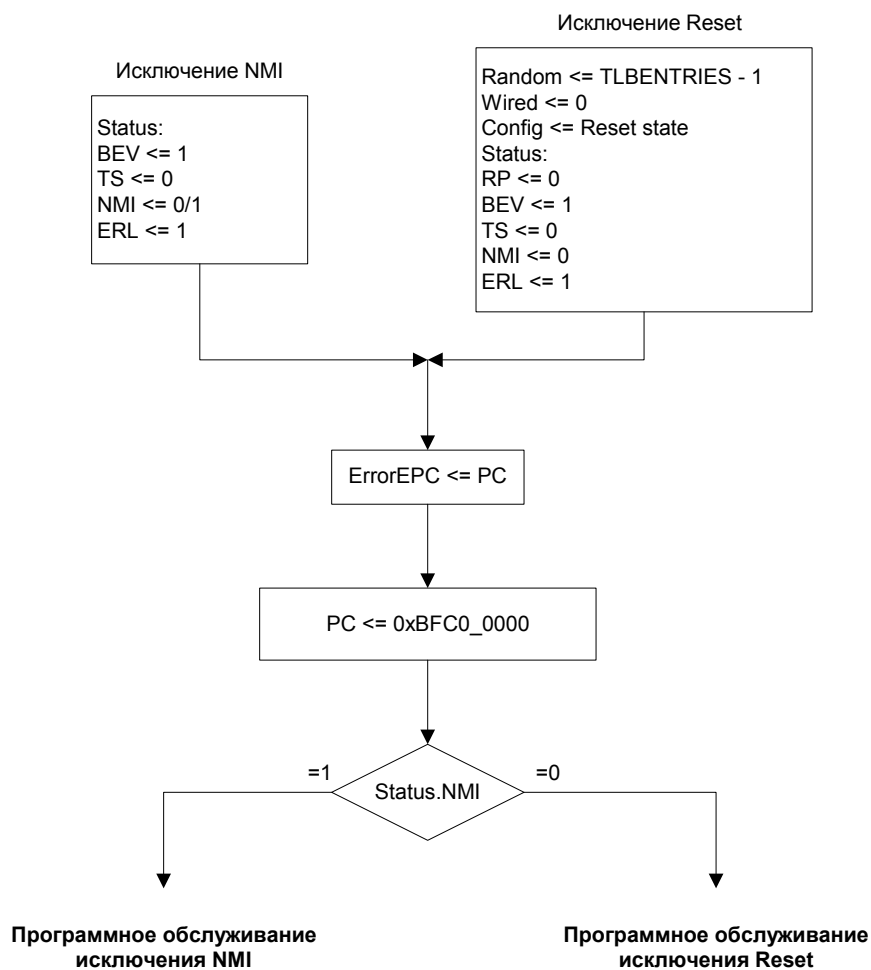


Рисунок 2.23. Обработка исключений Reset и NMI

## 2.8 Регистры CP0

### 2.8.1 Назначение

Системный Управляющий Сопроцессор (CP0) обеспечивает регистровый интерфейс с процессорным ядром MIPS32 и поддерживает управление памятью, преобразование адреса, обработку исключений и другие привилегированные операции. Каждому регистру CP0 соответствует определяющий его уникальный номер; этот номер называется *номером регистра*. Например, регистру PageMask соответствует 5-й номер регистра.

После записи нового значения в регистр CP0 (с помощью команды MTC0), его обновление происходит не сразу, а по прошествии периода от 0 и более команд. Этот период называется периодом особой ситуации.

### 2.8.2 Обзор регистров CP0

В Таблица 2.24. приведены все регистры CP0 в порядке возрастания нумерации. В разделе 5.3 каждый из этих регистров описан отдельно.

Таблица 2.24. Регистры CP0

Номер регистра	Название Регистра	Функция
0	Index <sup>1</sup>	Индекс матрицы TLB (режим TLB)
1	Random <sup>1</sup>	Случайным образом сгенерированный индекс для буфера TLB (режим TLB)
2	EntryLo0 <sup>1</sup>	Младшая часть строки TLB для виртуальных страниц с четными номерами (режим TLB)
3	EntryLo1 <sup>1</sup>	Младшая часть строки TLB для виртуальных страниц с нечетными номерами (режим TLB)
4	Context <sup>2</sup>	Указатель на строку в таблице страниц памяти (режим TLB)
5	PageMask <sup>1</sup>	Управление переменным размером страниц строк TLB (режим TLB)
6	Wired <sup>1</sup>	Управление количеством закрепленных “привязанных” строк TLB (режим TLB)
7	Reserved	Резерв
8	BadVAddr <sup>2</sup>	Содержит адрес, вызвавший последнее связанное с адресацией исключение
9	Count <sup>2</sup>	Счетчик процессорных циклов
10	EntryHi <sup>1</sup>	Старшая часть строки TLB (режим TLB)
11	Compare <sup>2</sup>	Управление прерыванием таймера
12	Status <sup>2</sup>	Состояние и управление процессором
13	Cause <sup>2</sup>	Причина последнего исключения
14	EPC <sup>2</sup>	Значение счетчика команд во время последнего исключения
15	PRId	Идентификация и ревизия процессора
16	Config/Config1	Конфигурационный регистр
17	LLAddr	Загрузка адреса сопряжения
18-19	Не реализованы	
20-22	Reserved	Резерв
23-24	Не реализованы	
25-27	Reserved	Резерв
28-29	Не реализованы	
30	ErrorEPC <sup>2</sup>	Значение счетчика команд при последней ошибке
31	Не реализован	

<sup>1</sup>Регистры, используемые при управлении памятью.

<sup>2</sup>Регистры, используемые при обработке исключений.

### 2.8.3 Регистры CP0

Регистры CP0 обеспечивают интерфейс между системой команд (ISA) и архитектурой процессора. Каждый регистр, описанный в этом разделе, представлен своим порядковым номером и значением поля select.

Все поля описанных регистров характеризуются свойствами записи / чтения, а также значением после аппаратного сброса. Свойства записи / чтения охарактеризованы в Таблица 2.25.

Таблица 2.25

Свойства записи/чтения	Аппаратная интерпретация	Программная интерпретация
R/W	<p>Поле, в котором все биты программно и аппаратно доступны по записи и чтению.</p> <p>Аппаратное обновление этого поля доступно для программы при чтении программой. Программное обновление этого поля доступно для процессора при чтении процессором.</p> <p>Если значение поля после сброса не определено, программа или процессор должны проинициализировать это поле, чтобы первое чтение возвратило предсказуемое значение.</p>	
R	<p>Поле, значение которого постоянно или обновляется только процессором.</p> <p>Значение поля после начальной установки восстанавливается также при включении питания.</p> <p>Если значение поля не определено после начальной установки, процессор обновляет его только при условиях, определенных при описании поля.</p>	<p>Поле, для которого значение, записанное программой, процессором игнорируется. Программное прочтение этого поля возвращает последнее обновленное процессором значение.</p> <p>Если значение поля не определено после начальной установки, программное прочтение этого поля возвратит непредсказуемое значение кроме тех случаев, когда произошло обновление процессором значения этого поля по возникновению условий, определенных в описании поля условий.</p>
0	Поле, значение которого процессором не обновляется и всегда равно нулю.	Программное чтение всегда возвращает нуль.

### 2.8.3.1 Регистр Index (Регистр 0 CP0, Select 0).

Регистр Index является 32-х разрядным регистром, доступным для чтения и записи. Он содержит индекс доступа к TLB для команд TLBP, TLBR и TLBWI. Ширина поля индекса зависит от количества строк TLB и равна 4.

Функционирование процессора НЕОПРЕДЕЛЕНО, если в регистр Index записано значение большее или равное количеству строк TLB.

#### Формат регистра Index

31	30				4	3	0
P				0			Index

Таблица 2.26. Описание полей регистра Index

Поля		Описание	Чтение/Запись	Начальное состояние
Имя	Биты			
P	31	Неудачная проба. Устанавливается в 1, если предыдущей командой TLBProbe (TLBP) не было найдено соответствия в TLB.	R	Не определено
0	30:4	При чтении возвращается нуль	0	0
Index	3:0	Индекс строки TLB, к которой относятся команды TLBRead и TLBWrite	R/W	Не определено

### 2.8.3.2 Регистр Random (Регистр CP0 1, Select 0).

Регистр Random доступен только для чтения, и его значение используется как индекс TLB для команды TLBWR. Ширина поля Random определяется таким же образом, как для регистра Index.

Значение этого регистра изменяется между верхней и нижней границами следующим образом:

- Нижняя граница определяется количеством строк TLB, зарезервированных для использования операционной системой (содержимое регистра Wired). Строка, чей индекс равен значению Wired, является первой из доступных для записи командой TLB Write Random (TLBWR).
- Верхняя граница равна общему количеству строк TLB минус 1.

Регистр Random уменьшается на 1 при продвижении конвейера RISC, возвращаясь к максимальному значению по достижению величины, равной значению регистра Wired.

Процессор инициализирует регистр Random значением, равным верхней границе по возникновению исключения Reset и по записи в регистр Wired.

**Формат регистра Random**

31	4 3 0
0	
Random	

**Таблица 2.27. Описание полей регистра Random**

Поля		Описание	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
0	31:4	При чтении возвращается нуль	0	0
Random	3:0	Случайный индекс строки TLB	R	TLB Entries - 1

### 2.8.3.3 EntryLo0, EntryLo1 (Регистры 2 и 3 CP0, Select 0)

Пара регистров EntryLo действует как интерфейс между TLB и командами TLBR, TLBWI, TLBWR.

В режиме TLB EntryLo0 содержит строки для четных страниц TLB, а EntryLo1 – для нечетных страниц.

После ошибки адресации и возникновения исключений TLB refill, TLB invalid и TLB modified, содержимое регистров EntryLo0 и EntryLo1 не определено.



**Формат регистров EntryLo0, EntryLo1**

31	30	29	26	25	6	5	3	2	1	0	
R		0			PFN			C	D	V	G

**Таблица 2.28. Описание полей регистров EntryLo0 и EntryLo1**

Поля		Описание	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
R	31:30	Резервные. При чтении возвращается нуль	R	0
0	29:26	При чтении возвращается нуль	R	0
PFN	25:6	Номер страничного кадра. Соответствует битам 31:12 физического адреса.	R/W	Не определено
C	5:3	Атрибут когерентности страницы. См. табл.2.18.	R/W	Не определено
D	2	“Dirty” – бит, разрешающий запись. Указывает на то, что в страницу была сделана запись, и/или страница открыта для записи. Если этот бит равен 1, разрешается сохранение в этой странице. Если он равен 0, сохранение в этой странице вызывает исключение TLB Modified.	R/W	Не определено
V	1	Бит валидности. Указывает, на то, что строка TLB и, соответственно, отображение виртуальной страницы, является действительным. Если этот бит равен 1, доступ к странице разрешается. Если этот бит равен 0, доступ к странице вызывает исключение TLB Invalid.	R/W	Не определено
G	0	Бит глобальности. При записи в TLB битом G в строке TLB становится логическое “И” битов G EntryLo0 и EntryLo1. Если бит G строки TLB равен 1, результат сравнения полей ASID игнорируется при поиске по TLB. При чтении строки TLB биты G EntryLo0 и EntryLo1 отражают состояние бита G TLB.	R/W	Не определено

В Таблица 2.29. приведена кодировка для поля C регистров EntryLo0 и EntryLo1 и полей K0, K23 и KU регистра Config.

**Таблица 2.29. Атрибуты когерентности Кэш**

Значение C[5:3]	Описание
0, 1, 3*, 4, 5, 6	Кэшируемая, некогерентная область
2*, 7	Некэшируемая область
* - Архитектура MIPS32 предусматривает только эти два значения. Остальные значения не используются и отображаются в используемые значения. Например, 0, 1, 4, 5 и 6 отображаются в 3, а 7 – в 2.	

### 2.8.3.4 Регистр Context (Регистр 4 CP0, Select 0)

Регистр Context доступен для чтения и записи, и содержит указатель на строку в матрице PTE (page table entry). Эта матрица является структурой данных операционной системы, в которой содержатся преобразования виртуального адреса в физический. При возникновении промаха TLB, операционная система загружает в TLB недостающее преобразование из матрицы PTE. Регистр Context дублирует часть информации, содержащейся в регистре BadVAddr, но организован таким образом, что операционная система может прямо ссылаться к 8-байтной матрице PTE в памяти.

При возникновении исключения TLB (TLB Refill, TLB Invalid, или TLB Modified) биты VA<sub>31:13</sub> виртуального адреса записываются в поле BadVPN2 регистра Context. Поле PTEBase записывается и используется операционной системой.

После возникновения исключения ошибки адресации значение поля BadVPN2 регистра Context не определено.

#### Формат регистра Context

31	23	22	4	3	0
PTEBase		BadVPN2			

Таблица 2.30. Описание полей регистра Context

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
PTEBase	31:23	Это поле используется операционной системой и обычно содержит значение, позволяющее операционной системе использовать регистр Context в качестве указателя на текущую матрицу PTE в памяти.	R/W	Не определено
BadVPN2	22:4	Это поле заполняется процессором при промахе TLB. Оно содержит биты VA <sub>31:13</sub> пропущенного виртуального адреса	R	Не определено
0	3:0	При чтении возвращается нуль	0	0

### 2.8.3.5 Регистр PageMask (Регистр 5 CP0, Select 0)

Регистр PageMask доступен для чтения и записи, и используется для чтения TLB и записи в TLB. Он содержит маску сравнения, которая устанавливает переменную размера страниц для каждой строки TLB, как показано в Таблица 2.32.

Если значение регистра отлично от значений, приведенных в таблице, поведение процессора при поиске по TLB не определено.

Формат регистра PageMask

31	25	24	13	12	0
0	Mask			0	

Таблица 2.31. Описание полей регистра PageMask

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
Mask	24:13	Бит маски, содержащий “1”, указывает на то, что соответствующий бит виртуального адреса не должен принимать участие при поиске соответствия по TLB	R/W	Не определено
0	31:25, 12:0	При чтении возвращается нуль	0	0

Таблица 2.32. Таблица возможных значений поля Mask регистра PageMask

Размер страницы	Бит											
	24	23	22	21	20	19	18	17	16	15	14	13
4 Кбайт	0	0	0	0	0	0	0	0	0	0	0	0
16 Кбайт	0	0	0	0	0	0	0	0	0	0	1	1
64 Кбайт	0	0	0	0	0	0	0	0	1	1	1	1
256 Кбайт	0	0	0	0	0	0	1	1	1	1	1	1
1 Мбайт	0	0	0	0	1	1	1	1	1	1	1	1
4 Мбайт	0	0	1	1	1	1	1	1	1	1	1	1
16 Мбайт	1	1	1	1	1	1	1	1	1	1	1	1

### 2.8.3.6 Регистр Wired (Регистр 6 CP0, Select 0)

Регистр Wired доступен для чтения и записи. Этот регистр определяет границу между случайными и “привязанными” строками TLB, как показано на Рисунок 2.24. Ширина поля Wired определяется так же, как для описанного выше регистра Index. “Привязанные” строки зафиксированы, то есть они не являются удаляемыми и не могут быть перезаписаны командой TLBWR. Эти строки могут быть перезаписаны только командой TLBWI.

Регистр Wired устанавливается в нулевое состояние исключением по аппаратному сбросу (Reset). Запись в регистр Wired вызывает установку регистра Random в значение, равное его верхней границе.

Если значение, записанное в регистр Wired, больше или равно числу строк TLB, операция процессора не определена.

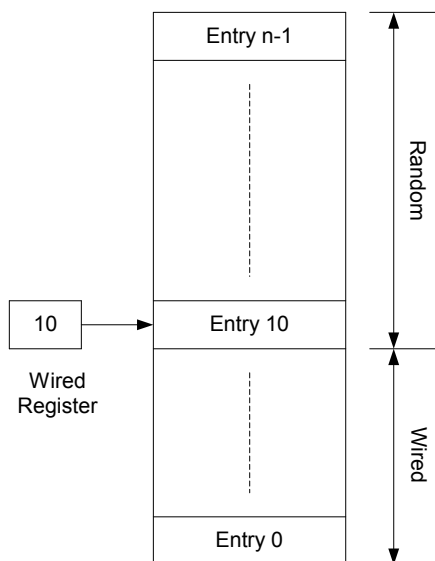


Рисунок 2.24. “Привязанные” и случайные строки TLB

## Формат регистра Wired

31	4 3	0
0		Wired

Таблица 2.33. Описание полей регистра Wired

Поля		Описание	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
0	31:4	При чтении возвращается нуль	0	0
Wired	3:0	Граница между “привязанными” и случайны- ми строками TLB.	R/W	0

**2.8.3.7 Регистр BadVAddr (Регистр 8 CP0, Select 0)**

Регистр BadVAddr доступен только для чтения и содержит последний виртуальный адрес, вызвавший одно из следующих исключений:

- Ошибка адреса (AdEL или AdES)
- TLB Refill
- TLB Invalid
- TLB Modified

## Формат регистра BadVAddr

31	0
BadVAddr	

Таблица 2.34. Описание полей регистра BadVAddr

Поля		Описание	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
BadVAddr	31:0	Виртуальный адрес, вызвавший исключе- ние	R	Не определено

### 2.8.3.8 Регистр Count (Регистр 9 CP0, Select 0)

Регистр Count действует как таймер, увеличивающий свое значение каждый такт.

Регистр Count может быть записан в функциональных или диагностических целях, включая установку или синхронизацию процессора.

Формат регистра Count

31	0
COUNT	

Таблица 2.35. Описание полей регистра Count

Поля		ОПИСАНИЕ	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
Count	31:0	Счетчик	R/W	Не определено

### 2.8.3.9 Регистр EntryHi (Регистр 10 CP0, Select 0)

Регистр EntryHi содержит информацию соответствия виртуального адреса, используемая при чтении, записи и операциях доступа к TLB.

При возникновении исключений TLB (TLB Refill, TLB Invalid или TLB Modified) биты VA<sub>31:13</sub> виртуального адреса записываются в поле VPN2 регистра EntryHi. В поле ASID, которое используется в процессе сравнения при поиске по TLB, программно записывается идентификатор текущего адресного пространства.

Поле VPN2 регистра EntryHi не определено после прерывания по ошибке адресации.

Формат регистра EntryHi

31	0
VPN2	0
	ASID

Таблица 2.36. Описание полей регистра EntryHi

Поля		Описание	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
VPN2	31:13	Разряды VA <sub>31:0</sub> виртуального адреса (виртуальный номер страницы, деленный на 2). Это поле записывается аппаратно при исключении TLB или при чтении TLB, и программно перед записью в TLB.	R/W	Не определено
0	12:8	При чтении возвращается нуль	0	0
ASID	7:0	Идентификатор адресного пространства. Это поле записывается аппаратно при чтении TLB, и программно при установке текущего значения ASID для записи в TLB и для сравнения при поиске по TLB с соответствующими полями ASID в строках TLB.	R/W	Не определено

### 2.8.3.10 Регистр Compare (Регистр 11 CP0, Select 0)

Регистр Compare действует совместно с регистром Count с целью реализации функции таймера и прерывания по таймеру.

Результат сравнения регистров Count и Compare заведен на 15 разряд регистра Cause. Когда значение регистра Count равняется значению регистра Compare, этот бит имеет единичное состояние. Он остается в этом состоянии, пока в регистр Compare не будет произведена запись.

Для диагностических целей регистр Compare доступен для чтения и записи. Однако при нормальном функционировании регистр Compare используется только для записи. При записи значения в регистр Compare в качестве побочного эффекта происходит очистка бита прерывания по таймеру.

Формат регистра Compare

31	0
Compare	

Таблица 2.37. Описание полей регистра Compare

Поля		Описание	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
Compare	31:0	Период счета таймера	R/W	Не определено

### 2.8.3.11 Регистр Status (Регистр 12 CP0, Select 0)

Регистр Status (SR) является регистром, доступным для чтения и записи. Он содержит поля рабочего режима, разрешения прерываний и диагностические состояния процессора. Для задания режимов функционирования процессора, поля этого регистра объединяются следующим образом:

**Разрешение прерываний:** Прерывания разрешаются, когда истинны все следующие условия:

- IE = 1
- EXL = 0
- ERL = 0

Если эти условия выполнены, прерывания разрешаются установкой битов IM.

**Рабочие режимы:** Процессор всегда находится в одном из двух режимов – Kernel или User. Режим задается установкой следующих битов регистра Status CPU.

- Режим User: UM = 1, EXL = 0, and ERL = 0
- Режим Kernel: UM = 0 или EXL = 1 или ERL = 1

Формат Status регистра

31	28	27	26	23	22	21	20	19	18	16	15	8	7	5	4	3	2	1	0
CU3-CU0	0	0	BEV	TS	0	NMI	0	IM7-IM0	0	UM	0	ERL	EXL	IE					

Таблица 2.38. Описание полей регистра Status

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
CU3-CU0	31:28	Не используются	R/W	Не определено
-	27	Не используется	0	0
-	26:23	При чтении возвращается нуль	0	0
BEV	22	Управление размещением векторов исключения: 0: Нормальный 1: Начальная загрузка	R/W	1
TS	21	TLB-закрытие системы. Этот бит устанавливается, если при выполнении команд TLBWI или TLBWR образуется команда, которая приводит к условию закрытия, если оно разрешено. Программа может записывать в этот разряд только 0, чтобы очистить его, и не может вызвать переход этого бита из 0 в 1.	R/W	0
NMI	19	Указывает, что вход в вектор исключения начальной установки был осуществлен по причине возникновения NMI. 0: Не NMI (Аппаратный сброс) 1: NMI Программное обеспечение может записывать в этот бит только 0, чтобы очистить его, и не может записать 1.	R/W	1 для NMI, иначе 0
-	18:16	При чтении возвращается нуль	0	0
IM[7:0]	15:8	Маска прерываний: управление разрешением внешних, внутренних и программных прерываний. Прерывание принимается в случае, если установлен бит IE регистра Status и установлены соответствующие биты как в поле IM[7:0] регистра Status, так и в поле IP[7:0] регистра Cause. 0: Запрос на прерывание не разрешен. 1: Запрос на прерывание разрешен.	R/W	Не определено
-	7:5	При чтении возвращается нуль	0	0
UM	4	Указывает на то, что процессор работает в непривилегированном режиме (User): 0: Процессор работает в привилегированном режиме (Kernel) 1: Процессор работает в непривилегированном режиме (User) Замечание: процессор может также находиться в режиме Kernel, если установлены биты EXL или ERL. Это условие не влияет на состояние бита UM.	R/W	Не определено
-	3	При чтении возвращается нуль	0	0

Таблица 2.38. Продолжение.

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
ERL	2	Уровень ошибки. Устанавливается процессором при возникновении исключений Reset и NMI. 0: Нормальный уровень 1: Уровень ошибки Когда бит ERL установлен: Процессор находится в режиме Kernel. Прерывания запрещены. Команда ERET использует адрес возврата, содержащийся в EtopEPC вместо EPC. kuseg используется как неотображаемая и неэкэшируемая область. Это позволяет иметь доступ к главной памяти при ошибках кэш. Поведение процессора не определено, если бит ERL установлен при выполнении кода из useg/kuseg.	R/W	1
EXL	1	Уровень Исключения. Устанавливается процессором при возникновении любого исключения, кроме Reset и NMI. 0: Нормальный уровень 1: Уровень исключения Когда бит EXL установлен: Процессор переходит в привилегированный режим (Kernel). Прерывания запрещены. Исключения TLB Refill используют общий вектор исключения вместо вектора TLB Refill. Если происходит другое исключение, EPC не модифицируется.	R/W	Не определено
IE	0	Разрешение Прерывания. 0: Отключает прерывания 1: Разрешает прерываниям	R/W	Не определено

### 2.8.3.12 Регистр Cause (Регистр 13 CP0, Select 0)

Регистр Cause, в основном, описывает причину последнего исключения. Кроме того, поля регистра управляют запросами на программные прерывания и определяют вектор, которым обрабатываются прерывания. Все поля регистра Cause, за исключением IP[1:0], IV и WP, доступны только для чтения.

#### Формат регистра Cause

<b>31</b>	<b>30</b>	<b>24</b>	<b>23</b>	<b>22</b>	<b>16</b>	<b>15</b>	<b>10</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>2</b>	<b>1</b>	<b>0</b>
BD	0	IV		0			IP[7:2]	IP[1:0]	0		Exc Code		0	



Таблица 2.39. Описание полей регистра Cause

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
BD	31	Указывает на то, что последнее исключение произошло в слоте задержки перехода: 0: Не в слоте задержки 1: В слоте задержки Замечание: бит BD не модифицируется на новом исключении, если установлен бит EXL.	R	Не определено
0	30:24	При чтении возвращается нуль	0	0
IV	23	Указывает, какой вектор используется для обслуживания исключений прерывания – общий или специальный вектор прерываний: 0: Используется общий вектор исключения (0x180) 1: Используется специальный вектор прерываний (0x200)	R/W	Не определено
0	22:16	При чтении возвращается нуль	0	0
IP[7:2]	15:10	Указывает, какое прерывание установлено: 15 – COMPARE; 14 — прерывание от DSP; 13 - прерывания от SWIC1, SWIC0, MFBSP1, MFBSP0, объединенные по схеме ИЛИ; 12 - прерывания от SRIO1, SRIO0, объединенные по схеме ИЛИ; 11 - прерывания от DMAMem1, DMAMem0, объединенные по схеме ИЛИ; 10 - прерывания от IT, RTT, WDT, Vpout, Vpin, ENET, MBR, PMCh, I2C, UART, nIRQ[3:0], объединенные по схеме ИЛИ	R	Не определено
IP[1:0]	9:8	Управляет запросами программных прерываний (посредством записи «1» в данные разряды): 9: Запрос программного прерывания 1; 8: Запрос программного прерывания 0.	R/W	Не определено
ID	7	Прерывание от встроенных средств отладки программ (OnCD).	R/W	0
Exc Code	6:2	Код исключения — см. Таблица 2.40		
0	1:0	При чтении возвращается нуль	0	0

Таблица 2.40. Описание поля Exc Code регистра Cause

Значение Exc Code	Мнемоника	Описание
0	Int	Прерывание
1	Mod	TLB-исключение модификации
2	TLBL	TLB-исключение (загрузка или вызов команды)
3	TLBS	TLB-исключение (сохранение)
4	AdEL	Прерывание по ошибке адресации (загрузка или вызов команды)
5	AdES	Прерывание по ошибке адресации (сохранение)
6-7	-	Не используются
8	Sys	Системное исключение
9	Bp	Исключение Breakpoint
10	RI	Исключение зарезервированной команды
11	SpU	Исключение недоступности сопроцессора
12	Ov	Исключение целочисленного переполнения
13	Tr	Исключение Trap
14	-	Не используются

Значение Exc Code	Мнемоника	Описание
15	FPE	Исключение от сопроцессора арифметики в формате с плавающей точкой (FPU)
16-23	-	Не используются
24	MCheck	Аппаратный контроль
25-31	-	Не используются

### 2.8.3.13 Регистр EPC (Регистр 14 CP0, Select 0)

Программный счетчик исключения (EPC) является регистром, доступным для чтения и записи. EPC содержит адрес, начиная с которого возобновляется исполнение программы после завершения обработки исключения. Все биты регистра EPC значимы и должны перезаписываться.

Для синхронных (точных) исключений, EPC содержит одно из следующего:

- Виртуальный адрес команды, которая была прямой причиной исключения;
- Виртуальный адрес команды перехода (Branch или Jump), непосредственно предшествующей исключению, если команда, вызвавшая исключение, находится в слоте задержки перехода и установлен бит BD в регистре Cause.

Если установлен бит EXL в регистре Status, процессор не записывает адрес в регистр EPC при возникновении новых исключений. Однако, новое значение можно записать в EPC командой MTC0.

#### Формат регистра EPC

<b>31</b>	<b>0</b>
EPC	

Таблица 2.41. Описание полей регистра EPC

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
EPC	31:0	Программный счетчик исключения	R/W	Не определено

### 2.8.3.14 Регистр PRId (Регистр 15 CP0, Select 0)

Регистр идентификации процессора (PRId) – это 32-х разрядный регистр, доступный только для чтения. Он содержит информацию, идентифицирующую изготовителя, опции изготовителя, идентификацию процессора, и версию процессора.

#### Формат регистра PRId

<b>31</b>	<b>24 23</b>	<b>16 15</b>	<b>8 7</b>	<b>0</b>
R	Company ID	Processor ID	Revision	

Таблица 2.42. Описание полей регистра PRId

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
R		При чтении возвращается нуль	R	0
Company ID	23:16	Идентификация компании, которая проектировала или изготавливала процессор.	R	1010
Processor ID	15:8	Идентификация типа процессора.	R	10010
Revision	7:0	Номер версии процессора. Позволяет программам различать разные версии одного типа процессора.	R	0

### 2.8.3.15 Регистр Config (Регистр 16 CP0, Select 0)

Регистр Config определяет различную конфигурационную информацию, а также информацию о возможностях процессора. Большинство полей регистра Config инициализируется аппаратно при выполнении исключения Reset или имеет постоянное значение, и только поле K0 должно быть проинициализировано программно обработчиком исключения Reset.

Формат регистра Config

31	30	28	27	25	24	21	20	19	18	17	16	15	14	13	12	10	9	7	6	3	2	0
M	K23	KU	0	MDU	R	MM	BM	BE	AT	AR	MT	0	K0									

Таблица 2.43. Описание полей регистра Config

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
M	31	Этот бит аппаратно устанавливается в высокий уровень, указывая на наличие регистра Config1	R	1
K23	30:28	Это поле управляет кэшируемостью адресных сегментов kseg2 и kseg3 в режиме FM. В режиме TLB не используется. См. табл.2.33.	FM:R/W	FM:010
			TLB:R	TLB:000
KU	27:25	Это поле управляет кэшируемостью адресных сегментов kuseg и useg в режиме FM. В режиме TLB не используется. См. табл.2.33.	FM:R/W	FM:010
			TLB:R	TLB:000
0	24:21	Не используются	0	0
MDU	20	Тип MDU: итеративный умножитель и делитель	R	1
R	19	При чтении возвращается нуль	0	0
MM	18:17	Режим No Merging для 32 bit collapsing write buffer	R	0
BM	16	Тип передачи Burst: последовательный	R	0
BE	15	Режим endian: Little endian	R	0
AT	14:13	Тип архитектуры, реализованной процессором: MIPS32.	R	0
AR	12:10	Номер версии: 1	R	0
MT	9:7	Тип MMU: 1: Стандартный TLB (FM = 0) 3: Фиксированное отображение (FM = 1) 0, 2, 4-7: зарезервированы	R	TLB: 01
				FM: 11
R	6:3	При чтении возвращается нуль	0	0
K0	2:0	Алгоритм когерентности для kseg0, см. Таблица 2.29.	R/W	010

Таблица 2.44. Атрибуты когерентности кэш

Значение C[5:3]	
0, 1, 3*, 4, 5, 6	Кэшируемая, некогерентная область
2*, 7	Некэшируемая область
* - Архитектура MIPS32 предусматривает только эти два значения. Остальные значения не используются и отображаются в используемые значения. Например, 0, 1, 4, 5 и 6 отображаются в 3, а 7 – в 2.	

### 2.8.3.16 Регистр Config1 (Регистр 16 CP0, Select 1)

Регистр Config1 является дополнением к регистру Config и кодирует дополнительную информацию о возможностях процессора. Все поля регистра Config1 доступны только для чтения.

**Формат регистра Config1**

31	30	25	24	22	21	19	18	16	15	13	12	10	9	7	6	5	4	3	2	1	0
R	MMUSize	IS	IL	IA	DS	DL	DA	R	PC	WR	CA	EP	FP								

**Таблица 2.45. Описание полей Config1 регистра**

Поля		Описание	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
R	31	При чтении возвращается нуль	0	0
Размер MMU	30:25	Это поле содержит количество строк TLB минус 1. В режиме TLB возвращается код 15 в десятичном формате, в режиме Fixed Mapping – 0.	R	001111 (FM =0)
				000000 (FM =1)
IS	24:22	Количество наборов кэш команд: резервная опция	R	111
IL	21:19	Размер строки кэш команд: 16 байт	R	011
IA	18:16	Тип кэш команд: Direct mapped	R	0
DS	15:13	Нет кэш данных	R	0
DL	12:10	Нет кэш данных	R	0
DA	9:7	Нет кэш данных	R	0
R	6:5	При чтении возвращается нуль	0	0
PC	4	Нет регистра Performance Counter	R	0
WR	3	Нет регистра WATCH	R	0
CA	2	Не реализовано	R	0
EP	1	EJTAG не реализован	R	0
FP	0	Нет плавающей арифметики	R	0

**2.8.3.17 Регистр LLAddr – Load Linked Address (Регистр 17 CP0, Select 0)**

Регистр LLAddr содержит физический адрес последней команды Load Linked (LL). Этот регистр используется только для диагностических целей.

**Формат LLAddr регистра**

31	28	27	0
0	Paddr[31:4]		

**Таблица 2.46. Описание полей LLAddr регистра**

Поля		Описание	Чтение/ Запись	Начальное со- стояние
Имя	Биты			
0	31:28	При чтении возвращается нуль	0	0
Paddr[31:4]	27:0	Физический адрес последней команды LL	R	Не определено

### 2.8.3.18 Регистр ErrorEPC (Регистр 30 CP0, Select 0)

Доступный для чтения и записи, регистр ErrorEPC полностью подобен регистру EPC, но используется при возникновении исключений ошибок. Все биты регистра ErrorEPC значимы и должны перезаписываться. Регистр ErrorEPC также используется для сохранения значения счетчика команд при возникновении исключений Reset и немаскируемого прерывании (NMI).

Регистр ErrorEPC содержит виртуальный адрес, начиная с которого может возобновиться исполнение программы после обработки ошибочной ситуации.

Этот адрес может быть:

- Виртуальным адресом команды, вызвавшей исключение;
- Виртуальным адресом команды перехода (Branch или Jump), непосредственно предшествующей исключению, если команда, вызвавшая ошибку, находится в слоте задержки перехода.

В отличие от регистра EPC, для регистра ErrorEPC не имеется соответствующего признака слота задержки перехода.

#### Формат регистра ErrorEPC

31		0
ErrorEPC		

Таблица 2.47. Описание полей регистра ErrorEPC

Поля		Описание	Чтение/ Запись	Начальное состояние
Имя	Биты			
ErrorEPC	31:0	Счетчик команд при исключении ошибки	R/W	Не определен

Регистры WatchLo, WatchHi, Debug, DEPC, TagLo, DataLo, DeSave не реализованы

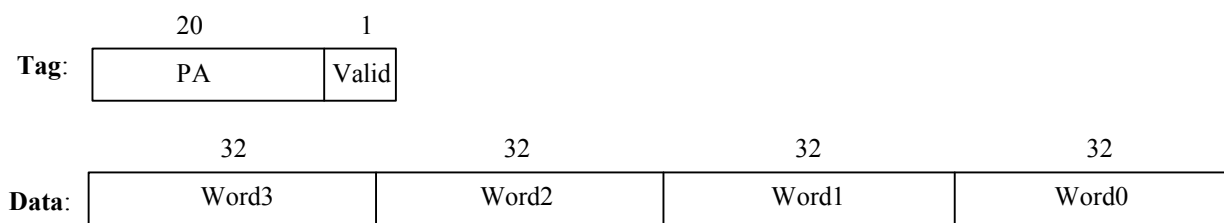
## 2.9 Кэш

CPU имеет кэш команд и кэш данных типа direct mapped объемом по 16 Кбайт. Кэш данных работает по протоколу write-through.

Кэш состоит из двух массивов – массива тэгов и массива данных. Кэш индексируется виртуально, поскольку для выбора соответствующей строки в обоих массивах используется виртуальный адрес. Это позволяет осуществлять доступ к кэш параллельно с преобразованием виртуального адреса в физический. Контроль осуществляется по физическому тэгу, так-так массив тэгов содержит физический, а не виртуальный адрес.

На **Рисунок 2.25** представлен формат каждой строки массивов тэгов и данных. Тэговая строка содержит 20 старших бита физического адреса (биты [31:12]) и бит валидности.

Строка данных содержит 4 32-х разрядных слова – всего 16 байт. До получения всей строки кэш конвейер останавливается.



**Рисунок 2.25** Формат массива кэш

Кэш имеет только два атрибута кэшируемости. Область может быть либо кэшируемой, либо некэшируемой (см. Таблица 2.44).

## 2.10 Карта памяти CPU

Карта физической памяти CPU приведена в Таблица 2.48. Здесь и далее, если это не оговорено специально, коды адреса и данных указаны в шестнадцатеричной системе счисления. Объемы областей памяти указаны с учетом ее дальнейшего расширения.

Таблица 2.48. Карта физической памяти CPU

Диапазон адресов	Название области	Объем области, Мбайт
FFFF_FFFF 2000_0000	Внешняя память	3584
1FFF_FFFF 1C00_0000	Внешняя память (как правило, постоянное запоминающее устройство - ПЗУ)	64
1BFF_FFFF 1800_0000	Внутренняя память	64
17FF_FFFF 0000_0000	Внешняя память	384

Внешняя память доступна через порт внешней памяти общего назначения (MPORT) и порты DDR\_PORT.

Для CPU все адресное пространство памяти является 32-разрядным. Память SRAM, а также внешняя память, могут адресоваться с точностью до байта.

Для указания разрядности блока внешней памяти в регистрах CSCON0:CSCON3 порта внешней памяти MPORT имеется бит W64: 0 – блок 32-разрядный, 1 – блок 64-разрядный. Данные в 64-разрядном сегменте располагаются следующим образом:

Номер 64-разрядного слова	Адрес старшей 32-разрядной части (H)	Адрес младшей 32-разрядной части (L)
0	0x0000_0004	0x0000_0000
1	0x0000_000C	0x0000_0008
2	0x0000_0014	0x0000_0010
3	0x0000_001C	0x0000_0018

Адресом 64-разрядного слова является адрес его младшей части. Для программ CPU разрядность блоков внешней памяти неразличима.

Карта внутренней памяти микросхемы 1892ВМ7Я приведена в Таблица 2.49.

**Таблица 2.49. Карта внутренней памяти**

Диапазон адресов	Название области
1BFF_FFFF 1B00_0000	Окно для программного обмена данными CPU с шиной PCI
1AFF_FFFF	
1908_028F 1908_027F	Резерв
1840_0000	
183F_FFFF 1830_0000	Память и регистры DSP
182F_FFFF	
182F_0000 182E_FFFF	Резерв
1802_0000	
1801_FFFF 1800_0000	Регистры CPU
	Резерв
	Память СРАМ

Перечень программно доступных регистров для CPU приведен в Таблица 2.50.



Таблица 2.50

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры DMA MEM_CH0</u>		
CSR_MEM_CH00	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0000
CP_MEM_CH00	Регистр указателя цепочки	182F_0004
IR0_MEM_CH00	Регистр индекса 0	182F_0008
IR1_MEM_CH00	Регистр индекса 1	182F_000C
OR_MEM_CH00	Регистр смещений	182F_0010
Y_MEM_CH00	Регистр параметров направления Y при двухмерной адресации	182F_0014
RUN_MEM_CH00	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH00 На чтение: Регистр управления и состояния канала MEM_CH0 без сброса битов "END" и "DONE"	182F_0018
CSR_MEM_CH01	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0080
CP_MEM_CH01	Регистр указателя цепочки	182F_0084
IR0_MEM_CH01	Регистр индекса 0	182F_0088
IR1_MEM_CH01	Регистр индекса 1	182F_008C
OR_MEM_CH01	Регистр смещений	182F_0090
Y_MEM_CH01	Регистр параметров направления Y при двухмерной адресации канала	182F_0094
RUN_MEM_CH01	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH01 На чтение: Регистр управления и состояния канала MEM_CH1 без сброса битов "END" и "DONE"	182F_0098
CSR_MEM_CH02	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0100
CP_MEM_CH02	Регистр указателя цепочки	182F_0104
IR0_MEM_CH02	Регистр индекса 0	182F_0108
IR1_MEM_CH02	Регистр индекса 1	182F_010C
OR_MEM_CH02	Регистр смещений канала MEM_CH2	182F_0110
Y_MEM_CH02	Регистр параметров направления Y при двухмерной адресации канала MEM_CH2	182F_0114
RUN_MEM_CH02	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH02 На чтение: Регистр управления и состояния канала MEM_CH2 без сброса битов "END" и "DONE"	182F_0118
CSR_MEM_CH03	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0180
CP_MEM_CH03	Регистр указателя цепочки	182F_0184
IR0_MEM_CH03	Регистр индекса 0	182F_0188
IR1_MEM_CH03	Регистр индекса 1	182F_018C
OR_MEM_CH03	Регистр смещений	182F_0190
Y_MEM_CH03	Регистр параметров направления Y при двухмерной адресации	182F_0194
RUN_MEM_CH03	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH03 На чтение: Регистр управления и состояния канала MEM_CH3 без сброса битов "END" и "DONE"	182F_0198

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры DMA MEM_CH0</u>		
CSR_MEM_CH04	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0200
CP_MEM_CH04	Регистр указателя цепочки	182F_0204
IR0_MEM_CH04	Регистр индекса 0	182F_0208
IR1_MEM_CH04	Регистр индекса 1	182F_020C
OR_MEM_CH04	Регистр смещений	182F_0210
Y_MEM_CH04	Регистр параметров направления Y при двухмерной адресации	182F_0214
RUN_MEM_CH04	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH00 На чтение: Регистр управления и состояния канала MEM_CH0 без сброса битов "END" и "DONE"	182F_0218
CSR_MEM_CH05	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0280
CP_MEM_CH05	Регистр указателя цепочки	182F_0284
IR0_MEM_CH05	Регистр индекса 0	182F_0288
IR1_MEM_CH05	Регистр индекса 1	182F_028C
OR_MEM_CH05	Регистр смещений	182F_0290
Y_MEM_CH05	Регистр параметров направления Y при двухмерной адресации канала	182F_0294
RUN_MEM_CH05	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH01 На чтение: Регистр управления и состояния канала MEM_CH1 без сброса битов "END" и "DONE"	182F_0298
CSR_MEM_CH06	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0300
CP_MEM_CH06	Регистр указателя цепочки	182F_0304
IR0_MEM_CH06	Регистр индекса 0	182F_0308
IR1_MEM_CH06	Регистр индекса 1	182F_030C
OR_MEM_CH06	Регистр смещений канала MEM_CH2	182F_0310
Y_MEM_CH06	Регистр параметров направления Y при двухмерной адресации канала MEM_CH2	182F_0314
RUN_MEM_CH06	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH02 На чтение: Регистр управления и состояния канала MEM_CH2 без сброса битов "END" и "DONE"	182F_0318
CSR_MEM_CH07	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0380
CP_MEM_CH07	Регистр указателя цепочки	182F_0384
IR0_MEM_CH07	Регистр индекса 0	182F_0388
IR1_MEM_CH07	Регистр индекса 1	182F_038C
OR_MEM_CH07	Регистр смещений	182F_0390
Y_MEM_CH07	Регистр параметров направления Y при двухмерной адресации	182F_0394
RUN_MEM_CH07	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH03 На чтение: Регистр управления и состояния канала MEM_CH3 без сброса битов "END" и "DONE"	182F_0398

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры DMA MEM_CH1</u>		
CSR_MEM_CH10	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0400
CP_MEM_CH0	Регистр указателя цепочки	182F_0404
IR0_MEM_CH10	Регистр индекса 0	182F_0408
IR1_MEM_CH10	Регистр индекса 1	182F_040C
OR_MEM_CH10	Регистр смещений	182F_0410
Y_MEM_CH10	Регистр параметров направления Y при двухмерной адресации	182F_0414
RUN_MEM_CH10	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH10 На чтение: Регистр управления и состояния канала MEM_CH0 без сброса битов "END" и "DONE"	182F_0418
CSR_MEM_CH11	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0480
CP_MEM_CH11	Регистр указателя цепочки	182F_0484
IR0_MEM_CH11	Регистр индекса 0	182F_0488
IR1_MEM_CH11	Регистр индекса 1	182F_048C
OR_MEM_CH11	Регистр смещений	182F_0490
Y_MEM_CH11	Регистр параметров направления Y при двухмерной адресации канала	182F_0494
RUN_MEM_CH11	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH11 На чтение: Регистр управления и состояния канала MEM_CH1 без сброса битов "END" и "DONE"	182F_0498
CSR_MEM_CH12	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0500
CP_MEM_CH12	Регистр указателя цепочки	182F_0504
IR0_MEM_CH12	Регистр индекса 0	182F_0508
IR1_MEM_CH12	Регистр индекса 1	182F_050C
OR_MEM_CH12	Регистр смещений канала MEM_CH2	182F_0510
Y_MEM_CH12	Регистр параметров направления Y при двухмерной адресации канала MEM_CH2	182F_0514
RUN_MEM_CH12	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH12 На чтение: Регистр управления и состояния канала MEM_CH2 без сброса битов "END" и "DONE"	182F_0518
CSR_MEM_CH13	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0580
CP_MEM_CH13	Регистр указателя цепочки	182F_0584
IR0_MEM_CH13	Регистр индекса 0	182F_0588
IR1_MEM_CH13	Регистр индекса 1	182F_058C
OR_MEM_CH13	Регистр смещений	182F_0590
Y_MEM_CH13	Регистр параметров направления Y при двухмерной адресации	182F_0594
RUN_MEM_CH13	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH13 На чтение: Регистр управления и состояния канала MEM_CH3 без сброса битов "END" и "DONE"	182F_0598

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры DMA MEM_CH1</u>		
CSR_MEM_CH14	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0600
CP_MEM_CH14	Регистр указателя цепочки	182F_0604
IR0_MEM_CH14	Регистр индекса 0	182F_0608
IR1_MEM_CH14	Регистр индекса 1	182F_060C
OR_MEM_CH14	Регистр смещений	182F_0610
Y_MEM_CH14	Регистр параметров направления Y при двухмерной адресации	182F_0614
RUN_MEM_CH14	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH10 На чтение: Регистр управления и состояния канала MEM_CH0 без сброса битов "END" и "DONE"	182F_0618
CSR_MEM_CH15	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0680
CP_MEM_CH15	Регистр указателя цепочки	182F_0684
IR0_MEM_CH15	Регистр индекса 0	182F_0688
IR1_MEM_CH15	Регистр индекса 1	182F_068C
OR_MEM_CH15	Регистр смещений	182F_0690
Y_MEM_CH15	Регистр параметров направления Y при двухмерной адресации канала	182F_0694
RUN_MEM_CH15	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH11 На чтение: Регистр управления и состояния канала MEM_CH1 без сброса битов "END" и "DONE"	182F_0698
CSR_MEM_CH16	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0700
CP_MEM_CH16	Регистр указателя цепочки	182F_0704
IR0_MEM_CH16	Регистр индекса 0	182F_0708
IR1_MEM_CH16	Регистр индекса 1	182F_070C
OR_MEM_CH16	Регистр смещений канала MEM_CH2	182F_0710
Y_MEM_CH16	Регистр параметров направления Y при двухмерной адресации канала MEM_CH2	182F_0714
RUN_MEM_CH16	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH12 На чтение: Регистр управления и состояния канала MEM_CH2 без сброса битов "END" и "DONE"	182F_0718
CSR_MEM_CH17	Регистр управления и состояния (по чтению сброс битов "END" и "DONE")	182F_0780
CP_MEM_CH17	Регистр указателя цепочки	182F_0784
IR0_MEM_CH17	Регистр индекса 0	182F_0788
IR1_MEM_CH17	Регистр индекса 1	182F_078C
OR_MEM_CH17	Регистр смещений	182F_0790
Y_MEM_CH17	Регистр параметров направления Y при двухмерной адресации	182F_0794
RUN_MEM_CH17	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH13 На чтение: Регистр управления и состояния канала MEM_CH3 без сброса битов "END" и "DONE"	182F_0798

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры DMA VPIN_CH</u>		
CSR_VPIN_CH	Регистр управления и состояния. При чтении: сброс битов END и DONE	182F_6000
CP_VPIN_CH	Регистр указателя цепочки	182F_6004
IR_VPIN_CH	Регистр индекса	182F_6008
RUN_VPIN_CH	При записи: псевдорегистр управления состоянием бита RUN регистра CSR. При чтении: регистр управления и состояния CSR без сброса битов “END” и ”DONE”	182F_600C
<u>Регистры DMA VPOUT_CH</u>		
CSR_VPOUT_CH	Регистр управления и состояния. При чтении: сброс битов END и DONE	182F_7000
CP_VPOUT_CH	Регистр указателя цепочки	182F_7004
IR_VPOUT_CH	Регистр индекса	182F_7008
RUN_VPOUT_CH	При записи: псевдорегистр управления состоянием бита RUN регистра CSR. При чтении: регистр управления и состояния CSR без сброса битов “END” и ”DONE”	182F_700C
<u>Регистры DMA EMAC_RX_CH</u>		
CSR_EMAC_RX	Регистр управления и состояния. При чтении: сброс битов END и DONE	182F_E800
CP_EMAC_RX	Регистр указателя цепочки	182F_E804
IR_EMAC_RX	Регистр индекса	182F_E808
RUN_EMAC_RX	При записи: псевдорегистр управления состоянием бита RUN регистра CSR. При чтении: регистр управления и состояния CSR без сброса битов “END” и ”DONE”	182F_E80C
<u>Регистры DMA EMAC_TX_CH</u>		
CSR_EMAC_TX	Регистр управления и состояния. При чтении: сброс битов END и DONE	182F_E840
CP_EMAC_TX	Регистр указателя цепочки	182F_E844
IR_EMAC_TX	Регистр индекса	182F_E848
RUN_EMAC_TX	При записи: псевдорегистр управления состоянием бита RUN регистра CSR. При чтении: регистр управления и состояния CSR без сброса битов “END” и ”DONE”	182F_E84C

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры DMA MFBSP_TX_CH0</u>		
CSR_MFBSP_TX_CH0	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_8800
CP_MFBSP_TX_CH0	Регистр указателя цепочки	182F_8804
IR_MFBSP_TX_CH0	Регистр индекса	182F_8808
RUN_MFBSP_TX_CH0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_880C
<u>Регистры DMA MFBSP_RX_CH0</u>		
CSR_MFBSP_RX_CH0	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_8840
CP_MFBSP_RX_CH0	Регистр указателя цепочки	182F_8844
IR_MFBSP_RX_CH0	Регистр индекса	182F_8848
RUN_MFBSP_RX_CH0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_884C
<u>Регистры DMA MFBSP_TX_CH1</u>		
CSR_MFBSP_TX_CH1	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_9800
CP_MFBSP_TX_CH1	Регистр указателя цепочки	182F_9804
IR_MFBSP_TX_CH1	Регистр индекса	182F_9808
RUN_MFBSP_TX_CH1	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_980C
<u>Регистры DMA MFBSP_RX_CH1</u>		
CSR_MFBSP_RX_CH1	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_9840
CP_MFBSP_RX_CH1	Регистр указателя цепочки	182F_9844
IR_MFBSP_RX_CH1	Регистр индекса	182F_9848
RUN_MFBSP_RX_CH1	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_984C

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<b>Регистры DMA SWIC0</b>		
Канал записи в память дескрипторов принимаемых пакетов		
CSR_SWIC_RX_DES0	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_C800
CP_SWIC_RX_DES0	Регистр указателя цепочки	182F_C804
IR_SWIC_RX_DES0	Регистр индекса	182F_C808
RUN_SWIC_RX_DES0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_C80C
Канал записи в память принимаемых слов данных		
CSR_SWIC_RX_DAT0	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_C840
CP_SWIC_RX_DAT0	Регистр указателя цепочки	182F_C844
IR_SWIC_RX_DAT0	Регистр индекса	182F_C848
RUN_SWIC_RX_DAT0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_C84C
Канал чтения из памяти дескрипторов передаваемых пакетов		
CSR_SWIC_TX_DES0	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_C880
CP_SWIC_TX_DES0	Регистр указателя цепочки	182F_C884
IR_SWIC_TX_DES0	Регистр индекса	182F_C888
RUN_SWIC_TX_DES0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_C88C
Канал чтения из памяти передаваемых слов данных		
CSR_SWIC_TX_DAT0	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_C8C0
CP_SWIC_TX_DAT0	Регистр указателя цепочки	182F_C8C4
IR_SWIC_TX_DAT0	Регистр индекса	182F_C8C8
RUN_SWIC_TX_DAT0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_C8CC



Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры DMA SWIC1</u>		
<u>Канал записи в память дескрипторов принимаемых пакетов</u>		
CSR_SWIC_RX_DES1	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_D800
CP_SWIC_RX_DES1	Регистр указателя цепочки	182F_D804
IR_SWIC_RX_DES1	Регистр индекса	182F_D808
RUN_SWIC_RX_DES1	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_D80C
<u>Канал записи в память принимаемых слов данных</u>		
CSR_SWIC_RX_DAT1	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_D840
CP_SWIC_RX_DAT1	Регистр указателя цепочки	182F_D844
IR_SWIC_RX_DAT1	Регистр индекса	182F_D848
RUN_SWIC_RX_DAT1	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_D84C
<u>Канал чтения из памяти дескрипторов передаваемых пакетов</u>		
CSR_SWIC_TX_DES1	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_D880
CP_SWIC_TX_DES1	Регистр указателя цепочки	182F_D884
IR_SWIC_TX_DES1	Регистр индекса	182F_D888
RUN_SWIC_TX_DES1	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_D88C
<u>Канал чтения из памяти передаваемых слов данных</u>		
CSR_SWIC_TX_DAT1	Регистр управления и состояния (по чтению сброс битов “END” и ”DONE”)	182F_D8C0
CP_SWIC_TX_DAT1	Регистр указателя цепочки	182F_D8C4
IR_SWIC_TX_DAT1	Регистр индекса	182F_D8C8
RUN_SWIC_TX_DAT1	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_ На чтение: Регистр управления и состояния без сброса битов “END” и ”DONE”	182F_D8CC



Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры контроллера SWIC0</u>		
HW_VER0	Регистр аппаратной версии контроллера	182F_C000
STATUS0	Регистр состояния	182F_C004
RX_CODE0	Регистр принятого управляющего символа	182F_C008
MODE_CR0	Регистр управления режимом работы	182F_C00C
TX_SPEED0	Регистр управления скоростью передачи	182F_C010
TX_CODE0	Регистр передаваемого управляющего символа	182F_C014
RX_SPEED0	Регистр измерителя скорости приема	182F_C018
CNT_RX0_PACK0	Регистр счетчика принятых пакетов нулевой длины	182F_C01C
CNT_RX_PACK0	Регистр счетчика принятых пакетов ненулевой длины	182F_C020
ISR_L0	Регистр кодов распределенных прерываний (младшая часть)	182F_C024
ISR_H0	Регистр кодов распределенных прерываний (старшая часть)	182F_C028
TRUE_TIME0	Регистр достоверного маркера времени	182F_C02C
TOUT_CODE0	Регистр размера таймаутов	182F_C030
ISR_tout_L0	Младшие разряды регистра флагов таймаутов ISR	182F_C034
ISR_tout_H0	Старшие разряды регистра флагов таймаутов ISR	182F_C038
LOG_ADDR0	Регистр логического адреса	182F_C03C
<u>Регистры контроллера SWIC1</u>		
HW_VER1	Регистр аппаратной версии контроллера	182F_D000
STATUS1	Регистр состояния	182F_D004
RX_CODE1	Регистр принятого управляющего символа	182F_D008
MODE_CR1	Регистр управления режимом работы	182F_D00C
TX_SPEED1	Регистр управления скоростью передачи	182F_D010
TX_CODE1	Регистр передаваемого управляющего символа	182F_D014
RX_SPEED1	Регистр измерителя скорости приема	182F_D018
CNT_RX0_PACK1	Регистр счетчика принятых пакетов нулевой длины	182F_D01C
CNT_RX_PACK1	Регистр счетчика принятых пакетов ненулевой длины	182F_D020
ISR_L1	Регистр кодов распределенных прерываний (младшая часть)	182F_D024
ISR_H1	Регистр кодов распределенных прерываний (старшая часть)	182F_D028
TRUE_TIME1	Регистр достоверного маркера времени	182F_D02C
TOUT_CODE1	Регистр размера таймаутов	182F_D030
ISR_tout_L1	Младшие разряды регистра флагов таймаутов ISR	182F_D034
ISR_tout_H1	Старшие разряды регистра флагов таймаутов ISR	182F_D038
LOG_ADDR1	Регистр логического адреса	182F_D03C

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры VPIN</u>		
CSR	Регистр управления и состояния	182F_6000
Line_cnt/Pix_cnt	Счетчик строк / счетчик пикселей	182F_6004
Frame_cnt	Счетчик кадров	182F_6008
FIFO_OUT	Выход FIFO	182F_600C
<u>Регистры VPOUT</u>		
CSR	Регистр управления и состояния	182F_7000
DIV	Регистр периода сигнала VCLKO_out	182F_7004
Hstart/Hend	Регистр начала/конца активной части строки	182F_7008
Vstart/Vend	Регистр начала/конца активной части кадра	182F_700C
Line_cnt/Pix_cnt	Счетчик строк / счетчик пикселей	182F_7010
Frame_cnt (R) FIFO_IN (W)	Счетчик кадров Адрес записи в FIFO по команде Store	182F_7014
EHstart/EHend	Регистр начала/конца сигнала VDEN в строке	182F_7018
EVstart/EVend	Регистр начала/конца сигнала VDEN в строке в кадре	182F_701C
<u>Регистры I2C</u>		
PRER[15:0]	Регистр предделителя частоты	182F_3800
CTR[8:0]	Регистр управления	182F_3804
TXR[7:0]	Регистр передачи данных	182F_3808
RXR[7:0]	Регистр приема данных	182F_380C
CR[7:0]	Регистр команд	182F_3810
SR[7:0]	Регистр состояния	182F_3814
PR_CNT[15:0]	Счетчик предделителя частоты	182F_3818
SYNC[16:0]	Регистр синхронизации	182F_381C
<u>Регистры DDR_PORT0</u>		
DDR_CON0	Регистр конфигурации DDRAM	182F_1200
DDR_BAR0	Регистр базового адреса	182F_1204
DDR_CSR0	Регистр управления и состояния	182F_1208
DDR_TMR0	Регистр параметров DDRAM	182F_120C
DDR_MOD0	Регистр режимов	182F_1210
<u>Регистры DDR_PORT1</u>		
DDR_CON1	Регистр конфигурации DDRAM	182F_1300
DDR_BAR1	Регистр базового адреса	182F_1304
DDR_CSR1	Регистр управления и состояния	182F_1308
DDR_TMR1	Регистр параметров DDRAM	182F_130C
DDR_MOD1	Регистр режимов	182F_1310

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры Ethernet MAC</u>		
MAC_CONTROL[11:0]	Регистр управления MAC	182F_E000
MAC_ADDR_L[31:0]	Регистр младшей части исходного адреса MAC	182F_E004
MAC_ADDR_H[15:0]	Регистр старшей части исходного адреса MAC	182F_E008
DADDR_L[31:0]	Регистр младшей части адреса назначения	182F_E00C
DADDR_H[15:0]	Регистр старшей части адреса назначения	182F_E010
FCS_CLIENT[31:0]	Регистр контрольной суммы кадра	182F_E014
TYPE[15:0]	Регистр типа кадра	182F_E018
IFS_COLL_MODE[23:0]	Регистр IFS и режима обработки коллизии	182F_E01C
TX_FRAME_CONTROL[16:0]	Регистр управления передачи кадра	182F_E020
STATUS_TX[26:0]	Регистр статуса передачи кадра	182F_E024
UCADDR_L[31:0]	Регистр младшей части уникального адреса MAC	182F_E028
UCADDR_H[15:0]	Регистр старшей части уникального адреса MAC	182F_E02C
MCADDR_L[31:0]	Регистр младшей части группового адреса	182F_E030
MCADDR_H[15:0]	Регистр старшей части группового адреса	182F_E034
MCADDR_MASK_L[31:0]	Регистр младшей части маски группового адреса	182F_E038
MCADDR_MASK_H[15:0]	Регистр старшей части маски группового адреса	182F_E03C
HASHT_L[31:0]	Регистр младшей части хэш-таблицы	182F_E040
HASHT_H[31:0]	Регистр старшей части хэш-таблицы	182F_E044
RX_FRAME_CONTROL[9:0]	Регистр управления приема кадра	182F_E048
RX_FR_MaxSize[11:0]	Регистр максимального размера принимаемого кадра	182F_E04C
STATUS_RX[29:0]	Регистр статуса приема кадра	182F_E050
RX_FRAME_STATUS_FIFO [26:0]	FIFO статусов принятых кадров	182F_E054
MD_CONTROL[31:0]	Регистр управления порта MD	182F_E058
MD_STATUS[31:0]	Регистр статуса порта MD	182F_E05C
MD_MODE[8:0]	Регистр режима работы порта MD	182F_E060
TX_TEST_CSR[14:0]	Регистр управления и состояния режима тестирования TX_FIFO	182F_E064
TX_FIFO[31:0]	Передающее TX_FIFO	182F_E068
RX_TEST_CSR[14:0]	Регистр управления и состояния режима тестирования RX_FIFO	182F_E06C
RX_FIFO[31:0]	Принимающее RX_FIFO	182F_E070
<u>Регистры SRIO0</u>		
-	Базовый адрес. Перечень регистров контроллера с указанием адреса относительно базового приведен в разделе 11	182F_A000 – 182F_AFFC
<u>Регистры SRIO1</u>		
-	Базовый адрес. Перечень регистров контроллера с указанием адреса относительно базового приведен в разделе 11	182F_B000 – 182F_BFFC

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры MFBSP0</u>		
TX_MFBSP0	Буфер передачи данных	182F_8000
RX_MFBSP0	Буфер приёма данных	182F_8000
CSR_MFBSP0	Регистр управления и состояния	182F_8004
DIR_MFBSP0	Регистр управления направлением выводов порта ввода-вывода	182F_8008
GPIO_DR0	Регистр данных порта ввода-вывода	182F_800C
TCTR0	Регистр управления передатчиком	182F_8010
RCTR0	Регистр управления приёмником	182F_8014
TSR0	Регистр состояния передатчика	182F_8018
RSR0	Регистр состояния приёмника	182F_801C
TCTR_RATE0	Регистр управления темпом передачи данных	182F_8020
RCTR_RATE0	Регистр управления темпом приёма данных	182F_8024
TSTART0	псевдорегистр ten – запуск/останов передатчика без изменения настроек передатчика	182F_8028
RSTART0	псевдорегистр gen – запуск/останов приемника без изменения настроек приемника	182F_802C
EMERG_MFBSP0	Регистр аварийного управления портом	182F_8030
IMASK_MFBSP0	Регистр маски прерываний от порта	182F_8034
<u>Регистры MFBSP1</u>		
TX_MFBSP1	Буфер передачи данных	182F_9000
RX_MFBSP1	Буфер приёма данных	182F_9000
CSR_MFBSP1	Регистр управления и состояния	182F_9004
DIR_MFBSP1	Регистр управления направлением выводов порта ввода-вывода	182F_9008
GPIO_DR1	Регистр данных порта ввода-вывода	182F_900C
TCTR1	Регистр управления передатчиком	182F_9010
RCTR1	Регистр управления приёмником	182F_9014
TSR1	Регистр состояния передатчика	182F_9018
RSR1	Регистр состояния приёмника	182F_901C
TCTR_RATE1	Регистр управления темпом передачи данных	182F_9020
RCTR_RATE1	Регистр управления темпом приёма данных	182F_9024
TSTART1	псевдорегистр ten – запуск/останов передатчика без изменения настроек передатчика	182F_9028
RSTART1	псевдорегистр gen – запуск/останов приемника без изменения настроек приемника	182F_902C
EMERG_MFBSP1	Регистр аварийного управления портом	182F_9030
IMASK_MFBSP1	Регистр маски прерываний от порта	182F_9034

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры PMSC</u>		
Device ID/ Vendor ID	Регистр идентификации устройства	182F_F000
Status/Command	Регистр состояния и управления	182F_F004
Class Code/Revision ID	Регистр кода классификации	182F_F008
Latency Timer	Регистр времени транзакции в режиме Master	182F_F00C
BAR0 (Base Address Register 0)	Регистр базового адреса 0	182F_F010
BAR1 (Base Address Register 1)	Регистр базового адреса 1	182F_F014
Subsystem ID/ Subsystem Ven- dor ID	Регистр идентификации подсистемы	182F_F02C
Interrupt Line	Код прерывания	182F_F03C
IR_TARGET	Регистр адреса памяти в режиме Target	182F_F040
SEM	Регистр семафора	182F_F044
MBR_PCI	Регистр почтового ящика шины PCI	182F_F048
CSR_PCI	Регистр управления и состояния шины PCI	182F_F04C
CSR_MASTER	Регистр управления режимом Master	182F_F050
IR_MASTER	Регистр адреса памяти в режиме Master	182F_F054
AR_PCI	Регистр адреса шины PCI в режиме Master	182F_F058
QSTR_PCI	Регистр системных прерываний.	182F_F05C
MASKR_PCI	Регистр маскирования прерываний	182F_F060
STATUS_ MASTER	Регистр состояния обмена с шиной PCI в режиме Master	182F_F064
TMR_PCI	Регистр параметров	182F_F068
CSR_WIN	Регистр управления обменом с PCI через адресное окно	182F_F06C
<u>Регистры UART</u>		
RBR	Приемный буферный регистр	182F_3000
THR	Передающий буферный регистр	182F_3000
IER	Регистр разрешения прерываний	182F_3004
IIR	Регистр идентификации прерывания	182F_3008
FCR	Регистр управления FIFO	182F_3008
LCR	Регистр управления линией	182F_300C
MCR	Регистр управления модемом	182F_3010
LSR	Регистр состояния линии	182F_3014
SPR	Регистр Scratch Pad	182F_301C
DLL	Регистр делителя младший	182F_3000
DLM	Регистр делителя старший	182F_3004
SCLR	Регистр предделителя (scaler)	182F_3014

Таблица 2.50. Продолжение

Условное обозначение регистра	Название регистра	Адрес регистра
<u>Регистры IT0</u>		
ITCSR	Регистр управления	182F_5000
ITPERIOD	Регистр периода работы таймера	182F_5004
ITCOUNT	Регистр счетчика	182F_5008
ITSCALE	Регистр предделителя	182F_500C
<u>Регистры WDT</u>		
WTCSR	Регистр управления	182F_5010
WTPERIOD	Регистр периода работы таймера	182F_5014
WTCOUNT	Регистр счетчика	182F_5018
WTSCALE	Регистр предделителя	182F_501C
<u>Регистры IT1</u>		
ITCSR	Регистр управления	182F_5020
ITPERIOD	Регистр периода работы таймера	182F_5024
ITCOUNT	Регистр счетчика	182F_5028
ITSCALE	Регистр предделителя	182F_502C
<u>Регистры MPORT</u>		
CSCON0	Регистр конфигурации блока внешней памяти, подключаемого к выводу nCS[0]	182F_1000
CSCON1	Регистр конфигурации блока внешней памяти, подключаемого к выводу nCS[1]	182F_1004
CSCON2	Регистр конфигурации блока внешней памяти, подключаемого к выводу nCS[2]	182F_1008
CSCON3	Регистр конфигурации блока внешней памяти, подключаемого к выводу nCS[3]	182F_100C
CSCON4	Регистр конфигурации внешней памяти, не вошедшей в блоки памяти, определяемые регистрами CSCON3 - CSCON0	182F_1010
SDRCON	Регистр конфигурации типа SDRAM	182F_1014
SDRTMR	Регистр временных параметров памяти типа SDRAM	182F_1018
SDRCSR	Регистр управления режимами памяти типа SDRAM	182F_101C
FLY_WS	Регистр определяет количество дополнительных тактов ожидания в обменах внешних устройств с асинхронной памятью (режим FLYBY)	182F_1020
<u>Системные регистры</u>		
CR_PLL	Регистр управления PLL	182F_4000
CLK_EN	Регистр управления отключением частоты от устройств	182F_4004
CSR	Регистр управления и состояния	182F_4008
MASKR0	Регистр маски прерываний из регистра QSTR0	182F-4010
QSTR0	Регистр запросов прерываний от IT0, IT1, WDT, I2C, Vpin, Vpout, ENET, UART, nIRQ[3:0]	182F-4014
MASKR1	Регистр маски прерываний из регистра QSTR1	182F-4018
QSTR1	Регистр запросов прерываний от каналов DMA MEM_CH	182F-401C
MASKR2	Регистр маски прерываний из регистра QSTR2	182F-4020
QSTR2	Регистр запросов прерываний от SRIO0, SRIO1	182F-4024
MASKR3	Регистр маски прерываний из регистра QSTR3	182F-4028
QSTR3	Регистр запросов прерываний от SWIC0, SWIC1, MFBSP0, MFBSP1	182F-402C
IRQM	Регистр управления режимом приема внешних прерываний nIRQ[3:0]	182F-4030

### 3. ЦИФРОВОЙ СИГНАЛЬНЫЙ ПРОЦЕССОР

В состав микросхемы 1892ВМ7Я входит сигнальный сопроцессор-акселератор QELcore-28, представляющий собой кластер (симметричный мультипроцессор) из 4-х DSP-ядер ELcore-28 (DSP0 – DSP3), работающих на общем поле памяти данных, содержащий набор общих для всего кластера регистров управления и состояния, а также буфер обмена XBUF.

#### 3.1 Основные технические характеристики DSP-кластера QELcore-28

- 4 вычислительных ядра DSP ELcore-28;
- объем общей памяти данных 512 Кбайт (128 Кбайт на ядро);
- объем памяти программ 32 Кбайт на ядро;
- максимальная пропускная способность коммутатора ядер с памятью – 1024 бит за такт;
- максимальная скорость обмена внешних устройств с памятью кластера – 256 бит за такт;
- пиковая производительность:
  - 32 32-разрядных операций с плавающей точкой (IEEE 754) за такт;
  - 32 32-разрядных операций с фиксированной точкой за такт;
  - 128 16-разрядных операций с фиксированной точкой за такт;
  - 192 8-разрядных операций с фиксированной точкой за такт.

#### 3.2 Структурная схема

Структурная схема 4-ядерного DSP-кластера QELcore-28 приведена на Рисунок 3.1.

На схеме приняты следующие обозначения:

DSP0 – DSP3 – четыре DSP-ядра ELcore-28;

PMEM – память программ;

XMEM – память данных;

CDB – контроллер шины CDB (slave);

MEM\_EXT\_PORT, MEM\_MUX\_OUT – распределенный контроллер портов AXI SWITCH (slave);

XBUF\_04 – буфер обмена (регистровый файл 32 слова по 64 разряда, 6 портов);

ArbBuf, MA\_LocalArb – распределенный арбитр;

DSP\_logic – вычислительное ядро;

AGU, AGU-Y – адресные генераторы памяти данных;

PAG – адресный генератор памяти программ;

PDC\_17 – программный декодер;

RF9 – регистровый файл 32 слова по 128 разрядов, 9 портов;

COMM5 – коммутатор входных данных операционных устройств;

OP1\_unit, OP2\_unit – операционные (вычислительные) устройства;

CCR\_REG, PDN – регистры признаков результата операции и параметра денормализации;

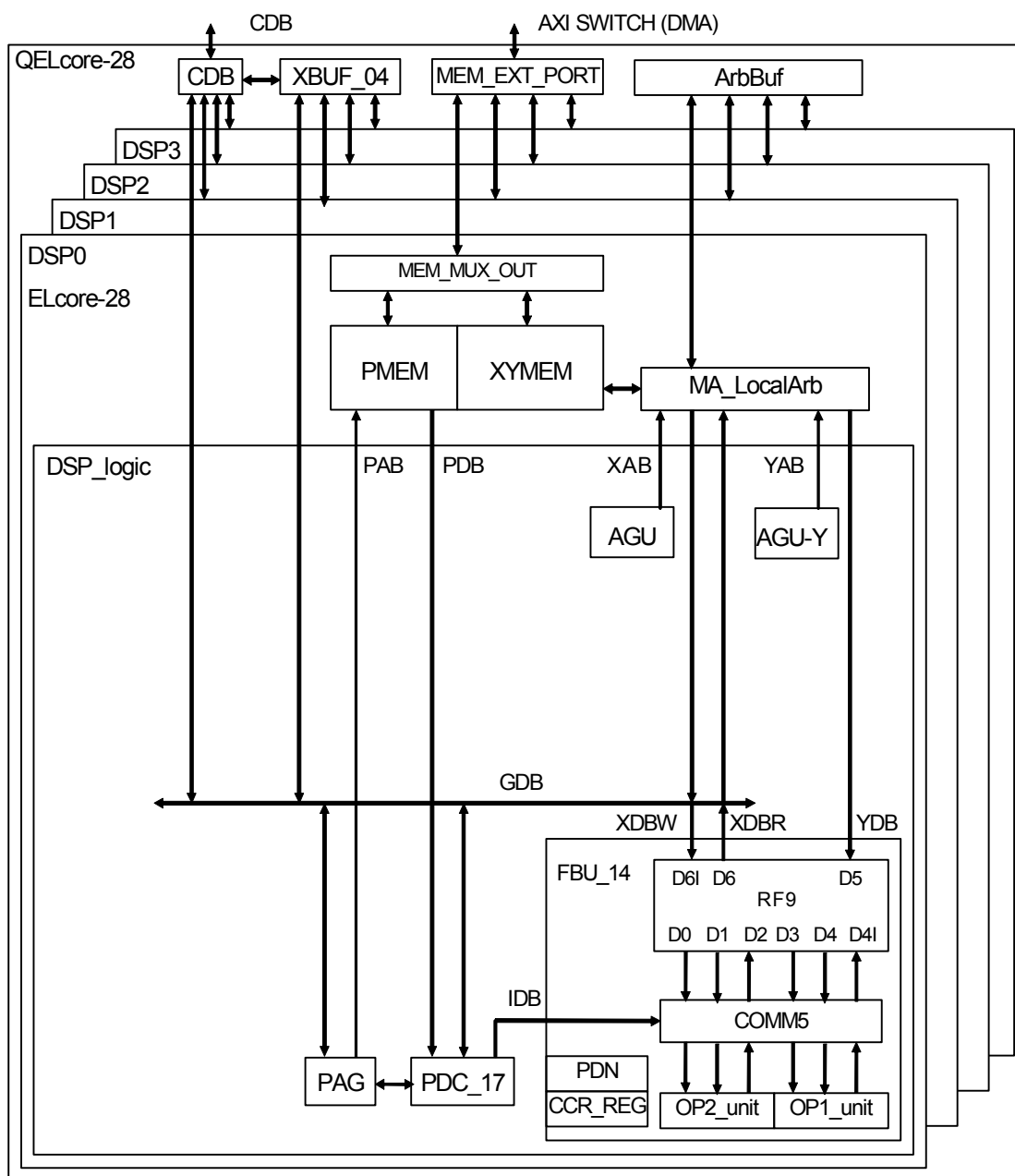


Рисунок 3.1. Структурная схема 4-ядерного DSP-кластера QELcore-28

### 3.2.1 Интерфейсы DSP-кластера QELcore-28

Управление кластером DSP осуществляется CPU. Внешний доступ ко всем регистрам DSP ядер, регистрам обменного буфера XBUF, а так же контрольным регистрам общим для всех ядер DSP кластера осуществляется по шине CDB.

Доступ к программной памяти и памяти данных осуществляется через порты AXI SWITCH. При этом в кластере DSP предусмотрено 4 независимых порта с AXI SWITCH, каждый из которых позволяет передавать по 64 бита за такт. По каждому порту производится доступ к памяти определенного ядра. Такая организация позволяет одновременно производить несколько DMA обменов с памятью DSP кластера. При



этом каждое DSP ядро может запустить DMA обмен, используя один из восьми доступных контроллеров DMA, а также получить прерывание от контроллера DMA, закончившего обмен. Для этих целей в интерфейсе кластера предусмотрены четыре пары векторных выводов, по которым передается информация, о том какой контроллер DMA должен быть запущен, и от какого именно контроллера поступило прерывание для конкретного DSP ядра.

Для каждого из DSP ядер кластера предусмотрен собственный тактовый сигнал (сигнал синхронизации), поэтому кроме системного такового сигнала шины CDB и портов AXI SWITCH, в кластер заводятся 4 тактовых сигнала для каждого из 4-х вычислительных ядер. Это сделано для обеспечения возможности независимого отключения тактовой частоты от каждого из DSP ядер с целью снижения энергопотребления.

### 3.2.2 Организация работы DSP-кластера QELcore-28

Кластер DSP представляет собой четырехядерную MIMD систему. Каждое DSP ядро обладает собственной программной памятью, и может работать независимо от остальных ядер.

Для синхронизации работы DSP ядер в кластере предусмотрено два механизма: механизм прерываний и механизм обменов через XBUF в синхронном режиме.

Каждое DSP ядро может сформировать прерывание для любого другого ядра в кластере. Ядро, получившее прерывание, переходит в состояние RUN, если было остановлено, и начинает исполнение подпрограммы, адрес которой храниться в специальном регистре этого ядра.

Для оперативных обменов данными между CPU, DSP0 – DSP3 в составе 1892BM7Я имеется буфер обмена XBUF, состоящий из 32-х 64-разрядных регистров X0-X31, доступных по записи и чтению для всех процессорных ядер.

Буфер обмена XBUF представляет собой многопортовую память и допускает одновременное чтение одной и той же ячейки со стороны нескольких абонентов - CPU, DSP0 – DSP3. При одновременном запросе на запись в одну и ту же ячейку приоритет отдается CPU, затем - DSP0, затем - DSP1, затем - DSP2, затем - DSP3.

Обменный буфер может работать в обычном режиме, когда при обмене данными через него не происходит никаких блокировок и в синхронном режиме. В синхронном режиме для конкретного регистра XBUF обязательно должны чередоваться операции чтения записи, если какое либо ядро пытается осуществить запись после записи или чтение после чтения – оно блокируется. Обмен через XBUF в синхронном режиме является дополнительным программным способом синхронизации ядер DSP.

Программная память и память данных кластера DSP физически организована как двух-портовая. По одному порту производятся внешние обращения от RISC ядра и контроллеров DMA, по другому порту производятся обращения от ядер DSP. Такая организация позволяет производить бесконфликтный фоновый обмен данными между памятью кластера DSP и внешними устройствами. Более подробно организация памяти в кластере описана в следующем параграфе.

### 3.3 Организация общего поля памяти данных

Кластер DSP организован как система с ассиметричным доступом к памяти (NUMA). Общее адресное пространство кластера состоит из локальных памяти XYRAM0-XYRAM3 каждого из DSP ядер. Таким образом, вся память разбита на 4 сегмента, при этом для каждого DSP ядра есть ближний (свой) сегмент памяти, обращения к которому в случае, если нет конфликтов с другими ядрами, не приводят к простоям ядра. Остальные же сегменты для него являются дальними (чужими) и обращения к ним могут приводить к простоям ядра даже в отсутствии конфликтов между ядрами. Обращения к чужим сегментам памяти проходят через очередь обращений (для 1892BM7Я глубина очереди обращений к дальним сегментам равняется 2).

Операция записи является буферизованной, т.е. в отсутствии конфликтов между ядрами запись в дальний сегмент памяти не приводит к простоям ядра. Однако программисту следует учитывать, что физически запись в память происходит не сразу после исполнения инструкции, а через время, требуемое для прохождения данных по очереди обращений и на разрешение конфликтов (в отсутствии конфликтов запись корректных данных в дальнюю память осуществляется через 2 такта после исполнения инструкции записи в память). При возникновении конфликтов при обращениях к памяти простой ядер возможен даже при выполнении записи.

В данной реализации кластера DSP операция чтения не является буферизованной, поэтому при чтении из дальнего сегмента памяти ядро останавливается на 4 такта (при возникновении конфликтных ситуаций к этому времени добавляется время, требуемое для разрешения конфликтов).

### 3.3.1 Карта памяти

Карта памяти DSP0-DSP3 в составе 1892BM7Я приведена на Рисунок 3.2.

Адреса в пространстве CPU					Внутренние адреса DSP
DSP0	DSP1	DSP2	DSP3		
0x187F_FFFC 0x187F_FF00				Буфер обмена XBUF (32*64)	
				Резерв	
0x1848_027C 0x1848_0000	0x1888_027C 0x1888_0000	0x18C8_027C 0x18C8_0000	0x1908_027C 0x1908_0000	Регистры данных и управления	
				Резерв	
0x1844_7FFC 0x1844_0000	0x1884_7FFC 0x1884_0000	0x18C4_7FFC 0x18C4_0000	0x1904_7FFC 0x1904_0000	Память программ PRAM (4K*64)	0x0FFF = PC_max PC 0x0000 = PC_min
0x1901_FFFC 0x1900_0000				Память данных XYRAM сегмент 3 (16K*64)	0x1FFFF 0x18000
0x18C1_FFFC 0x18C0_0000				Память данных XYRAM сегмент 2 (16K*64)	0x17FFF 0x10000
0x1881_FFFC 0x1880_0000				Память данных XYRAM сегмент 1 (16K*64)	0x0FFFF 0x08000
0x1841_FFFC 0x1840_0000				Память данных XYRAM сегмент 0 (16K*64)	0x07FFF 0x00000

Рисунок 3.2. Карта памяти DSP0-DSP3 в составе 1892BM7Я

Каждое из DSP-ядер имеет свою программную память (PRAM) объемом 4К 64-разрядных слов (32 Кбайт) и общую для всех память данных XYRAM объемом 64К 64-разрядных слов (всего 512 Кбайт).

- Объем PRAM (DSP0) – 8К 32-разрядных слов (32 Кбайт).
- Объем PRAM (DSP1) – 8К 32-разрядных слов (32 Кбайт).
- Объем PRAM (DSP2) – 8К 32-разрядных слов (32 Кбайт).
- Объем PRAM (DSP3) – 8К 32-разрядных слов (32 Кбайт).
- Объем XYRAM – 128К 32-разрядных слов (512 Кбайт).

Для обеспечения возможности одновременного доступа к памяти программ и данных DSP как со стороны CPU (DMA), так и со стороны DSP блоки памяти XYRAM и PRAM аппаратно реализованы как 2-портовые. С внешней стороны возможны как 32-разрядные (CPU), так и 64-разрядные обращения (DMA); со стороны DSP0–DSP3 возможны 32/64/128-разрядные обращения.

Особенностью архитектуры процессора 1892BM7Я является то, что 4 входящих в его состав DSP-ядра (DSP0 – DSP3) работают на общем поле памяти данных. Для каждого DSP-ядра сегмент памяти с соответствующим номером (XYRAM0 – XYRAM3) является «ближней» памятью, доступ к которой осуществляется с наименьшей задержкой. Доступ к остальной («дальней») памяти производится с дополнительной задержкой, необходимой для выполнения арбитража.

При этом отсутствует разделение памяти данных на X-память и Y-память, имевшее место в предшествующих версиях DSP-ядер ELcore-xx. Указатели (адресные регистры) A0-A7, AT полностью равноправны, т.е. по указателям A0-A7, AT каждому из DSP-ядер доступна вся память данных XYRAM.

Начальное состояние регистров A0-A7, AT каждого из DSP-ядер приведено в Таблица 3.1.

**Таблица 3.1 Начальное состояние адресных регистров A0-A7, AT**

Условное обознач.	Разрядность	Наименование	Начальное состояние			
			DSP0	DSP1	DSP2	DSP3
A0-A7	32 R/W	Адресный регистр AGU	0x00000	0x08000	0x10000	0x18000
AT	32 R/W	Адресный регистр AGU-Y	0x04000	0x0C000	0x14000	0x1C000

Таким образом, при начальной установке регистры A0-A7 указывают на начало, а регистры AT – на середину ближней (локальной) памяти соответствующего DSP-ядра.

### **3.3.2 Дисциплина отработки одновременных обращений к общему полю памяти данных со стороны DSP-ядер (арбитраж).**

Так как память данных XYRAM является общим ресурсом для четырех DSP-ядер, при одновременном обращении к ней со стороны нескольких DSP-ядер возможны коллизии.

Для уменьшения числа таких коллизий память данных XYRAM разделена на 4 сегмента, каждый из которых содержит 4 страницы объемом 2К 128-разрядных слов. Таким образом, доступ к каждой из страниц может осуществляться независимо от других, и обращение различных DSP-ядер к различным страницам памяти может происходить одновременно и не приводит к коллизиям и задержкам.

Коллизии возникают лишь при одновременном обращении различных DSP-ядер к одной и той же странице, либо при одновременном обращении X-указателя (A0-A7) и Y-указателя (AT) одного из DSP к одной и той же странице памяти. Для разрешения возникающих коллизий вводится дополнительное устройство – арбитр памяти. Процедура арбитража позволяет корректно отработать все обращения, однако приводит к некоторому замедлению работы программы из-за введения дополнительных тактов ожидания обмена.

Подробнее дисциплина отработки одновременных обращений к одной и той же странице памяти данных со стороны нескольких DSP-ядер (арбитраж) рассматривается в п. 3.9.17.

### 3.4 Регистры управления и состояния QELcore-28

На верхнем уровне кластера DSP имеются 4 регистра управления и состояния. Назначение и адреса этих регистров указаны в Таблица 3.2.

Таблица 3.2 Назначение и адреса регистров управления и состояния кластера DSP

Имя	Разрядность	Тип обращений	Назначение	Адрес
MASKR_DSP	32	R/W	Регистр маски прерываний	0x1848_1000
QSTR_DSP	32	R	Регистр запросов прерываний	0x1848_1004
CSR_DSP	32	R/W	Регистр управления и состояния	0x1848_1008
TOTAL_RUN_CNTR	32	R/W	Счетчик тактов в состоянии RUN	0x1848_100C
TOTAL_CLK_CNTR	32	R/W	Счетчик тактов	0x1848_1010

#### 3.4.1 Регистр маски прерываний (MASKR\_DSP)

Регистр маски прерываний MASKR\_DSP содержит 32 разряда, каждый из которых разрешает («1») либо запрещает («0») запрос на прерывание в CPU от соответствующего разряда регистра запросов прерываний QSTR\_DSP. Регистр доступен по чтению и записи. Начальное состояние регистра MASKR\_DSP=0x0.

#### 3.4.2 Регистр запросов прерываний (QSTR\_DSP)

Регистр запросов прерываний QSTR\_DSP доступен только по чтению и содержит флаги запросов прерываний от 4-х DSP-ядер. Назначение разрядов регистра QSTR\_DSP приведено в Таблица 3.3.

Таблица 3.3 Назначение разрядов регистра QSTR\_DSP

Номер разряда	Наименование разряда	Назначение
0	PI0	Программное прерывание DSP0
1	SE0	Прерывание по ошибке стека DSP0
2	BREAK0	Прерывание по останову BREAK DSP0
3	STP0	Прерывание по останову STOP DSP0
4-7	-	Резерв
8	PI1	Программное прерывание DSP1
9	SE1	Прерывание по ошибке стека DSP1
10	BREAK1	Прерывание по останову BREAK DSP1
11	STP1	Прерывание по останову STOP DSP1
12-15	-	Резерв
16	PI2	Программное прерывание DSP2
17	SE2	Прерывание по ошибке стека DSP2
18	BREAK2	Прерывание по останову BREAK DSP2
19	STP2	Прерывание по останову STOP DSP2
20-23	-	Резерв
24	PI3	Программное прерывание DSP3
25	SE3	Прерывание по ошибке стека DSP3
26	BREAK3	Прерывание по останову BREAK DSP3
27	STP3	Прерывание по останову STOP DSP3
28	WAIT	Прерывание по состоянию ожидания DSP0 - DSP3
29-31	-	Резерв

Начальное состояние регистра QSTR\_DSP=0x0.

### 3.4.3 Регистр управления и состояния (CSR\_DSP)

Регистр управления и состояния CSR\_DSP доступен по чтению и записи и содержит биты управления кластером DSP-ядер. Назначение разрядов регистра CSR\_DSP приведено в Таблица 3.4.

Таблица 3.4 Назначение разрядов регистра CSR\_DSP

Номер разряда	Наименование разряда	Назначение
0	SYNSTART	Одновременный старт DSP0 – DSP1
1	SYNWORK	Работа XBUF в синхронном режиме
2-3	PMCONFIG	Конфигурация программной памяти
4-15	-	Резерв
16	HEN	Включение режима определения высокой плотности потоков
17	DEN	Разрешение установки явного приоритета (статический режим)
18	LEN	Бит разрешения ограничителя
21:20	DPTR	Номер ядра, обладающего наивысшим приоритетом
29:24	Limit	Максимальное значение счетчика обращений
25-31	-	Резерв

Начальное состояние регистра CSR\_DSP=0x0.

Запись «1» в разряд SYNSTART приводит к одновременному запуску четырёх DSP-ядер. При этом в регистрах DCSR каждого из DSP-ядер бит RUN устанавливается в «1», состояние других разрядов не изменяется.

Запись «1» в разряд SYNWORK устанавливает буфер обмена XBUF в синхронный режим.

Для управления арбитражем введены разряды HEN, DEN, LEN, DPTR, Limit.

HEN – Включение режима определения высокой плотности потоков. Используется в режиме захвата (LEN = 0). Если HEN = 1, то включаются счетчики, определяющие плотность обращений ядер к данному сегменту. Если плотность обращений хотя бы от одного ядра больше 75% – то при значениях HEN = 1 и LEN = 0 передача приоритета происходит каждый такт.

DEN – разрешение установки явного приоритета (статический режим). Если данный бит установлен в 1, то при возникновении конфликта приоритет отдается обращению от ядра, номер которого определяется битами DPTR.

DPTR – определяет номер ядра, обладающего наивысшим приоритетом при обращении к сегменту памяти данного DSP. DPTR = 0 задает высший приоритет для данного ядра, 1 – высший приоритет для соседа с меньшим номером, далее циклически в сторону уменьшения номера ядра.

LEN – бит разрешения ограничителя. Если данный бит установлен в 1, арбитр работает в режиме ограничения, если бит установлен в 0 арбитр работает в режиме захвата.

Limit – задает максимальное значение счетчика обращений, в режиме ограничения. В этом режиме предусмотрена автоматическая смена приоритета.



### 3.4.4 Счетчик тактов в состоянии RUN (*TOTAL\_RUN\_CNTR*)

32-разрядный счетчик тактов (*TOTAL\_RUN\_CNTR*) выполняет подсчет числа тактов, в течение которых хотя бы одно из DSP-ядер находилось в состоянии RUN. Любая запись в данный счетчик приводит к его обнулению.

Начальное состояние счетчика тактов также равно нулю:  $TOTAL\_RUN\_CNTR = 0x0$ .

### 3.4.5 Счетчик тактов (*TOTAL\_CLK\_CNTR*)

32-разрядный счетчик тактов (*TOTAL\_CLK\_CNTR*) выполняет подсчет числа тактов. Любая запись в данный счетчик приводит к его обнулению.

Начальное состояние счетчика тактов также равно нулю:  $TOTAL\_CLK\_CNTR = 0x0$ .

## 3.5 Буфер обмена XBUF

Для оперативных обменов данными между CPU, DSP0 – DSP3 в составе 1892BM7Я имеется буфер обмена XBUF, состоящий из 32-х 64-разрядных регистров X0-X31, доступных по записи и чтению для всех процессорных ядер.

Буфер обмена XBUF представляет собой многопортовую память и допускает одновременное чтение одной и той же ячейки со стороны нескольких абонентов - CPU, DSP0 – DSP3. При одновременном запросе на запись в одну и ту же ячейку приоритет отдается CPU, затем - DSP0, затем - DSP1, затем - DSP2, затем - DSP3.

Особенностью работы XBUF в составе 1892BM7Я является то, что обмены со стороны DSP0 – DSP3 – 64-разрядные, а со стороны CPU – 32-разрядные. Размещение 64-разрядных регистров X0-X31 в адресном пространстве CPU приведено в Таблица 3.18.

В ассемблере DSP-ядра ELcore-28 регистры XBUF (регистры обмена) составляют подмножество регистров управления. Для обозначения этих регистров в ассемблер DSP вводятся специальные мнемонические имена – X0, ..., X31. Для обращения к регистрам XBUF используются форматы команд 2t, 8d и 9d:

Формат 2t:	MOVE.cc Rn, Xi	;(запись в XBUF),
	MOVE.cc Xi, Rn	;(чтение из XBUF).
Форматы 8d, 9d:	<OP2> <OP1> Rn, Xi	;(запись в XBUF),
	<OP2> <OP1> Xi, Rn	;(чтение из XBUF).

### 3.5.1 Регистр флагов обмена (*EFR*)

Регистр флагов обмена (*EFR*) предназначен для отображения флагов обменов через буфер XBUF. Регистр *EFR* содержит 32 бита, доступных только по чтению каждому из DSP-ядер и CPU, начальное состояние  $EFR=0x0$ .

Каждый разряд этого регистра формируется аппаратно и отображает тип последней транзакции, выполненной с соответствующей ячейкой XBUF (0 – чтение из XBUF, 1 – запись). Заметим, что при 32-разрядных обращениях со стороны CPU *изменение состояния EFR происходит только при обращении к младшей половине 64-разрядной ячейки XBUF*.

### 3.5.2 Режимы обменов с XBUF

Имеются два режима обменов с XBUF – обычный и синхронный (семафорный).

*В обычном режиме* (устанавливается битом 1 регистра CSR\_DSP SYNWORK=0) любой из абонентов - CPU, DSP0/1/2/3 - в любое время может обращаться к любой ячейке XBUF, и это обращение немедленно исполняется (с учетом приоритета по записи).

*В синхронном режиме* (устанавливается битом 1 регистра CSR\_DSP SYNWORK=1):

- CPU обращается к XBUF так же, как и в обычном режиме;

- обращения со стороны DSP0 – DSP3 могут выполняться с задержкой в зависимости от состояния регистра EFR и типа обращения. Если тип обращения не совпадает с типом последней транзакции, выполненной с данной ячейкой XBUF (то есть если за записью следует чтение, а за чтением - запись) то исполнение такого обращения происходит без задержки. Если же за записью вновь следует запрос на запись в ту же ячейку (либо за чтением – вновь запрос на чтение), то такое обращение выполняется с задержкой. Выдавшееся запрос DSP переводится в состояние ожидания, продолжающееся до тех пор, пока соответствующий бит EFR не сменит свое значение на противоположное.

В регистре DCSR имеется бит WT=DCSR[4], указывающий на то, что DSP находится в состоянии ожидания при обращении к XBUF. Одновременная установка битов WT в состояние «1» во всех четырех DSP-ядрах (то есть зависание программы) вызывает прерывание WAIT в CPU (разряд 16 регистра QSTR\_DSP).

## 3.6 Программная модель DSP-ядра ELcore-28

Программная модель DSP-ядра включает в себя память (программ и данных) и программно-доступные регистры. Регистры обменного буфера XBUF и регистр флагов обмена EFR являются общими для всего DSP-кластера, остальные регистры принадлежат конкретному DSP-ядру и входят в состав одного из его исполнительных устройств. К исполнительным устройствам DSP-ядра относятся:

- вычислительная секция ALU;
- адресные генераторы для XY-памяти данных (AGU и AGU-Y);
- устройство программного управления PCU.

По своему назначению все регистры делятся на регистры данных, объединенные в регистровый файл (RF), и регистры управления (все остальные). Регистры управления разделены на четыре подмножества:

- регистры адресных генераторов AGU, AGU-Y;
- регистры обменного буфера XBUF;
- регистры устройства управления PCU;
- регистры-аккумуляторы (в составе ALU);

Программно-доступные регистры DSP-ядра (включая стеки и регистровый файл) приведены на Рисунок 3.3.

## 3.7 Регистры ALU

Каждая вычислительная секция ALU содержит регистровый файл RF – реконфигурируемый массив (16x128 или 32x64 или 32x32 или 32x16) регистров данных, регистр параметра денормализации PDNR, регистр кодов условий (регистр признаков) CCR, реконфигурируемый массив (8x64 или 16x32) регистров-аккумуляторов.



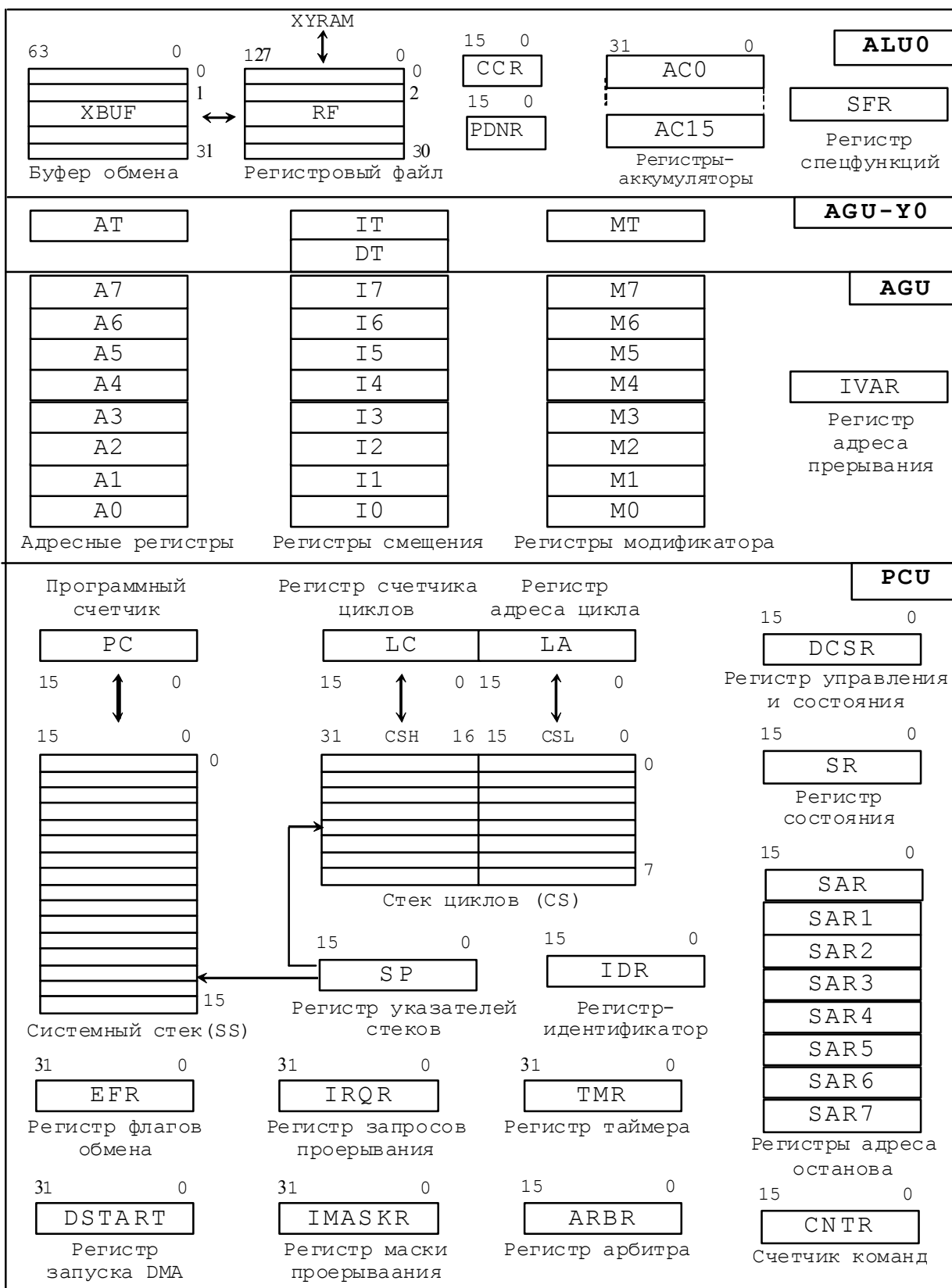


Рисунок 3.3. Программно-доступные регистры DSP-ядра Elcore-28

### 3.7.1 Регистровый файл

Исходные данные и результаты всех операций ALU хранятся в регистровом файле (RF), который представляет собой реконфигурируемый массив регистров данных (16 регистров по 128 разрядов; или 32 регистра по 64 разряда; или 32 регистра по 32 разряда; или 32 регистра по 16 разрядов). Структура регистрового файла приведена на Рисунок 3.4.

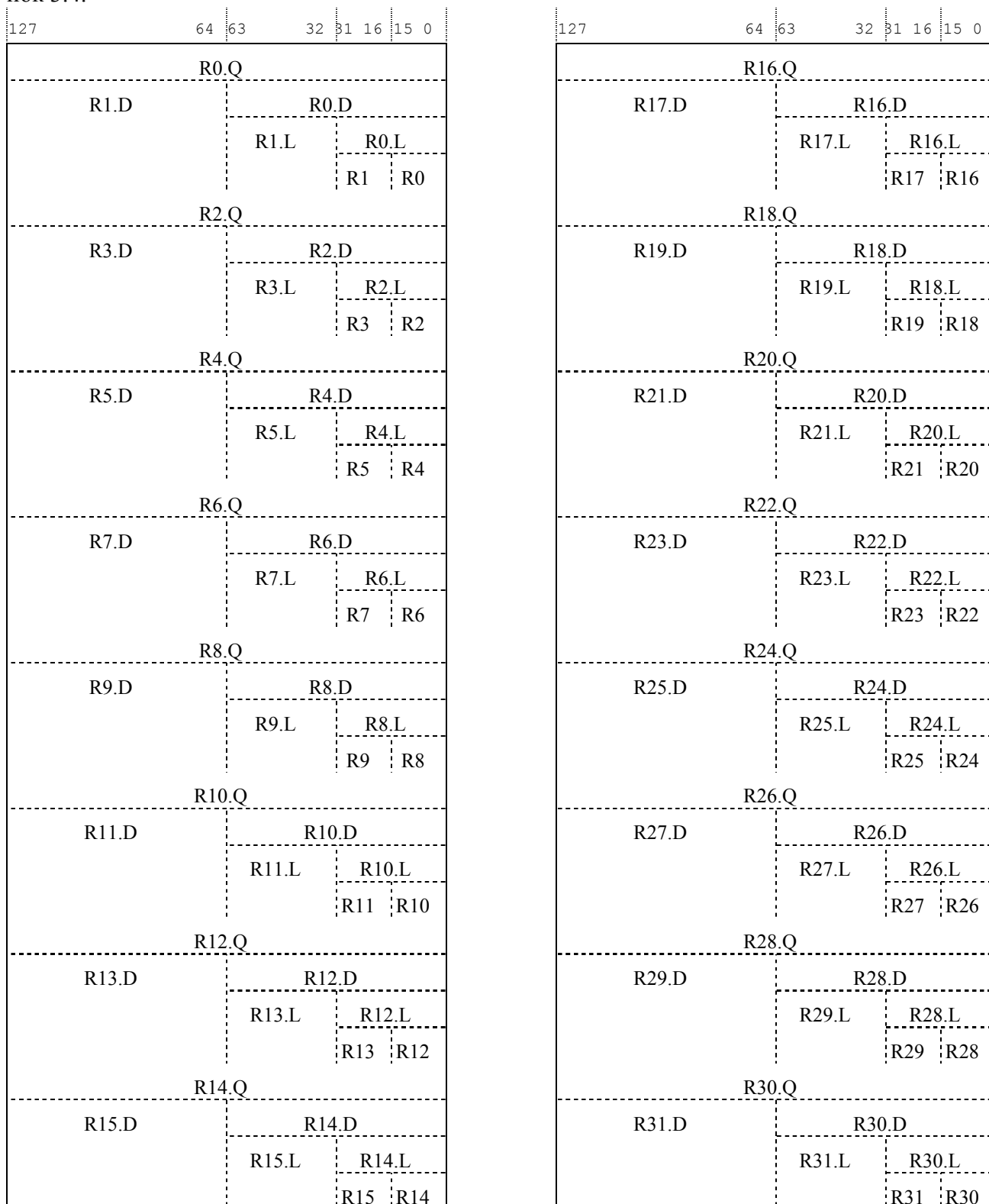


Рисунок 3.4. Структура регистрового файла

Для определения форматов регистров используются следующие мнемоники:

R – 16-разрядные регистры;

R.L – 32-разрядные регистры;

R.D – 64-разрядные регистры;

R.Q – 128-разрядные регистры.

16/32/64-разрядные регистры данных могут иметь номера с R0 по R31, а 128-разрядные регистры – только четные номера с R0 по R30. Четный и нечетный (с номером, большим на единицу) регистры одинаковой разрядности объединяются попарно и образуют 16 регистров большей разрядности с четными номерами, например, два 16-разрядных регистра R0 и R1 образуют 32-разрядный регистр R0.L.

### 3.7.2 Регистры-аккумуляторы

Регистры-аккумуляторы являются специализированными 32/64-разрядными регистрами данных, предназначенными для накопления результата в операциях умножения с накоплением.

DSP-ядро ELCORE-28 содержит шестнадцать 32-разрядных регистров-аккумуляторов AC0-AC15, которые могут попарно объединяться в восемь 64-разрядных, либо четыре 128-разрядных регистра. Два 32-разрядных регистра, четный и нечетный (с номером, большим на единицу), объединяются в один 64-разрядный регистр для получения 64-разрядного результата.

Структура регистрового файла регистров-аккумуляторов приводится на Рисунок 3.5.

ACn.L – 32-разрядные регистры; n=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15;

ACn.D – 64-разрядные регистры; n=0,2,4,6,8,10,12,14;

ACn.Q – 128-разрядные регистры; n=0,4,8,12.

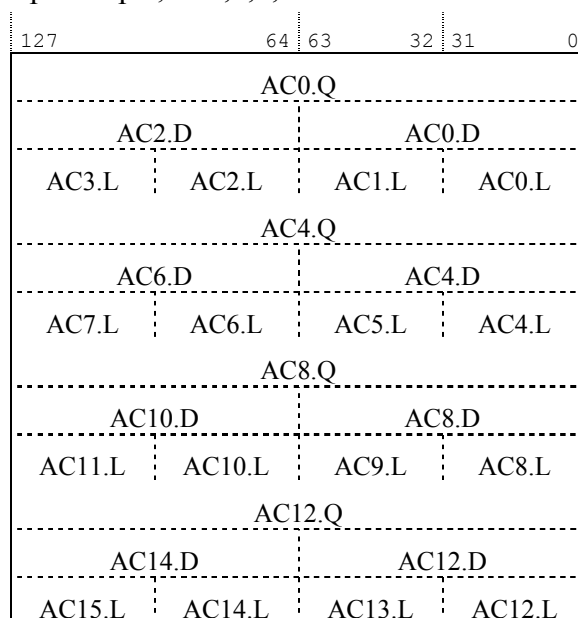


Рисунок 3.5. Структура регистрового файла регистров-аккумуляторов ELCORE-28

Регистры-аккумуляторы доступны по записи и по чтению как со стороны CPU, так и со стороны DSP-ядра.

Начальное состояние регистров-аккумуляторов равно нулю.

### 3.7.3 Регистр PDNR

Регистр PDNR - регистр управления, предназначенный для измерения параметра денормализации (PDN) и управления режимом блочной экспоненты и режимом масштабирования (Scaling).

Назначение разрядов регистра PDNR приведено в Таблица 3.5.

Начальное состояние регистра PDNR = 0x0000.

**Таблица 3.5. Назначение разрядов регистра PDNR**

Разряды регистра	Идентификатор	Назначение
0 – 4	Cpdn	Текущий код PDN
5	F	(X/L) – формат анализируемой информации (0 – Long, 1 – X16)
7	Epdn	Программный признак разрешения детектирования и изменения PDN (0 – нет разрешения, 1 – разрешение)
8,9	SC	Величина масштабирования результата (00 – нет сдвига, 01 - сдвиг на 1 разряд, 10 - сдвиг на 2 разряда)
15	Esc	Признак разрешения масштабирования результата (0 – нет разрешения, 1 – разрешение)
6,10-14	-	Не используются

### 3.7.4 Регистр CCR

Регистр CCR - регистр управления, предназначенный для хранения признаков результатов вычислительных операций. Регистр CCR содержит два поля признаков: основное {Ev,U,N,Z,V,C} (разряды [5:0]) и дополнительное {Evm,Um,Nm,Zm,Vm,Cm} (разряды [15:10]). Поле признаков в младшем байте регистра CCR является основным, т.к. на его основе формируются условия исполнения команд.

Поля признаков формируются по следующим правилам:

- при исполнении одной операции типа OP1 (AU/LU/FASU) ее признаки помещаются только в основное поле;
- при исполнении одной операции типа OP2 (MS/SH/FMU) ее признаки помещаются в оба поля;
- при одновременном выполнении двух вычислительных операций признаки, формируемые операцией типа OP1 поступают в основное поле, признаки операции типа OP2 - в дополнительное поле;
- в тех случаях, когда операция типа OP1 заполняет только часть признаков в основном поле, оставшиеся формируются операцией OP2.

Регистр CCR содержит также специальные признаки E, t и управляющие разряды RND и S.

Назначение разрядов регистра CCR приведено в Таблица 3.6.

Начальное состояние регистра CCR = 0x0000.

**Таблица 3.6. Назначение разрядов регистра CCR**

Разряды регистра	Идентификатор	Назначение
0	C	Признак переноса, сформированного в результате выполнения операции (0 – нет переноса, 1 – есть перенос)
1	V	Признак переполнения результата (0 – нет переполнения, 1 – есть переполнение)
2	Z	Признак нулевого результата (0 – результат не нулевой, 1 – результат нулевой)
3	N	Знак результата (0 – знак положительный, 1 – знак отрицательный)
4	U	Признак ненормализованного результата (0 – нормализованный результат, 1 – ненормализованный результат)
5	Ev	Запомненный ранее возникший признак переполнения результата (0 – не было переполнения, 1 – было переполнение)
6	E	Экспоненциальный признак (формируется командой CMPE)
7	t	Признак истинности условия после исполнения условной команды (t=0 – безусловная команда либо условие ложно; t=1 – условие истинно)
8	S	Бит включения режима насыщения результата (0 – отключение режима насыщения, 1 – включение режима насыщения)
9	RND	Бит управления режимом округления результата (0 – CR (Convergent Rounding), 1 – TCR (Two's-Complement Rounding))
10	Cm	Признак переноса сформированного в результате выполнения операции OP2 (0 – нет переноса, 1 – есть перенос)
11	Vm	Признак переполнения результата операции OP2 (0 – нет переполнения, 1 – есть переполнение)
12	Zm	Наличие нулевого результата операции OP2 (0 – результат не нулевой, 1 – результат нулевой)
13	Nm	Значение знака результата операции OP2 (0 – знак положительный, 1 – знак отрицательный)
14	Um	Признак ненормализованного результата операции OP2 (0 – нормализованный результат, 1 – ненормализованный результат)
15	Ev <sub>m</sub>	Запомненный ранее возникший признак переполнения результата операции OP2 (0 – не было переполнения, 1 – было переполнение)

## 3.8 Регистры адресных генераторов и виды адресной арифметики

### 3.8.1 Регистры AGU

Генератор адреса AGU содержит восемь наборов по три регистра (Рисунок 3.6): регистр адреса  $A_n$ , регистр смещения  $I_n$ , регистр модификатора  $M_n$  ( $n=0-7$ ). Эти регистры могут использоваться для хранения адресных указателей или других данных. При косвенной адресации операндов в памяти автоматически включается механизм обновления адресных указателей. Адресные регистры могут быть запрограммированы для линейной адресации, модульной адресации или реверсивной адресации.



Рисунок 3.6. Программная модель AGU

Эти регистры могут также использоваться для хранения произвольных данных.

### 3.8.2 Регистры AGU-Y

Генератор адреса AGU-Y содержит набор из четырех регистров (Рисунок 3.7): регистра адреса AT, регистров смещения IT и DT, регистра модификатора MT.



Рисунок 3.7. Программная модель AGU-Y

### 3.8.3 Назначение регистров адресных генераторов

Особенностью DSP-ядра Elcore-28 по сравнению с предшествующими модификациями DSP-ядер Elcore-xx платформы «Мультикор» является то, что расширен до 32 разрядов формат адресных регистров A0 – A7, AT. Это вызвано расширением адресного пространства DSP-кластера и выходом его за пределы доступности 16-разрядных адресных регистров, существовавших в предшествующих модификациях DSP Elcore-xx. При этом регистры смещения I0–I7, IT, DT и регистры модификаторов M0–M7, MT по-прежнему остаются 16-разрядными.

32-разрядные адресные регистры A0-A7, AT содержат адреса памяти данных. Содержимое адресного регистра может непосредственно указывать на данные в памяти либо используется для формирования указателя со смещением. Адресный регистр обновляется после формирования адресного указателя (пост-модификация).

16-разрядные регистры смещений I0-I7, IT, DT содержат значения смещений, используемых для инкрементации или декрементации адресных регистров при выполнении обновления адреса.

16-разрядные регистры модификаторов M0-M7, MT определяют тип адресной арифметики, применяемой при модификации адреса.

Адресные АЛУ поддерживают три типа арифметики: *линейную, модульную и арифметику с обратным переносом*. Для модульной арифметики содержимое регистров модификаторов определяет также модуль.

Значения модификатора Mn и соответствующие им типы адресной арифметики указаны в таблице 3.7.

**Таблица 3.7. Типы адресной арифметики**

Модификатор Mn	Адресная арифметика
0x0000	Арифметика с обратным переносом
0x0001	Модуль 2
0x0002	Модуль 3
...	...
0x7FFE	Модуль 32767 ( $2^{15} - 1$ )
0x7FFF	Модуль 32768 ( $2^{15}$ )
0x8001	Модуль 2 с кратным обращением
0x8003	Модуль 4 с кратным обращением
0x8007	Модуль 8 с кратным обращением
...	...
0x9FFF	Модуль $2^{13}$ с кратным обращением
0xBFFF	Модуль $2^{14}$ с кратным обращением
0xFFFF	Линейная арифметика (Модуль $2^{16}$ )
Остальные комбинации – резерв	

Линейная адресная арифметика ( $M_n = 0xFFFF$ )

Модификация адреса выполняется с использованием линейной адресной арифметики. 16-разрядное смещение,  $In$ ,  $+1$  или  $-1$  используется для модификации адреса. Диапазон значений  $In$  рассматривается как знаковый и находится в пределах от  $-32768$  до  $+32767$ .

*Важной особенностью линейной адресной арифметики ELcore-28 является то, что операции инкремента и декремента выполняются в 16-разрядном формате и диапазон изменения адресов находится в пределах от 0 до 0xFFFF. Таким образом, путем операций инкремента и декремента возможен переход из адресного пространства одного сегмента в адресное пространство другого сегмента в пределах всего DSP-кластера.*

Адресная арифметика с обратным переносом ( $M_n = 0x0000$ )

Этот вариант адресной арифметики выбирается посредством установки регистра модификатора в 0. Модификация адреса в этом случае выполняется аппаратно с распространением переноса в обратном направлении – от старших разрядов к младшим.

Операция модификации адреса с обратным переносом эквивалентна последовательному выполнению следующих процедур:

- изменению на обратный порядок следования разрядов в регистрах адреса и смещения (при этом старший бит становится младшим и т.д.);
- модификации адреса посредством нормальной операции сложения;
- возвращению первоначального порядка следования разрядов адреса.
- в случае, когда величина смещения составляет  $2^{(k-1)}$  (целая степень двойки), такая модификация адреса эквивалентна:
- обращению порядка следования  $k$  младших разрядов  $An$ ;
- увеличению на 1;
- возвращению исходного порядка следования  $k$  младших разрядов  $An$ .

Рассматриваемый режим адресной арифметики удобен при реализации алгоритма быстрого преобразования Фурье (БПФ).

Модульная адресная арифметика ( $M_n = \text{Modulus} - 1$ )

Модификация адреса выполняется по модулю  $M$ , где  $M$  - целое число в пределах от 2 до 32768. Арифметика по модулю  $M$  вынуждает значение адреса оставаться в пределах диапазона значений, отличающихся друг от друга не более чем на  $M-1$ .

Величина  $M-1$  хранится в регистре модификатора адреса. Нижняя граница диапазона (базовый адрес) должна иметь нули в младших  $k$  разрядах, где  $2^k \geq M$ . Верхняя граница диапазона определяется как сумма нижней границы и модуля минус единица (базовый адрес  $+ M - 1$ ). Нижняя и верхняя границы диапазона определяются значением  $An$ .

При этом необязательно устанавливать  $An$  равным базовому адресу. Достаточно того, чтобы величина  $An$  находилась в пределах требуемого диапазона.

Если при вычислении адреса в этом режиме используется смещение  $In$ , его величина не должна превышать  $M$ .

Рассматриваемый тип адресной арифметики удобен при организации циклических буферов для реализации на их основе структур данных типа очередей (FIFO), линий задержки и т.п.



Кратная модификация адреса по модулю

Этот тип адресной арифметики выбирается посредством установки в «1» 15-го разряда регистра модификатора Mn, как это показано в таблице 2.3.

Модификация адреса выполняется по модулю M, где M - степень двойки в пределах от  $2^1$  до  $2^{14}$ . Арифметика по модулю M вынуждает значение адреса оставаться в пределах диапазона значений, отличающихся друг от друга не более чем на M-1.

Величина M-1 хранится в младших 15-ти разрядах регистра модификатора адреса Mn. Нижняя граница диапазона (базовый адрес) должна иметь нули в младших k разрядах, где  $2^k \geq M$ . Верхняя граница диапазона определяется как сумма нижней границы и модуля минус единица (базовый адрес + M - 1).

Нижняя и верхняя границы диапазона определяются значением An. При этом необязательно устанавливать An равным базовому адресу. Достаточно того, чтобы величина An находилась в пределах требуемого диапазона.

### 3.8.4 Особенности X- и Y- указателей

Виды адресации памяти данных XRAM сведены в Таблица 3.7. Режим адресации определяется полем “mode” командного слова инструкции.

Таблица 3.7. Виды X-адресации памяти данных (указатели A0-A7)

Код режима адресации (mode)	Обозначение	Пояснение
000	-	Отмена пересылки
001	(An)	Косвенная
010	(An)+	Пост - автоинкремент
011	(An)-	Пост - автодекремент
100	(An)+In	Пост - автоувеличение
101	(An)-In	Пост - автоуменьшение
110	(An+In)	Индексирование (An не меняется)
111	(An+dspl)	С непосредственным смещением (A не меняется)

Примечание. По установленному признаку “u” в командном слове вычисляется исполнительный адрес без выполнения самой пересылки

Виды Y-адресации сведены в Таблица 3.8. Режим адресации определяется полем “AT” инструкции и управляющим параметром YM (11-й разряд регистра SR).

Таблица 3.8. Виды Y-адресации памяти данных (указатель AT)

Код режима адресации (поле “AT”)	YM	Обозначение	Пояснение
00	X	-	Отмена пересылки
01	X	(AT)	Косвенная
10	X	(AT)+IT	Пост - автоувеличение
11	0	(AT)+IT	Индексирование (An не меняется)
11	1	(AT)+DT	Пост - автоувеличение

### 3.8.5 Начальное состояние регистров адреса A0-A7, AT

Начальное состояние регистров A0-A7, AT DSP-ядер приведено в Таблица 3.9.

Таблица 3.9. Начальное состояние регистров A0-A7, AT

Условное обозначение	Разрядность	Наименование	Начальное состояние	
			DSP0	DSP1
A0-A7	32 R/W	Адресный регистр AGU	0x00000	0x08000
AT	32 R/W	Адресный регистр AGU-Y	0x04000	0x0C000

Таким образом, при начальной установке регистры A0-A7 указывают на начало, а регистры AT – на середину ближней (локальной) памяти соответствующего DSP.

### 3.8.6 Регистр адреса вектора прерывания IVAR

В ELcore-28 реализован механизм прерываний, рассмотренный подробнее в п.2.3.12. При отработке прерывания автоматически выполняется команда JSR IVAR, по которой происходит переход на подпрограмму обработки прерываний, находящуюся по адресу, содержащемуся в регистре адреса вектора прерывания IVAR (16 бит, запись/чтение).

Начальное состояние регистра IVAR=0x1F00.

## 3.9 Регистры устройства программного управления PCU

Устройство программного управления PCU включает в себя набор управляющих регистров и стеков:

- регистр управления и состояния DCSR;
- программный счетчик PC;
- регистр состояния SR;
- регистр-идентификатор IDR;
- регистр флагов обмена EFR;
- регистр запуска DMA DSTART;
- регистр запросов на прерывание IRQR;
- регистры масок запросов на прерывания IMASKR, QMASKR0, QMASKR1, QMASKR2;
- регистр управления арбитром памяти ARBR;
- регистр таймера TMR;
- регистр адреса окончания цикла LA;
- регистр счетчика циклов LC;
- системный стек SS;
- стеки циклов CSL, CSH;
- регистр указателей стека SP;
- регистры адреса останова SAR, SAR1 – SAR7;
- счетчик команд CNTR;
- регистр спецфункций SFR.

### 3.9.1 Регистр управления и состояния DCSR

Регистр управления и состояния (DCSR) содержит разряды управления, определяющие состояние и режим работы DSP-ядра, а также прерывания, формируемые DSP-ядром для обработки в RISC-ядре.

Назначение разрядов регистра DCSR указано в Таблица 3.10.

Начальное состояние DCSR = 0x0000.

**Таблица 3.10. Назначение разрядов регистра DCSR**

Разряды регистра	Идентификатор	Назначение
0	PI	Программное прерывание PI.
1	SE	Прерывание по ошибке стека SE
2	BRK	Прерывание по останову BREAK
3	STP	Прерывание по останову STOP
4	WT	Состояние ожидания обмена с XBUF
5–13	-	Не используется
14	RUN	Состояние исполнения программы
15	-	Не используется

### 3.9.2 Программный счетчик PC

Регистр программного счетчика PC предназначен для хранения 16-разрядного адреса инструкции в программной памяти. Инкрементированное значение PC заносится в системный стек при инициализации нового программного цикла DO, DOFOR и при входе в подпрограмму.

Начальное состояние PC = 0x0000.

### 3.9.3 Регистр состояния SR

Регистр состояния SR содержит параметры управления и состояния DSP-ядра. Разряды [7:0] регистра SR доступны только по чтению, остальные - по записи/чтению.

Назначение разрядов регистра SR указано в Таблица 3.11.

Таблица 3.11. Назначение разрядов регистра SR

Разряды регистра	Идентификатор	Назначение
0	C	Перенос
1	V	Признак переполнения
2	Z	Признак нулевого результата
3	N	Признак отрицательного результата
4	U	Признак ненормализованного результата
5	Ev	Флаг переполнения
6	E	Экспоненциальный признак
7	t	Признак истинности последнего условия
8	nBLKmod	Управление отключением блокировки конвейера на такт в случае предмодификации адреса при обращении к памяти (индексация)
9	DD	Управление режимом записи результата в инструкциях ADDSUB, ADDSUBL, ADDSUBX, FAS, CVFE (Double Destination)
10	BD	Управление блокировкой конвейера (Blocking Disabled)
11	YM	Управление режимом адресации памяти YRAM
12-15	-	Не используются

Начальное состояние регистра SR = 0x0000.

Разряды [7:0] регистра SR содержат интегральные признаки предыдущей арифметической операции.

Бит nBLKmod предназначен для управления отключением блокировки конвейера на такт в случае предмодификации адреса при обращении к памяти (режим индексации):

1) nBLKmod = 0. Каждое обращение к памяти DSP с предмодификацией адреса приводит к блокировке ядра на 1 такт. Обращение обрабатывается корректно, ограничений для модификации адреса нет.

2) nBLKmod = 1. Обращения к памяти DSP с предмодификацией адреса не приводят к блокировке ядра, однако существует ограничение на модификацию адреса: исходный адрес и адрес полученный после модификации обязательно должны находиться в одной странице памяти. Для обращений без модификации и с постмодификацией адреса ограничений нет.

Бит DD (Double Destination) = SR[9] предназначен для выбора режимов исполнения вычислительных команд, формирующих двойной результат: ADDSUB, ADDSUBL, ADDSUBX, FAS, CVFE. При DD=0 (по умолчанию) указанные команды выполняются в варианте с двумя результатами и двумя адресами записи, при DD=1 один результат удвоенного формата записывается по одному адресу D.L(D.D). (Более подробную информацию можно получить из описания указанных инструкций).

**Бит BD** (Blocking Disabled) = SR[10] предназначен для управления автоматической блокировкой программного конвейера: при  $BD = 0$  блокировка включена, при  $BD = 1$  отключена.

Пояснение: автоматическая блокировка (включена по умолчанию при  $BD=0$ ) вызывает торможение программного конвейера в тех случаях, когда последующая инструкция использует еще не сформированный результат предыдущей инструкции. Отключение автоматической блокировки ( $BD=1$ ) может производиться с целью ускорения работы программы при условии хорошего понимания работы программного конвейера ELcore-28.

Отключение автоматической блокировки не оказывает влияния на остановы вычислительного ядра, вызванные конфликтами при обращении к памяти.

Назначение бита  $YM = SR[11]$  описано в Таблица 3.8.

### 3.9.4 Регистр-идентификатор IDR

Состояние регистров-идентификаторов DSP-ядер ELcore-28 в составе DSP-кластера:  $IDR=0xn309$ , где  $n=0,1,2,3$  – номер DSP-ядра.

### 3.9.5 Регистр адреса окончания цикла LA

Регистр адреса окончания цикла LA содержит адрес последней инструкции в программном цикле DO, DOFOR. Этот регистр заносится в стек SS по команде DO, DOFOR и извлекается обратно по окончании вложенного цикла либо по команде ENDDO.

Начальное состояние  $LA = 0x0000$ .

### 3.9.6 Регистр счетчика циклов LC

Формат регистра LC приведен в Таблица 3.12.

Таблица 3.12. Назначение разрядов регистра LC

Разряды регистра	Идентификатор	Назначение
0 - 13	Nc	Текущее значение 14-разрядного счетчика программных циклов Nc – разряды 0-13 регистра LC
14	LF	Флаг цикла DO – разряд 14 регистра LC
15	FV	Флаг цикла DOFOR – разряд 15 регистра LC

Значение счетчика программных циклов Nc определяет количество повторений программного цикла DO, в пределах от 1 до  $(2^{14} - 1)$ . Этот регистр заносится в верхнюю (старшую) половину стека циклов CSL по команде DO (образуется вложенный программный цикл) и извлекается обратно по окончании вложенного цикла либо по команде ENDDO.

Начальное состояние  $LC = 0x0000$ .

### 3.9.7 Стеки SS, CSL, CSH

Устройство программного управления содержит системный стек SS и стеки циклов CSL, CSH. Системный стек SS имеет объем 15 16-разрядных слов и используется для автоматического сохранения содержимого регистра программного счетчика PC при входе в подпрограмму или в цикл DO, DOFOR. Стеки циклов имеют объем по 7×16 бит и предназначены для хранения соответственно длины цикла и адреса последней инструкции цикла (LC и LA). Стеки участвуют в обменах как 16-разрядные регистры управления – SS, CSL и CSH.

### 3.9.8 Регистр указателей стека SP

Регистр указателей стека SP содержит указатели на последнее записанное в стеки SS, CSH слово. Назначение разрядов регистра SP указано в Таблица 3.13.

Таблица 3.13. Назначение разрядов регистра SP

Разряды регистра	Идентификатор	Назначение
0 - 3	SP	указатель системного стека
4	SSE	флаг ошибки системного стека
5	UFS	флаг переполнения системного стека
6, 7	-	не используются
8-10	CP[2:0]	указатель стека циклов
11	CSE	флаг ошибки стека циклов
12	UFC	флаг переполнения стека циклов
13-15	-	не используются

Младший байт регистра SP содержит указатель и флаги системного стека; старший байт - указатель и флаги стека циклов.

Начальное состояние SP = 0x0000.

### 3.9.9 Регистры адреса останова SAR, SAR1-SAR7

Регистры адреса останова SAR, SAR1–SAR7 являются специализированными 16-разрядными регистрами, используемыми при отладке DSP-ядра. Регистры SAR, SAR1–SAR7 определяют точки останова (Breakpoint) - адрес инструкции, непосредственно перед исполнением которой должен произойти останов DSP-ядра. Перед исполнением инструкции с указанным адресом DSP-ядро переходит в состояние останова (RUN=0) и флаг прерывания BRK устанавливается в «1».

Начальное состояние SAR, SAR1–SAR7 = 0xFFFF.

### 3.9.10 Счетчик команд CNTR

Счетчик команд CNTR - специализированный 16-разрядный регистр, предназначенный для отладки DSP-ядра. Регистр CNTR задает пошаговый режим исполнения программ в соответствии с Таблица 3.14.

Начальное состояние CNTR = 0x0000.

Таблица 3.14. Назначение разрядов регистра CNTR

Счетчик CNTR	Режим исполнения программ
0x0000	Нормальный режим исполнения программ. Число исполняемых команд не ограничено.
N > 0	Пошаговый режим исполнения программ. После исполнения N инструкций DSP-ядро переходит в состояние останова (RUN=0) и флаг прерывания BRK устанавливается в "1".

### 3.9.11 Регистры управления прерываниями и DMA-обменами

В ELcore-28 имеется механизм прерываний, с помощью которого, в частности, осуществляется запуск DSP со стороны DMA. Кроме того, прерывания в DSP ELcore-28 могут поступать также со стороны CPU, другого DSP-ядра, таймеров.

Для управления DMA-обменами и прерываниями имеется следующий набор регистров:

- регистр запросов на прерывание DSP со стороны DMA, CPU, других DSP-ядер, таймеров – IRQR;
- регистр маски запросов на прерывание DSP – IMASKR;
- псевдорегистр (только запись) запуска со стороны DSP каналов DMA и других DSP-ядер – DSTART.

### 3.9.12 Механизм отработки прерываний

Отработка запросов на прерывание (в том числе на запуск DSP со стороны DMA) обрабатывается одинаковым образом:

- 1) аппаратно взводится в состояние «1» соответствующий бит регистра IRQR;
- 2) аппаратно переводится в состояние «1» бит RUN регистра DCSR (если он еще не находится в этом состоянии);
- 3) автоматически выполняется команда JSR IVAR, по которой происходит переход на подпрограмму обработки прерываний, находящуюся по адресу, содержащемуся в регистре адреса вектора прерывания IVAR. Подпрограмма обработки прерываний должна оканчиваться командой возврата из подпрограммы обработки прерывания RTI.

Поступающие прерывания не имеют иерархии приоритетов и обрабатываются последовательно. Если во время обработки прерывания приходит новый запрос, то обработка его начнется только после завершения текущей подпрограммы обработки прерывания.

### 3.9.13 Регистр запросов на прерывание DSP (IRQR)

Регистр IRQR содержит флаги запросов («1» - наличие запроса, «0» - отсутствие запроса) на прерывание DSP со стороны DMA, CPU, другого DSP-ядра, таймера. Назначение разрядов регистра IRQR приведено в Таблица 3.15.

Регистр IRQR доступен по записи и чтению со стороны CPU и DSP.

Таким образом, состояние разрядов регистра IRQR может изменяться как аппаратно – при приходе соответствующего сигнала запроса на прерывание, так и программно – при записи со стороны CPU или DSP.

Таблица 3.15. Назначение разрядов регистра IRQR

Номер разряда	Наименование разряда	Назначение
0	DRQ0	Запрос на прерывание DSP со стороны канала DMA MemCh0
1	DRQ1	Запрос на прерывание DSP со стороны канала DMA MemCh1
2	DRQ2	Запрос на прерывание DSP со стороны канала DMA MemCh2
3	DRQ3	Запрос на прерывание DSP со стороны канала DMA MemCh3
4-23	-	Резерв
24	IRQ0	Запрос на прерывание DSP со стороны DSP0
25	IRQ1	Запрос на прерывание DSP со стороны DSP1
26-27	-	Резерв
28	INT_TMR	Запрос на прерывание DSP со стороны таймера TMR
29	FPE	Исключение при исполнении операции в формате плавающей точки (V=1)
30	QT0	Запрос на прерывание DSP со стороны CPU (QSTR0)
31	QT1	Запрос на прерывание DSP со стороны CPU (QSTR1, QSTR2)

Начальное состояние регистра IRQR =0x0.

### **3.9.14 Регистры масок запросов на прерывание DSP (IMASKR, QMASKR0, QMASKR1, QMASKR2)**

Регистр IMASKR содержит 32 разряда, каждый из которых разрешает («1») либо запрещает («0») запрос на прерывание DSP от соответствующего разряда регистра IRQR. Регистр доступен по чтению и записи со стороны CPU или DSP.

Регистр маски запросов на прерывание QMASKR0 содержит 32 разряда, каждый из которых разрешает («1») либо запрещает («0») прерывание DSP от соответствующего разряда регистра запросов прерываний со стороны CPU (регистр QSTR0).

Регистр маски запросов на прерывание QMASKR1 содержит 32 разряда, каждый из которых разрешает («1») либо запрещает («0») прерывание DSP от соответствующего разряда регистра запросов прерываний со стороны CPU (регистр QSTR1).

Регистр маски запросов на прерывание QMASKR2 содержит 32 разряда, каждый из которых разрешает («1») либо запрещает («0») прерывание DSP от соответствующего разряда регистра запросов прерываний со стороны CPU (регистр QSTR2).

Начальное состояние регистров IMASKR, QMASKR0, QMASKR1, QMASKR2 - нулевое.

### **3.9.15 Регистр запуска DMA со стороны DSP (DSTART)**

Регистр DSTART доступен по только записи и предназначен для запуска соответствующего канала DMA со стороны DSP. Назначение разрядов регистра DSTART приведено в Таблица 3.16.



Таблица 3.16. Назначение разрядов регистра DSTART

Номер разряда	Наименование разряда	Назначение
0	DE0	Запрос со стороны DSP на запуск канала DMA MemCh0
1	DE1	Запрос со стороны DSP на запуск канала DMA MemCh1
2	DE2	Запрос со стороны DSP на запуск канала DMA MemCh2
3	DE3	Запрос со стороны DSP на запуск канала DMA MemCh3
4-23	-	Резерв
24	DSP0	Запрос на прерывание DSP0
25	DSP1	Запрос на прерывание DSP1
26-31	-	Резерв

### 3.9.16 Регистр таймера (TMR)

Регистр таймера TMR (32 разряда, запись/чтение) предназначен для формирования периодических запросов на прерывание DSP. Период запросов определяется значением, содержащимся в регистре TMR по формуле:

$$T_{INT} = (TMR + 1) * T_{CLK},$$

где  $T_{CLK}$  - период тактовой частоты DSP.

При  $TMR = 0$  запросы на прерывание DSP не формируются.

Регистр TMR доступен по записи и чтению. Начальное состояние регистра  $TMR = 0x0$ .

### 3.9.17 Регистр управления локальным арбитром (ARBR)

Принципы арбитража и режимы работы

Вся память DSP кластера разбита на 2 сегмента, каждый из которых соответствует определенному DSP ядру и состоит из 4 страниц каждый. Таким образом, для каждого ядра существует сегмент “своей” или ближней памяти. В архитектуре глобального коммутатора предусмотрены два локальных арбитра, каждый из них осуществляет арбитраж обращений к определенному сегменту памяти. Каждый из локальных арбитров настраивается и работает независимо от другого арбитра. Таким образом, одно ядро может иметь высший приоритет для обращений к одному сегменту памяти и низший для обращений к другому.

Каждая страница памяти состоит из четырёх физических блоков по 4К 32 разрядных слов каждый. Для организации чтения 128 разрядных слов, а так же для повышения производительности при 32 разрядных обменах с памятью применена технология расслоения памяти. Т.е. любые четыре последовательно идущих адреса одной страницы располагаются в 4-х разных физических блоках.

В случае если оба ядра обращаются к одной странице памяти, обрабатывается обращение от ядра, имеющего на данный момент высший приоритет (другое ядро останавливается до момента получения высшего приоритета). Если обращения идут к разным страницам (даже внутри одного сегмента), конфликтов не возникает. Конфликтов так же не возникает при обращении одного ядра по X и Y указателям к одной странице памяти, при условии, что обращения идут к разным физическим блокам (условие бескон-

фликтоно обращения одного DSP кодной странице памяти: для 32-х и 64-х разрядных обращений  $XAB \% 4 \neq YAB \% 4$ ).

Обращения к своей памяти не приводят к останову конвейера, если отсутствуют конфликты с другими ядрами, либо для данного ядра явно установлен высший приоритет для обращений к своей памяти (заданы значения бит  $DEN=1$  и  $DPTR = 0$  в регистре ARBR данного ядра).

Остальная память является для текущего ядра дальней. Чтение из дальней памяти неизбежно приводит к останову конвейера на четыре дополнительных такта. Одиночная запись в дальнюю память буферизуется и не приводит к блокировкам. Поддерживается пакетная запись в дальнюю память, которая так же проходит без дополнительных блокировок конвейера. Поддержка пакетных обращений имеет место при работе в режиме захвата, либо при явном задании высшего приоритета для данного ядра. При работе в режиме ограничения, максимальная длина пакета определяется значением ограничителя.

Локальный арбитр может работать в режиме *захвата* (режим по умолчанию). В этом режиме, ядро, получившее разрешение для обращений к определенному сегменту памяти, получает высший приоритет, и сохраняет его до тех пор, пока есть обращения к данному сегменту памяти. Как только обращения от текущего ядра прекращаются, право на захват циклически передается следующему ядру.

Так же предусмотрен режим *ограничения*. В этом режиме включаются счетчики обращений для каждого ядра. Если значение счетчика обращений от ядра, обладающего высшим приоритетом, превышает заданный лимит, то высший приоритет автоматически передается следующему ядру, осуществляющему обращение к памяти. Если обращений со стороны других ядер нет – счетчик сбрасывается, и передачи приоритета не происходит.

В *статическом* режиме приоритет ядер задается явно.

Регистры управления локальными арбитрами располагаются в каждом из DSP ядер и задают режим работы соответствующего локального арбитра.

### 3.9.17.1 Назначение разрядов регистра ARBR

Назначение разрядов регистра ARBR приведено в Таблица 3.17.

Таблица 3.17. Назначение разрядов регистра ARBR

Номер разряда	Наименование разряда	Назначение
0	HEN	Включение режима определения высокой плотности потоков
1	DEN	Разрешение установки явного приоритета (статический режим)
2	LEN	Бит разрешения ограничителя
3, 6, 7	-	Резерв
4-5	DPTR	Номер ядра, обладающего наивысшим приоритетом
8-13	Limit	Максимальное значение счетчика обращений
14-15	-	резерв

HEN – Включение режима определения высокой плотности потоков. Используется в режиме захвата ( $LEN = 0$ ). Если  $HEN = 1$ , то включаются счетчики, определяющие

плотность обращений ядер к данному сегменту. Если плотность обращений хотя бы от одного ядра больше 75% – то при значениях  $HEN = 1$  и  $LEN = 0$  передача приоритета происходит каждый такт.

$DEN$  – разрешение установки явного приоритета (статический режим). Если данный бит установлен в 1, то при возникновении конфликта приоритет отдается обращению от ядра, номер которого определяется битами  $DPTR$ .

$DPTR$  – определяет номер ядра, обладающего наивысшим приоритетом при обращении к сегменту памяти данного DSP.  $DPTR = 0$  задает высший приоритет для данного ядра, 1 – высший приоритет для соседа с меньшим номером, далее циклически в сторону уменьшения номера ядра.

$LEN$  – бит разрешения ограничителя. Если данный бит установлен в 1, арбитр работает в режиме ограничения, если бит установлен в 0 арбитр работает в режиме захвата.

$Limit$  – задает максимальное значение счетчика обращений, в режиме ограничения. В этом режиме предусмотрена автоматическая смена приоритета.

### **3.9.17.2 Механизм передачи приоритета**

Передача приоритета осуществляется циклически, между ядрами, осуществляющими обращение к памяти. Механизм передачи приоритета срабатывает в следующих случаях:

- ядро, обладавшее высшим приоритетом, не обращается к текущему сегменту памяти,
- в режиме захвата при  $LEN = 0$  и  $HEN = 1$  плотность обращений хотя бы от одного ядра больше 75%.
- в режиме ограничения  $LEN = 1$ , если значение счетчика обращений от ядра с высшим приоритетом достигло значения  $Limit$ .

В статическом режиме передачи приоритета не осуществляется.

Начальное состояние регистра  $ARBR = 0x0F01$ .

### **3.9.18 Регистр спецфункций (SFR)**

Регистр спецфункций SFR (32 разряда, запись/чтение) предназначен для реализации специальных вычислительных функций. Назначение разрядов регистра SFR определяется реализуемой функцией.

### 3.10 Программный конвейер DSP-ядра ELcore-28

Программный конвейер DSP-ядра ELcore-28 содержит 7 фаз.

Отличие его от программного конвейера DSP-ядра ELcore-18 состоит в том, что исполнение вычислительных команд выполняется не за три, а за два такта (на 6-й и 7-й фазе конвейера).

1) Исполнение вычислительных команд

1 фаза (A)	2 фаза (F)	3 фаза (D)	4 фаза (E)	5 фаза (E1)	6 фаза (E2)	7 фаза (E3)
Выдача адреса на PRAM	Чтение инструкции из PRAM	Декодирование инструкции	Формирование бло-кировок	Выборка данных из RF	Исполнение инструкции (1 фаза)	Исполнение инструкции (2 фаза)

2) Исполнение команд MOVE XRAM, YRAM -> RF

1 фаза (A)	2 фаза (F)	3 фаза (D)	4 фаза (E)	5 фаза (E1)	6 фаза (E2)	7 фаза (E3)
Выдача адреса на PRAM	Чтение инструкции из PRAM	Декодирование инструкции	Модификация адреса XRAM	Выдача адреса на XRAM	Чтение данных из XRAM	Запись данных в RF

3) Исполнение команд MOVE RF -> XRAM

1 фаза (A)	2 фаза (F)	3 фаза (D)	4 фаза (E)	5 фаза (E1)	6 фаза (E2)	7 фаза (E3)
Выдача адреса на PRAM	Чтение инструкции из PRAM	Декодирование инструкции	Модификация адреса XRAM	Запись данных в XRAM	-	-

4) Исполнение команд MOVE RF, RC, #16/32 -> RF

1 фаза (A)	2 фаза (F)	3 фаза (D)	4 фаза (E)	5 фаза (E1)	6 фаза (E2)	7 фаза (E3)
Выдача адреса на PRAM	Чтение инструкции из PRAM	Декодирование инструкции	Формирование бло-кировок	Выборка данных из RC	Запись данных в RF	-

5) Исполнение команд MOVE RF, #16/32 -> RC(кр.CCR,PDNR,AC)

1 фаза (A)	2 фаза (F)	3 фаза (D)	4 фаза (E)	5 фаза (E1)	6 фаза (E2)	7 фаза (E3)
Выдача адреса на PRAM	Чтение инструкции из PRAM	Декодирование инструкции	Выборка данных из RF	Запись данных в RC	-	-

Таким, образом, при исполнении различных операций фазы конвейера DSP-ядра ELcore-28 имеют следующее содержание:

а) при выполнении вычислительной операции:

- 1 фаза (A): Формирование адреса памяти программ.
- 2 фаза (F): Выборка инструкции из программной памяти.
- 3 фаза (D): Декодирование инструкции.
- 4 фаза (E): Формирование блокировок конвейера.
- 5 фаза (E1): Чтение данных из RF.
- 6 фаза (E2): Исполнение инструкции.
- 7 фаза (E3): Исполнение инструкции, запись данных в RF.

б) при чтении из памяти данных:

- 1 фаза (A): Формирование адреса памяти программ.
- 2 фаза (F): Выборка инструкции из программной памяти.
- 3 фаза (D): Декодирование инструкции.
- 4 фаза (E): Формирование адреса памяти данных.
- 5 фаза (E1): Выдача адреса на память данных.
- 6 фаза (E2): Чтение из памяти данных в буферный регистр.
- 7 фаза (E3): Запись данных в RF.

в) при записи в память данных:

- 1 фаза (A): Формирование адреса памяти программ.
- 2 фаза (F): Выборка инструкции из программной памяти.
- 3 фаза (D): Декодирование инструкции.
- 4 фаза (E): Формирование адреса памяти данных.
- 5 фаза (E1): Выдача адреса на память данных и запись в память данных.

г) при записи в регистр RF:

- 1 фаза (A): Формирование адреса памяти программ.
- 2 фаза (F): Выборка инструкции из программной памяти.
- 3 фаза (D): Декодирование инструкции.
- 4 фаза (E): Формирование блокировок конвейера.
- 5 фаза (E1): Чтение данных из RF или регистра управления.
- 6 фаза (E2): Запись в RF.

д) при записи в регистр управления:

- 1 фаза (A): Формирование адреса памяти программ.
- 2 фаза (F): Выборка инструкции из программной памяти.
- 3 фаза (D): Декодирование инструкции.
- 4 фаза (E): Чтение данных из RF.
- 5 фаза (E1): Запись в регистр управления.

Примечание. При записи/чтении памяти данных арбитром могут вводиться дополнительные такты ожидания.

### 3.11 Перечень адресуемых регистров DSP-кластера

Перечень адресуемых регистров DSP-кластера в составе 1892BM7Я приведен в Таблица 3.18.

Таблица 3.18 Перечень адресуемых регистров DSP-кластера в составе 1892BM7Я

(i=0,1,2,3 – номер DSP; BASE(0)=0x1848\_0000; BASE(1)=0x1888\_0000; BASE(2)=0x18C8\_0000; BASE(3)=0x1908\_0000)

Условное обозначение	Тип	Назначение регистра	Адрес регистра
<u>Общие регистры управления и состояния</u>			
MASKR_DSP	R/W	Регистр маски прерываний	0x1848_1000
QSTR_DSP	R	Регистр запросов прерываний	0x1848_1004
CSR_DSP	R/W	Регистр управления и состояния	0x1848_1008
<u>Регистры буфера обмена XBUF</u>			
X0[31:0]	R/W	Регистр обмена X0	0x187F_FF00
X0[63:32]	R/W	Регистр обмена X0	0x187F_FF04
X1[31:0]	R/W	Регистр обмена X1	0x187F_FF08
X1[63:32]	R/W	Регистр обмена X1	0x187F_FF0C
X2[31:0]	R/W	Регистр обмена X2	0x187F_FF10
X2[63:32]	R/W	Регистр обмена X2	0x187F_FF14
X3[31:0]	R/W	Регистр обмена X3	0x187F_FF18
X3[63:32]	R/W	Регистр обмена X3	0x187F_FF1C
X4[31:0]	R/W	Регистр обмена X4	0x187F_FF20
X4[63:32]	R/W	Регистр обмена X4	0x187F_FF24
X5[31:0]	R/W	Регистр обмена X5	0x187F_FF28
X5[63:32]	R/W	Регистр обмена X5	0x187F_FF2C
X6[31:0]	R/W	Регистр обмена X6	0x187F_FF30
X6[63:32]	R/W	Регистр обмена X6	0x187F_FF34
X7[31:0]	R/W	Регистр обмена X7	0x187F_FF38
X7[63:32]	R/W	Регистр обмена X7	0x187F_FF3C
X8[31:0]	R/W	Регистр обмена X8	0x187F_FF40
X8[63:32]	R/W	Регистр обмена X8	0x187F_FF44
X9[31:0]	R/W	Регистр обмена X9	0x187F_FF48
X9[63:32]	R/W	Регистр обмена X9	0x187F_FF4C
X10[31:0]	R/W	Регистр обмена X10	0x187F_FF50
X10[63:32]	R/W	Регистр обмена X10	0x187F_FF54
X11[31:0]	R/W	Регистр обмена X11	0x187F_FF58
X11[63:32]	R/W	Регистр обмена X11	0x187F_FF5C
X12[31:0]	R/W	Регистр обмена X12	0x187F_FF60
X12[63:32]	R/W	Регистр обмена X12	0x187F_FF64
X13[31:0]	R/W	Регистр обмена X13	0x187F_FF68
X13[63:32]	R/W	Регистр обмена X13	0x187F_FF6C
X14[31:0]	R/W	Регистр обмена X14	0x187F_FF70
X14[63:32]	R/W	Регистр обмена X14	0x187F_FF74
X15[31:0]	R/W	Регистр обмена X15	0x187F_FF78
X15[63:32]	R/W	Регистр обмена X15	0x187F_FF7C
X16[31:0]	R/W	Регистр обмена X16	0x187F_FF80
X16[63:32]	R/W	Регистр обмена X16	0x187F_FF84
X17[31:0]	R/W	Регистр обмена X17	0x187F_FF88
X17[63:32]	R/W	Регистр обмена X17	0x187F_FF8C
X18[31:0]	R/W	Регистр обмена X18	0x187F_FF90
X18[63:32]	R/W	Регистр обмена X18	0x187F_FF94
X19[31:0]	R/W	Регистр обмена X19	0x187F_FF98
X19[63:32]	R/W	Регистр обмена X19	0x187F_FF9C
X20[31:0]	R/W	Регистр обмена X20	0x187F_FFA0
X20[63:32]	R/W	Регистр обмена X20	0x187F_FFA4

Условное обозначение	Тип	Назначение регистра	Адрес регистра
X21[31:0]	R/W	Регистр обмена X21	0x187F_FFA8
X21[63:32]	R/W	Регистр обмена X21	0x187F_FFAC
X22[31:0]	R/W	Регистр обмена X22	0x187F_FFB0
X22[63:32]	R/W	Регистр обмена X22	0x187F_FFB4
X23[31:0]	R/W	Регистр обмена X23	0x187F_FFB8
X23[63:32]	R/W	Регистр обмена X23	0x187F_FFBC
X24[31:0]	R/W	Регистр обмена X24	0x187F_FFC0
X24[63:32]	R/W	Регистр обмена X24	0x187F_FFC4
X25[31:0]	R/W	Регистр обмена X25	0x187F_FFC8
X25[63:32]	R/W	Регистр обмена X25	0x187F_FFCC
X26[31:0]	R/W	Регистр обмена X26	0x187F_FFD0
X26[63:32]	R/W	Регистр обмена X26	0x187F_FFD4
X27[31:0]	R/W	Регистр обмена X27	0x187F_FFD8
X27[63:32]	R/W	Регистр обмена X27	0x187F_FFDC
X28[31:0]	R/W	Регистр обмена X28	0x187F_FFE0
X28[63:32]	R/W	Регистр обмена X28	0x187F_FFE4
X29[31:0]	R/W	Регистр обмена X29	0x187F_FFE8
X29[63:32]	R/W	Регистр обмена X29	0x187F_FFEC
X30[31:0]	R/W	Регистр обмена X30	0x187F_FFF0
X30[63:32]	R/W	Регистр обмена X30	0x187F_FFF4
X31[31:0]	R/W	Регистр обмена X31	0x187F_FFF8
X31[63:32]	R/W	Регистр обмена X31	0x187F_FFFC
		<u>PCU</u>	
DCSR	R/W	Регистр режима работы	BASE(i)+0x0100
SR	R/W	Регистр состояния	BASE(i)+0x0104
IDR	R	Регистр-идентификатор	BASE(i)+0x0108
EFR	R	Регистр флагов обмена	BASE(i)+0x010C
DSTART	W	Регистр запуска DMA со стороны DSP и запросов на прерывания других DSP	BASE(i)+0x010C
IRQR	R/W	Регистр запросов на прерывание DSP	BASE(i)+0x0110
IMASKR	R/W	Регистр маски запросов на прерывания DSP	BASE(i)+0x0114
TMR	R/W	Регистр таймера DSP	BASE(i)+0x0118
ARBR	R/W	Регистр управления арбитром памяти DSP	BASE(i)+0x011C
PC	R/W	Программный счетчик	BASE(i)+0x0120
SS	R/W	Стек программного счетчика	BASE(i)+0x0124
LA	R/W	Регистр адреса цикла	BASE(i)+0x0128
CSL	R/W	Стек адреса цикла	BASE(i)+0x012C
LC	R/W	Счетчик циклов	BASE(i)+0x0130
CSH	R/W	Стек счетчика циклов	BASE(i)+0x0134
SP	R/W	Регистр указателя стека	BASE(i)+0x0138
SAR	R/W	Регистр адреса останова	BASE(i)+0x013C
CNTR	R/W	Счетчик исполненных команд	BASE(i)+0x0140
SAR1	R/W	Регистр адреса останова	BASE(i)+0x0144
SAR2	R/W	Регистр адреса останова	BASE(i)+0x0148
SAR3	R/W	Регистр адреса останова	BASE(i)+0x014C
SAR4	R/W	Регистр адреса останова	BASE(i)+0x0150
SAR5	R/W	Регистр адреса останова	BASE(i)+0x0154
SAR6	R/W	Регистр адреса останова	BASE(i)+0x0158
SAR7	R/W	Регистр адреса останова	BASE(i)+0x015C
		Регистры состояния ALU	
CCR	R/W	Регистр кодов условий	BASE(i)+0x0160
PDNR	R/W	Регистр параметра денормализации	BASE(i)+0x0164
SFR	R/W	Регистр специальных функций	BASE(i)+0x0168
		<u>AGU, AGU-Y</u>	
A0	R/W	Регистр адреса A0	BASE(i)+0x0080
A1	R/W	Регистр адреса A1	BASE(i)+0x0084



Условное обозначение	Тип	Назначение регистра	Адрес регистра
A2	R/W	Регистр адреса A2	BASE(i)+0x0088
A3	R/W	Регистр адреса A3	BASE(i)+0x008C
A4	R/W	Регистр адреса A4	BASE(i)+0x0090
A5	R/W	Регистр адреса A5	BASE(i)+0x0094
A6	R/W	Регистр адреса A6	BASE(i)+0x0098
A7	R/W	Регистр адреса A7	BASE(i)+0x009C
I0	R/W	Регистр индекса I0	BASE(i)+0x00A0
I1	R/W	Регистр индекса I1	BASE(i)+0x00A4
I2	R/W	Регистр индекса I2	BASE(i)+0x00A8
I3	R/W	Регистр индекса I3	BASE(i)+0x00AC
I4	R/W	Регистр индекса I4	BASE(i)+0x00B0
I5	R/W	Регистр индекса I5	BASE(i)+0x00B4
I6	R/W	Регистр индекса I6	BASE(i)+0x00B8
I7	R/W	Регистр индекса I7	BASE(i)+0x00BC
M0	R/W	Регистр модификатора M0	BASE(i)+0x00C0
M1	R/W	Регистр модификатора M1	BASE(i)+0x00C4
M2	R/W	Регистр модификатора M2	BASE(i)+0x00C8
M3	R/W	Регистр модификатора M3	BASE(i)+0x00CC
M4	R/W	Регистр модификатора M4	BASE(i)+0x00D0
M5	R/W	Регистр модификатора M5	BASE(i)+0x00D4
M6	R/W	Регистр модификатора M6	BASE(i)+0x00D8
M7	R/W	Регистр модификатора M7	BASE(i)+0x00DC
AT	R/W	Регистр адреса AT	BASE(i)+0x00E0
IT	R/W	Регистр индекса IT	BASE(i)+0x00E4
MT	R/W	Регистр модификатора MT	BASE(i)+0x00E8
DT	R/W	Регистр модификатора DT	BASE(i)+0x00EC
IVAR	R/W	Регистр адреса вектора прерывания	BASE(i)+0x00FC
		<u>Регистры данных RF</u>	
R0.L		Регистр данных	BASE(i)+0x0000
R2.L		Регистр данных	BASE(i)+0x0004
R4.L		Регистр данных	BASE(i)+0x0008
R6.L		Регистр данных	BASE(i)+0x000C
R8.L		Регистр данных	BASE(i)+0x0010
R10.L		Регистр данных	BASE(i)+0x0014
R12.L		Регистр данных	BASE(i)+0x0018
R14.L		Регистр данных	BASE(i)+0x001C
R16.L		Регистр данных	BASE(i)+0x0020
R18.L		Регистр данных	BASE(i)+0x0024
R20.L		Регистр данных	BASE(i)+0x0028
R22.L		Регистр данных	BASE(i)+0x002C
R24.L		Регистр данных	BASE(i)+0x0030
R26.L		Регистр данных	BASE(i)+0x0034
R28.L		Регистр данных	BASE(i)+0x0038
R30.L		Регистр данных	BASE(i)+0x003C
R1.L		Регистр данных	BASE(i)+0x0040
R3.L		Регистр данных	BASE(i)+0x0044
R5.L		Регистр данных	BASE(i)+0x0048
R7.L		Регистр данных	BASE(i)+0x004C
R9.L		Регистр данных	BASE(i)+0x0050
R11.L		Регистр данных	BASE(i)+0x0054
R13.L		Регистр данных	BASE(i)+0x0058
R15.L		Регистр данных	BASE(i)+0x005C
R17.L		Регистр данных	BASE(i)+0x0060
R19.L		Регистр данных	BASE(i)+0x0064
R21.L		Регистр данных	BASE(i)+0x0068
R23.L		Регистр данных	BASE(i)+0x006C



Условное обозначение	Тип	Назначение регистра	Адрес регистра
R25.L		Регистр данных	BASE(i)+0x0070
R27.L		Регистр данных	BASE(i)+0x0074
R29.L		Регистр данных	BASE(i)+0x0078
R31.L		Регистр данных	BASE(i)+0x007C
R1.D[31:0]		Регистр данных	BASE(i)+0x0180
R1.D[63:32]		Регистр данных	BASE(i)+0x0184
R3.D[31:0]		Регистр данных	BASE(i)+0x0188
R3.D[63:32]		Регистр данных	BASE(i)+0x018C
R5.D[31:0]		Регистр данных	BASE(i)+0x0190
R5.D[63:32]		Регистр данных	BASE(i)+0x0194
R7.D[31:0]		Регистр данных	BASE(i)+0x0198
R7.D[63:32]		Регистр данных	BASE(i)+0x019C
R9.D[31:0]		Регистр данных	BASE(i)+0x01A0
R9.D[63:32]		Регистр данных	BASE(i)+0x01A4
R11.D[31:0]		Регистр данных	BASE(i)+0x01A8
R11.D[63:32]		Регистр данных	BASE(i)+0x01AC
R13.D[31:0]		Регистр данных	BASE(i)+0x01B0
R13.D[63:32]		Регистр данных	BASE(i)+0x01B4
R15.D[31:0]		Регистр данных	BASE(i)+0x01B8
R15.D[63:32]		Регистр данных	BASE(i)+0x01BC
R17.D[31:0]		Регистр данных	BASE(i)+0x01C0
R17.D[63:32]		Регистр данных	BASE(i)+0x01C4
R19.D[31:0]		Регистр данных	BASE(i)+0x01C8
R19.D[63:32]		Регистр данных	BASE(i)+0x01CC
R21.D[31:0]		Регистр данных	BASE(i)+0x01D0
R21.D[63:32]		Регистр данных	BASE(i)+0x01D4
R23.D[31:0]		Регистр данных	BASE(i)+0x01D8
R23.D[63:32]		Регистр данных	BASE(i)+0x01DC
R25.D[31:0]		Регистр данных	BASE(i)+0x01E0
R25.D[63:32]		Регистр данных	BASE(i)+0x01E4
R27.D[31:0]		Регистр данных	BASE(i)+0x01E8
R27.D[63:32]		Регистр данных	BASE(i)+0x01EC
R29.D[31:0]		Регистр данных	BASE(i)+0x01F0
R29.D[63:32]		Регистр данных	BASE(i)+0x01F4
R31.D[31:0]		Регистр данных	BASE(i)+0x01F8
R31.D[63:32]		Регистр данных	BASE(i)+0x01FC
		<u>Регистры-аккумуляторы</u>	
AC0	R/W	Регистр-аккумулятор AC0	BASE(i)+0x0200
AC1	R/W	Регистр-аккумулятор AC1	BASE(i)+0x0204
AC2	R/W	Регистр-аккумулятор AC2	BASE(i)+0x0208
AC3	R/W	Регистр-аккумулятор AC3	BASE(i)+0x020C
AC4	R/W	Регистр-аккумулятор AC4	BASE(i)+0x0210
AC5	R/W	Регистр-аккумулятор AC5	BASE(i)+0x0214
AC6	R/W	Регистр-аккумулятор AC6	BASE(i)+0x0218
AC7	R/W	Регистр-аккумулятор AC7	BASE(i)+0x021C
AC8	R/W	Регистр-аккумулятор AC8	BASE(i)+0x0220
AC9	R/W	Регистр-аккумулятор AC9	BASE(i)+0x0224
AC10	R/W	Регистр-аккумулятор AC10	BASE(i)+0x0228
AC11	R/W	Регистр-аккумулятор AC11	BASE(i)+0x022C
AC12	R/W	Регистр-аккумулятор AC12	BASE(i)+0x0230
AC13	R/W	Регистр-аккумулятор AC13	BASE(i)+0x0234
AC14	R/W	Регистр-аккумулятор AC14	BASE(i)+0x0238
AC15	R/W	Регистр-аккумулятор AC15	BASE(i)+0x023C

Условное обозначение	Тип	Назначение регистра	Адрес регистра
		<u>Отладочные регистры</u>	
dbDCSR	R/W	Регистр управления в режиме отладки	BASE(i)+0x0500
Cnt_RUN	R	Счетчик тактов	BASE(i)+0x0518
dbPCa	R	Программный счетчик, стадия a	BASE(i)+0x0524
dbPCf	R	Программный счетчик, стадия f	BASE(i)+0x0528
dbPCd	R	Программный счетчик, стадия d	BASE(i)+0x052C
dbPCe	R	Программный счетчик, стадия e	BASE(i)+0x0520
dbPCe1	R	Программный счетчик, стадия e1	BASE(i)+0x0530
dbPCe2	R	Программный счетчик, стадия e2	BASE(i)+0x0534
dbPCe3	R	Программный счетчик, стадия e3	BASE(i)+0x0538
dbSAR	R/W	Регистр адреса останова 0 в режиме отладки	BASE(i)+0x053C
dbCNTR	R/W	Счетчик исполненных команд в режиме отладки	BASE(i)+0x0540
dbSAR1	R/W	Регистр адреса останова 1 в режиме отладки	BASE(i)+0x0544
dbSAR2	R/W	Регистр адреса останова 2 в режиме отладки	BASE(i)+0x0548
dbSAR3	R/W	Регистр адреса останова 3 в режиме отладки	BASE(i)+0x054C
dbSAR4	R/W	Регистр адреса останова 4 в режиме отладки	BASE(i)+0x0550
dbSAR5	R/W	Регистр адреса останова 5 в режиме отладки	BASE(i)+0x0554
dbSAR6	R/W	Регистр адреса останова 6 в режиме отладки	BASE(i)+0x0558
dbSAR7	R/W	Регистр адреса останова 7 в режиме отладки	BASE(i)+0x055C

## 4. СИСТЕМНОЕ УПРАВЛЕНИЕ

### 4.1 Система синхронизации

#### 4.1.1 Входы синхронизации и умножители частоты

Микросхема 1892ВМ7Я имеет следующие входы синхронизации:

- ХТІ - частота 10 МГц для синхронизации всех умножителей частоты микросхемы;
- RTC\_XTI - частота таймера реального времени 32 КГц;
- PCLK - частота работы шины PCI величиной от 33 до 66 МГц;
- SRIO\_CLK - частота работы контроллера интерфейса Serial RapidIO (частота передачи кодовых групп) 125 МГц;
- PIXCLK – синхронизация пикселей порта VPIN.

Для синхронизации работы узлов микросхемы 1892ВМ7 используются умножители частоты на основе схемы фазовой автоподстройки частоты (PLL). Имеются следующие умножители частоты:

- PLL\_CORE – формирует тактовую частоту работы CPU, UART, IT, WDT, I2C, коммутатора AXI, системной части всех устройств микросхемы;
- PLL\_DSP – формирует тактовую частоту работы основной части DSP;
- PLL\_DDR – формирует тактовую частоту работы основной части портов DDR\_PORT и памяти, подключенной к ним;
- PLL\_MPORT – формирует выходную тактовую частоту SCLK для работы памяти типа SDRAM, подключенной к MPORT;
- PLL\_TX\_SWIC0, PLL\_TX\_SWIC1 – формирует тактовую частоту для передачи последовательного кода из контроллеров SWIC0, SWIC1 соответственно, в сеть SpaceWire;

Частота, поступающая на вход, ХТІ делится на 2 и далее поступает на входы всех PLL.

### 4.1.2 Управление работой PLL

Управление работой PLL осуществляется при помощи регистра CR\_PLL, формат которого приведен в Таблица 4.1.

Таблица 4.1 Формат регистра CR\_PLL

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	PLL_DDR_EN	Выбор источника тактовой частоты для работы памяти типа DDR SDRAM, подключенной к DDR_PORT0, DR_PORT1: 1 – PLL_DDR; 0 – вход XTI.	R/W	0
30:24	CLK_SEL_DDR[6:0]	Коэффициент умножения/деления входной частоты PLL DDR (частота XTI, деленная на 2): 00 – 1/16; 01 – 1 02 – 2; 03 – 3; ... 7E – 126; 7F – 127.	R/W	1
23	PLL_DSP_EN	Выбор источника тактовой частоты для работы ядер DSP: 1 – PLL_DSP; 0 – вход XTI.	R/W	0
22:16	CLK_SEL_DSP[6:0]	Коэффициент умножения/деления входной частоты PLL_DSP (частота XTI, деленная на 2): 00 – 1/16; 01 – 1 02 – 2; 03 – 3; ... 7F – 127.	R/W	1
15	-	Резерв	-	0
14:8	CLK_SEL_MPORT[6:0]	Коэффициент умножения/деления входной частоты PLL_MPORT (частота XTI, деленная на 2): 00 – 1/16; 01 – 1 02 – 2; 03 – 3; ... 7F – 127.	R/W	1
7	-	Резерв	-	0
6:0	CLK_SEL_CORE[6:0]	Коэффициент умножения/деления входной частоты PLL_CORE (частота XTI, деленная на 2): 00 – 1/16; 01 – 1 02 – 2; 03 – 3; ... 7F – 127.	R/W	1

Номерация разрядов всех регистров соответствует нумерации разрядов памяти CPU. Если разряды регистров доступны только по записи или не используются (резерв), то при чтении из них считываются нули. Если разряды регистров доступны только по чтению или не используются, то при записи в них необходимо указывать нули.

Выбор источника тактовой частоты (далее CLK) для работы CPU, UART, IT0, IT1, WDT, коммутатора AXI и системной части всех устройств микросхемы определяется входом микросхемы PLL\_EN:

1 – PLL\_CORE;

0 – вход XTI.

Выбор источника формирования выходной частоты SCLK также определяется входом микросхемы PLL\_EN:

1 – PLL\_MPORT;

0 – вход XTI.

### 4.1.3 Отключение и включение тактовой частоты

В данной микросхеме имеется два режима энергосбережения:

- уменьшение тактовой частоты работы устройств;
- отключение тактовой частоты работы устройств.

Уменьшение тактовой частоты устройств выполняется при записи необходимого кода в поле CLK\_SEL регистра CR\_PLL. При этом значение тактовой частоты изменится через время не более чем 2 мс.

Отключение тактовой частоты от устройств выполняется при помощи регистра CLK\_EN, формат которого приведен в Таблица 4.2.

**Таблица 4.2 Формат регистра CLK\_EN**

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	-	Резерв	-	0
29:28	CLKEN_SRIO[1:0]	Управление включением тактовой частоты контроллеров SRIO1 и SRIO0 соответственно, поступающей с входа микросхемы SRIO_CLK: 1 – частота включена; 0 – частота выключена.	R/W	0
27:26	-	Резерв	-	0
25:24	CLKEN_SWIC[1:0]	Управление включением тактовых частот SWIC1 и SWIC0 соответственно, поступающих от PLL_CORE и PLL_MPORT: 1 – частота включена; 0 – частота выключена.	R/W	0
23:21	-	Резерв	-	0
20	CLKEN_EMAC	Управление включением тактовой частоты Ethernet MAC, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена.	R/W	0
19	CLKEN_VPOUT	Управление включением тактовой частоты VPOUT, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена.	R/W	0
18	CLKEN_VPIN	Управление включением тактовой частоты VPIN, поступающей от PLL_CORE:	R/W	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
		1 – частота включена; 0 – частота выключена.		
17	-	Резерв	-	0
16	CLKEN_PMSC	Управление включением тактовой частоты PMSC, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена.	R/W	0
15:14	-	Резерв	-	0
13:12	CLKEN_DMA[1:0]	Управление включением тактовой частоты каналов DMA MEM_CH10 – MEM_CH17 и MEM_CH00 – MEM_CH07 соответственно, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена.	R/W	0
11:10	-	Не используется	-	0
9	CLKEN_MFBSP1	Управление включением тактовой частоты MFBSP1 (и его DMA), поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена	R/W	0
8	CLKEN_MFBSP0	Управление включением тактовой частоты MFBSP0 (и его DMA), поступающая от PLL_CORE: 1 – частота включена; 0 – частота выключена	R/W	0
7:4	CLKEN_DSP[3:0]	Управление включением тактовой частоты DSP3, DSP2, DSP1 и DSP0 соответственно, поступающей от PLL_DSP: 1 – частота включена; 0 – частота выключена. При выключении частоты соответствующего DSP его регистры становятся недоступны для CPU.	R/W	0
3:2	CLKEN_DDR[1:0]	Управление включением тактовой частоты портов DDR_PORT1 и DDR_PORT0 соответственно, поступающей от PLL_DDR: 1 – частота включена; 0 – частота выключена. При выключении частоты соответствующего DDR_PORT его регистры доступны для CPU. Память, подключенная к данному порту DDR_PORT, становится не доступной, и все передачи данных переадресуются в MPORT	R/W	0
1	-	Не используется	-	0
0	CLKEN_CORE	Управление включением тактовой частоты, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена.	R/W	1

При CLKEN\_CORE = 1 частота от PLL\_CORE (при PLL\_EN = 1) всегда поступает на CPU, UART, IT0, IT1, WDT, I2C, коммутатор AXI и системную часть всех устройств микросхемы.

При CLKEN\_CORE = 1 частота от PLL\_CORE (при PLL\_EN = 1), поступающая на DMA, PMSC, VPIN, VPOUT, MFBSP, Ethernet MAC, SWIC, SRIO, может быть отключена при помощи соответствующего разряда регистра CLK\_EN.

Частота от PLL\_DDR (при PLL\_DDR\_EN = 1), поступающая на основную часть портов DDR\_PORT[1:0], может быть отключена при помощи регистра CLK\_EN.

Частота от PLL\_DSP (при PLL\_DSP\_EN = 1), поступающая на основную часть ядер DSP[3:0], может быть отключена при помощи регистра CLK\_EN.

Устройство, входная частота которого отключается, должно быть в неактивном состоянии. Все передачи данных, выполняемые им, должны быть завершены.

Отключение внутренней тактовой частоты ядра микросхемы, должно выполняться следующим образом:

- программа CPU должна выполняться из кэш программ или из внутренней памяти SRAM;
- DMA, все контроллеры и порты переводятся в неактивное состояние. Все передачи данных должны быть завершены;
- записать 1 в разряд SREF регистра SDRCSR MPORT. По данной операции SDRAM переводится в режим саморегенерации;
- произвести запись 0 в разряд CLKEN\_CORE регистра CLK\_EN. По этой операции внутренняя тактовая частота ядра микросхемы отключается. За этой командой должна стоять команда NOP.

Включение внутренней тактовой частоты осуществляется по любому внешнему прерыванию nIRQ[3:0] или NMI. Обработка исключения по данным прерываниям в этом случае должна выполняться следующим образом:

- записать 1 в разряд EXIT регистра SDRCSR MPORT. По данной операции SDRAM выводится из режима саморегенерации;
- выполнить 10 команд NOP.

## 4.2 Контроллер прерываний

Все сигналы внутренних и внешних прерываний поступают на входы псевдорегистров. Эти регистры не имеют элементов памяти и доступны только по чтению.

Каждый разряд регистров QSTR содержит запрос прерывания от внутренних узлов микросхемы и от внешних сигналов прерывания nIRQ[3:0] в не зависимости от состояния соответствующих разрядов регистров MASKR:

0 – нет запроса;

1 – есть запрос.

Сигналы внутренних прерываний формируются в соответствующих устройствах при выполнении определенных условий. В процессе обслуживания прерывания необходимо проанализировать состояние устройства для определения причины его возникновения. Сброс прерывания осуществляется в момент исключения причины возникновения данного прерывания. Например, прерывание от LPORT сбрасывается при записи данных в буфер LTx или при чтении данных из буфера LRx.

Все незамаскированные прерывания объединяются по «или» и поступают в поле IP[7:2] регистр Cause CPU.

Исходное состояние регистров QSTR – нули.

Каждое внутреннее прерывание можно замаскировать. Для этого имеются 4 регистра маски MASKR0, MASK1, MASK2 и MASK3 форматы которых аналогичны форматам соответствующих регистров QSTR0, QSTR1, QSTR2 и QSTR3. Исходное состояние ре-

гистров маски – нули (все прерывания запрещены). Регистры маски доступны по записи и чтению.

Форматы регистров QSTR приведены в Таблица 4.3 - Таблица 4.6.

**Таблица 4.3 Формат регистра QSTR0**

Номер разряда	Условное обозначение прерывания	Название прерывания
31:24	-	Не используется
23	I2C	Прерывание от контроллера шины I2C
22	IT1	Прерывание от таймера IT1
21	IT0	Прерывание от таймера IT0
20	WDT	Прерывание от таймера WDT
19	VPOUT_CH	Прерывание от канала DMA VPOUT_CH по передаче массива данных
18	VPOUT	Прерывание от контроллера VPOUT
17	VPIN_CH	Прерывание от канала DMA VPIN_CH по приему массива данных
16	VPIN	Прерывание от контроллера VPIN
15	EMAC_TX_CH	Прерывание от DMA контроллера Ethernet MAC по завершению передачи данных
14	EMAC_RX_CH	Прерывание от DMA контроллера Ethernet MAC по завершению приема данных
13	EMAC_TX_FRAME	Прерывание от контроллера Ethernet MAC по завершению попытки передачи пакета
12	EMAC_RX_FRAME	Прерывание от контроллера Ethernet MAC по приему кадра или по переполнению входного FIFO
11:7	-	Не используется
6	PMSC_MASTER	Прерывание от PMSC в режиме Master
5	PMSC_MBR	Прерывание от PMSC при записи в почтовый ящик
4	UART	Прерывание от UART
3	IRQ3	Внешнее прерывание nIRQ[3]
2	IRQ2	Внешнее прерывание nIRQ[2]
1	IRQ1	Внешнее прерывание nIRQ[1]
0	IRQ0	Внешнее прерывание nIRQ[0]



**Таблица 4.4 Формат регистра QSTR1**

Номер разряда	Условное обозначение прерывания	Название прерывания
31:16	-	Не используется
15	MEM_CH17	Прерывание от канала DMA MEM_CH17
	...	
8	MEM_CH10	Прерывание от канала DMA MEM_CH10
7	MEM_CH07	Прерывание от канала DMA MEM_CH07
	...	
0	MEM_CH00	Прерывание от канала DMA MEM_CH00

**Таблица 4.5 Формат регистра QSTR2**

Номер разряда	Условное обозначение прерывания	Название прерывания
31:16	-	Не используется
15	SRIO1_MCE_DEC	В SRIO1 поступил символ Multicast-Event. Повторяет состояние бита MCE_DEC регистра LPU_CSR
14	SRIO1_RESET_DEVICE_CMD	В SRIO1 поступили 4 команды Reset-Device Command. Повторяет состояние бита RESET_DEVICE_CMD регистра LPU_CSR
13	SRIO1_PORT_ERROR	LPU SRIO1 находится в нерабочем состоянии из-за обнаружения невозстанавливаемой ошибки. Повторяет состояние бита PORT_ERROR регистра ERROR_STATUS_CSR
12	SRIO1_MPU_TX	Прерывание от MPU_TX SRIO1
11	SRIO1_MPU_RX	Прерывание от MPU_RX SRIO1
10	SRIO1_LSU	Прерывание от LSU SRIO1
9	SRIO1_DOORBELL	В SRIO1 поступил пакет типа DOORBELL
8	SRIO1_PWRITE	В SRIO1 поступил пакет типа PORT_WRITE
7	SRIO0_MCE_DEC	В SRIO0 принял символ Multicast-Event. Повторяет состояние бита MCE_DEC регистра LPU_CSR
6	SRIO0_RESET_DEVICE_CMD	В SRIO0 поступили 4 команды Reset-Device Command. Повторяет состояние бита RESET_DEVICE_CMD регистра LPU_CSR
5	SRIO0_PORT_ERROR	LPU SRIO0 находится в нерабочем состоянии из-за обнаружения невозстанавливаемой ошибки. Повторяет состояние бита PORT_ERROR регистра ERROR_STATUS_CSR
4	SRIO0_MPU_TX	Прерывание от MPU_TX SRIO0
3	SRIO0_MPU_RX	Прерывание от MPU_RX SRIO0
2	SRIO0_LSU	Прерывание от LSU SRIO0
1	SRIO0_DOORBELL	В SRIO0 поступил пакет типа DOORBELL
0	SRIO0_PWRITE	В SRIO0 поступил пакет типа PORT_WRITE

**Таблица 4.6 Формат регистра QSTR3**

Номер разряда	Условное обозначение прерывания	Название прерывания
31	SW_TX_DAT_CH1	Прерывание от канала DMA SW_TX_DAT_CH1
30	SW_TX_DES_CH1	Прерывание от канала DMA SW_TX_DES_CH1
29	SW_RX_DAT_CH1	Прерывание от канала DMA SW_RX_DAT_CH1
28	SW_RX_DES_CH1	Прерывание от канала DMA SW_RX_DES_CH1
27	-	Не используется
26	SW_TIME1	Прерывание от SWIC1 при получении маркер времени или распределенного прерывания
25	SW_ERR1	Прерывание от SWIC1 при ошибке в канале
24	SW_LINK1	Прерывание от SWIC1 при установлении соединения
23	SW_TX_DAT_CH0	Прерывание от канала DMA SW_TX_DAT_CH0
22	SW_TX_DES_CH0	Прерывание от канала DMA SW_TX_DES_CH0
21	SW_RX_DAT_CH0	Прерывание от канала DMA SW_RX_DAT_CH0
20	SW_RX_DES_CH0	Прерывание от канала DMA SW_RX_DES_CH0
19	-	Не используется
18	SW_TIME0	Прерывание от SWIC0 при получении маркер времени или распределенного прерывания
17	SW_ERR0	Прерывание от SWIC0 при ошибке в канале
16	SW_LINK0	Прерывание от SWIC0 при установлении соединения
15:14	-	Не используется
13	MFBSR_RX_CH1	Прерывание от канала DMA порта MFBSR1 при приеме данных
12	MFBSR_TX_CH1	Прерывание от канала DMA порта MFBSR1 при передаче данных
11	-	Не используется
10	MFBSR_RXBUF1	Формируется, если порт MFBSR1 включен на прием данных (в одном из режимов), а число 64-х разрядных слов в буфере приёма больше чем RLEV (RLEV устанавливается в регистре состояния приёмника RSR)
9	MFBSR_TXBUF1	Формируется, если порт MFBSR1 включен на передачу данных (в одном из режимов), а число 64-х разрядных слов, находящихся в буфере передачи меньше, либо равно TLEV (TLEV устанавливается в регистре состояния передатчика TSR)
8	SRQ1	Запрос обслуживания от порта MFBSR1. Формируется, если порт выключен (LEN=0, SPI_I2S_EN=0), а на выводах LACK или LCLK присутствует сигнал высокого уровня
7:6	-	Не используется
5	MFBSR_RX_CH0	Прерывание от канала DMA порта MFBSR0 при приеме данных
4	MFBSR_TX_CH0	Прерывание от канала DMA порта MFBSR0 при передаче данных
3	-	Не используется
2	MFBSR_RXBUF0	Формируется, если порт MFBSR0 включен на прием данных (в одном из режимов), а число 64-х разрядных слов в буфере приёма больше чем RLEV (RLEV устанавливается в регистре состояния приёмника RSR)
1	MFBSR_TXBUF0	Формируется, если порт MFBSR0 включен на передачу данных (в одном из режимов), а число 64-х разрядных слов, находящихся в буфере передачи меньше, либо равно TLEV (TLEV устанавливается в регистре состояния передатчика TSR)
0	SRQ0	Запрос обслуживания от порта MFBSR0. Формируется, если порт выключен (LEN=0, SPI_I2S_EN=0), а на выводах LACK или LCLK присутствует сигнал высокого уровня

Регистры запросов прерывания от DSP и их регистры маски находятся в адресном пространстве DSP.

Для управления режимом приема внешних прерываний nIRQ[3:0] имеется регистр IRQM, формат которого приведен в таблице 4.7.

**Таблица 4.7. Формат регистра IRQM**

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:12	-	Резерв	-	0
11:8	IRQ_MODE	Режим приема внешних прерываний nIRQ[3:0]: 0 - потенциальные сигналы, активный низкий уровень; 1 – прерывание формируется при переходе состояния входного сигнала с высокого уровня на низкий уровень. Прерывание запоминается на регистре. Регистр обнуляется при помощи разрядов IRQ_NULL	R/W	0
7:4	-	Резерв	-	0
3:0	IRQ_NULL	Обнуление запомненных прерываний при IRQ_MODE = 1. Прерывания nIRQ[3:0] обнуляются при записи 1 в разряды [3:0] соответственно.	RW1C	0

### 4.3 Системные регистры

Формат регистра управления и состояния CSR приведен в Таблица 4.7.

Таблица 4.7

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:28	-	Не используется	-	0
27	SRIO_LBE	Режим работы PMA SRIO: 0 – нормальный режим; 1 – режим петли. Используется для тестирования PMA.	RW	0
26:15	-	Не используется	-	0
14	FLUSH_D	При записи 1 в данный разряд кэш данных CPU останавливается в исходное состояние, то есть ее содержимое девалидируется. Эта процедура может использоваться для обеспечения когерентности кэш при работе DMA.	W	0
13	-	Не используется	-	0
12	FLUSH_I	При записи 1 в данный разряд кэш команд CPU останавливается в исходное состояние, то есть ее содержимое девалидируется. Эта процедура может использоваться для обеспечения когерентности кэш при работе DMA.	W	0
11	TST_CACHE	Режим работы кэш программ и кэш данных: 0 – нормальный режим; 1 – режим тестирования. Используется только при технологическом тестировании кэш программ. Пользователям устанавливать этот режим запрещено	R/W	0
10:2	-	Не используется	-	0
1	TR_CRAM	Режим размещения векторов прерываний при BEV = 0 (регистр Status CPU): 0 – вектора прерываний размещаются во внешней памяти (базовый адрес 0x80000000); 1 – вектора прерываний размещаются во внутренней памяти CRAM (базовый адрес 0xB8000000)	RW	0
0	FM	Режим преобразования виртуальных адресов CPU в физические адреса: 0 – с использованием TLB; 1 – Fixed Mapped (FM).	RW	1

## 4.4 Процедура начальной загрузки

По сигналу nRST (низкий уровень) все устройства микросхемы устанавливаются в исходное состояние. После его снятия (высокий уровень), дальнейшие действия определяются состоянием сигналов на входах микросхемы PBOOT и BOOT[1:0]. Устанавливать BOOT[1:0]=11 при PBOOT=1 запрещено.

При BOOT[1:0]=11 начальная загрузка выполняется из памяти Flash, подключенной к шине SPI MFBSP0. При этом в память CRAM из Flash загружается 64 32-разрядных слова, начиная с адреса 0x1800\_0000. Затем CPU стартует с этого адреса.

При PBOOT=1 после снятия сигнала nRST CPU находится в состоянии останова. Запуск CPU осуществляется после записи стартового адреса в регистр QSTR\_PCI командой Memory Write. После запуска CPU начинает выполнять стартовую программу с адреса из регистра QSTR\_PCI. Предварительно, из шины PCI должны быть загружены необходимые программы и данные во внутреннюю память и/или в блок памяти, подключенный к выводу nCS[3] (или nCS[4]). После старта CPU регистр QSTR\_PCI аппаратно обнуляется.

Следует отметить, что если микросхема 1892BM7Я используется в плате в составе персонального компьютера, обязательно должен быть установлен режим PBOOT=1, поскольку при PBOOT=0 после снятия сигнала nRST частота контроллера PMSC отключена и он не опознается на шине PCI при выполнении процедуры PLUG&PLAY.

Программа начальной загрузки должна обеспечивать конфигурирование всех устройств микросхемы.

## 4.5 Логика взаимодействия CPU и DSP

### 4.5.1 Функции CPU

CPU является ведущим. Он имеет свою операционную систему (планировщик или монитор) и выполняет основную программу.

CPU имеет доступ к следующим ресурсам DSP:

- памяти данных;
- памяти программ;
- архитектурным регистрам.

Обмен данными с этими ресурсами выполняется по командам Load, Store. Память DSP и его регистры для CPU являются словными, то есть состояние двух младших разрядов адреса является безразличным.

При штатной, работе доступ к архитектурным регистрам DSP, как правило, не используется, а применяется только для его диагностики или для отладки программного обеспечения.

DSP выдает следующие прерывания в CPU, которые поступают на регистр QSTR:

- программное;
- по переполнению стека;
- при выполнении команды STOP;

- при достижении адреса останова при исполнении программы до адреса останова или завершении требуемого числа шагов при пошаговом исполнении программы.

CPU в DSP прерываний не формирует.

CPU управляет работой DSP посредством передачи ему задания (макрокоманды) и его запуска (перевод из режима STOP в режим RUN). Данная процедура выполняется в следующей последовательности:

- CPU передает в память DSP данные и параметры их обработки. Эта операция может отсутствовать;
- CPU передает в программную память DSP программный код, который должен быть выполнен. Эта операция может отсутствовать;
- CPU передает в DSP адрес первой выполняемой команды посредством записи в программный счетчик. Эта операция может отсутствовать, например, если следующая макрокоманда DSP должна выполняться с его текущего состояния;
- CPU переводит DSP в состояние RUN посредством записи в его регистр управления и состояния DCSR.

#### 4.5.2 Функции DSP

DSP является ведомым. Он работает под управлением CPU и выполняет его макрокоманды (задания). Операционной системы и какого-либо монитора не имеет.

Для управления его работы DSP имеет программно доступный регистр управления и состояния DCSR. Формат этого регистра приведен в главе 3.

DSP может находиться в состояниях STOP или RUN и работает в старте стоповом режиме. То есть, после выполнения очередного задания CPU он останавливается и переходит в режим STOP посредством выполнения одноименной команды. DSP из состояния STOP в состояние RUN может перейти:

- по команде CPU;
- по сигналам от каналов DMA MEM\_CH.

DSP может выполнить запуск работы каналов DMA MEM\_CH посредством записи 1 в соответствующие разряды регистра DCSR.

## 5. ИНТЕРВАЛЬНЫЙ ТАЙМЕР

### 5.1 Назначение

Интервальный таймер (ИТ) предназначен для выработки периодических прерываний на основе деления тактовой частоты CPU либо внешней тактовой частоты – ХТІ или RTCХТІ. Основные характеристики таймера:

- Число разрядов делителя – 32;
- Число разрядов предделителя – 8;
- Программное управление стартом и остановкой таймера;
- Доступ ко всем регистрам обеспечивается в любой момент времени.

В 1892ВМ7Я имеется два интервальных таймера ИТ0, ИТ1.

### 5.2 Структурная схема ИТ

Структурная схема ИТ представлена на рисунке 5.1.

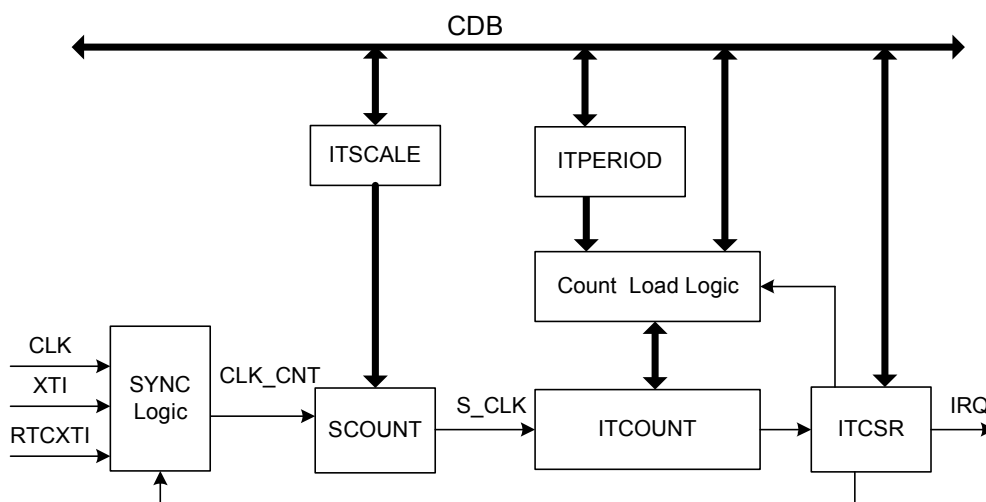


Рисунок 5.1. Структурная схема ИТ

В состав таймера входят следующие основные узлы:

- ИТCSR - регистр управления и состояния;
- ИТСАУНТ - счетчик основного делителя;
- ИТПЕРІОД - регистр периода основного делителя;
- ИТСАЛЕ - регистр предделителя;
- SСАУНТ – счетчик предделителя;
- SYNC Logic – логика синхронизации частот;
- Count Load Logic - логика загрузки счетчика основного делителя;

На структурной схеме интервального таймера использованы следующие обозначения:

- CDB – шина данных CPU;
- CLK – тактовая частота работы CPU;
- ХТІ – внешняя тактовая частота;
- RTCХТІ – внешняя тактовая частота;
- CLK\_CNT – выходная частота логики синхронизации;
- S\_CLK – выходная частота предделителя;
- IRQ – запрос на прерывание от таймера реального времени.

На вход интервального таймера поступает тактовая частота CPU и внешние тактовые частоты: XTI, RTCXTI. Для правильной работы таймера должны выполняться соотношения:

$f_{XTI} \leq \frac{f_{CLK}}{4}$ ,  $f_{RTCXTI} \leq \frac{f_{CLK}}{4}$ , где  $f_{XTI}$ ,  $f_{RTCXTI}$  и  $f_{CLK}$  значения частот XTI, RTCXTI и CLK соответственно. Как правило, RTCXTI имеет частоту 32,768 кГц.

### 5.3 Описание регистров интервального таймера

В таблице 5.1. приведен перечень программно-доступных регистров ИТ.

Таблица 5.1. Перечень регистров ИТ

Условное обозначение регистра	Название регистра	Тип доступа	Исходное состояние
ITCSR[4:0]	Регистр управления и состояния	W/R	0
ITPERIOD[31:0]	Регистр периода	W/R	FFFF_FFFF
ITCOUNT[31:0]	Регистр счетчика основного делителя частоты	W/R	0000_0000
ITSCALE[7:0]	Регистр предделителя частоты	W/R	0000

Формат регистра ITCSR приведен в таблице 5.2.

Таблица 5.2. Формат регистра ITCSR

Номер разряда	Условное обозначение	Описание
0	EN	Разрешение работы таймера: 0 – запрещение работы (неактивное состояние таймера); 1 – разрешение работы (активное состояние таймера).
1	INT	Признак срабатывания таймера. Состояние данного разряда транслируется в биты IT0 или IT1 регистра QSTR0. Сбрасывается при записи нуля в этот разряд
2	TICK	Бит тестирования регистра счетчика ITCOUNT и регистра предделителя ITSCALE. При записи 1 в бит TICK декрементируется значение счетчика ITCOUNT и предделителя ITSCALE. Поле доступно только по записи.
4:3	CLK_SEL	Задаёт тактовую частоту от которой работает ИТ: 00 – CLK – тактовая частота CPU; 01 – XTI – внешняя тактовая частота; 10 – RTCXTI – внешняя тактовая частота;

8-разрядный регистр ITSCALE используется для задания коэффициента предделения тактовой частоты CLK\_CNT, которая поступает на вход счетчика SCOUNT.

32-разрядные регистр ITPERIOD используется для задания периода работы основного делителя.

32-разрядный счетчик основного делителя частоты ITCOUNT работает в режиме декремента. На вход этого счетчика поступает частота (S\_CLK) с выхода счетчика предделителя.

Если ITPERIOD = 0000\_7FFF, ITSCALE = 0000, при этом в регистре ITCSR задана работа от внешней частоты RTCXTI (ITCSR[4:3]=10), а частота RTCXTI = 32,768 кГц, то интервальный таймер формирует прерывание каждую секунду.



## 5.4 Программирование IT

Перед началом работы с таймером необходимо задать источник тактовой частоты в регистре `ITCSR[4:3]=CLK_SEL`. Затем необходимо загрузить значение периода в регистр `ITPERIOD` и значение коэффициента предделения частоты в регистр `ITSCALE`.

Для активизации таймера необходимо в бит `EN` регистра `ITCSR` записать 1. В момент этой записи содержимое регистров `ITSCALE` и `ITPERIOD` переписывается в счетчики `SCOUNT` и `ITCOUNT` соответственно. После этого оба счетчика начинают работать в режиме декремента. При этом предделитель работает от частоты `CLK_CNT`, а счетчик `ITCOUNT` – от частоты `S_CLK`, формируемой предделителем.

Когда оба счетчика `SCOUNT` и `ITCOUNT` достигают нулевого состояния, в регистре `ITCSR` устанавливается бит `INT` и формируется запрос на прерывание, а содержимое регистров `ITSCALE` и `ITPERIOD` снова переписывается в счетчики `SCOUNT` и `ITCOUNT` соответственно. Далее таймер работает аналогичным образом.

Запрос на прерывание формируется каждые  $\{(irtperiod + 1) * (irtscale + 1)\}$  тактов `CLK_CNT`, где `irtperiod` и `irtscale` – содержимое регистров `ITPERIOD` и `ITSCALE` соответственно.

При необходимости, в любой момент времени в регистры `ITCOUNT` и `ITPERIOD` можно произвести запись новых данных и тем самым изменить значение обрабатываемого временного интервала.

## 6. СТОРОЖЕВОЙ ТАЙМЕР

### 6.1 Назначение

Сторожевой таймер (WDT) предназначен для:

- вывода системы из зависания, если программное обеспечение зациклилось и не формирует соответствующих управляющих воздействий;
- выработки прерываний на основе деления тактовой частоты CPU.

Основные характеристики таймера:

- число разрядов основного делителя – 32;
- число разрядов предделителя – 8;
- программное управление стартом и остановкой таймера;
- два режима работы: режим сторожевого таймера (WDM) и режим интервального таймера (ITM);
- два режима отработки временных интервалов: однократный и периодический;
- доступ ко всем регистрам обеспечивается в любой момент времени.

### 6.2 Структурная схема

Структурная схема сторожевого таймера приведена на Рисунок 6.1.

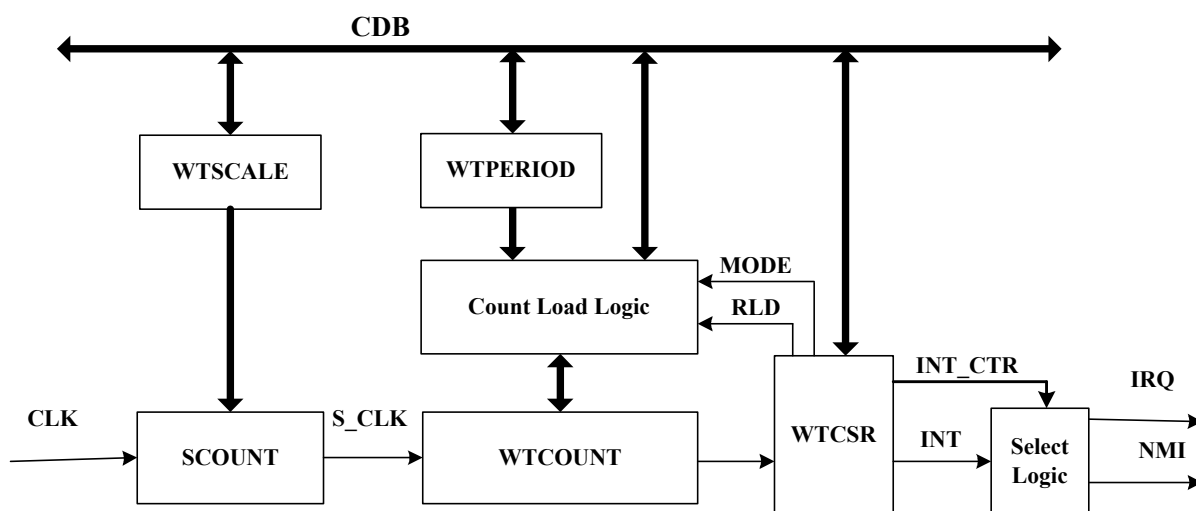


Рисунок 6.1. Структурная схема сторожевого таймера.

В состав сторожевого таймера входят следующие основные узлы:

- WTCSR - регистр управления и состояния;
- WTCOUNT - счетчик основного делителя;
- WTPERIOD - регистр периода основного делителя;
- WTSCALE - регистр предделителя;
- SCOUNT – счетчик предделителя;
- Count Load Logic - логика загрузки счетчика основного делителя.

На структурной схеме интервального таймера использованы следующие обозначения:

- CDB – шина данных CPU;
- CLK – тактовая частота работы CPU;
- S\_CLK – выходная частота делителя;
- IRQ – запрос на прерывание от интервального таймера;
- NMI – немаскируемое прерывание.

### 6.3 Описание регистров WDT

В Таблица 6.1 приведен перечень программно-доступных регистров WDT.

Таблица 6.1. Перечень программно-доступных регистров WDT

Условное обозначение регистра	Название регистра	Тип доступа	Исходное состояние
WTCSR[14:0]	Регистр управления и состояния	W/R	0000
WTPERIOD[31:0]	Регистр периода	W/R – в неактивном состоянии; R – в активном состоянии.	FFFF_FFFF
WTCOUNT[31:0]	Регистр счетчика основного делителя частоты	W/R – в неактивном состоянии; R – в активном состоянии.	0000_0000
WTSCALE[15:0]	Регистр делителя частоты	W/R – в неактивном состоянии; R – в активном состоянии.	0000

8-разрядный регистр WTSCALE используется для задания коэффициента деления тактовой частоты CPU (CLK), которая поступает на вход счетчика SCOUNT.

32-разрядные регистр WTPERIOD используется для задания периода работы основного делителя.

32-разрядный счетчик основного делителя частоты WTCOUNT работает в режиме декремента. На вход этого счетчика поступает частота S\_CLK с выхода счетчика делителя.

Формат регистра WTCSR приведен в Таблица 6.2.

Таблица 6.2. Формат регистра WTCSR.

Номер разряда	Условное обозначение	Описание
7: 0	KEY	Поле для записи ключей. Запись в это поле последовательности кодов A0 (ключ KEY1) и F5 (ключ KEY2) приводит к переключению таймера из режима сторожевого таймера (WDM) в режим интервального таймера (ITM). Поле доступно по чтению и записи. Поле доступно по записи только в режиме WDM: когда EN=1 или когда таймер находится в состоянии Timeout. Сбрасывается в ноль при переводе таймера из режима ITM в режим WDM. Значение в исходном состоянии – 0.
8	EN	Разрешение работы таймера: 0 – запрещение работы (неактивное состояние таймера); 1 – разрешение работы (активное состояние таймера). Доступен по чтению и записи. Запись нуля в этот бит при работе таймера в режиме WDM не имеет эффекта. Значение в исходном состоянии – 0.
9	INT	Признак срабатывания таймера. В зависимости от содержимого поля INT_CTR состояние данного разряда транслируется или в бит WDT регистра QSTR0, или в немаскируемое прерывание (NMI). Сбрасывается при записи нуля в этот разряд, а также при переводе таймера из режима ITM в режим WDM. Доступен по чтению и записи в режиме ITM и только по чтению в режиме WDM. Значение в исходном состоянии – 0.
10	MODE	Режим работы таймера: 0 – режим сторожевого таймера (WDM); 1 – режим обычного таймера (ITM). Доступен по чтению и записи при EN=0 и только по чтению при EN=1. Значение в исходном состоянии – 0.
11	RLD	Бит управления перезагрузкой SCOUNT и WTCOUNT при работе в режиме ITM: 0 – таймер однократно обрабатывает временной интервал и останавливается; 1 – таймер обрабатывает заданный временной интервал периодически. После обработки очередного временного интервала содержимое WTSCALE и WTPERIOD загружается в SCOUNT и WTCOUNT соответственно. Доступен по чтению и записи при EN=0 и только по чтению при EN=1. Значение в исходном состоянии – 0.
13: 12	INT_CTR	Управления типом прерывания, которое формируется таймером WDT: 00, 11 – прерывание не формируется; 01 – обычное прерывание (QSTR[29]). Как правило, используется в режиме ITM; 10 – немаскируемое прерывание (NMI). Как правило, используется в режиме WDM. Поле доступно по чтению и записи при EN=0 и только по чтению при EN=1. Значение в исходном состоянии – 0.

## 6.4 Программирование WDT

Диаграмма состояний WDT приведена на рис 7.2.

В исходном состоянии WDT находится в режиме сторожевого таймера. Для перевода его в режим интервального таймера необходимо записать 1 в бит MODE регистра WTCSR. Следует отметить, что смена режима работы таймера посредством записи в бит MODE возможна, если таймер не активен (EN=0).

Перед началом работы с таймером WDT необходимо загрузить значение периода в регистр WTPERIOD и значение коэффициента предделения частоты в регистр WTSCALE.

Для активизации таймера необходимо в бит EN регистра WTCSR записать 1. В момент этой записи содержимое регистров WTSCALE и WTPERIOD переписывается в счетчики SCOUNT и WTCOUNT соответственно. После этого оба счетчика начинают работать в режиме декремента. При этом предделитель работает от частоты CLK, а счетчик WTCOUNT – от частоты S\_CLK, формируемой предделителем.

После активизации таймера, WTCOUNT, WTPERIOD, WTSCALE, а также поля INT\_CTR, MODE, RLD регистра WTCSR, становятся не доступными по записи.

Сторожевой таймер в режиме WDM необходимо периодически обслуживать. То есть, если он был активизирован в режиме WDM, то для того, чтобы не возникло состояния Timeout необходимо периодически выполнять следующую последовательность действий:

- переключить таймер из режима WDM в режим ITM посредством последовательной записи в поле KEY регистра WTCSR кодов A0 и F5;
- остановить таймер посредством записи 0 в бит EN регистра WTCSR;
- установить MODE=0;

Если вслед за значением A0 в поле KEY будет записано значение  $\neq$  F5, то таймер перейдет в состояние Timeout.

Если после активизации таймера в режиме WDM, он не будет переведен в режим ITM, то, когда оба счетчика SCOUNT и WTCOUNT достигнут нулевого значения, таймер перейдет в состояние Timeout.

В состоянии Timeout таймер формирует признак INT и останавливается, а запись в какой-либо из его регистров блокируется. Для вывода WDT из состояния Timeout необходимо его переключить в режим ITM посредством последовательной записи в поле KEY регистра WTCSR кодов A0 и F5.

При переключении таймера из неактивного состояния в режиме ITM в режим WDM путем записи 0 в поле MODE регистра WTCSR происходит обнуление полей KEY и INT.

При работе таймера в режиме ITM при RLD=0 он однократно обрабатывает заданный временной интервал, устанавливает INT=1 и останавливается (когда оба счетчика SCOUNT и WTCOUNT достигают нулевого состояния). Если RLD=1, то каждый раз после достижения счетчиками нулевого состояния и установки INT=1, происходит перезагрузка значений периода и коэффициента предделения частоты. То есть, таймер обрабатывает заданный временной интервал периодически до тех пор, пока он не будет остановлен.

Запрос на прерывание формируется каждые  $\{(wtperiod + 1) * (wtscale + 1)\}$  тактов работы CPU, где wtperiod и wtscale – содержимое регистров WTPERIOD и WTSCALE соответственно.

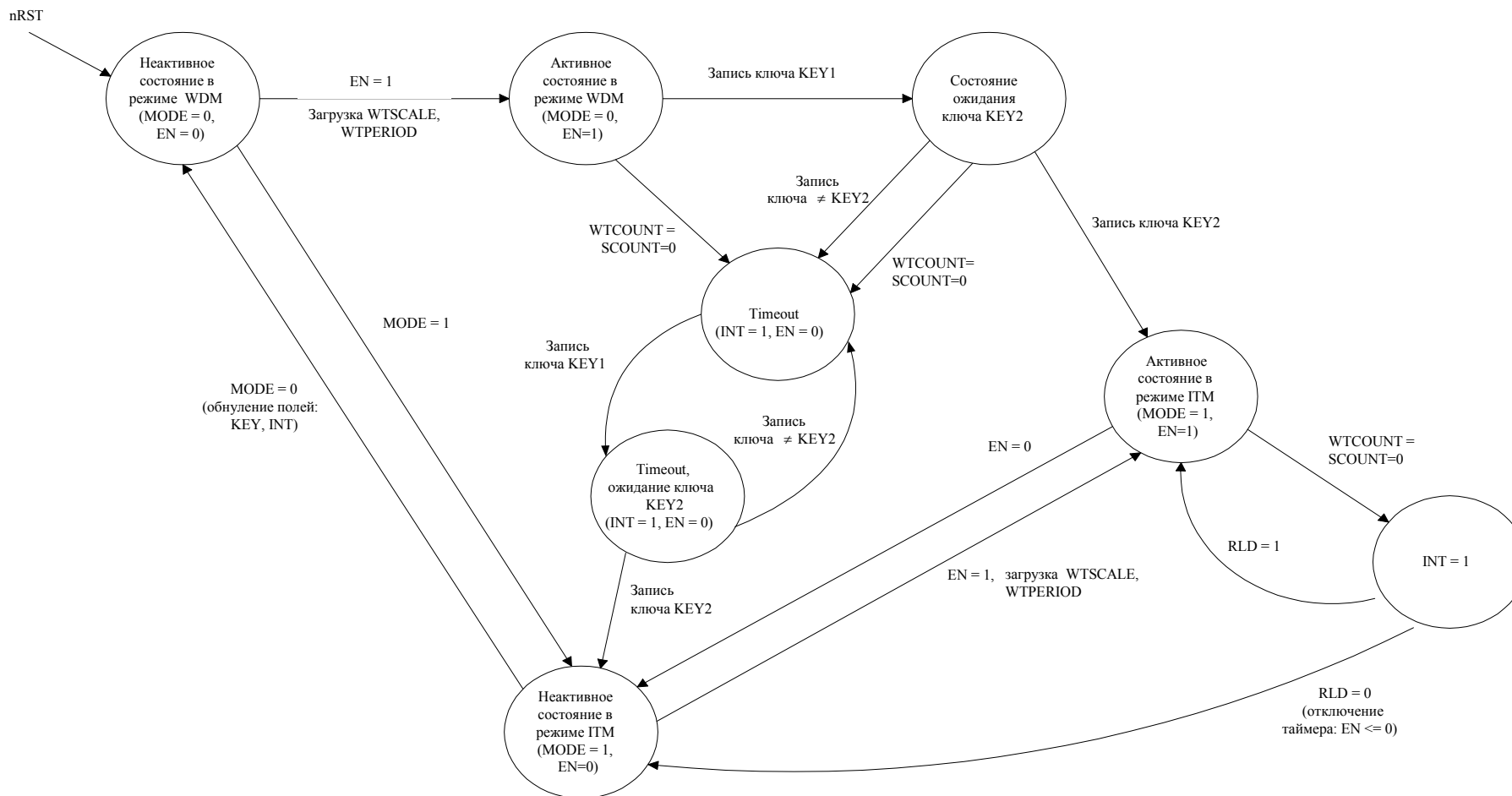


Рисунок 6.2. Диаграмма состояний WDT.

## 7. КОНТРОЛЛЕР ПРЯМОГО ДОСТУПА В ПАМЯТЬ (DMA)

### 7.1 Перечень каналов DMA

Контроллер DMA микросхемы 1892BM7Я имеет 40 каналов. Перечень каналов приведен в Таблица 7.1.

Таблица 7.1. Перечень каналов DMA

Условное обозначение канала	Назначение канала
VPIN_CH	Прием данных из контроллера VPIN в память
VPOUT_CH	Передача данных из памяти в контроллер VPOUT
SWIC_TX_DAT_CH0	Передача данных из памяти в SWIC0
SWIC_TX_DES_CH0	Передача дескрипторов из памяти в SWIC0
SWIC_RX_DAT_CH0	Передача данных из SWIC0 в память
SWIC_RX_DES_CH0	Передача дескрипторов из SWIC0 в память
SWIC_TX_DAT_CH1	Передача данных из памяти в SWIC0
SWIC_TX_DES_CH1	Передача дескрипторов из памяти в SWIC0
SWIC_RX_DAT_CH1	Передача данных из SWIC0 в память
SWIC_RX_DES_CH1	Передача дескрипторов из SWIC0 в память
EMAC_TX_CH	Передача данных из памяти в контроллер Ethernet
EMAC_RX_CH	Прием данных из контроллера Ethernet в память
MFBSP_RX_CH1	Прием данных из MFBSP1 в память
MFBSP_TX_CH1	Передача данных из памяти в MFBSP1
MFBSP_RX_CH0	Прием данных из MFBSP0 в память
MFBSP_TX_CH0	Передача данных из памяти в MFBSP0
MEM_CH10 – MEM_CH17	Обмен данными типа память-память
MEM_CH00 – MEM_CH07	Обмен данными типа память-память

Памятью могут быть CRAM, блоки памяти сопроцессоров DSP: XRAM, YRAM и PRAM, внешняя память, доступная через порты MPORT, DDR\_PORT0, DDR\_PORT1.

Если при работе DMA изменяется программный код в памяти, то когерентность кэш программ CPU (ICACHE) аппаратно не обеспечивается. В этом случае для обеспечения когерентности используется бит FLUSH в системном регистре CSR.

### 7.2 Организация обмена данными в микросхеме

Для передачи данных в микросхеме 1892BM7Я имеются: шина CDB (CPU Data Bus) и коммутатор AXI Switch (см. Рисунок 1.1).

CPU без конфликтов с DMA обменивается данными с памятью CRAM, с системными регистрами (CSR, MASKR, QSTR и т.д.), а также с регистрами устройств IT0, IT1, WDT, DSP, MPORT, DDR\_PORT0, DDR\_PORT1, MFBSP1, MFBSP0, EMAC, VPIN, VPOUT, I2C, UART, SWIC.

Коммутатор обеспечивает передачу данных между любым исполнительным устройством (Slave) и любым задатчиком (Master). Исполнительными устройствами являются блоки внутренней памяти (CRAM, память DSP) или любая внешняя память, доступная через MPORT и DDR\_PORT0, DDR\_PORT1. Задатчиками могут быть CPU, DSP, каналы DMA.

Процесс передачи данных между любыми парами Slave ↔ Master выполняется параллельно и без конфликтов. Конфликт между задатчиками возникает, если они через коммутатор пытаются обменяться данными с одним и тем же исполнительным устройством.

### 7.3 Каналы DMA типа память - память

В микросхеме 1892ВМ7Я имеется два DMA по 8 каналов MEM\_CH каждый, которые обеспечивают обмен данными между двумя областями любых блоков памяти (внутренней или внешней).

Для управления работой каждого канала MEM\_CH имеются следующие регистры:

- \* регистр управления и состояния – CSR\_MEM\_CH;
- \* регистры индекса (физический адрес памяти) - IR0, IR1;
- \* регистры смещения - OR, Y;
- \* регистр начального физического адреса блока параметров DMA передачи для выполнения процедуры самоинициализации (CP);
- \* псевдорегистр управления состоянием бита RUN регистра CSR (RUN\_MEM\_CH).

Исходное состояние регистров CSR\_MEM\_CH: разряды [15:0] – нули, а состояние разрядов 31:16 не определено. Исходное состояние остальных регистров не определено.

Формат регистров CSR\_MEM\_CH этих каналов приведен в Таблица 7.2.



**Таблица 7.2. Формат регистра управления и состояния каналов MEM\_CN**

Номер разряда	Условное Обозначение	Назначение
0	RUN	Состояние работы канала DMA: 0 – состояние останова; 1 – состояние обмена данными. Устанавливается в 1 при записи 1 в этот разряд. Устанавливается в 0: при записи 0 в этот разряд и после окончания передачи данных, оставшихся в канале; при завершении передачи блока данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
1	DIR	Направление обмена данными: 0 – память по IR0 => память по IR1; 1 – память по IR1 => память по IR0.
5:2	WN	Пакет данных, который передается по коммутатору AXI Switch за одно предоставление прямого доступа: 0 – 1 слово; F – 16 слов. Посредством этого параметра можно плавно изменять приоритет каналов DMA относительно CPU, DSP и относительно друг друга
6	EN64	Формат передаваемых данных по коммутатору AXI Switch: 0 – 32 разряда; 1 – 64 разряда. При передаче 32-разрядными словами: WCX – число 32-разрядных слов; адрес в IR0, IR1 должен быть выровнен по границе 32-разрядного слова. При передаче 64-разрядными словами: WCX – число 64-разрядных слов; адрес в IR0, IR1 должен быть выровнен по границе 64-разрядного слова
7	START_DSP	Разрешение запуска работы DSP-ядра (перевод из состояния STOP в состояние RUN) после завершения передачи блока данных: 0 – запуск запрещен; 1 – запуск разрешен.
8	MODE	Режим модификации адреса регистра IR0 0 – линейный режим; 1 – режим с обратным переносом.
9	2D	Режим модификации адреса регистра IR1: 0 – одномерный режим; 1 – двухмерный режим.
10	MASK	Маска внешнего запроса прямого доступа nDMAR: 0 – запрос запрещен; 1 – запрос разрешен. Если разряд равен нулю, то канал работает только под управлением бита RUN. Если разряд равен 1, то для инициализации канала необходимо также наличие запроса nDMAR (низкий уровень).

Номер разряда	Условное Обозначение	Назначение
11	FLYBY	Признак выполнения обмена данными в режиме Flyby: 0 – обычный режим; 1 – режим Flyby. Обмен данными между внешней памятью и внешним устройством
12	CHEN	Разрешение выполнения очередной процедуры самоинициализации: 0 – выполнение очередной процедуры самоинициализации запрещено; 1 – выполнение очередной процедуры самоинициализации разрешено. Используется только при обмене цепочкой блоков данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
13	IM	Маска разрешение установки признака END: 0 – установки признака запрещено; 1 – установки признака разрешено. Используется только при обмене цепочкой блоков данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
14	END	Признак завершения передачи блока данных. Устанавливается в 1 при завершении передачи блока данных (при IM=1). Устанавливается в 0 при чтении содержимого этого регистра. Доступен по записи и чтению.
15	DONE	Признак завершения передачи блока данных. Устанавливается в 1 при завершении передачи блока данных при CHEN=0 (CHEN=1 может быть только при использовании процедуры самоинициализации). Устанавливается в 0 при чтении содержимого этого регистра
31:16	WCX	Число слов данных, которые должен передать канал DMA при одномерной адресации (блок данных). Число слов в строке при двухмерной адресации. Количество передаваемых слов = WCX + 1. Содержимое этого поля уменьшается на 1 после передачи каналом DMA очередного слова данных

Все разряды регистра CSR\_MEM\_CH доступны по записи и чтению.

Состоянием разряда 0 регистра CSR\_MEM\_CH можно управлять, используя адрес псевдорегистра RUN. При этом остальные разряды этого регистра не изменяются. Эта процедура может быть использована для временной приостановки канала DMA. При чтении по адресу псевдорегистра RUN считывается содержимое регистра CSR\_MEM\_CH без сброса битов END и DONE.

32-разрядные регистры индекса IR0, IR1 содержат начальные физические адреса источника и приемника данных (или, наоборот, в зависимости от содержимого разряда DIR регистра

CSR\_MEM\_CH) памяти микросхемы. В зависимости от содержимого разряда EN64 адреса в этих регистрах должны быть выровнены по границе 32 или 64-разрядного слова.

Формат регистра смещения OR приведен в Таблица 7.3.

Таблица 7.3. Формат регистра индекса и смещения каналов MEM\_CH

Номер разряда	Условное Обозначение	Назначение
15:0	OR0	Смещение (приращение) адреса для индексного регистра IR0 после передачи каждого слова данных
31:16	OR1	Смещение (приращение) адреса для индексного регистра IR1 после передачи каждого слова данных

Модификация индексного регистра IR0 при помощи смещения OR0 обеспечивается в режимах с прямым или обратным переносами. Режим с обратным переносом используется при реализации алгоритма быстрого преобразования Фурье (БПФ). Модификация индексного регистра IR1 при помощи смещения OR1 обеспечивается только в режиме с прямым переносом.

В режиме модификации индексного регистра с прямым переносом смещение, задаваемое полями OR0, OR1, рассматривается как число со знаком в диапазоне  $-32768$  до  $+32767$  слов данных (32 или 64-разрядных). Алгоритм модификации адреса с прямым переносом:

```
for ( x = 0; x < WCX; x++ ) { пересылка по адресу IR0;
                                модификация адреса для 64-х разрядного обмена: IR0 = IR0 +
                                {{13{OR0[15]}},OR0,000};
                                модификация адреса для 32-х разрядного обмена: IR0 = IR0 +
                                {{14{OR0[15]}},OR0,00};
                                пересылка по адресу IR1;
                                модификация адреса для 64-х разрядного обмена: IR1 = IR1 +
                                {{13{OR1[15]}},OR1,000};
                                модификация адреса для 32-х разрядного обмена: IR1 = IR1 +
                                {{14{OR1[15]}},OR1,00};
                                }
```

В режиме модификации индексного регистра с обратным переносом смещение, задаваемое полем OR0, имеет диапазон от 0 до +65535. Модификация адреса в этом случае выполняется с распространением переноса в обратном направлении – от старших разрядов к младшим. Операция модификации адреса с обратным переносом эквивалентна последовательному выполнению следующих процедур:

- 16-разрядное смещение OR0 дополняется до 32 разрядов: со стороны младших – двумя или тремя нулями, (для 32 или 64-разрядного обменов соответственно, а со стороны старших разрядов – четырнадцатью или тринадцатью нулями, для 32 и 64-разрядного обменов соответственно);
- изменение на обратный порядок следования разрядов в регистрах адреса и смещения. При этом старший бит становится младшим;
- модификация адреса посредством операции сложения с прямым переносом;
- восстановление первоначального порядка следования разрядов регистра адреса.

Модификацию адреса с обратным переносом можно описать при помощи следующих выражений:

$IR0 [0:31] = IR0[0:31] + \{000, OR0[0:15], 000000000000\}$  – для 64-разрядного обмена;

$IR0 [0:31] = IR0[0:31] + \{00, OR0[0:15], 0000000000000000\}$  – для 32-разрядного обмена.

Канала MEM\_CN обеспечивают передачу двумерных массивов (матриц  $W[m;n]$ ). При этом, память (внутренняя или внешняя) адресуется в двухмерном режиме. Для этого имеется 32-разрядный регистр Y, формат которого приведен в Таблица 7.4.

**Таблица 7.4. Формат регистра Y**

Номер разряда	Условное Обозначение	Назначение
15:0	OY	Смещение (приращение) адреса памяти в 32-разрядных словах по направлению Y. Используется только при двухмерной адресации.
31:16	WCY	Число строк по Y направлению. Используется только при двухмерной адресации. Количество передаваемых строк = WCY + 1.

При двухмерном режиме адресации поле WCX регистра CSR содержит число слов в строке (X направление), а поле WCY регистра Y содержит число строк (Y направление). Пересылка каждого слова данных осуществляется по индексному регистру IR1 с его последующей инкрементацией на величину, соответствующую содержимому поля OR1 регистра OR (X направление) или поля OY регистра Y. Двухмерная адресация выполняется следующим образом:

Содержимое счетчика WCX сохраняется в буферном регистре;

1 цикл. Индексный регистр внешней памяти модифицируется с использованием смещения OR1. Счетчик WCX декрементируется. Если он равен 0, то переход ко второму циклу.

2 цикл. Состояние счетчика WCX восстанавливается из буферного регистра. Индексный регистр внешней памяти модифицируется с использованием смещения OY. Счетчик WCY декрементируется. Если он не равен 0, то переход к первому циклу. Если он равен 0, то работа канала завершается.

Функционально двумерная адресация эквивалентна следующему двойному циклу (реализуется только по IR1, OR1):

```

for ( y = 0; y <= WCY; y++ ) {
    for ( x = 0; x < WCX; x++ ) { пересылка по адресу IR1
                                для 64-х разрядного обмена : IR1 =
    IR1 + {{13{OR1[15]}},OR1,3'h0};
                                для 32-х разрядного обмена : IR1 =
    IR1 + {{14{OR1[15]}},OR1,2'h0}
                                };
                                пересылка по адресу IR1
                                для 64-х разрядного обмена : IR1 = IR1 +
    {{13{ORY[15]}},ORY,3'h0};
                                для 32-х разрядного обмена : IR1 = IR1 +
    {{14{ORY[15]}},ORY,2'h0};
                                };
};

//общее кол-во пересылок (WCX=1)*(WCY+1)
    
```

Микросхема 1892ВМ7Я имеет 8 внешних сигналов запроса прямого доступа nDMAR[7:0]. Эти сигналы поступают на каналы DMA MEM\_CH следующим образом:

nDMAR[0] - на каналы MEM\_CH00, MEN\_CH10;  
 nDMAR[1] - на каналы MEM\_CH01, MEN\_CH11;  
 nDMAR[2] - на каналы MEM\_CH02, MEN\_CH12;  
 nDMAR[3] - на каналы MEM\_CH03, MEN\_CH13;  
 nDMAR[4] - на каналы MEM\_CH04, MEN\_CH14;  
 nDMAR[5] - на каналы MEM\_CH05, MEN\_CH15;  
 nDMAR[6] - на каналы MEM\_CH06, MEN\_CH16;  
 nDMAR[7] - на каналы MEM\_CH07, MEN\_CH17.

То есть, один сигнал запроса может запустить сразу два канала DMA MEM\_CH, если они настроены для работы в этом режиме.

Для настройки работы канала DMA MEM\_CH по внешним запросам необходимо в регистре CSR\_MEM\_CH установить: MASK=1, RUN=1. Внешнее устройство необходимо активизировать на формирование сигналов nDMAR, только после настройки соответствующего канала DMA MEM\_CH.

По каждому переходу сигнала nDMAR из 1 в 0 канал DMA MEM\_CH выполняет процедуру передачи одного пакета слов данных размером в соответствии с полем WN регистра CSR\_MEM\_CH.

Необходимо иметь в виду, что факт перехода сигнала nDMAR из 1 в 0 запоминается в DMA только при RUN=1, MASK=1. При выполнении любой операции записи в регистр CSR\_MEM\_CH, сбрасывается запомненный в DMA факт перехода сигнала nDMAR из 1 в 0, если он не был принят к исполнению к этому моменту.

Каналы DMA MEM\_CH совместно с портом MPORT обеспечивают передачу данных в режиме Flyby. При передаче данных в режиме Flyby шина данных микропроцессора переводится в 3 состояние, и одновременно активизируется внешняя память и внешнее устройство ввода-вывода. Память управляется как обычно, а устройство ввода-вывода – при помощи

сигналов nFLYBY (признак данного режима) и nOE (активизация выходных формирователей устройства ввода-вывода).

Для выполнения передачи данных в режиме Flyby в регистре CSR\_MEM\_CH необходимо установить бит FLYBY=1.

Микросхема 1892BM17Я имеет 4 пары внешних сигналов nFLYBY и nOE. Эти сигналы связаны с каналами DMA MEM\_CH следующим образом:

nFLYBY[0], nOE[0] - каналы MEM\_CH00, MEM\_CH04, MEM\_CH10, MEM\_CH14;

nFLYBY[0], nOE[0] - каналы MEM\_CH01, MEM\_CH05, MEM\_CH11, MEM\_CH15;

nFLYBY[0], nOE[0] - каналы MEM\_CH02, MEM\_CH06, MEM\_CH12, MEM\_CH16;

nFLYBY[0], nOE[0] - каналы MEM\_CH03, MEM\_CH07, MEM\_CH13, MEM\_CH17.

В случае если в каналах DMA, управляющих одним внешним устройством, одновременно установлен бит FLYBY=1, то очередность управления происходит согласно приоритету между каналами DMA.

## 7.4 Каналы DMA периферийных портов

Для обслуживания портов VPIN, VPOUT, EMAC, MFBSP, SWIC, USBIC имеются следующие каналы DMA:

VPIN\_CH, VPOUT\_CH,

EMAC\_TX\_CH, EMAC\_RX\_CH,

MFBSP\_RX\_CH1, MFBSP\_TX\_CH1, MFBSP\_RX\_CH0, MFBSP\_TX\_CH0,

SWIC\_TX\_DAT\_CH0, SWIC\_TX\_DES\_CH0, SWIC\_RX\_DAT\_CH0, SWIC\_RX\_DES\_CH0,

SWIC\_TX\_DAT\_CH1, SWIC\_TX\_DES\_CH1, SWIC\_RX\_DAT\_CH1, SWIC\_RX\_DES\_CH1.

Для управления работой каналы DMA портов содержат следующие регистры:

- \* регистр управления и состояния (CSR);
- \* регистр индекса (физический адрес памяти) (IR);
- \* регистр начального адреса блока параметров DMA передачи для сомоинициализации (CP);
- \* псевдорегистр управления состоянием бита RUN регистра CSR.

Исходное состояние регистров CSR: разряды 15:0 – нули, а состояние разрядов 31:16 не определено. Исходное состояние остальных регистров не определено.

Каналы DMA портов передают данные по коммутатору AXI Switch 64-разрядными словами.

32-разрядный индексный регистр IR содержат физический адрес внутренней или внешней памяти. После передачи каждого слова данных к индексу IR прибавляется смещение на одно 64-х разрядное слово.

Памятью могут быть CRAM, блоки памяти DSP: XRAM, YRAM и PRAM, внешняя память, доступная через MPORT, DDR\_PORT0, DDR\_PORT1.

Формат регистров управления и состояния CSR каналов DMA этих портов приведен в таблице 7.5.

**Таблица 7.5. Формат регистров управления и состояния DMA портов**

Номер разряда	Условное обозначение	Назначение
0	RUN	Состояние работы канала DMA: 0 – состояние останова; 1 – состояние обмена данными. Устанавливается в 1 при записи 1 в этот разряд.

Номер разряда	Условное обозначение	Назначение
		Устанавливается в 0: при завершении передачи блока данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
1	-	Не используется
5:2	WN	Число слов данных (пачка), которое передается за одно предоставление прямого доступа: 0 – 1 слово, F – 16 слов. Посредством этого параметра можно плавно изменять приоритет каналов DMA относительно других устройств и относительно друг друга
11:6	-	Не используется
12	CHEN	Разрешение выполнения очередной процедуры самоинициализации: 0 – выполнение очередной процедуры самоинициализации запрещено; 1 – выполнение очередной процедуры самоинициализации разрешено. Используется только при обмене цепочкой блоков данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
13	IM	Маска разрешение установки признака END: 0 – установки признака запрещено; 1 – установки признака разрешено. Используется только при обмене цепочкой блоков данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
14	END	Признак завершения передачи блока данных. Устанавливается в 1 при завершении передачи блока данных (при IM=1). Устанавливается в 0 при чтении содержимого этого регистра.
15	DONE	Признак завершения передачи блока данных. Устанавливается в 1 при завершении передачи блока данных при CHEN=0 (CHEN=1 может быть только при использовании процедуры самоинициализации). Устанавливается в 0 при чтении содержимого этого регистра
31:16	WCX	Кроме EMAC_CH: число 64-разрядных слов данных, которые должен передать канал DMA (блок данных); количество передаваемых слов: WCX + 1; содержимое этого поля уменьшается на 1 после передачи каналом DMA очередного слова данных. Для EMAC_CH: число байт данных, которое должен передать канал DMA (блок данных); количество передаваемых байт: WCX + 1; содержимое этого поля уменьшается на число переданных байт данных. Исходное состояние поля не определено.

Все разряды регистра CSR доступны по записи и чтению.



Бит RUN может быть использован для остановки работы канала DMA портов. Для этого в любой момент времени в него необходимо записать 0. Эта процедура возможна, если длина массива данных, указанного в канале DMA порта, равна длине массива данных, который порт передаст (например, MFBSP). Для продолжения работы в бит RUN необходимо записать 1.

Если порт прекратил обмен данными по внешней причине, то длина массива данных, указанного в канале DMA порта, будет не равна длине массива данных, который порт действительно передаст. В этом случае для остановки работы порта и его канала DMA необходимо использовать следующие алгоритмы.

Алгоритм остановки MFBSP и его канала DMA:

1. Остановить MFBSP, для чего в регистр CSR\_MFBSP необходимо записать 0.
2. Выполнить операцию записи 0 в бит RUN регистра CSR соответствующего канала DMA MFBSP (при этом, бит RUN может в 0 не установиться).
3. Установить в 1 бит RX\_DBG (TX\_DBG) регистра EMERG\_MFBSP.
4. Дождаться установки в 0 бита RUN регистра CSR соответствующего канала DMA MFBSP.
5. Установить в 0 бит RX\_DBG (TX\_DBG) регистра EMERG\_MFBSP.

Алгоритм остановки SWIC и его каналов DMA:

1. Выполнить операцию записи 0 в биты RUN регистров CSR каналов DMA SWIC (канал записи в память дескрипторов принимаемых пакетов, канал записи в память принимаемых слов данных, канал чтения из памяти дескрипторов передаваемых пакетов, канал чтения из памяти передаваемых слов данных).
2. Установить в регистре MODE\_CR SWIC в 1 биты Link\_disable (остановка работы SWIC) и RDY\_MODE.
3. Дождаться установки в 0 битов RUN регистров CSR каналов DMA SWIC.
4. Установить в регистре MODE\_CR SWIC в 0 бит RDY\_MODE.

Следует отметить, что при выполнении этого алгоритма «хвост» передаваемых данных из порта теряется, а в «хвосте» приемного буфера данные будут недостоверны.

Состоянием разряда 0 регистра CSR можно управлять, используя адрес псевдорегистра RUN. При этом остальные разряды этого регистра не изменяются. Эта процедура может быть использована для временной приостановки канала DMA. При чтении по адресу псевдорегистра RUN считывается содержимое регистра CSR без сброса битов END и DONE.

## 7.5 Процедура самоинициализации

Все каналы DMA могут выполнять процедуру самоинициализации (выполнение цепочки передач DMA).

Для выполнения самоинициализации в каналах DMA имеется 32-разрядный регистр CP, в котором хранится физический начальный адрес блока параметров очередного DMA обмена. Младшие три разряда регистра CP игнорируются (адреса выровнены по границе 64-



разрядного слова). Младший (нулевой разряд) регистра CP используется для старта режима самоинициализации. Эти параметры при самоинициализации аппаратно загружаются в 64-разрядном формате в соответствующие регистры канала DMA. Процедура этой загрузки ничем не отличается от обычного DMA обмена. Блок параметров может размещаться в любой памяти микросхемы.

Если необходимо продолжить цепочку команд, то необходимо указать CHEN=1. В режиме самоинициализации при записи параметров в регистр CSR биты END и DONE недоступны.

Для запуска работы канала DMA в режиме с самоинициализацией необходимо в регистр CP записать адрес первого блока параметров DMA передачи. При этом 0 разряд записываемых данных должен содержать 1 (признак пуска самоинициализации). В результате этого, соответствующий канал загрузит в свои регистры параметры DMA передачи и начнет обмен данными.

После окончания передачи блока данных бит END в регистре CSR устанавливается в единичное состояние, если бит IM = 1 - выдается прерывание. По окончании передачи блока данных также проверяется состояние бита CHEN. Если он равен 1, то будет загружен следующий блок параметров DMA передачи и т.д. В противном случае цепочка DMA обменов закончится и в регистре CSR бит DONE установится в единичное состояние и выдается прерывание.

Параметры для самоинициализации каналов DMA MEM\_CH размещаются в памяти в трех последовательных 64-разрядных словах, следующим образом (в порядке возрастания адресов):

```

64 _____ 0
   {      IR132,      IR032      };
   { {WCY16, ORY16}, { OR116, OR016 } };
   {      CSR32,      CP32      }.
    
```

Параметры для самоинициализации каналов DMA портов размещаются в памяти в двух последовательных 64-разрядных словах, следующим образом (в порядке возрастания адресов):

```

64 _____ 0
   { IR32,      -32 };
   { CSR32, CP32 } .
    
```

При необходимости каналы DMA могут инициализироваться программно. Для этого CPU должен записать адрес в регистр IR, а затем регистр CSR. При загрузке регистра CSR бит RUN необходимо установить в единичное состояние.

## 7.6 Прерывания DMA

Канал DMA формирует прерывание в соответствующем регистре QSTR (при условии, если установлены соответствующие биты в регистре MASKR и в поле IM[12:10] регистра STATUS CPU) при единичном состоянии битов DONE или END.

Обнуление битов DONE и END (и снятие соответствующего прерывания) выполняется посредством чтения содержимого регистра CSR или записью в эти биты нулей.

## 8. ПОРТ ВНЕШНЕЙ ПАМЯТИ ОБЩЕГО НАЗНАЧЕНИЯ

### 8.1 Введение

Порт внешней памяти общего назначения (MPORT) позволяет организовать обмен данными с широким набором устройств памяти и периферии. Внешний интерфейс порта обеспечивает подключение без дополнительной логики синхронной статической и динамической (SDRAM) памяти, а также асинхронной памяти, например EPROM и FLASH.

Порт внешней памяти имеет следующие основные характеристики:

- шина данных внешней памяти – 64 разряда;
- шина адреса внешней памяти – 32 разряда;
- программное конфигурирование типа блока памяти и его объема;
- интерфейс с синхронной динамической памятью типа SDRAM;
- интерфейс с синхронной статической памятью типа SBSRAM;
- интерфейс с асинхронной памятью (SRAM, EPROM, FLASH и т.д.);
- режим передачи данных Flyby(обмены памяти с устройством ввода-вывода);
- управление числом тактов ожидания при обмене с асинхронной памятью;
- формирование сигналов выборки 5 блоков внешней памяти.

### 8.2 Состав и назначение регистров порта внешней памяти

Перечень регистров порта внешней памяти приведен в таблице 8.1.

Таблица 8.1 Регистры порта внешней памяти

Условное обозначение регистра	Название регистра
CSCON0	Регистр конфигурации 0
CSCON1	Регистр конфигурации 1
CSCON2	Регистр конфигурации 2
CSCON3	Регистр конфигурации 3
CSCON4	Регистр конфигурации 4
SDRCON	Регистр конфигурации памяти типа SDRAM
SDRTMR	Регистр параметров SDRAM
SDRCSR	Регистр управления и состояния SDRAM
FLY_WS	Регистр внешних устройств

При описании полей и значений регистров используются обозначения:

- R – только чтение;
- W1 – пуск операции, реальная запись не производится;
- RW – чтение и запись;
- RW1 – Чтение, пуск операции;
- [ i ] – номер разряда;
- i:j – неразрывная группа разрядов, i –старший разряд группы, j –младший;
- 0x – далее следует шестнадцатеричный код;
- SCLK– частота SDRAM.

Термины и обозначения временных параметров и команд управления SDRAM соответствуют стандарту JESD79C.

### 8.2.1 Регистр конфигурации CSCON0

Регистр CSCON0 предназначен для конфигурирования блока внешней памяти, подключаемого к выводу nCS[0].

Формат регистра приведен в Таблица 8.2.

Таблица 8.2 Назначение разрядов регистра CSCON0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:24	-	Резерв	R	0
23	W64	Разрядность блока памяти: 0 – 32 разряда; 1 – 64 разряда	RW	0
22:21	T	Тип памяти блока памяти: 00 – асинхронная без ожидания сигнала ACK; 10 – асинхронная с ожиданием сигнала ACK; 01 – синхронная динамическая; 11 – синхронная статическая	RW	0
20	E	Разрешение формирования сигнала nCS[0]: 0 – запрещено; 1 – разрешено	RW	0
19:16	WS	Число тактов ожидания при обращении к блоку памяти, если он сконфигурирован как асинхронная память	RW	0xF
15:8	CSBA	Разряды 31:24 базового адреса блока памяти. Младшие разряды базового адреса равны нулю	RW	0
7:0	CSMASK	Разряды маски 31:24 при определении базового адреса блока памяти. Младшие разряды маски равны нулю	RW	0

Сигнал nCS[0] формируется, если при  $E = 1$  выполнено условие  $PHA[31:24] \& CSMASK = CSBA$ , где PHA – 32-разрядный физический адрес.

Если это условие выполнено, но  $E = 0$ , то обмен будет произведен с блоком внешней памяти, подключенным к выводу nCS[4].

Минимальный размер блока – 16 Мбайт (при  $CSMASK = 0xFF$ ). Для увеличения размера блока в младшие разряды поля CSMASK необходимо записать соответствующее число нулей. Например, для блока размером в 128 Мбайт, разряды 2:0 CSMASK должны быть равны нулю.

Регистры CSCON должны быть сконфигурированы таким образом, чтобы определяемые ими блоки памяти занимали уникальные адресные пространства. Если эти пространства перекрываются, то результат обмена данными будет непредсказуем.

В поле WS регистров CSCON задается количество тактов ожидания в тактах частоты SCLK, которое необходимо добавить в цикл шины при обращении к асинхронной внешней памяти. При аппаратном сбросе микропроцессора в поле WS всех регистров CSCON устанавливается значение 0xF (15 тактов). При  $WS = 0$  цикл шины составляет 2 такта SCLK.

Внешнее управление длительностью цикла обмена микропроцессора с асинхронной памятью осуществляется сигналом ACK. Сигнал ACK позволяет вставлять такты ожидания непосредственно в начатый цикл обмена данными. Количество вставленных тактов ожидания равно максимальному количеству дополнительных тактов, заданных полем WS и сигналом ACK.

### 8.2.2 Регистр конфигурации CSCON1

Регистр CSCON1 предназначен для конфигурирования блока внешней памяти, подключаемого к выводу nCS[1].

Формат регистра приведен в таблице 8.3.

Таблица 8.3 Назначение разрядов регистра CSCON1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:24	-	Резерв	R	0
23	W64	Разрядность блока памяти: 0 – 32 разряда; 1 – 64 разряда	RW	0
22:21	T	Тип памяти блока памяти: 00 – асинхронная без ожидания сигнала ACK; 10 – асинхронная с ожиданием сигнала ACK; 01 – синхронная динамическая; 11 – синхронная статическая	RW	0
20	E	Разрешение формирования сигнала nCS[1]: 0 – запрещено; 1 – разрешено	RW	0
19:16	WS	Число тактов ожидания при обращении к блоку памяти, если он сконфигурирован как асинхронная память	RW	0xF
15:8	CSBA	Разряды 31:24 базового адреса блока памяти. Младшие разряды базового адреса равны нулю	RW	0
7:0	CSMASK	Разряды маски 31:24 при определении базового адреса блока. Младшие разряды маски равны нулю	RW	0

### 8.2.3 Регистр конфигурации CSCON2

Регистр CSCON2 предназначен для конфигурирования блока внешней памяти, подключаемого к выводу nCS[2].

Формат регистра приведен в таблице 8.4.

Таблица 8.4 Назначение разрядов регистра CSCON2

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:24	-	Резерв	R	0
23	W64	Разрядность блока памяти: 0 – 32 разряда; 1 – 64 разряда	RW	0
22:21	T	Тип памяти блока памяти: 00 – асинхронная без ожидания сигнала ACK; 10 – асинхронная с ожиданием сигнала ACK; 01, 11 – синхронная статическая	RW	0
20	E	Разрешение формирования сигнала nCS[2]: 0 – запрещено; 1 – разрешено	RW	0
19:16	WS	Число тактов ожидания при обращении к блоку памяти, если он сконфигурирован как асинхронная память	RW	0xF
15:8	CSBA	Разряды 31:24 базового адреса блока памяти. Младшие разряды базового адреса равны нулю	RW	0
7:0	CSMASK	Разряды маски 31:24 при определении базового адреса блока. Младшие разряды маски равны нулю	RW	0

Память, подключаемая к выводу nCS[2], может быть асинхронной или синхронной статической.

## 8.2.4 Регистр конфигурации CSCON3

Регистр CSCON3 предназначен для конфигурирования блока памяти, подключаемого к выводу nCS[3].

Формат регистра приведен в таблице 8.5.

Таблица 8.5 Назначение разрядов регистра CSCON3

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	-	Резерв	RW	0
30:29	-	Резерв	R	0
28	W64	Разрядность блока памяти при BOOT = 11: 0 – 32 разряда; 1 – 64 разряда	RW	0
27:26	-	Резерв	R	0
25:24	BOOT[1:0]	Состояние сигналов на одноименных входах микропроцессора. Они определяют источник и разрядность данных при начальной загрузке программ микропроцессора после снятия сигнала nRST: 00 – загрузка производится из 32-разрядного блока памяти, подключенного к выводу nCS[3]. В этом случае разрядность этого блока памяти изменить нельзя; 01 – загрузка производится из 8-разрядного блока памяти, подключенного к выводу nCS[3]. В этом случае разрядность этого блока памяти изменить нельзя; 10 – загрузка производится из 64-разрядного блока памяти, подключенного к выводу nCS[3]. В этом случае разрядность этого блока памяти изменить нельзя; 11 – загрузка производится из порта SPI MFBSP0. При этом к выводу nCS[3] может быть подключен 32-разрядный или 64-разрядный блок памяти. Его разрядность определяет бит W64 этого регистра	R	Определяется состоянием сигналов на одноименных входах микропроцессора
23:22	-	Резерв	R	0
21:20	ADDR[1:0]	Используются при программной записи данных в 8-разрядную асинхронную память (в том числе и NOR FLASH): при выполнении команды Store Word на линии адреса A[1:0] микропроцессора выдается содержимое поля ADDR[1:0] соответственно	RW	0
19:16	WS	Число тактов ожидания при обращении к памяти блока	RW	0
15:0	-	Резерв	R	0

Область памяти, определяемая регистром CSCON3, размещается в диапазоне физических адресов от 0x1C00\_0000 до 0x1FFF\_FFFF (64 Мбайт). Память данного блока может быть только асинхронной. Доступ к данному блоку памяти всегда разрешен. При обмене данными с этим блоком сигнал ACK безразличен.

Как правило, к выводу nCS[3] подключается блок памяти программ, реализованный на NOR FLASH, PROM, EEPROM и т.д. Разрядность этого блока, в зависимости от состояния сигналов на выводах микросхемы BOOT и бита W64, может быть 8, 32 или 64.

8-разрядная память подключается к выводам D[7:0] микропроцессора. Шину адреса A[31:0] к этой памяти необходимо подключать, начиная с 0 разряда (к 32 и 64-разрядной памяти адрес подключается, начиная со 2 разряда). 32 или 64-разрядное слово из 8-разрядной памяти считывается байтами, причем сначала считывается старший байт слова. Запись данных в 8-разрядную память выполняется побайтно в соответствии с рекомендациями п. 8.4.2.

### 8.2.5 Регистр конфигурации CSCON4

Регистр CSCON4 предназначен для конфигурирования внешней памяти, не вошедшей в блоки памяти, определяемые регистрами CSCON3 - CSCON0.

Данный блок памяти подключается к выводу nCS[4].

Формат регистра приведен в таблице 8.6.

Таблица 8.6 Назначение разрядов регистра CSCON4

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:20	-	Резерв	R	0
19:16	WS	Число тактов ожидания при обращении к памяти блока	RW	0
15:0	-	Резерв	R	0

Память данного блока может быть только асинхронной разрядности 32. Доступ к данному блоку памяти всегда разрешен. При обмене данными с этим блоком сигнал АСК безразличен.

### 8.2.6 Регистр конфигурации SDRCON

Регистр SDRCON предназначен для программирования конфигурационных параметров синхронной памяти типа SDRAM.

Формат регистра приведен в таблице 8.7.

Таблица 8.7 Формат регистра SDRCON

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	-	Резерв	R	0
29:16	tRFR	Период регенерации SDRAM в тактах частоты SCLK	RW	0
15:13	-	Резерв	R	0
12	-	Резерв	RW	0
11:7	-	Резерв	R	0
6:4	CL	Задержка данных при чтении (CAS latency): 010 – 2 такта SCLK; 011 – 3 такта SCLK. Остальные значения этого поля – резерв. Записанное значение передается в SDRAM при выполнении команды инициализации SDRAM. При чтении считывается значение, установленное в SDRAM при её инициализации. Запись резервных кодов игнорируется	RW	0
3	-	Резерв	R	0
2:0	PS	Размер страницы микросхем SDRAM, подключенных к MPORT: 100 – 256; 000 – 512; 001 – 1024; 010 – 2048; 011 – 4096. Число банков SDRAM – 4	RW	0

Память данного типа может быть размещена только в блоке памяти, подключенном к выводам nCS[0] или/и nCS[1].

Преобразование физического адреса в адрес 64-разрядной памяти SDRAM при различных значениях параметра PS представлено в таблицах 8.8-8.10. Разряды физического адреса в таблицах обозначены строчными буквами “а”.

**Таблица 8.8 Отображение адреса строки для 64-разрядной памяти**

PS	Адрес SDRAM												
	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
100	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13
000	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14
001	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15
010	a28	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16
011	a29	a28	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17

**Таблица 8.9 Отображение адреса столбца для 64-разрядной памяти**

PS	Адрес SDRAM												
	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
100	0	0	0	0	0	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3
000	0	0	0	0	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3
001	0	0	0	a12	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3
010	0	a13	0	a12	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3
011	a14	a13	0	a12	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3

**Таблица 8.10 Отображение адреса банка для 64-разрядной памяти**

PS	Адрес банка SDRAM	
	BA1	BA0
100	a12	a11
000	a13	a12
001	a14	a13
010	a15	a14
011	a16	a15

Преобразование физического адреса в адрес 32-разрядной памяти SDRAM представлено в таблицах 8.11-8.13. Разряды физического адреса в таблицах обозначены строчными буквами “а”.

**Таблица 8.11 Отображение адреса строки для 32-разрядной памяти**

PS	Адрес SDRAM												
	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
100	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12
000	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13
001	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14
010	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15
011	a28	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16

**Таблица 8.12 Отображение адреса столбца для 32-разрядной памяти**

PS	Адрес SDRAM												
	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
100	0	0	0	0	0	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2
000	0	0	0	0	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2
001	0	0	0	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2
010	0	a12	0	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2
011	a13	a12	0	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2

**Таблица 8.13 Отображение адреса банка для 32-разрядной памяти**

PS	Адрес банка SDRAM	
	BA1	BA0
100	a11	a10
000	a12	a11
001	a13	a12
010	a14	a13
011	a15	a14

Период регенерации tRFR должен определяться индивидуально для используемой конфигурации памяти. Например, при тактовой частоте SCLK 200 МГц для обеспечения 8 192 цикловой регенерации за 64 мс необходимо в поле tRFR записать код 0x61A, что соответствует 7, 81 мкс на строку, а при частоте 100 МГц - 0x30D.

После инициализации SDRAM MPORT аппаратно выполняет процедуру регенерации с периодом tRFR тактов SCLK. Режим регенерации отключается при tRFR =0 или при переводе SDRAM в режим саморегенерации или пониженного потребления.



### 8.2.7 Регистр параметров *SDRTMR*

Регистр *SDRTMR* предназначен для задания интервалов (в тактах частоты *SCLK*) между различными командами *SDRAM*.

Формат регистра приведен в таблице 8.14.

**Таблица 8.14 Формат регистра *SDRTMR***

Номер разряда	Условное Обозначение параметра	Назначение	Доступ	Исходное состояние
31:28	-	Резерв	R	0
27:24	tRC	Минимальный период команд ACTIVE для одного и того же банка	RW	0
23:20	tRFC	Минимальный интервал между командами AUTO REFRESH.	RW	0
19:16	tRAS	Минимальная задержка между командами ACTIVE и PRECHARGE	RW	0
15:14	-	Резерв	R	0
13:12	-	Резерв	RW	0
11:10	-	Резерв	R	0
9:8	tRCD	Минимальная задержка между командами ACTIVE и READ/WRITE	RW	0
7:6	-	Резерв	R	0
5:4	tRP	Минимальный период команд PRECHARGE	RW	0
3:2	-	Резерв	R	0
1:0	tWR	Минимальная задержка между последней записью данных и командой PRECHARGE (Write recovery).	RW	0

Значения 0, 1, ..., n параметра в таблице соответствуют интервалу в 1, 2, ..., n+1 тактов.

Например, значение 0xF параметра tRFC задает интервал 16 тактов между командами AUTO REFRESH, а значение 0 – интервал в один такт.

При вычислении параметров в соответствии с рабочей частотой и со спецификацией используемой памяти, полученные значения необходимо округлять до ближайшего меньшего целого. Например, если в спецификации указано время tRCD = 20 нсек, то при частоте *SCLK* 133 МГц (период 7.5 нсек) минимальный интервал в 2.7 такта нужно округлить до 2 и в поле tRCD регистра *SDRTMR* записать код 0x2.

### 8.2.8 Регистр состояний и управления SDRCSR

Регистр SDRCSR предназначен для запуска команд изменения режимов SDRAM и индикации их исполнения.

Формат регистра SDRCSR приведен в Таблица 8.15.

Таблица 8.15 Формат регистра SDRCSR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:5	-	Резерв.	R	0
4	EXIT	При записи 1 в данный разряд MPORT выполняет последовательность команд вывода SDRAM из режимов саморегенерации и пониженного потребления. При чтении - признак выполнения команды выхода SDRAM из указанных режимов: устанавливается в 1 после завершения команды; сбрасывается при записи любой команды	RW1	0
3	PWDN	При записи 1 в данный разряд MPORT переводит SDRAM в режим пониженного потребления. При чтении - признак окончания данной команды: устанавливается в 1 после завершения команды; сбрасывается записью команды EXIT	RW1	0
2	SREF	При записи 1 в данный разряд MPORT переводит SDRAM в режим саморегенерации. При чтении - признак окончания данной команды: устанавливается в 1 после завершения команды; сбрасывается записью команды EXIT	RW1	0
1	AREF	При записи 1 в данный разряд MPORT выполняет команду авторегенерации SDRAM. При чтении - признак окончания команды авторегенерации: устанавливается в 1 после завершения данной команды; сбрасывается при записи любой команды	RW1	0
0	INIT	При записи 1 в данный разряд MPORT выполняет инициализацию SDRAM с параметрами: Burst Length – 1; Burst Type – Sequential; CAS Latency – поле CL регистра SDRCON; Operation Mode – Standart Operation WB – Programmed Burst Length. При чтении - признак окончания команды инициализации: устанавливается в 1 после завершения данной команды; сбрасывается при записи любой команды	RW1	0

Команды кодируются унитарным кодом в разрядах 4:0. Запись других кодов или запись новой команды до завершения предыдущей игнорируются.

При запуске любой команды изменения режимов MPORT ожидает завершения текущего обмена (в том числе регенерации), приостанавливает выполнение очередного обмена с SDRAM и выполняет необходимую последовательность команд SDRAM. Во время исполнения команды значение регистра SDRCSR - 0

По команде INIT выполняется последовательность команд инициализации:

- PRECHARGE;
- Пауза tRP, AUTO REFRESH;
- Пауза tRFC, AUTO REFRESH;
- Пауза tRFC, Load Mode Register;
- Пауза tMRD, установка индикатора INIT.

Длительность выполнения команды INIT составляет порядка 30 тактов SCLK.

До выполнения начальной инициализации необходимо записать все параметры в регистры SDRCON, SDRTMR и сконфигурировать регистры CSCON0 и/или CSCON1.

MPORT не контролирует задержку 200 мкс между установкой стабильного питания и запуском команды INIT.

По команде AREF контроллер выполняет:

- PRECHARGE;
- пауза tRP, AUTO REFRESH;
- пауза tRFC, установка индикатора AREF.

По команде PWDN MPORT выполняет:

- PRECHARGE;
- Пауза 1 такт SCLK;
- Сброс CKE, NOP;
- Пауза tRFC, установка индикатора PWDN.

После выполнения данной команды память находится в “режиме precharge power down”.

По команде SREF MPORT выполняет:

- PRECHARGE;
- Пауза tRP;
- SELF REFRESH;
- Пауза tRFC, установка индикатора SREF.

После выполнения команд PWDN и SREF MPORT находится в состоянии ожидания команды EXIT и игнорирует другие команды изменения режимов SDRAM. В этом состоянии MPORT не контролирует выполнение интервала tREF.

По команде EXIT контроллер устанавливает CKE и, после паузы tXSNR(или 2 такта SCLK при выходе из режима PWDN), выполняет AREF и устанавливается индикатор EXIT.

MPORT игнорирует команду EXIT при сброшенных индикаторах PWDN и SREF.

### 8.2.9 Регистр FLY\_WS

Данный регистр определяет количество дополнительных тактов ожидания в обменах внешних устройств с асинхронной памятью.

Формат регистра FLY\_WS приведен в Таблица 8.16.

Таблица 8.16 Формат регистра FLY\_WS

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16		Резерв	R	0
15:11	FWS3	Число тактов ожидания для внешнего устройства 3 при обмене с асинхронной памятью	RW	0
11:7	FWS2	Число тактов ожидания для внешнего устройства 2 при обмене с асинхронной памятью	RW	0
7:4	FWS1	Число тактов ожидания для внешнего устройства 1 при обмене с асинхронной памятью.	RW	0
3:0	FWS0	Число тактов ожидания для внешнего устройства 0 при обмене с асинхронной памятью	RW	0

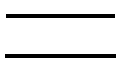



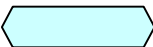

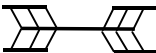


Количество вставленных тактов ожидания равно максимальному количеству дополнительных тактов, заданных сигналом ACK и полями WS и FWS участников обмена.

## 8.3 Временные диаграммы обмена данными

### 8.3.1 Общие положения

При описании временных диаграмм используются условные обозначения в соответствии с Таблица 8.17.

Таблица 8.17. Условные обозначения

Условное обозначение	Описание
	Стабильное значение
	Возможное значение
	область изменения из «0» в «1»
	область изменения из «1» в «0»
	Достоверное значение
	Для входов: Не воспринимается, допустимо любое переключение Для выходов: состояние не определено
	Переключение выхода из (в) высокоимпедансное состояние (центральная линия)
	Повторение сигнала в течение неопределенного времени
<b>T<sub>i</sub></b>	$i = 1, 2, \dots$ фаза обмена на временной диаграмме
<b>n</b>	Количество дополнительных тактов ожидания, задаваемых полем WS регистров CCON
<b>w</b>	Количество тактов ожидания высокого уровня сигнала ACK
<b>nCS<sub>x</sub></b>	Один из четырёх сигналов nCS[3:0]
<b>nOЕ<sub>x</sub></b>	Один из четырёх сигналов nOE[3:0]
<b>nFLYBY<sub>x</sub></b>	Один из четырёх сигналов nFLYBY [3:0]
	Момент приема данных

### 8.3.2 Обмен данными с асинхронной памятью

Временные диаграммы записи данных в асинхронную память приведены на Рисунок 8.1 - Рисунок 8.3.

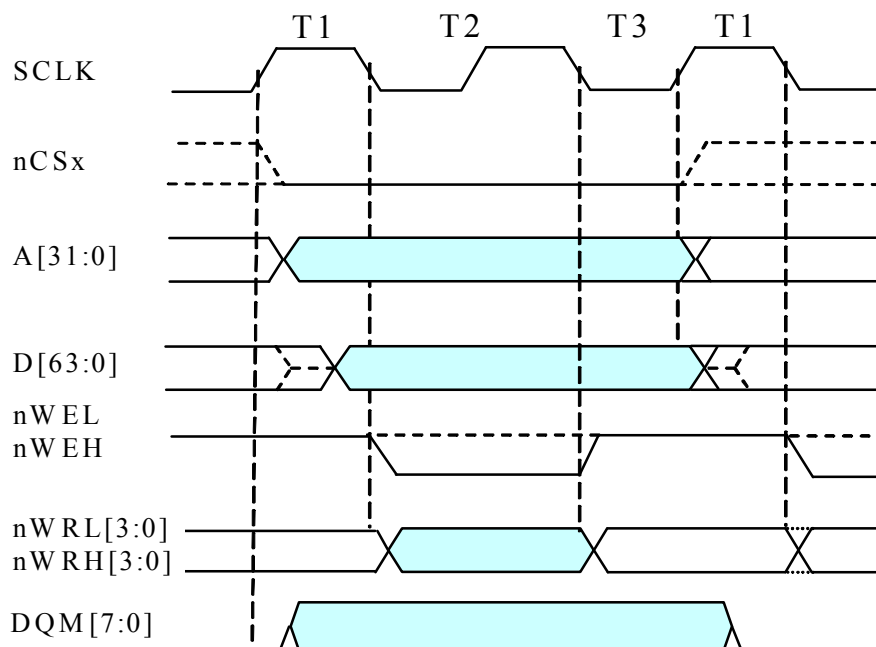


Рисунок 8.1 .Запись в асинхронную память без дополнительных тактов ожидания.

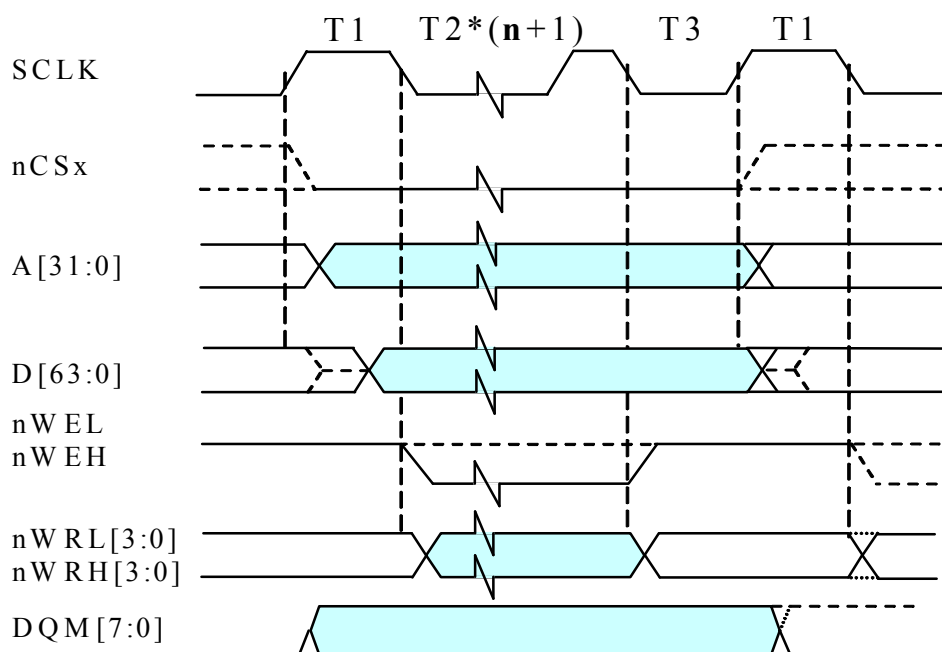


Рисунок 8.2. Запись в асинхронную память с n дополнительными тактами ожидания.

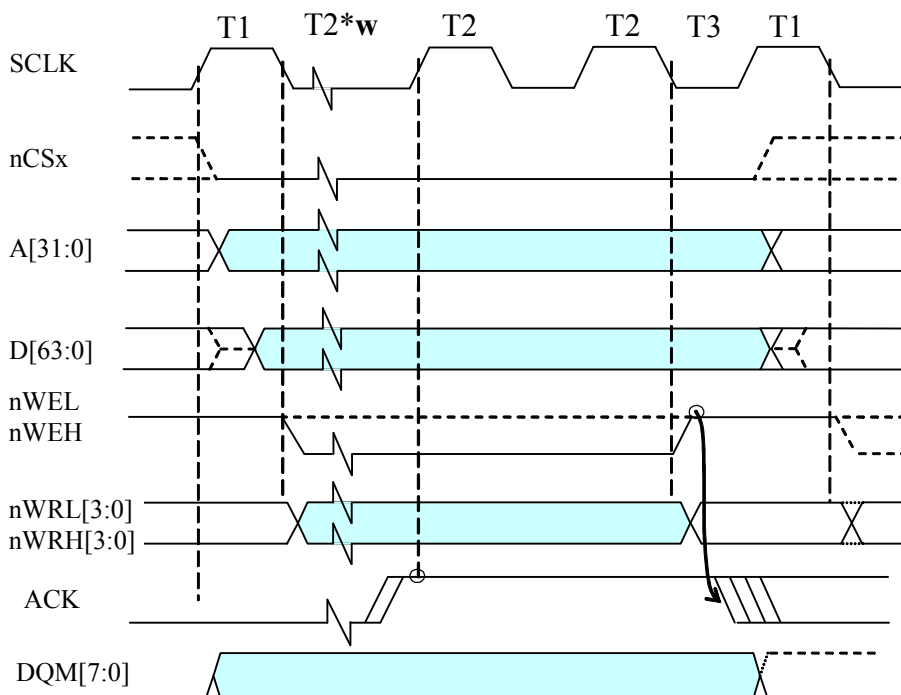


Рисунок 8.3. Запись в асинхронную память с ожиданием сигнала ACK.

Временные диаграммы чтения данных из асинхронной памяти приведены на Рисунок 8.4 - Рисунок 8.6. Выводы DQM[7:0] не изменяются.

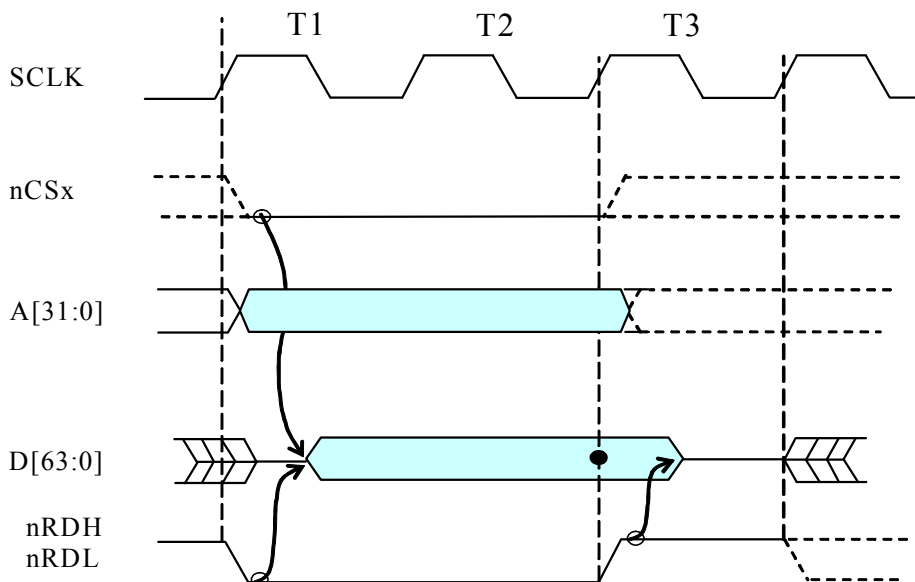
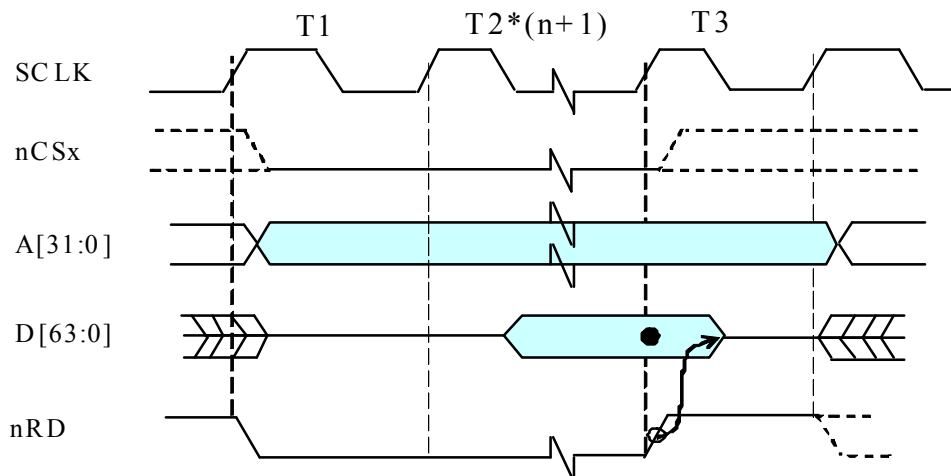
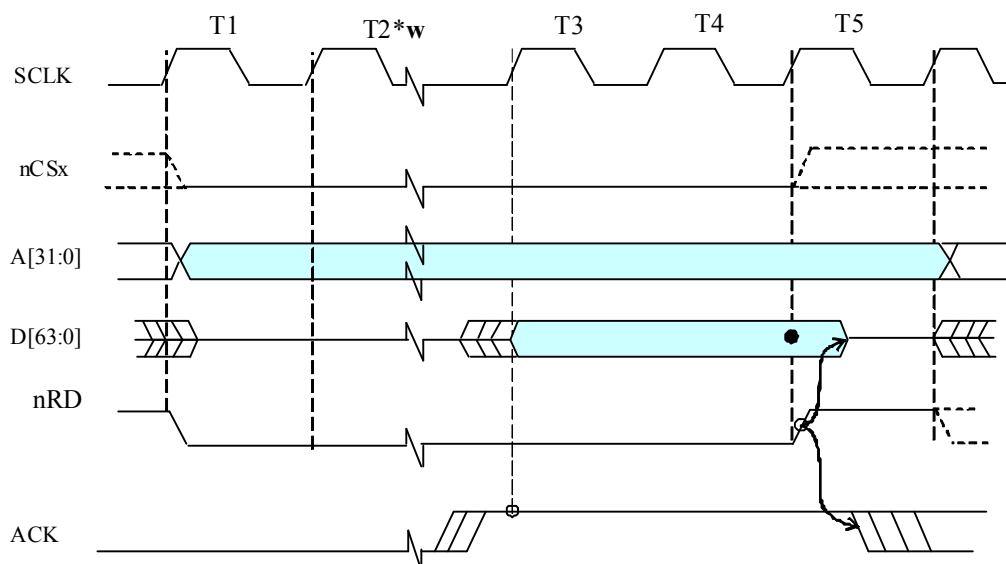


Рисунок 8.4. Чтение асинхронной памяти без дополнительных тактов ожидания.



**Рисунок 8.5. Чтение асинхронной памяти с  $n$  дополнительными тактами ожидания.**

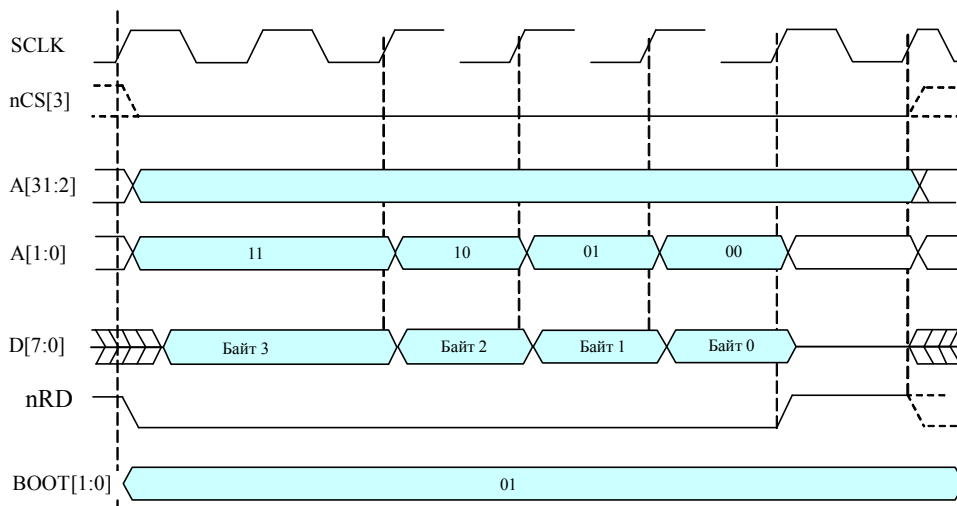


**Рисунок 8.6. Чтение данных из асинхронной памяти с ожиданием сигнала ACK.**

Как правило, в блоке внешней памяти, подключенному к сигналу выборки памяти  $nCS[3]$ , размещается постоянное запоминающее устройство (ПЗУ), реализованное на FLASH, PROM, EEPROM и т.д.

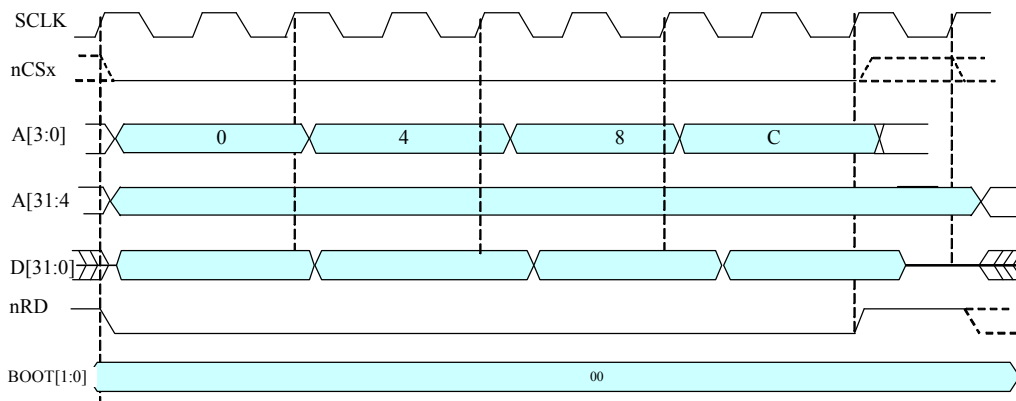
В зависимости от состояния выводов микросхемы BOOT этот блок внешней памяти может быть 8, 32 или 64 - разрядным. На Рисунок 8.7 приведена временная диаграмма чтения 32-разрядного слова из 8-разрядного ПЗУ при  $BOOT = 01$ .



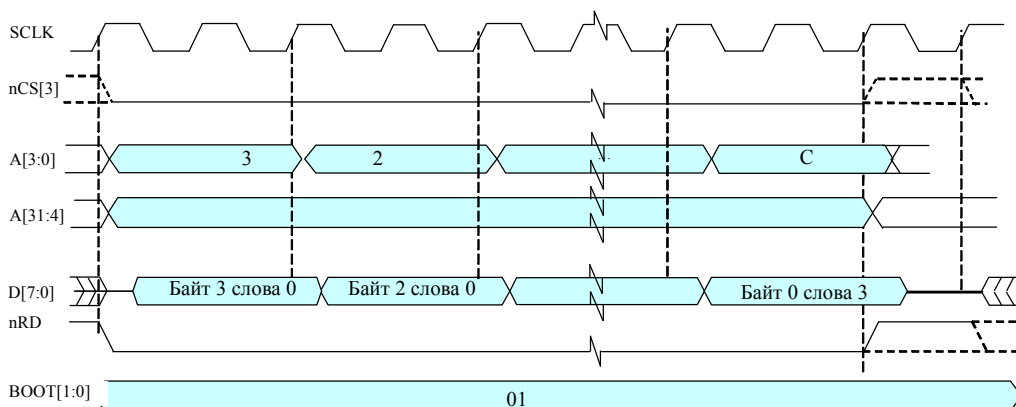


**Рисунок 8.7. Чтение 32-разрядного слова из 8-разрядного ПЗУ (n = 0).**

Если CPU выполняет программу из кэшируемой области внешней памяти, то загрузка строки кэш (процедура Refill) выполняются посредством чтения двух 64 -разрядных слов в режиме burst. Адрес, по которому начинается burst, выровнен по 16-байтной границе. На Рисунок 8.8 приведена временная диаграмма выполнение процедуры Refill из 32-разрядной асинхронной памяти. На рисунке 8.9 приведена временная диаграмма выполнение процедуры Refill из 8-разрядного ПЗУ.



**Рисунок 8.8. Выполнение процедуры Refill из 32-разрядной асинхронной памяти (n = 0).**



**Рисунок 8.9. Выполнение процедуры Refill из 8-разрядного ПЗУ (n = 0) Обмен данными с синхронной памятью**

Временные диаграммы с синхронной памятью приведены на рисунках 8.10-8.16. Временные диаграммы инициализации и регенерации SDRAM приведены на рисунках 8.17 и 8.18 соответственно.

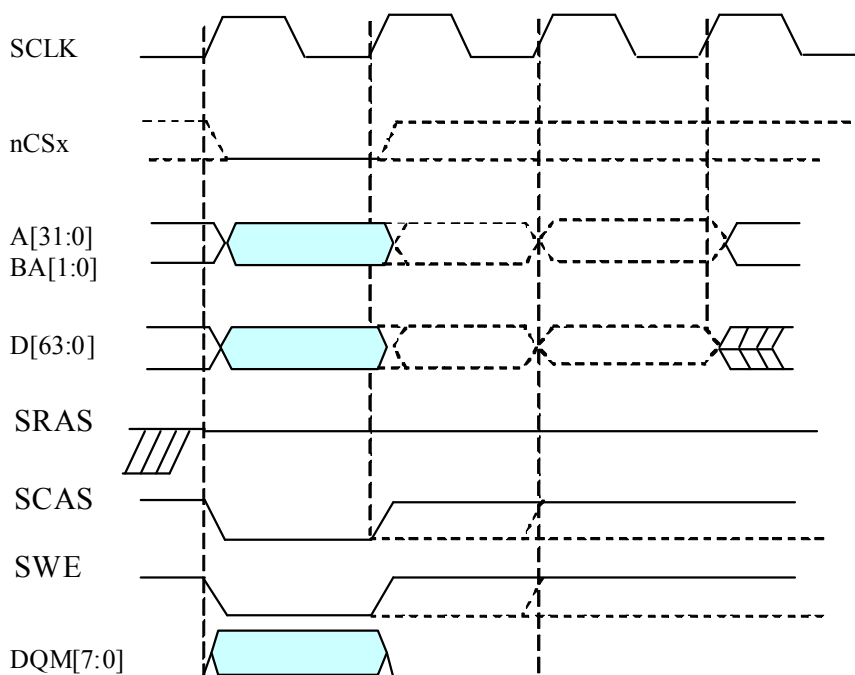


Рисунок 8.10. Запись одного слова данных в синхронную память.

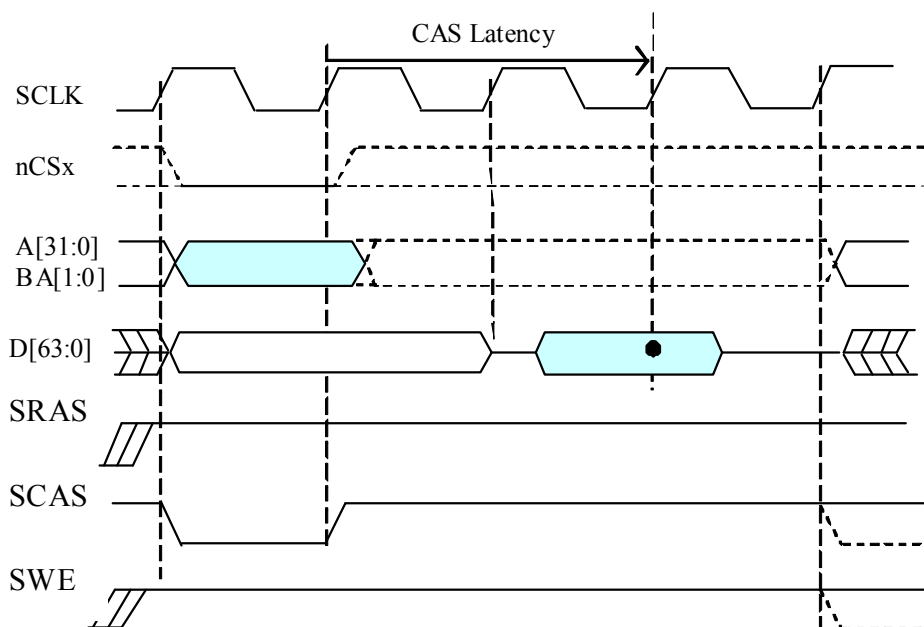
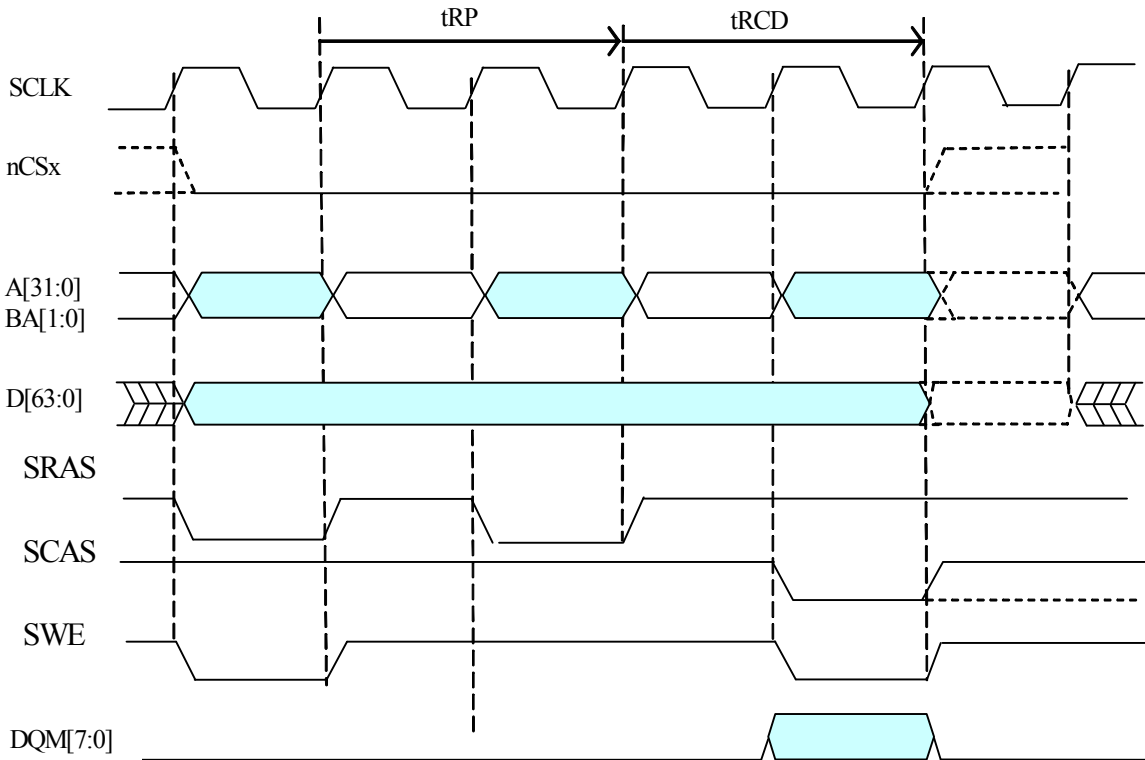
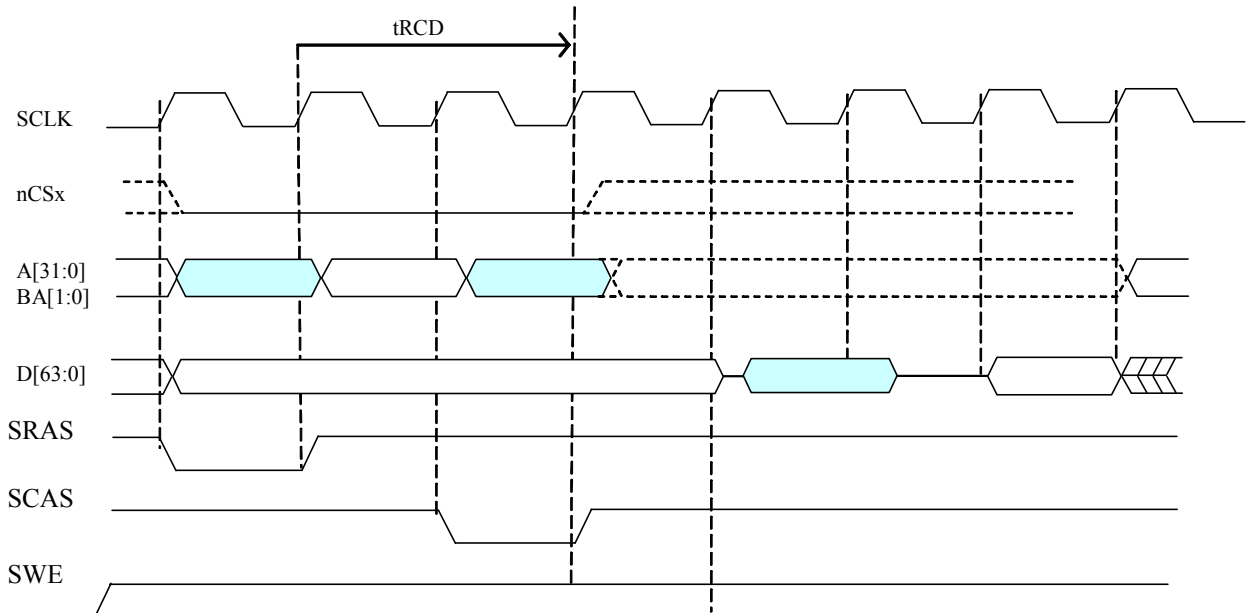


Рисунок 8.11. Чтение одного слова данных из синхронной памяти (здесь и далее CAS latency = 2)



**Рисунок 8.12. Запись одного слова данных в синхронную память с деактивизацией строки**



**Рисунок 8.13. Чтение одного слова данных из синхронной памяти с активизацией строки.**

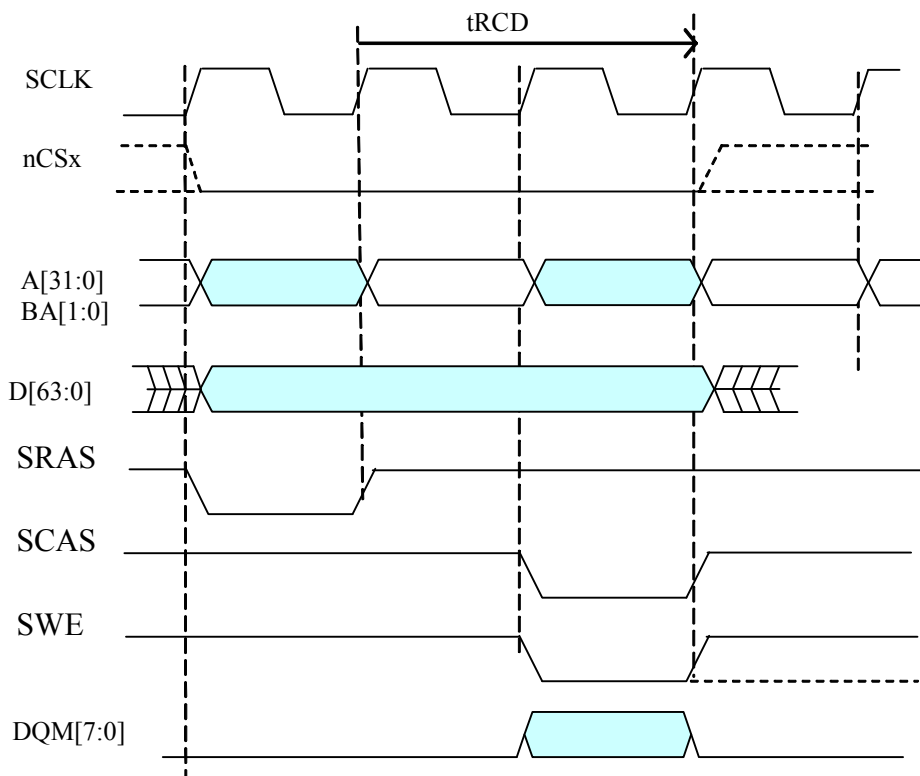


Рисунок 8.14. Запись одного слова данных в синхронную память с активизацией строки.

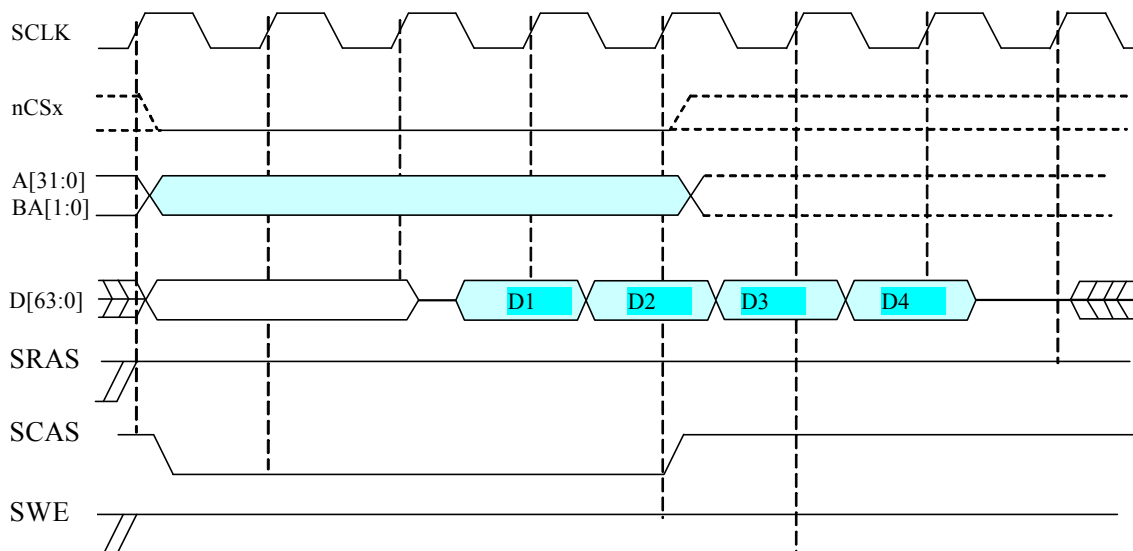


Рисунок 8.15. Чтение 4-х слов данных из синхронной памяти в режиме "burst".

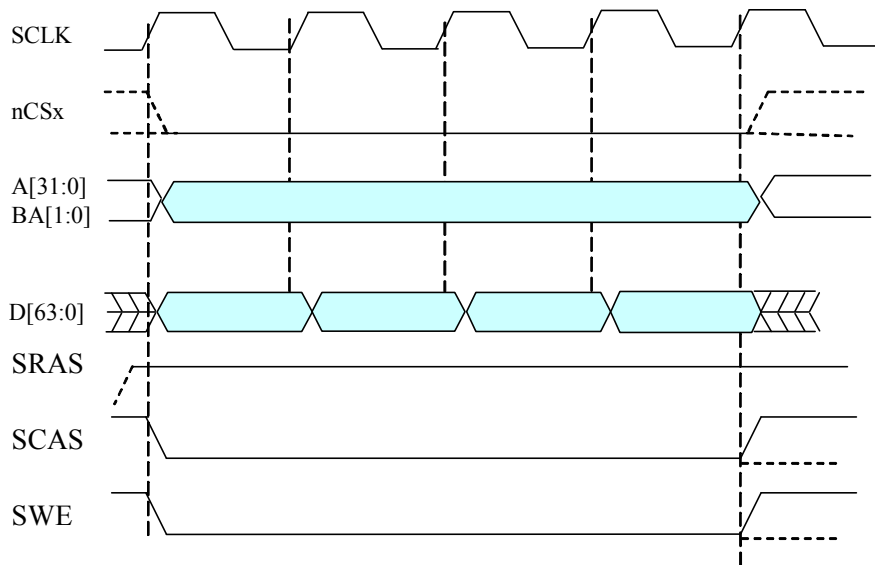


Рисунок 8.16. Запись 4-х слов данных в синхронную память в режиме "burst".

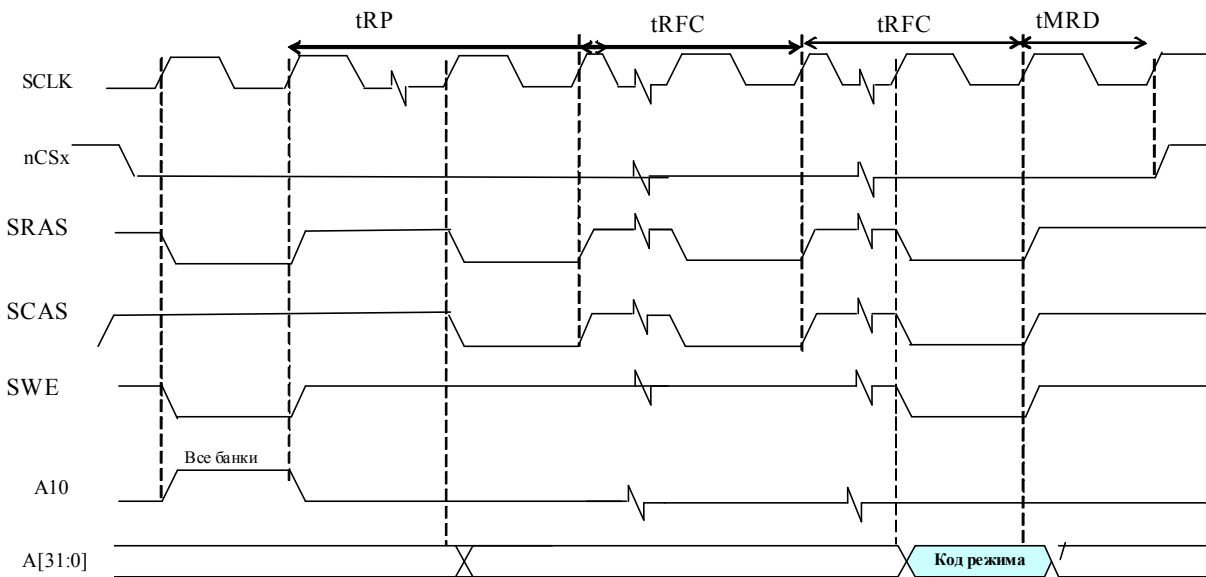


Рисунок 8.17. Инициализация синхронной памяти

Рису-

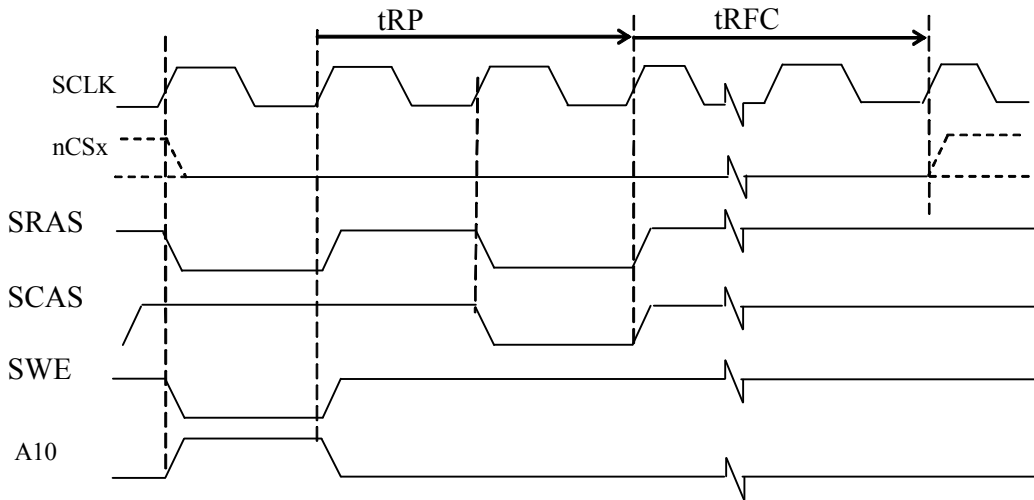


Рисунок 8.18. Временная диаграмма регенерация синхронной памяти.

### 8.3.3 Обмен данными в режиме Flyby

Режим Flyby используется контроллером DMA (каналы MEM\_CH) для передачи данных между внешним устройством ввода-вывода и внешней памятью (как асинхронной, так и синхронной). Например, контроллер DMA может быть запрограммирован для передачи данных из аналого-цифрового преобразователя в SDRAM. Для выполнения передачи данных в режиме Flyby в соответствующем регистре CSR\_MEM\_CH необходимо установить бит FLYBY.

При передаче данных в режиме Flyby MPORT активизирует внешнюю память и внешнее устройство ввода-вывода одновременно. Память управляется как обычно, а устройство ввода-вывода – при помощи сигналов nFLYBY (признак данного режима) и nOE (активизация выходных формирователей устройства ввода-вывода).

В режиме Flyby MPORT выполняет обмен данными полными словами памяти. Объем передаваемой информации определяется форматом передачи (бит EN64 регистра CSR\_MEM\_CH), количеством передаваемых слов (биты WN регистра CSR\_MEM\_CH) и разрядностью памяти (бит W64 соответствующего регистра CSCON). При EN64 =0 и W64 =1 поле WN должно определять четное число слов, а начальный адрес передачи должен быть выровнен до границы 64-разрядного слова. Например, при EN64 =0, WN = 3 и W64 =1 MPORT выполнит передачу 2 слов памяти,

при EN64 =1, WN = 3 и W64 =1 MPORT выполнит передачу 4 слов памяти, а

при EN64 =1, WN = 3 и W64 =0 MPORT выполнит передачу 8 слов памяти.

Для 8-разрядной памяти EN64 определяет количество байтов в слове передачи: при EN64=0 из памяти передается 4 байта, при EN64=1 передается 8 байт. Например, если WN = 3, то при EN64=0 во внешнее устройство будет передано 16 байт, а при EN64=1 будет передано 32 байта.

Временные диаграммы обмена данными в режиме Flyby приведены на рисунках 8.19-8.24 (WS=0, WSF=0, AE=0, CL=2).

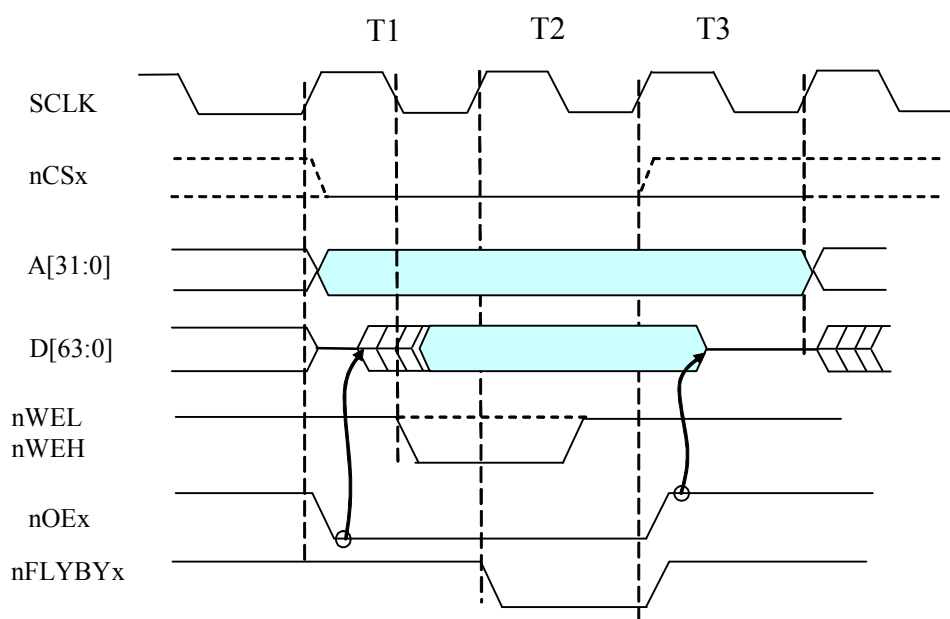


Рисунок 8.19. Передача одного слова данных из устройства ввода-вывода в асинхронную память.

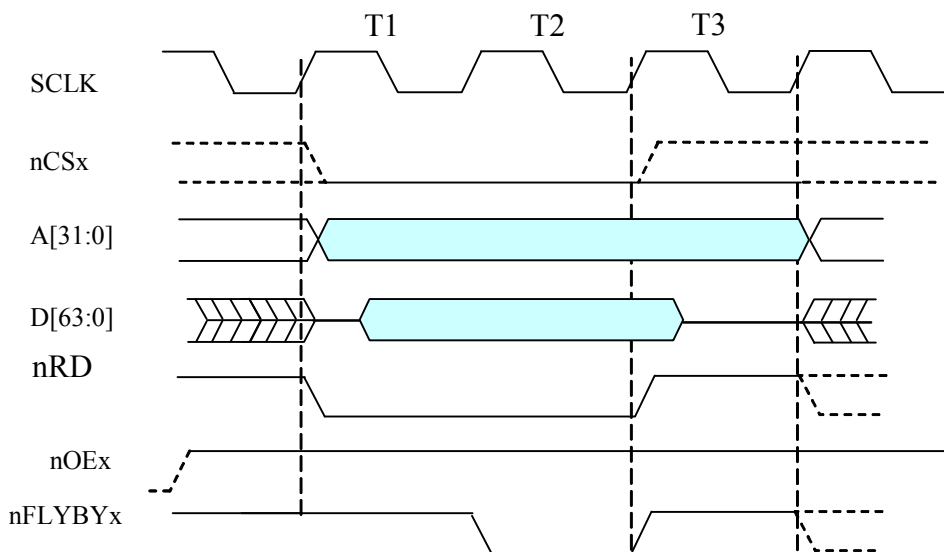


Рисунок 8.20. Передача одного слова данных из асинхронной памяти в устройство ввода-вывода.

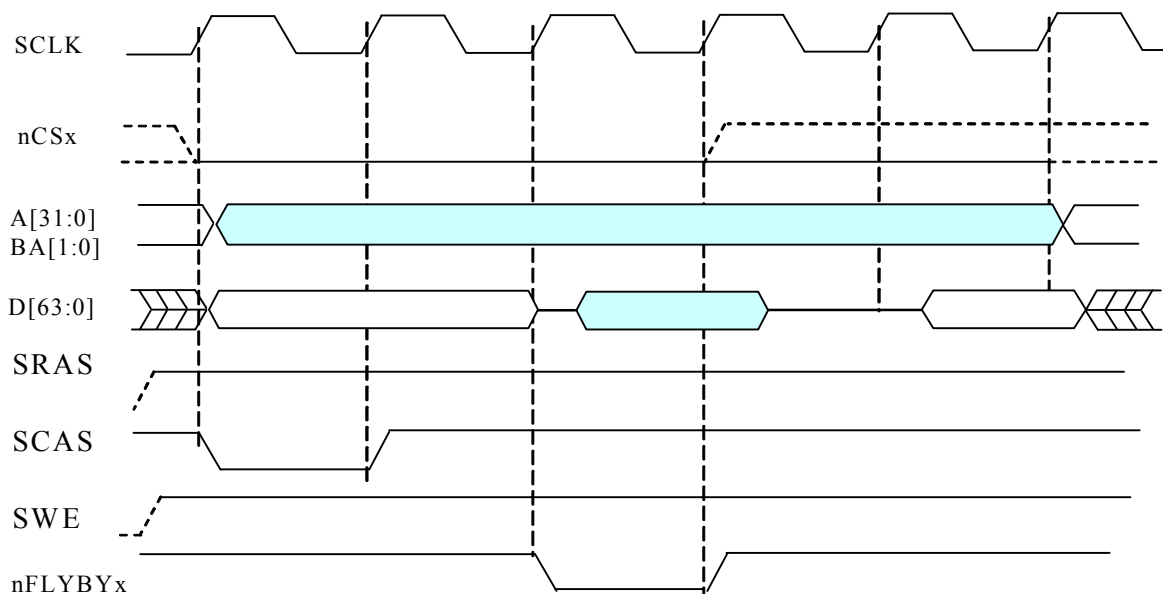


Рисунок 8.21. Передача одного слова данных из синхронной памяти в устройство ввода-вывода.

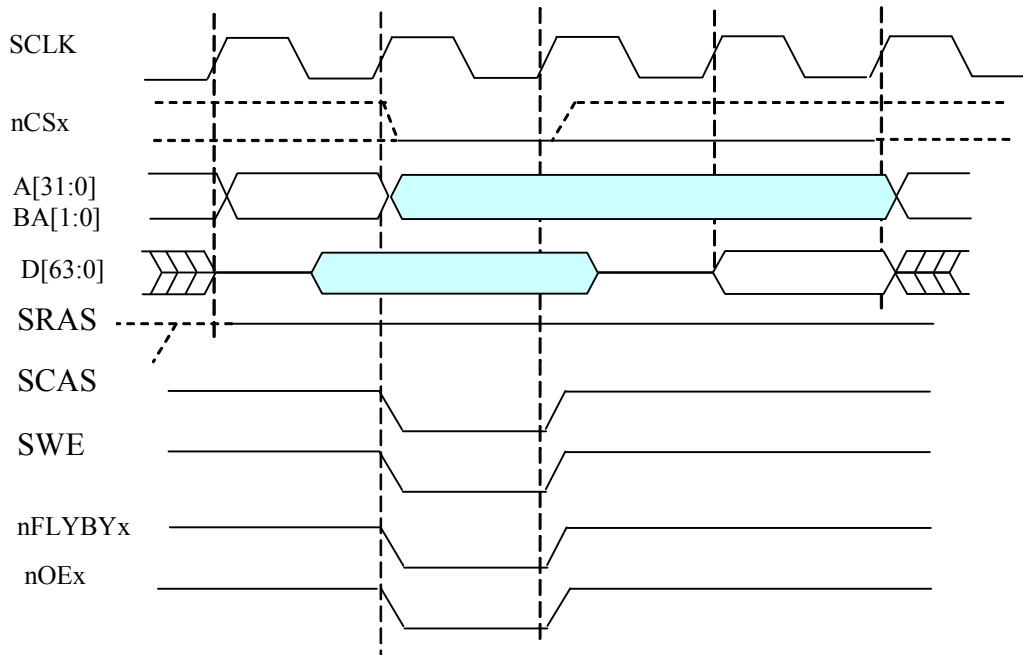


Рисунок 8.22. Передача одного слова данных из устройства ввода-вывода в синхронную память.

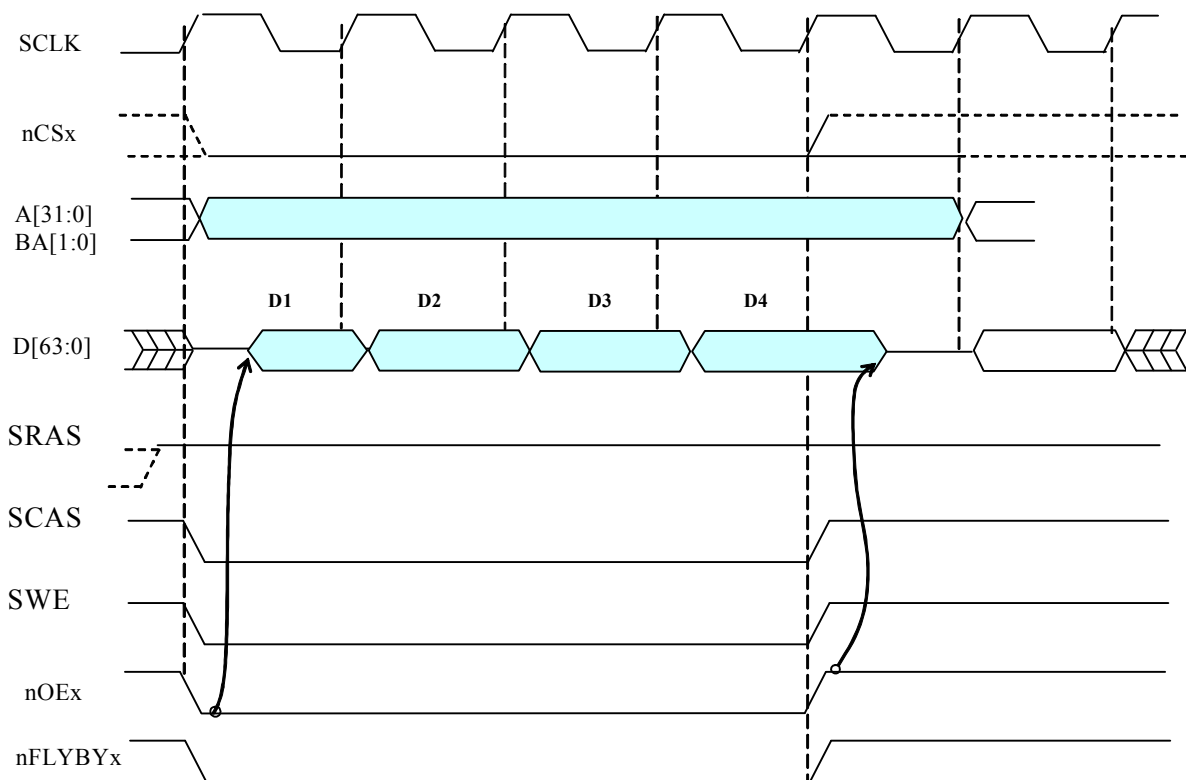


Рисунок 8.23. Передача 4-х слов данных из устройства ввода-вывода в синхронную память.



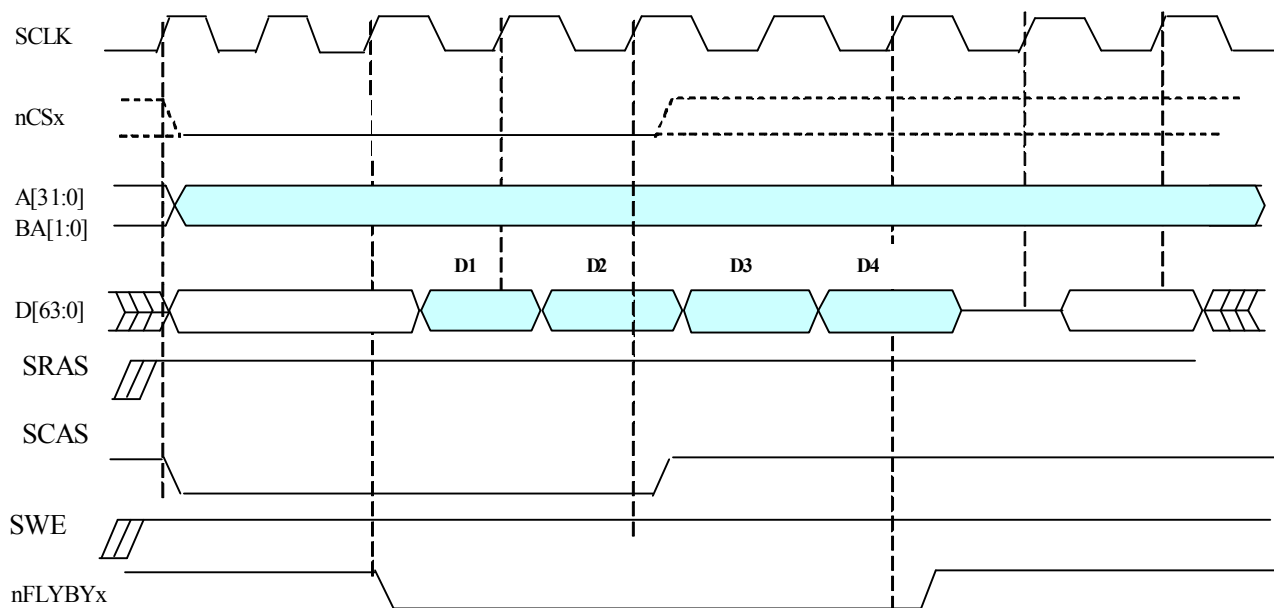


Рисунок 8.24. Передача 4-слов данных из синхронной памяти в устройство ввода-вывода.

### 8.3.4 Обмен данными с синхронной статической памятью

Временные диаграммы с синхронной памятью приведены на рисунках 8.25-8.26. Задержка данных составляет 2 такта SCLK.

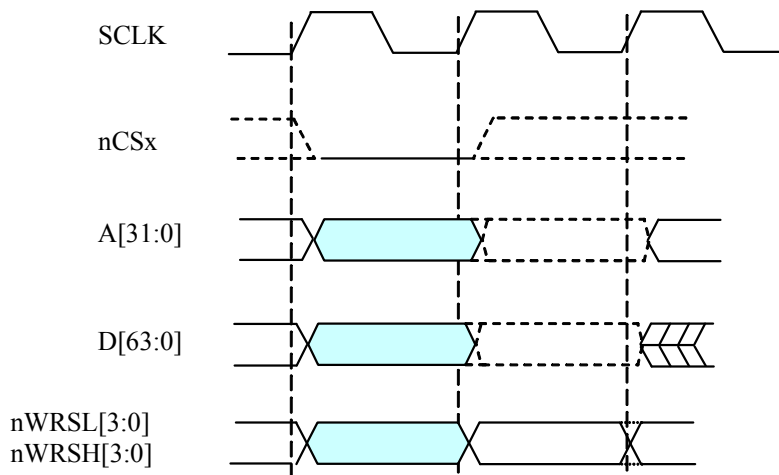


Рисунок 8.25. Запись одного слова данных в синхронную статическую память.

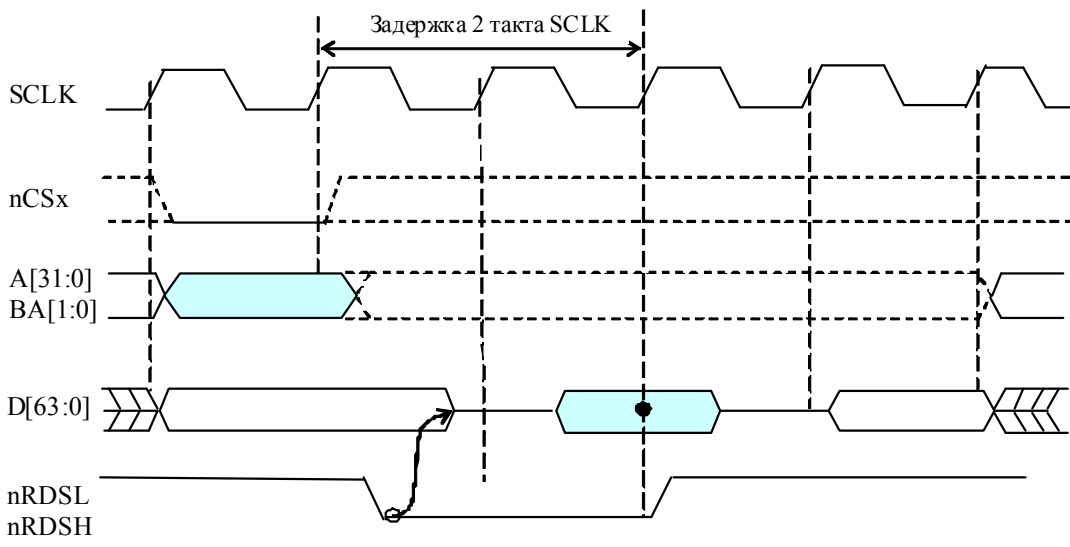


Рисунок 8.26. Чтение одного слова данных из синхронной статической памяти.

## 8.4 Рекомендации по подключению внешней памяти

### 8.4.1 Память типа SDRAM

Выводы адреса микросхем типа SDRAM подключаются к выводам шины адреса порта внешней памяти следующим образом:

- номер банка SDRAM – к выводам BA[1:0];
- адрес A[12:0] SDRAM – к выводам A[14:13], A10, A[11:2] соответственно.

### 8.4.2 Память типа NOR FLASH

К микропроцессору можно подключать 64-разрядную, 32-разрядную или 8-разрядную память типа NOR FLASH.

32- или 64-разрядная память FLASH подключается к микропроцессору аналогично асинхронной памяти. Как правило, она подключается к сигналу выборки памяти nCS[3] и используется для старта микропроцессора. Но при необходимости память FLASH может быть подключена к любому из четырех сигналов выборки памяти nCS[3:0].

8-разрядная память Flash подключается только к сигналу выборки памяти nCS[3]. При этом входы BOOT микропроцессора необходимо установить в состояние 01, а адресную шину микропроцессора подключить к памяти Flash, начиная с 0 разряда (к 32 или 64-разрядной памяти адрес подключается, начиная со 2 разряда). Вывод nWE микропроцессора необходимо подключить к входу разрешения записи памяти.

При использовании 8-разрядной памяти Flash возможны следующие варианты ее программирования:

1. Микросхемы этой памяти программируются на программаторе и потом распаиваются на плату или устанавливаются в контактирующее устройство.
2. Микросхемы этой памяти программируются на плате программно с использованием команды Store Byte. В этом случае MPORT выдает на выводы A[1:0] номер байта и коммутирует заказанный байт на выводы D[7:0]. При использовании других модификаций команды Store(например, Store Word, Store Halfword) MPORT выдает на разряды адреса A[1:0] состояние, заданное полем ADDR регистра CSCON3, а на выводы D[7:0] коммутирует младший байт операнда.
3. Микросхемы этой памяти программируются на плате через порт JTAG микропроцессора. В этом случае запись в память производится командой Store Word, поэтому перед каждой записью необходимо устанавливать в разрядах 21:20 регистра CSCON3 необходимое значение адреса байта. Для процесса программирования через порт JTAG необходим специальный драйвер, который не входит в состав MC Studio.

## 9. ПОРТ ВНЕШНЕЙ ПАМЯТИ ТИПА DDR SDRAM

### 9.1 Общие положения

В микросхеме имеется два порта внешней памяти типа DDR SDRAM (DDR\_PORT0, DDR\_PORT1).

Внешний интерфейс порта обеспечивает подключение памяти типам DDR SDRAM, соответствующей стандарту JESD79C, с параметрами:

- Организация – x8 или x16;
- Количество банков – 4;
- Разрядность адреса строки – не более 13;
- Разрядность адреса столбца – 9, 10, 11, 12;
- Задержка данных при чтении – 2, 2.5 и 3 такта.

Контроллер имеет следующие основные характеристики:

- Шина данных – 64 разряда;
- Шина адреса – 13 разрядов;
- Шина адреса банка – 2 разряда;
- Количество стробов DQS – 8;
- Количество стробов DM – 8;
- Количество каналов выдачи частоты синхронизации СК, СКп – 2;
- Программируемая установка параметров памяти;
- Программная и аппаратная подстройка “окна” приема данных;
- Аппаратный контроль открытия страниц.

### 9.2 Регистры DDR\_PORT

Состав регистров приведен в Таблица 9.1

Таблица 9.1 Регистры контроллера

Условное обозначение	Название регистра
DDR_CON	Регистр конфигурации DDRAM
DDR_BAR	Регистр базового адреса
DDR_CSR	Регистр управления и состояния
DDR_TMR	Регистр параметров DDRAM
DDR_MOD	Регистр режимов

При описании полей и значений регистров используются обозначения:

- R – только чтение;
- R0 – сброс при чтении;
- W1 – пуск операции, реальная запись не производится;
- RW – чтение и запись;
- RW1 – Чтение, пуск операции;
- ox – далее следует шестнадцатеричный код.

Термины и обозначения временных параметров и команд управления SDRAM соответствуют стандарту JESD79C.

### 9.2.1 Регистр конфигурации DDRAM

Формат регистра DDR\_CON приведен в Таблица 9.2

Таблица 9.2 Формат регистра DDR\_CON

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	-	Резерв	R	0
29:16	tRFR	Период авторегенерации DDRAM в тактах частоты СК	RW	0
15:13	-	Резерв	R	0
12	tWTR	Внутренняя задержка DDRAM между командами WRITEe и READ в тактах частоты СК: 0 – 1 такт СК; 1 – 2 такта СК	RW	0
11:9	-	Резерв.	R	0
8	DS	Мощность выходов микросхем DDRAM, подключенных к контроллеру ( Drive Strength): 0 – нормальная STTL class II; 1 – пониженная ( ~54% от нормальной )	RW	0
7	-	Резерв	R	0
6:4	CL	Задержка данных при чтении ( CAS latency ): 010 – 2 такта СК; 011 – 3 такта СК; 110 – 2.5 такта СК; 000:001, 100:101, 111 – резерв	RW	0
3:2	-	Резерв	R	0
1:0	PS	Размер страницы микросхем DDRAM, подключенных к контроллеру: 00 – 512; 01 – 1024; 10 – 2048; 11 – 4096. Число банков DDRAM – 4	RW	0

Преобразование физического адреса в адрес SDRAM представлено в таблицах 9.3-9.5. Разряды физического адреса обозначены строчными буквами “a” .

Таблица 9.3 Отображение адреса строки

PS	Адрес DDRAM												
	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
00	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14
01	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15
10	a28	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16
11	a29	a28	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17

Таблица 9.4 Отображение адреса столбца

PS	Адрес DDRAM												
	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
00	0	0	0	0	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3
01	0	0	0	a12	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3
10	0	a13	0	a12	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3
11	a14	a13	0	a12	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3

**Таблица 9.5 Отображение адреса банка**

PS	Адрес банка DDRAM	
	BA1	BA0
00	a13	a12
01	a14	a13
10	a15	a14
11	a16	a15

Период авторегенерации должен определяться индивидуально для используемой конфигурации DDRAM. Например, при тактовой частоте СК 200 МГц для обеспечения 8 192 цикловой регенерации за 64 мс необходимо в поле tRFR записать код 0x61A, что соответствует 7, 81 мкс на строку. При tRFR = 0 режим авторегенерации отключен.

### 9.2.2 Регистр базового адреса DDR\_BAR

Формат регистра DDR\_BAR приведен в Таблица 9.6

**Таблица 9.6 Формат регистра DDR\_BAR**

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	-	Резерв	R	0
15:8	CSBAR	Разряды 31:24 базового адреса DDRAM. Младшие разряды базового адреса равны нулю	RW	0xFF
7:0	CSMASK	Разряды 31:24 маски, используемые при определении базового адреса DDRAM. Младшие разряды маски равны нулю	RW	00

Физический адрес попадает в область памяти DDRAM, если PHA & CSMASK = CSBAR, где PHA – 32-разрядный физический адрес. Если физический адрес не попадает в область памяти DDRAM, то эта передача данных переадресуется в MPORT.

Минимальный размер сегмента – 16 Мбайт (при CSMASK = 0xFF). Для увеличения размера сегмента в младшие разряды поля CSMASK необходимо записать соответствующее число нулей. Например, для сегмента в 128 Мбайт, разряды 2:0 CSMASK должны быть равны нулю.

### 9.2.3 Регистр параметров DDRAM DDR\_TMR

Регистр DDR\_TMR предназначен для задания интервалов (в тактах частоты СК) между различными командами DDRAM.

Формат регистра DDR\_TMR приведен в Таблица 9.7.

**Таблица 9.7 Формат регистра DDR\_TMR**

Номер разряда	Условное Обозначение параметра	Назначение	Доступ	Исходное состояние
31:24	-	Резерв.	R	0
23:20	tRFC	Минимальный интервал между командами AUTO REFRESH.	RW	0
19:16	tRAS	Минимальная задержка между командами ACTIVE и PRECHARGE.	RW	0
15:14	-	Резерв.	R	0
13:12	tRTW	Дополнительная задержка команды READ после WRITE.	RW	0
11:10	-	Резерв.	R	0
9:8	tRCD	Минимальная задержка между командами ACTIVE и READ / WRITE.	RW	0
7:6	-	Резерв.	R	0
5:4	tRP	Минимальный период команд PRECHARGE	RW	0

3:2	-	Резерв.	R	0
1:0	tWR	Минимальная задержка между записью данных и командой PRECHARGE ( Write recovery )	RW	0

Значения 0, 1, ..., n параметра в таблице соответствуют интервалу в 1, 2, ..., n+1 тактов. Например, значение 0xF параметра tRFC задает интервал 16 тактов между командами AUTO REFRESH, а значение 0 – интервал в один такт.

При вычислении параметров в соответствии с рабочей частотой и со спецификацией используемой памяти, полученные значения необходимо округлять до ближайшего меньшего целого. Например, если в спецификации указано время tRCD = 20 ns, то при частоте СК 133 МГц ( период 7.5ns ) минимальный интервал в 2.7 такта нужно округлить до 2 и в поле tRCD регистра DDR\_TMR записать код 0x2.

### 9.2.4 Регистр состояний и управления DDR\_CSR

Регистр DDR\_CSR предназначен для запуска команд изменения режимов DDRAM или контроллера и индикации их исполнения. Формат регистра DDR\_CSR приведен в Таблица 9.8

Таблица 9.8 Формат регистра DDR\_CSR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:6	-	Резерв.	R	0
5	EYEW	Запись 1 в данный разряд запускает команду контроллера “подстройка частоты приема данных”. При чтении - признак окончания команды “подстройка частоты приема данных”: Устанавливается в 1 после завершения команды; сбрасывается при записи любой команды. Данная команда также запускается автоматически при выполнении команд инициализации и выхода из режима саморегенерации при сброшенном бите MODE регистра DDR_MOD. При этом бит EYEW не устанавливается, а биты INIT и EXIT устанавливается только после завершения подстройки частоты	RW	0
4	EXIT	Запись 1 в данный разряд запускает команду выхода DDRAM из режимов саморегенерации и пониженного потребления. При чтении - признак выполнения команды выхода DDRAM из режимов саморегенерации и пониженного потребления: устанавливается в 1 после завершения команды; сбрасывается при записи любой команды	RW	0
3	PWDN	Запись 1 в данный разряд запускает команду перевода DDRAM в режим пониженного потребления. При чтении - признак окончания данной команды: устанавливается в 1 после завершения команды; сбрасывается записью команды EXIT	RW	0
2	SREF	Запись 1 в данный разряд запускает команду перевода DDRAM в режим саморегенерации. При чтении - признак окончания данной команды: устанавливается в 1 после завершения команды; сбрасывается записью команды EXIT	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
1	AREF	При записи 1 в данный разряд контроллер выполняет команду регенерации DDRAM. При чтении - признак окончания данной команды: устанавливается в 1 после завершения команды; сбрасывается при записи любой команды	RW	0
0	INIT	Запись 1 в данный разряд запускает команду инициализации DDRAM с параметрами: Burst Length – 2; Burst Type – sequential; CAS Latency – поле CL регистра DDR_CON; Drive Strength – поле DS регистра DDR_CON. При чтении - признак окончания команды инициализации: устанавливается в 1 после завершения данной команды; сбрасывается при записи любой команды	RW	0

Команды кодируются унитарным кодом в разрядах 5-0. Запись других кодов или запись новой команды до завершения предыдущей игнорируются. Выражение “Запись 1 в данный разряд” в графе **Назначение** означает корректную запись унитарного кода с единицей в данном разряде.

При запуске любой команды изменения режимов DDR\_PORT ожидает завершения текущей операции и выполняет необходимую последовательность команд DDRAM.

Выполнение команд INIT, EYEW и EXIT зависит от состояния поля MODE регистра DDR\_MOD.

По команде AREF контроллер выполняет:

- PRECHARGE;
- пауза tRP, AUTO REFRESH;
- пауза tRFC, установка индикатора AREF.

По команде EYEW при MODE = 0 контроллер выполняет:

- запуск аппаратуры подстройки частоты приема данных;
- циклическое чтение памяти и ожидает (~30-40 тактов) окончания подстройки;
- завершает цикл чтения и устанавливает индикатор EYEW.

По команде EYEW при MODE = 1 контроллер выполняет запуск аппаратуры подстройки

частоты приема данных и устанавливает индикатор EYEW. В этом режиме аппаратура

аппаратуры подстройки выполняет сдвиг окна приема данных для каналов, заданных полем SEL регистра DDR\_MOD на величину 1/34 периода частоты СК.

В режиме MODE = 0 аппаратура подстройки автоматически центрирует окно приема данных в середину стробов DQS.

По команде INIT после специфицированной последовательности команд инициализации,

DDR\_PORT дополнительно выполняет

при MODE = 0:

- команду LOAD MODE REGISTER с битом DLL=0;
- пауза 200 тактов СК;
- команду EYEW и устанавливает индикатор INIT,



при  $MODE = 1$ :

- команду  $LOAD\ MODE\ REGISTER$  с битом  $DLL=0$ ;
- пауза  $tMRD$  тактов  $CK$  и устанавливает индикатор  $INIT$ .

До выполнения начальной инициализации необходимо записать все параметры в регистры  $DDR\_CON$ ,  $DDR\_TMR$  и  $DDR\_MOD$ .

$DDR\_PORT$  не контролирует выполнение интервала 200 мкс между установкой стабильного питания и командой  $INIT$ .

По команде  $PWDN$  контроллер выполняет:

- $PRECHARGE$ ;
- Пауза  $tRP$ ,  $AUTO\ REFRESH$ ;
- Пауза 1 такт  $CK$ ;
- Сброс  $SKE$ ;
- Пауза  $tRFC$ , установка индикатора  $PWDN$ .

После выполнения данной команды память находится в режиме  $precharge\ power-down$ .

Аналогично выполняется команда  $SREF$ . Отличие в том, что сброс  $SKE$  происходит одновременно с  $AUTO\ REFRESH$  и устанавливается индикатор  $SREF$ .

После выполнения команд  $PWDN$  и  $SREF$  контроллер находится в состоянии останова до выполнения команды  $EXIT$ . В этом состоянии  $DDR\_PORT$  не контролирует выполнение интервала  $tREFC$ .

По команде  $EXIT$  контроллер устанавливает  $SKE$  и после паузы  $tXSNR$  выполняет  $AREF$ .

При выходе из  $PWDN$  или из  $SREF$  при  $MODE = 1$  команда на этом завершается установкой индикатора  $EXIT$ , а при выходе из  $SREF$  при  $MODE = 0$ , контроллер ждет 200 тактов  $CK$ , выполняет команду  $EYEW$  и устанавливает индикатор  $EXIT$ .

$DDR\_PORT$  игнорирует команду  $EXIT$  при сброшенных индикаторах  $PWDN$  и  $SREF$ .

### 9.2.5 Регистр режимов $DDR\_MOD$

Регистр  $DDR\_MOD$  предназначен для управления аппаратурой настройки окна приема данных и задания режимов выполнения специальных команд. Формат регистра  $DDR\_MOD$  приведен в Таблица 9.9.

Таблица 9.9 Формат регистра  $DDR\_MOD$

Номер разряда	Условное обозначение параметра	Назначение	Доступ	Исходное состояние
31:16	$tEYE$	Период автоподстройки частоты приема данных в циклах $tRFR$	RW	0
15	$MODE$	Режим выполнения специальных команд: 0- автоматический; 1- пошаговый.	RW	0
14:4	-	Резерв.	R	0
3:0	$SEL$	Выбор канала приема данных в пошаговом режиме: $SEL[i] = 1$ –сдвиг окна для строба $DQS[i]$ разрешен, $i = 0, 1, 2, 3$ ; $SEL[i] = 0$ – сдвиг окна для строба $DQS[i]$ запрещен, $i = 0, 1, 2, 3$	RW	0

При  $MODE = 0$  контроллер аппаратно выполняет команду  $EYEW$  через каждые  $tRFR$  \*  $tEYE$  тактов частоты  $CK$ .

При  $tEYE = 0$  или  $tRFR = 0$  режим автоподстройки частоты приема данных отключен.

## 10. УНИВЕРСАЛЬНЫЙ АСИНХРОННЫЙ ПОРТ (UART)

### 10.1 Общие положения

Универсальный асинхронный порт (далее UART) имеет следующие характеристики:

- по архитектуре совместим с UART 16550;
- частота приема и передачи данных – от 50 до 1 Мбод;
- FIFO для приема и передачи данных имеют объем по 16 байт;
- полностью программируемые параметры последовательного интерфейса: длина символа от 5 до 8 бит; генерация и обнаружение бита четности; генерация стопового бита длиной 1, 1.5 или 2 бита;
- диагностический режим внутренней петли;
- эмуляция символьных ошибок;
- функция управления модемом (CTS, RTS, DSR, DTR, RI, DCD).

Структурная схема порта UART приведена на Рисунок 10.1.

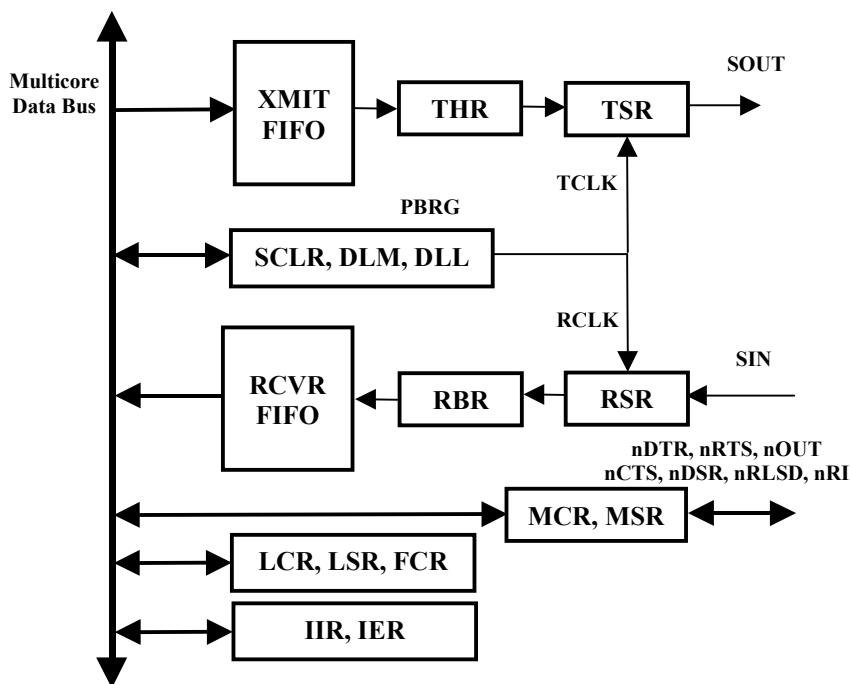


Рисунок 10.1. Структурная схема UART.

Передаваемые данные записываются в регистр THR, а затем аппаратно переписываются в передающий сдвигающий регистр (TSR), если он пуст. После этого в регистр THR могут быть записаны следующие данные.

После приема данных в приемный сдвигающий регистр (RSR) данные переписываются в регистр RBR, если он не занят.

Назначение внешних выводов UART приведено в Таблица 10.1.

**Таблица 10.1. Внешние выводы UART.**

Название вывода	Тип вывода	Описание
SIN	I	Вход последовательных данных
SOUT	O	Выход последовательных данных
nDTR	O	Готовность UART к установлению связи (Data Terminal Ready)
nRTS	O	Готовность UART к обмену данными (Request To Send)
nOUT1	O	Выход общего назначения
nOUT2	O	Выход общего назначения
nCTS	I	Готовность модема к обмену данными (Clear To Send)
nDSR	I	Готовность модема к установлению связи (Data Set Ready)
nDCD	I	Признак обнаружения модемом несущей частоты (Receiver Line Signal Detect)
nRI	I	Признак обнаружения модемом телефонного звонка (Ring Indicator)

## 10.2 Регистры UART

### 10.2.1 Общие положения

Перечень регистров UART приведен в Таблица 10.2.

**Таблица 10.2. Перечень регистров UART**

Условное обозначение регистра	Название регистра	Смещение	Доступ (R-чтение, W-запись)
RBR	Приемный буферный регистр	0 (DLAB=0)	R
THR	Передающий буферный регистр	0 (DLAB=0)	W
IER	Регистр разрешения прерываний	1 (DLAB=0)	R/W
IIR	Регистр идентификации прерывания	2	R
FCR	Регистр управления FIFO	2	W
LCR	Регистр управления линией	3	R/W
MCR	Регистр управления модемом	4	R/W
LSR	Регистр состояния линии	5	R
SPR	Регистр Scratch Pad	7	R/W
DLL	Регистр делителя младший	0 (DLAB=1)	R/W
DLM	Регистр делителя старший	1 (DLAB=1)	R/W
SCLR	Регистр предделителя (scaler)	5	W

## 10.2.2 Регистр LCR

Формат регистра LCR приведен в Таблица 10.3.

Таблица 10.3. Формат регистра LCR

Номер бита	Условное обозначение	Назначение
1:0	WLS (Word Length Select)	Количество бит данных в передаваемом символе: 00 -5 бит, 01 -6 бит, 10 -7 бит, 11 -8 бит.
2	STB (Number Stop Bits)	Количество стоп-бит: 0 - 1 стоп-бит, 1 - 2 стоп-бита (для 5-битного символа стоп-бит имеет длину 1,5 бита). Приемник анализирует только первый стоп бит.
3	PEN (Parity Enable)	Разрешение генерации (передатчик) или проверки (приемник) контрольного бита: 1 – контрольный бит (паритет или постоянный) разрешен, 0 – запрещен.
4	EPS (Even Parity Select)	Выбор типа контроля (при PEN=1): 0 – нечетность, 1 – четность.
5	STP (Stick Parity)	Принудительное формирование бита паритета: 0 – контрольный бит генерируется в соответствии с паритетом выводимого символа, 1 – постоянное значение контрольного бита: при EPS=1 - нулевое, при EPS=0 – единичное.
6	SBC (Set Break Control)	Формирование обрыва линии: 0 – нормальная работа; 1 – на выходе SOUT устанавливается низкий уровень (Spacing level). Это влияет только на выход SOUT, а не на логику передачи символа.
7	DLAB (Divisor Latch Access bit)	Управление доступом к регистрам: 0 – разрешен доступ к регистрам RBR, THR, IER; 1 – разрешен доступ к регистрам DLL, DLM

Исходное состояние регистра LCR – нули.

Бит SBC используется как признак «Внимание» для приемного терминала, подключенному к выходу UART. Для того чтобы не было передано ошибочного символа при использовании бита SBC, необходимо выполнять следующую последовательность действий:

- Загрузить в регистр THR все нули по признаку THRE=1;
- Установить SBC=1 по следующему THRE=1;
- Дождаться TEMT=1.

Для восстановления нормальной передачи необходимо установить SBC=0.

### 10.2.3 Регистр FCR

Формат регистра FCR приведен в Таблица 10.4.

**Таблица 10.4. Формат регистра FCR**

Номер бита	Условное обозначение	Назначение
0	FEWO (FIFO Enable)	Разрешение работы XMIT и RCVR FIFO: 0 – символьный режим; 1 – режим FIFO. При изменении состояния этого бита, данные из FIFO, не удаляются. Запись в биты RFR, TFR, RFTL выполняется, если FEWO=1.
1	RFR (Receiver FIFO Reset)	Установка RCVR FIFO в исходное состояние. Регистр RSR не обнуляется. После записи 1 в этот бит он автоматически сбрасывается.
2	TFR (Transmitter FIFO Reset)	Установка XMIT FIFO в исходное состояние. Регистр TSR не обнуляется. После записи 1 в этот бит он автоматически сбрасывается.
5:3	-	Резерв
7:6	RFTL (RCVR FIFO Trigger Level)	Порог заполнения RCVR FIFO (в байтах), при котором формируется прерывание: 00 – 1; 01 – 4; 10 – 8; 11 – 14.

Исходное состояние регистра FCR – нули.

### 10.2.4 Регистр LSR

Формат регистра LSR приведен в Таблица 10.5.

**Таблица 10.5. Формат регистра LSR**

Номер бита	Условное обозначение	Назначение
0	RDR (Receiver Data Ready)	Готовность данных. Устанавливается после приема символа данных и передачи его в регистр RBR или FIFO. Сбрасывается после чтения регистра RBR (в символьном режиме) или чтения всего содержимого RCVR FIFO (в режиме FIFO)
1	OE (Overrun Error)	Ошибка переполнения. Устанавливается, если содержимое регистра RBR не было прочитано, в сдвигающий регистр принят следующий символ и начат прием очередного символа. При этом новый символ записывается в сдвигающий регистр вместо старого. В режиме FIFO устанавливается, если после перехода порогового (trigger) уровня FIFO заполнено до конца, во входной сдвигающий регистр полностью принят следующий символ и начат прием очередного символа. При этом в FIFO ничего не передается. Бит сбрасывается при чтении содержимого регистра LSR.
2	PE (Parity Error)	Ошибка контрольного бита (паритета или фиксированного). В режиме FIFO этот бит показывает на ошибку в символе, находящемся наверху FIFO. Бит сбрасывается при чтении содержимого регистра LSR.
3	FE (Framing Error)	Ошибка кадра. Устанавливается, если стоп-бит равен нулю (Spacing level). В режиме FIFO этот бит показывает на ошибку в символе, находящемся наверху FIFO. После этой ошибки UART пересинхронизируется. Бит сбрасывается при чтении содержимого регистра LSR.
4	BI (Break Interrupt)	Обрыв линии. Устанавливается, если вход приема данных находится в состоянии 0 (Spacing level) не менее чем время передачи всего символа. В режиме FIFO этот бит показывает на ошибку в символе, находящемся наверху FIFO. При возникновении этой ситуации, в FIFO загружается только один нулевой символ. Прием следующих символов разрешается после того, как вход приема данных перейдет в единичное состояние (Marking state) и будет принят действительный стартовый бит. Бит сбрасывается при чтении содержимого регистра LSR.
5	THRE (Transmitter Holding Register Empty)	Передающий буферный регистр пуст. Показывает, что UART готов принять следующий символ для передачи. Устанавливается, когда содержимое регистра THR передается в передающий сдвигающий регистр. Одновременно с этим генерируется прерывание THREI, если оно разрешено. Бит сбрасывается при записи символа в регистр THR. В режиме FIFO этот бит устанавливается, когда XMIT FIFO пусто, и сбрасывается, если в XMIT FIFO записывается хотя бы один символ.
6	TEMT (Transmitter Empty)	Передатчик пуст. Устанавливается, если регистры THR и TSR пусты. Имеет нулевое состояние, если хотя бы один из регистров THR и TSR не пуст. В режиме FIFO этот бит устанавливается, если нет символов ни в XMIT FIFO, ни в регистре TSR.
7	EIRF (Error in RCVR FIFO)	Наличие хотя бы одного признака ошибки в FIFO. В символьном режиме этот бит всегда равен нулю. Бит сбрасывается при чтении содержимого регистра LSR, если в FIFO нет больше признаков ошибок.

Исходное состояние бит THRE, TEMT – 1, остальных – 0.

Установка бит OE, PE, FE, VI приводит к формированию прерыванию по состоянию входа приема данных (Receiver Line Status Interrupt), если это прерывание разрешено.

### 10.2.5 Регистр IER

Формат регистра IER приведен в Таблица 10.6. Исходное состояние регистра IER – нули.

Таблица 10.6. Формат регистра IER

Номер бита	Условное обозначение	Назначение
0	ERBI	Разрешение прерывания по наличию принятых данных (RDAI), а также по таймауту (CTI)
1	ETBEI	Разрешение прерывания по отсутствию данных в регистре THR (THREI)
2	ERLSI	Разрешение прерывания по статусу приема данных (RLSI)
3	EMSI	Разрешение прерывания по статусу модема (MSI)
7:4	-	Резерв

### 10.2.6 Регистр IIR

Формат регистра IIR приведен в Таблица 10.7.

Таблица 10.7. Формат регистра IIR

Номер бита	Условное обозначение	Назначение
0	IP (Interrupt Pending)	Признак наличия прерывания: 0 – есть прерывание; 1 – нет прерывания.
3:1	IID[2:0]	Код идентификации прерывания в соответствии с Таблица 10.8.
5:4	-	Резерв
7:6	FE	Признак разрешения работы RCVR и XMIT FIFO

Исходное состояние бита IP – 1, остальных – 0.

**Таблица 10.8. Идентификация прерываний**

Код поля ID[2:0]	Уровень приоритета (1 – наивысший)	Тип прерывания	Причина прерывания	Условие сброса прерывания
011	1	Статус приема данных (RLSI – Receiver Line Status Interrupt)	OE - Overrun Error; PE - Parity Error; FE - Framing Error; BI - Break Interrupt.	Чтение содержимого регистра LSR. Чтение из FIFO символа, по которому сформировано это прерывание. Обнуление FIFO.
010	2	Наличие принятых данных (RDAI – Received Data Available Interrupt)	Наличие данных в регистре RBR или достижение заданного порога FIFO	Чтение содержимого регистра RBR. Считывание данных из FIFO до уровня ниже порогового.
110	2	Таймаут (CTI – Character Timeout Interrupt)	С момента приема последнего символа в RCVR FIFO прошло время, равное длительности передачи 4-х символов и не было ни чтения FIFO, ни приема очередного символа.	Чтение содержимого регистра RBR. Прием очередного символа. Сброс FIFO.
001	3	Регистр THR пуст (THREI – Transmitter Holding Register Empty Interrupt)	Регистр THR пуст	Чтение содержимого регистра IIR, если источником прерывания является это условие. Запись символа в регистр THR
000	4	Статус модема (MSI – Modem Status Interrupt)	Изменение состояния сигналов на входах порта nCTS, nDSR, nRI, nDCD	Чтение содержимого регистра MSR.



### 10.2.7 Регистр MCR

Формат регистра MCR приведен в Таблица 10.9.

Таблица 10.9. Формат регистра MCR

Номер бита	Условное Обозначение	Назначение
0:3	-	Не используется
4	LOOP	Режим петли. Используется для тестирования UART. При установке этого бита в 1 выполняется следующее: На выходе SOUT UART устанавливается высокий уровень; Вход SIN UART отключается от внешнего вывода; Выход регистра TSR подключается к входу регистра RSR; В режиме петли передаваемые данные немедленно принимаются. В режиме петли все прерывания формируются как обычно.
7:5	-	Не используется

Исходное состояние регистра MCR – нули.

### 10.2.8 Программируемый генератор скорости обмена

В UART имеется программируемый генератор скорости обмена данными (PBRG – Programmable Baud Rate Generator). Он состоит из 8-разрядного предделителя и 16-разрядного основного делителя частоты. На вход предделителя поступает системная тактовая частота CLK, на которой работает CPU, UART и другие устройства. Выходная частота предделителя поступает на вход основного делителя. Выходная частота генератора PBRG в 16 раз больше частоты обмена последовательными данными.

Значение частоты на выходе предделителя равно  $CLK/(SCLR + 1)$ . Коэффициент деления основного делителя задается 16-разрядным регистром, который является конкатенацией регистров DLM и DLL.

Период частот передачи и приема (TCLK и RCLK) UART вычисляется по формуле:

$CLK/(SCLR + 1) / ((\text{конкатенация содержимого регистров DLM и DLL}) * 16)$ . Минимальная величина, которая может быть записана в регистры {DLM, DLL}, равна 1.

Исходное состояние регистров DLL, DLM, SCLR – нули.

## 10.3 Работа с FIFO по прерыванию

Если установлен режим работы с FIFO (EFWO=1 в регистре FCR) и разрешены прерывания по приему (бит ERI=1 в регистре IER), то в процессе приема:

- формируется прерывание, если число символов в RCVR FIFO достигло запрограммируемого порога. Это прерывание сбрасывается, если при чтении из FIFO число символов оставшихся в нем, станет меньше запрограммируемого порога;
- одновременно с этим в регистре IIR устанавливается индикатор наличия принятых данных RDAI. Индикатор обнуляется, при чтении из FIFO до снижения запрограммируемого порога;
- может возникнуть прерывание по статусу приема данных (RLSI), приоритет которого выше, чем RDA;
- бит RDR в регистре LSR устанавливается в момент передачи символа из регистра RSR в RCVR FIFO. Этот бит обнуляется при считывании из FIFO всех символов данных.

Если установлен режим работы с FIFO (EFWO=1 в регистре FCR) и разрешены прерывания по приему (ERI=1 в регистре IER), то генерируется прерывание по таймауту, если с момента приема последнего символа в RCVR FIFO прошло время, равное длительности передачи 4-х символов и за это время не было:

- ни чтения RCVR FIFO;
- ни приема в RCVR FIFO очередного символа.

При 12-битном символе и скорости передачи 300 бод, прерывание по этой причине возникнет через 160 мс.

При возникновении прерывания по таймауту оно обнуляется при считывании символа из RCVR FIFO. При этом обнуляется и таймер, генерирующий данное прерывание. Если прерывание по таймауту не возникло, то таймер таймаута обнуляется при приеме нового символа или при считывании символа из RCVR FIFO.

Если установлен режим работы с FIFO (EFWO=1 в регистре FCR) и разрешены прерывания по передаче данных (бит ETI=1 в регистре IER), то генерируется прерывание по передаче следующим образом:

- формируется прерывание THREI, если XMIT FIFO пусто. Это прерывание обнуляется, как только выполняется запись символа в регистр THR (при приеме данного прерывания в XMIT FIFO можно записать от 1 до 16 символов);
- индикатор TEMT в регистре LSR установится в единичное состояние через время равное длительности одного символа минус последний стоп бит, после установки THRE=1. Первое прерывание по передаче (если оно разрешено) формируется немедленно после установки EFWO=1.

#### 10.4 Работа с FIFO по опросу

Если установлен режим работы с FIFO (EFWO=1 в регистре FCR) и запрещены прерывания, то обмен данными выполняется по опросу, а управление FIFO приема и передачи (RCVR, XMIT) выполняется раздельно.

В этом режиме опрос состояния RCVR и XMIT FIFO осуществляется программно, посредством считывания содержимого регистра LSR:

- бит RDR=1, пока есть данные в RCVR FIFO;
- биты OE, PE, FE, VI указывают на ошибки. Эти ошибки обрабатываются так же, как и при работе по прерыванию;
- бит THRE=1, если XMIT FIFO пусто;
- бит TEMT=1, если в XMIT FIFO и TSR нет данных.

При работе по опросу нет индикации таймаута и факта достижения порога RCVR FIFO. Однако оба RCVR и XMIT FIFO могут хранить символы данных.

## 11. КОНТРОЛЛЕР ИНТЕРФЕЙСА Serial RapidIO (SRIO)

### 11.1 Общие положения

Контроллер интерфейса Serial RapidIO (SRIO) соответствует следующим стандартам:

- RapidIO Interconnect Specification V1.2 Part I: Input/Output Logical Specification;
- RapidIO Interconnect Specification V1.2 Part II: Message Passing Logical Specification;
- RapidIO Interconnect Specification V1.2 Part III: Common Transport Specification;
- RapidIO Interconnect Specification V1.2 Part VI: Physical Layer 1x/4x
- LP-Serial Specification.

Контроллер SRIO не реализует функциональные параметры и возможности по стандарту «RapidIO Interconnect Specification V1.2 Part V: Globally Shared Memory Logical Specification».

Контроллеры SRIO имеют следующие функциональные параметры и возможности:

- \* 1-канальный порт с автоматической адаптацией на одноканальную работу;
- \* аппаратная обработка ошибок, включая проверку CRC (Cyclic Redundancy Code);
- \* дифференциальные приемопередатчики с поддержкой возможности развязки по постоянному току;
- \* скорость передачи - 1,25 Гбод;
- \* режим энергосбережения для неиспользуемых каналов;
- \* режим энергосбережения для всего порта при его не использовании;
- \* выполнение операций NREAD, NWRITE, WRITE\_R, SWRITE, ATOMIC, MESSAGE, DOORBELL, PORT\_WRITE, MAINTENANCE;
- \* поддержка 8 и 16-разрядных device ID;
- \* поддержка 34 разрядного адреса при приеме пакетов;
- \* поддержка 34 и 66 разрядного адреса при передаче пакетов;
- \* прием и передача сообщений, содержащих до 16 пакетов;
- \* поддержка расширения Error Management Extensions;
- \* поддержка Congestion Control Extensions;
- \* поддержка четырёх multicast ID.

В микропроцессоре имеется два контроллера SRIO: SRIO0 и SRIO1.

Для обеспечения гальванической развязки к дифференциальным входам SRIO необходимо последовательно подключать конденсаторы номиналом 0,01 мкФ, 10 В.

Обмен данными между интерфейсом Serial RapidIO и AXI Switch SRIO выполняет в режиме прямого доступа в память.

## 11.2 Структурная схема

Структурная схема контроллера SRIO приведена на Рисунок 11.1.

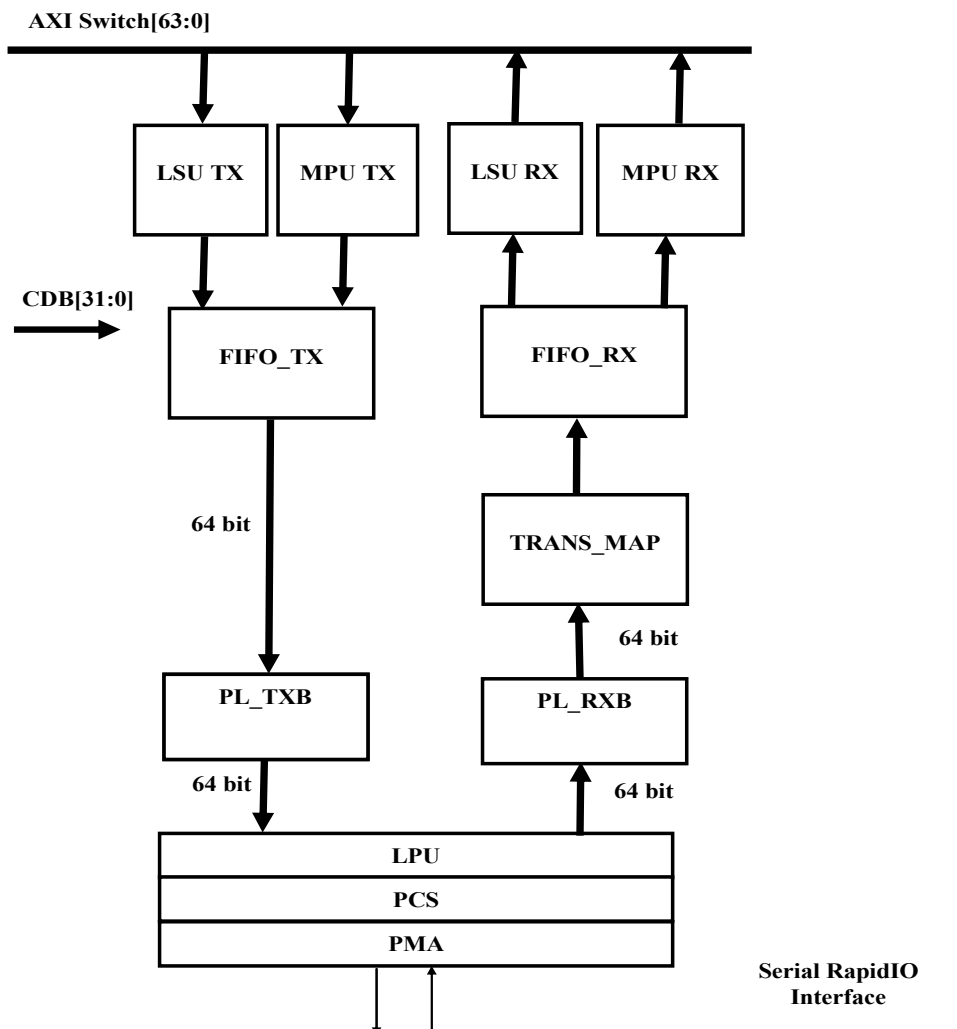


Рисунок 11.1. Структурная схема контроллера SRIO.

В состав SRIO входят следующие основные узлы:

- \* LSU\_TX (LSU – Load-Store Unit) – устройство передачи пакетов ввода-вывода;
- \* LSU\_RX (LSU – Load-Store Unit) – устройство приема пакетов ввода-вывода;
- \* MPU\_TX (MSU – Message Passing Unit) – устройство передачи сообщений;
- \* MPU\_RX (MSU – Message Passing Unit) – устройство приема сообщений;
- \* FIFO\_TX, FIFO\_RX – буфера типа FIFO для передачи и приема пакетов;
- \* TRANS\_MAP (Transaction Mapping) – устройство анализа заголовка входных пакетов;
- \* PL\_TXB, PL\_RXB (PL – Physical Layer) – приемный и передающий буфера физического уровня RapidIO;
- \* LPU (Link Protocol Unit) – устройство реализации протокола канала связи;
- \* PCS (Physical Coding Sublayer) – устройство подуровня кодирования;
- \* PMA (Physical Media Attachment) – устройство сопряжения со средой передачи данных.

## 11.3 Регистры SRIO

### 11.3.1 Общие положения

Перечень регистров SRIO приведен в Таблица 11.1.

Таблица 11.1. Перечень регистров SRIO

Условное обозначение регистра	Название регистра	Адрес относительно базового
Регистры устройства приема и передачи пакетов ввода-вывода (LSU)		
LSU0_CR0	Регистр управления 0 LSU0	000
LSU0_CR1	Регистр управления 1 LSU0	004
LSU0_CR2	Регистр управления 2 LSU0	008
LSU0_CR3	Регистр управления 3 LSU0	00C
LSU0_CR4	Регистр управления 4 LSU0	010
LSU0_CR5	Регистр управления 5 LSU0	014
LSU0_CR6	Регистр управления 6 LSU0	018
LSU1_CR0	Регистр управления 0 LSU1	020
LSU1_CR1	Регистр управления 1 LSU1	024
LSU1_CR2	Регистр управления 2 LSU1	028
LSU1_CR3	Регистр управления 3 LSU1	02C
LSU1_CR4	Регистр управления 4 LSU1	030
LSU1_CR5	Регистр управления 5 LSU1	034
LSU1_CR6	Регистр управления 6 LSU1	038
LSU2_CR0	Регистр управления 0 LSU2	040
LSU2_CR1	Регистр управления 1 LSU2	044
LSU2_CR2	Регистр управления 2 LSU2	048
LSU2_CR3	Регистр управления 3 LSU2	04C
LSU2_CR4	Регистр управления 4 LSU2	050
LSU2_CR5	Регистр управления 5 LSU2	054
LSU2_CR6	Регистр управления 6 LSU2	058
LSU3_CR0	Регистр управления 0 LSU3	060
LSU3_CR1	Регистр управления 1 LSU3	064
LSU3_CR2	Регистр управления 2 LSU3	068
LSU3_CR3	Регистр управления 3 LSU3	06C
LSU3_CR4	Регистр управления 4 LSU3	070
LSU3_CR5	Регистр управления 5 LSU3	074
LSU3_CR6	Регистр управления 6 LSU3	078
LSU_IRQ_SR	Регистр состояния запросов прерывания LSU	080
LSU_IRQ_CLR	Регистр обнуления запросов прерывания LSU	084
PORT_WRITE_SR	Регистр состояния буфера FIFO PORT_WRITE	088
PORT_WRITE_FIFO	Выходной регистр буфера FIFO PORT_WRITE	08C
IN_FLTR0	Регистр фильтрации входящих операций ввода данных	090
IN_FLTR1	Регистр фильтрации входящих операций ввода данных	094
MNT_WR_IRQ	Регистр состояния запроса прерывания по входной операции Maintenance Write	098

Продолжение Таблица 11.1

Условное обозначение регистра	Название регистра	Адрес относительно базового
Регистры устройства операций приема и передачи сообщений (MPU)		
RXU_MAP_L0	Регистр 0 отображения номера почтового ящика на номер очереди (low)	100
RXU_MAP_H0	Регистр 0 отображения номера почтового ящика на номер очереди (high)	104
...		
RXU_MAP_L31	Регистр 31 отображения номера почтового ящика на номер очереди (low)	1F8
RXU_MAP_H31	Регистр 31 отображения номера почтового ящика на номер очереди (high)	1FC
RXQ_HDP0	Указатель на первый дескриптор очереди принимаемых сообщений 0	200
RXQ_CDP0	Указатель на последний обработанный дескриптор очереди принимаемых сообщений 0	204
...		
RXQ_HDP15	Указатель на первый дескриптор очереди принимаемых сообщений 15	278
RXQ_CDP15	Указатель на последний обработанный дескриптор очереди принимаемых сообщений 15	27C
TXQ_HDP0	Указатель на первый дескриптор очереди передаваемых сообщений 0	280
TXQ_CDP0	Указатель на последний обработанный дескриптор очереди передаваемых сообщений 0	284
...		
TXQ_HDP15	Указатель на первый дескриптор очереди передаваемых сообщений 15	2F8
TXQ_CDP15	Указатель на последний обработанный дескриптор очереди передаваемых сообщений 15	2FC
RX_CR	Регистр режима приема пакетов в много пакетных сообщениях	300
RX_QTCR	Регистр команд прекращения приема сообщений в данную очередь	304
TX_QUEUE_CTR0	Регистр 0 управления передачей из очередей сообщений	308
TX_QUEUE_CTR1	Регистр 1 управления передачей из очередей сообщений	30C
TX_QUEUE_CTR2	Регистр 2 управления передачей из очередей сообщений	310
TX_QUEUE_CTR3	Регистр 3 управления передачей из очередей сообщений	314
TX_QTCR	Регистр команд прекращения передачи сообщений	318
MPU_IRQ_SR	Регистр состояния запросов прерывания MPU	31C
DOORBELL_FIFO_LOW	Выходной регистр DOORBELL_FIFO_LOW	320
DOORBELL_FIFO_HIGN	Выходной регистр DOORBELL_FIFO_HIGH	324

Продолжение Таблица 11.1

Условное обозначение регистра	Название регистра	Адрес относительно базового
Системные регистры		
SRIO_CSR	Регистр управления и состояния SRIO	328
MUL-TICAST_DEVICE_ID0	Регистр 0 идентификатора устройства для входных пакетов типа Multicast	32C
MUL-TICAST_DEVICE_ID1	Регистр 1 идентификатора устройства для входных пакетов типа Multicast	330
MUL-TICAST_DEVICE_ID2	Регистр 2 идентификатора устройства для входных пакетов типа Multicast	334
MUL-TICAST_DEVICE_ID3	Регистр 3 идентификатора устройства для входных пакетов типа Multicast	338
TEST_FR	Регистр тестовых полей пакета	33C
TIMER_COUNT	Счетчик таймера ответных пакетов	340
PACKET_FR	Регистр запоминания полей входных пакетов. Используется для тестирования	344
EXTND_ADDR	Регистр запоминания поля extended address входных пакетов. Используется для тестирования	348
Архитектурные регистры логического и транспортного уровней RapidIO		
DEV_ID_CAR	Device Identity Capability Register	400
DEV_INFO_CAR	Device Information Capability Register	404
ASBLY_ID_CAR	Assembly Identity Capability Register	408
ASBLY_INFO_CAR	Assembly Information Capability Register	40C
PE_FEATURES_CAR	Processing Element Features Capability Register	410
SRC_OP_CAR	Source Operation Capability Register	418
DEST_OP_CAR	Destination Operation Capability Register	41C
PE_LOG_CSR	Processing Element Logical Layer Control CSR	44C
BASE_DEVICE_ID_CSR	Base Device ID Command and Status Register	460
HOST_BASEID_LOCK_CSR	Host Base Device ID Lock Command and Status Register	468
COPM_TAG_CSR	Component TAG Command and Status Register	46C
Архитектурные регистры физического уровня RapidIO		
BLOCK_HEADER	1x/4x LP-Serial Register Block Header	500
LINK_TIMEOUT	Port Link Time-out Control CSR	520
RESP_TIMEOUT	Port Response Time-out Control CSR	524
GENERAL_CSR	Port General Control CSR	53C
ERROR_STATUS_CSR	Port Error and Status CSR	558
CONTROL_CSR	Port Control CSR	55C
Дополнительные регистры физического уровня (из RapidIO не доступны)		
PCS_CSR	Регистр статуса и управления блока PCS	560
LPU_CSR	Регистр статуса и управления блока LPU	564
LPU_ERROR	Регистр ошибок приёма/передачи блока LPU	568
ACKID_CSR	Регистр контроля AckId переданных и подтверждённых пакетов: TBUF_STAT, TLAST_ACK, TNEXT_ACK. Используется для целей тестирования и контроля	56C
USER_SYMBOL	Регистр выдачи управляющих символов	570
RX_SYMBOL	Регистр хранения управляющего символа принятого от соседнего устройства	574
TRB_CSR	Регистр статуса, управления и тестирования буфера приёма и передачи пакетов TRB	578

Условное обозначение регистра	Название регистра	Адрес относительно базового
TXB_TEST	Регистр тестирования буфера передачи пакетов PL_TXB. Опрос внутренних регистров буфера	57C
LINK_TIMER_COUNT	Регистр таймера ответных символов. Используется для тестирования	580
LPU_PCS_TEST	Регистр диагностики и тестового управления блоков LPU_PCS	584
RX_BUF_CSR	Регистр управления буферами: RX_BUF0, RX_BUF1, RX_BUF2, RX_BUF3 в PCS по каждому каналу	588
RX_BUF0	Буфер приема символов от приёмника PMA_RX0	58C
RX_BUF1	Буфер приема символов от приёмника PMA_RX1	590
RX_BUF2	Буфер приема символов от приёмника PMA_RX2	594
RX_BUF3	Буфер приема символов от приёмника PMA_RX3	598
TX_CODE0	Регистр передачи символа на передатчик PMA_TX0	59C
TX_CODE1	Регистр передачи символа на передатчик PMA_TX1	5A0
TX_CODE2	Регистр передачи символа на передатчик PMA_TX2	5A4
TX_CODE3	Регистр передачи символа на передатчик PMA_TX3	5A8

В данном разделе используются следующие обозначения типа доступа:

- R – только чтение;
- RW – чтение и запись;
- W1 – пуск операции. Реальная запись не производится;
- RW1C – Чтение, запись 1 для сброса.

Базовый адрес SRIO0 – 182F\_A000, SRIO1 – 182F\_B000.



## 11.3.2 Регистры системные

### 11.3.2.1 Регистр SRIO\_CSR

Формат регистра SRIO\_CSR (SRIO Command and Status Register) приведен в Таблица 11.2.

Таблица 11.2 Формат регистра SRIO\_CSR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	MPU_TX_INT	Прерывание от MPU_TX (узел передачи пакетов message)	R	0
30	MPU_RX_INT	Прерывание от MPU_RX (узел приёма пакетов message)	R	0
29	LSU_INT	Прерывание от LSU (4-ре узла передачи пакетов input/output)	R	0
28	DRBL_INT	Признак наличия прерывания от порта приема пакетов DOORBELL. Повторяет состояние разряда NEMPTY регистра DOORBEEL_FIFO_LOW.	R	0
27	MNT_INT	Признак наличия прерывания MAINTENANCE PORT-WRITE либо MAINTENANCE WRITE. Объединяет по “ИЛИ” разряд NEMPTY регистра PORT_WRITE_SR и разряд MNT_WR_INT регистра MNT_WR_IRQ.	R	0
26	PORT_ERROR	Признак наличия прерывания из-за того, что LPU находится в нерабочем состоянии. Это происходит из-за обнаружения невосстанавливаемой ошибки, либо переполнился счетчик повторов символов “Packet Not Accepted”. Объединяет по “ИЛИ” разряд PORT_ERROR регистра ERROR_STATUS_CSR и разряд PNA_CNT_OVER регистра LPU_CSR	R	0
25	RST_DEV_CMD	Поступили 4 команды Reset-Device Command. Повторяет состояние одноименного бита регистра LPU_CSR	R	0
24	MCE_DEC	Признак того, что LPU принял символ Multicast-Event. Повторяет состояние одноименного бита регистра LPU_CSR	R	0
23:21	WN	Число слов, на которое DMA SRIO захватывает коммутатор AXI при передаче данных: 000 – 1; 001 – 2; 010 – 4; 011 – 8; 100:111 – 16.	RW	0
20	BUSY_DMA	Внутренний контроллер DMA SRIO выполняет обмен.	R	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
19:16	PRESCALER	Коэффициент деления частоты CLK для тактирования таймеров ответных пакетов логического уровня (регистр TIMER_COUNT): 0000 – 1; 0001 – 2; ... 1111 – 16.	RW	0
15	TICK_TIMER	Если EN_TIMER = 0, то при записи 1 в этот разряд выполняется программная инкрементация таймера ответных пакетов на 1. Считывается всегда ноль	W1	0
14	EN_TIMER	Разрешение работы таймера: 0 – работа таймера запрещена. В этом случае таймер инкрементируется на 1 при записи 1 в разряд TICK_TIMER; 1 – работа таймера разрешена.	RW	0
13	TEST_RESPONSE	Разрешение передачи тестовых ответных пакетов: 0 – нормальный режим работы. Пакеты ответа передаются в соответствии с протоколом; 1 – тестовый режим работы. Пакеты ответа передаются с подменной полем из регистра TEST_FR. TEST_RESPONSE используется только в узлах LSU_RX и MPU_RX.	RW	0
12	TEST_REQUEST	Разрешение передачи тестовых пакетов запроса: 0 – нормальный режим работы. Пакеты запроса передаются в соответствии с заданием; 1 – тестовый режим работы. Пакеты запроса передаются в соответствии с заданием, но с подменной полем из регистра TEST_FR. TEST_REQUEST используется только в узле LSU_TX0.	RW	0
11	DSBL_RESPONSE	Запрещение передачи ответного пакета: 0 – ответный пакет передается; 1 – ответный пакет не передается. Используется для тестирования в узлах LSU_RX и MPU_RX.	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
10	DSBL_CMPR_ID	Запрещение сравнения содержимого поля <b>destinationID</b> входных пакетов запроса и ответа с содержимым регистров BASE_DEVICE_ID_CSR, MULTICAST_DEVICE_ID: 0 – сравнение разрешено. Входные пакеты принимаются, если содержимое поля <b>destinationID</b> входных пакетов запроса или ответа равно содержимому регистров BASE_DEVICE_ID_CSR или MULTICAST_DEVICE_ID (MULTICAST_DEVICE_ID используется только для пакетов NWRITE и SWRITE) ; 1 – сравнение запрещено. Входные пакеты принимаются вне зависимости от содержимого поля <b>destinationID</b> входных пакетов запроса или ответа. Этот режим может быть использован для приема пакетов Multicast.	RW	0
9	–	Не используется	R	0
8:4	RESP_CANCEL	Число ответных пакетов, которые не были отосланы из-за занятости буфера PL_TXB. Сработал таймер на доступ к PL_TXB, и в течении всего времени PL_TXB был занят. Сбрасывается посредством записи всех 1 в это поле.	RWIC	0
3	–	Не используется	R	0
2	EN_MPU_RX	Разрешение приема входных пакетов запроса типа MESSAGE: 0 – прием запрещен, в случае приёма пакета-запроса выдаётся ответный пакет со статусом «ERROR»; 1 – прием разрешен.	RW	0
1	EN_LSU_RX	Разрешение приема входных пакетов запроса типа NWRITE, NWRITE_R, SWRITE, NREAD, ATOMIC, MAINTENANCE, PORT_WRITE, DOORBELL: 0 – прием запрещен, в случае приёма пакета-запроса выдаётся ответный пакет со статусом «ERROR»; 1 – прием разрешен.	RW	0
0	EN_SRIO	Программная установка SRIO в исходное состояние: 0 – SRIO находится в исходном состоянии, программный сброс SRIO; 1 – SRIO находится в рабочем состоянии.	RW	0

Для начала работы порта SRIO необходимо перевести его в рабочее состояние. Для этого в регистр SRIO\_CSR только в бит EN\_SRIO записывают 1, при этом запись в другие поля регистра SRIO\_CSR произведена не будет, так как в этот момент во всём порте SRIO действует программный сброс.

Когда EN\_SRIO = 1, можно производить запись в SRIO\_CSR и все остальные регистры порта.

Когда EN\_SRIO = 0, весь порт SRIO сбрасывается в исходное состояние. Для уверенного сброса EN\_SRIO должен находиться в 0 несколько тактов.

### 11.3.2.2 Регистры MULTICAST\_DEVICE\_IDn

Имеется 4 регистра MULTICAST\_DEVICE\_ID0:MULTICAST\_DEVICE\_ID3 (Multicast Base Device ID). Используется для приема пакетов типа Multicast (пакеты типа Multicast – NWRITE, SWRITE). Формат этих регистров приведен в Таблица 11.3.

Таблица 11.3 Формат регистров MULTICAST\_DEVICE\_IDn

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:25	–	Не используется	R	0
24	EN	Разрешение сравнения содержимого поля входного пакета <b>destinationID</b> с содержимым этого регистра: 0 – сравнение запрещено, поля регистра BASE_DEV_ID и LARGE_DEV_ID не участвуют в сравнении с полем <b>destinationID</b> входного пакета NWRITE или SWRITE. Если нет ни одного сравнения, то пакет запроса выбрасывается; 1 – сравнение разрешено, и входные пакеты NWRITE или SWRITE с данными идентификаторами принимаются.	RW	0
23:16	BASE_DEV_ID	8-разрядный идентификатор устройства для небольшой транспортной системы	RW	0
15:0	LARGE_DEV_ID	16-разрядный идентификатор устройства для большой транспортной системы	RW	0

### 11.3.2.3 Регистр TEST\_FR

Формат регистра TEST\_FR (Test Fields Register) приведен в Таблица 11.4.

Таблица 11.4 Формат регистра TEST\_FR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	–	Не используется	R	0
30	WDPTR	Для пакетов-запросов содержит код поля <b>wdptr</b> , выдаваемого в тестовом режиме	RW	0
29:24	WORD_COUNT	Количество передаваемых 64-разрядных слов: 00 – 0 слов; 01 - 1 слово; 02 - 2 слова; ... 20 – 32 слова; 21 – 33 слова; ... 3F – 63 слова.	RW	0
23:16	TARGET_TID	Для пакетов message содержит код поля <b>target_info</b> , для пакетов запроса содержит код поля <b>srcTID</b> , для пакетов ответов содержит код поля пакета <b>targetTID (target_info)</b> , выдаваемого в тестовом режиме	RW	0
15:12	SSIZE_STATUS	Для пакетов message содержит код поля <b>size</b> , для пакетов запроса содержит код поля <b>rdsize/wrsize</b> , для пакетов ответов содержит код поля пакета <b>status</b> , выдаваемого в тестовом режиме.	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
11:8	TRANS_MSGLEN	Для пакетов message содержит код поля <b>msglen</b> , для всех остальных пакетов содержит код поля <b>transaction</b> , выдаваемого в тестовом режиме	RW	0
7:4	FTYPE	Для всех типов пакетов содержит код поля пакета <b>ftype</b> , выдаваемого в тестовом режиме	RW	0
3:2	ID_SIZE	Для всех типов пакетов содержит код поля пакета <b>tt</b> , выдаваемого в тестовом режиме	RW	0
1:0	PRIORITY	Для всех типов пакетов содержит код поля пакета <b>prio</b> , выдаваемого в тестовом режиме	RW	0

Регистр TEST\_FR используется для передачи тестовых пакетов запроса и ответа.

При TEST\_REQUEST = 1 из LSU\_TX0 можно передать тестовый пакет любого типа: message, input/output, response. Формат пакета и состав полей определяется заданием из регистров LSU0\_CR0 : LSU\_CR5. При TEST\_REQUEST = 1 в поле FTYPE регистра LSU0\_CR5 можно записывать b'1011 для формирования тестовых пакетов message и b'1101 для ответных пакетов.

Регистры LSU0\_CR0 : LSU\_CR5 должны быть заданы корректно, так как LSU\_TX0 всегда проверяет задание на корректность. Если по заданию требуется ответный пакет, то запускается таймер и ожидается ответный пакет. В регистре LSU\_CR6 всегда выставляется статус выполнения передачи.

При передаче из LSU\_TX0 сформированного на основе задания пакета, происходит следующая подмена полей пакета:

- \* поле **prio** берётся из поля PRIORITY регистра TEST\_FR;
- \* поле **tt** берётся из поля ID\_SIZE регистра TEST\_FR;
- \* поле **ftype** берётся из поля FTYPE регистра TEST\_FR;
- \* поле **destinationID** формируется как обычно из регистра LSU0\_CR4, при этом поле ID\_SIZE из регистра LSU0\_CR4 влияет на размер **destinationID**. Здесь необходимо помнить, что в тестовом режиме поле **tt** берётся из поля ID\_SIZE регистра TEST\_FR, поэтому в пакете размер **destinationID** и значение **tt** могут не соответствовать;
- \* поле **sourceID** берётся из регистра MULTICAST\_DEVICE\_ID0, а поле ID\_SIZE из регистра LSU0\_CR4 определяет размер **sourceID**. Здесь возможно не соответствие размера **sourceID** и значения **tt** как в ситуации с полем **destinationID**;
- \* поле **transaction(msglen** для пакетов message) берётся из поля TRANS\_MSGLEN регистра TEST\_FR;
- \* поля **wrsize/rdsize(ssize** в пакетах message, **status** в ответных пакетах) берутся из поля SSIZE\_STATUS регистра TEST\_FR;
- \* поле **scrTID(target\_info** в пакетах message, **targetTID** в ответных пакетах) берётся из поля TARGET\_TID регистра TEST\_FR;
- \* поля **extended\_address, address, xamsbs** формируются как обычно из регистров LSU0\_CR0, LSU0\_CR1, LSU0\_CR4 соответственно;
- \* поле **hop\_count, config\_offset** в пакетах запросах Maintenance формируются как обычно из регистров LSU0\_CR1, LSU0\_CR5 соответственно;
- \* поле **wdptr** берётся из поля WDPTR регистра TEST\_FR;

- \* число слов в пакете определяется полем WORD\_COUNT регистра TEST\_FR и не зависит от типа пакета(с данными или без). Операция на пакеты не разбивается. Из-за особенностей аппаратуры в пакете может быть 34 или 35 слов, зависит от длины заголовка пакета. Если указать больше слов, то последние 2 слова будут затираться новыми словами;

При TEST\_RESPONSE = 1 узлы LSU\_RX и MPU\_RX в ответ на поступивший пакет запроса выдают ответные пакеты с кодами полей, которые формируются следующим образом:

- \* поле **prio** берётся из поля PRIORITY регистра TEST\_FR;
- \* поле **tt** берётся из поля ID\_SIZE регистра TEST\_FR;
- \* поле **ftype** берётся из поля FTYPE регистра TEST\_FR;
- \* поле **destinationID** формируется как обычно на основе полей **sourceID** и **tt** входного пакета запроса. Так как в тестовом режиме поле **tt** берётся из поля ID\_SIZE регистра TEST\_FR, то в пакете размер **destinationID** и значение **tt** могут не соответствовать;
- \* поле **sourceID** берётся из регистра MULTICAST\_DEVICE\_ID0, а поле **tt** входного пакета определяет размер **sourceID**. Здесь возможно не соответствие размера **sourceID** и значения **tt** как в ситуации с полем **destinationID**;
- \* поле **transaction** берётся из поля TRANS\_MSGLEN регистра TEST\_FR;
- \* поле **status** берётся из поля SSIZE\_STATUS регистра TEST\_FR;
- \* поле **targetTID** (**target\_info** в ответных пакетах message) берётся из поля TARGET\_TID регистра TEST\_FR;
- \* поле **hop\_count** в ответных пакетах Maintenance всегда равно h'FF;
- \* наличие данных и их количество определяется входным пакетом запросом, на который выдаётся ответный пакет. Для ответных пакетов message данных никогда нет.

#### 11.3.2.4 Регистр TIMER\_COUNT

Формат регистра TIMER\_COUNT приведен в Таблица 11.5.

Таблица 11.5 Формат регистра TIMER\_COUNT

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	TIMECODE_CNT	Счетчик таймера ответных пакетов.	RW	0
27:4	DEVIDER_CNT	Основной делитель счётчика таймера. Коэффициент деления – значение поля RTIMEOUT из регистра RESP_TIMEOUT	RW	0
3:0	PRESCALER_CNT	Предварительный делитель счётчика таймера. Коэффициент деления – значение поля PRESCALER из регистра SRIO_CSR	RW	0

Регистр TIMER\_COUNT используется для тестирования таймера ответных пакетов логического уровня и обеспечивает программный доступ к счетчику таймера и его делителям. Процесс счёта управляется битами EN\_TIMER и TICK\_TIMER регистра SRIO\_CSR.

### 11.3.2.5 Регистр PACKET\_FR

Формат регистра PACKET\_FR приведён в Таблица 11.6.

Таблица 11.6 Формат регистра PACKET\_FR (Packet Fields Register)

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31	–	Не используется	R	0
30:28	REM	Количество значащих байт в последнем слове принятого пакета при передаче его из буфера PL_RXB в SRIO_CTR	RW	0
27	–	Не используется	R	0
26:25	XAMBS	Значение поля <b>xambs</b> последнего принятого пакета. Если в пакете нет поля <b>xambs</b> , то записывается нулевое значение	RW	0
24	WDPTR	Значение поля <b>wdptr</b> последнего принятого пакета. Если в пакете нет поля <b>wdptr</b> , то записывается нулевое значение.	RW	0
23:16	HOP_COUNT	Значение поля <b>hop_count</b> последнего принятого пакета ( <b>hop_count</b> присутствует в пакетах MAINTENANCE). Если в пакете нет поля <b>hop_count</b> , то записывается нулевое значение.	RW	0
15:8	SRCTID	Значение поля <b>srcTID</b> , <b>targetTID</b> или <b>target_info</b> в зависимости от типа принятого пакета. Если в пакете нет таких полей (например пакет Swrite), то записывается нулевое значение.	RW	0
7:4	FTYPE	Значение поля <b>ftype</b> последнего принятого пакета.	RW	0
3:2	TT	Значение поля <b>tt</b> последнего принятого пакета.	RW	0
1:0	PRIО	Значение поля <b>prio</b> последнего принятого пакета.	RW	0

Регистр PACKET\_FR содержит значения некоторых полей последнего принятого пакета. Регистр PACKET\_FR в основном используется при тестировании, для проверки правильности передачи полей. Поле REM позволяет проверить правильность вычисления значащих байт в последнем слове пакета. Эти вычисления происходят в LPU и через буфер PL\_RXB передаются в SRIO\_CTR, где пакет обрабатывается.

В поля REM и PRIО, TT, FTYPE всегда записываются значения при приёме очередного пакета из PL\_RXB.

В поля SRCTID, HOP\_COUNT, WDPTR, XAMBS могут прописаться 0000 в нескольких случаях: пакет неопределённого типа, пакет по типу не содержит таких полей. В других случаях записываются принятые значения этих полей.

При каждом приёме пакета предыдущие значения затираются новыми значениями. Если новый пакет неопределённого типа или там нет таких полей, то записываются 000. Также все поля регистра в любой момент доступны для программной записи.



### 11.3.2.6 Регистр EXTND\_ADDR

Формат регистра EXTND\_ADDR приведён в Таблица 11.7.

Таблица 11.7 Формат регистра EXTND\_ADDR

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:0	EXTENDED_ADDR	Значение поля <b>extended_address</b> входящих пакетов. Если в пакете нет поля <b>extended_address</b> , то записывается нулевое значение	RW	0

Регистр EXTND\_ADDR в основном используется при тестировании, для проверки правильности передачи поля **extended\_address**.

При приёме пакета в поле EXTENDED\_ADDR могут прописаться 0000 в нескольких случаях: пакет неопределённого типа, пакет по типу не содержит поля **extended\_address**. В других случаях записывается принятое значение этого поля.

При каждом приёме пакета предыдущее значение затирается новым значением. Если новый пакет неопределённого типа или там нет поля **extended\_address**, то записываются 0000. Также регистр в любой момент доступен для программной записи.

### 11.3.3 Регистры устройства выполнения операций ввода-вывода (LSU)

Устройство LSU осуществляет передачу следующих пакетов:

- 2-ой тип – NREAD, ATOMIC POST-INCREMENT, ATOMIC POST-DECREMENT, ATOMIC SET, ATOMIC CLEAR;
- 5-ий тип – NWRITE, NWRITE\_R, ATOMIC TEST-AND-SWAP;
- 6-ой тип – SWRITE;
- 8-ой тип – MAINTENANCE READ REQUEST, MAINTENANCE WRITE REQUEST, MAINTENANCE PORT-WRITE REQUEST;
- 10-ый тип – DOORBELL.

Устройство LSU состоит из 4-ёх узлов: LSU\_TX0, LSU\_TX1, LSU\_TX2, LSU\_TX3. Каждому узлу соответствует свой набор управляющих регистров LSU<sub>n</sub>\_CR0 : LSU<sub>n</sub>\_CR6.

#### 11.3.3.1 Регистр LSU<sub>n</sub>\_CR0

Формат регистра LSU<sub>n</sub>\_CR0 приведен в Таблица 11.8.

Таблица 11.8. Формат регистра LSU<sub>n</sub>\_CR0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	EXTENDED_ADDR	Поле <b>extended address</b> для пакетов типа 2,5 и 6. Это поле может присутствовать либо отсутствовать в этих пакетах. Наличие в пакете поля <b>extended address</b> зависит от состояния поля EXT_ADDR_CTR регистра PE_LOG_CSR	RW	0



### 11.3.3.2 Регистр L<sub>SUn</sub>\_CR1

Формат регистра L<sub>SUn</sub>\_CR1 приведен в Таблица 11.9.

Таблица 11.9. Формат регистра L<sub>SUn</sub>\_CR1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	ADRR/CONFIG	<p>Используется для формирования поля <b>address</b> пакетов типа NREAD, NWRITE, SWRITE. При этом младшие 3-и разряда этого регистра должны быть нулевыми, так как передаются 64-разрядные данные.</p> <p>Для пакетов типа ATOMIC также используется для формирования поля <b>address</b>, но при этом младшие 2-а разряда должны быть нулевыми. 3-ий разряд определяет адрес 32-разрядного слова участвующего в операции ATOMIC и влияет на состояние поля <b>wdptr</b> пакета, которое формируется автоматически.</p> <p>Для пакетов MAINTENANCE READ/WRITE REQUEST используется для формирования поля <b>config_offset</b>, но при этом младшие 2-а разряда должны быть нулевыми. 3-ий разряд определяет адрес 32-разрядного регистра участвующего в операции MAINTENANCE READ/WRITE и влияет на состояние поля <b>wdptr</b> пакета.</p> <p>Для пакетов MAINTENANCE PORT-WRITE REQUEST используется для формирования поля <b>config_offset</b>. При этом младшие 3-и разряда этого регистра должны быть нулевыми, так как передаются 64-разрядные данные.</p> <p>Для пакетов DOORBELL этот регистр не имеет значения.</p>	RW	0

### 11.3.3.3 Регистр L<sub>SUn</sub>\_CR2

Формат регистра L<sub>SUn</sub>\_CR2 приведен в Таблица 11.10.

Таблица 11.10. Формат регистра L<sub>SUn</sub>\_CR2

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	ADDR	<p>Адрес памяти данного микропроцессора, выровненный по границе 64-разрядного слова (младшие 3 разряда этого поля нулевые).</p> <p>Это начальный адрес для чтения передаваемых данных в пакетах типа NWRITE, ATOMIC TEST-AND-SWAP и т.д.</p> <p>Для пакетов типа NREAD, ATOMIC, MAINTENANCE READ REQUEST это начальный адрес для записи принимаемых данных из ответного пакета.</p> <p>Для пакетов, где 3-ий разряд регистра L<sub>SUn</sub>_CR1 определяет положение 32-разрядного результирующего слова, это слово должно находиться по адресу ADDR в старшей или младшей его части согласно 3-ему разряду регистра L<sub>SUn</sub>_CR1. Это правило относится к чтению и записи данных.</p> <p>Для пакетов DOORBELL этот регистр не имеет значения.</p>	RW	0

### 11.3.3.4 Регистр LSUn\_CR3

Формат регистра LSUn\_CR3 приведен в Таблица 11.11.

Таблица 11.11. Формат регистра LSUn\_CR3

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	–	Не используется	R	0
15:0	WORD_COUNT	<p>Количество передаваемых 64-разрядных слов. Используется для формирования полей пакета <b>wrsizе/rdsizе</b> и <b>wdptr</b>:</p> <p>0000 – 65536 слов;                      0001 - 1 слово;                      0002 - 2 слова;                      ...                      ffff – 65535 слов.</p> <p>Поле WORD_COUNT участвует в проверке корректности задания.</p> <p>Для пакетов типа NREAD, NWRITE, SWRITE количество слов может быть любым. Если оно больше 32 слов, то операция является многопакетной.</p> <p>Для пакетов типа ATOMIC, MAINTENANCE READ/WRITE REQUEST количество слов должно быть 1.</p> <p>Для пакетов MAINTENANCE PORT-WRITE количество слов может быть 1, 2, 4, 8.</p> <p>Для пакетов DOORBELL этот регистр не имеет значения.</p>	RW	0

### 11.3.3.5 Регистр L<sub>SUn</sub>\_CR4

Формат регистра L<sub>SUn</sub>\_CR4 приведен в Таблица 11.12.

Таблица 11.12. Формат регистра L<sub>SUn</sub>\_CR4

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	–	Не используется	R	0
29:28	PRIORITY	Формирует поле пакета <b>priority</b> : 00 – самый низкий; 11 – самый высокий. Пакеты запроса не должны иметь приоритет «11» для исключения тупиковых ситуаций в системе. Поле PRIORITY участвует в проверке корректности задания.	RW	0
27:26	XAMBS	Формирует поле пакета <b>xambs</b> (старшие разряды расширенного адреса).	RW	0
25:24	ID_SIZE	Формирует поле пакета <b>tt</b> , которое определяет длину полей <b>sourceID</b> и <b>destinationID</b> пакета: 00 – 8 разрядов; 01 – 16 разрядов; 10, 11 – резерв. Поле ID_SIZE участвует в проверке корректности задания.	RW	0
23:8	DEST_ID	Формирует поле пакета <b>destinationID</b> . В зависимости от ID_SIZE из поля DEST_ID берутся все 16 разрядов или только младшие 8 разрядов.	RW	0
7:1	–	Не используется	R	0
0	INT_MASK	Маска прерывания после завершения операции ввода-вывода: 0 – прерывание запрещено; 1 – прерывание разрешено.	RW	0

### 11.3.3.6 Регистр L<sub>SUn</sub>\_CR5

Формат регистра L<sub>SUn</sub>\_CR5 приведен в Таблица 11.13.

Таблица 11.13. Формат регистра L<sub>SUn</sub>\_CR5

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	DRBLL_INFO	Формирует поле <b>info</b> пакетов DOORBELL(10-ый тип). Для пакетов других типов это поле значения не имеет.	RW	0
15:8	HOP_COUNT	Формирует поле <b>hop_count</b> всех пакетов MAINTENANCE(8-ой тип). Для пакетов других типов это поле значения не имеет.	RW	0
7:4	FTYPE	Формирует поле <b>ftype</b> пакетов всех типов. Поле FTYPE участвует в проверке корректности задания.	RW	0
3:0	TRANSACTION	Формирует поле пакетов <b>transaction</b> . Для пакетов WRITE и DOORBELL не имеет значения, так как в этих пакетах нет поля <b>transaction</b> . Поле TRANSACTION участвует в проверке корректности задания.	RW	0

Запись в регистр L<sub>SUn</sub>\_CR5 запускает задание на выполнение.

### 11.3.3.7 Регистр LSUn\_CR6

Формат регистра LSUn\_CR6 приведен в Таблица 11.14.

Таблица 11.14. Формат регистра LSUn\_CR6

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:5	–	Не используется	R	0
4:1	OP_STATUS	<p>Состояние выполнения задания, записывается по завершении:</p> <p>0000 – операция выполнена без ошибок (Posted/Non-Posted). (Тестовый режим в LSU_TX0: получен ответный пакет со статусом «DONE», проверка длины и сохранение данных не производится);</p> <p>0001 – при выполнении Non-Posted операции (операции, в которых требуется ответный пакет) произошел таймаут (используется в тестовом режиме в LSU_TX0 для тех заданий, где требуется ответный пакет);</p> <p>0010 – в ответ на пакет запроса DOORBELL принят ответный пакет, содержащий в поле <b>status</b> код «RETRY». (Тестовый режим в LSU_TX0: ответный пакет DOORBELL или MESSAGE со статусом «RETRY»);</p> <p>0011 – при выполнении операции получен ответный пакет (типа 8 или 13), содержащий в поле <b>status</b> код «ERROR» или ответные данные имеют неправильную длину. (Тестовый режим в LSU_TX0: ответный пакет со статусом «ERROR», длина ответных данных не проверяется);</p> <p>0100 – операция не может быть выполнена, так как она не реализована, или регистры LSU неправильно запрограммированы (проверка корректности задания по регистрам LSUn_CR0 : LSUn_CR5 проводится также в тестовом режиме в LSU_TX0);</p> <p>0101 – в ответ на пакет запроса DOORBELL принят ответный пакет, содержащий в поле <b>status</b> код «ERROR» (Тестовый режим в LSU_TX0: ответный пакет DOORBELL или MESSAGE со статусом «ERROR»);</p> <p>0110 - операция «ATOMIC Test-and-Swap» не может быть выполнена из-за занятости семафора. (В тестовом режиме в LSU_TX0 этот код не используется);</p> <p>0111 – пакет не может быть передан из-за занятости буфера PL_TXB в момент программного запуска операции или в случае выполнения многопакетной операции (используется в тестовом режиме в LSU_TX0, так как это событие может произойти).</p> <p>0111 - устанавливается только при срабатывании таймера доступа к буферу PL_TXB. Этот таймер повторяет таймер таймаута ответных пакетов и работает по полю TIMECODE регистра TIMER_COUNT.</p>	R	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
0	BUSY	Признак занятости регистров LSUn_CR0 : LSUn_CR5, признак выполнения задания. Устанавливается в «1» в момент записи данных в регистр LSUn_CR5 и соответственно в начале выполнения задания. Сбрасывается в «0» при окончании выполнения задания (с ошибкой или без нее). В этом момент в поле OP_STATUS выставляется состояние выполнения.	R	0

*Примечание:* Операция является Posted, если она не требует ответного пакета. Операция является Non-Posted, если она требует ответного пакета.

### 11.3.3.8 Регистры IN\_FLTR0, IN\_FLTR1

Регистры IN\_FLTR0, IN\_FLTR1(Inbound Filter Register) предназначены для фильтрации входящих пакетов ввода-вывода данных по адресу, который находится в поле пакета **address**. На пакеты DOORBELL и MAINTENANCE, где нет поля **address**, эти регистры влияния не оказывают.

Формат регистров IN\_FLTR0, IN\_FLTR1 приведен в Таблица 11.15.

Таблица 11.15. Формат регистров IN\_FLTR0, IN\_FLTR1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:24	MASK1	Разряды маски 31:24 при определении базового адреса блока. Младшие разряды маски равны нулю.	RW	0
23:16	BA1	Разряды 31:24 базового адреса блока. Младшие разряды базового адреса равны нулю.	RW	0
15:8	MASK0	Разряды маски 31:24 при определении базового адреса блока. Младшие разряды маски равны нулю.	RW	0
7:0	BA0	Разряды 31:24 базового адреса блока. Младшие разряды базового адреса равны нулю.	RW	0

Эти регистры могут определять 4 области памяти, в которые ввод/вывод данных запрещен. Размер области от 16 Мбайт до 2 Гбайт.

Условие попадания адреса входного пакета (содержимое поля **address**) в запрещенную область, выполняется, если  $\text{address}[28:21] \& \text{MASK} = \text{BA}$ . Минимальный размер области, равен 16 Мбайт (при MASK = FF). Для увеличения размера области в младшие разряды поля MASK необходимо записать соответствующее число нулей. Например, для блока размером в 128 Мбайт, разряды 2:0 MASK должны быть равны нулю. Активность области определяется единичным состоянием старшего разряда поля MASK. То есть, размер области не может быть больше 2 Гбайт.

Если по данному адресу доступ запрещен, то входной пакет далее не обрабатывается, и имеющиеся в нем данные выбрасываются. Если по этой операции требуется ответный пакет, то он выдается с признаком ERROR в поле **status** и без данных.

### 11.3.3.9 Регистр *PORT\_WRITE\_SR*

LSU может принять только один пакет MAINTENANCE PORT-WRITE с данными до 8-ми 64-разрядных слов (16-ть 32-разрядных слов), которые помещаются в очередь. В регистре *PORT\_WRITE\_SR* (Port Write Status Register) отображается статусная информация этой очереди. Формат этого регистра приведен в Таблица 11.16

Таблица 11.16. Формат регистра *PORT\_WRITE\_SR*

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	SOURCE_ID_PW	Содержимое поля <b>sourceID</b> входного пакета MAINTENANCE PORT-WRITE.	R	0
15	NEMPTY_PW	Признак наличия данных в очереди (регистр <i>PORT_WRITE_FIFO</i> ). NEMPTY_PW = 0 – очередь пустая, NEMPTY_PW = 1 – очередь не пустая.	R	0
14:13	TT_PW	Содержимое поля <b>tt</b> входного пакета MAINTENANCE PORT-WRITE. Определяет размер <b>sourceID</b> .	R	0
12:8	SIZE_PW	Количество 32-разрядных слов данных пакета в очереди. При чтении из очереди поле SIZE_PW указывает реальное количество оставшихся в очереди слов.	R	0
7:0	SRC_TID_PW	Содержимое поля <b>srcTID</b> входного пакета MAINTENANCE PORT-WRITE.	R	0

### 11.3.3.10 Регистр *PORT\_WRITE\_FIFO*

Регистр *PORT\_WRITE\_FIFO*[31:0] обеспечивает доступ к данным принятого пакета MAINTENANCE PORT-WRITE, которые находят в специальной очереди. Исходное состояние регистра *PORT\_WRITE\_FIFO*: XXXX.

При чтении из регистра, в него загружаются следующие данные из очереди. Оставшееся количество данных в очереди отображается в поле *SIZE\_PW* регистра *PORT\_WRITE\_SR*. При чтении последнего слова данных из регистра *PORT\_WRITE\_FIFO* происходят следующие установки в *PORT\_WRITE\_SR*: *NEMPTY\_PW* = 0, *SIZE\_PW* = 0. Остальные поля регистра *PORT\_WRITE\_SR* остаются неизменными до прихода следующего пакета MAINTENANCE PORT-WRITE. При чтении последнего слова данных из регистра *PORT\_WRITE\_FIFO*, он переходит в неизвестное значение.

### 11.3.3.11 Регистр *DOORBELL\_FIFO\_LOW*

LSU может принять 16-ть пакетов DOORBELL, которые помещаются в специальную очередь. Данные из этой очереди отображаются в двух регистрах *DOORBELL\_FIFO\_LOW* и *DOORBELL\_FIFO\_HIGH*. Формат регистра *DOORBELL\_FIFO\_LOW* приведен в Таблица 11.17.

Таблица 11.17 Формат регистра DOORBELL\_FIFO\_LOW

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31	NEMPTY_DRBL	Признак наличия пакетов DOORBELL в очереди.	R	0
30:12	–	Не используется	R	0
11:10	TT_DRBL	Содержимое поля <b>tt</b> очередного пакета из очереди.	R	X
9:8	PRIO_DRBL	Содержимое поля <b>prio</b> очередного пакета из очереди.	R	X
7:0	SRC_TID_DRBL	Содержимое поля <b>srcTID</b> очередного пакета из очереди.	R	X

### 11.3.3.12 Регистр DOORBELL\_FIFO\_HIGH

Формат регистра DOORBELL\_FIFO\_HIGH приведен в Таблица 11.18.

Таблица 11.18 Формат регистра DOORBELL\_FIFO\_HIGH

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	SOURCE_ID_DRBL	Содержимое поля <b>sourceID</b> очередного пакета из очереди..	R	X
15:0	INFO_DRBL	Содержимое поля <b>info</b> очередного пакета из очереди..	R	X

При чтении из регистра DOORBELL\_FIFO\_HIGH, в регистры DOORBELL\_FIFO\_LOW и DOORBELL\_FIFO\_HIGH загружаются данные следующего пакета DOORBELL из очереди. При чтении последних данных устанавливается NEMPTY\_DRBL = 0, остальные поля регистров переходят в неизвестные значения.



### 11.3.3.13 Регистр LSU\_IRQ\_SR

Формат регистра LSU\_IRQ\_SR (LSU Interrupt Request Status Register) приведен в Таблица 11.19. В регистре LSU\_IRQ\_SR представлены запросы на прерывания от всех 4-ёх узлов LSU: LSU\_TX0 : LSU\_TX3. На принадлежность поля к конкретному узлу указывает цифра в названии.

Таблица 11.19. Формат регистра LSU\_IRQ\_SR

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31	OKEY3	Операция выполнена без ошибок (Posted/Non-Posted).	RW	0
30	ERROR3	При выполнении операции получен ответный пакет (типа 8 или 13), содержащий в поле <b>status</b> код «ERROR» или его поле данных имеет неправильную длину.	RW	0
29	DRBL_RETRY3	В ответ на пакет DOORBELL принят пакет, содержащий в поле <b>status</b> код «RETRY».	RW	0
28	UNSUPPORT-ED3	Операция не может быть выполнена, так как она не реализована, или регистры LSU неправильно запрограммированы.	RW	0
27	TIMEOUT3	При выполнении Non-Posted операции (операции, в которых требуется ответный пакет) произошел таймаут.	RW	0
26	DRBL_ERROR3	В ответ на пакет DOORBELL принят пакет, содержащий в поле <b>status</b> код «ERROR».	RW	0
25	NOT_ALLOWED3	Операция «ATOMIC Test-and-Swap» не может быть выполнена из-за занятости семафора.	RW	0
24	NO_CREDIT3	Пакет не может быть передан из-за занятости буфера PL_TXB в момент программного запуска операции или при срабатывании таймера в случае выполнения многопакетной операции.	RW	0
23	OKEY2	Операция выполнена без ошибок (Posted/Non-Posted).	RW	0
22	ERROR2	При выполнении операции получен ответный пакет (типа 8 или 13), содержащий в поле <b>status</b> код «ERROR» или его поле данных имеет неправильную длину.	RW	0
21	DRBL_RETRY2	В ответ на пакет DOORBELL принят пакет, содержащий в поле <b>status</b> код «RETRY».	RW	0
20	UNSUPPORTED2	Операция не может быть выполнена, так как она не реализована, или регистры LSU неправильно запрограммированы.	RW	0
19	TIMEOUT2	При выполнении Non-Posted операции (операции, в которых требуется ответный пакет) произошел таймаут.	RW	0
18	DRBL_ERROR2	В ответ на пакет DOORBELL принят пакет, содержащий в поле <b>status</b> код «ERROR».	RW	0
17	NOT_ALLOWED2	Операция «ATOMIC Test-and-Swap» не может быть выполнена из-за занятости семафора.	RW	0

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
16	NO_CREDIT2	Пакет не может быть передан из-за занятости буфера PL_TXB в момент программного запуска операции или при срабатывании таймера в случае выполнения многопакетной операции.	RW	0
15	OKEY1	Операция выполнена без ошибок (Posted/Non-Posted).	RW	0
14	ERROR1	При выполнении операции получен ответный пакет (типа 8 или 13), содержащий в поле <b>status</b> код «ERROR» или его поле данных имеет неправильную длину.	RW	0
13	DRBL_RETRY1	В ответ на пакет DOORBELL принят пакет, содержащий в поле <b>status</b> код «RETRY».	RW	0
12	UNSUPPORTED1	Операция не может быть выполнена, так как она не реализована, или регистры LSU неправильно запрограммированы.	RW	0
11	TIMEOUT1	При выполнении Non-Posted операции (операции, в которых требуется ответный пакет) произошел таймаут.	RW	0
10	DRBL_ERROR1	В ответ на пакет DOORBELL принят пакет, содержащий в поле <b>status</b> код «ERROR».	RW	0
9	NOT_ALLOWED1	Операция «ATOMIC Test-and-Swap» не может быть выполнена из-за занятости семафора.	RW	0
8	NO_CREDIT1	Пакет не может быть передан из-за занятости буфера PL_TXB в момент программного запуска операции или при срабатывании таймера в случае выполнения многопакетной операции.	RW	0
7	OKEY0	Операция выполнена без ошибок (Posted/Non-Posted).	RW	0
6	ERROR0	При выполнении операции получен ответный пакет (типа 8 или 13), содержащий в поле <b>status</b> код «ERROR» или его поле данных имеет неправильную длину.	RW	0
5	DRBL_RETRY0	В ответ на пакет DOORBELL принят пакет, содержащий в поле <b>status</b> код «RETRY».	RW	0
4	UNSUPPORTED0	Операция не может быть выполнена, так как она не реализована, или регистры LSU неправильно запрограммированы.	RW	0
3	TIMEOUT0	При выполнении Non-Posted операции (операции, в которых требуется ответный пакет) произошел таймаут.	RW	0
2	DRBL_ERROR0	В ответ на пакет DOORBELL принят пакет, содержащий в поле <b>status</b> код «ERROR».	RW	0
1	NOT_ALLOWED0	Операция «ATOMIC Test-and-Swap» не может быть выполнена из-за занятости семафора.	RW	0
0	NO_CREDIT0	Пакет не может быть передан из-за занятости буфера PL_TXB в момент программного запуска операции или при срабатывании таймера в случае выполнения многопакетной операции.	RW	0

### 11.3.3.14 Регистр *LSU\_IRQ\_CLR*

Для обнуления разрядов регистра *LSU\_IRQ\_SR* имеется 32-разрядный регистр *LSU\_IRQ\_CLR* (LSU Interrupt Request Clear Register). Его формат полностью повторяет формат регистра *LSU\_IRQ\_SR*. Он доступен только по записи 1. При записи 1 в *LSU\_IRQ\_CLR* соответствующий разряд регистра *LSU\_IRQ\_SR* обнуляется. Из *LSU\_IRQ\_CLR* всегда считываются нули.

### 11.3.3.15 Регистр *MNT\_WR\_IRQ*

Формат регистра *MNT\_WR\_IRQ* (Maintenance Write Interrupt Request) приведен в Таблица 11.20

Таблица 11.20. Формат регистра *MNT\_WR\_IRQ*

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:16	SOURCE_ID	Содержимое поля <b>sourceID</b> входного пакета MAINTENANCE WRITE REQUEST	R	00
15	WNT_WR_INT	Устанавливается при обработке входного пакета MAINTENANCE WRITE REQUEST, который осуществляет запись нового значения в Maintenance регистр. Сбрасывается посредством записи 1 в этот бит	RWIC	0
14	–	Не используется	R	0
13:12	TT	Содержимое поля <b>tt</b> входного пакета MAINTENANCE WRITE REQUEST. Используется для определения размера SOURCE_ID	R	00
11:0	ADDR	Младшие 12 разрядов адреса Maintenance регистра, в который была произведена запись при обработке пакета MAINTENANCE WRITE REQUEST. Данный порт SRIO не содержит maintenance регистров, у которых старшие 12 разрядов адреса отличны от 0. Поэтому они равны 0.	R	0

### 11.3.4 Регистры устройства MPU

#### 11.3.4.1 Регистры RXU\_MAP\_Ln

Имеется 32 регистра RXU\_MAP\_Ln (Mailbox to Queue Mapping Register Low). Формат этих регистров приведен в Таблица 11.21. Регистры RXU\_MAP – это регистры отображения входных пакетов MESSAGE в одну из 16-ти очередей.

Таблица 11.21. Формат регистра RXU\_MAP\_Ln

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	LETTER_MASK	Маска номера письма: 0 – соответствующий бит не участвует в сравнении; 1 – соответствующий бит участвует в сравнении.	RW	0
29:24	MAILBOX_MASK	Маска номера почтового ящика: 0 – соответствующий бит не участвует в сравнении; 1 – соответствующий бит участвует в сравнении.	RW	0
23:22	LETTER	Номер письма для сравнения с полем <b>letter</b> входного пакета MESSAGE с учётом маски LETTER_MASK.	RW	0
21:16	MAILBOX	Номер почтового ящика для сравнения с полем { <b>xmailbox,mailbox</b> } входного пакета с учётом маски MAILBOX_MASK. Поле <b>xmailbox</b> (расширение поля <b>mailbox</b> ) может присутствовать только в однопакетных сообщения на месте поля <b>msgseg</b> .	RW	0
15:0	SOURCEID	Идентификатор источника входного пакета для сравнения с полем <b>sourceID</b> входного пакета с учётом бита PROMISCUOUS соответствующего регистра RXU_MAP_Ln.	RW	0

### 11.3.4.2 Регистры RXU\_MAP\_Hn

Имеется 32 регистра RXU\_MAP\_Hn. (Mailbox to Queue Mapping Register High). Формат этих регистров приведен в Таблица 11.22.

Таблица 11.22. Формат регистра RXU\_MAP\_Hn

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:10	–	Не используется	R	0
9:8	TT	Длина поля <b>sourceID</b> : 0 – 8 разрядов; 1 – 16 разрядов. Сравнивается с полем <b>tt</b> входного пакета MESSAGE.	RW	0
7:6	–	Не используется	R	0
5:2	QUEUE_ID	Номер очереди – от 0 до 15. Пакет будет приниматься в заданную очередь, если произошло сравнение пакета по указанным параметрам. Если сравнения не произошло, то входной пакет выкидывается и выдаётся ответный пакет со статусом ERROR	RW	0
1	PROMISCUOUS	Разрешение не сравнивать содержимое поля пакета <b>sourceID</b> и поля SOURCEID регистра RXU_MAP_Ln: 0 – сравнение выполняется; 1 – сравнение не выполняется.	RW	0
0	SEGMENT_MAPPING	Режим сегментации: 0 – в очередь принимаются только однопакетные сообщения; 1 – в очередь принимаются только многопакетные сообщения.	RW	0

### 11.3.4.3 Регистры RXQ\_HDPn

Имеется 16 регистров RXQ\_HDPn (Receive Queue Head Descriptor Pointer). Формат этих регистров приведен в Таблица 11.23.

Таблица 11.23. Формат регистра RXQ\_HDPn

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	RX_HDP	Указатель(адрес) на первый дескриптор очереди. Адрес дескриптора должен отличаться от нуля, а младшие 3 разряда должны быть нулевыми. Для инициализации данной очереди необходимо записать в регистр адрес первого дескриптора очереди. После исчерпания очереди, то есть после обработки последнего дескриптора, регистр аппаратно обнуляется. Пока регистр не обнулится, запись в него аппаратно запрещена.	RW	0

#### 11.3.4.4 Регистры RXQ\_CDPn

Имеется 16 регистров RXQ\_CDPn (Receive Queue Completion Descriptor Pointer). Формат этих регистров приведен в Таблица 11.24.

Таблица 11.24. Формат регистра RXQ\_CDPn

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	RX_CDP	<p>Указатель(адрес) на последний обработанный дескриптор. Младшие 3 разряда должны быть нулевыми.</p> <p>После приема очередного сообщения в этот регистр аппаратно записывается адрес дескриптора данного сообщения и формируется прерывание, если оно заказано в дескрипторе.</p> <p>После обработки прерывания по дескриптору в регистр RXQ_CDPn необходимо программно записать адрес этого дескриптора, хотя регистр доступен только для чтения. Аппаратной записи на самом деле нет, а если записываемый адрес совпал с содержимым регистра, то соответствующее прерывание сбрасывается. После этого, буфер, определяемый дескриптором, может быть вновь использован SRIO.</p> <p>Следует помнить, что во время обработки прерывания в регистре RXQ_CDPn может измениться значение адреса из-за приёма сообщения по следующему дескриптору.</p>	R	0

#### 11.3.4.5 Регистры TXQ\_HDPn

Имеется 16 регистров TXQ\_HDPn (Transmit Queue Head Descriptor Pointer). Формат этих регистров приведен в Таблица 11.25.

Таблица 11.25. Формат регистра TXQ\_HDPn

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	TX_HDP	<p>Указатель(адрес) на первый дескриптор очереди. Адрес дескриптора должен отличаться от нуля, а младшие 3 разряда должны быть нулевыми.</p> <p>Для инициализации (запуска процесса передачи сообщений) данной очереди необходимо записать в регистр адрес первого дескриптора очереди. После исчерпания очереди, то есть после обработки последнего дескриптора, регистр аппаратно обнуляется.</p> <p>Пока регистр не обнулится, запись в него аппаратно запрещена.</p>	RW	0

#### 11.3.4.6 Регистры TXQ\_CDPn

Имеется 16 регистров TXQ\_CDPn (Transmit Queue Completion Descriptor Pointer). Формат этих регистров приведен в Таблица 11.26.

Таблица 11.26. Формат регистра TXQ\_CDPn

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	TX_CDP	<p>Указатель(адрес) на последний обработанный дескриптор. Младшие 3 разряда должны быть нулевыми.</p> <p>После передачи очередного сообщения в этот регистр аппаратно записывается адрес дескриптора данного сообщения и формируется прерывание, если оно заказано в дескрипторе.</p> <p>После обработки прерывания по дескриптору в регистр TXQ_CDPn необходимо программно записать адрес этого дескриптора, хотя регистр доступен только для чтения. Аппаратной записи на самом деле нет, а если записываемый адрес совпал с содержимым этого регистра, то соответствующее прерывание сбрасывается. После этого, буфер, определяемый этим дескриптором, может быть вновь использован SRIO.</p> <p>Следует помнить, что во время обработки прерывания в регистре TXQ_CDPn может измениться значение адреса из-за передачи сообщения по следующему дескриптору.</p>	R	0

#### 11.3.4.7 Регистры TX\_QUEUE\_CTR

Имеется 4 регистра управления переходами между очередями передачи TX\_QUEUE\_CTR0, TX\_QUEUE\_CTR1, TX\_QUEUE\_CTR2, TX\_QUEUE\_CTR3. Формат этих регистров приведен в Таблица 11.27 - Таблица 11.30.

Таблица 11.27. Формат регистра TX\_QUEUE\_CTR0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	MSG_NMBR0	Количество сообщений (дескрипторов) очереди 0, которые передаются перед переходом к другой очереди.	RW	0
27:24	POINTER0	Указатель номера очереди, на которую выполняется переход от очереди 0.	RW	0
23:20	MSG_NMBR1	Количество сообщений (дескрипторов) очереди 1, которые передаются перед переходом к другой очереди.	RW	0
19:16	POINTER1	Указатель номера очереди, на которую выполняется переход от очереди 1.	RW	0
15:12	MSG_NMBR2	Количество сообщений (дескрипторов) очереди 2, которые передаются перед переходом к другой очереди.	RW	0
11:8	POINTER2	Указатель номера очереди, на которую выполняется переход от очереди 2.	RW	0
7:4	MSG_NMBR3	Количество сообщений (дескрипторов) очереди 3, которые передаются перед переходом к другой очереди.	RW	0
3:0	POINTER3	Указатель номера очереди, на которую выполняется переход от очереди 3.	RW	0

Таблица 11.28. Формат регистра TX\_QUEUE\_CTR1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	MSG_NMBR4	Количество сообщений (дескрипторов) очереди 4, которые передаются перед переходом к другой очереди.	RW	0
27:24	POINTER4	Указатель номера очереди, на которую выполняется переход от очереди 4.	RW	0
23:20	MSG_NMBR5	Количество сообщений (дескрипторов) очереди 5, которые передаются перед переходом к другой очереди.	RW	0
19:16	POINTER5	Указатель номера очереди, на которую выполняется переход от очереди 5.	RW	0
15:12	MSG_NMBR6	Количество сообщений (дескрипторов) очереди 6, которые передаются перед переходом к другой очереди.	RW	0
11:8	POINTER6	Указатель номера очереди, на которую выполняется переход от очереди 6.	RW	0
7:4	MSG_NMBR7	Количество сообщений (дескрипторов) очереди 7, которые передаются перед переходом к другой очереди.	RW	0
3:0	POINTER7	Указатель номера очереди, на которую выполняется переход от очереди 7.	RW	0

Таблица 11.29. Формат регистра TX\_QUEUE\_CTR2

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	MSG_NMBR8	Количество сообщений (дескрипторов) очереди 8, которые передаются перед переходом к другой очереди.	RW	0
27:24	POINTER8	Указатель номера очереди, на которую выполняется переход от очереди 8.	RW	0
23:20	MSG_NMBR9	Количество сообщений (дескрипторов) очереди 9, которые передаются перед переходом к другой очереди.	RW	0
19:16	POINTER9	Указатель номера очереди, на которую выполняется переход от очереди 9.	RW	0
15:12	MSG_NMBR10	Количество сообщений (дескрипторов) очереди 10, которые передаются перед переходом к другой очереди.	RW	0
11:8	POINTER10	Указатель номера очереди, на которую выполняется переход от очереди 10.	RW	0
7:4	MSG_NMBR11	Количество сообщений (дескрипторов) очереди 11, которые передаются перед переходом к другой очереди.	RW	0
3:0	POINTER11	Указатель номера очереди, на которую выполняется переход от очереди 11.	RW	0



Таблица 11.30. Формат регистра TX\_QUEUE\_CTR3

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	MSG_NMBR12	Количество сообщений (дескрипторов) очереди 12, которые передаются перед переходом к другой очереди.	RW	0
27:24	POINTER12	Указатель номера очереди, на которую выполняется переход от очереди 12.	RW	0
23:20	MSG_NMBR13	Количество сообщений (дескрипторов) очереди 13, которые передаются перед переходом к другой очереди.	RW	0
19:16	POINTER13	Указатель номера очереди, на которую выполняется переход от очереди 13.	RW	0
15:12	MSG_NMBR14	Количество сообщений (дескрипторов) очереди 14, которые передаются перед переходом к другой очереди.	RW	0
11:8	POINTER14	Указатель номера очереди, на которую выполняется переход от очереди 14.	RW	0
7:4	MSG_NMBR15	Количество сообщений (дескрипторов) очереди 15, которые передаются перед переходом к другой очереди.	RW	0
3:0	POINTER15	Указатель номера очереди, на которую выполняется переход от очереди 15.	RW	0

Если MSG\_NMBR = 0, то передаётся одно сообщение (один дескриптор) перед переходом к другой очереди. При MSG\_NMBR = F передаётся 16 сообщений. Если в очереди меньше сообщений(дескрипторов), чем указано в MSG\_NMBR, то после передачи последнего сообщения произойдёт автоматический переход на следующую очередь указанную в POINTER.

#### 11.3.4.8 Регистр RX\_CR

Формат регистра RX\_CR (Receive Control Register) приведен в Таблица 11.31.

Таблица 11.31. Формат регистра RX\_CR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	–	Не используется	R	0
15:0	RX_QUEUE_IN_ORDER	Режим приема пакетов в много пакетном сообщении: 0 – прием пакетов в любом порядке; 1 – прием пакетов только в порядке возрастания номера пакета (поле <b>msgseg</b> в пакете). Каждой очереди соответствует один разряд регистра. Используется для применений, со специальными информационными потоками (flows).	RW	0

### 11.3.4.9 Регистр RX\_QTCR

Формат регистра RX\_QTCR (Receive Queue Teardown Command Register) приведен в Таблица 11.32.

Таблица 11.32. Формат регистра RX\_QTCR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	–	Не используется	R	0
15:0	RX_QUEUEn_TEARDWN	Программное прекращение приема сообщений. При записи 1 в разряд n начинает выполняться процедура прекращения приема сообщений в очередь n. После ее окончания разряд аппаратно обнуляется.	RW	0

### 11.3.4.10 Регистр TX\_QTCR

Формат регистра TX\_QTCR (Transmit Queue Teardown Command Register) приведен в Таблица 11.33.

Таблица 11.33. Формат регистра TX\_QTCR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	–	Не используется	R	0
15:0	TX_QUEUEn_TEARDWN	Программное прекращение передачи сообщений. При записи 1 в разряд n начинает выполняться процедура прекращения передачи сообщений в очереди n. После ее окончания разряд аппаратно обнуляется.	RW	0

### 11.3.4.11 Регистр MPU\_IRQ\_SR

Формат регистра MPU\_IRQ\_SR (MPU Interrupt Request Status Register) приведен в Таблица 11.34.

Таблица 11.34 Формат регистра MPU\_IRQ\_SR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	TX_QUEUE_INT	<p>Признак наличия прерывания от соответствующей очереди передачи пакетов MESSAGE.</p> <p>Если в дескрипторе заказано формирование прерывания, то оно устанавливается аппаратно при окончании передачи сообщения и сброса признака OWNERSHIP в дескрипторе.</p> <p>Прерывание сбрасывается программно. Для этого, после обработки прерывания по вызвавшему его дескриптору в регистр TXQ_CDPn необходимо записать адрес этого дескриптора. Если он совпал с содержимым этого регистра, то соответствующее прерывание сбрасывается.</p> <p>Следует помнить, что во время обработки прерывания в регистре TXQ_CDPn может измениться значение адреса из-за передачи сообщения по следующему дескриптору.</p>	R	0
15:0	RX_QUEUE_INT	<p>Признак наличия прерывания от соответствующей очереди приёма пакетов MESSAGE.</p> <p>Если в дескрипторе заказано формирование прерывания, то оно устанавливается аппаратно при окончании приема сообщения и сброса признака OWNERSHIP в дескрипторе.</p> <p>Прерывание сбрасывается программно. Для этого, после обработки прерывания по вызвавшему его дескриптору в регистр RXQ_CDPn необходимо записать адрес этого дескриптора. Если он совпал с содержимым этого регистра, то соответствующее прерывание сбрасывается.</p> <p>Следует помнить, что во время обработки прерывания в регистре RXQ_CDPn может измениться значение адреса из-за приёма сообщения по следующему дескриптору.</p>	R	0

### 11.3.5 Архитектурные регистры логического и транспортного уровней RapidIO

#### 11.3.5.1 Регистр DEV\_ID\_CAR

Формат регистра DEV\_ID\_CAR (Device Identity Capability Register) приведен в Таблица 11.35.

Таблица 11.35 Формат регистра DEV\_ID\_CAR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	DEV_ID	Идентификатор устройства. Присваивается разработчиком устройства.	RW	0
15:0	DEV_VNDR_ID	Идентификатор предприятия разработчика (изготовителя) устройства. Присваивается организацией RapidIO.	RW	0

#### 11.3.5.2 Регистр DEV\_INFO\_CAR

Формат регистра DEV\_INFO\_CAR (Device Information Capability Register) приведен в Таблица 11.36.

Таблица 11.36 Формат регистра DEV\_INFO\_CAR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	DEV_REV	Версия устройства.	RW	0

#### 11.3.5.3 Регистр ASBLY\_ID\_CAR

Формат регистра ASBLY\_ID\_CAR (Assembly Identity Capability Register) приведен в Таблица 11.37.

Таблица 11.37 Формат регистра ASBLY\_ID\_CAR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	ASSY_ID	Идентификатор системы. Присваивается разработчиком устройства.	RW	0
15:0	ASSY_VNDR_ID	Идентификатор предприятия разработчика (изготовителя) системы, в которой используется данное устройство. Присваивается организацией RapidIO.	RW	0

#### 11.3.5.4 Регистр ASBLY\_INFO\_CAR

Формат регистра ASBLY\_INFO\_CAR (Assembly Information Capability Register) приведен в Таблица 11.38.

Таблица 11.38 Формат регистра ASBLY\_INFO\_CAR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	ASSY_REV	Версия системы. Присваивается разработчиком системы.	RW	0
15:0	EF_PTR	Указатель на следующий блок структуры данных	RW	100

### 11.3.5.5 Регистр PE\_FEATURES\_CAR

Формат регистра PE\_FEATURES\_CAR (Processing Element Features Capability Register) приведен в Таблица 11.39.

Таблица 11.39 Формат регистра PE\_FEATURES\_CAR (32'h 6000\_041D)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	–	Не используется	R	0
30	MEMORY	Признак того что данное устройство содержит локальную память	R	1
29	PROCESSOR	Признак того, что данное устройство является процессором	R	1
28:11	–	Не используется	R	0
10	MULTICAST_SUPPORT	Признак того, что данное устройство поддерживает расширение MULTICAST	R	1
9:7	–	Не используется	R	0
6	RETRANS_SUPPRES_SUPPORT	Признак того, что данное устройство не поддерживает подавление повторной передачи при ошибках CRC в пакетах. Т.е. восстановление из ошибок CRC и повторная передача пакетов всегда разрешены.	R	0
5	CRF_SUPPORT	Признак того, что данное устройство не поддерживает индикаторы Critical Request Flow (CRF)	R	0
4	LARGE_SUPPORT	Признак того, что данное устройство поддерживает транспортную систему с 16-разрядным адресом	R	1
3	EXT_FEATURES	Признак того, что имеется указатель на следующий блок структуры данных	R	1
2:0	EXT_ADDR	Признак того, что данное устройство поддерживает 34- и 66-разрядный адрес в исходящих и входящих операциях	R	101

### 11.3.5.6 Регистр SRC\_OP\_CAR

Формат регистра SRC\_OP\_CAR (Source Operation Capability Register) приведен в Таблица 11.40. Регистр SRC\_OP\_CAR показывает, какие типы выходных операций может инициировать устройство.

Таблица 11.40 Формат регистра SRC\_OP\_CAR (32'h0000\_FDF4)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	–	Не используется	R	0
15	NREAD	Признак того, что данное устройство обеспечивает выполнение операции NREAD	R	1
14	NWRITE	Признак того, что данное устройство обеспечивает выполнение операции NWRITE	R	1
13	SWRITE	Признак того, что данное устройство обеспечивает выполнение операции SWRITE	R	1
12	NWRITE_R	Признак того, что данное устройство обеспечивает выполнение операции NWRITE_R	R	1
11	MESSAGE	Признак того, что данное устройство обеспечивает выполнение операции MESSAGE	R	1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
10	DOORBELL	Признак того, что данное устройство обеспечивает выполнение операции DOORBELL	R	1
9	ATOMIC COMPARE-AND-SWAP	Признак того, что данное устройство НЕ обеспечивает выполнение операции ATOMIC COMPARE-AND-SWAP	R	0
8	ATOMIC TEST-AND-SWAP	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC TEST-AND-SWAP	R	1
7	ATOMIC INCREMENT	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC POST-INCREMENT-THE-DATA	R	1
6	ATOMIC DECREMENT	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC POST-DECREMENT-THE-DATA	R	1
5	ATOMIC SET	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC SET-THE-DATA	R	1
4	ATOMIC CLEAR	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC CLEAR-THE-DATA	R	1
3	ATOMIC SWAP	Признак того, что данное устройство НЕ обеспечивает выполнение операции ATOMIC SWAP	R	0
2	MAINTENANCE PORT_WRITE	Признак того, что данное устройство обеспечивает выполнение операции MAINTENANCE PORT-WRITE	R	1
1:0	–	Не используется	R	0

### 11.3.5.7 Регистр DEST\_OP\_CAR

Формат регистра DEST\_OP\_CAR (Destination Operation Capability Register) приведен в Таблица 11.41. Регистр DEST\_OP\_CAR показывает, какие типы входных операций разрешены для обработки.

Таблица 11.41 Формат регистра DEST\_OP\_CAR (32'h0000\_FDF4)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	–	Не используется	R	0
15	NREAD	Признак того, что данное устройство обеспечивает выполнение операции NREAD	RW	1
14	NWRITE	Признак того, что данное устройство обеспечивает выполнение операции NWRITE	RW	1
13	SWRITE	Признак того, что данное устройство обеспечивает выполнение операции SWRITE	RW	1
12	NWRITE_R	Признак того, что данное устройство обеспечивает выполнение операции NWRITE_R	RW	1
11	MESSAGE	Признак того, что данное устройство обеспечивает выполнение операции MESSAGE	RW	1
10	DOORBELL	Признак того, что данное устройство обеспечивает выполнение операции DOORBELL	RW	1
9	ATOMIC COMPARE-AND-SWAP	Признак того, что данное устройство НЕ обеспечивает выполнение операции ATOMIC COMPARE-AND-SWAP	R	0
8	ATOMIC	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC	RW	1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
	TEST-AND-SWAP	печивает выполнение операции ATOMIC TEST-AND-SWAP		
7	ATOMIC INCREMENT	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC POST-INCREMENT-THE-DATA	RW	1
6	ATOMIC DECREMENT	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC POST-DECREMENT-THE-DATA	RW	1
5	ATOMIC SET	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC SET-THE-DATA	RW	1
4	ATOMIC CLEAR	Признак того, что данное устройство обеспечивает выполнение операции ATOMIC CLEAR-THE-DATA	RW	1
3	ATOMIC SWAP	Признак того, что данное устройство НЕ обеспечивает выполнение операции ATOMIC SWAP	R	0
2	MAINTENANCE PORT_WRITE	Признак того, что данное устройство обеспечивает выполнение операции MAINTENANCE PORT-WRITE	RW	1
1:0	–	Не используется	R	0

При необходимости, содержимое этого регистра может быть изменено для запрещения обработки некоторых входящих операций. Если входящая операция запрещена и на неё требуется ответный пакет, то он выдаётся со статусом «ERROR» и без данных.

### 11.3.5.8 Регистр PE\_LOG\_CSR

Формат регистра PE\_LOG\_CSR (Processing Element Logical Layer Control CSR) приведен в Таблица 11.42.

Таблица 11.42 Формат регистра PE\_LOG\_CSR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:3	–	Не используется	R	0
2:0	EXT_ADDR_CTR	Число разрядов адреса в исходящих и входящих операциях: 001 – 34 разряда, поля {xambs, address} формируют 34-ёх разрядный адрес в пакетах типа 2,5 и 6; 100 – 66 разрядов, при этом присутствует поле <b>extended address</b> в пакетах типа 2,5 и 6, поля {xambs, extended address, address} формируют 66-ти разрядный адрес. Остальные коды не используются.	RW	001

### 11.3.5.9 Регистр *BASE\_DEVICE\_ID\_CSR*

Формат регистра *BASE\_DEVICE\_ID\_CSR* (Base Device ID Command and Status Register) приведен в Таблица 11.43.

Таблица 11.43 Формат регистра *BASE\_DEVICE\_ID\_CSR*

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:24	–	Не используется	R	0
23:16	BASE_DEV_ID	8-разрядный идентификатор устройства для небольшой транспортной системы	RW	FF или 00
15:0	LARGE_DEV_ID	16-разрядный идентификатор устройства для большой транспортной системы	RW	FFFF или 0000

Исходное состояние полей *BASE\_DEV\_ID*, *LARGE\_DEV\_ID* этого регистра зависит от состояния входа *HOST*. Если *HOST*=0, то их исходное состояние – все 1, иначе – все 0. Если в процессе работы в любой момент времени вход *HOST* изменит своё состояние из 1->0, то в регистр *BASE\_DEVICE\_ID\_CSR* пропишутся все 1.

### 11.3.5.10 Регистр *HOST\_BASEID\_LOCK\_CSR*

Формат регистра *HOST\_BASEID\_LOCK\_CSR* (Host Base Device ID Lock Command and Status Register) приведен в Таблица 11.44.

Таблица 11.44 Формат регистра *HOST\_BASEID\_LOCK\_CSR*

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	–	Не используется	R	0
15:0	HOST_BASE_DEVICEID_LOCK	Идентификатор устройства для внешнего процессорного элемента (Host), который инициализирует данное устройство. Запись нового значения может быть выполнена только тогда, когда в этом поле записан код FFFF. Все другие записи игнорируются, за исключением случая, когда записываемая величина равна содержимому этого регистра. В этом случае в поле устанавливается код FFFF и далее можно записать новое значение.	RW	FFFF или 0000

Исходное состояние поля *HOST\_BASE\_DEVICEID\_LOCK* этого регистра зависит от состояния входа *HOST*. Если *HOST*=0, то его исходное состояние – все 1, иначе – все 0. Если в процессе работы в любой момент времени *HOST* изменит своё состояние из 1->0, то в регистр *HOST\_BASEID\_LOCK\_CSR* пропишутся все 1.

### 11.3.5.11 Регистр *COPM\_TAG\_CSR*

Формат регистра *COPM\_TAG\_CSR* (Component TAG Command and Status Register) приведен в Таблица 11.45. В регистр *COMP\_TAG\_CSR* можно записать любую информацию по усмотрению пользователя.

Таблица 11.45 Формат регистра *COMP\_TAG\_CSR*

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:0	COMPONENT_TAG	Тэг устройства. Устанавливается программно при инициализации.	RW	0



### 11.3.6 Архитектурные регистры физического уровня RapidIO

#### 11.3.6.1 Регистр BLOCK\_HEADER

Формат регистра BLOCK\_HEADER приведен в Таблица 11.46.

Таблица 11.46. Регистр BLOCK\_HEADER (1x/4x LP-Serial Register Block Header)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	EF_PTR	Указатель на следующий блок структуры данных	R	0
15:0	EF_ID	ID расширенных возможностей.	R	001

#### 11.3.6.2 Регистр LINK\_TIMEOUT

Формат регистра LINK\_TIMEOUT приведен в Таблица 11.47.

Таблица 11.47. Регистр LINK\_TIMEOUT (Port Link Time-out Control CSR)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:8	LTIMEOUT	Значение Time-out для ожидания символа подтверждения приема пакета или получения символа Link-Response. Является коэффициентом деления для основного делителя счётчика таймера в регистре LINK_TIMER_COUNT.	RW	FFFFFF
7:0	–	Резерв.	R	0

#### 11.3.6.3 Регистр RESP\_TIMEOUT

Формат регистра RESP\_TIMEOUT приведен в Таблица 11.48.

Таблица 11.48. Регистр RESP\_TIMEOUT (Port Response Time-out Control CSR)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:8	RTIMEOUT	Значение Time-out для ожидания ответного пакета. Используется на логическом уровне. Является коэффициентом деления для основного делителя счётчика таймера в регистре TIMER_COUNT.	RW	FFFFFF
7:0	–	Резерв.	R	0

### 11.3.6.4 Регистр GENERAL\_CSR

Формат регистра GENERAL\_CSR приведен в Таблица 11.49.

Таблица 11.49. Регистр GENERAL\_CSR (Port General Control CSR)

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31	HOST	Тип данного устройства: 0 – agent или slave устройство; 1 – host устройство. Host устройство ответственно за диагностику, инициализацию и поддержку системы. Обычно host устройство инициализирует agent или slave устройства.	RW	0 или 1
30	MASTER_ENABLE	Бит MASTER_ENABLE показывает, может ли устройство посылать запросы в систему: 0 – устройство не выдает запросы; 1 – устройство может выдавать запросы. Если MASTER_ENABLE сброшен, то устройство может только отвечать на внешние запросы.	RW	0 или 1
29	DISCOVERED	Устройство было обнаружено процессорным элементом ответственным за конфигурацию системы. 0 – устройство не обнаружено; 1 – устройство обнаружено другим процессорным элементом.	RW	0 или 1
28:0	–	Резерв	R	0

Исходное состояние полей HOST, MASTER\_ENABLE и DISCOVERED этого регистра зависит от состояния входа HOST. Если HOST=0, то их исходное состояние – все 0, иначе – все 1. Если в процессе работы в любой момент времени вход HOST изменит своё состояние из 0->1, то в регистр GENERAL\_CSR пропишутся все 1.

Поля HOST, MASTER\_ENABLE и DISCOVERED являются обычными регистрами для чтения/записи и на функционирование LPU\_PCS не влияют. Обработка этих полей должна проводиться программно. На основе значений этих полей программа управляет портом SRIO. Например, запускает выдачу пакетов, только после установки бита MASTER\_ENABLE и т.д.

### 11.3.6.5 Регистр ERROR\_STATUS\_CSR

Формат регистра ERROR\_STATUS\_CSR приведен в Таблица 11.50.

Таблица 11.50. Регистр ERROR\_STATUS\_CSR (Port Error and Status CSR)

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:21	–	Резерв	R	0
20	OUT_RTY_ENC	Устанавливается одновременно с битом 18. Сбрасывается посредством записи 1 в этот бит.	RW1C	0
19	OUT_RETRIED	Устанавливается одновременно с битом 18. Сбрасывается после получения символа Packet-Accepted или символа Packet-Not-Accepted.	R	0
18	OUT_RTY_STOP	Выходной порт находится в состоянии Output Retry Stop (после получения символа Packet-Retry).	R	0

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
17	OUT_ERR_ENC	Устанавливается одновременно с битом 16. Сбрасывается посредством записи 1 в этот бит.	RWIC	0
16	OUT_ERR_STOP	Выходной порт находится в состоянии Output Error Stop.	R	0
15:12	–	Резерв	R	0
11	INP_RTY_ENC	Устанавливается одновременно с битом 10. Сбрасывается посредством записи 1 в этот бит. Этого бита в стандарте нет, он добавлен разработчиком.	RWIC	0
10	INP_RTY_STOP	Входной порт находится в состоянии Input Retry Stop.	R	0
9	INP_ERR_ENC	Устанавливается одновременно с битом 8. Сбрасывается посредством записи 1 в этот бит.	RWIC	0
8	INP_ERR_STOP	Входной порт находится в состоянии Input Error Stop.	R	0
7:5	–	Резерв	R	0
4	PWRITE_PEND	Устанавливается, если порт оказался в состоянии, при котором он должен инициализировать операцию Maintenance Port-write. Т.е. бит устанавливается, когда порт передал пакет Maintenance Port-write и не зависит от наличия подтверждения приёма этого пакета от соседнего устройства. Сбрасывается посредством записи 1 в этот бит.	RWIC	0
3	–	Резерв	R	0
2	PORT_ERROR	Устанавливается при возникновении невосстанавливаемой ошибки Fatal Error и при потере инициализации, когда бит PORT_UNINIT переходит из 0 в 1. Этот сигнал идёт на прерывание. При установке данного бита, его можно сбросить только установкой всего порта в исходное состояние, путём общего асинхронного сброса, либо при помощи бита CLR из регистра PCS_CSR. Хотя в стандарте просят сброс этого бита путём записи 1 в него.	R	0
1	PORT_OK	Порт инициализирован, обменивается достоверными символами и готов к передаче пакетов. Устанавливается после инициализации, когда порт принял 7 достоверных безошибочных символов статуса и сам передал 15 символов статуса. Сбрасывается при потере инициализации. Здесь ушли от требования стандарта, чтобы биты PORT_OK и PORT_UNINIT являлись взаимно-исключаемыми.	R	0
0	PORT_UNINIT	Порт не инициализирован. При установке инициализации этот бит сбрасывается независимо от передачи/приёма символов Status. При потере инициализации этот бит устанавливается. Здесь ушли от требования стандарта, чтобы биты PORT_OK и PORT_UNINIT являлись взаимно-исключаемыми.	R	1

### 11.3.6.6 Регистр CONTROL\_CSR

Формат регистра CONTROL\_CSR приведен в Таблица 11.51.

Таблица 11.51. Регистр CONTROL\_CSR (Port Control CSR)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	PORT_WIDTH	Количество каналов порта - аппаратно 4 канала.	R	01
29:27	INIT_PORT_WIDTH	Конфигурация порта после инициализации: 000 – инициализирован канал 0; 001 – инициализирован канал 2; 010 – инициализированы все 4 канала; 111 – инициализации нет; 011:110 – резерв.	R	111
26:24	PWIDTH_OVERRIDE	Согласно стандарту: программная инициализация/переинициализация порта в указанный режим: 000 – не было программной настройки порта; 001 – резерв; 010 – инициализирован канал 0; 011 – инициализирован канал 2; 100:111 – резерв. Реализация: Это поле аппаратно не используется, просто регистр для чтения/записи. Должна быть программная обработка этого поля.	RW	000
23	PORT_DIS	Режим работы порта: 0 – порт находится в рабочем состоянии; 1 – порт находится в нерабочем состоянии. Реализация: Этот бит аппаратно не используется, просто регистр для чтения/записи. Должна быть программная обработка этого поля.	RW	0
22	OUT_PENA	Согласно стандарту: разрешение работы выходного порта: 0 – порт может инициировать на передачу только пакеты Maintenance. 1 – порт может инициировать на передачу любые пакеты. Реализация: Этот бит аппаратно не используется, просто регистр для чтения/записи. Должна быть программная обработка этого бита.	RW	0
21	IN_PENA	Разрешение работы входного порта: 0 – входной порт может принимать только пакеты типа MAINTENANCE. В ответ на остальные пакеты выдаются символы Packet-Not-Accepted. Символы принимаются и выдаются нормально. 1 – входной порт может принимать любые пакеты.	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
20	ERR_CHK_DIS	Запрещение контроля всех ошибок передачи: 0 – контроль ошибок и восстановление разрешено; 1 – контроль ошибок и восстановление запрещено. В общем случае, при отключенном контроле ошибок поведение устройства при возникновении ошибочной ситуации не определено. Реализация: Маскирование всех возможных ошибок Input Error, Output Error и Fatal Error.	RW	0
19:18	–	Резерв	R	0
17	ENUM_BOUNDARY	Этот бит аппаратно не используется, просто регистр для чтения/записи. Это флаг используется в специальных программных алгоритмах.	RW	0
16:12	–	Резерв	R	0
11:4	RE-TRANS_SUPPRES_MASK	Подавление повторной передачи пакетов при ошибках CRC. Подавление не поддерживается, повторная передача пакетов всегда разрешена.	R	00
3:1	–	Резерв	R	0
0	PORT_TYPE	Тип порта – последовательный.	R	1

### 11.3.7 Дополнительные регистры физического уровня

#### 11.3.7.1 Регистр PCS\_CSR

Формат регистра PCS\_CSR приведен в Таблица 11.52.

Таблица 11.52. Регистр PCS\_CSR (PCS Control and Status Register)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:26	-	Не используется	R	00
25:24	SYNC_STATE	Состояние выполнения процедуры синхронизации: 00 – NO SYNC (после сброса); 01 – NO SYNC 1; 10 – SYNC; 11 – SYNC 1.	R	00
23:21	-	Не используется	R	00
20	SYNC_ERROR	Устанавливается при потере линейной синхронизации, т.е. синхронизация была, а потом пропала. Сбрасывается посредством записи 1 в этот бит. SYNC_ERROR может возникнуть только после завершения инициализации. В процессе проведения инициализации потеря синхронизации не устанавливает SYNC_ERROR.	RW1C	0
19:15	-	Не используется	R	00

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
14:12	INIT_STATE	Состояние выполнения процедуры инициализации: 001 – SILENT (после сброса); 010 – SEEK; 011 – DISCOVERY1; 100 – DISCOVERY2; 101 – INIT – канал инициализирован	R	001
11	FORCE_REINIT	Сигнал запуска инициализации PCS. Запись 1 вызывает инициализацию. Всегда считывается 0. Для переинициализации необходимо провести программный сброс CLR в регистре PCS_CSR и заново запустить инициализацию FORCE_REINIT. Если случилась потеря инициализации и установлен бит PORT_ERROR, то его необходимо сбросить с помощью бита CLR из регистра PCS_CSR. Только после этого можно заново провести инициализацию. <b>Результат инициализации отображается в поле INIT_STATE регистра PCS_CSR либо в поле INIT_PORT_WIDTH регистра CONTROL_CSR.</b>	W1	0
10:8	-	Не используется	R	000
7:4	GEN_ERR	Тестовый сигнал генерации одиночной ошибки в первом слове пакета до кодирования 8B/10B: GEN_ERR[0] – инвертируется 5 разряд в 1-ом байте пакета (1-ый байт содержит поле ackID); GEN_ERR[1] – инвертируется 5 разряд во 2-ом байте пакета; GEN_ERR[2] – инвертируется 5 разряд во 3-ем байте пакета; GEN_ERR[3] – инвертируется 5 разряд во 4-ом байте пакета. После генерации ошибки в одном пакете все разряды GEN_ERROR сбрасываются.	RW	0000
3	–	Резерв	R	0
2	INVLD_CHK_DIS	Запрещение контроля ошибок приёма в PCS: 0 – контроль ошибок разрешён; 1 – контроль ошибок запрещён. Если установлен бит INVLD_CHK_DIS, то при проведении инициализации и в режиме передачи некорректные кодовые группы INVALID игнорируются. INVLD_CHK_DIS не отключает контроль ошибок Input/Output Error.	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
1	COMPARE_EN	Разрешение сравнения в приёмнике PMA_RX несущей частоты поступающих кодовых групп с опорной частотой функционирования PMA_RX (125МГц): 1 – После начального этапа настройки выполняется сравнение частот. Если разность частот более 3%, то повторяется начальный этап. 0 – После начального этапа сравнение частот не выполняется. Начальный этап состоит из выполнения синхронизации по опорной частоте и синхронизации по перепадам входных данных. На начальном этапе сравнение частот не производится независимо от состояния этого бита.	RW	0
0	CLR	Установка блоков физического уровня (LPU и PCS) в исходное состояние. Запись 1 вызывает синхронный сброс на частоте работы LPU_PCS CLK.	W1	0

### 11.3.7.2 Регистр LPU\_CSR

Формат регистра LPU\_CSR приведен в Таблица 11.53.

Таблица 11.53. Регистр LPU\_CSR (LPU Control and Status Register)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	–	Резерв	R	0
29	IDLE_ERR_DIS	Запрещение контроля ошибок в Idle последовательности: 0 – контроль разрешён. Idle последовательность должна состоять из символов  A ,  K  и  R . При приёме любого другого символа порт входит в состояние Input-Error-Stop; 1 – контроль запрещён. В idle последовательности игнорируются любые non-idle или invalid символы.	RW	0
28	REQ_INPUT_ERR	При установке имитирует ошибку приема пакета, что вызывает генерацию символа Packet-Not-Accepted и переход в состояние Input-Error-Stop. Используется для целей тестирования, при получении символа Packet-Not-Accepted соседнее устройство переходит в состояние Output Error Stop.	RW	0
27:26	STATUS_MODE	Режим передачи символов Status: 00 – нормальный режим передачи символов Status каждые 1024 кодовые группы; 01 – блокировка передачи; 10 – передача с неправильным ackID; 11 – резерв. Неправильный ackID формируется путём инверсии старшего разряда в корректном ackID. Используется для целей тестирования.	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
25:24	LRESP_MODE	Режим передачи символа Link-Response: 00 – нормальный режим; 01 – блокировка передачи; 10 – передача с неправильным ackID; 11 – резерв. Установка режима передачи для символа Link-Response приводит к переходу соседнего устройства из состояния Output Error Stop в состояние Fatal Error. Используется для целей тестирования.	RW	0
23:22	ACK_MODE	Режим передачи символов Packet-Accepted, Packet-Retry, Packet-Not-Accepted: 00 – нормальный режим; 01 – блокировка передачи; 10 – передача с неправильным ackID; 11 – резерв. Неправильный ackID формируется путём инверсии старшего разряда в корректном ackID. Используется для целей тестирования. Блокировка передачи или передача с неправильным ackID символов Packet-Accepted, Packet-Retry приводит к переходу соседнего устройства в состояние Output Error Stop. При блокировке Packet-Retry можно осуществить выход из Input Retry Stop по символу Link-Request	RW	0
21:20	CRC_MODE	Режим формирования ошибки CRC на передачу: 00 – нормальный режим; 01 – формирование ошибки CRC пакетов данных; 10 – формирование ошибки CRC управляющих символов; 11 – формирование ошибки CRC пакетов данных и управляющих символов. Ошибка CRC формируется путём инверсии старшего разряда в корректной CRC. Используется для целей тестирования, при приёме пакетов и символов с ошибкой CRC соседнее устройство переходит в состояние Input Error Stop.	RW	0
19:16	PRESCALER	Коэффициент предварительного деления частоты TXCLK(125МГц) для таймера ответных управляющих символов: 0000 – 1; 0001 – 2; ... 1111 – 16. Это поле относится к предварительному делителю PRESCALER_CNT в регистре LINK_TIMER_COUNT.	RW	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
15	TICK_TIMER	Если EN_TIMER = 0, то при записи 1 в этот разряд выполняется программная инкрементация таймера ответных управляющих символов на 1 в регистре LINK_TIMER_COUNT. Считывается всегда ноль. Используется для целей тестирования таймера в регистре LINK_TIMER_COUNT.	W1	0
14	EN_TIMER	Разрешение работы таймера ожидания ответных управляющих символов в регистре LINK_TIMER_COUNT: 0 – работа таймера запрещена. В этом случае таймер инкрементируется на 1 при записи 1 в разряд TICK_TIMER; 1 – работа таймера разрешена.	RW	0
13	–	Резерв	R	0
12	TX_FLOW_CTR_EN	Разрешение работы в режиме TRANSMITTER FLOW CONTROL в соответствии с п. 5.6 спецификации “Serial RapidIO”: 0 – работа только в режиме Receiver Flow Control; 1 – работа в режиме Transmitter Flow Control, если не получается, то в режиме Receiver Flow Control.	RW	0
11	–	Резерв	R	0
10:8	RETRANS_CNT	Максимальное число раз приёма символов Packet-Not-Accepted, не разделенных символами Packet-Accepted или Packet-Retry. А также максимальное число раз приёма символов Packet-Retry, не разделенных символами Packet-Accepted. При RETRANS_CNT = 0 счётчики PR_CNT_OVER и PNA_CNT_OVER не работают.	RW	000
7	PR_CNT_OVER	Устанавливается, когда переполняется счетчик приёма символов Packet-Retry не разделённых символами Packet-Accepted. Максимальное количество повторов символов Packet-Retry указано в поле RETRANS_CNT. Сбрасывается посредством записи 1 в этот бит.	RWIC	0
6	PNA_CNT_OVER	Устанавливается, когда переполняется счетчик приёма символов Packet-Not-Accepted не разделённых символами Packet-Accepted или Packet-Retry. Максимальное количество повторов символов Packet-Not-Accepted указано в поле RETRANS_CNT. Сбрасывается посредством записи 1 в этот бит. Приёмник отправляет PNA не только для пакетов, но и для обнаруженных ошибок Input Error. Поэтому получается, что при отсутствии передачи пакетов, но при наличии ошибок, может установиться PNA_CNT_OVER.	RWIC	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
5	PR	Устанавливается при приёме символа Packet-Retry. Сбрасывается посредством записи 1 в этот бит.	RW1C	0
4	PNA	Устанавливается при приеме символа Packet-Not-Accepted. Сбрасывается посредством записи 1 в этот бит.	RW1C	0
3	RST_DEV_CMD	Устанавливается, если LPU принял 4 символа Link-Request/Reset-Device без промежуточных ошибок. Значит, что соседнее устройство просит осуществить сброс. Сбрасывается посредством записи 1 в этот бит.	RW1C	0
2	MCE_DEC	Устанавливается при приеме символа Multicast-Event. Сбрасывается посредством записи 1 в этот бит.	RW1C	0
1	LINK_RST_REQ	Запрос выдачи 4 символов Link-Request/Reset-Device посредством записи 1 в этот бит. Передача символов осуществляется, когда порт не передаёт пакеты, либо в промежутке между пакетами. После завершения передачи 4-ёх символов Link-Request/Reset-Device этот бит сбрасывается. По стандарту передача символов отличных от Status разрешена после завершения полной инициализации, когда PORT_OK = 1. Но по запросу LINK_RESET_REQ символы Link-Request/Reset-Device передаются независимо от PORT_OK.	RW	0
0	MCAST_REQ	Запрос выдачи символа Multicast-Event посредством записи 1 в этот бит. Символ Multicast-Event передаётся сразу, если необходимо он встраивается в пакет. После передачи символа этот бит сбрасывается. По стандарту передача символов отличных от Status разрешена после завершения полной инициализации, когда PORT_OK = 1. Но по запросу MCAST_REQ символ Multicast-Event передаётся независимо от PORT_OK.	RW	0

### 11.3.7.3 Регистр LPU\_ERROR

Формат регистра LPU\_ERROR приведен в Таблица 11.54.

Таблица 11.54. Регистр LPU\_ERROR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:27	PORT_STATUS	Отображает поле Port_status принятого символа Link-Response: 00000, 00001 – резерв; 00010 – порт обнаружил невосстанавливаемую ошибку и не способен принимать пакеты; 00011 – резерв; 00100 – порт передал символ Packet-Retry и находится в состоянии Retry-Stopped в ожидании возобновления передачи; 00101 – порт обнаружил ошибку при приеме, передал Packet-Not-Accepted и находится в состоянии Error-Stopped в ожидании возобновления передачи; 00110:01111 – резерв; 10000 – порт нормально принимает пакеты; 10001:11111 – резерв. Поле полностью доступно на запись и чтение.	RW	00000
26:25	–	Резерв	R	00
24:20	PNA_CAUSE	Отображает причину, по которой пакет не принят приёмником соседнего устройства. Отображает поле Cause принятого символа Packet-Not-Accepted : 00000 – резерв; 00001 – принят пакет с неожиданным значением ackID; 00010 – принят символ с неправильным CRC; 00011 – прерван прием пакета данных, так как остановлен приём non-maintenance пакетов; 00100 – принят пакет с неправильным CRC; 00101 – принят ошибочный (invalid) символ информации, или символ принят без ошибок, но он является недопустимым (illegal); 00110:11110 – резерв; 11111 – ошибка общего плана (general error), все остальные ошибки выявленные приёмником соседнего устройства. Поле полностью доступно на запись и чтение.	RW	00000

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
19	FATAL_ERR_STOP	<p>Порт находится в состоянии Fatal Error и не может передавать пакеты, при этом порт может обмениваться символами и принимать пакеты. Переход в Fatal Error осуществляется из состояния Output Error Stop при 2-ой неудачной попытке восстановления. Причина перехода указана в поле FATAL_ERR_CAUSE.</p> <p>При входе в Fatal Error устанавливается бит PORT_ERROR в регистре ER-ROR_STATUS_CSR.</p> <p>Сбросить этот бит можно только общим асинхронным сбросом, либо битом CLR в регистре PCS_CSR.</p>	R	0
18	–	Резерв	R	0
17:16	FATAL_ERR_CAUSE	<p>Причина перехода порта из состояния Output Error Stop в состояние Fatal Error:</p> <p>1 – Сработал таймер ожидания символа Link-Response в ответ на выданный Link-Request/Input-Status;</p> <p>2 – В ответ на выданный Link-Request/Input-Status принят символ Link-Response с некорректным значением поля ackID_status.</p> <p>Сбросить этого поле можно только общим асинхронным сбросом, либо битом CLR в регистре PCS_CSR.</p>	R	00
15:12	OUT_ERR_CAUSE	<p>Причина перехода порта в состояние Output Error Stop:</p> <p>1 – Принят символ Packet-Not-Accepted;</p> <p>2 – Сработал таймер ожидания символа подтверждения;</p> <p>3 – Принят неожиданный символ Packet-Accepted;</p> <p>4 – Принят неожиданный символ Packet-Retry;</p> <p>5 – Принят неожиданный символ Link-Response;</p> <p>6 – Принят символ Packet-Accepted с неожиданным значением поля packet_ackID;</p> <p>7 – Принят символ Packet-Retry с неожиданным значением поля packet_ackID;</p> <p>8 – Принят символ Status с неожиданным значением поля ackID_status.</p> <p>Поле показывает причину перехода порта в ближайшее предыдущее состояние Output Error Stop. Устанавливается одновременно с битом 16 из регистра ER-ROR_STATUS_CSR и хранит текущее значение до следующего перехода в Output Error Stop.</p> <p>Поле полностью доступно на запись и чтение.</p>	RW	00

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
11:8	INP_ERR_CAUSE	<p>Причина перехода порта в состояние Input Error Stop:</p> <p>1 – Idle последовательность содержит invalid или non-idle 8-битный символ;</p> <p>2 – Idle последовательность содержит 4 байта данных;</p> <p>3 – Символ содержит invalid или non-data 8-битный символ внутри 24-разрядного кода символа;</p> <p>4 – Символ с неправильной CRC;</p> <p>5 – Принят неожиданный символ Restart-From-Retry;</p> <p>6 – Принят неожиданный символ End-Of-Packet;</p> <p>7 – Принят неожиданный символ Stomp;</p> <p>8 – Пакет содержит invalid или non-data 8-битный символ;</p> <p>9 – Пакет содержит 4 символа idle последовательности;</p> <p>10 – В пакете неверное значение ackID или старший из зарезервированных бит не ноль;</p> <p>11 – Длина пакета менее 8 байт или более 276 байт;</p> <p>12 – Пакет с неправильной промежуточной или конечной CRC;</p> <p>13 – Приём Non-Maintenance пакетов остановлен битом IN_PENA;</p> <p>14 – Имитация ошибки приёма пакетов REQ_INPUT_ERR.</p> <p>Поле показывает причину перехода порта в ближайшее предыдущее состояние Input Error Stop. Устанавливается одновременно с битом 8 из регистра ERROR_STATUS_CSR и хранит текущее значения до следующего перехода в Input Error Stop.</p> <p>Поле полностью доступно на запись и чтение.</p>	RW	00
7	OUT_RTY_ENC	Повторяет состояние одноимённого бита из регистра ERROR_STATUS_CSR.	R	0
6	OUT_RTY_STOP	Повторяет состояние одноимённого бита из регистра ERROR_STATUS_CSR.	R	0
5	OUT_ERR_ENC	Повторяет состояние одноимённого бита из регистра ERROR_STATUS_CSR.	R	0
4	OUT_ERR_STOP	Повторяет состояние одноимённого бита из регистра ERROR_STATUS_CSR.	R	0
3	INP_RTY_ENC	Повторяет состояние одноимённого бита из регистра ERROR_STATUS_CSR.	R	0
2	INP_RTY_STOP	Повторяет состояние одноимённого бита из регистра ERROR_STATUS_CSR.	R	0
1	INP_ERR_ENC	Повторяет состояние одноимённого бита из регистра ERROR_STATUS_CSR.	R	0
0	INP_ERR_STOP	Повторяет состояние одноимённого бита из регистра ERROR_STATUS_CSR.	R	0

### 11.3.7.4 Регистр ACKID\_CSR

Формат регистра ACKID\_CSR приведен в Таблица 11.55.

Таблица 11.55. Регистр ACKID\_CSR (Acknowledge ID Control and Status Register)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:27	RBUF_FREE	Количество пакетов максимальной длины, которые может принять буфер приёма PL_RXB. Здесь указывается реальное число, которое не зависит от режима передачи Transmitter/Receiver Flow Control.	R	h10
26:22	TBUF_STAT	Статус буфера приёма пакетов соседнего устройства. Значение зависит от режима передачи Transmitter/Receiver Flow Control, в котором работает порт и соседнее устройство: <ul style="list-style-type: none"> <li>• Если Receiver Flow Control, то TBUF_STAT = 5'h1F</li> <li>• Если Transmitter Flow Control, то TBUF_STAT = число пакетов максимальной длины, которое может принять соседнее устройство.</li> </ul> Это поле buf_status из принимаемых символов: Packet-accepted, Packet-retry, Status.	RW	0
21	–	Резерв	R	0
20:16	ACKID_STAT	Значение AckID следующего ожидаемого на приём пакета. AckID_Status из приёмника LPU_RX.	R	0
15:11	SYMBOL_ACKID	Значение поля AckID из принятого символа: Packet-accepted, Packet-retry, Status, Link-response. Символ Packet-not-accepted не учитывается, так как в нём это поле не определено.	R	0
10	–	Резерв	R	0
9:5	TLAST_ACK	Код ackID последнего подтвержденного пакета. Может изменяться вперёд на одно или несколько значений. Всегда TLAST_ACK < TNEXT_ACK.	RW	h1F
4:0	TNEXT_ACK	Код ackID следующего пакета на передачу. Может изменяться вперёд только на одну единицу, назад на одно или несколько значений. Всегда TLAST_ACK < TNEXT_ACK.	RW	0

Регистр ACKID\_CSR используется для тестирования блока LPU\_PCS и буфера TRB. Путём записи определённых значений в поля TBUF\_STAT, TLAST\_ACK, TNEXT\_ACK создаются некоторые ситуации работы порта, которые далее отслеживаются.

В штатном режиме работы запись значений в регистр ACKID\_CSR запрещена. Запись произвольных значений в поля TBUF\_STAT, TLAST\_ACK, TNEXT\_ACK может привести к неправильной работе или зависанию блока LPU\_PCS.

### 11.3.7.5 Регистр USER\_SYMBOL

Формат регистра USER\_SYMBOL приведен в Таблица 11.56.

Таблица 11.56. Регистр USER\_SYMBOL

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	SYMBOL_REQ	Запрос передачи символа посредством записи 1 в этот бит. Передаётся символ с кодом SYMBOL_CODE, с разделителем StartControl (1C), CRC символа вычисляется автоматически. Передача осуществляется сразу по запросу, не смотря на передачу пакетов и других управляющих символов. Т.е. символ может встраиваться в пакет, либо в поток других символов. После передачи символа этот бит сбрасывается и только тогда можно передать следующий символ. По стандарту передача символов отличных от Status разрешена после завершения полной инициализации, когда PORT_OK = 1. Это ограничение на передачу USER_SYMBOL не влияет.	RW	0
30:19	–	Резерв	R	0
18:0	SYMBOL_CODE	Код символа для передачи. Длина символа 19 бит. На основе кода символа автоматически вычисляется его CRC. В это CRC можно ввести ошибку с помощью поля CRC_MODE регистра LPU_CSR. Символ передаётся с разделителем StartControl (1C).	RW	0

### 11.3.7.6 Регистр RX\_SYMBOL

Формат регистра RX\_SYMBOL приведен в Таблица 11.57.

Таблица 11.57. Регистр RX\_SYMBOL

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:24	RX_SYMBOL_DELIMITER	Разделитель принятого управляющего символа: 1C – Start Control (SC) разделитель символов, которые не являются разделителями пакетов; 7C – Packet Delimiter (PD) разделитель символов, которые являются разделителями пакетов. Поле доступно на чтение/запись.	RW	0
23:5	RX_SYMBOL_CODE	Код принятого символа. Поле доступно на чтение/запись.	RW	0
4:0	RX_SYMBOL_CRC	CRC принятого символа. Поле доступно на чтение/запись.	RW	0

### 11.3.7.7 Регистр TRB\_CSR

Формат регистра TRB\_CSR приведен в Таблица 11.58.

Таблица 11.58. Регистр TRB\_CSR (Transmitter-Receiver Buffer Control and Status Register)

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:27	REQUEST_COUNT	Количество пакетов запросов, находящихся в буфере передачи PL_TXB.	R	0
26	EMPTY	Буфер PL_TXB полностью пустой. Если в нём были пакеты, то они все переданы и их приём подтверждён.	R	1
25	FULL	Признак того, что в буфере передачи PL_TXB нет места для любых пакетов, как запросов, так и ответов. Если установлен бит DSBL_PL_TXB = 1, то считается, что буфер полон и FULL = 1.	R	0
24	FULL_REQUEST	Признак того, что в буфере передачи PL_TXB нет места для пакетов запросов, но место для ответного пакета должно быть. Если установлен бит DSBL_PL_TXB = 1, FULL_REQUEST = 1.	R	0
23	EMPTY_TX	В буфере PL_TXB нет пакетов ожидающих передачу. При этом в буфере могут находиться пакеты ожидающие подтверждение приёма.	R	1
22	SHUFFLE_EN	Разрешение процедуры перемещения пакетов в буфере PL_TXB: 0 – перемещение пакетов в PL_TXB запрещено, пакеты передаются в порядке их поступления в PL_TXB; 1 – перемещение пакетов в PL_TXB разрешено, самый подходящий для передачи пакет перемещается к выходу PL_TXB. Этот пакет будет передаваться первым. Самый подходящий для перемещения/передачи пакет определяется по следующим признакам: <ul style="list-style-type: none"> <li>• приоритет пакета;</li> <li>• тип пакета (пакет ответ либо пакет запрос).</li> </ul> Если по приоритету нет претендентов, то для перемещения выбирается любой ответный пакет. Перемещение пакетов запросов осуществляется, когда REQUEST_REORDER_EN = 1 и SHUFFLE_EN = 1. Перемещение пакетов ответов осуществляется, когда SHUFFLE_EN = 1. При этом может повышаться их приоритет, когда PROMOTION_EN = 1.	RW	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
21	REQUEST_REORDER_EN	Разрешение выполнения процедуры перемещения (shuffle) пакетов запросов: 0 – перемещение запрещено; 1 – перемещение разрешено. Перемещение пакетов запросов осуществляется, когда REQUEST_REORDER_EN = 1 и SHUFFLE_EN = 1.	RW	0
20	PROMOTION_EN	Разрешение повышения приоритета ответных пакетов: 0 – повышение запрещено; 1 – повышение разрешено. Приоритет может повышаться только при перемещении ответного пакета, когда SHUFFLE_EN = 1.	RW	0
19	DSBL_TXB	Запрещение приема исходящих из LSU_TX и MPU_TX пакетов в буфер PL_TXB: 0 – прием разрешен; 1 – прием запрещен. При DSBL_TXB = 1 устанавливаются биты FULL = 1 и FULL_REQUEST = 1. Используется для тестирования.	RW	0
18	TX_DSBL	Запрещение передачи пакетов из буфера PL_TXB в LPU для передачи в соседнее устройство: 0 – передача разрешена; 1 – передача запрещена, при этом можно осуществить пошаговую передачу пакетов (бит TX_STEP). Используется для тестирования.	RW	0
17	TX_STEP	Пошаговая передача пакета. Когда TX_DSBL = 1, при записи 1 в этот бит передается один пакет из PL_TXB в LPU для его передачи в соседнее устройство. Считывается всегда ноль. Используется для тестирования.	W1	0
16	–	Резерв	R	0
15	INSUF_PRIO_EN	Режим приема пакетов в буфер PL_RXB: 0 – пакеты принимаются вне зависимости от их приоритета. Когда буфер заполнится, на принимаемый пакет выдается символ Packet_Retry; 1 – при недостаточном месте в буфере PL_RXB, пакеты с низким приоритетом отбрасываются и выдается символ Packet-Retry, при этом пакеты с более высоким приоритетом будут приниматься. Используется в штатном режиме работы.	RW	0
14	DSBL_RXB	Запрещение приема входящих пакетов в буфер PL_RXB из LPU: 0 – прием разрешен; 1 – прием запрещен. При DSBL_RXB = 1 для принимаемого пакета LPU выдает символ Packet-Retry. Используется для тестирования.	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
13	RX_DSBL	Запрещение выдачи пакетов из буфера приёма PL_RXB в SRIO_CTR для обработки: 0 – выдача разрешена; 1 – выдача запрещена, при этом можно осуществить пошаговую передачу пакетов (бит RX_STEP). Используется для тестирования.	RW	0
12	RX_STEP	Пошаговая передача пакета. Когда RX_DSBL = 1, при записи 1 в этот бит выдаётся один пакет из PL_RXB в SRIO_CTR для обработки. Считывается всегда ноль. Используется для тестирования.	W1	0
11	–	Резерв	R	0
10:8	WATERMARK2	Код WM2, определенный в п. 5.6.2.3 стандарта. WM0>WM1>WM2	RW	0
7	–	Резерв	R	0
6:4	WATERMARK1	Код WM1, определенный в п. 5.6.2.3 стандарта. WM0>WM1>WM2	RW	0
3	–	Резерв	R	0
2:0	WATERMARK0	Код WM0, определенный в п. 5.6.2.3 стандарта. WM0>WM1>WM2	RW	0

### 11.3.7.8 Регистр TXB\_TEST

Формат регистра TXB\_TEST приведен в Таблица 11.59.

Таблица 11.59. Регистр TXB\_TEST

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	SHUFFLE_FROM	Номер слота буфера с наиболее подходящим для перемещения пакетом	R	0
27:26	–	Не используется	R	0
25	BEST_VLD	Найден наилучший пакет с приоритетом выше уровня MIN_PRI для перемещения. Здесь учитываются и пакеты запросы и пакеты ответы. Перемещение по BEST_VLD всегда приоритетней	R	0
24	ANY_VLD	Найден ответный пакет любого приоритета для перемещения. ANY_VLD учитываются только ответные пакеты. У перемещения по ANY_VLD самый низкий приоритет	R	0
23:20	NEXT_WR_ADDR	Адрес свободного слота буфера PL_TXB для записи пакета поступающего в буфер PL_TXB	R	0
19:16	NEXT_RD_ADDR	Адрес слота буфера PL_TXB занятого пакетом для чтения этого пакета на передачу в LPU	R	0
15:14	NEXT_RD_PRIO	Приоритет пакета выходящего на передачу из PL_TXB	R	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
13:12	MIN_PRIO	Минимальный уровень приоритета для передаваемых пакетов. Он вычисляется внутри буфера PL_TXB в процессе передачи пакетов. Процесс вычисления зависит от режима передачи Transmitter Flow Control/Receiver Flow Control. В штатном режиме работы порта запись в это поле запрещена.	RW	0
11:9	–	Не используется	R	0
8:4	FREE_BUF_CNT	Количество свободных слотов буфера в приёмнике с другой стороны. Оно вычисляется в процессе передачи пакетов и зависит от режима передачи. В режиме Transmitter Flow Control: FREE_BUF_CNT = TBUF_STAT – OUT_PCKT_CNT В режиме Receiver Flow Control: FREE_BUF_CNT = h'1F	R	0
3:0	OUT_PCKT_CNT	Количество переданных пакетов, на которые не пришло подтверждение приёма TLAST_ACK. Оно вычисляется в процессе передачи пакетов. В штатном режиме работы порта запись в это поле запрещена.	RW	0

Регистр TXB\_TEST используется только для тестирования буфера PL\_TXB (Physical Layer Transmitter Buffer) и отражает некоторые его аппаратные регистры. Путём записи определённых значений в поля MIN\_PRI и OUT\_PCKT\_CNT создаются некоторые ситуации работы буфера, которые отслеживаются через поля доступные для чтения.

В штатном режиме работы запись значений в регистр TXB\_TEST запрещена, так как это может привести к неправильной работе или зависанию буфера PL\_TXB и всего порта.

### 11.3.7.9 Регистр LINK\_TIMER\_COUNT

Формат регистра LINK\_TIMER\_COUNT приведен в Таблица 11.60.

Таблица 11.60. Регистр LINK\_TIMER\_COUNT

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	TIMECODE_CNT	Счетчик таймера ожидания ответных символов	RW	0
27:4	DEVIDER_CNT	Основной делитель счётчика таймера. Коэффициент деления – значение поля LTIMEOUT из регистра LINK_TIMEOUT.	RW	0
3:0	PRESCALER_CNT	Предварительный делитель счётчика таймера. Коэффициент деления - значение поля PRESCALER из регистра LPU_CSR.	RW	0

Регистр LINK\_TIMER\_COUNT используется для тестирования таймера ожидания символов подтверждения приема пакетов и символа Link-Response в ответ на Link-Request. Регистр обеспечивает программный доступ к счетчику таймера и его делителям. Счётчик работает на частоте TXCLK (125МГц).

### 11.3.7.10 Регистр LPU\_PCS\_TEST

Формат регистра LPU\_PCS\_TEST приведен в Таблица 11.61.

Таблица 11.61. Регистр LPU\_PCS\_TEST

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	LOOPBACK_PMA	Бит управления петлей передачи через приёмо-передатчики PMA_TX, PMA_RX по каналам: 0 – нормальный режим; 1 – режим петли через PMA, порт SRIO передаёт данные сам в себя. Коммутация в режиме петли осуществляется по каналам и только через цифровые части PMA_TX, PMA_RX.	RW	0
30	LOOPBACK_PCS	Режим работы PCS: 0 – нормальный режим; 1 – режим петли (LOOPBACK). Выходы кодировщика PCS коммутируются на входы декодировщика PCS, образуя петлю передачи "сам в себя". Частота приёма RXCLK коммутируется с основной частоты работы PCS (125МГц)	RW	0
29:28	SET_MODE	Программная установка режима инициализации: 00 – нет принудительной установки; 01 – установка в режим MODE_1X0; 10 – установка в режим MODE_1X2; 11 – установка в режим MODE_4X. При установке SET_MODE[1:0] в любое отличное от 00 значение, в PCS происходит следующее: машины синхронизации всех каналов устанавливаются в SYNC, машина выравнивания в ALIGN, основная машина инициализации переходит в указанный режим. Т.е. происходит установка всех машин, даже тех которые могут быть не нужны. Пока SET_MODE[1:0] не равен нулю, все машины находятся в указанных состояниях не зависимо от наличия ошибок и нарушений протокола. Используется для целей тестирования. В штатном режиме должно быть SET_MODE = 00.	RW	00

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
27:26	INIT_MODE	<p>Режим предыдущей инициализации, если она была :</p> <p>00 – исходное состояние:</p> <p>01 – инициализация в режиме MODE_1X0;</p> <p>10 – инициализация в режиме MODE_1X2;</p> <p>11 – инициализация в режиме MODE_4X.</p> <p>По этому полю можно будет понять, была ли достигнута инициализация или нет.</p> <p>Сбрасывается записью двух 11 в это поле.</p>	RW1C	00
25	–	Резерв	R	0
24	ALIGN_ERROR	<p>Аналог бита ALIGN_ERROR из регистра PCS_CSR, но без привязки к машине инициализации.</p> <p>Отслеживает потерю выравнивания по машине выравнивания на всём промежутке работы порта и при инициализации и после её завершения.</p> <p>Сбрасывается посредством записи 1 в этот бит.</p>	RW1C	0
23:20	SYNC_ERROR	<p>Аналог поля SYNC_ERROR из регистра PCS_CSR, но без привязки к машине инициализации.</p> <p>Устанавливается при потере линейной синхронизации по каналам 3:0 соответственно. Отслеживает потерю синхронизации по машине синхронизации на всём промежутке работы порта и при инициализации и после её завершения.</p> <p>Сбрасывается посредством записи 1 в соответствующий бит.</p>	RW1C	0
19:16	FULL_AFIFO	<p>Устанавливается при заполнении принятыми символами асинхронного FIFO соответствующего канала. При заполнении FIFO запись в него всё равно продолжается, что в дальнейшем должно привести к потере синхронизации.</p> <p>Сбрасывается путём записи 1 в соответствующий бит. Используется в диагностических целях.</p>	RW1C	0000

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
15:12	RXCLK_ERROR	<p>Устанавливается при отсутствии частоты приёма RXCLK от PMA_RX соответствующего канала. Определяется отсутствие RXCLK в течении 7-ми тактов основной частоты функционирования PCS(125МГц). Сбрасывается путём записи 1 в соответствующий бит. Используется в диагностических целях.</p> <p>При LOOPBACK = 1 в регистре PCS_CSR, частота RXCLK коммутируется с основной частотой работы PCS(125МГц). Поэтому в этом режиме, потери частоты RXCLK не может быть.</p> <p>При инициализации через внешнюю среду всегда устанавливается RXCLK_ERROR для соответствующего канала. Так как при включении приёмник PMA_RX не сразу генерирует частоту RXCLK, а контроль её отсутствия уже ведётся. Чтобы убедиться в отсутствии частоты RXCLK, нужно после запуска инициализации сбросить RXCLK_ERROR, а затем опросить.</p>	RW1C	0000
11:8	COMMADET8	<p>Устанавливается, когда PMA_RX соответствующего канала обнаружил 8-мь символов COMMA и засинхронизовался по ним. 8-мь символов нужны для исключения влияния переходных процессов протекающих в начале синхронизации PMA_RX. Этот бит используется в PCS, он запускает приём и декодирование поступающих от PMA_RX символов.</p> <p>Сбрасывается путём записи 1 в этот бит. В штатном режиме работы порта сброса не должно быть.</p> <p>Сброс COMMADET8 будет происходить только при наличии частоты RXCLK для соответствующего канала.</p>	RW1C	0000
7:4	COMMAEN	<p>Разрешение синхронизации по символу COMMA в PMA_RX соответствующего канала.</p> <p>Этот бит управляется программно, либо машиной инициализации PCS, причём приоритет у программного управления. В штатном режиме работы порта регистром управляет только машина инициализации, а программной записи не должно быть.</p>	RW	0000
3:0	EN_POWER	<p>Включение пары приемо-передатчиков PMA_RXn, PMA_TXn соответствующего канала.</p> <p>Этот бит управляется программно, либо машиной инициализации PCS, причём приоритет у программного управления. Используется для тестирования. В штатном режиме работы порта регистром управляет только машина инициализации, а программной записи не должно быть.</p>	RW	0000

Регистр LPU\_PCS\_TEST используется для тестирования и диагностики работы LPU\_PCS.

### 11.3.7.11 Регистр RX\_BUF\_CSR

Формат регистра RX\_BUF\_CSR приведен в Таблица 11.62.

Таблица 11.62. Регистр RX\_BUF\_CSR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:12	–	Резерв	R	0
11:8	FULL_RX_BUF	Соответствующий буфер RX_BUF <sub>n</sub> полный. Если буфер RX_BUF <sub>n</sub> заполнится, то запись в него прекратится, пока не появится свободное место. На свободное место запишется текущий символ принимаемый от PMA_RX.	R	0000
7:4	EMPTY_RX_BUF	Соответствующий буфер RX_BUF <sub>n</sub> пустой. При чтении из пустого буфера RX_BUF <sub>n</sub> выдаётся неверное значение, но к сбою в работе буфера это не приводит. Достоверные данные читаются, когда буфер не пустой.	R	1111
3:0	EN_RX_BUF	Разрешение записи в буфер RX_BUF <sub>n</sub> символов принимаемых от PMA_RX <sub>n</sub> по соответствующим каналам: 0 – запись в буфер запрещена; 1 – запись в буфер разрешена. Независимо от состояния EN_RX_BUF программное чтение из буферов разрешено всегда.	RW	0000

Регистр RX\_BUF\_CSR используется для управления буферами RX\_BUF0:RX\_BUF3. А эти буфера используются для диагностики работы приёмников PMA\_RX0:PMA\_RX3.

### 11.3.7.12 Регистры RX\_BUF<sub>n</sub>

Имеется 4-ре специальных асинхронных буфера типа FIFO для записи символов принятых с приёмника PMA\_RX соответствующего канала. Объем - 256 12-разрядных слов: 10 разрядов для хранения принятого символа, 1 разряд для запоминания сигнала COMMADET, 1 разряд для сигнала CTRL, чтобы видеть, как COMMADET и CTRL изменяются в динамике.

Буферы управляются регистром RX\_BUF\_CSR. Запись значений в буфер происходит на частоте приёма RXCLK (~125МГц) идущей от приёмника PMA\_RX. Для чтения значений из буферов существует 4-ре регистра: RX\_BUF0 : RX\_BUF3. Для того чтобы загрузить следующее значение в соответствующий регистр RX\_BUF<sub>n</sub>, необходимо произвести запись в этот регистр.

Формат регистров RX\_BUF0 : RX\_BUF3 приведен в Таблица 11.63.

Таблица 11.63. Регистр RX\_BUFn

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:14	–	Резерв	R	0
13	RX_CTRL	Принятый сигнал CTRL от PMA_RX. CTRL – признак обнаружения несущей частоты во входных данных. Несущая частота экстрагируется из изменения входных данных.	R	X
12	RX_COMMADET	Принятый сигнал COMMADET от PMA_RX. COMMADET – признак обнаружения синхронизации по символу COMMA. COMMADET соответствует принятому символу RX_CODE[9:0].	R	X
11:10	–	Резерв	R	0
9:0	RX_CODE	Код символа принятого от PMA_RX.	R	X

### 11.3.7.13 Регистры TX\_CODEn

Имеется 4-ре регистра TX\_CODE0 : TX\_CODE3 для передачи символов на PMA\_TX соответствующих каналов. По сути, они являются выходными регистрами кодировщика PCS, с возможностью программной записи и разрешением постоянной передачи EN\_TX\_CODE. Код с выхода этих регистров напрямую идёт на входы передатчиков PMA\_TX, откуда передаётся во внешнюю среду.

Формат регистров TX\_CODE0 : TX\_CODE 3 приведен в Таблица 11.64.

Таблица 11.64. Регистр TX\_CODEn

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	EN_TX_CODE	Разрешение постоянной передачи записанного символа TX_CODE[9:0] на PMA_TX. Если EN_TX_CODE = 0, то записанный программно TX_CODE[9:0] передаётся на PMA_TX только один такт, на следующих тактах в поле TX_CODE[9:0] записывается очередное значение от кодировщика PCS. Когда EN_TX_CODE =1, то запись значений от кодировщика блокируется и на PMA_TX всё время передается записанный программно TX_CODE[9:0]. В штатном режиме работы порта программной записи не должно быть и всегда EN_TX_CODE = 0.	RW	0
30:10	–	Резерв	R	0
9:0	TX_CODE	Код символа передаваемый на PMA_TX. Код записывается программно или поступает от кодировщика PCS. Приоритет у программной записи. В штатном режиме работы порта программной записи не должно быть и в TX_CODE[9:0] всегда записывается значение от кодировщика PCS.	RW	0



## 11.4 Устройство выполнения операций ввода-вывода (LSU)

### 11.4.1 Общие положения

Устройство ввода-вывода пакетов (LSU – Load-Store Unit) выполняет передачу и прием пакетов ввода-вывода в соответствии с требованиями RapidIO Interconnect Specification V1.2 Part I: Input/Output Logical Specification. LSU обеспечивает также передачу пакетов типа DOORBELL в соответствии с требованиями RapidIO Interconnect Specification V1.2 Part II: Message Passing Logical Specification.

Пакеты операций ввода-вывода содержат конкретный физический адрес памяти, по которому в указанное устройство системы RapidIO необходимо произвести запись данных или их чтение. Пакеты типа DOORBELL обрабатывает отдельная аппаратура (очередь пакетов DOORBELL).

Для выполнения исходящих операций (передачи пакетов запросов) ввода-вывода в LSU имеется 4 узла передачи LSU\_TX0 – LSU\_TX3, каждому из которых соответствует свой набор управляющих регистров LSU\_n\_CR0 – LSU\_n\_CR6. Благодаря этому в LSU может находиться до 4 операций, находящихся в процессе выполнения. Например, ожидающих ответных пакетов.

### 11.4.2 Описание операций ввода-вывода

Перечень выполняемых операций ввода-вывода приведен в Таблица 11.65.

Таблица 11.65. Перечень выполняемых операций ввода-вывода

Вид операции	Тип пакета запроса	Тип пакета ответа	Описание
Write (ftype=5)	NWRITE	–	Запись в память устройства RapidIO от 8 до 256 байт данных. Объем данных кратен 8 байтам.
	NWRITE_R	RESPONSE without DATA	Запись в память устройства RapidIO от 8 до 256 байт данных. Объем данных кратен 8 байтам.
	ATOMIC Test-and-Swap	RESPONSE with DATA	Чтение из памяти устройства RapidIO 4 байт данных и одновременно запись 4 байт данных по этому же адресу, если считанные данные равны 0.
Streaming-write (ftype=6)	SWRITE	–	Запись в память устройства RapidIO от 8 до 256 байт. Объем данных кратен 8 байтам. Заголовок пакета меньшего размера по сравнению с NWRITE: нет полей transaction, wrsize, wdptr, srcTID.
Read (ftype=2)	NREAD	RESPONSE with DATA	Чтение из памяти устройства RapidIO от 8 до 256 байт. Объем данных кратен 8 байтам.
Atomic read-modify-write (ftype=2)	Post-Increment The Data		Чтение из памяти устройства RapidIO 4 байт данных, а затем инкрементирование этих данных в памяти этого устройства.
	Post-Decrement The Data		Чтение из памяти устройства RapidIO 4 байт данных, а затем декрементирование этих данных в памяти этого устройства.
	Set The Data		Чтение из памяти устройства RapidIO 4 байт данных, а затем запись «1» во все разряды этой ячейки памяти устройства.
	Clear The Data		Чтение из памяти устройства RapidIO 4 байт данных, а затем запись «0» во все разряды этой ячейки памяти устройства.

Вид операции	Тип пакета запроса	Тип пакета ответа	Описание
Maintenance (ftype=8)	MAINTENANCE READ REQUEST	MAINTENANCE READ RESPONSE	Чтение 4 байт данных из регистров логического и физического уровней, а так же структуры данных.
	MAINTENANCE WRITE REQUEST	MAINTENANCE WRITE RESPONSE	Запись 4 байт данных в регистры логического и физического уровней, а так же структуры данных.
	MAINTENANCE PORT-WRITE REQUEST	–	Запись от 8 до 64 байт данных в специальную область памяти устройства RapidIO. Объем данных кратен 8 байтам. Используется для передачи информации об ошибках или статусной информации, например из коммутатора (switch).

SRIO обеспечивает передачу и прием пакетов запроса типа 2,5 и 6 с числом байт 8 и больше. Объем данных кратен 8 байтам. При приеме этих пакетов запроса с числом байт меньше чем 8, формируется ответный пакет со статусом ERROR.

SRIO обеспечивает передачу и прием пакетов запроса типа 8 с числом байт 4. При приеме этих пакетов запроса с числом байт не равном 4, формируется ответный пакет со статусом ERROR.

#### 11.4.2.1 Описание полей пакетов

Описание полей пакетов приведено в Таблица 11.66.

Таблица 11.66. Описание полей пакетов запроса

Поле	Число бит	Описание
prio	2	Приоритет пакета
tt	2	Определяет размерность идентификаторов устройств <b>destinationID</b> и <b>sourceID</b> : tt = 00 – 8 бит; tt = 01 – 16 бит
ftype	4	Тип формата пакета, может использоваться совместно с полем <b>transaction</b> . В пакетах 6, 10 и 11 типов поле <b>transaction</b> отсутствует
destinationID	8 или 16	Идентификатор устройства назначения пакета
sourceID	8 или 16	Идентификатор устройства источника пакета
transaction	4	Тип транзакции, используется совместно с полем <b>ftype</b> . В пакетах 6, 10 и 11 типов поле <b>transaction</b> отсутствует
wrsize	4	Размер поля данных для операций записи. Используется совместно с полем <b>wdptr</b> . Если размер данных больше чем 8 байт, то это поле определяет максимальный размер поля данных, которые должно ожидать приемное устройство RapidIO, а данных может быть меньше
rdsizе	4	Размер поля данных для операций чтения. Используется совместно с полем <b>wdptr</b>
wdptr	1	Указатель слова. Используется совместно с полями <b>rdsizе</b> или <b>wrsize</b>
srcTID	8	Номер транзакции в пакетах запроса
targetTID	8	Номер транзакции в пакетах ответа
info	16	Поле информации пакета DOORBELL
xambs	2	Расширение поля адреса до 34 или 66 разрядов (старшие 2 разряда)
extended address	32	Расширенный 32-разрядный адрес для формирования 66-разрядного адреса { <b>xambs, extended address, address</b> }

Поле	Число бит	Описание
address	29	Старшие 29 разрядов адреса двойного слова. Полный 32-разрядный адрес выровнен по границе 8 байт (младшие 3 разряда равны нулю). Формирует 34 разрядный адрес { <b>xambs, address</b> } либо 66 разрядный адрес { <b>xambs, extended address, address</b> }.
hop_count	8	Это поле относится к пакетам 8-ого типа (MAINTENANCE). Определяет число коммутаторов, через которые должен пройти пакет запроса MAITENANCE. Используется для адресации коммутаторов, которые не имеют <b>deviceID</b> . При получении пакета запроса MAITENANCE коммутатор проверяет <b>hop_count</b> . Если он равен 0, то этот пакет предназначен данному коммутатору. Если нет, то это поле декрементируется на 1 и пакет передается дальше.
config-offset	21	Адрес архитектурных регистров RapidIO, выровненный по границе двойного слова. Используется в операциях типа MAINTENANCE.
status	4	Статус выполнения операции. Используется в ответных пакетах.
rsrv	–	Резерв. Согласно стандарту зарезервированные поля должны быть нулевыми.
double-word	64	Двойное 64-разрядное слово данных пакета.

Кодировка поля **status** в ответных пакетах приведена в Таблица 11.67.

Таблица 11.67. Кодировка поля **status**

Код	Условное обозначение	Назначение
0000	DONE	Нормальное завершение операции
0001-0010	–	Резерв
0011	RETRY	Запрос не обработан, должен быть повтор запроса. Статус RETRY может быть только в ответных пакетах для запроса DOORBELL
0100-0110	–	Резерв
0111	ERROR	Обнаружена невосстанавливаемая ошибка
1000-1111	–	Резерв

Кодировка поля **rdsize** в пакетах запроса приведена в Таблица 11.68.

Таблица 11.68. Описание поля **rdsize**

Код поля wptr	Код поля rdsize	Число байт	Расположение байт
0	0000	1	10000000
0	0001	1	01000000
0	0010	1	00100000
0	0011	1	00010000
1	0000	1	00001000
1	0001	1	00000100
1	0010	1	00000010
1	0011	1	00000001
0	0100	2	11000000
0	0101	3	11100000
0	0110	2	00110000
0	0111	5	11111000
1	0100	2	00001100
1	0101	3	00000111
1	0110	2	00000011
1	0111	5	00011111
0	1000	4	11110000
1	1000	4	00001111
0	1001	6	11111100
1	1001	6	00111111
0	1010	7	11111110

Код поля wdptr	Код поля rdsize	Число байт	Расположение байт
1	1010	7	01111111
0	1011	8	11111111
1	1011	16	11111111
0	1100	32	11111111
1	1100	64	11111111
0	1101	96	11111111
1	1101	128	11111111
0	1110	160	11111111
1	1110	192	11111111
0	1111	224	11111111
1	1111	256	11111111

Кодировка поля **wrsz** в пакетах запроса приведена в Таблица 11.69.

Таблица 11.69. Описание поля **wrsz**

Код поля wdptr	Код поля wrsz	Число байт	Расположение байт
0	0000	1	10000000
0	0001	1	01000000
0	0010	1	00100000
0	0011	1	00010000
1	0000	1	00001000
1	0001	1	00000100
1	0010	1	00000010
1	0011	1	00000001
0	0100	2	11000000
0	0101	3	11100000
0	0110	2	00110000
0	0111	5	11111000
1	0100	2	00001100
1	0101	3	00000111
1	0110	2	00000011
1	0111	5	00011111
0	1000	4	11110000
1	1000	4	00001111
0	1001	6	11111100
1	1001	6	00111111
0	1010	7	11111110
1	1010	7	01111111
0	1011	8	11111111
1	1011	16 максимум	11111111
0	1100	32 максимум	11111111
1	1100	64 максимум	11111111
0	1101	Не используется	11111111
1	1101	128 максимум	11111111
0	1110	Не используется	11111111
1	1110	Не используется	11111111
0	1111	Не используется	11111111
1	1111	256 максимум	11111111

SRIO обеспечивает передачу и прием пакетов запроса типа 2,5 и 6 с полем данных размером 8 и более байт. Объём данных кратен 8 байтам. При приеме этих пакетов запроса с числом байт меньше чем 8, формируется ответный пакет со статусом ERROR и без данных.

SRIO обеспечивает передачу и прием пакетов запроса MAINTENANCE READ/WRITE REQUEST с полем данных размером только 4 байта. При приеме этих пакетов запроса с числом байт не равном 4, формируется ответный пакет со статусом ERROR и без данных.

SRIO обеспечивает передачу и прием пакетов запроса MAINTENANCE PORT-WRITE REQUEST с полем данных размером от 8 до 64 байт. Объем данных кратен 8 байтам. При приеме этого пакета с неверным числом байт, этот пакет выкидывается, ответный пакет не выдается, так как он не требуется.

#### 11.4.2.2 Форматы пакетов

Сводная таблица состава пакетов приведена в Таблица 11.70.

Таблица 11.70. Состав пакетов

Тип пакета	Поле пакета							
	ftype	transaction	wrsize/ rdsize/ status	wdptr	srcTID/ target- TID	address/ config- offset	xambs	data
NWRITE	0101	0100	wrsize	+	rsrv	address	+	+
NWRITE_R		0101			srcTID			
ATOMIC Test-and-Swap		1110						
SWRITE	0110	–	–	rsrv	–	address	+	+
NREAD	0010	0100	rdsize	+	srcTID	address	+	–
ATOMIC Post-Increment The Data		1100						
ATOMIC Post-Decrement The Data		1101						
ATOMIC Set The Data		1110						
ATOMIC Clear The Data		1111						
RESPONSE with DATA	1101	1000	status	–	target- TID	–	–	+
RESPONSE without DATA		0000						–
MAINTENANCE READ REQUEST	1000	0000	rdsize	+	srcTID	config- offset	rsrv	–
MAINTENANCE WRITE REQUEST		0001	wrsize					+
MAINTENANCE READ RESPONSE		0010	status	rsrv	target- TID	rsrv		+
MAINTENANCE WRITE RESPONSE		0011						–
MAINTENANCE PORT-WRITE REQUEST		0100	wrsize	+	rsrv	rsrv		+
DOORBELL	1010	rsrv	rsrv	–	srcTID	–	–	поле info

**rsrv** – зарезервированные поля. Они присутствуют в пакете, но не несут никакой смысловой нагрузки. Согласно стандарту зарезервированные поля должны быть равны нулю, в LSU это делается автоматически при формировании заголовка пакета.

Все пакеты имеют поля **prio**, **tt**, **sourceID**, **destinationID**. В пакетах DOORBELL поле **info** входит в состав заголовка пакета.

Форматы пакетов, приведенные в данном пункте, определены для входов и выходов буферов PL\_TXB, PL\_RXB. Пакеты внутри SRIO передаются по 64-разрядным шинам. Поля пакетов передаются в порядке их номеров, начиная со старших разрядов, то есть prio[1:0], tt[1:0] и т.д.

#### 11.4.2.2.1 Пакеты запроса

Формат пакетов запроса NREAD, ATOMIC POST-INCREMENT THE DATA, ATOMIC POST-DECREMENT THE DATA, ATOMIC SET THE DATA, ATOMIC CLEAR THE DATA приведен в Таблица 11.71.

**Таблица 11.71. Формат пакетов запроса NREAD, ATOMIC POST-INCREMENT THE DATA, ATOMIC POST-DECREMENT THE DATA, ATOMIC SET THE DATA, ATOMIC CLEAR THE DATA**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=0010	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction	4
7	rdsizе	4
8	srcTID	8
9	extended address	0/32
10	address	29
11	wdptr	1
12	xambs	2

Формат пакетов запроса NWRITE, NWRITE\_R, ATOMIC TEST-AND-SWAP приведен в Таблица 11.72.

**Таблица 11.72. Формат пакетов запроса NWRITE, NWRITE\_R, ATOMIC TEST-AND-SWAP**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=0101	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction	4
7	wrsizе	4
8	srcTID (rsrv для NWRITE)	8
9	extended address	0/32
10	address	29
11	wdptr	1
12	xambs	2
13	double-word 0	64
...	...	...
N	double-word n	64

Формат пакетов запроса SWRITE приведен в Таблица 11.73.

**Таблица 11.73. Формат пакета запроса SWRITE**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=0110	4
4	destinationID	8/16
5	sourceID	8/16
6	extended address	0/32
7	address	29
8	rsrv	1
9	xambs	2
10	double-word 0	64
...	...	...
N	double-word n	64

Формат пакетов запроса MAINTENANCE WRITE REQUEST, MAINTENANCE PORT-WRITE REQUEST приведен в Таблица 11.74.

**Таблица 11.74. Формат пакетов запроса MAINTENANCE WRITE REQUEST, MAINTENANCE PORT-WRITE REQUEST**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=1000	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction	4
7	wrsize	4
8	srcTID (rsrv для PORT-WRITE)	8
9	hop_count	8
10	config_offset (rsrv для PORT-WRITE)	21
11	wdptr	1
12	rsrv	2
13	double-word 0	64
...	...	...
N	double-word n	64

Формат пакетов запроса MAINTENANCE READ REQUEST приведен в Таблица 11.75.

**Таблица 11.75. Формат пакетов запроса MAINTENANCE READ REQUEST**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=1000	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction	4
7	rdsizе	4
8	srcTID	8
9	hop_count	8
10	config_offset	21
11	wdptr	1
12	rsrv	2

Формат пакета запроса DOORBELL приведен в Таблица 11.76. Поле Info берется из поля DRBLL\_INFO регистра LUn\_CR5.

**Таблица 11.76. Формат пакета DOORBELL**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype = 1010	4
4	destinationID	8/16
5	sourceID	8/16
6	rsv	8
7	srcTID	8
8	info	16

#### 11.4.2.2.2 Пакеты ответа

Формат ответного пакета WRITE\_R RESPONSE приведен в Таблица 11.77.

**Таблица 11.77. Формат ответного пакета WRITE\_R RESPONSE**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=1101	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction=0000	4
7	status	4
8	targetTID	8

Формат ответного пакета NREAD, ATOMIC RESPONSE приведен в Таблица 11.78.

**Таблица 11.78. Формат ответного пакета NREAD, ATOMIC RESPONSE**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=1101	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction=1000	4
7	status	4
8	targetTID	8
9	double-word 0	64
...	...	...
N	double-word n	64

Количество данных в ответных пакетах ATOMIC RESPONSE равно одному 64-разрядному слову (double-word).

Формат ответного пакета MAINTENANCE WRITE RESPONSE приведен в Таблица 11.79.



**Таблица 11.79. Формат ответного пакета MAINTENANCE WRITE RESPONSE**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=1000	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction=0011	4
7	status	4
8	targetTID	8
9	hop_count	8
10	rsrv	24

В ответных пакетах MAINTENANCE поле **hop\_count** равно FF.

Формат ответного пакета MAINTENANCE READ RESPONSE приведен в Таблица 11.80.

**Таблица 11.80. Формат ответного пакета MAINTENANCE READ RESPONSE**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype=1000	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction=0010	4
7	status	4
8	targetTID	8
9	hop_count	8
10	rsrv	24
11	double-word 0	64
...	...	...
N	double-word n	64

В ответных пакетах MAINTENANCE поле **hop\_count** равно FF. Для данного порта SRIO количество данных в ответных пакетах MAINTENANCE READ RESPONSE равно одному 64-разрядному слову (double-word).

Формат ответного пакета DOORBELL RESPONSE приведен в Таблица 11.81.

**Таблица 11.81. Формат ответного пакета DORRBELL RESPONSE**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype = 1101	4
4	destinationID	8/16
5	sourceID	8/16
6	transaction = 0000	4
7	status	4
8	targetTID	8

### 11.4.3 Выполнение операций ввода-вывода

#### 11.4.3.1 Выполнение исходящих операций

##### 11.4.3.1.1 Запуск передачи пакетов запросов

Исходящие операции начинаются с передачи пакетов запросов. Для этого в LSU имеется 4 набора регистров L<sub>SUn</sub>\_CR0 – L<sub>SUn</sub>\_CR6. Благодаря этому в LSU может находиться до 4 операций, находящихся в процессе выполнения. То есть, ожидающие ответные пакеты.

Формат регистров L<sub>SUn</sub>\_CR0 – L<sub>SUn</sub>\_CR6 описан в п. 11.3. Поля регистров L<sub>SUn</sub>\_CR0 – L<sub>SUn</sub>\_CR6 приведены в Таблица 11.82.

Таблица 11.82. Поля регистров L<sub>SUn</sub>\_CR0 – L<sub>SUn</sub>\_CR6

Условное обозначение	Назначение
EXTENDED_ADRR[31:0]	Поле «extended address» для пакетов типа 2,5 и 6
ADRR/CONFIG[31:0]	Используется для формирования поля <b>address</b> пакетов типа 2,5 и 6. При этом младшие 3 разряда этого поля должны быть нулевыми. Используется для формирования поля <b>config offset</b> пакетов типа 8. При этом младшие 2 разряда этого поля должны быть нулевыми. 3 разряд этого поля определяет состояние бита <b>wdptr</b> пакета.
ADDR[31:0]	Адрес памяти данного микропроцессора, выровненный по границе 64-разрядного слова (младшие 3 разряда этого поля нулевые).
WORD_COUNT[15:0]	Количество передаваемых 64-разрядных слов. Используется для формирования полей пакета <b>wsize/rdsize</b> и <b>wdptr</b> : 0000 – 65536 слов; 0001 - 1 слово; 0002 - 2 слова; ... ffff – 65535 слов. Операции MAINTENANCE и ATOMIC ограничены одним пакетом с одним 32-разрядным словом.
PRIORITY[1:0]	Поле «priority» пакета: 00 – самый низкий; 11 – самый высокий. Пакеты запроса не должны иметь приоритет «11» для исключения тупиковых ситуаций в системе.
XAMSBS[1:0]	Поле «xamsbs» пакета (старшие разряды расширенного адреса).
ID_SIZE[1:0]	Поле «tt» пакета, которое определяет длину полей «sourceID» и «destination ID» пакета: 00 – 8 разрядов; 01 – 16 разрядов; 10, 11 – резерв.
DEST_ID[15:0]	Поле «destination ID» пакета. Поле «source ID» формируется аппаратно. Оно берется из регистра BDIDR (Base Device ID CSR).
INT_MASK	Маска прерывания после завершения операции ввода-вывода: 0 – прерывание запрещено; 1 – прерывание разрешено.
DRBLL_INFO[15:0]	Поле «info» пакетов типа 10 (DOORBELL)
HOP_COUNT[7:0]	Поле «hop count» пакетов 8 (MAINTENANCE)

Условное обозначение	Назначение
FTYPE[3:0]	Поле «ftype» пакетов
TRANSACTION[3:0]	Поле «transaction» пакетов
OP_STATUS[3:0]	Состояние о выполнении операции: 0000 – операция выполнена без ошибок (Posted/Non-Posted); 0001 – при выполнении Non-Posted операции (операции, в которых требуется ответный пакет) произошел таймаут; 0010 – в ответ на пакет запроса DOORBELL принят ответный пакет, содержащий в поле «status» код «RETRY». 0011 – при выполнении операции получен ответный пакет (типа 8 или 13), содержащий в поле «status» код «ERROR» или его поле данных имеет неправильную длину; 0100 – операция не может быть выполнена, так как она не реализована, или регистры LSU неправильно запрограммированы; 0101 – в ответ на пакет запроса DOORBELL принят ответный пакет, содержащий в поле «status» код «ERROR»; 0110 - операция «ATOMIC test-and-swap» не может быть выполнена из-за занятости семафора; 0111 – пакет не может быть передан из-за занятости буфера PL_TXB в момент программного запуска операции или при срабатывании таймера в случае выполнения многопакетной операции. 0111 устанавливается только при срабатывании таймера доступа к буферу PL_TXB. Этот таймер повторяет таймер таймаута ответных пакетов и работает по полю TIMECODE регистра TIMER_COUNT.
BUSY	Признак занятости регистров LSU <sub>n</sub> -CR0 – LSU <sub>n</sub> CR5. Устанавливается в «1» в момент записи данных в регистр LSU <sub>n</sub> и соответственно начале выполнения операции ввода-вывода. Сбрасывается в «0» при окончании выполнения данной операции (с ошибкой или без нее).

Для выполнения операции ввода-вывода в регистры LSU<sub>n</sub>\_CR0 – LSU<sub>n</sub>\_CR5 необходимо записать необходимую информацию. Операция начинает выполняться в момент записи данных в регистр LSU<sub>n</sub>\_CR5, поэтому это необходимо делать в последнюю очередь. В этот же момент устанавливается в «1» бит BUSY в регистре LSU<sub>n</sub>\_CR6.

На основе содержимого регистров LSU<sub>n</sub>\_CR0 – LSU<sub>n</sub>\_CR5 формируются следующие поля (заголовки) передаваемого пакета: **extended address**, **address**, **config offset**, **wrsize/rdsize** и **wdptr**, **priority**, **xambs**, **tt**, **destination ID**, **info**, **hop count**, **ftype**.

При формировании заголовка пакетов типа 2,5 и 6 в поле **address** пакета размещается содержимое ADRR/CONFIG[31:3], а при формировании заголовка пакетов типа 8 в поле **config-offset** пакета размещается содержимое ADRR/CONFIG[23:3].

Если формируется пакеты типа MAINTENANCE и ATOMIC (WORD\_COUNT должен быть равен 1), то бит **wdptr** пакета является инверсией 3 разряда поля ADRR/CONFIG.

Число разрядов поля адреса в пакете определяется регистром PE\_CSR (Processing Element Logical Layer Control CSR).

Для пакетов типа 2, 5, 8 и 10 поле **srcTID** повторяет номер LSU, из которого осуществляется передача. Для LSU0 – srcTID = 0, для LSU1 – srcTID = 1 и т.д.

Пакеты типа 6 (SWRITE) не имеют ответных пакетов. Поэтому при их передаче поле **srcTID** аппаратно устанавливается в 0 независимо от номера LSU.

Поле **targetTID** ответных пакетов повторяет поле **srcTID** соответствующего пакета-запроса.

Поле передаваемого пакета запроса **sourceID** формируется аппаратно. Оно определяется содержимым регистра BDIDR (Base Device ID CSR).

#### 11.4.3.1.2 Формирование и передача пакетов запросов

Если это пакет запроса вывода данных, то необходимые данные по DMA считываются из памяти данного микропроцессора, начиная с адреса, указанного в регистре LSUn\_CR2. Пакет формируется в буфере FIFO\_TX шириной 64-разряда с использованием содержимого регистров LSUn\_CR0 – LSUn\_CR5 и принимаемых данных по DMA. По мере формирования 64-разрядных слов пакета они передаются в буфер PL\_TXB. Для пакетов типа 8 32-разрядное слово данных для передачи размещается в памяти с учетом 3 разряда ADDR/CONFIG.

Обмен данными по DMA с памятью микропроцессора выполняется пачками по WN слов (см. регистр CSR\_SRIO).

Если это пакет запроса ввода данных, то пакет формируется только с использованием содержимого регистров LSUn\_CR0 – LSUn\_CR5. Далее он передается в буфер PL\_TXB.

LSU не обеспечивает аппаратного повтора передачи пакетов запроса. При выполнении всех операций буфер FIFO\_TX освобождается от пакета запроса, как только он переписывается в буфер PL\_TXB. То есть, если ответный пакет пришел с признаками ERROR или RETRY, то при необходимости CPU должен повторно выполнить эту операцию. Для этого достаточно произвести запись только в регистр LSUn\_CR5.

#### 11.4.3.1.3 Таймирование ответных пакетов

В соответствии с требованиями стандарта Serial RapidIO время ожидания ответных пакетов должно быть от 3 до 6 сек. Для этого имеется таймер, структурная схема которого приведена на Рисунок 11.2.

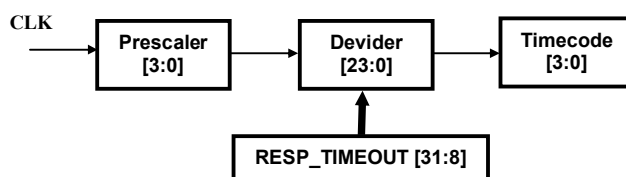


Рисунок 11.2. Структурная схема таймера

На вход таймера поступает системная частота CLK (200 – 300 МГц). Делитель Prescaler программируются (см. регистр SRIO\_CSR), а делитель Timecode имеет коэффициент деления 16. Для обеспечения времени от 3 до 6 сек, частота на выходе Prescaler должна быть от 50 до 90 МГц, а на выходе Divider – от 2,5 до 5 Гц.

Период всего таймера определяется величиной:  $\text{Timeout} = \text{период CLK} * 14 * ((\text{Prescale value} + 1) * (\text{RTIMEOUT} + 1))$ . 14 полных тактов счётчика Timecode с периодом  $((\text{Prescale value} + 1) * (\text{RTIMEOUT} + 1))$ . Когда Prescaler = 0, то входная частота на счётчик Divider передаётся без изменений. Когда RTIMEOUT = 0, то Divider передаёт входную частоту со своего входа на вход счётчика Timecode.

В каждом из 4-х LSU имеется 4-разрядный регистр. В момент окончания передачи пакета запроса, требующего ответного пакета (Non-Posted), в буфер PL\_TXB соответствующий регистр загружается содержимым делителя Timescode. При каждом изменении делителя Timescode производится сравнение 4-х разрядного регистра и делителя Timescode. При их совпадении и отсутствия ответного пакета фиксируется состояние таймаута.

#### 11.4.3.1.4 Прием ответных пакетов

Ответные пакеты могут быть с данными или без. Ответные пакеты могут поступать в любом порядке, а не только в том, в котором были переданы соответствующие им пакеты запроса.

При приеме заголовка входного пакета из буфера PL\_RXB анализируется формат пакета по содержимому полей ftupe и transaction. Если это ответный пакет, то, содержимое поля destination ID сравнивается с содержимым регистра BASE\_DEVICE\_ID\_CSR с учетом поля пакета tt. Если они не равны, то пакет выбрасывается.

Далее поля пакета srcTID и sourceID сравнивается соответственно с запомненными полями srcTID и destinationID переданных пакетов запросов. Если сравнения не произошло, то формируется соответствующий признак ошибки, и этот пакет выбрасывается. Пакет выбрасывается, и формируются соответствующие признаки ошибок, если обнаружен недопустимый код транзакции (поле transaction), или эта транзакция не реализована.

Если сравнение произошло, то:

- \* сбрасывается соответствующий таймер;
- \* содержимое поля status переписывается в регистр LSUn\_CR6;
- \* данные, при их наличии, из буфера PL\_RXB передаются в буфер FIFO\_RX для формирования 64-разрядных слов;
- \* сформированные 64-разрядные слова при помощи канала DMA записываются в память данного микропроцессора, начиная с адреса, указанного в регистре LSUn\_CR2. При этом номер этого регистра определяется содержимым поля srcTID пакета. 32-разрядное слово данных пакета типа 8 размещается в памяти с учетом 3 разряда ADDR/CONFIG;
- \* если длина пакета не соответствует его формату, то формируется соответствующий признак ошибки.

Данные в память микропроцессора не передаются, если:

- \* поле status пакета содержит код ERROR;
- \* это ответный пакет без данных, а они присутствуют в нем.

Лишние данные пакета выбрасываются.

#### 11.4.3.1.5 Окончание выполнения операции

Операции, не требующие ответных пакетов (Posted), заканчиваются сразу же после передачи пакета запроса в буфер PL\_TXB. Операции типа Posted: NWRITE, SWRITE, MAINTENANCE PORT-WRITE.

Операции, требующие ответных пакетов (Non-Posted), заканчиваются после приема достоверного ответного пакета, а если это ответный пакет с данными, то после их записи в память процессора по DMA. Операции типа Non-Posted: NREAD, NWRITE\_R, ATOMIC, MAINTENANCE WRITE/READ, DOORBELL. Ответные пакеты с данными поступают при выполнении операций NREAD и ATOMIC.

Операция ATOMIC TEST-AND-SWAP заканчивается после того, как ответные данные приняты в LSU, проверены на равенство с нулем и установлен соответствующий код в регистре LSUn\_CR6. В память микропроцессора эти данные не передаются.

Для пакетов типа 8 принятое 32-разрядное слово данных размещается в памяти с учетом 3 разряда ADDR/CONFIG.

После выполнения операции бит BUSY сбрасывается в «0». Состояние о выполнении операции содержится в поле OP\_STATUS регистра LSUn\_CR6.

После окончания выполнения операции ввода-вывода может быть сформировано прерывание, если в регистре LSUn\_CR4 бит INT\_MASK=1.

#### **11.4.3.2 Выполнение входящих операций**

##### 11.4.3.2.1 Прием входных пакетов запросов

Входные пакеты запроса поступают из буфера PL\_RXB. Прием пакета начинается с запоминания следующих полей его заголовка: ftype, transaction, wrsize/rdsize/status, wdptr, srcTID, sourceID, destinationID, address/config-offset, xamsbs. После этого заголовок анализируется. Проверяется возможность его обработки.

Прием пакетов выполняется в соответствии с содержимым регистра DEST\_OP\_CAR. Если прием этого пакета не разрешен, то пакет далее не обрабатывается, и имеющиеся в нем данные выбрасываются. Ответный пакет выдается со статусом ERROR.

Если прием этого типа пакета разрешен, то содержимое поля destination ID сравнивается с содержимым регистра BASE\_DEVICE\_ID\_CSR с учетом поля пакета tt. Если они не равны, то пакет далее не обрабатывается, а имеющиеся в нем данные выбрасываются.

Если сравнение произошло, то операция анализируется на допустимость и реализуемость.

Если операция является недопустимой, то пакет далее не обрабатывается, и имеющиеся в нем данные выбрасываются. Ответный пакет не выдается.

Если операция является нереализованной, то пакет далее не обрабатывается, и имеющиеся в нем данные выбрасываются. Если по этой операции требуется ответный пакет, то он выдается с признаком ERROR в поле status и без поля данных.

Далее анализируется поле address. Если по данному адресу доступ запрещен (см. регистры IN\_FLTR в п. 11.3.3.8), то пакет далее не обрабатывается, и имеющиеся в нем данные выбрасываются. Если по этой операции требуется ответный пакет, то он выдается с признаком ERROR в поле status и без поля данных.



Данные, при их наличии, из буфера PL\_RXB передаются в буфер FIFO\_RX для формирования 64-разрядных слов. Сформированные 64-разрядные слова при помощи канала DMA записываются в память данного микропроцессора, начиная с адреса, указанного в поле пакета address.

Если длина поля данных не соответствует формату пакета, то пакет далее не обрабатывается, и имеющиеся в нем данные выбрасываются. Если по этой операции требуется ответный пакет, то он выдается с признаком ERROR в поле status и без поля данных.

Если пакет имеет формат без данных, а они присутствуют в нем, то пакет далее не обрабатывается, а имеющиеся в нем данные выбрасываются. Если по этой операции требуется ответный пакет, то он выдается с признаком ERROR в поле status и без поля данных.

Лишние данные, присутствующие в пакете выбрасываются.

#### 11.4.3.2.2 Выполнение обмена данными с памятью микропроцессора

Если ошибок в формате пакета нет, то далее выполняется обмен данными с памятью данного микропроцессора. Обмен выполняется при помощи канала DMA. Обмен данными по DMA с памятью микропроцессора выполняется пачками по WN слов (см. регистр CSR\_SRIO).

Следует иметь в виду, что при приеме данных их объем может быть меньше, чем указано в полях wsize, wdptr.

Если принят пакет запроса NWRITE, NWRITE\_R или SWRITE, то данные, запомненные в буфере FIFO\_RX, записываются в память, начиная с адреса, указанного в поле пакета address.

Если принят пакет запроса NREAD, то данные из памяти, начиная с адреса, указанного в поле пакета address, считываются в буфер FIFO\_TX. После окончания считывания всех данных, пакет передается в буфер PL\_TXB.

Если принят пакет запроса ATOMIC TEST-AND-SWAP то, из памяти микропроцессора по адресу, указанному в поле пакета address, считываются одно 64-разрядное слово. Из него выделяется 32-разрядное слово, с учетом бита **wdptr** пакета. Если оно равно нулю, то вместо него вставляется 32-разрядное слово из принятого пакета, и результирующее 64-разрядное слово записывается в память по этому же адресу.

Если принят пакет запроса ATOMIC POST-INCREMENTP-THE-DATA, из памяти микропроцессора по адресу, указанному в поле пакета address, считывается одно 64-разрядное слово. Из него выделяется 32-разрядное слово, с учетом бита **wdptr** пакета, и к нему прибавляется 1, а полученный результат записывается в память по этому же адресу.

Если принят пакет запроса ATOMIC POST-DECREMENT-THE-DATA, из памяти микропроцессора по адресу, указанному в поле пакета address, считывается одно 64-разрядное слово. Из него выделяется 32-разрядное слово, с учетом бита **wdptr** пакета, и из него вычитается 1, а полученный результат записывается в память по этому же адресу.

Если принят пакет запроса ATOMIC SET-THE-DATA, из памяти микропроцессора по адресу, указанному в поле пакета address, считывается одно 64-разрядное слово. Из него выделяется 32-разрядное слово, с учетом бита **wdptr** пакета, и в нем устанавливаются все 1. Результирующее 64-разрядное слово записывается в память по этому же адресу.

Если принят пакет запроса ATOMIC CLEAR-THE-DATA, из памяти микропроцессора по адресу, указанному в поле пакета address, считывается одно 32-разрядное слово. Затем в память по этому же адресу записывается все 0.

#### 11.4.3.2.3 Выполнение операций типа MAINTENANCE

При выполнении операций типа MAINTENANCE осуществляется обмен данными с регистрами SRIO. Относительные адреса этих регистров указаны в Таблица 11.1. Длина поля данных этих пакетов ограничена одним 32-разрядным словом.

Если принят пакет запроса MAINTENANCE WRITE REQUEST, то данные из буфера PL\_RXB записываются в регистр SRIO по адресу, указанному в поле **config-offset** с учетом бита **wdptr**.

Если принят пакет запроса MAINTENANCE READ REQUEST, то содержимое регистра SRIO по адресу, указанному в поле **config-offset** с учетом бита **wdptr**, передается в буфер PL\_TXB вместе с соответствующим заголовком.

#### 11.4.3.2.4 Выполнение операции MAINTENANCE PORT-WRITE REQUEST

Операции MAINTENANCE PORT-WRITE REQUEST не имеют гарантированной доставки и не имеют ответных пакетов. Эти операции полезны для передачи информации об ошибках или статусной информации от коммутаторов.

В SRIO имеется буфер для приема одного пакета MAINTENANCE PORT-WRITE REQUEST. Он состоит из регистра PORT\_WRITE\_CSR и FIFO объемом 16 32-разрядных слов (64 байта). Формат регистра описан в п. 11.3. Поля регистра PORT\_WRITE приведены в Таблица 11.83.

Таблица 11.83 Поля регистра PORT\_WRTITE

Условное обозначение	Назначение
SOURCE_ID_PW[15:0]	Поле пакета sourceID
NEMPTY_PW	Признак наличия данных в FIFO
TT_PW[1:0]	Поле TT пакета, определяет размерность sourceID
SIZE_PW[5:0]	Размер поля данных пакета в 32-разрядных словах
SRC_TID_PW[7:0]	Поле пакета srcTID

После приема пакета в FIFO формируется прерывание. Оно сбрасывается после считывания всех данных из FIFO. Если в момент занятости FIFO поступит очередной пакет MAINTENANCE PORT-WRITE REQUEST, то он выбрасывается.

Чтение данных из FIFO осуществляется программно через 32-разрядный регистр PORT\_WRITE\_FIFO. Этот регистр доступен только по чтению. Исходное состояние регистра не определено.

После каждого чтения данных через регистр PORT\_WRITE\_FIFO, значение поля SIZE\_PW[5:0] декрементируется и отображает реальное количество данных оставшихся в FIFO.

Если в принятом пакете Port-Write данных больше чем указано в полях wrtsize, wdptr, то лишние данные выкидываются, а очередь PORT-WRITE FIFO сбрасывается.



#### 11.4.3.2.5 Формирование ответных пакетов

Поля ответного пакета формируются следующим образом:

- \* `prio` – равно аналогичному полю пакета запроса;
- \* `tt` – равно аналогичному полю пакета запроса;
- \* `ftype`, `transaction` – по типу ответного пакета (см. Таблица 11.77 - Таблица 11.81);
- \* в поле `status` – формируется код `ERROR`, если обнаружена ошибка в формате входного пакета запроса, при недопустимом коде транзакции или если эта транзакция не реализована;
- \* в поле `status` – формируется код `ERROR`, если доступ к памяти микропроцессора по этому адресу в настоящий момент запрещен;
- \* `targetTID` берется из поля `srcTID` принятого пакета запроса;
- \* `sourceID` берется из регистра `BASE_DEVICE_ID_CSR`;
- \* `destinationID` берется из поля `sourceID` принятого пакета запроса.

Если в поле `status` ответного пакета сформирован код `ERROR`, то пакет выдается без данных.

Ответный пакет не выдается, если операция, заданная входным пакетом запроса является недопустимой, т.е. контроллер не может определить тип пакета по полям `ftype` и `transaction`. В этой ситуации не известно, надо ли выдавать ответный пакет.

## 11.5 Устройство выполнения операций передачи сообщений (MPU)

### 11.5.1 Общие положения

Устройство выполнения операций передачи сообщений (MPU – Message Passing Unit) в соответствии с требованиями RapidIO Interconnect Specification V1.3 Part II: Message Passing Logical Specification.

При передаче сообщений адрес внутри устройства RapidIO (address) в пакете запроса не указывается. Вместо этого используется идентификатор почтового ящика (mailbox). Почтовый ящик отображается на память самим устройством RapidIO, которое приняло это сообщение.

Стандарт RapidIO регламентирует в устройстве 4 почтовых ящиков. Каждый почтовый ящик может содержать 4 письма. Однопакетные сообщения обеспечивают организацию 64 почтовых ящиков по 4 письма в каждом, то есть 256 сообщений. Почтовые ящики могут быть определены для различных типов данных или приоритетов.

### 11.5.2 Описание операций передачи сообщений

Перечень выполняемых операций обмена сообщениями приведен в Таблица 11.84.

Таблица 11.84. Перечень выполняемых операций ввода-вывода

Вид операции	Тип пакета запроса	Тип пакета ответа	Описание
DOORBELL	DOORBELL	RESPONSE	Передача короткого сообщения с 16-разрядным полем info.
DATA MESSAGE	MESSAGE	RESPONSE	Передача от 8 до 256 байт данных в одном пакете. Объем данных кратен 8 байтам. Сообщение может содержать до 16 пакетов.

Описание полей пакета MESSAGE приведено в Таблица 11.85.

Таблица 11.85. Описание полей пакета MESSAGE

Поле	Число бит	Описание
msglen	4	Число пакетов в сообщении: 0 – 1 пакет; 1 – 2 пакета; ... 15 – 16 пакетов.
ssize	4	Объем данных во всех пакетах много пакетного сообщения, за исключением последнего пакета (он может быть меньше): 0000:1000 – резерв; 1001 – 1 64-разрядное слово; 1010 – 2 64-разрядных слов; 1011 – 4 64-разрядных слов; 1100 – 8 64-разрядных слов; 1101 – 16 64-разрядных слов; 1110 – 32 64-разрядных слов; 1111 – резерв.
letter	2	Номер слота в почтовом ящике.
mbox	2	Номер почтового ящика.
msgseg	4	Номер пакета в сообщении: 0 – 1 пакет; 1 – 2 пакет; ... 15 – 16 пакет.

Поле	Число бит	Описание
xmbox	4	Для сообщений, состоящих из одного пакета, старшие 4 разряда номера почтового ящика. То есть, номер почтового ящика определяется конкатенацией {xmbox, mbox}- всего 64 почтовые ящика.

Описание полей ответного пакета приведено в Таблица 11.86.

**Таблица 11.86. Описание полей ответного пакета**

Поле	Число бит	Описание
transaction	4	Тип ответного пакета: 0000 – ответный пакет DOORBELL; 0001 – ответный пакет MESSAGE.
status	4	Результат выполнения транзакции: 0000 – DONE, транзакция завершилась успешно; 0001:0010 – резерв; 0011 – RETRY, транзакция не выполнена, требуется повтор; 0100:0110 – резерв; 0111 – ERROR, обнаружена не восстанавливаемая ошибка; 1000:1111 – резерв.
targetTID		Используется в ответном пакете DOORBELL. Содержимое поля равно содержимому поля srcTID входного пакета DOORBELL.
letter	2	Номер слота принятого пакета.
mbox	2	Номер почтового ящика принятого пакета.
msgseg	4	Номер пакета в сообщении.

Формат пакета MESSAGE приведен в Таблица 11.87.

**Таблица 11.87. Формат пакета MESSAGE**

Номер поля	Название поля	Длина поля, бит
1	prio	2
2	tt	2
3	ftype = 1011	4
4	destinationID	8/16
5	sourceID	8/16
6	msglen	4
7	ssize	4
8	letter	2
9	mbox	2
10	msgseg/xmbox	4
11	double-word 0	64
...	...	
N	double-word 0	64

Формат ответного пакета MESSAGE приведен в Таблица 11.88.

Таблица 11.88. Формат ответного пакета MESSAGE

Номер поля	Название поля	Длина поля, бит	
1	prio	2	
2	tt	2	
3	ftype = 1101	4	
4	destinationID	8/16	
5	sourceID	8/16	
6	transaction = 0001	4	
7	status	4	
8	target_info	letter	2
9		mbox	2
10		msgseg	4

### 11.5.3 Прием сообщений

#### 11.5.3.1 Описание дескрипторов приема сообщений

Пакеты обрабатываются в порядке их поступления.

Для приема входных пакетов MESSAGE в памяти микропроцессора организуются очереди. Отображение этих пакетов в соответствующую очередь осуществляется посредством сравнения содержимого полей пакета sourceID, msglen, mbox, letter и mbox с содержимым всех регистров RXU\_MAP\_Ln RXU\_MAP\_Hn. Всего имеется 32 пары этих регистров. Формат регистров RXU\_MAP\_Ln и RXU\_MAP\_Hn описан в п. 11.3. Поля этих регистров приведены в Таблица 11.89.

Таблица 11.89. Описание полей регистров RXU\_MAP\_Ln и RXU\_MAP\_Hn

Поле	Описание
LETTER_MASK[1:0]	Маска номера письма: 0 – соответствующий бит не участвует в сравнении; 1 - соответствующий бит участвует в сравнении.
MAILBOX_MASK[5:0]	Маска номера почтового ящика: 0 – соответствующий бит не участвует в сравнении; 1 - соответствующий бит участвует в сравнении.
LETTER[1:0]	Номер письма
MAILBOX[5:0]	Номер почтового ящика
SOURCEID[15:0]	Идентификатор источника входного пакета
TT[1:0]	Длина поля sourceID: 0 – 8 разрядов; 1 – 16 разрядов.
QUEUE_ID[3:0]	Номер очереди – от 0 до 15
PROMISCUOUS	Разрешение не сравнивать содержимое поля пакета sourceID и поля SOURCEID регистра RXU_MAP_Ln: 0 – сравнение выполняется; 1 – сравнение не выполняется.
SEGMENT_MAPPING	Режим сегментации: 0 – однопакетное сообщение; 1 – многопакетное сообщение.

Если для принимаемого сообщения есть несколько отображений, то номер очереди берётся из регистров RXU\_MAP\_Ln и RXU\_MAP\_Hn с наименьшим номером.

Если не произошло ни одного совпадения, то пакет выбрасывается и ответный пакет выдается со статусом ERROR.

32 пары регистров RXU\_MAP\_Ln и RXU\_MAP\_Hn обеспечивают отображение на 16 очередей. Каждая очередь может быть использована для приема одно пакетных или много пакетных сообщений. Для каждой из этих очередей имеется два 32-разрядных регистра:

- \* Регистр указателя на первый дескриптор входной очереди (RXQ\_HDPn – Receive Queue Head Descriptor Pointer). Для инициализации данной очереди в него необходимо записать адрес первого дескриптора очереди. Младшие 3 разряда этого адреса должны быть нулевыми. После исчерпания очереди, то есть после обработки последнего дескриптора, он аппаратно обнуляется. Если в него произведена запись, а он не равен нулю, то результат приема пакетов будет неопределенным;
- \* Регистр указателя обработанного дескриптора (RXQ\_CDPn – Receive Queue Completion Descriptor Pointer). Младшие 3 разряда этого указателя должны быть нулевыми. После приема очередного сообщения в этот регистр аппаратно записывается адрес дескриптора данного сообщения и формируется прерывание. После обработки прерывания по данному дескриптору в этот регистр программно необходимо записать его адрес. Если он совпал с содержимым этого регистра, (то есть это последний обработанный дескриптор) то соответствующее прерывание сбрасывается. После этого, буфер, определяемый этим дескриптором, может быть вновь использован SRIO.

Дескрипторы очереди состоят из 4 32-разрядных слов, последовательно расположенных в памяти. Дескриптор имеет следующий состав (приведен в порядке расположения в памяти):

- \* RX\_NDP (Next Descriptor Pointer) – содержит указатель (адрес) на следующий дескриптор. Младшие 3 разряда этого адреса должны быть нулевыми. Если RX\_NDP=0, то это последний дескриптор;
- \* RX\_BP (Buffer Pointer) – содержит адрес начала буфера. Младшие 3 разряда этого адреса должны быть нулевыми.

Форматы 3-го и 4-го слова дескриптора приведены в Таблица 11.90 и Таблица 11.91.

**Таблица 11.90. Формат 3-го слова дескриптора**

Номер разряда	Условное обозначение	Назначение
31:16	SOURCE_ID	Содержит поле sourceID принятого сообщения. Записывается аппаратно после приема первого пакета сообщения.
15:14	PRIО	Содержит поле prі принятого пакета. Записывается аппаратно после приема первого пакета сообщения.
13:12	TT	Содержит поле tt принятого пакета. Записывается аппаратно после приема первого пакета сообщения.

Номер разряда	Условное обозначение	Назначение
11:8	SSIZE	Содержит поле ssize принятого сообщения. Означает объем данных во всех пакетах много пакетного сообщения, за исключением последнего пакета (он может быть меньше): 0000:1000 – резерв; 1001 – 1 64-разрядное слово; 1010 – 2 64-разрядных слов; 1011 – 4 64-разрядных слов; 1100 – 8 64-разрядных слов; 1101 – 16 64-разрядных слов; 1110 – 32 64-разрядных слов; 1111 – резерв. Записывается аппаратно после приема первого пакета сообщения.
7:6	LETTER	Содержит поле letter принятого сообщения. Записывается аппаратно после приема первого пакета сообщения.
5:0	MAILBOX	Содержит конкатенацию полей {xmbbox, mbox} принятого сообщения. Для многопакетного сообщения действительны только два младших разряда. Записывается аппаратно после приема первого пакета сообщения.

**Таблица 11.91. Формат 4-го слова дескриптора**

Номер разряда	Условное обозначение	Назначение
31:30	-	Не используется
29	OWNERSHIP	Признак владения дескриптором. Устанавливается в 1 программно при инициализации дескриптора. Сбрасывается в 0 аппаратно после приема всего сообщения в память.
28	-	Не используется
27	TEARDOWN_COPMLETE	Признак того, что процедура прекращения приема сообщений, завершена: 0 – процедура не окончена; 1 – процедура закончена. Устанавливается в 0 программно при инициализации дескриптора. Устанавливается в 1 аппаратно при окончании процедуры.
26	RESP_CANCEL	Ответный пакет не выдан по причине занятости буфера PL_TXB. При этом сообщение может быть принято без ошибок и поле CC = 000.
25:20	-	Не используется
19:16	MSGLEN	Поле msglen принятого сообщения. Означает длину сообщения в пакетах: 0000 – сообщение состоит из 1-ого пакета; 0001 – сообщение состоит из 2-ух пакетов; 0010 – сообщение состоит из 3-ех пакетов; ..... 1110 – сообщение состоит из 15-ти пакетов; 1111 – сообщение состоит из 16-ти пакетов.
15:13	-	Не используется
12	INT_MASK	Маска формирования прерывания после приема сообщения: 0 – прерывание запрещено; 1 – прерывание разрешено.

Номер разряда	Условное обозначение	Назначение
11:9	СС	Признак завершения приема сообщения: 000 – сообщение успешно принято; 001 – ошибка. Длина сообщения больше чем это обеспечивается дескриптором. То есть, больше чем указано в поле MSG_LENGTH; 010 – длина не последнего пакета многопакетного сообщения не равна содержимому поля <b>ssize</b> , для последнего пакета – его длина превышает <b>ssize</b> ; 011 – ошибка. Зафиксирован таймаут при приеме пакета в случае многопакетного сообщения; 100 – ошибка. Нарушен порядок следования номеров пакетов при приеме многопакетного сообщения. 101 – прием сообщения прекращен программно. Данные не действительны; 110:111 – не используется.
8:0	MSG_LENGTH	Длина сообщения: 000 – 512 64-разрядных слов; 001 – 1 64-разрядное слово; 002 – 2 64-разрядных слова; ... 1FF – 511 64-разрядных слов. Устанавливается программно при инициализации дескриптора и указывает максимальную длину сообщения, которое может быть принято. После приема всего сообщения аппаратно устанавливается действительная длина принятого сообщения.

### 11.5.3.2 Порядок приема сообщений

Дескриптор используется для одного сообщения. В данный момент времени очередь для многопакетных сообщений может обеспечивать прием только одного сообщения.

Если дескриптор для много пакетного сообщения в данный момент не свободен, (то есть, поступили еще не все пакеты сообщения) и начало поступать другое много пакетное сообщение (с другими полями sourceID или mailbox или letter) в эту же очередь, то пакеты этого сообщения выбрасываются и выдаются ответные пакеты со статусом RETRY.

Если дескриптор для много пакетного сообщения в данный момент не свободен, и начало поступать следующее много пакетное сообщение с теми же полями sourceID, mailbox и letter, то последний пакет выбрасывается и выдается ответный пакет со статусом ERROR. Это обычно является признаком того, что на передающей стороне допущена ошибка, то есть пакет передан повторно или пакет передан с выходом за диапазон msglen.

Однопакетные сообщения никогда не вызывают передачу ответного пакета со статусом RETRY.

После успешного приема сообщения выдается ответный пакет со статусом DONE.

Если длина принимаемого сообщения больше чем указано в дескрипторе, то выдается ответный пакет со статусом ERROR. Одновременно устанавливается соответствующий код в поле CC 4-го слова дескриптора.

При приеме пакетов каждого много пакетного сообщения выполняется таймирование. Таймер реализован аналогично тому, как это описано в п. 11.4.3.1.3. Таймер запускается при выдаче ответного пакета на предыдущий пакет и сбрасывается при приеме следующего пакета данного сообщения.

При приеме всего сообщения признак OWNERSHIP обнуляется аппаратно и формируется прерывание. По этому прерыванию CPU обрабатывает очередь, обнаруживая принятые пакеты, анализируя признаки OWNERSHIP в каждом дескрипторе. Обработка выполняется до обнаружения OWNERSHIP=1 или EOQ=1.

В каждой очереди порядок приема пакетов много пакетных сообщений управляется регистром RX\_CR (Receive Control Register). Формат регистра приведен в п. 11.3. Пакеты в пределах сообщения могут приниматься в любом порядке или только в порядке возрастания номера пакета (поле msgseg). Последний вариант используется для применений, со специальными информационными потоками (flows). В этом случае, если номер принятого пакет не соответствует порядку, то выдается ответный пакет со статусом RETRY. То же происходит и с последующими пакетами, пока не будет принят ожидаемый пакет.

### 11.5.3.3 Программная деактивизация очереди принимаемых пакетов

Можно программно прекратить прием пакетов данной очереди (TEARDOWN), посредством записи в регистр RX\_QTCR. Если эта команда поступила в SRIO, когда его автомат приема сообщений находится в неактивном состоянии, то аппаратно выполняются следующие действия:



- \* если очередь находится в ожидании очередного пакета много пакетного сообщения, то прием новых пакетов прекращается и в текущем дескрипторе устанавливается  $CC=100$ . Все остальные поля дескриптора являются не действительными. Устанавливается:  $RXQ\_HDPn=0$ ,  $RXQ\_CDPn=FFFF\_FFFC$ , формируется прерывание по данной очереди и соответствующий бит  $QUEUEn\_TEARDWN$  в регистре  $RX\_QTCR$  обнуляется;
- \* если очередь не находится в ожидании очередного пакета много пакетного сообщения, но является активной (есть не обработанные дескрипторы), то в очередном дескрипторе устанавливается  $CC=100$ . Все остальные поля дескриптора являются не действительными. Устанавливается:  $RXQ\_HDPn=0$ ,  $RXQ\_CDPn=FFFF\_FFFC$ , формируется прерывание по данной очереди и соответствующий бит  $QUEUEn\_TEARDWN$  в регистре  $RX\_QTCR$  обнуляется;
- \* если очередь не находится в ожидании очередного пакета много пакетного сообщения и не является активной то содержимое регистров  $RXQ\_HDPn=0$  и  $RXQ\_CDPn$  не изменяется. Прерывания не формируется. Соответствующий бит  $QUEUEn\_TEARDWN$  в регистре  $RX\_QTCR$  не обнуляется.

Если эта команда поступила в SRIO, когда его автомат приема сообщений находится в активном состоянии, то ожидается переход его в неактивное состояние.

После окончания процесса TEARDOWN программное обеспечение должно инициализировать очередь снова.

#### 11.5.3.4 Прием пакетов запроса DOORBELL

Для приема пакетов DOORBELL в SRIO имеется буфер типа FIFO объемом 16 слов, что позволяет принять до 16 пакетов. Чтение данных из FIFO осуществляется через два регистра DOORBELL\_FIFO\_LOW и DOORBELL\_FIFO\_HIGH. Поля этих регистров приведены в Таблица 11.92.

Таблица 11.92 Поля регистров DOORBELL\_FIFO\_L, DOORBELL\_FIFO\_H

Условное обозначение	Назначение
NEMPTY	Признак наличия пакетов DOORBELL в FIFO
-	Не используется
TT[1:0]	Содержимое поля пакета tt
PRIO[1:0]	Содержимое поля пакета prio
SRC_TID[7:0]	Содержимое поля пакета srcTID
SOURCE_ID[15:0]	Содержимое поля пакета sourceID
INFO[15:0]	Содержимое поля пакета info

После приема пакета в FIFO формируется прерывание. Оно сбрасывается после считывания всех данных из FIFO. Если в момент занятости FIFO поступит очередной пакет DOORBELL, то выдается ответный пакет со статусом RETRY.

## 11.5.4 Передача сообщений

### 11.5.4.1 Описание дескрипторов передачи сообщений

Передача сообщений осуществляется аналогично приему сообщений с использованием дескрипторов. Каждый дескриптор предназначен для передачи одного сообщения, однопакетного или многопакетного. Дескрипторы должны быть инициализированы программно.

Имеется 16 очередей дескрипторов. Для каждой из этих очередей имеется два 32-разрядных регистра:

- \* Регистр указателя на первый дескриптор входной очереди (TXQ\_HDPn – Transmit Queue Head Descriptor Pointer). Для инициализации данной очереди в него необходимо записать адрес первого дескриптора очереди. Младшие 3 разряда этого адреса должны быть нулевыми. После исчерпания очереди, то есть после обработки последнего дескриптора, он аппаратно обнуляется. Если в него произведена запись, а он не равен нулю, то результат передачи пакетов будет неопределенным;
- \* Регистр указателя обработанного дескриптора (TXQ\_CDPn – Transmit Queue Completion Descriptor Pointer). Младшие 3 разряда этого указателя должны быть нулевыми. После передачи очередного сообщения в этот регистр аппаратно записывается адрес дескриптора данного сообщения и формируется прерывание. После обработки прерывания по данному дескриптору в этот регистр программно необходимо записать его адрес. Если он совпал с содержимым этого регистра, (то есть это последний обработанный дескриптор) то соответствующее прерывание сбрасывается. После этого, буфер, определяемый этим дескриптором, может быть вновь использован SRIO.

Дескрипторы очереди состоят из 4 32-разрядных слов, последовательно расположенных в памяти. Дескриптор имеет следующий состав (приведен в порядке расположения в памяти):

- \* TX\_NDP (Next Descriptor Pointer) – содержит указатель (адрес) на следующий дескриптор. Младшие 3 разряда этого адреса должны быть нулевыми. Если NDP = 0, то это последний дескриптор;
- \* TX\_BP (Buffer Pointer) – содержит адрес начала буфера данных. Младшие 3 разряда этого адреса должны быть нулевыми;

Форматы 3-го и 4-го слова дескриптора приведены в Таблица 11.93, Таблица 11.94.

**Таблица 11.93. Формат 3-го слова дескриптора**

Номер разряда	Условное обозначение	Назначение
31:16	DESTINATION_ID	Содержит поле destinationID для пакетов передаваемого сообщения. Определяется программно при инициализации дескриптора.
15:14	PRIO	Содержит поле prі для пакетов передаваемого сообщения. Определяется программно при инициализации дескриптора. Не может иметь значение 3.
13:12	TT	Содержит поле tt для пакетов передаваемого сообщения. Определяется программно при инициализации дескриптора.
11:8	SSIZE	<p>Объем данных во всех пакетах много пакетного сообщения, за исключением последнего пакета (он может быть меньше):</p> <p>0000:1000 – резерв;                      1001 – 1 64-разрядное слово;                      1010 – 2 64-разрядных слов;                      1011 – 4 64-разрядных слов;                      1100 – 8 64-разрядных слов;                      1101 – 16 64-разрядных слов;                      1110 – 32 64-разрядных слов;                      1111 – резерв.</p> <p>Содержимое поля MSG_LENGTH 4-го слова дескриптора деленное на 16 должно быть меньше или равно содержимому этого поля. Определяется программно при инициализации дескриптора.</p>
7:6	LETTER	Содержит поле letter для пакетов передаваемого сообщения. Определяется программно при инициализации дескриптора.
5:0	MAILBOX	Содержит конкатенацию полей {xmbx, mbox} для пакетов передаваемого сообщения. Для многопакетного сообщения действительны только два младших разряда. Определяется программно при инициализации дескриптора.

Таблица 11.94. Формат 4-го слова дескриптора

Номер разряда	Условное обозначение	Назначение
31:30	-	Не используется
29	OWNERSHIP	Признак владения дескриптором. Устанавливается в 1 программно при инициализации дескриптора. Сбрасывается в 0 аппаратно после передачи всего сообщения. При этом в поле CC устанавливается соответствующий код.
28	-	Не используется
27	TEARDOWN_COPMLETE	Признак того, что процедура прекращения передачи сообщений, завершена: 0 – процедура не окончена; 1 – процедура закончена. Устанавливается в 0 программно при инициализации дескриптора. Устанавливается в 1 аппаратно при окончании процедуры.
26:23	-	Не используется
22:16	RETRY_COUNT	Число повторных передач пакетов данного сообщения (общее число на все пакеты): 0000000 – без повторов; 0000001 – один повтор; 0000010 – два повтора; ... 0111111 – 63 повтора; 1000000 – неограниченное число; Определяется программно при инициализации дескриптора.
15:13	-	Не используется
12	INT_MASK	Маска формирования прерывания после передачи сообщения: 0 – прерывание запрещено; 1 – прерывание разрешено.
11:9	CC	Признак завершения передачи сообщения: 000 – сообщение успешно передано. Ответные пакеты приняты со статусом DONE; 001 – ошибка. Принят ответный пакет со статусом ERROR. 010 – потребовалось большее число повторов, чем указано в поле RETRY_COUNT. 011 – возник таймаут при передаче пакета; 100 – ошибка программирования дескриптора; 101 – передача сообщения прекращена программно (teardown); 110 – в буфере PL_TXB нет свободного места; 111 – не используется. Устанавливается в 0 программно при инициализации дескриптора.
8:0	MSG_LENGTH	Длина сообщения: 000 – 512 64-разрядных слов; 001 – 1 64-разрядное слово; 002 – 2 64-разрядных слова; ... 1FF – 511 64-разрядных слов. Устанавливается программно при инициализации дескриптора и указывает длину сообщения для передачи.

#### 11.5.4.2 Порядок передачи сообщений

Порядок передачи очередей сообщений изменяется циклически с использованием весовых коэффициентов. Для управления этим механизмом имеется 4 32-разрядных регистра TX\_QUEUE\_CTR0, TX\_QUEUE\_CTR1, TX\_QUEUE\_CTR2, TX\_QUEUE\_CTR3, формат которых описан в п. 11.3. Поля этих регистров приведены в Таблица 11.95. Описание полей регистров TX\_QUEUE\_CTR0, TX\_QUEUE\_CTR1, TX\_QUEUE\_CTR2, TX\_QUEUE\_CTR3

**Таблица 11.95. Описание полей регистров TX\_QUEUE\_CTR0, TX\_QUEUE\_CTR1, TX\_QUEUE\_CTR2, TX\_QUEUE\_CTR3**

Условное обозначение	Назначение
MSG_NMBRn[3:0]	Количество сообщений (дескрипторов) очереди n, которые передаются перед переходом к другой очереди: 0 – одно сообщение; ... F – 16 сообщений.
POINTERn[3:0]	Указатель номера очереди, на которую выполняется переход от очереди n.

После инициализации дескриптора анализируются поля DESTINATION\_ID и PRI дескриптора для определения того, заблокировано ли (Xoffd) это Flow. Если оно заблокировано, то эта очередь пропускается.

Действия, связанные с поступлением ответных пакетов и их таймированием имеют самый высокий приоритет.

Если получен ответный пакет со статусом RETRY, то повторная передача соответствующего пакета выполняется немедленно, то есть является более приоритетной, чем передача новых пакетов. Новые пакеты передаются, не ожидая ответных пакетов на предыдущие пакеты.

Ответные пакеты могут поступать не в том порядке, в котором соответствующие пакеты выданы. Тем не менее, действия по обновлению содержимого дескрипторов и формированию прерываний выполняются только после поступления ответов на все предыдущие пакеты.

Передача многопакетного сообщения прекращается, как только текущий пакет передан с ошибкой (таймаут или принят ответный пакет со статусом ERROR). Передача многопакетного сообщения считается законченной, если приняты все ответные пакеты.

#### 11.5.4.3 Программная деактивизация очереди передаваемых пакетов

Можно программно прекратить передавать пакеты очереди (TEARDOWN), посредством записи в регистр TX\_QTCR. В результате этого аппаратно выполняются следующие действия:

- \* Передача новых сообщений прекращается;
- \* все начатые сообщения передаются как обычно;
- \* если очередь была активной, то в следующем дескрипторе устанавливается CC=101. Устанавливается: TXQ\_HDPn=0, TXQ\_CDPn=FFFF\_FFFC, формируется прерывание по данной очереди и соответствующий бит QUEUE\_n\_TEARDWN в регистре TX\_QTCR обнуляется;
- \* если очередь была неактивной (нет больше дескрипторов), или она становится неактивной после окончания передачи текущего сообщения, то обнуляется только соответствующий бит QUEUE\_n\_TEARDWN в регистре TX\_QTCR.

После окончания процесса TEARDOWN программное обеспечение должно инициализировать очередь снова.

#### 11.5.4.4 Функции программного обеспечения при передаче сообщений

При инициализации очереди:

- \* Сформировать в памяти очереди дескрипторов;
- \* В регистрах TX\_QUEUE\_CTRn сформировать таблицу переходов между очередями передачи;
- \* Записать в регистры TX\_HDPn адреса начальных дескрипторов очередей. После записи, очереди передаются согласно таблицы переходов.

При обработке прерываний:

- \* проверяется состояние битов OWNERSHIP, если он нулевой, то сообщение передано;
- \* очередь обрабатывается до обнаружения EOQ=1 или OWNERSHIP=1;
- \* то, что все сообщения в данной очереди переданы, определяется по EOQ=1, OWNERSHIP=0, TX\_NDP=0 в последнем обработанном дескрипторе;
- \* подтверждение обработки прерывания выполняется посредством записи в регистр TXQ\_CDPn адрес дескриптора, по которому обработано прерывание. При этом прерывание, выставленное SRIO, сбрасывается.

## 11.6 Формирование и обработка прерываний

Перечень сигналов прерываний, которые формирует SRIO приведен в Таблица 11.96.

Таблица 11.96 Сигналы прерываний SRIO

Условное обозначение	Назначение
RESET_DEVICE_CMD	Поступили 4 команды Reset-Device Command. Сбрасывается посредством записи 1 в одноименный бит регистра LPU_CSR.
PORT_ERROR	Признак того, что LPU находится в нерабочем состоянии из-за обнаружения невозстанавливаемой ошибки. Сбрасывается посредством записи 1 в одноименный бит регистра ERROR_STATUS_CSR
MCE_DEC	Признак того, что LPU принял символ Multicast-Event. Сбрасывается посредством записи 1 в одноименный бит регистра LPU_CSR
MPU_INT	Прерывание от MPU
LSU_INT	Прерывание от LSU
DBL_INT	Признак наличия прерывания от порта приема пакетов DOORBELL. Повторяет состояние разряда NEMPTY регистра DBL_FIFO_LOW.
MNT_INT	Признак наличия прерывания MAINTENANCE PORT-WRITE либо MAINTENANCE WRITE.

Сигналы прерывания SRIO поступают с разрядов регистра SRIO\_CSR.

Для идентификации прерываний от LSU имеется регистр LSU\_IRQ. Перечень прерываний от каждого LSU приведен в Таблица 11.97.

Таблица 11.97 Перечень прерываний, формируемых LSU

Условное обозначение	Назначение
OKEY	Операция выполнена без ошибок (Posted/Non-Posted)
ERROR	При выполнении операции получен ответный пакет (типа 8 или 13), содержащий в поле «status» код «ERROR» или его поле данных имеет неправильную длину
DRBL_RETRY	В ответ на пакет DOORBELL принят пакет, содержащий в поле «status» код «RETRY».
UNSUPPORTED	Операция не может быть выполнена, так как она не реализована, или регистры LSU неправильно запрограммированы
TIMEOUT	При выполнении Non-Posted операции (операции, в которых требуется ответный пакет) произошел таймаут
DRBL_ERROR	В ответ на пакет DOORBELL принят пакет, содержащий в поле «status» код «ERROR»
NOT_ALLOWED	Операция «ATOMIC test-and-swap» не может быть выполнена из-за занятости семафора
NO_CREDIT	Пакет не может быть передан из-за занятости буфера PL_TXB в момент программного запуска операции или при срабатывании таймера в случае выполнения многопакетной операции.

Для обнуления разрядов регистра LSU\_IRQ\_SR имеется 32-разрядный регистр LSU\_IRQ\_CLR (LSU Interrupt Request Clear Register). Его формат полностью повторяет формат регистра LSU\_IRQ\_SR. Он доступен только по записи 1. При этом соответствующий разряд регистра LSU\_IRQ\_SR обнуляется.

Для идентификации очереди, от которой сформировано прерывание, имеется регистр MPU\_IRQ (MPU Interrupt Request Register). Он содержит два 16-разрядных поля TX\_QUEUE и RX\_QUEUE.

Обнуление условия соответствующего прерывания от MPU осуществляется посредством записи в регистр RXQ\_CDPn или TXQ\_CDPn адреса обработанного дескриптора (см. п 11.5.3.1).



## 11.7 Инициализация SRIO

### 11.7.1 Запуск инициализации SRIO

Перед началом работы порта SRIO необходимо перевести его в рабочее состояние, установив бит EN\_SRIO в регистре SRIO\_CSR.

Процесс первоначальной установки параметров, запуска и проверки инициализации порта SRIO состоит из следующих шагов:

1. В бите TX\_FLOW\_CTR\_EN регистра LPU\_CSR установить режим передачи порта: Transmitter Flow Control /Receiver Flow Control. Режим работы порта необходимо задавать до запуска инициализации.
2. Установить бит COMPARE\_EN в регистре PCS\_CSR, тем самым разрешая сравнение частот в приёмнике PMA\_RX. В данной версии порта SRIO это необходимо делать до запуска инициализации. Иначе инициализация может не пройти, зависит от начала инициализации соседнего устройства.
3. В поле FORCE\_REINIT\_MODE в регистре PCS\_CSR установить режим инициализации: канал 0 (режим 1X0), канал 2 (режим 1X2), все 4-ре канала (режим 4X). Для требуемого режима инициализации у порта должна быть обеспечена физическая связь по проводам с соседним устройством.
4. Запустить инициализацию SRIO, установив бит FORCE\_REINIT в регистре PCS\_CSR.
5. После запуска инициализации в поле INIT\_STATE регистра PCS\_CSR проверить состояние выполнения инициализации. Возможные следующие варианты:
  - INIT\_STATE = 001 (состояние SILENT) – порт был инициализирован, но из-за возникших ошибок инициализация пропала. Необходимо сбросить весь порт с помощью бита EN\_SRIO в регистре SRIO\_CSR или сбросить блок PCS порта с помощью бита CLR в регистре PCS\_CSR. Затем заново запустить процесс инициализации.
  - INIT\_STATE = 010 (состояние SEEK) – порт посылает idle последовательность и ожидает ответа от соседнего устройства. Порт SRIO может находиться в этом состоянии бесконечно долго, пока соседнее устройство не запустит свой процесс инициализации и начнёт передавать idle последовательность. Можно выполнить выход из этого состояния только при помощи сброса EN\_SRIO в регистре SRIO\_CSR или CLR в регистре PCS\_CSR.
  - INIT\_STATE = 011 (состояние DISCOVERY1) или INIT\_STATE = 100 (состояние DISCOVERY2) – порт продолжает инициализацию в режиме 4X.
  - INIT\_STATE = 101 (состояние MODE1X0) – порт инициализировал 0-ой канал. Такая инициализация проходит в двух случаях. При установке FORCE\_REINIT\_MODE = 000 (режим 1X0) и соседнее устройство инициализируется в режиме 1X0 или 4X. А также при установке FORCE\_REINIT\_MODE = 010 (режим 4X), а соседнее устройство инициализируется в режиме 1X0.
  - INIT\_STATE = 110 (состояние MODE1X2) – порт инициализировал 2-ой канал. Такая инициализация проходит в двух случаях. При установке FORCE\_REINIT\_MODE = 001 (режим 1X2) и соседнее устройство инициализируется в режиме 1X2 или 4X. А также при установке FORCE\_REINIT\_MODE = 010 (режим 4X), а соседнее устройство инициализируется в режиме 1X2.
  - INIT\_STATE = 111 (состояние MODE4X) – порт инициализировал все 4-ре канала. Такая инициализация проходит только в одном случае. При установке



FORCE\_REINIT\_MODE = 010 и соседнее устройство инициализируется в режиме 4X.

В зависимости от заданного режима инициализации в поле FORCE\_REINIT\_MODE проверить корректность состояния выполнения инициализации в поле INIT\_STATE регистра PCS\_CSR.

6. Запретить сравнение частот в приёмнике PMA\_RX, сбросив бит COMPARE\_EN в регистре PCS\_CSR.
7. В регистре ERROR\_STATUS\_CSR дождаться, когда установятся биты PORT\_OK = 1, PORT\_UNINIT = 0. Это значит, что порт инициализировал соединение и обменялся требуемым количеством символов с соседним устройством. В процессе ожидания может возникнуть ситуация, когда PORT\_OK = 0 и PORT\_UNINIT = 0. Значит, что порт инициализировал соединение с соседним устройством, но символами ещё не обменялся, необходимо подождать.
8. Проверить поле TBUF\_STAT регистра ACKID\_CSR для определения режима передачи: Transmitter Flow Control /Receiver Flow Control. Режим Transmitter Flow Control устанавливается только в одном случае, когда порт и соседнее устройство работают в режиме Transmitter Flow Control. Иначе устанавливается режим Receiver Flow Control. Если TBUF\_STAT = 1F – порт и соседнее устройство работают в режиме Receiver Flow Control. Если TBUF\_STAT отлично от 1F, то порт и соседнее устройство работают в режиме Transmitter Flow Control.
9. Выполнить остальные базовые настройки порта согласно требованиям к работе порта и требованиям ко всей системе. К базовым настройкам относятся: установка счётчиков таймаутов, задание Device ID.

### 11.7.2 Потеря инициализации SRIO

При потере инициализации теряется связь между устройствами. Поэтому ни порт SRIO, ни соседнее устройство не могут передавать и принимать пакеты.

При потере инициализации в регистре ERROR\_STATUS\_CSR устанавливаются биты PORT\_ERROR и PORT\_UNINIT и сбрасывается бит PORT\_OK. А в регистре PCS\_CSR поле INIT\_STATE устанавливается в значение 001 (состояние SILENT). Значение бита PORT\_ERROR из регистра ERROR\_STATUS\_CSR дублируется в одноимённом бите PORT\_ERROR в регистре SRIO\_CSR. А бит PORT\_ERROR из регистра SRIO\_CSR является прерыванием для RISK ядра. Таким образом, потеря инициализации приводит к возникновению прерывания в RISK ядре.

Причиной потери инициализации в режимах 1X0 и 1X2 является потеря синхронизации в соответствующем канале. Т.е. по каналу принято много ошибочных данных в заданный интервал времени, поэтому порт SRIO теряет инициализацию, устанавливает необходимые признаки (PORT\_ERROR, PORT\_OK, PORT\_UNINIT, SYNC\_ERROR и т.д.) и прекращает приём и передачу. Так передача остановлена, то соседнее устройство теряет синхронизацию и прекращает приём и передачу. Возможна ситуация, что сначала происходит потеря синхронизации в соседнем устройстве и оно прекращает приём и передачу, что ведёт к потере синхронизации в порте SRIO и уже порт теряет инициализацию и прекращает приём и передачу.

Поле SYNC\_ERROR регистра PCS\_CSR отображает потерю синхронизации по каналам. Для режима 1X0 при потере инициализации должно быть SYNC\_ERROR = 0001. Для режима 1X2 при потере инициализации должно быть SYNC\_ERROR = 0100.

Поля SYNC\_STATE0:SYNC\_STATE3 регистра PCS\_CSR отображают этапы синхронизации в каждом канале соответственно:

- SYNC\_STATE = 00 (состояние NO\_SYNC) – синхронизации в канале нет, это состояние после сброса или после потери синхронизации.
- SYNC\_STATE = 01 (состояние NO\_SYNC1) – в канале выполняется синхронизация.
- SYNC\_STATE = 10 (состояние SYNC) – синхронизация в канале выполнена, для режимов 1X0 и 1X2 получение синхронизации в соответствующем канале означает успешное завершение процесса инициализации.
- SYNC\_STATE = 11(состояние SYNC1) – синхронизация в канале есть, но присутствуют одиночные ошибки.

Для режима 1X0 после выполнения инициализации должно быть SYNC\_STATE0 = 10 или 11, а после потери инициализации SYNC\_STATE0 = 00. Для режима 1X2 после выполнения инициализации должно быть SYNC\_STATE2 = 10 или 11, а после потери инициализации SYNC\_STATE2 = 00.

У потери инициализации в режиме 4X есть несколько причин: потеря синхронизации в каком-либо из каналов, нарушение выравнивания между каналами. Если теряется синхронизация в любом канале порта SRIO, то порт теряет инициализацию, устанавливает все необходимые признаки(PORT\_ERROR, PORT\_OK, PORT\_UNINIT, SYNC\_ERROR и т.д.) и прекращает приём и передачу по всем каналам. Следовательно, соседнее устройство также теряет инициализацию и прекращает приём и передачу по всем каналам. Возможна ситуация, что сначала соседнее устройство прекращает приём и передачу из-за потери инициализации, что ведёт к потере инициализации в порте SRIO и прекращению приёма и передачи. Поля SYNC\_STATE0:SYNC\_STATE3, SYNC\_ERROR регистра PCS\_CSR отображают этапы синхронизации и потерю синхронизации по каналам.

Если синхронизация по всем каналам устойчивая, то возможно нарушение выравнивания между каналами что ведёт к потере инициализации. Для выравнивания каналов передатчик соседнего устройства по всем каналам одновременно передаёт специальные символы |A|. Из-за разности длины коммутирующих проводов, может получиться “разбег” между передаваемыми данными по каждому каналу. Поэтому приёмник, используя символы |A|, в процессе инициализации в режиме 4X единожды выполняет выравнивание каналов так, чтобы со всех каналов считывался столбец символов |A|. Если в процессе приёма это условие нарушается несколько раз за определённый интервал времени, то порт SRIO теряет инициализацию и прекращает передачу и приём. Символы |A| могут считываться не столбцом из-за ошибки в данных или из-за нарушения протокола передачи в передатчике.

Бит ALIGN\_ERROR регистра PCS\_CSR информирует о нарушении выравнивания между каналами после завершения инициализации. Поле ALIGN\_STATE регистра PCS\_CSR отражает состояние выполнения процедуры выравнивания в режиме 4X:

- ALIGN\_STATE = 00 (состояние NOT\_ALIGNED) – выравнивания каналов нет, это состояние после сброса или после потери инициализации 4X.
- ALIGN\_STATE = 01 (состояние NOT\_ALIGNED 1) – при инициализации 4X в каналах выполняется выравнивание.
- ALIGN\_STATE = 10 (состояние ALIGNED) – выравнивание во всех каналах выполнено, для режима 4X получение выравнивания во всех каналах означает успешное завершение процесса инициализации 4X.
- ALIGN\_STATE = 11(состояние ALIGNED1) – выравнивание во всех каналах есть, но присутствуют одиночные ошибки при считывании столбца символов |A|.

Для режима 4X после выполнения инициализации должно быть для всех

SYNC\_STATE<sub>n</sub> = 10 или 11 и ALIGN\_STATE = 10 или 11. После потери инициализации в зависимости от причины устанавливается соответствующий бит в поле SYNC\_ERROR или бит ALIGN\_ERROR в регистре PCS\_CSR. Остальные поля SYNC\_STATE<sub>n</sub> = 00 и ALIGN\_STATE = 00.

Если после потери инициализации для продолжения работы необходимо запустить новую инициализацию, то сначала необходимо выполнить сброс:

- если была передача пакетов через порт SRIO, то необходимо сбросить весь порт с помощью бита EN\_SRIO регистра SRIO\_CSR;
- если передачи пакетов не было, то можно сбросить только блок PCS порта с помощью бита CLR регистра PCS\_CSR или с помощью бита EN\_SRIO регистра SRIO\_CSR сбросить весь порт.

После сброса порта SRIO заново запускаем инициализацию.

## 12. КОНТРОЛЛЕР ИНТЕРФЕЙСА SpaceWire (SWIC)

### 12.1 Общие положения

Контроллер интерфейса SpaceWire (далее по тексту SWIC – Space Wire Interface Controller) предназначен для обеспечения аппаратной поддержки функций внутрисистемных коммуникаций с использованием протокола SpaceWire.

В микропроцессоре имеется два контроллера SWIC: SWIC0 и SWIC1.

Основные особенности контроллера:

- Контроллер разработан в соответствии с международным стандартом ECSS-E-50-12;
- Обеспечивает функционирование одного дуплексного канала связи со скоростью от 2 до 300 Мбит/с в каждую сторону;
- Реализация контроллера охватывает уровни стека протоколов SpaceWire, от сигнального до сетевого (частично) уровня;
- Аппаратное детектирование ошибок связи: рассоединение, ошибки четности;
- Встроенные LVDS приемопередатчики в соответствии со стандартом стандарта ANSI/TIA/EIA-644(LVDS);
- Встроенные в приемник LVDS резисторы-терминаторы;
- Содержит десятиразрядный регистр управления синтезатором частоты передачи;
- Четыре канала DMA (два канала данных и два канала дескрипторов пакетов);
- Обмен данными через DMA с памятью словами по 64 бита.

### 12.2 Блок-схема

Структура контроллера коммуникационного канала по стандарту SpaceWire приведена на Рисунок 12.1. Основой контроллера является DS-макроячейка, реализующая функции кодера/декодера SpaceWire. Кодер/декодер SpaceWire через драйверы LVDS подключен к физическим линиям связи.

Контроллер канала SW взаимодействует с центральным процессором через шину АНВ (работа с программно-доступными регистрами контроллера) и FIFO-подобный интерфейс с DMA (прием/передача пакетов данных). Для взаимодействия с внутренней памятью используются блоки DMA, поддерживающие интерфейс буферов. На шине CDB SWIC представлен интерфейсом ведомого устройства.

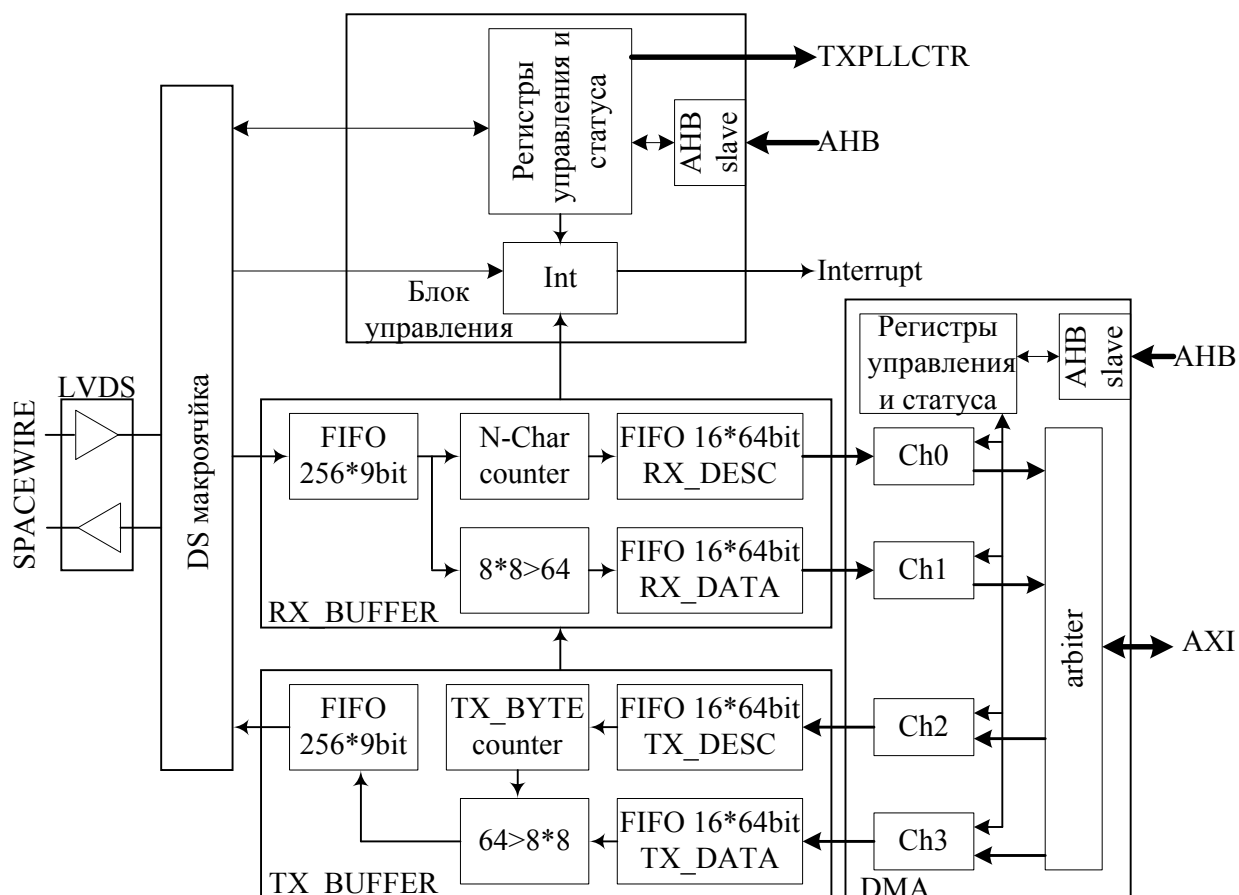


Рисунок 12.1 Структурная схема SWIC

Блок управления по командам центрального процессора задает режимы работы приемопередатчика SpaceWire (DS-макроячейки). В этом блоке содержатся программно управляемый регистр, содержащий коэффициент скорости передачи данных, и доступный программному обеспечению на чтение регистр, в который записывается коэффициент скорости приема данных. Передача управляющих кодов; контроль состояние последнего полученного извне маркера времени, кода распределенного прерывания и кода подтверждения производится через соответствующие регистры блока управления.

Блок формирования прерываний INT формирует необходимые прерывания по состоянию DS-макроячейки.

Буфер приема RX\_BUFFER имеет конвейерную организацию и состоит из двух ступеней. Сначала в FIFO\_256\*9bit буферизируются восьмиразрядные данные, принимаемые от DS-макроячейки. Девятый служебный разряд несет информацию о признаке символа данных N-Char или символе конца пакета EOP. Затем в блоке преобразования формируются 64-разрядные слова данных и поступают в FIFO RX\_DATA. Дескриптор пакета формируется в счетчике N-Char\_counter. При поступлении символа данных N-Char счетчик увеличивается на 1, при поступлении символа конца пакета значение счетчика переписывается в выходной буфер RX\_DESC, а сам счетчик сбрасывается в 0.

В буфер передачи TX\_BUFFER с помощью канала передаваемых данных DMA записываются 64-разрядные слова данных. Содержимое пакетов и их дескрипторы буферизируются в двух FIFO TX\_DATA и TX\_DESC соответственно. Данные из буфера передачи в DS-

макроячейку выдаются побайтно через FIFO 256\*9bit. Преобразование 64-хразрядных слов в байты осуществляется в блоке преобразования под управлением счетчика TX\_BYTE counter. В счетчик заносится размер пакета из дескриптора передаваемого пакета. После передачи каждого байта этот счетчик уменьшается на 1. По достижении счетчиком значения 0, в поток передаваемых данных вставляется символ конца пакета EOP, а в счетчик заносится размер следующего передаваемого пакета из следующего дескриптора.

Буферы приема-передачи предназначены для согласования скоростей передачи данных между коммутатором AXI Switch и каналом SpaceWire.

### 12.3 Прерывания

Контроллер SWIC формирует три прерывания, описание которых сведено в Таблица 12.1.

Таблица 12.1 Источники прерываний в SWIC

Условное обозначение	Причина	Примечание
LINK	Соединение установлено	В регистре STATUS указана причина прерывания: - CONNECTED.
ERR	Обнаружена ошибка в канале связи	В регистре STATUS указана причина прерывания: -DC_ERR; -P_ERR; -ESC_ERR; -CREDIT_ERR.
TIM	Получен управляющий код	В регистре STATUS указана причина прерывания: -принят маркер времени (GOT_TIME); -принят код распределенного прерывания; (GOT_INT) -принят код подтверждения (GOT_ACK) -принят управляющий код C[7..6]=01 (при включенном режиме 5-и разрядных распределенных прерываний) (CC_01) -принят управляющий код C[7..6]=11 (CC_11) -истекло время ожидания таймаута приема кода распределенного прерывания (регистр ISR_tout).

Схема формирования и маскирования прерываний следующая. Источники прерываний формируют импульс (лог. «1») признака какого-либо состояния, этот импульс фиксируется в триггере и хранится на его выходе до тех пор, пока не будет произведен сброс прерывания записью “1” в соответствующий причине прерывания разряд регистра STATUS. После сброса контроллера все прерывания являются замаскированными. Для того чтобы демаскировать прерывание, необходимо установить соответствующий разряд регистра режима (IRQ\_0\_mask, IRQ\_1\_mask, IRQ\_2\_mask соответственно) в 1.

С выхода триггеров сигналы прерываний доступны процессору по чтению в регистре STATUS в разрядах [19:17].

## 12.4 Перечень регистров SWIC

### 12.4.1 Общие положения

Перечень программно-доступных регистров контроллера SWIC приведен в Таблица 12.2.

Таблица 12.2 Перечень регистров блока SWIC

Условное обозначение регистра	Название регистра	Тип доступа
HW_VER	Номер версии контроллера	RD
STATUS	Регистр состояния	WR/RD
RX_CODE	Регистр управляющего символа, принятого из сети (маркера времени, кода распределенного прерывания или кода подтверждения распределенного прерывания)	RD
MODE_CR	Регистр режима работы	WR
TX_SPEED	Регистр коэффициента скорости передачи	WR
TX_CODE	Регистр управляющего символа (маркера времени, кода распределенного прерывания, кода подтверждения) для передачи в сеть	WR
RX_SPEED	Регистр скорости приема данных в канале SpaceWire.	RD
CNT_RX_PACK	Регистр счетчика принятых пакетов ненулевой длины	RD/WR
CNT_RX0_PACK	Регистр счетчика принятых пакетов нулевой длины (идущих подряд символов концов пакетов)	RD/WR
ISR_L	Младшие разряды регистра ISR	RD/WR
ISR_H	Старшие разряды регистра ISR	RD/WR
TRUE_TIME	Регистр достоверного маркера времени	RD
TOUT_CODE	Регистр размера таймаутов	RD/WR
ISR_tout_L	Младшие разряды регистра флагов таймаутов ISR	RD/WR
ISR_tout_H	Старшие разряды регистра флагов таймаутов ISR	RD/WR
LOG_ADDR	Регистр логического адреса	RD/WR

## 12.5 Описание регистров SWIC

### 12.5.1 Регистр HW\_VER

Регистр номера версии SWIC. При чтении этого регистра выводится номер версии аппаратной реализации SWIC. В 1892BM7Я аппаратная версия SWIC – “0x0000 0003”.

Таблица 12.3 Назначение разрядов регистра HW\_VER

Номер разряда	Условное обозначение	Описание
31:0	HW_VER	Номер версии SWIC



## 12.5.2 Регистр STATUS

Регистр состояния блока SWIC предназначен для оперативного контроля состояния фаз работы контроллера. Регистр доступен как на чтение, так и на запись. Заполнение регистра выполняется побитно по сигналам от DS-макроячейки, блока приема данных из канала SpaceWire, блока передачи данных в канал SpaceWire. Назначение разрядов регистра приведено в Таблица 12.4.

Таблица 12.4 Назначение разрядов регистра STATUS

Номер разряда	Условное обозначение	Описание
0	DC_ERR	Признак ошибки разъединения (DisconnectError): «1» – Ошибка произошла «0» – Нет ошибки (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в «0». Используется для сброса прерывания ERR посредством записи «1» в этот разряд. Исходное состояние - «0»
1	P_ERR	Признак ошибки четности: «1» – Ошибка произошла «0» – Нет ошибки (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в "0". Используется для сброса прерывания ERR посредством записи «1» в этот разряд. Исходное состояние - «0»
2	ESC_ERR	Признак ошибки в ESC последовательности: «1» – Ошибка произошла «0» – Нет ошибки (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в «0». Используется для сброса прерывания ERR посредством записи «1» в этот разряд. Исходное состояние - «0»
3	CREDIT_ERR	Признак ошибки кредитования: «1» – Ошибка произошла «0» – Нет ошибки (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в «0». Используется для сброса прерывания ERR посредством записи «1» в этот разряд. Исходное состояние - «0»
4	-	Не используется
5 - 7	DS_STATE	Состояние DS-макроячейки. Исходное состояние «000».
8	RX_BUF_FULL	Состояние буфера приема: «1» – буфер полон; «0» – в буфере есть свободное место (после сигнала сброса). Исходное состояние - «0»
9	RX_BUF_EMPTY	Буфер приема пуст «1» – Пуст (после сигнала сброса) «0» – В буфере есть данные Исходное состояние - «1»
10	TX_BUF_FULL	Состояние буфера передачи: «1» – буфер полон; «0» – в буфере есть свободное место (после сигнала сброса). Исходное состояние - «0»
11	TX_BUF_EMPTY	Буфер передачи пуст «1» – Пуст (после сигнала сброса) «0» – В буфере есть данные. Исходное состояние - «1»



Номер разряда	Условное обозначение	Описание
12	GOT_FIRST_BIT	Состояние принятого первого бита из канала «1» – бит принят «0» – приемный канал не активен (не было изменений фронтов din/sin после последнего сброса макроячейки по сбросу или в связи с ошибкой) Запись «1» в этот бит сбрасывает прерывание INT_LINK, если оно было установлено, но не изменяет состояние GOT_FIRST_BIT. Исходное состояние - «0»
13	CONNECTED	Соединение установлено (DS_STATE=5). Исходное состояние - «0»
14	GOT_TIME	Принят маркер времени из сети «1» – Принят маркер времени «0» – Маркер времени не принят (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в «0». Используется для сброса прерывания TIM посредством записи «1» в этот разряд. Исходное состояние - «0»
15	GOT_INT	Принят код распределенного прерывания из сети «1» – Принят код распределенного прерывания времени «0» – Код распределенного прерывания не принят (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в «0». Используется для сброса прерывания TIM посредством записи 1 в этот разряд. Исходное состояние «0».
16	GOT_ACK	Принят код подтверждения из сети «1» – Принят код подтверждения «0» – код подтверждения не принят (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в «0». Используется для сброса прерывания TIM посредством записи «1» в этот разряд. Исходное состояние - «0»
17	FL_CONTROL	Если данный флаг сброшен в «0», SWIC готов к отправке управляющего кода (маркера времени, кода распределенного прерывания, кода подтверждения). Если управляющий код записывается в SWIC при установленном флаге, его передача в сеть не гарантируется. Исходное состояние - «0»
18	IRQ_0	Значение сигнала прерывания 0 (установка соединения). Устанавливается при установке соответствующего прерывания, если оно не замаскировано в регистре режима. Сбрасывается при сбросе соответствующего прерывания. Исходное состояние - «0»
19	IRQ_1	Значение сигнала прерывания 1 (разрыв соединения). Устанавливается при установке соответствующего прерывания, если оно не замаскировано в регистре режима. Сбрасывается при сбросе соответствующего прерывания. Исходное состояние - «0»
20	IRQ_2	Значение сигнала прерывания 2 (принят управляющий код). Устанавливается при установке соответствующего прерывания, если оно не замаскировано в регистре режима. Сбрасывается при сбросе соответствующего прерывания. Исходное состояние - «0»
21	CC_11	Признак принятия управляющего кода C[7..6]=11 «1» – Принят управляющий код «0» – управляющий код не принят (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в «0». Исходное состояние - «0»
22	CC_01	Признак принятия управляющего кода C[7..6]=01 «1» – Принят управляющий код «0» – Упр. код не принят (после сигнала сброса) Запись «1» в этот разряд сбрасывает этот разряд в «0». Исходное состояние - «0»
23..29	-	Резерв
30	S LVDS_RX	Значение линии LVDS Sin при MODE_CR[29]=1.
31	D LVDS_RX	Значение линии LVDS Din при MODE_CR[29]=1

### 12.5.3 Регистр RX\_CODE

Регистр принятого из сети управляющего кода. Назначение разрядов регистра приведено в Таблица 12.5. Исходное состояние регистра не определено.

Таблица 12.5 Назначение разрядов регистра RX\_CODE

Номер разряда	Условное обозначение	Описание
7:0	TIME_CODE	Значение маркера времени, принятого из сети последним
15:8	INT_CODE	Значение кода распределенного прерывания, принятого из сети последним
23:16	ACK_CODE	Значение кода подтверждения, принятого из сети последним
31:24	CC_11	Значение кода C[7..6]=11 принятого из сети последним

### 12.5.4 Регистр MODE\_CR

Регистр режима работы. Назначение разрядов регистра приведено в Таблица 12.6.

Таблица 12.6 Назначение разрядов регистра MODE\_CR

Номер разряда	Условное обозначение	Назначение
0	LinkDisabled	Установка LinkDisabled для блока DS-кодирования
1	AutoStart	Установка Autostart для блока DS-кодирования
2	LinkStart	Установка LinkStart для блока DS-кодирования
3	-	Не используется
4	-	Не используется
5	DSM_RST	Сброс DS-макроячейки
6	RDY_MODE	Режим формирования признака готовности обмена данными с DMA SWIC: 0 – штатный режим работы. Признак готовности SWIC формирует аппаратно; 1 – признак готовности установлен в 1. Используется для приведение DMA SWIC в исходное состояние, если: произошло разъединение; необходимо программно остановить SWIC и его DMA
7	-	Не используется
8	TEST_TYPE	Тип режима работы ('0' – рабочий, '1' – тестовый)
9	TX_SINGLE	Включение режима Single на передачу
10	RX_SINGLE	Включение режима Single на прием
11	LVDS_Loopback	Loopback (перед LVDS)
12	CODEC_Loopback	Loopback (перед кодеком)
13	DS_Loopback	Loopback (перед DS-макроячейкой)
14	COEFF_10_wr	Разрешение модификации регистра коэффициента для подсчета таймаутов
15	AUTO_SPEED	Если этот бит установлен в 1, то при разрыве соединения коэффициент скорости передачи будет автоматически устанавливаться на 10МГц, а при установке соединения автоматически переходит на базовое значение скорости
16	dirQ_regime	Режим передачи/приема кодов распределенных прерываний. Если этот бит установлен в 0, то используются 6-и битные коды распределенных прерываний, если в 1 – то используются 5-и битные коды распределенных прерываний
17	-	Не используется
18	IRQ_0_mask	Маска прерывания IRQ0. Если значение маски установлено в 1, то значение прерывания отображается в регистр STATUS и участвует в формировании сигнала IRQ_all. Если значение 0, значение прерывания не отображается в регистр STATUS и не участвует в формировании сигнала IRQ_all.

Номер разряда	Условное обозначение	Назначение
19	IRQ_1_mask	Маска прерывания IRQ1. Если значение маски установлено в 1, то значение прерывания отображается в регистр STATUS и участвует в формировании сигнала IRQ_all. Если значение 0, значение прерывания не отображается в регистр STATUS и не участвует в формировании сигнала IRQ_all.
20	IRQ_2_mask	Маска прерывания IRQ2. Если значение маски установлено в 1, то значение прерывания отображается в регистр STATUS и участвует в формировании сигнала IRQ_all. Если значение 0, значение прерывания не отображается в регистр STATUS и не участвует в формировании сигнала IRQ_all.
21	CTR	Если этот бит установлен в 1, то установка соединения выполняется без ожидания таймаутов (используется в отладочном режиме)
22	TCode_mask	Если этот разряд установлен в 0, то прерывание IRQ2 при получении тайм-кода не устанавливается
23	INT_mask	Если этот разряд установлен в 0, то прерывание IRQ2 при получении кода распределенного прерывания или кода подтверждения не устанавливается
24	CC_11_mask	Если этот разряд установлен в 0, то прерывание IRQ2 при получении управляющего кода C[7..6]=11 не устанавливается
25	CC_01_mask	Если этот разряд установлен в 0, то прерывание IRQ2 при получении управляющего кода C[7..6]=01 (dIRQ_regime=1) не устанавливается
26	INT_tout_mask	Если этот разряд установлен в 0, то прерывание IRQ2 по факту истечения таймаута получения кода подтверждения не устанавливается
28:27	INT_tout_allow	Разрешение контроля таймаутов получения кодов подтверждения: 00 – контроль таймаутов запрещен 01 – выполняется контроль таймаутов и установка флагов истечения таймаутов 10 – выполняется контроль таймаутов, установка флагов истечения таймаутов и отправка кода подтверждения в сеть
29	LVDS_regime	Режим LVDS – если этот бит установлен в 0 – штатный режим работы, на выходные драйверы LVDS подаются сигналы от передатчика, разряды регистра STATUS[31:30] равны «0»; 1 – тестовый режим работы, на Sout, Dout LVDS подаются значения из разрядов 30, 31 регистра MODE_CR, в регистр STATUS[31:30] отображаются входные линии Sin и Din LVDS.
30	S_LVDS_TX	Значение для передачи на линию Sout LVDS
31	D_LVDS_TX	Значение для передачи на линию Dout LVDS

После того, как в результате разрешения AutoStart или LinkStart блок DS-кодирования установил соединение (при LinkDisabled='0'), буфер передачи в сеть начинает принимать данные из DMA. Если DMA передал все данные, то далее в сеть передаются символы NULL. Соединение при этом не прекращается. Соединение прекращается, если процессор осуществляет запись единицы в бит LinkDisabled.

### 12.5.5 Регистр TX\_SPEED

Регистр коэффициентов скорости передачи. В разряды 9:0 записывается коэффициент, который передается на TXPLL при полностью программном управлении скоростью передачи. При использовании автоматического перехода на 10 МГц при разрыве соединения, коэффициент, записанный в разряды 9:0 устанавливается, когда соединение установлено. При разрыве соединения в этом режиме автоматически устанавливается коэффициент, записанный в разряды 19:10, он должен соответствовать скорости передачи 10 МГц.

В разряды 28:20 этого регистра записывается значение коэффициента для подсчета таймаутов установки соединения (6,4 мкс и 12,8 мкс). Значение данного коэффициента зависит от тактовой частоты работы CPU (см. п.4.1), на которой осуществляется подсчет таймаутов.

Значение после сброса для этого регистра – 0x0A, что соответствует тактовой частоте CPU 100 МГц.

Запись нового значения в этот регистр возможно только, если бит COEFF\_10\_wt (14) регистра MODE\_CR (режима) установлен в 1.

**Таблица 12.7 Назначение разрядов регистра TX\_SPEED**

Номер разряда	Условное обозначение	Назначение
7:0	TX_SPEED	Определяет скорость передачи данных (в режиме авто установки скорости используется как базовое значение после установки соединения), Мбит/с: 0x01 - 5; 0x02 – 10; .... 0x4F – 395; 0x50 – 400.
8	PLL_TX_EN	Управление работой PLL_TX_SWIC: 1 – работа разрешена; 0 – работа запрещена. PLL_TX_SWIC находится в режиме пониженного энергопотребления
9	LVDS_EN	Управление работой приемопередатчиков LVDS SWIC: 1 – работа разрешена; 0 – работа запрещена. LVDS SWIC находятся в режиме пониженного энергопотребления
19:10	TX_SPEED_10	Определяет скорость передачи данных при установке соединения (в режиме авто установки скорости). Должен быть записан код 0x02
28:20	COEFF_10	Значение коэффициента для подсчета таймаутов установки соединения. Содержимое этого поля должно быть целым от деления тактовой частоты работы CPU (CLK) на 10: 0x05 - при CLK = 50 или 55 МГц; 0x0A - при CLK = 100 или 105 МГц; ... 0x14 - при CLK = 200 или 205 МГц и т.д.
31..29	-	Резерв

### 12.5.6 Регистр TX\_CODE

Регистр управляющего кода для передачи в канал. Сразу же после записи в этот регистр начинается передача управляющего символа в DS-макроячейку и далее в канал.

**Таблица 12.8 Назначение разрядов регистра TX\_CODE**

Номер разряда	Условное обозначение	Описание
5:0	CODE_VAL	Значение управляющего кода для отправки в сеть
7:6	CODE_TYPE	Тип управляющего кода для отправки в сеть: 00 – код времени; 01 – код прерывания; 10 – код подтверждения прерывания.
31:8	-	Резерв

### 12.5.7 Регистр *RX\_SPEED*

Назначение разрядов регистра *RX\_SPEED* приведено в Таблица 12.9.

Таблица 12.9 Назначение разрядов регистра *RX\_SPEED*

Номер разряда	Условное обозначение	Назначение
31:10	-	Не используется
9:0	<i>RX_SPEED</i>	Скорость приема данных из канала связи SpaceWire равна ( $RX\_SPEED * 800 / 1024$ ) Мбит/с. Например, если $RX\_SPEED = 128$ (десятичное), то скорость приема данных равна 100 Мбит/с

Исходное состояние регистра – нули.

### 12.5.8 Регистр *CNT\_RX\_PACK*

Регистр счетчика принятых пакетов. Значение регистра увеличивается на 1 каждый раз, когда из DS макроячейки прочитывается символ конца пакета, если ему предшествовал один или более символ данных. Исходное состояние регистра «0».

При записи, значение регистра обнуляется. Процессор может обнулить содержимое этого регистра для того, чтобы начать счет пакетов заново. Рекомендуется выполнять сброс регистра каждый раз при выполнении новой настройки DMA для передачи данных в сеть.

Таблица 12.10 Назначение разрядов регистра *CNT\_RX\_PACK*

Номер разряда	Условное обозначение	Описание
31:0	<i>CNT</i>	Число принятых пакетов

### 12.5.9 Регистр *CNT\_RX0\_PACK*

Регистр счетчика принятых пустых пакетов. Значение регистра увеличивается на 1 каждый раз, когда из DS макроячейки прочитывается символ конца пакета, если ему не предшествовал хотя бы один символ данных. Исходное состояние регистра «0».

При записи, значение регистра обнуляется. Процессор может обнулить содержимое этого регистра для того, чтобы начать счет пакетов заново. Рекомендуется выполнять сброс регистра каждый раз при выполнении новой настройки DMA для передачи данных в сеть.

Таблица 12.11 Назначение разрядов регистра *CNT\_RX0\_PACK*

Номер разряда	Условное обозначение	Описание
31:0	<i>CNT</i>	Число принятых пустых пакетов

### 12.5.10 Регистр *ISR\_L*

В этот регистр отображается младшая (31..0) часть регистра *ISR*. Регистр *ISR* содержит информацию о принятых и отправленных кодах распределенных прерываний и подтверждения. Если из сети получено распределенное прерывание, то бит регистра *ISR*, соответствующий номеру распределенного прерывания устанавливается в 1 (если он уже не был установлен в 1). Аналогично, если в регистр *TX\_CODE* осуществляется запись кода распределенного прерывания, соответствующий бит регистра *ISR* устанавливается в 1.

Если из сети получен код подтверждения, то бит регистра *ISR*, соответствующий номеру кода подтверждения устанавливается в 0 (если он уже не был установлен в 0). Аналогично, ес-

ли в регистр TX\_CODE осуществляется запись кода подтверждения, соответствующий бит регистра ISR устанавливается в 0.

Необходимость данного регистра связана с тем, что коды распределенных прерываний и коды подтверждения могут приходиться из сети очень часто, быстрее, чем процессор может среагировать на очередное прерывание и прочитать код. Если даже в регистре RX\_CODE код распределенного прерывания или код подтверждения будет перезаписан следующим, информация о нем не будет утрачена – она сохранится в регистре ISR.

Существует возможность программного сброса отдельных битов ISR. Для этого необходимо записать в соответствующие биты 1. (Если в бит записывается значение 0, то его значение не меняется).

**Таблица 12.12 Назначение разрядов регистра ISR\_L**

Номер разряда	Условное обозначение	Описание
31:0	ISR_L	Младшая часть регистра ISR

### 12.5.11 Регистр ISR\_H

В этот регистр отображается старшая [63:32] часть регистра ISR.

**Таблица 12.13 Назначение разрядов регистра ISR\_H**

Номер разряда	Условное обозначение	Описание
31:0	ISR_H	Старшая часть регистра ISR

### 12.5.12 Регистр TRUE\_TIME

В этот регистр записывается значение последнего правильного маркера времени, в отличие от разрядов 5:0 регистра RX\_CODE, в котором регистрируются все принятые маркеры времени. Назначение разрядов регистра приведено в Таблица 12.14. Исходное состояние регистра «0».

**Таблица 12.14 Назначение разрядов регистра TRUE\_TIME**

Номер разряда	Условное обозначение	Описание
5:0	TRUE_TIME	Значение последнего правильного маркера времени
31:6	Не используется	

### 12.5.13 Регистр TOUT\_CODE

В этот регистр записываются значение периода для глобального счетчика таймаутов (в количестве тактов локальной частоты) и максимальные значения локальных счетчиков таймаутов ожидания кодов подтверждения распределенных прерываний. Отдельный локальный счетчик таймаутов соответствует каждому разряду ISR. Если в SWIC поступает код распределенного прерывания, то запускается соответствующий ему счетчик локальных таймаутов. Он декрементируется каждый раз при завершении очередного периода счета глобального счетчика таймаутов.

**Таблица 12.15 Назначение разрядов регистра TOUT\_CODE**

Номер разряда	Условное обозначение	Описание
15..0	GLOB_COU	Значение периода глобального счетчика (задается в тактах локальной частоты)



Номер разряда	Условное обозначение	Описание
20..16	LOC_COU1	Значение таймаута ожидания кода подтверждения (на код прерывания, отправленный процессором через SWIC)
25..21	LOC_COU2	Значение таймаута ожидания кода подтверждения (на код прерывания, принятый из сети)
31:26	Не используется	

### 12.5.14 Регистр *ISR\_tout\_L*

В этот регистр отображается младшая (31..0) часть регистра флагов *ISR\_tout*. Если в регистре *ISR* регистрируется код распределенного прерывания, то для него запускается счет таймаута (каждому разряду *ISR* соответствует отдельный счетчик). В зависимости от того, был ли код распределенного прерывания принят из сети или отправлен процессором, начальное значение счетчика устанавливается в *LOC\_TOUT1* или *LOC\_TOUT2*. (значение счетчика декрентируется каждый раз, когда глобальный счетчик досчитывает до определенного для него максимального значения). Если за время счета из сети не поступает соответствующий код подтверждения, то соответствующий разряд регистра *ISR\_tout* устанавливается в 1. Для того чтобы его сбросить, необходимо записать в этот разряд регистра *ISR\_tout* 1. (При записи в бит значения 0, его значение не меняется)

Таблица 12.16 Назначение разрядов регистра *ISR\_tout\_L*

Номер разряда	Условное обозначение	Описание
31:0	<i>ISR_tout_L</i>	Младшая часть регистра <i>ISR_tout</i>

### 12.5.15 Регистр *ISR\_tout\_H*

В этот регистр отображается старшая (63..32) часть регистра *ISR\_tout*.

Таблица 12.17 Назначение разрядов регистра *ISR\_tout\_H*

Номер разряда	Условное обозначение	Описание
31:0	<i>ISR_tout_H</i>	Старшая часть регистра <i>ISR_tout</i>

После сброса содержимое регистров «0».

### 12.5.16 Регистр *LOG\_ADDR*

В этом регистре хранится значение логического адреса, добавляемого к пакету по умолчанию, если установлен соответствующий режим. Длина логического адреса может быть от одного до 4 байтов, она определяется значением дескриптора пакета.

Таблица 12.18 Назначение разрядов регистра *LOG\_ADDR*

Номер разряда	Условное обозначение	Описание
31:0	<i>LOG_ADDR</i>	Значение логического адреса.

## 12.6 Работа со SWIC. Пакеты данных, дескрипторы пакетов.

В этой главе описывается формирование пакетов данных в памяти для передачи в канал, формат пакетов данных, дескрипторов, передача данных из памяти в канал SpaceWire, прием данных из канала SpaceWire в память, интерпретирование принятых данных, системные сообщения.

### **12.6.1 Расположение данных в памяти.**

Рассмотрим пример (см. Рисунок 12.4) представления данных в системной памяти, если для данных выделен один сегмент памяти. Пусть в системную память из канала SpaceWire было записано 3 пакета. Первый пакет имеет размер 10 байт и заканчивается символом EOP. Второй пакет имеет размер 8 байт и заканчивается символом EEP. Третий пакет имеет размер 11 байт и заканчивается символом EOP. Собственно пакеты хранятся в сегменте памяти, выделенном процессором для записи данных. Для выравнивания по границам 64-х разрядных слов, первый и третий пакет дополнены двумя и одним байтом соответственно.

Дескрипторы хранятся в сегменте памяти, выделенном процессором для записи дескрипторов. В дескрипторах указаны размеры пакетов в байтах – 0Ah, 08h и 0Bh соответственно. В дескрипторах хранится так же информация о типе конца пакета. В разряд 31 дескриптора записывается 1, что указывает процессору на то, что дескриптор заполнен действительными данными.

### **12.6.2 Схема обработки данных процессором**

В данном примере пакеты могут быть обработаны процессором в соответствии со следующей схемой. Процессор прочитывает первое слово из блока, выделенного для дескрипторов – первый дескриптор. По дескриптору он определяет тип конца пакета, в соответствии с этим решает, как его обрабатывать. По дескриптору он определяет действительный размер пакета и извлекает данные, относящиеся к пакету 1. Для того чтобы вычислить начальный адрес второго пакета к начальному адресу блока данных добавляется размер первого пакета и выполняется округление до границы ближайшего слова. После того, как первый пакет полностью обработан, процессор прочитывает дескриптор второго пакета. Обработка остальных пакетов выполняется аналогично. Процесс обработки очереди пакетов заканчивается, когда 31 разряд очередного дескриптора равен 0.

### **12.6.3 Прием данных из канала SpaceWire.**

Маршрут принимаемых данных и схема их обработки приведены на Рисунок 12.2.



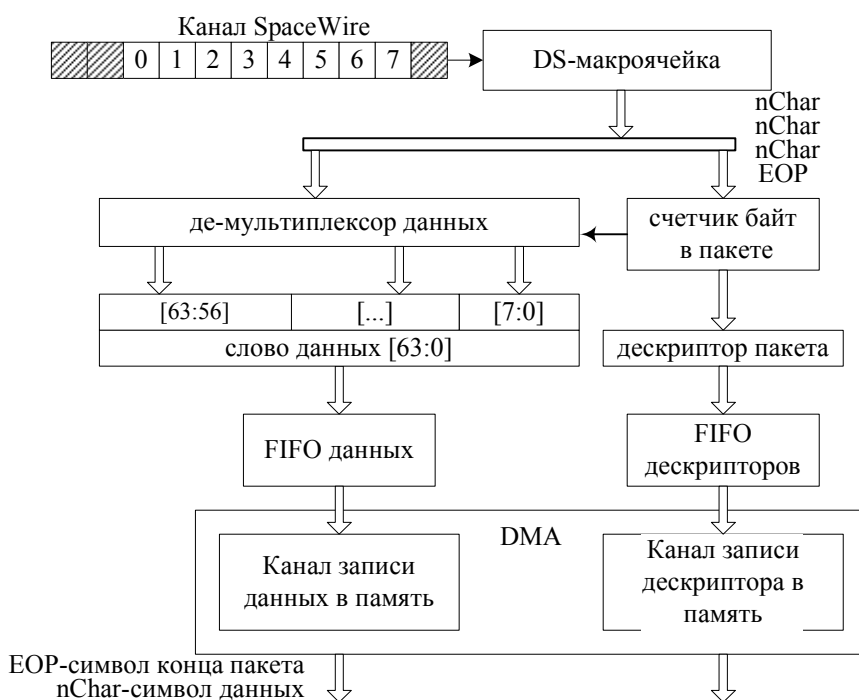


Рисунок 12.2 Схема приема данных из канала SpaceWire

Из DS-линков в DS-макроячейку символы данных поступают последовательно (по-битно). DS-макроячейка выделяет из последовательности приходящих символов символы данных и символы концов пакетов и передает их в блок приема. По DS-линку байты данных передаются младшими разрядами вперед.

Передача всех разрядов символа (9 разрядов, из них 8 используется для представления собственно байта данных, девятый бит является дополнительным и указывает, является ли этот байт символом данных nChar или символом конца пакета EOP) от DS-макроячейки в блок приема осуществляется в параллельном коде.

Подсчет числа символов nChar и формирование дескриптора при приеме символа конца пакета осуществляется в счетчике байт в пакете.

В блоке приема из байтов данных формируются слова разрядности 64. При формировании слов первый поступивший байт размещается в разрядах 7:0, второй – в разрядах 15:8, третий – в разрядах 23:16, четвертый – в разрядах 31:24 и т.д. Распределение символов данных по разрядам слова данных производится по счетчику байт.

Для того чтобы сократить загрузку процессора в ходе последующей обработки пакетов данных, в этом блоке выполняется выравнивание границ пакетов по границам слов и формирование дескрипторов пакетов, позволяющих процессору распознать границы отдельных пакетов.

Собственно пакеты данных и дескрипторы пакетов могут храниться в различных областях памяти. Местоположение этих областей в памяти определяется процессором при настройке каналов DMA. Дескрипторы пакетов записываются в память друг за другом и логически организованы в очередь.

### 12.6.4 Передача данных в канал SpaceWire

Процесс передачи пакетов данных из системной памяти в канал через контроллер, а также преобразование форматов данных показаны на Рисунок 12.3.

Пакеты данных загружаются из системной памяти в буфер передачи через каналы DMA чтения данных из памяти и чтения дескриптора из памяти.

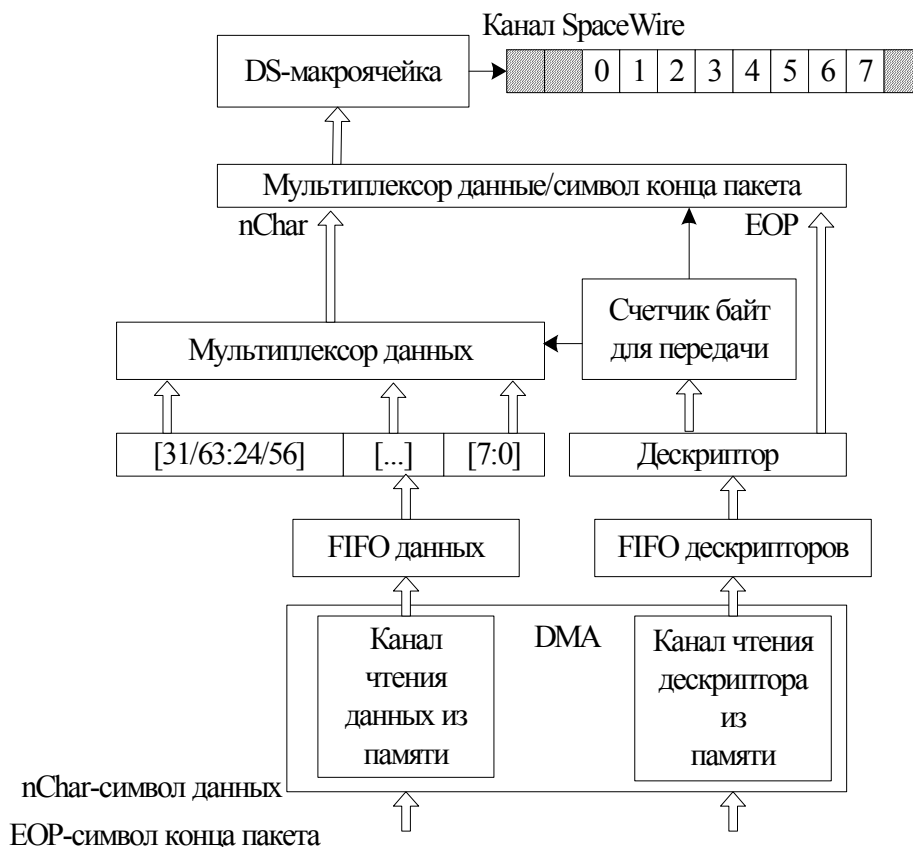


Рисунок 12.3 Передача данных из системной памяти в DS-линк

Блок передачи разбивает слова на отдельные байты. При этом из последовательности байтов в соответствии с информацией, содержащейся в дескрипторе, удаляются “лишние” байты – байты, добавленные для выравнивания пакетов по границам слов, и вставляются символы концов пакетов EOP или EEP. Если в DS-линк передаются пакеты, сгенерированные в данном узле, то предполагается, что они всегда должны заканчиваться символом EOP. Однако пакеты могут проходить через данный процессорный модуль транзитом. В этом случае они могут заканчиваться символом EEP. Коды маркеров EOP или EEP формируются контроллером аппаратно, на основании кодов дескриптора пакета на передачу (разряды 29:30 дескриптора пакета). Сами дескрипторы пакетов на передачу в сеть из основной памяти формируются программно.

Распаковка 64-разрядного слова в последовательность из 8 байт при передаче из контроллера выполняется по правилу, согласованному с правилом упаковки байтов при приеме данных из канала в контроллер.

Блок передачи вначале передает в DS-макроячейку байт данных, находящийся в разрядах 7:0 слова, затем байт, находящийся в разрядах 15:8, затем байт, находящийся в разрядах 23:15, затем байт из разрядов 31:24 и т.д. 64-разрядного слова.

Символы данных и концов пакетов передаются блоком передачи в блок DS-макроячейки. DS-макроячейка преобразует полученные символы в соответствии с алгоритмом DS кодирования и передает их в канал. Символы передаются младшими разрядами вперед.

### 12.6.5 Выравнивание границ пакетов по границам слов

Рассмотрим выравнивание пакетов данных на примере Рисунок 12.4. Если очередное слово данных сформировано не полностью (действительными данными заполнены один, два или три байта слова), а следующий символ в последовательности – символ конца пакета, то заполнение данного слова прекращается. Первый символ следующего пакета будет записан в первый байт нового слова. Действительный размер пакета в байтах записывается в дескриптор пакета. Это позволяет процессору при обработке пакета исключить из рассмотрения “лишние” байты – байты, добавленные для выравнивания пакетов по границам слов. В дескриптор заносится также информация о типе конца пакета (нормальный конец пакета – EOP, или признак завершения пакета с ошибкой – EEP).

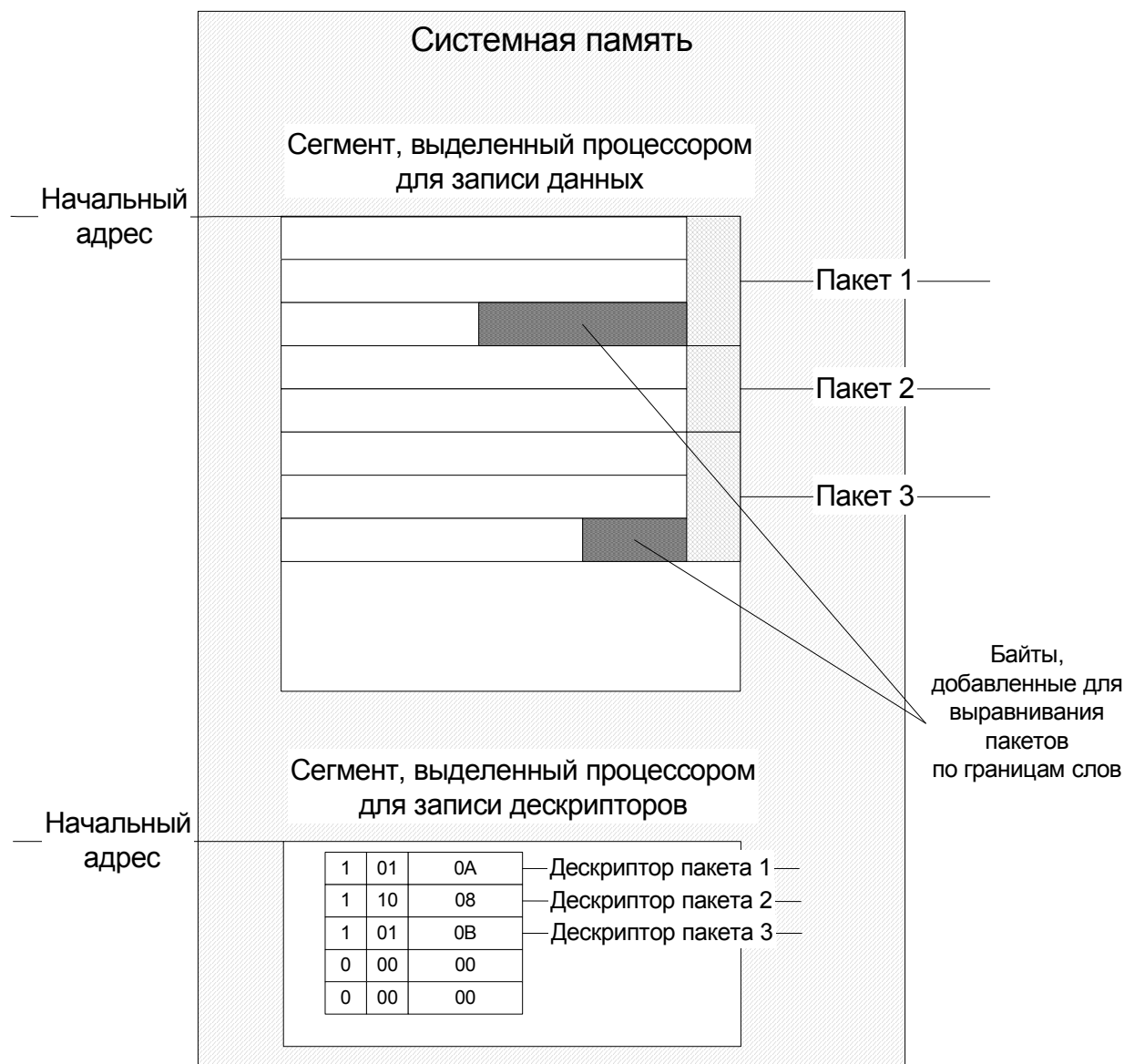


Рисунок 12.4 Представление данных в памяти (пример)

### 12.6.6 Формат дескриптора пакета

Дескриптор пакета имеет следующую структуру:

[63:32] – не используются;

[31] – признак заполнения дескриптора действительными данными. Бит учитывается только при приёме пакетов (позволяет процессору идентифицировать конец очереди дескрипторов в памяти). При передаче пакетов этот бит не учитывается (DMA вычитывает всю область дескрипторов, заданную процессором). До запуска приёма, все 31-е биты дескрипторов области приёма должны быть обнулены программно; DMA не обнуляет 31-е биты не принятых дескрипторов, DMA только записывает '1' в 31-е биты принятых дескрипторов;

[30:29] – тип конца пакета:

00 – передавать данные пакета из памяти и не вставлять конец пакета

01 – EOP;

10 – EEP.

11 – передавать данные пакета из регистра LOG\_ADDR и не вставлять конец пакета

[28:25] – не используется «0b0000»;

[24:0] – размер пакета в байтах.

При использовании 64-разрядной версии SWIC биты [63:32] не используются и на приеме могут содержать случайные значения.

Тип конца пакета 00 рекомендуется использовать для того, чтобы формировать заголовки пакетов, используемые для маршрутизации при передаче пакетов через сеть, отдельно от собственно передаваемых данных. Заголовок пакета может включать в себя произвольное количество байтов (не кратное 4). Оформление такого заголовка как отдельного пакета позволяет избежать выравнивания собственно передаваемых данных при длине заголовка не кратной 4. В дальнейшем будем называть заголовок пакета, оформленный как отдельный пакет, коммуникационным пакетом.

Слова данных из буфера приема передаются в канал DMA записи данных в память. Дескрипторы из блока приема передаются в канал DMA записи дескриптора в память. Блок DMA записывает данные и дескрипторы в системную память в соответствии с настройками, выполненными процессором (через регистры DMASWIC).

Процессор для канала записи дескрипторов в память определяет начальный адрес блока памяти и размер блока памяти. Для записи собственно пакетов данных в память может быть задан один блок памяти (так же, как и для канала записи дескриптора в память) или последовательность блоков памяти, физически расположенных в разных местах памяти.

### 12.6.7 Возможность передачи коммуникационного пакета

Дескриптор пакета в битах [30:29] содержит информацию о типе передаваемого пакета. Пакет может иметь нормальное окончание (EOP, код 01), ошибочное окончание (EEP, код 10), конец пакета может отсутствовать (00), и пакет может иметь тип *коммуникационного пакета* LOG\_ADDR (11).

Если конец пакета отсутствует (код 00), то после передачи всех байт данных пакета, соответствующего дескриптору с битами [30:29]=00, конец пакета SpaceWire не посылается в канал. Эта возможность используется, чтобы слить два пакета в один. Например, первый пакет может иметь статус коммуникационного, второй - содержать передаваемые данные. Дескрип-

тор первого пакета в этом случае должен содержать длину коммуникационного пакета. Второй, замыкающий, пакет (пакет данных) должен содержать действительное значение числа байт в основном блоке данных, и тип пакета 01 или 10 (т.е. так же как при стандартной передаче данных). Описанная возможность позволяет отдельно формировать данные для коммуникационного пакета и данные основного пакета. При этом оба пакета располагаются друг за другом, каждому соответствует свой дескриптор, и данные пакетов выровнены по длине 64-разрядных слов.

Для тех случаев, когда программист предпочитает иметь заранее сформированный коммуникационный пакет, который бы вставлялся перед передаваемым пакетом данных из памяти, предусмотрен режим передачи коммуникационного пакета из регистра LOG\_ADDR. В этом случае нет необходимости формировать коммуникационный пакет для каждого пакета данных. Программисту следует записать в регистр LOG\_ADDRS данные коммуникационного пакета (максимум - 4 байта) и сформировать для него дескриптор по описанной выше схеме (в области дескрипторов на передачу, перед дескрипторами данных, для которых требуется вставка коммуникационного пакета LOG\_ADDR). Всегда, встречая дескриптор с кодом 11, SWIC передает число байт, указанное в этом дескрипторе, из регистра LOG\_ADDR, а не из памяти. После передачи данных из этого регистра в канал не высылаются конца пакета, таким образом, пакет из регистра сольется с данными следующего пакета.

В обоих случаях (при передаче пакета с дескриптором 11 или 00) при слиянии пакетов на приемной стороне будет принят пакет длиной, равной сумме длин переданных пакетов, первому из которых соответствовал дескриптор 11 или 00.

Передача подряд нескольких пакетов с дескрипторами 11 и 00 допустима, при этом все переданные пакеты с этими дескрипторами - сольются в один пакет на приемной стороне. После пакетов с идентификаторами 11 или 00 обязательно должен следовать пакет с идентификатором EOP или EEP.

### **12.6.8 Использование симплексного режима**

Блок SWIC позволяет осуществлять передачу данных в симплексном режиме. В этом режиме предусмотрено две возможности – передача в симплексном режиме и прием в симплексном режиме. При этом в симплексном режиме передающая сторона не получает информации о состоянии приемной стороны, например, передающая сторона не способна определить возникла ли ошибка на приемной стороне, и не может принять решение о перезапуске канала. Для гарантированного перезапуска (в случае разрыва связи на приемной стороне) используется механизм автоматического снижения передающей частоты и посылки в канал символов NULL, один из которых должен быть определен на приемной стороне как первый NULL. Далее в автоматическом режиме скорость снова может быть поднята.

Рассмотрим работу блоков приема и передачи в симплексном режиме подробнее.

При работе в симплексном режиме на прием (установка MODE\_CR[10]='1') блок приема работает так же как в обычном режиме. Он должен принять первый символ NULL на скорости 10 Мбит/с как в начале работы блока, так и при разрыве связи.

При активизации возможности передачи данных в симплексном режиме (установка MODE\_CR[9]='1') блок SWIC осуществляет запуск канала без участия приемника. Блок начинает передачу символов NULL на скорости 10 Мбит/с в течение 12.8 мкс. Затем устанавливается скорость из регистра скорости передачи и в канал передаются данные без участия системы кредитования по стандарту SpaceWire. Считается, что блок может посылать неограниченное число данных в канал. Через предустановленный интервал времени примерно 100 мкс блок автоматически снижает скорость до 10 Мбит/с на время 12.8 мкс и передает

только символы NULL. Эта схема при работе в симплексном режиме на передачу повторяется циклически.

Кратковременный переход на низкую скорость позволяет установить связь с приемной стороной, если на ней по каким-то причинам произошел разрыв связи. 12.8 мкс достаточно чтобы в канале передачи появился как минимум один символ NULL, который приемное устройство обязано трактовать как первый NULL и установить прием данных по симплексному каналу SpaceWire.

Блок SWIC может быть настроен одновременно на работу в симплексном режиме сразу по обоим каналам – приема и передачи. При этом два канала приема и передачи будут работать независимо (т.е. принимаемые данные никоим образом не влияют на работу передающего устройства).

Если настроен на симплексный режим только один из каналов – приема или передачи, то работа второго канала блокируется. Т.о. при работе в симплексном режиме канала приема передатчик выдает в канал низкие уровни сигналов DOUT и SOUT. При работе в симплексном режиме только канала передачи работа приемника автоматически запрещается.

### **12.6.9 Маркеры времени**

Маркеры времени - системная функция стандарта SpaceWire. Они предназначены для синхронизации системных часов взаимодействующих систем.

При передаче данных маркеры времени имеют наивысший приоритет. Маркер времени записывается в регистр TX\_CODE. Этот же регистр используется и для передачи в сеть кодов распределенных прерываний и кодов подтверждения. После записи DS-макроячейка дожидается окончания передачи символа данных или служебного символа и начинает передачу маркера времени, после окончания передачи маркера времени продолжается передача потока данных. Для того, чтобы не произошло утраты управляющего символа в результате перезаписи его в регистре TX\_CODE следующим управляющим символом до передачи в сеть необходимо программно отслеживать значение бита [17] (FL\_CONTROL) регистра состояния. Если этот бит установлен в 0, то SWIC готов к передаче следующего управляющего символа. Если в момент записи в регистр TX\_CODE нового значения этот бит был установлен в 1, то существует вероятность того, что предыдущий управляющий код не будет передан в сеть.

В канале приема маркер времени выделяется из потока данных и при безошибочном приеме заносится в регистр RX\_CODE (разряды 7 - 0) с выставлением соответствующего прерывания, если маркер времени является корректным. Корректным признается маркер времени на 1 больше, чем предыдущий, если предыдущий маркер времени имел значение меньше 63. Если предыдущий маркер времени имел значение 63, то следующий корректный маркер времени должен иметь значение 0. Если маркер времени не является корректным, то его значение так же заносится в соответствующие разряды регистра RX\_CODE, однако, прерывание для процессора в данном случае не устанавливается. В начале работы устройства или после сброса маркер времени со значением 1 рассматривается как корректный.



### **12.6.10 Коды распределенных прерываний**

Коды распределенных прерываний являются расширением стандарта SpaceWire. Механизм передачи кодов распределенных прерываний в сеть аналогичен механизму передачи маркеров времени.

При приеме кода распределенного прерывания из сети выполняются следующие действия.

Если соответствующий коду распределенного прерывания разряд регистра ISR установлен в 1, то данное прерывание игнорируется (никаких действий не выполняется). Если соответствующий разряд регистра установлен в 0, то в него записывается 1 и код распределенного прерывания записывается в разряды [15:8] регистра RX\_CODE. В этом случае устанавливается прерывание.

### **12.6.11 Коды подтверждения распределенных прерываний**

Коды подтверждения распределенных прерываний являются расширением стандарта SpaceWire. Механизм передачи кодов подтверждения в сеть аналогичен механизму передачи маркеров времени.

При приеме кода подтверждения прерывания из сети выполняются следующие действия. Если соответствующий коду подтверждения разряд регистра ISR установлен в 0, то данный код игнорируется (никаких действий не выполняется). Если соответствующий разряд регистра установлен в 1, то в него записывается 0 и код записывается в разряды [23:16] регистра RX\_CODE. В этом случае устанавливается прерывание.

### **12.6.12 Установка скорости передачи данных**

Управление скоростью передачи осуществляется посредством регистра TX\_SPEED.

Если не установлен режим автоматического контроля скорости (разряд AUTO\_SPEED регистра управления MODE\_CR), то установка скорости передачи осуществляется путем записи коэффициента скорости в разряды 9:0 регистра TX\_SPEED. Этот коэффициент напрямую передается в TX\_PLL. До установки соединения в эти разряды должен быть записан коэффициент, соответствующий скорости передачи 10Мбит/с. После установки соединения в эти разряды регистра могут быть записаны другие значения (соответствующие скорости передачи от 2 до 400МГц, в соответствии со стандартом SpaceWire). Если происходит разрыв соединения, то в этот регистр снова необходимо записать коэффициент, соответствующий 10 Мбит/с.

Если установлен режим автоматического контроля скорости, то до установки соединения на TX\_PLL подается коэффициент TX\_SPEED\_10 из разрядов 19:10 регистра TX\_SPEED. Он должен соответствовать 10Мбит/с. После установки соединения на TX\_PLL будет подаваться коэффициент из разрядов 9:0 регистра TX\_SPEED. В эти разряды регистра могут быть записаны значения соответствующие скорости передачи от 2 до 400МГц. При разрыве соединения переход на коэффициент TX\_SPEED\_10 выполняется автоматически, при повторной установке соединения переход на TX\_SPEED так же выполняется автоматически.



### **12.6.13 Установление соединения**

Для разрешения процесса установки соединения необходимо записать лог "0" в разряд LinkDisabled и "1" в разряд LinkStart регистра режима работы MODE\_CR – для запуска канала, WORK\_TYPE = "1".

Критерием успешного установления соединения является прохождение прерывания INT\_LINK и отсутствие прерывания INT\_ERR.

После обнаружения прерывания INT\_LINK, необходимо считать регистр STATUS и проверить биты DC\_ERR, P\_ERR, ESC\_ERR, CREDIT\_ERR на равенство «0». Бит CONNECTED должен быть равен «1». При выполнении этих условий - соединение с удаленной системой установлено.

Для активации функции автоматического восстановления соединения после обрыва связи дополнительно в разряд AutoStart записывается «1». В этом случае после рассоединения из-за ошибок будет выставлено прерывание INT\_ERR и система будет производить повторное установление соединения. Однако следует учитывать что повторное соединение на скорости выше 10 Мбит/с не предусмотрено стандартом SpaceWire, вследствие этого при обнаружении рассоединения необходимо снова установить скорость передачи равной 10 Мбит/с.

### **12.6.14 Определение скорости приема данных**

Оценка скорости приема выполняется при разрешенной работе канала и установленном соединении. Скорость приема данных отображается в регистре RX\_SPEED[9:0]. После установления соединения скорость должна составлять  $10 \pm 1$  Мбит/с при этом регистр RX\_SPEED[9:0] будет равен  $0x0000000A \pm 1$  МЗР. Разряды регистра с 8 по 31 не используются и при чтении содержат ноль.

## 13. КОНТРОЛЛЕР ETHERNET MAC 10/100

### 13.1 Введение

#### 13.1.1 Назначение

Контроллер Ethernet MAC 10/100 (EMAC) предназначен для использования в качестве порта Ethernet для обмена данными через приемопередатчик РНУ в сети Ethernet. Контроллер Ethernet MAC поддерживает обмен данными в сети Ethernet с быстродействием 10 Мбит/с, либо 100 Мбит/с.

#### 13.1.2 Основные характеристики

Контроллер Ethernet MAC 10/100 имеет следующие основные характеристики:

- Соответствует стандарту Ethernet IEEE Std 802.3-2005;
- Поддерживает полудуплексный (CSMA/CD), дуплексный режимы работы;
- В состав контроллера входит буферное FIFO передаваемых данных размером 0,5К 64-разрядных слов или 4К байт;
- В состав контроллера входит буферное FIFO принятых данных размером 0,5К 64-разрядных слов или 4К байт;
- Передача данных из памяти в FIFO передаваемых данных обеспечивается каналом DMA (передача данных из памяти осуществляется с точностью до байта);
- Передача данных из FIFO принятых данных в память обеспечивается каналом DMA (передача данных в память осуществляется с точностью до байта);
- Передаваемый кадр MAC целиком помещается в FIFO, поэтому при возникновении коллизии повторная передача кадра будет выполняться из FIFO;
- Поддерживает режим зацикливания тракта приема данных на тракт передачи, в этом режиме контроллер принимает только передаваемые от него данные;
- Поддерживает различные режимы фильтрации принимаемых кадров MAC по адресу назначения: распознавание уникального адреса MAC, широковещательный адрес, распознавание группового адреса по маске либо по хэш-таблице;
- Поддерживает различные режимы отбрасывания принятых кадров MAC, при проверке которых были обнаружены ошибки: слишком короткий кадр, слишком длинный кадр, кадр с ошибкой в контрольной сумме, кадр с ошибкой длины;
- В состав контроллера входит FIFO статусов принятых кадров MAC размером 64 слова статуса.

#### 13.1.3 Особенности использования

При использовании контроллера Ethernet необходимо соблюдать следующие условия:

- при работе порта Ethernet в режиме 100 Мбит/с частота CPU должна быть больше 100 МГц;
- при работе порта Ethernet в режиме 10 Мбит/с частота CPU должна быть больше 10 МГц.

## 13.2 Функциональное описание

### 13.2.1 Структурная схема

Контроллер Ethernet MAC 10/100 включает:

- Блок управления и состояния;
- Контроллер DMA;
- Блок передачи кадров – TransmitFrame;
- Блок приема кадров – ReceiveFrame;
- Порт управления PHY – MD\_PORT.

На Рисунок 13.1 приведена структурная схема контроллера MAC 10/100.

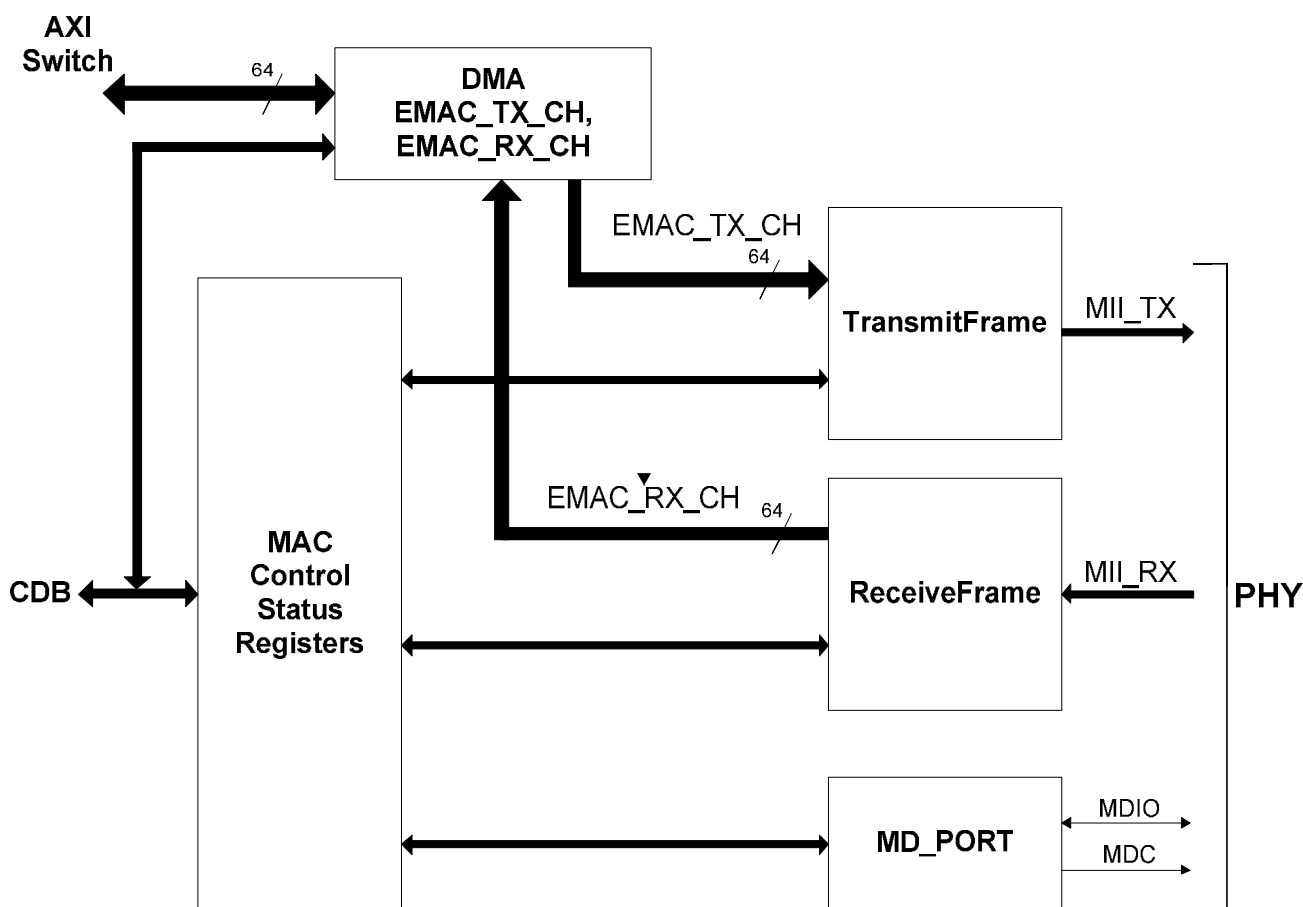


Рисунок 13.1. Структурная схема контроллера MAC 10/100

Блок управления и состояния содержит регистры управления и состояния контроллера MAC.

Каналы DMA EMAC\_TX\_CH, EMAC\_RX\_CH обеспечивают обмен данными между FIFO передаваемых/принятых данных и памятью (внешней или внутренней).

Блок передачи кадров – TransmitFrame – выполняет передачу кадров MAC по шине MII. В состав блока передачи кадров входит TX\_FIFO размером 4К байт, блок вычисления временной задержки перед повторной передачей кадра при обнаружении коллизии – BACKOFF, а также блок вычисления контрольной суммы передаваемого кадра – CALC\_CRC32.

На Рисунок 13.2 приведена структурная схема блока передачи кадров.

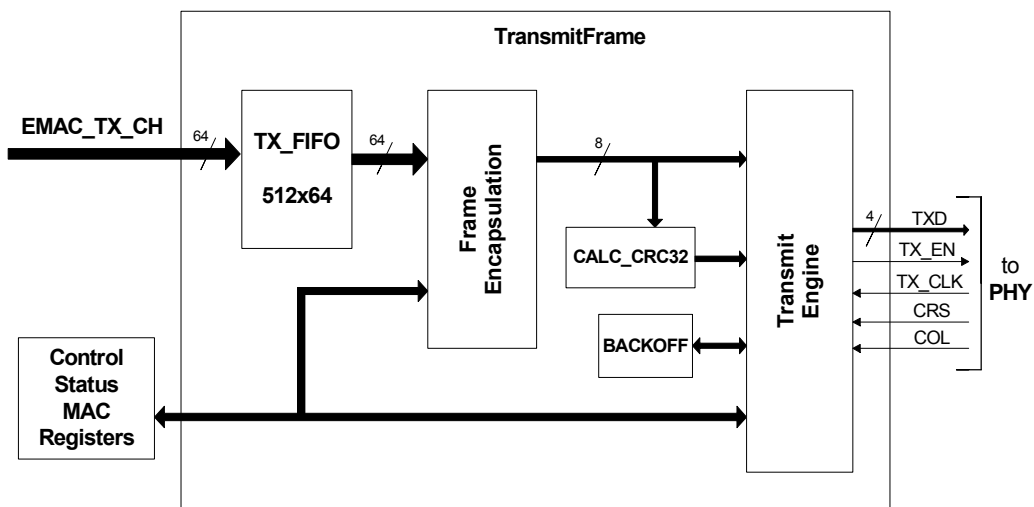


Рисунок 13.2. Структурная схема блока передачи кадров.

Блок приема кадров – ReceiveFrame – выполняет прием кадров MAC по шине MII. В состав блока приема кадров входит RX\_FIFO размером 4К байт, блок распознавания адреса назначения принятого кадра MAC – DADDR\_CHECK, блок вычисления и проверки контрольной суммы принятого кадра – CRC32\_CHECK, а также FIFO статусов принятых кадров размером 64 слова статуса.

На Рисунок 13.3 приведена структурная схема блока приема кадров.

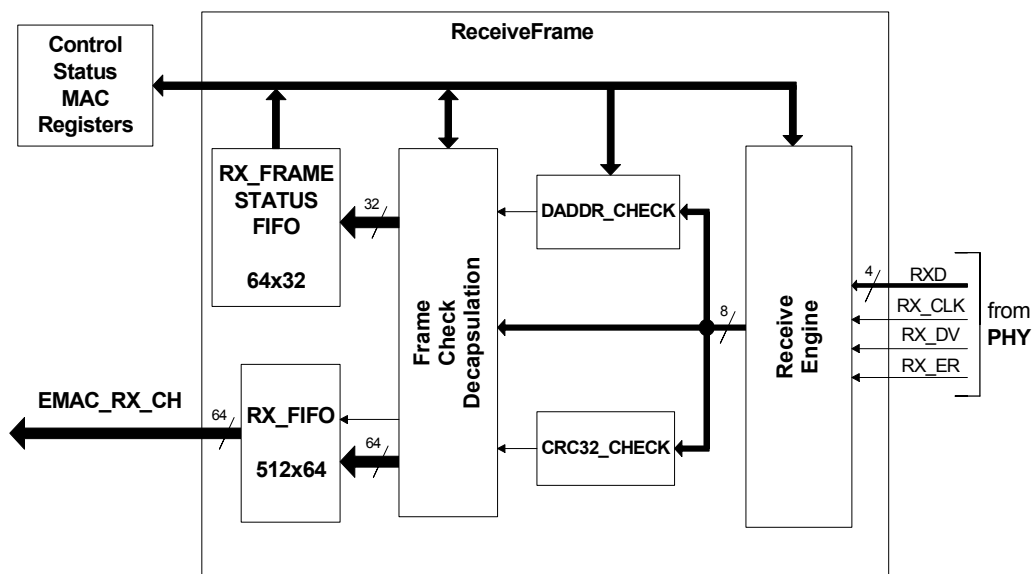


Рисунок 13.3. Структурная схема блока приема кадров.

Порт управления PHY – MD\_PORT – выполняет обмен управляющими и статусными данными с приемопередатчиком PHY.

## 13.3 Программная модель

### 13.3.1 Программирование контроллера Ethernet MAC 10/100

#### 13.3.1.1 Контроллер прямого доступа (DMA)

DMA обеспечивает:

- по каналу EMAC\_TX\_CN передачу данных из памяти (внешней или внутренней) в TX\_FIFO;
- по каналу EMAC\_RX\_CN передачу данных из RX\_FIFO в память (внешнюю или внутреннюю).

Для передачи данных по каналу EMAC\_TX\_CN необходимо в регистре MAC\_CONTROL установить EN\_TX\_DMA = 1, чтобы разрешить работу TX\_FIFO с каналом DMA.

#### 13.3.1.2 Порт управления PHY – MD\_PORT

Порт управления PHY предназначен для обмена управляющими и статусными данными с приемопередатчиком PHY.

Обмен данными с приемопередатчиком PHY осуществляется по последовательному двухпроводному интерфейсу управления MD. Интерфейс управления MD состоит из двунаправленного сигнала для обмена данными MDIO и сигнала тактовой частоты MDC.

Тактовая частота MDC интерфейса управления MD формируется портом управления PHY и передается в приемопередатчик PHY для тактирования данных, передаваемых по сигналу MDIO. Для формирования тактовой частоты MDC используется делитель системной частоты HCLK, входящий в состав порта управления PHY.

Коэффициент деления системной частоты при формировании тактовой частоты MDC задается в разрядах регистра MD\_MODE<7:0> = MDC\_Divider. Для корректной работы порта управления PHY значение коэффициента деления системной частоты должно быть четным и не нулевым. Для корректного обмена данными по интерфейсу управления MD тактовая частота MDC не должна превышать 2,5 МГц.

Порт управления PHY выполняет следующие операции:

- запись в регистр приемопередатчика PHY;
- чтение регистра приемопередатчика PHY.

Для того чтобы запустить операцию на выполнение необходимо установить код операции в разрядах регистра управления порта – MD\_CONTROL<31:30> = MD\_OP. После завершения выполнения операции код операции MD\_OP автоматически сбрасывается.

Адрес приемопередатчика PHY, с которым выполняется обмен данными, задается в разрядах регистра управления порта MD\_CONTROL<28:24> = PHY\_ADDR.

Адрес регистра приемопередатчика PHY, в который выполняется запись, либо из которого выполняется чтение данных, задается в разрядах регистра управления порта MD\_CONTROL<20:16> = PHYREG\_ADDR.

При выполнении операции записи в регистр приемопередатчика PHY 16-разрядные данные для записи должны быть установлены в разрядах регистра управления порта MD\_CONTROL<15:0> = WR\_DT.

После завершения выполнения операции чтения регистра приемопередатчика PHY прочтенные 16-разрядные данные сохраняются в разрядах регистра статуса порта MD\_STATUS <15:0> = RD\_DT.

После задания кода операции MD\_OP порт начинает выполнять операцию и считается занятым, то есть недоступным для выполнения новой операции.

Для отслеживания состояния порта используется бит статусного регистра порта MD\_STATUS<29> = MD\_BUSY. Во время выполнения операции устанавливается бит занятости порта MD\_BUSY, а после завершения выполнения операции бит MD\_BUSY сбрасывается.

Обмен данными с приемопередатчиком PHY по интерфейсу управления MD выполняется в соответствии с форматом кадра управления. Формат кадра управления представлен в Таблица 13.1.

**Таблица 13.1. Формат кадра управления.**

Число бит	Название поля	Поле кадра управления	Значение при операции записи	Значение при операции чтения
32	Преамбула	<b>PRE</b>	1111...1111	1111...1111
2	Начало кадра	<b>ST</b>	01	01
2	Код операции	<b>OP</b>	01	10
5	Адрес PHY	<b>PHYAD</b>	PHY_ADDR	PHY_ADDR
5	Адрес регистра	<b>REGAD</b>	PHYREG_ADDR	PHYREG_ADDR
2	Разворот (turnaround)	<b>TA</b>	10	Z0
16	Данные	<b>DATA</b>	WR_DT	RD_DT

Таким образом, при выполнении операции портом по интерфейсу MD последовательно передаются 64 бита кадра управления в течение 64 тактов частоты MDC. То есть временная задержка на выполнение операции портом управления PHY составляет 64 такта частоты MDC.

По завершении выполнения операции порт выставляет соответствующий флаг в разрядах регистра статуса порта MD\_STATUS<31:30> = MD\_OP\_END. Флаги завершения выполнения операции MD\_OP\_END доступны для записи и могут быть сброшены записью нулей в соответствующие биты регистра MD\_STATUS.

Во время выполнения операции регистр управления порта MD\_CONTROL и разряды регистра статуса порта MD\_STATUS<31:30> = MD\_OP\_END не доступны для записи.

Флаги завершения выполнения операции MD\_OP\_END являются запросом на прерывание от порта управления PHY. Запрос на прерывание от порта управления PHY маскируется.

В бите MD\_CONTROL<29> = MD\_MASK устанавливается маска запроса на прерывание от порта управления PHY.

Бит MD\_MODE<8> = RST\_MD предназначен для программного сброса порта управления PHY, а также регистров MD\_MODE, MD\_CONTROL, MD\_STATUS. После установления бит RST\_MD автоматически сбрасывается.

### 13.3.1.3 Блок передачи кадров *TransmitFrame*

Перед началом работы необходимо сконфигурировать блок передачи кадров – в регистре управления MAC установить бит `MAC_CONTROL<0> = FULLD = 0/1` для задания полудуплексного/дуплексного режима работы контроллера. Также для разрешения работы блока передачи кадров должен быть установлен бит `MAC_CONTROL<2> = EN_TX = 1`.

Формирование кадра при передаче может выполняться в одном из двух режимов:

- Передаваемый кадр формируется в блоке передачи кадров;
- В блок передачи кадров передается уже сформированный кадр.

На Рисунок 13.4 приведен формат кадра MAC.

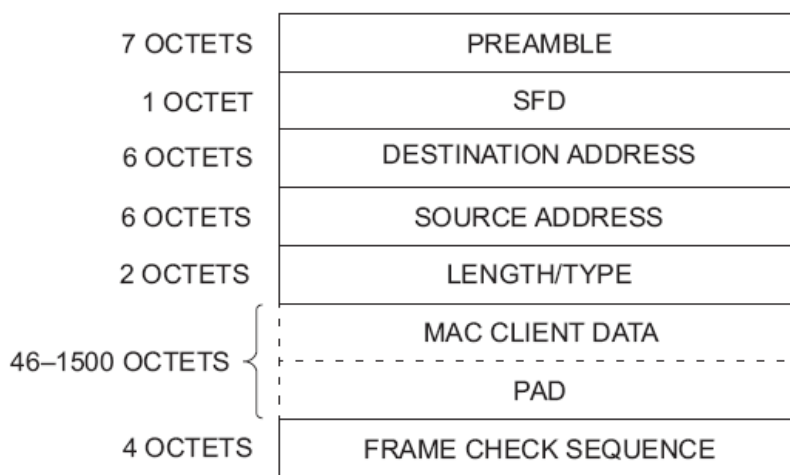


Рисунок 13.4. Формат кадра MAC.

При передаче кадра блок передачи кадров автоматически вставляет в начале каждого передаваемого кадра 8 байт полей `<PREAMBLE>` и `<SFD>`. Каждый байт поля `<PREAMBLE>` имеет значение `0x55`, а байт поля `<SFD>` имеет значение `0xD5`.

#### 13.3.1.3.1 Режим формирования передаваемого кадра в блоке передачи кадров.

По умолчанию кадр формируется в блоке передачи кадров, при этом бит `TX_FRAME_CONTROL<14> = DisEncapFR = 0`, то есть разрешен режим формирования кадра в блоке передачи кадров.

В этом режиме для формирования передаваемого кадра необходимо установить регистры `MAC_ADDR_L`, `MAC_ADDR_H`, `DADDR_L`, `DADDR_H`, `TYPE`, `FCS_CLIENT`, значение которых задает значение полей передаваемого кадра:

<code>{MAC_ADDR_H, MAC_ADDR_L}</code>	=> поле <code>&lt;SOURCE ADDRESS&gt;</code> ;
<code>{DADDR_H, DADDR_L}</code>	=> поле <code>&lt;DESTINATION ADDRESS&gt;</code> ;
<code>TYPE</code>	=> поле <code>&lt;LENGTH/TYPE&gt;</code> , используемое как поле <code>&lt;TYPE&gt;</code> ;
<code>FCS_CLIENT</code>	=> поле <code>&lt;FCS&gt;</code> – уже вычисленная клиентом

MAC контрольная сумма CRC32;

Разряды регистра

$\text{TX\_FRAME\_CONTROL}\langle 11:0 \rangle = \text{LENGTH} \Rightarrow$  задают значение поля  $\langle \text{LENGTH}/\text{TYPE} \rangle$ , используемое как поле  $\langle \text{LENGTH} \rangle$ ;

Содержание поля  $\langle \text{DATA} \rangle$  передается по DMA-каналу на запись  $\text{DMA\_EMAC\_CH1}$  в передающее FIFO –  $\text{TX\_FIFO}$  – в виде последовательности 64-разрядных слов. Каждое 64-разрядное слово состоит из 8 байт поля  $\langle \text{DATA} \rangle$ , начиная с байта, который должен быть передан первым, и заканчивая байтом, который должен быть передан последним:

63	56	55	48	47	40	39	32	31	24	23	16	15	8	7	0
byte <sub>(n+7)</sub>	byte <sub>(n+6)</sub>	byte <sub>(n+5)</sub>	byte <sub>(n+4)</sub>	byte <sub>(n+3)</sub>	byte <sub>(n+2)</sub>	byte <sub>(n+1)</sub>	byte <sub>(n)</sub>								

$\xrightarrow{\hspace{10em}}$   
 Байты передаются, начиная с младшего

В случае если последнее 64-разрядное слово поля  $\langle \text{DATA} \rangle$  содержит меньше чем 8 байт для передачи, то передаваемые байты помещаются в соответствующие младшие разряды слова: 1 байт – в разряды  $\langle 7:0 \rangle$ , 2 байта – в разряды  $\langle 15:0 \rangle$ , 3 байта – в разряды  $\langle 23:0 \rangle$ , 4 байта – в разряды  $\langle 31:0 \rangle$ , 5 байт – в разряды  $\langle 39:0 \rangle$ , 6 байт – в разряды  $\langle 47:0 \rangle$ , 7 байт – в разряды  $\langle 55:0 \rangle$ . Оставшиеся старшие разряды слова заполняются произвольными (нулевыми) значениями. Признаком того, что все данные кадра переданы в  $\text{TX\_FIFO}$  и, что можно аппаратно дополнить 64-разрядную строку нулями, является запись команды на передачу кадра  $\text{TX\_REQ}$ .

Бит регистра  $\text{TX\_FRAME\_CONTROL}\langle 12 \rangle = \text{TYPE\_EN}$  – задает в каком качестве используется поле  $\langle \text{LENGTH}/\text{TYPE} \rangle$  в передаваемом кадре.

Если бит  $\text{TYPE\_EN}=0$ , то в кадре используется поле  $\langle \text{LENGTH} \rangle$  и его значение определяется разрядами  $\text{TX\_FRAME\_CONTROL}\langle 11:0 \rangle$ .

Если бит  $\text{TYPE\_EN}=1$ , то в кадре используется поле  $\langle \text{TYPE} \rangle$  и его значение определяется значением регистра  $\text{TYPE}$ .

Независимо от значения бита  $\text{TYPE\_EN}$  необходимо установить разряды регистра  $\text{TX\_FRAME\_CONTROL}\langle 11:0 \rangle = \text{LENGTH}$  для задания числа байт в поле  $\langle \text{DATA} \rangle$  передаваемого кадра – этот параметр используется блоком передачи кадров при передаче кадра. Значение  $\text{LENGTH}$  должно быть не нулевым.

Бит регистра  $\text{TX\_FRAME\_CONTROL}\langle 13 \rangle = \text{FCS\_CLT\_EN}$  – задает источник формирования поля  $\langle \text{FCS} \rangle$ .

Если бит  $\text{FCS\_CLT\_EN}=0$ , то значение поля  $\langle \text{FCS} \rangle$  – контрольная сумма CRC32 передаваемого кадра – вычисляется в блоке  $\text{CALC\_CRC32}$  при передаче кадра.

Если бит  $\text{FCS\_CLT\_EN}=1$ , то значение поля  $\langle \text{FCS} \rangle$  – уже вычисленная клиентом MAC контрольная сумма CRC32, заданная в регистре  $\text{FCS\_CLIENT}$ .

Бит регистра  $\text{TX\_FRAME\_CONTROL}\langle 15 \rangle = \text{Dis\_PAD}$  – запрещает/разрешает автоматическое добавление в кадр поля  $\langle \text{PAD} \rangle$ , в случае когда число байт в поле  $\langle \text{DATA} \rangle$  меньше 46 байт (минимальный размер поля  $\langle \text{DATA} \rangle$  в соответствии со стандартом Ethernet).

Если бит  $\text{Dis\_PAD} = 0$ , тогда:

если бит  $\text{TX\_FRAME\_CONTROL}\langle 13 \rangle = \text{FCS\_CLT\_EN} = 0$ ,  
 а значение  $\text{TX\_FRAME\_CONTROL}\langle 11:0 \rangle = \text{LENGTH} < 46$  байт, }  $\Rightarrow$



=> то в кадр после поля <DATA> добавляется поле <PAD>.

Число байт в поле <PAD> определяется как разность  $(46 - LENGTH)$ .

Каждый байт поля <PAD> имеет значение 0x99.

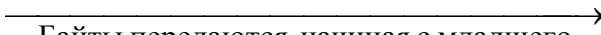
Если бит  $Dis\_PAD = 1$ , либо если бит  $TX\_FRAME\_CONTROL<13> = FCS\_CLT\_EN = 1$ , то, несмотря на число байт в поле <DATA>, автоматического добавления поля <PAD> в кадр выполняться не будет.

**13.3.1.3.2 Режим передачи, при котором в блок передачи кадров передается уже сформированный кадр.**

Для отключения режима формирования кадра в блоке передачи кадров необходимо установить бит  $TX\_FRAME\_CONTROL<14> = DisEncapFR = 1$ . В этом случае готовый для передачи сформированный кадр должен быть передан в блок передачи кадров.

Содержание кадра передается по DMA-каналу на запись  $DMA\_EMAC\_CH1$  в передающее FIFO –  $TX\_FIFO$  – в виде последовательности 64-разрядных слов. Каждое 64-разрядное слово состоит из 8 байт кадра, начиная с байта, который должен быть передан первым и заканчивая байтом, который должен быть передан последним:

63	56	55	48	47	40	39	32	31	24	23	16	15	8	7	0
byte <sub>(n+7)</sub>	byte <sub>(n+6)</sub>	byte <sub>(n+5)</sub>	byte <sub>(n+4)</sub>	byte <sub>(n+3)</sub>	byte <sub>(n+2)</sub>	byte <sub>(n+1)</sub>	byte <sub>(n)</sub>								


 Байты передаются, начиная с младшего

В случае если последнее 64-разрядное слово кадра содержит меньше чем 8 байт для передачи, то передаваемые байты помещаются в соответствующие младшие разряды слова: 1 байт – в разряды <7:0>, 2 байта – в разряды <15:0>, 3 байта – в разряды <23:0>, 4 байта – в разряды <31:0>, 5 байт – в разряды <39:0>, 6 байт – в разряды <47:0>, 7 байт – в разряды <55:0>. Оставшиеся старшие разряды слова аппаратно заполняются произвольными (нулевыми) значениями. Признаком того, что все данные кадра переданы в  $TX\_FIFO$  и, что можно аппаратно дополнить 64-разрядную строку нулями, является запись команды на передачу кадра  $TX\_REQ$ .

Кадр, переданный в  $TX\_FIFO$ , должен быть сформирован в соответствии с форматом кадра MAC, приведенным на Рисунок 13.4 и состоять из полей: <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE>, <DATA>. Таким образом, сначала в  $TX\_FIFO$  должно быть передано содержание поля <DESTINATION ADDRESS>, затем содержание поля <SOURCE ADDRESS>, далее содержание поля <LENGTH/TYPE> (старший байт первым), а затем содержание поля <DATA>. Также кадр, переданный в  $TX\_FIFO$ , может содержать уже вычисленное значение поля <FCS>. Тогда содержание поля <FCS> должно быть передано сразу же вслед за содержанием поля <DATA>. При этом при компоновке байт полей кадра в 64-разрядные слова не должно быть пустых байт на границах полей. Таким образом, кадр после разбиения на 64-разрядные слова должен иметь следующую структуру (когда в состав кадра не входит поле <FCS>), представленную в Таблица 13.2.

**Таблица 13.2. Структура кадра MAC, не включающего поле <FCS>.**

Word	63	48	47	32	31	0
0	SOURCE ADDRESS<15:0>		DESTINATION ADDRESS<47:32>		DESTINATION ADDRESS<31:0>	
1	DATA<byte1, byte0>		LENGTH/ TYPE<7:0>	LENGTH/ TYPE<15:8>	SOURCE ADDRESS<47:16>	
2	DATA<byte9, byte8, byte7, byte6>			DATA<byte5, byte4, byte3, byte2>		
...	...					
N	DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> >				DATA<byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> , byte <sub>(LEN-7)</sub> , byte <sub>(LEN-8)</sub> >	
либо: N	0x00, DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> >		DATA<byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> , byte <sub>(LEN-7)</sub> >			
либо: N	0x00, 0x00, DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> >		DATA<byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> >			
либо: N	0x00, 0x00, 0x00, DATA<byte <sub>(LEN-1)</sub> >		DATA<byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> >			
либо: N	0x00, 0x00, 0x00, 0x00		DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> >			
либо: N	0x00, 0x00, 0x00, 0x00		0x00, DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> >			
либо: N	0x00, 0x00, 0x00, 0x00		0x00, 0x00, DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> >			
либо: N	0x00, 0x00, 0x00, 0x00		0x00, 0x00, 0x00, DATA<byte <sub>(LEN-1)</sub> >			

Где LEN – число байт в поле <DATA>: byte0, byte1, ..., byte<sub>(LEN-1)</sub>.

В случае, когда кадр, переданный в TX\_FIFO, содержит уже вычисленное значение поля <FCS>, то кадр имеет следующую структуру, представленную в Таблица 13.3:

**Таблица 13.3. Структура кадра MAC, включающего поле <FCS>.**

Word	63	48	47	32	31	0
0	SOURCE ADDRESS<15:0>		DESTINATION ADDRESS<47:32>		DESTINATION ADDRESS<31:0>	
1	DATA<byte1, byte0>		LENGTH/ TYPE<7:0>	LENGTH/ TYPE<15:8>	SOURCE ADDRESS<47:16>	
2	DATA<byte9, byte8, byte7, byte6>				DATA<byte5, byte4, byte3, byte2>	
...	...					
N-1	DATA<byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> , byte <sub>(LEN-7)</sub> , byte <sub>(LEN-8)</sub> >				DATA<byte <sub>(LEN-9)</sub> , byte <sub>(LEN-10)</sub> , byte <sub>(LEN-11)</sub> , byte <sub>(LEN-12)</sub> >	
N	FCS<31:0>				DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> >	
либо: N-1	DATA<byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> , byte <sub>(LEN-7)</sub> >				DATA<byte <sub>(LEN-8)</sub> , byte <sub>(LEN-9)</sub> , byte <sub>(LEN-10)</sub> , byte <sub>(LEN-11)</sub> >	
N	0x00, FCS<31:8>				FCS<7:0>, DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> >	
либо: N-1	DATA<byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> >				DATA<byte <sub>(LEN-7)</sub> , byte <sub>(LEN-8)</sub> , byte <sub>(LEN-9)</sub> , byte <sub>(LEN-10)</sub> >	
N	0x00, 0x00, FCS<31:16>				FCS<15:0>, DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> >	
либо: N-1	DATA<byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> >				DATA<byte <sub>(LEN-6)</sub> , byte <sub>(LEN-7)</sub> , byte <sub>(LEN-8)</sub> , byte <sub>(LEN-9)</sub> >	
N	0x00, 0x00, 0x00, FCS<31:24>				FCS<23:0>, DATA<byte <sub>(LEN-1)</sub> >	
либо: N-1	DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> >				DATA<byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> , byte <sub>(LEN-7)</sub> , byte <sub>(LEN-8)</sub> >	
N	0x00, 0x00, 0x00, 0x00				FCS<31:0>	
либо: N-1	FCS<7:0>, DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> >				DATA<byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> , byte <sub>(LEN-7)</sub> >	
N	0x00, 0x00, 0x00, 0x00				0x00, FCS<31:8>	
либо: N-1	FCS<15:0>, DATA<byte <sub>(LEN-1)</sub> , byte <sub>(LEN-2)</sub> >				DATA<byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> , byte <sub>(LEN-6)</sub> >	
N	0x00, 0x00, 0x00, 0x00				0x00, 0x00, FCS<31:16>	
либо: N-1	FCS<23:0>, DATA<byte <sub>(LEN-1)</sub> >				DATA<byte <sub>(LEN-2)</sub> , byte <sub>(LEN-3)</sub> , byte <sub>(LEN-4)</sub> , byte <sub>(LEN-5)</sub> >	
N	0x00, 0x00, 0x00, 0x00				0x00, 0x00, 0x00, FCS<31:24>	

Бит регистра TX\_FRAME\_CONTROL<13> = FCS\_CLT\_EN – задает источник формирования поля <FCS>.

Если бит FCS\_CLT\_EN=0, то значение поля <FCS> – контрольная сумма CRC32 передаваемого кадра – вычисляется в блоке CALC\_CRC32 при передаче кадра.

При этом кадр, переданный в TX\_FIFO, содержит следующие поля: <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE>, <DATA>.

Если бит FCS\_CLT\_EN=1, то значение поля <FCS> – уже вычисленная клиентом MAC контрольная сумма CRC32, переданная вместе с остальными полями кадра в TX\_FIFO.

При этом кадр, переданный в TX\_FIFO, содержит поля: <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE>, <DATA>, <FCS>.

Также должны быть установлены разряды регистра TX\_FRAME\_CONTROL<11:0> = LENGTH для задания числа байт кадра, переданного в TX\_FIFO, – этот параметр используется блоком передачи кадров при передаче кадра. Значение LENGTH должно быть не нулевым.

В случае, когда FCS\_CLT\_EN=0, значение LENGTH соответствует числу байт полей <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE> и <DATA>, то есть (12 байт + число байт поля <DATA>).

В случае, когда FCS\_CLT\_EN=1, значение LENGTH соответствует числу байт всех полей кадра: <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE>, <DATA> и <FCS>, то есть (16 байт + число байт поля <DATA>).

Бит регистра TX\_FRAME\_CONTROL<15> = Dis\_PAD – запрещает/разрешает автоматическое добавление в кадр поля <PAD>, в случае когда число байт в кадре меньше 64 байт (минимальный размер кадра в соответствии со стандартом Ethernet).

Если бит Dis\_PAD = 0, тогда:

если бит TX\_FRAME\_CONTROL<13> = FCS\_CLT\_EN = 0,  
а значение TX\_FRAME\_CONTROL<11:0> = LENGTH < 60 байт } =>

(4 байта поля <FCS> вычисляются контроллером при передаче),

=> то во время передачи кадра перед передачей поля <FCS> передается поле <PAD>.

Число байт в поле <PAD> определяется как разность (60 – LENGTH).

Каждый байт поля <PAD> имеет значение 0x99.

Если бит Dis\_PAD = 1, либо если бит TX\_FRAME\_CONTROL<13> = FCS\_CLT\_EN = 1, то, несмотря на число байт в кадре, автоматического добавления поля <PAD> при передаче кадра выполняться не будет.

### 13.3.1.3.3 Передача кадра.

Для того чтобы запустить передачу кадра необходимо установить в регистре управления передачи кадра бит запроса на передачу кадра, то есть TX\_FRAME\_CONTROL<16> = TX\_REQ = 1.

Перед тем как будет установлен бит запроса на передачу кадра, в блок передачи кадров должны быть переданы данные, необходимые для формирования кадра.

В случае, когда разрешен режим формирования кадра в блоке передачи кадров, тогда необходимо установить регистры MAC\_ADDR\_L, MAC\_ADDR\_H, DADDR\_L, DADDR\_H, TYPE, FCS\_CLIENT, TX\_FRAME\_CONTROL, а также содержание поля <DATA> должно быть полностью передано в TX\_FIFO.

В случае, когда в блок передачи кадров передается уже сформированный кадр, тогда необходимо установить регистр TX\_FRAME\_CONTROL, а содержание кадра должно быть полностью передано в TX\_FIFO.

Перед тем как начать передавать данные в TX\_FIFO должна быть разрешена работа передающего TX\_FIFO с DMA-каналом на запись DMA\_EMAC\_CH1.

Для того чтобы разрешить работу передающего TX\_FIFO с каналом DMA\_EMAC\_CH1 необходимо установить в регистре управления MAC бит MAC\_CONTROL<1> = EN\_TX\_DMA = 1.

Число 64-разрядных слов в передающем FIFO – TX\_FIFO – отображается в разрядах регистра статуса STATUS\_TX<26:16> = TXW (TXW содержит информацию о количестве данных в TX\_FIFO с точностью до байта, но в регистре статуса отображена информация с точностью до 64-разрядного слова округленного в большую сторону).

Также, перед тем как будет установлен запрос на передачу кадра, должен быть сконфигурирован регистр IFS и режима обработки коллизий – IFS\_COLL\_MODE.

После выставления бита запроса на передачу кадра TX\_REQ = 1 в связи с синхронизацией системной частоты HCLK и частоты передачи TX\_CLK блоку передачи кадров требуется временная задержка, прежде чем он начнет обрабатывать запрос на передачу кадра. Для отслеживания состояния блока передачи кадров используется бит статусного регистра STATUS\_TX<0> = ONTX\_REQ. Как только блок передачи кадров начинает обработку запроса на передачу кадра устанавливается бит ONTX\_REQ и продолжает стоять в течение обработки запроса на передачу кадра. По завершении обработки запроса на передачу кадра бит ONTX\_REQ сбрасывается. Сразу после начала обработки запроса на передачу кадра блок передачи кадров буферизует содержимое регистров MAC\_ADDR\_L, MAC\_ADDR\_H, DADDR\_L, DADDR\_H, TYPE, FCS\_CLIENT, TX\_FRAME\_CONTROL, IFS\_COLL\_MODE. Таким образом, после того как был установлен бит запроса на передачу кадра TX\_REQ = 1 необходимо дождаться выставления бита ONTX\_REQ = 1 в статусном регистре, и после этого все регистры блока передачи кадров могут быть переустановлены для передачи следующего кадра. В передающее TX\_FIFO также может быть передано содержимое следующего кадра. В течении времени после того как был установлен бит TX\_REQ, но еще не выставился бит ONTX\_REQ попытка записи в регистры блока передачи кадров игнорируется.

После выставления бита запроса на передачу кадра TX\_REQ = 1 – он не может быть сброшен и будет продолжать стоять в течение обработки запроса на передачу кадра. По завершении обработки запроса на передачу кадра бит TX\_REQ автоматически сбрасывается. После этого бит запроса на передачу может быть выставлен снова для передачи следующего кадра.

Если бит разрешения работы блока передачи кадров MAC\_CONTROL<2> = EN\_TX будет сброшен, после того как блок передачи кадров начал обработку запроса на передачу кадра, то, не смотря на это, обработка текущего запроса на передачу будет продолжена.

Если был установлен бит запроса на передачу кадра TX\_REQ = 1 и при этом бит разрешения работы блока передачи кадров MAC\_CONTROL<2> = EN\_TX = 0, тогда блок передачи кадров сразу же завершает обработку запроса на передачу кадра и сообщает об этом выставлением в регистре статуса бита STATUS\_TX<3> = TX\_DONE = 1. По завершении обработки запроса на передачу кадра блок передачи кадров также сообщает результат передачи кадра в разрядах регистра статуса STATUS\_TX<8:4> = TX\_REZ = 0x01 – transmitDisabled – передача не разрешена.

Если был установлен бит запроса на передачу кадра TX\_REQ = 1 и при этом число слов в передающем TX\_FIFO – TXW меньше, чем значение разрядов регистра TX\_FRAME\_CONTROL<11:0> = LENGTH, то есть TXW < LENGTH, тогда блок передачи кадров сразу же завершает обработку запроса на передачу кадра и сообщает об этом выставлением в регистре статуса бита STATUS\_TX<3> = TX\_DONE = 1. По завершении обработки запроса на передачу кадра блок передачи кадров также сообщает результат передачи кадра в разрядах регистра статуса STATUS\_TX<8:4> = TX\_REZ = 0x02 – NotEnoughDataErr – в TX\_FIFO недостаточно данных для передачи.

Если контроллер MAC работает в полудуплексном режиме (бит  $MAC\_CONTROL<0> = FULLD = 0$ ), то когда блок передачи кадров начинает обработку запроса на передачу кадра ( $ONTX\_REQ = 1$ ), то сначала он проверяет занята ли среда передачи.

Для отслеживания занятости среды передачи используется бит статусного регистра  $STATUS\_TX<2> = BUSY$ . Когда в среде передачи обнаруживается наличие несущей, это означает, что в среде идет передача от одной из передающих станций (в том числе и от контроллера MAC), тогда устанавливается бит  $BUSY$  – среда занята. Как только среда передачи освобождается, бит  $BUSY$  сбрасывается.

В случае если блок передачи кадров обнаруживает занятость среды передачи, тогда он задерживает передачу кадра и ожидает когда среда передачи освободится, то есть когда другая станция закончит свою передачу. После того, как среда передачи освобождается, блок передачи кадров, перед тем как начать передавать кадр, выдерживает временную задержку, называемую межкадровым интервалом –  $interFrameSpacing$ .

Значение межкадрового интервала  $interFrameSpacing$  задается в разрядах регистра  $IFS\_COLL\_MODE<31:24> = IFS$ . В соответствии со стандартом Ethernet межкадровый интервал  $IFS$  по умолчанию равен времени передачи 96 бит, что соответствует 24 тактам частоты передачи  $TX\_CLK$ . Значение  $IFS$  должно быть не меньше 4 тактов частоты передачи  $TX\_CLK$ .

Межкадровый интервал рассматривается в качестве двух последовательных временных интервалов: начальный интервал, равный значению ( $IFS - 8$ ), что по умолчанию соответствует первым 16 тактам  $TX\_CLK$  после начала отсчета межкадрового интервала, и заключительный интервал, который соответствует последующим 8 тактам  $TX\_CLK$ . Блок передачи кадров начинает отсчитывать межкадровый интервал после того как освобождается среда передачи, если в течение начального интервала вновь обнаруживается занятость среды передачи, то блок передачи кадров снова ждет когда освободится среда и после этого заново начинает отсчитывать межкадровый интервал. Если же в течение начального интервала среда передачи остается свободной, то блок передачи кадров затем продолжает ожидать в течение заключительного интервала, но при этом уже не отслеживая занятость среды. Таким образом, как только истечет заключительный интервал межкадрового интервала блок передачи кадров сразу же начинает передачу своего кадра в среду передачи.

Бит статусного регистра  $STATUS\_TX<1> = ONTransmit$  позволяет отслеживать состояние блока передачи кадров. Когда блок передачи кадров передает кадр в среду передачи, тогда бит  $ONTransmit$  устанавливается и продолжает стоять в течение всей передачи кадра. Как только блок передачи кадров завершает передачу кадра, бит  $ONTransmit$  сбрасывается.

Если контроллер MAC работает в дуплексном режиме (бит  $MAC\_CONTROL<0> = FULLD = 1$ ), то среда передачи всегда доступна. Таким образом, в дуплексном режиме блок передачи кадров сразу же после начала обработки запроса на передачу начинает передавать кадр. Однако, в случае выполнения последовательных передач кадров блок передачи кадров между передачами выдерживает временную задержку – межкадровый интервал –  $interFrameSpacing$ . Межкадровый интервал  $interFrameSpacing$  в соответствии со стандартом Ethernet равен времени передачи 96 бит, что соответствует 24 тактам частоты передачи  $TX\_CLK$ .

Во время передачи блок передачи кадров последовательно передает байты всех полей кадра:  $\langle DESTINATION ADDRESS \rangle$ ,  $\langle SOURCE ADDRESS \rangle$ ,  $\langle LENGTH/TYPE \rangle$ ,  $\langle DATA \rangle$ ,  $\langle FCS \rangle$ .

Если контроллер MAC работает в полудуплексном режиме (бит  $MAC\_CONTROL<0> = FULLD = 0$ ) и во время передачи кадра не было обнаружено коллизии, либо если контроллер MAC работает в дуплексном режиме (бит  $MAC\_CONTROL<0> = FULLD = 1$ ), то блок передачи кадров, передав байты последнего поля  $\langle FCS \rangle$ , завершает передачу кадра и затем за-



вершает обработку запроса на передачу кадра и сообщает об этом выставлением в регистре статуса бита STATUS\_TX<3>=TX\_DONE= 1.

По завершении обработки запроса на передачу кадра блок передачи кадров также сообщает результат передачи кадра в разрядах регистра статуса STATUS\_TX<8:4> = TX\_REZ = 0x04 – transmitOK – передача кадра успешно выполнена.

По завершении обработки запроса на передачу кадра, если передача кадра была успешно выполнена, то число слов в передающем TX\_FIFO – TXW декрементируется в соответствии с размером данных переданного кадра.

Флаг завершения обработки запроса на передачу кадра TX\_DONE, а также код результата передачи кадра TX\_REZ после их установки блоком передачи кадров продолжают стоять, а при выставлении следующего запроса на передачу кадра автоматически сбрасываются.

Флаг завершения обработки запроса на передачу кадра TX\_DONE доступен по записи, когда блок передачи кадров не выполняет обработку запроса на передачу кадра, то есть когда бит TX\_REQ = 0. Таким образом, после завершения обработки запроса на передачу кадра флаг TX\_DONE может быть сброшен записью нуля в соответствующий бит регистра STATUS\_TX.

Код результата передачи кадра TX\_REZ доступен только по чтению.

Бит MAC\_CONTROL<9> = CP\_TX предназначен для сброса указателей передающего TX\_FIFO между передачами кадров. Когда установлен запрос на передачу кадра, то есть бит TX\_REQ = 1, бит CP\_TX не доступен по записи. В связи с синхронизацией системной частоты HCLK и частоты передачи TX\_CLK сброс указателей передающего TX\_FIFO происходит с временной задержкой. Также, если сброс указателей выполняется на фоне работы канала DMA на запись, то перед выполнением сброса указателей требуется временная задержка, необходимая для завершения запущенных на передачу по каналу DMA пачек данных. После установки бит CP\_TX продолжает стоять, при этом бит становится недоступным для записи и поэтому не может быть сброшен. После выполнения сброса указателей передающего TX\_FIFO бит CP\_TX автоматически сбрасывается, после чего бит снова доступен для записи. В результате сброса указателей число слов в передающем TX\_FIFO обнуляется – STATUS\_TX<26:16> = TXW = 0.

Флаг завершения обработки запроса на передачу кадра TX\_DONE является запросом на прерывание от блока передачи кадров. Запрос на прерывание от блока передачи кадров маскируется. В бите MAC\_CONTROL<3> = MASK\_TX\_DONE устанавливается маска запроса на прерывание от блока передачи кадров.

Бит MAC\_CONTROL<10> = RST\_TX предназначен для программного сброса блока передачи кадров, а также регистров MAC\_ADDR\_L, MAC\_ADDR\_H, DADDR\_L, DADDR\_H, TYPE, FCS\_CLIENT, IFS\_COLL\_MODE, TX\_FRAME\_CONTROL, STATUS\_TX и разрядов регистра MAC\_CONTROL<3:0>. В связи с синхронизацией системной частоты HCLK и частоты передачи TX\_CLK требуется временная задержка для выполнения программного сброса блока передачи кадров. Также, если программный сброс выполняется на фоне работы канала DMA на запись, то перед выполнением программного сброса требуется временная задержка, необходимая для завершения запущенных на передачу по каналу DMA пачек данных. После установки бит RST\_TX продолжает стоять, при этом бит становится недоступным для записи и поэтому не может быть сброшен. После выполнения программного сброса блока передачи кадров бит RST\_TX автоматически сбрасывается, после чего бит снова доступен для записи.

На Рисунок 13.5 приведен порядок обработки запроса на передачу кадра блоком передачи кадров.

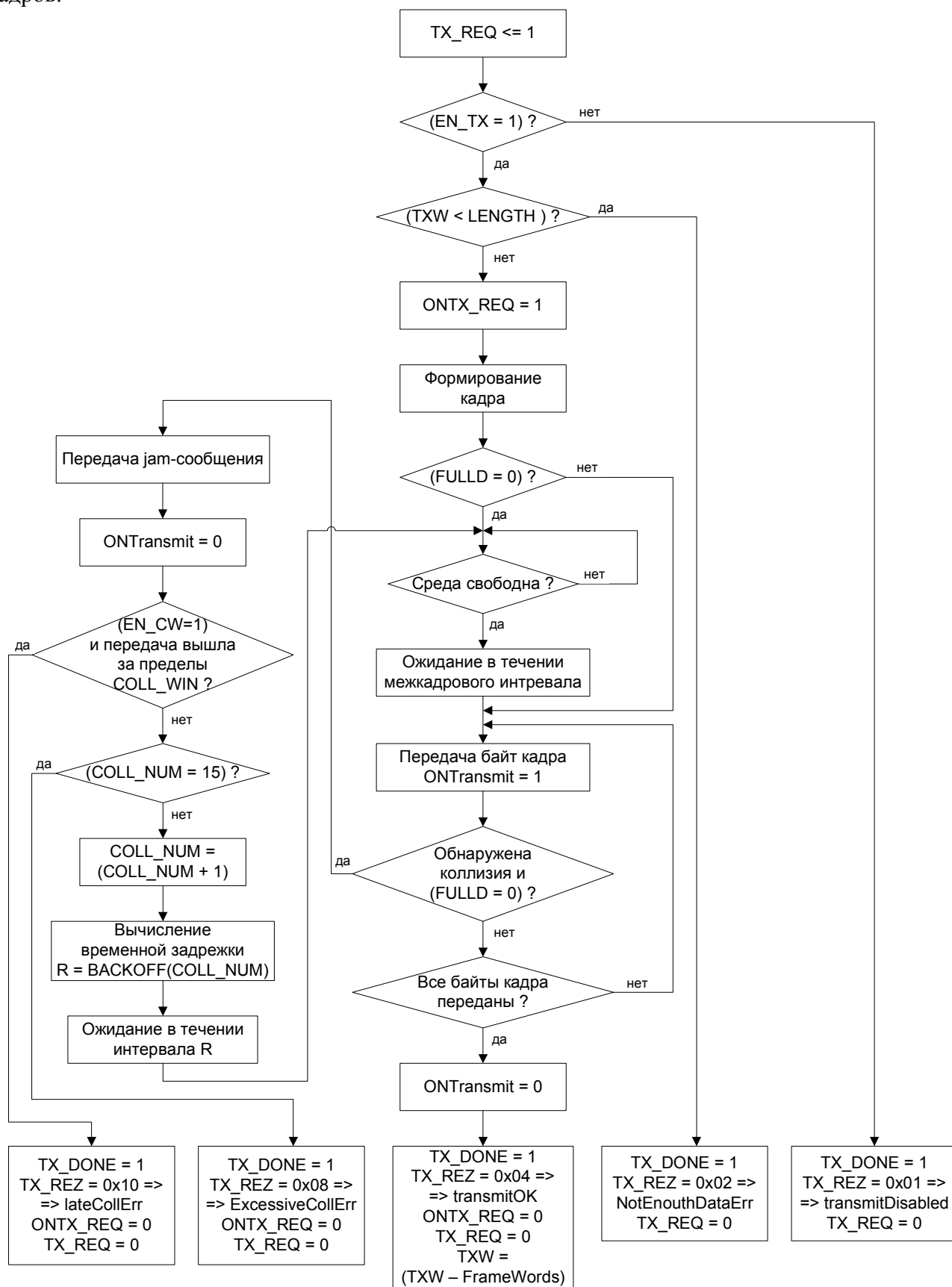


Рисунок 13.5. Порядок обработки запроса на передачу кадра.



#### 13.3.1.3.4 Обработка коллизий при передаче кадра.

Когда контроллер MAC работает в полудуплексном режиме

(бит  $MAC\_CONTROL<0> = FULLD = 0$ ), то во время передачи кадра в среде передачи может произойти коллизия. В случае обнаружения коллизии во время передачи кадра, блок передачи кадров вместо содержимого кадра начинает передавать 32-разрядное jam-сообщение, состоящее из 4 повторяющихся байт, чтобы сообщить другим станциям об обнаружении коллизии. После передачи jam-сообщения блок передачи кадров останавливает передачу и инкрементирует счетчик попыток повторных передач.

Значение повторяющегося байта jam-сообщения задается в разрядах регистра  $IFS\_COLL\_MODE<23:16> = JAMB$ .

Наличие коллизии в среде передачи отслеживается значением бита регистра статуса  $STATUS\_TX<3> = ONCOL$ .

Значение счетчика попыток повторных передач отображается в разрядах регистра статуса  $STATUS\_TX<15:12> = COLL\_NUM$ . Во время первой попытки передачи значение счетчика  $COLL\_NUM = 0$ . Счетчик попыток повторных передач  $COLL\_NUM$  доступен только по чтению. Значение счетчика попыток повторных передач  $COLL\_NUM$  автоматически сбрасывается при выставлении следующего запроса на передачу кадра.

После завершения передачи jam-сообщения блок передачи кадров переходит в состояние ожидания. Блок передачи кадров находится в состоянии ожидания в течение временной задержки, вычисленной в блоке  $BACKOFF$  в соответствии текущим значением номера попытки повторной передачи. По истечении временной задержки блок передачи кадров выполняет повторную попытку передачи кадра. В случае последующих обнаружений коллизий, блок передачи кадров будет выполнять повторные передачи кадра до тех пор, когда будет достигнуто максимальное количество попыток повторных передач кадра –  $ATTEMPT\_NUM$ . Максимальное количество попыток повторных передач кадра задается в разрядах регистра  $IFS\_COLL\_MODE<3:0> = ATTEMPT\_NUM$  и по умолчанию равно 15. Таким образом, по умолчанию блок передачи кадров выполняет до 16 попыток передачи кадра в соответствии со стандартом Ethernet.

В случае, когда при передаче кадра достигается максимальное количество попыток повторных передач кадра  $ATTEMPT\_NUM$ , и при этом последняя попытка передачи кадра также прерывается коллизией, тогда блок передачи кадров завершает обработку запроса на передачу кадра. Блок передачи кадров сообщает о завершении обработки запроса на передачу кадра выставлением в регистре статуса бита  $STATUS\_TX<3> = TX\_DONE = 1$ . По завершении обработки запроса на передачу кадра блок передачи кадров также сообщает результат передачи кадра в разрядах регистра статуса  $STATUS\_TX<8:4> = TX\_REZ = 0x08$  –  $ExcessiveCollErr$  – ошибка превышения максимального количества попыток повторных передач кадра.

Во время передачи кадра в среде передачи обычно может быть обнаружена коллизия в течение определенного временного промежутка после начала передачи, который требуется для распространения сигнала от передающей станции до всех остальных станций в среде передачи. Такой временной промежуток с начала передачи кадра называется окном коллизии. Размер окна коллизии задается как число байт кадра, для передачи которых требуется определенный промежуток времени, и устанавливается в разрядах регистра  $IFS\_COLL\_MODE<23:16> = COLL\_WIN$ . Размер окна коллизии должен быть больше 14 байт. В соответствии со стандартом Ethernet размер окна коллизии равен временному интервалу  $slotTime$ , который равен времени передачи 512 бит, что соответствует времени передачи 64 байт кадра. Таким образом, по умолчанию размер окна коллизии  $COLL\_WIN$  равен 64 байта. Для разрешения отслеживания окна коллизии должен быть установлен бит

IFS\_COLL\_MODE<4> = EN\_CW = 1. По умолчанию отслеживание окна коллизии разрешено.

В случае обнаружении коллизии во время передачи кадра, если разрешено отслеживание окна коллизии (IFS\_COLL\_MODE<4> = EN\_CW = 1), то блок передачи кадров проверяет вышла ли текущая передача за пределы окна коллизии. Таким образом, если обнаружена коллизия и при этом разрешено отслеживание окна коллизии (IFS\_COLL\_MODE<4> = EN\_CW = 1), а текущая передача вышла за пределы окна коллизии, то блок передачи кадров после завершения передачи jam-сообщения не делает повторных попыток передачи кадра, а завершает обработку запроса на передачу кадра. Блок передачи кадров сообщает о завершении обработки запроса на передачу кадра выставлением в регистре статуса бита STATUS\_TX<3> = TX\_DONE = 1.

По завершении обработки запроса на передачу кадра блок передачи кадров также сообщает результат передачи кадра в разрядах регистра статуса STATUS\_TX<8:4> = TX\_REZ = 0x10 – lateCollErr – ошибка поздней коллизии.

В случае, когда отслеживание окна коллизии не разрешено, то есть бит IFS\_COLL\_MODE<4> = EN\_CW = 0, тогда независимо от момента обнаружения коллизий, блок передачи кадров будет выполнять повторные попытки передачи кадра до тех пока передача кадра не будет успешно завершена или пока не будет достигнуто максимальное количество попыток повторных передач кадра.

Если коллизия обнаруживается в первые несколько тактов после успешного завершения передачи кадра, то блок передачи кадров завершает обработку запроса на передачу кадра и сообщает об этом выставлением в регистре статуса бита STATUS\_TX<3> = TX\_DONE = 1, а также сообщает результат передачи кадра в разрядах регистра статуса STATUS\_TX<8:4> = TX\_REZ = 0x14 – одновременно transmitOK и lateCollErr – передача кадра успешно выполнена и при этом ошибка поздней коллизии.

Когда контроллер MAC работает в дуплексном режиме (бит MAC\_CONTROL<0> = FULLD = 1), тогда в среде передачи не может возникать коллизий. Таким образом, передача кадра при работе в дуплексном режиме не может быть прервана и всегда успешно завершается с первой попытки передачи.

#### 13.3.1.4 Блок CALC\_CRC32

Блок CALC\_CRC32 вычисляет контрольную сумму CRC32 передаваемого кадра.

Контрольная сумма представляет собой 32-разрядное значение, которое вычисляется как функция от содержимого полей кадра: <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE>, <DATA>, <PAD>.

Алгоритм вычисления контрольной суммы CRC32 определяется полиномом:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 ;$$

Разряды вычисленной контрольной суммы CRC<31:0> помещаются в поле <FCS> так, что старший разряд CRC<31> помещается в младший разряд поля FCS<0>, а младший разряд CRC<0> помещается в старший разряд поля FCS<31>. Таким образом, поле FCS<31:0> = {CRC<0>, CRC<1>, ...12, CRC<30>, CRC<31>}.

Следует отметить, что если при передаче кадра используется регистр FCS\_CLIENT, то в этот регистр помещается непосредственно значение контрольной суммы CRC<31:0>, то есть FCS\_CLIENT<31:0> = CRC<31:0>.

Если же в TX\_FIFO передается сформированный кадр, содержащий уже вычисленное значение поля <FCS>, то в этом случае формат поля <FCS> должен соответствовать выражению:  $FCS<31:0> = \{CRC<0>, CRC<1>, \dots, CRC<30>, CRC<31>\}$ .

### 13.3.1.5 Блок BACKOFF

Блок BACKOFF вычисляет временную задержку перед повторной передачей кадра при обнаружении коллизии. Временная задержка определяется как целое число R временных интервалов slotTime. Временной интервал slotTime равен времени передачи 512 бит, что соответствует 128 тактам частоты передачи TX\_CLK.

R – целое число временных интервалов slotTime – вычисляется как случайное значение в диапазоне  $0 \leq R < 2K$ ,

где  $K = \min(n, 10)$ ,  $1 \leq n \leq 15$ , n – номер попытки повторной передачи.

Для блока BACKOFF предусмотрен тестовый режим работы. Для включения тестового режима работы блока BACKOFF необходимо установить бит IFS\_COLL\_MODE<7> = TM\_BACKOFF = 1. В тестовом режиме работы целое число временных интервалов slotTime – R – вычисляется в диапазоне:  $0 \leq R \leq 1$ .

### 13.3.1.6 Режим тестирования TX\_FIFO

Для тестирования записи данных по DMA-каналу в передающее TX\_FIFO предусмотрен режим тестирования TX\_FIFO. Для включения режима тестирования TX\_FIFO необходимо установить в регистре управления и состояния режима тестирования TX\_FIFO бит разрешения режима тестирования – TX\_TEST\_CSR<0> = TM\_TX\_FIFO = 1.

Когда разрешен режим тестирования передающего TX\_FIFO, то обмен по каналу DMA с TX\_FIFO невозможен. Данные поступающие на запись в TX\_FIFO при разрешенном режиме тестирования игнорируются.

Если разрешен режим тестирования, то TX\_FIFO доступно для чтения по адресу TX\_FIFO. Таким образом, в режиме тестирования последовательными чтениями 32-разрядных слов может быть вычитано содержимое TX\_FIFO. При этом чтение TX\_FIFO начинается с нулевой ячейки.

Число прочтенных 32-разрядных слов из TX\_FIFO отображается в разрядах регистра управления и состояния режима тестирования TX\_TEST\_CSR<14:4> = TM\_TX\_RDW. После сброса бита разрешения режима тестирования TX\_FIFO число прочтенных из TX\_FIFO слов – TM\_TX\_RDW – обнуляется.

### 13.3.1.7 Блок приема кадров ReceiveFrame

Для разрешения работы блока приема кадров должен быть установлен бит MAC\_CONTROL<4> = EN\_RX = 1.

Блок приема кадров может быть сконфигурирован для работы в режиме зацикливания блока приема кадров на блок передачи кадров. Для задания режима зацикливания в регистре управления MAC необходимо установить бит MAC\_CONTROL<5> = LOOPBACK = 1.

Для задания параметров фильтрации кадров по адресу назначения необходимо установить биты регистра RX\_FRAME\_CONTROL<9:6>, а также регистры UCADDR\_L, UCADDR\_H, MCADDR\_L, MCADDR\_H, MCADDR\_MASK\_L, MCADDR\_MASK\_H, HASHT\_L, HASHT\_H.

В регистре RX\_FR\_MaxSize необходимо установить значение максимального размера принимаемого кадра в байтах. По умолчанию максимальный размер принимаемого кадра равен 1518 байт в соответствии со стандартом Ethernet.

Также в разрядах регистра RX\_FRAME\_CONTROL<5:0> необходимо задать параметры проверки и обработки принятого кадра.

Блок приема кадров постоянно анализирует состояние сигнала RX\_DV для обнаружения трансляции кадра в среде передачи.

В случае, когда блок приема кадров обнаруживает, что установился сигнал RX\_DV и при этом бит разрешения работы блока приема кадров MAC\_CONTROL<4> = EN\_RX = 0, тогда блок приема кадров пропускает транслируемый кадр и сообщает об этом выставлением в регистре статуса бита STATUS\_RX<0> = RCV\_Disabled = 1. Бит RCV\_Disabled после выставления продолжает стоять и будет автоматически сброшен после завершения трансляции пропускаемого кадра в среде передачи, то есть когда снимется сигнал RX\_DV.

Когда блок приема кадров обнаруживает, что установился сигнал RX\_DV и при этом установлен бит разрешения работы блока приема кадров MAC\_CONTROL<4> = EN\_RX = 1, тогда блок приема кадров начинает прием кадра.

Если бит разрешения работы блока приема кадров MAC\_CONTROL<4> = EN\_RX будет сброшен после того как блок приема кадров начал прием кадра, то, несмотря на это, прием текущего кадра будет продолжен.

Когда контроллер MAC работает в полудуплексном режиме (бит MAC\_CONTROL<0> = FULLD = 0), то контроллер MAC может выполнять либо прием, либо передачу кадра. Таким образом, если в полудуплексном режиме блок передачи кадров выполняет передачу кадра, то во время передачи блок приема кадров пропускает транслируемые на прием кадры.

Бит регистра MAC\_CONTROL<6> = FULLD\_RX – включает тестовый режим работы блока приема кадров, при работе в котором блок приема кадров будет принимать транслируемые на прием кадры во время выполнения блоком передачи кадров передачи данных при работе контроллера в полудуплексном режиме (FULLD=0).

В начале приема кадра блок приема кадров ожидает на прием байты полей <PREAMBLE> и <SFD>. При этом поле <PREAMBLE> может содержать от 1 до 7 байт, либо поле <PREAMBLE> может отсутствовать, и тогда кадр начинается сразу с поля <SFD>.

Если после принятия 8 байт блок приема кадров не обнаружил поле <SFD>, 1 байт которого имеет значение 0xD5, то блок приема кадров прекращает прием транслируемых данных, которые не являются корректным кадром.

Как только блок приема кадров при приеме первых 8 байт обнаруживает поле <SFD>, блок приема кадров начинает прием 6 байт поля <DESTINATION ADDRESS> – адреса назначения. Принятый 48-разрядный адрес назначения поступает в блок DADDR\_CHECK. В блоке DADDR\_CHECK выполняется распознавание принятого адреса назначения в соответствии с заданными параметрами в битах регистра RX\_FRAME\_CONTROL<9:6>, а также в соответствии со значениями регистров UCADDR\_L, UCADDR\_H, MCADDR\_L, MCADDR\_H, MCADDR\_MASK\_L, MCADDR\_MASK\_H, HASHT\_L, HASHT\_H.

В случае, когда принятый адрес назначения не был распознан в блоке DADDR\_CHECK, тогда блок приема кадров прекращает прием текущего транслируемого кадра, так как данный кадр считается предназначенным для другой станции.

В случае, когда принятый адрес назначения был распознан в блоке DADDR\_CHECK, тогда текущий транслируемый кадр считается предназначенным для контроллера MAC и блок приема кадров продолжает прием остальных полей кадра.

Бит статусного регистра STATUS\_RX<1> = ONReceive позволяет отслеживать состояние блока приема кадров. Если был распознан адрес назначения и блок приема кадров выполняет прием кадра, то бит ONReceive устанавливается и продолжает стоять в течение приема кадра. Как только блок приема кадров завершает прием кадра, бит ONReceive сбрасывается.

Во время приема кадра по принимаемым байтам полей кадра, за исключением 4 байт поля <FCS>, в блоке CRC32\_CHECK вычисляется контрольная сумма CRC32. После завершения приема кадра в блоке CRC32\_CHECK контрольная сумма CRC32, вычисленная по данным принятого кадра, сравнивается со значением принятого поля <FCS>. В случае, если вычисленное значение не совпадает с принятым, то блок CRC32\_CHECK выставляет флаг ошибки контрольной суммы принятого кадра.

В случае если во время приема кадра устанавливается сигнал RX\_ER, то блок приема кадров определяет, что была обнаружена ошибка принятых данных.

В случае, когда объем транслируемых данных превышает максимальный допустимый размер принимаемого кадра, заданный в регистре RX\_FR\_MaxSize, тогда после приема объема данных, равного максимальному размеру принимаемого кадра + 1 байт, дальнейший прием транслируемого кадра прекращается.

При приеме кадра блок приема кадров компонует поступающие байты полей кадра в 64-разрядные слова и сохраняет их в принимающее FIFO – RX\_FIFO. Каждое 64-разрядное слово составляется из 8 принятых байт кадра в порядке их поступления, начиная с байта, который был принят первым:

63	56	55	48	47	40	39	32	31	24	23	16	15	8	7	0
byte <sub>(n+7)</sub>		byte <sub>(n+6)</sub>		byte <sub>(n+5)</sub>		byte <sub>(n+4)</sub>		byte <sub>(n+3)</sub>		byte <sub>(n+2)</sub>		byte <sub>(n+1)</sub>		byte <sub>(n)</sub>	


 Байты были приняты, начиная с младшего

В случае если для компоновки последнего 64-разрядного слова из принятых байт кадра остается меньше 8 принятых байт кадра, то последние принятые байты кадра помещаются в соответствующие младшие разряды слова: 1 байт – в разряды <7:0>, 2 байта – в разряды <15:0>, 3 байта – в разряды <23:0>, 4 байта – в разряды <31:0>, 5 байт – в разряды <39:0>, 6 байт – в разряды <47:0>, 7 байт – в разряды <55:0>. Оставшиеся старшие разряды слова заполняются нулевыми значениями.

Таким образом, при приеме кадра в принимающее RX\_FIFO последовательно записываются поступающие поля кадра: <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE>, <DATA>, <PAD>, <FCS>.

Если во время приема кадра при записи принятых байт кадра в принимающее RX\_FIFO происходит переполнение принимающего RX\_FIFO, то блок приема кадров прекращает прием транслируемого кадра, а уже принятые байты кадра отбрасываются. Для сообщения об этом блок приема кадров выставляет в регистре статуса флаг переполнения принимающего RX\_FIFO – STATUS\_RX<23> = RX\_FIFO\_OVF\_Err = 1, а также инкрементируется число пропущенных кадров из-за переполнения FIFO – NUM\_Missed\_FR. Число пропущенных кадров отображается в разрядах регистра статуса STATUS\_RX<29:24> = NUM\_Missed\_FR.



Как только сбрасывается сигнал RX\_DV блок приема кадров завершает прием кадра. После завершения приема кадра блок приема кадров выполняет проверку и обработку принятого кадра в соответствии с заданными параметрами в разрядах регистра RX\_FRAME\_CONTROL<5:0>.

В случае если во время приема кадра поступает нечетное число полубайт данных, то блок приема кадров принимает целое число байт данных кадра, а нечетный полубайт данных отбрасывает.

Порядок проверки принятого кадра блоком приема кадров:

- Если размер принятого кадра составляет меньше 18 байт, то такой кадр считается некорректным и блок приема кадров отбрасывает этот кадр.
- Если размер принятого кадра составляет меньше 64 байт (минимальный размер кадра в соответствии со стандартом Ethernet), то такой кадр определяется как слишком короткий кадр и для него устанавливается статусный флаг – RX\_FRAME\_STATUS<17> = frameTooShort = 1.
- Если во время приема кадра объем транслируемых данных превысил максимальный размер принимаемого кадра, заданный в регистре RX\_FR\_MaxSize, то такой кадр определяется как слишком длинный кадр и для него устанавливается статусный флаг – RX\_FRAME\_STATUS<16> = frameTooLong = 1.
- Если при приеме кадра поступило нечетное число полубайт, то есть нецелое число байт данных, то для такого кадра устанавливается статусный флаг – RX\_FRAME\_STATUS<18> = DribbleNibble = 1.
- Если блок CRC32\_CHECK выставляет флаг ошибки контрольной суммы принятого кадра, а при приеме кадра поступило нечетное число полубайт данных, то принятый кадр определяется как кадр с ошибкой выравнивания и для него устанавливается статусный флаг – RX\_FRAME\_STATUS<14> = alignmentError = 1.
- Если блок CRC32\_CHECK выставляет флаг ошибки контрольной суммы принятого кадра, и при приеме кадра поступило целое число байт данных, либо если во время приема кадра была обнаружена ошибка принятых данных (RX\_ER = 1), то принятый кадр определяется как кадр с ошибкой проверки кадра и для него устанавливается статусный флаг – RX\_FRAME\_STATUS<15> = frameCheckError = 1.
- Если в принятом кадре значение поля <LENGTH/TYPE> ≤ 1500 байт, то в соответствии со стандартом Ethernet поле <LENGTH/TYPE> в данном кадре трактуется как поле <LENGTH>. Для такого кадра устанавливается статусный флаг – RX\_FRAME\_STATUS<19> = LEN\_FIELD = 1.
- Если для принятого кадра установлен статусный флаг LEN\_FIELD = 1, в принятом кадре не обнаружено поле <PAD>, а число байт данных в поле <DATA> принятого кадра не совпадает со значением, принятого поля <LENGTH>, то принятый кадр определяется как кадр с ошибкой длины поля данных <DATA> и для него устанавливается статусный флаг – RX\_FRAME\_STATUS<13> = lengthError = 1.
- Если при проверке принятого кадра для него не выставляется ни один из статусных флагов: frameTooShort, frameTooLong, alignmentError, frameCheckError, lengthError, – тогда кадр считается успешно принятым без обнаружения ошибок и для такого кадра устанавливается статусный флаг – RX\_FRAME\_STATUS<12> = receiveOK = 1.

После проверки принятого кадра блок приема кадров выполняет затем его обработку в соответствии с заданными параметрами в разрядах регистра `RX_FRAME_CONTROL<5:0>`:

- Если для принятого кадра во время проверки был установлен статусный флаг `frameTooShort = 1`, а бит разрешения приема слишком коротких кадров `RX_FRAME_CONTROL<2> = Accept_TooShort = 0`, то принятый кадр отбрасывается.
- Если для принятого кадра во время проверки был установлен статусный флаг `frameTooLong = 1`, а бит разрешения отбрасывания слишком длинных кадров `RX_FRAME_CONTROL<3> = Discard_TooLong = 1`, то принятый кадр отбрасывается.
- Если для принятого кадра во время проверки был установлен статусный флаг `alignmentError = 1` или статусный флаг `frameCheckError = 1`, а бит разрешения отбрасывания кадров с ошибкой проверки контрольной суммы `RX_FRAME_CONTROL<4> = Discard_FCSErr = 1`, то принятый кадр отбрасывается.
- Если для принятого кадра во время проверки был установлен статусный флаг `lengthError = 1`, а бит разрешения отбрасывания кадров с ошибкой длины поля данных `RX_FRAME_CONTROL<5> = Discard_LengthErr = 1`, то принятый кадр отбрасывается.
- Если принятый кадр после проверки не был отброшен, а бит отключения сохранения поля <FCS> в принятом кадре `RX_FRAME_CONTROL<0> = Dis_RCV_FCS = 1`, то блок приема кадров удаляет из принятого кадра последние 4 байта – байты поля <FCS>. Блок приема кадров сообщает об удалении поля <FCS> в принятом кадре выставлением для него статусного флага – `RX_FRAME_STATUS<20> = FCS_Del = 1`.
- Если принятый кадр после проверки не был отброшен, и при этом в принятом кадре было обнаружено поле <PAD>, бит отключения сохранения поля <FCS> в принятом кадре `RX_FRAME_CONTROL<0> = Dis_RCV_FCS = 1`, а бит отключения удаления в принятом кадре поля <PAD> `RX_FRAME_CONTROL<1> = Dis_PAD_Del = 0`, то блок приема кадров удаляет из принятого кадра байты поля <PAD>. Блок приема кадров сообщает об удалении поля <PAD> в принятом кадре выставлением для него статусного флага – `RX_FRAME_STATUS<21> = PAD_Del = 1`.

Значение числа байт в принятом кадре сохраняется в разрядах статуса принятого кадра `RX_FRAME_STATUS<11:0> = RX_FR_LENGTH`.

В случае, когда после проверки принятого кадра блок приема кадров отбрасывает кадр, тогда блок приема кадров никак не сообщает о том, что кадр принимался и был отброшен, число слов в принимающем `RX_FIFO – RXW` остается неизменным.

Число 64-разрядных слов в принимающем `FIFO – RX_FIFO` – отображается в разрядах регистра статуса `STATUS_RX<22:12> = RXW` (`RXW` содержит информацию о количестве данных в `RX_FIFO` с точностью до байта, но в регистре статуса отображена информация с точностью до 64-разрядного слова округленного в меньшую сторону).

В случае, когда после проверки и обработки принятого кадра блоком приема кадров кадр не был отброшен, тогда считается, что блок приема кадров принял кадр.

В процессе проверки и обработки принятого кадра блок приема кадров формирует статус принятого кадра `RX_FRAME_STATUS`. По принятию кадра блок приема кадров записывает сформированный статус принятого кадра `RX_FRAME_STATUS` в `FIFO` статусов принятых кадров – `RX_FRAME_STATUS_FIFO`. `FIFO` статусов принятых кадров имеет объем в 64 слова статусов кадров.

При этом по принятию кадра инкрементируется число принятых кадров – NUM\_RX\_FR. Число принятых кадров отображается в разрядах регистра статуса STATUS\_RX<10:4>= NUM\_RX\_FR.

Также по принятию кадра число слов в принимающем RX\_FIFO – RXW инкрементируется в соответствии с размером данных принятого кадра. После этого данные принятого кадра доступны для вычитывания по DMA-каналу чтения DMA\_EMAC\_CH0. Данные принятого кадра вычитываются по DMA-каналу чтения из принимающего RX\_FIFO в виде последовательности 64-разрядных слов (с точностью до байта). Так как DMA может передавать данные с точностью до байта, то в случае когда длина кадра не кратна 8-ми байт, нет необходимости вычитывать нулевые байты дополняющие 64-разрядную строку. Выгрузку очередного кадра предваряет чтение FIFO статусов, что является командой к отбросу ненужных нулевых байтов. Для обнаружения наличия принятых кадров в принимающем RX\_FIFO используется бит статусного регистра STATUS\_TX<3> = RX\_DONE. Флаг наличия принятых кадров в принимающем RX\_FIFO – RX\_DONE устанавливается, когда в FIFO статусов принятых кадров имеются непрочитанные статусы принятых кадров, то есть FIFO статусов не пустое. После опустошения FIFO статусов принятых кадров флаг RX\_DONE автоматически сбрасывается. При вычитывании слова статуса кадра из FIFO статусов принятых кадров, число принятых кадров NUM\_RX\_FR декрементируется. FIFO статусов принятых кадров доступно только по чтению. Указатели FIFO статусов принятых кадров могут быть сброшены путем выполнения записи по адресу FIFO статусов произвольного значения. При сбросе указателей FIFO статусов число принятых кадров NUM\_RX\_FR обнуляется.

Если FIFO статусов принятых кадров полное, то есть NUM\_RX\_FR = 64, и при этом блок приема кадров завершает прием нового кадра, тогда при попытке записи статуса принятого кадра в заполненное FIFO статусов блок приема кадров обнаруживает переполнение FIFO статусов принятых кадров. При обнаружении переполнения FIFO статусов принятых кадров блок приема кадров отбрасывает принятый кадр и сообщает об этом выставлением в регистре статуса флага переполнения FIFO статусов принятых кадров – STATUS\_RX<11> = FR\_STATUS\_OVF\_Err = 1. Также при этом инкрементируется число пропущенных кадров из-за переполнения FIFO – NUM\_Missed\_FR. Так как принятый кадр отбрасывается, то число слов в принимающем RX\_FIFO – RXW остается неизменным.

Флаг переполнения FIFO статусов принятых кадров FR\_STATUS\_OVF\_Err и флаг переполнения принимающего RX\_FIFO – RX\_FIFO\_OVF\_Err доступны по записи и в случае их выставления могут быть сброшены записью нулей в соответствующие биты регистра STATUS\_RX.

Бит MAC\_CONTROL<11> = CP\_RX предназначен для сброса указателей принимающего RX\_FIFO между приемами кадров. Во время приема кадра (ONReceive = 1) бит CP\_RX не доступен по записи. В связи с синхронизацией системной частоты HCLK и частоты приема RX\_CLK сброс указателей принимающего RX\_FIFO происходит с задержкой. Также, если сброс указателей выполняется на фоне работы канала DMA на чтение, то перед выполнением сброса указателей требуется временная задержка, необходимая для завершения запущенных на передачу по каналу DMA пакетов данных. После установки бит CP\_RX продолжает стоять, при этом бит становится недоступным для записи и поэтому не может быть сброшен. После выполнения сброса указателей принимающего RX\_FIFO бит CP\_RX автоматически сбрасывается, после чего бит снова доступен для записи. В результате сброса указателей число слов в принимающем RX\_FIFO обнуляется – STATUS\_RX<22:12> = RXW = 0.

Флаг наличия принятых кадров в принимающем RX\_FIFO – RX\_DONE, а также флаги переполнения принимающего RX\_FIFO, FIFO статусов принятых кадров – RX\_FIFO\_OVF\_Err и



FR\_STATUS\_OVF\_Err – выставление одного из этих флагов является запросом на прерывание от блока приема кадров. Запрос на прерывание от блока приема кадров маскируется.

В бите MAC\_CONTROL<7> = MASK\_RX\_DONE устанавливается маска флага RX\_DONE (флаг наличия принятых кадров в принимающем RX\_FIFO), выставление которого является запросом на прерывание от блока приема кадров.

В бите MAC\_CONTROL<8> = MASK\_RX\_FIFO\_OVF\_ERR устанавливается маска флагов RX\_FIFO\_OVF\_Err и FR\_STATUS\_OVF\_Err (флагов переполнения принимающего RX\_FIFO и FIFO статусов принятых кадров), выставление одного из которых является запросом на прерывание от блока приема кадров. На рисунке 14.6 приведен порядок приема кадров блоком приема кадров.

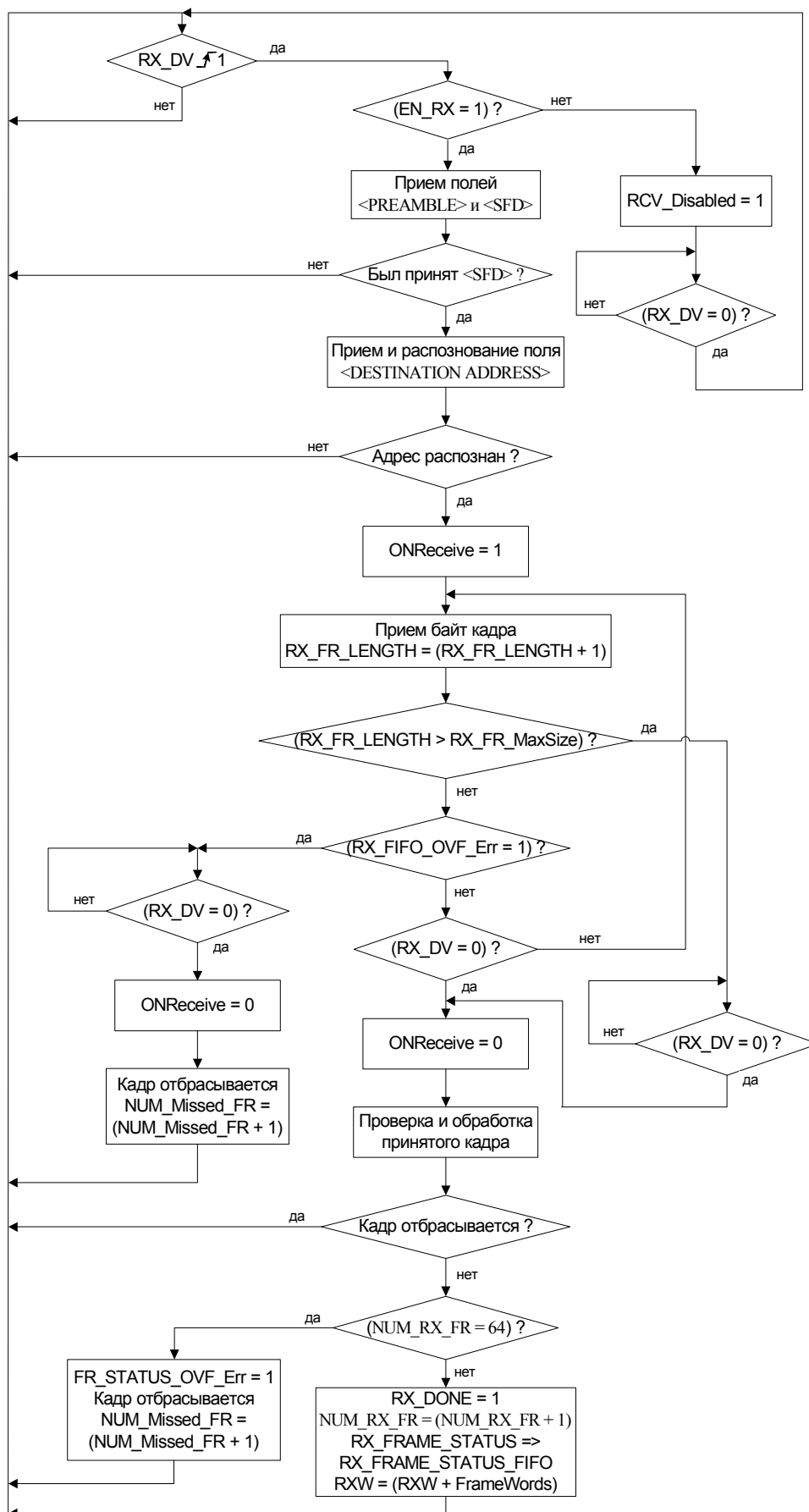


Рисунок 13.6. Порядок приема кадров

Бит  $MAC\_CONTROL<12> = RST\_RX$  предназначен для программного сброса блока приема кадров, а также регистров  $UCADDR\_L$ ,  $UCADDR\_H$ ,  $MCADDR\_L$ ,  $MCADDR\_H$ ,  $MCADDR\_MASK\_L$ ,  $MCADDR\_MASK\_H$ ,  $HASHT\_L$ ,  $HASHT\_H$ ,  $RX\_FR\_MaxSize$ ,  $RX\_FRAME\_CONTROL$ ,  $STATUS\_RX$ , разрядов регистра  $MAC\_CONTROL<8:4>$  и указателей FIFO статусов принятых кадров. В связи с синхронизацией системной частоты  $HCLK$  и частоты приема  $RX\_CLK$  требуется временная задержка для выполнения программного сброса блока приема кадров.

Также, если программный сброс выполняется на фоне работы канала DMA на чтение, то перед выполнением программного сброса требуется временная задержка, необходимая для завершения запущенных на передачу по каналу DMA пакетов данных. После установки бит  $RST\_RX$  продолжает стоять, при этом бит становится недоступным для записи и поэтому не может быть сброшен. После выполнения программного сброса блока приема кадров бит  $RST\_RX$  автоматически сбрасывается, после чего бит снова доступен для записи.

### 13.3.1.8 Блок $DADDR\_CHECK$

Блок  $DADDR\_CHECK$  после принятия в блоке приема кадров 6 байт поля  $<DESTINATION\ ADDRESS>$  выполняет распознавание принятого адреса назначения в соответствии с заданными параметрами в битах регистра  $RX\_FRAME\_CONTROL<9:6>$ , а также в соответствии со значениями регистров  $UCADDR\_L$ ,  $UCADDR\_H$ ,  $MCADDR\_L$ ,  $MCADDR\_H$ ,  $MCADDR\_MASK\_L$ ,  $MCADDR\_MASK\_H$ ,  $HASHT\_L$ ,  $HASHT\_H$ .

Порядок распознавания принятого адреса назначения:

- Если установлен бит разрешения приема кадров с любым адресом назначения  $RX\_FRAME\_CONTROL<9> = EN\_ALL = 1$ , то принятый адрес назначения считается распознанным и для принимаемого кадра устанавливается статусный флаг –  $RX\_FRAME\_STATUS<16> = ALL = 1$ .

- Если значение принятого 48-разрядного адреса назначения

$DA<47:0> = 0xFFFFFFFFFFFF$ ,

то такой адрес назначения является широковещательным. Если при этом не установлен бит запрещения приема кадров с широковещательным адресом назначения  $RX\_FRAME\_CONTROL<6> = Dis\_BC = 0$ , то принятый адрес назначения считается распознанным и для принимаемого кадра устанавливается статусный флаг –  $RX\_FRAME\_STATUS<25> = BC = 1$ .

- Если принятый адрес назначения  $DA$  является индивидуальным адресом

$(DA<0> = 0)$ ,

тогда принятый 48-разрядный адрес назначения  $DA<47:0>$  сравнивается с 48-разрядным значением уникального адреса MAC, сформированного из значения регистров  $UCADDR\_L$ ,  $UCADDR\_H$ :

?

$$DA<47:0> = \{UCADDR\_H<15:0>, UCADDR\_L<31:0>\}.$$

При совпадении значения принятого адреса назначения и значения уникального адреса MAC, адрес назначения считается распознанным и для принимаемого кадра устанавливается статусный флаг –  $RX\_FRAME\_STATUS<22> = UC = 1$ .

- Если принятый адрес назначения DA является групповым адресом ( $DA_{<0>} = 1$ ) и при этом установлен бит  $RX\_FRAME\_CONTROL_{<7>} = EN\_MCM = 1$ , тогда принятый 48-разрядный адрес назначения  $DA_{<47:0>}$  сравнивается с 48-разрядным значением группового адреса MAC, сформированного из значения регистров MCADDR\_L, MCADDR\_H с учетом наложения на 48-разрядные адреса маски, заданной в регистрах MCADDR\_MASK\_L, MCADDR\_MASK\_H. Таким образом, на значение принятого адреса назначения накладывается маска:

$$DA_{<47:0>} \& \{MCADDR\_MASK\_H_{<15:0>}, MCADDR\_MASK\_L_{<31:0>}\},$$

также на значение группового адреса MAC накладывается маска:

$$\{MCADDR\_H_{<15:0>}, MCADDR\_L_{<31:0>}\} \& \{MCADDR\_MASK\_H_{<15:0>}, MCADDR\_MASK\_L_{<31:0>}\},$$

а затем полученные замаскированные значения адресов сравниваются:

$$DA \& MCADDR\_MASK \stackrel{?}{=} MCADDR \& MCADDR\_MASK.$$

При совпадении замаскированных адресов, адрес назначения считается распознанным и для принимаемого кадра устанавливается статусный флаг –  $RX\_FRAME\_STATUS_{<23>} = MCM = 1$ .

- Если принятый адрес назначения DA является групповым адресом ( $DA_{<0>} = 1$ ) и при этом установлен бит

$$RX\_FRAME\_CONTROL_{<8>} = EN\_MCHT = 1,$$

тогда по принятому 48-разрядному адресу назначения  $DA_{<47:0>}$  в блоке CRC32\_CHECK вычисляется контрольная сумма  $DA\_CRC_{<31:0>}$ . Значение бита вычисленной контрольной суммы  $DA\_CRC_{<31>}$  определяет младшая или старшая часть хэш-таблицы будет использоваться для распознавания адреса назначения. Если бит  $DA\_CRC_{<31>} = 0$ , то для распознавания адреса используется младшая часть хэш-таблицы, заданная в регистре HASHT\_L. Если бит  $DA\_CRC_{<31>} = 1$ , то для распознавания адреса используется старшая часть хэш-таблицы, заданная в регистре HASHT\_H. Значение пяти бит вычисленной контрольной суммы  $DA\_CRC_{<30:26>}$  задает номер бита в используемой части (старшей или младшей) хэш-таблицы (HASHT\_L или HASHT\_H). Таким образом, из 64 разрядов хэш-таблицы, заданной в регистрах HASHT\_L и HASHT\_H, выбирается один бит. Если выбранный таким образом из хэш-таблицы бит установлен в 1, тогда адрес назначения считается распознанным и для принимаемого кадра устанавливается статусный флаг:

$$RX\_FRAME\_STATUS_{<24>} = MCHT = 1.$$

На рисунке 14.7 приведен порядок распознавания принятого адреса назначения.

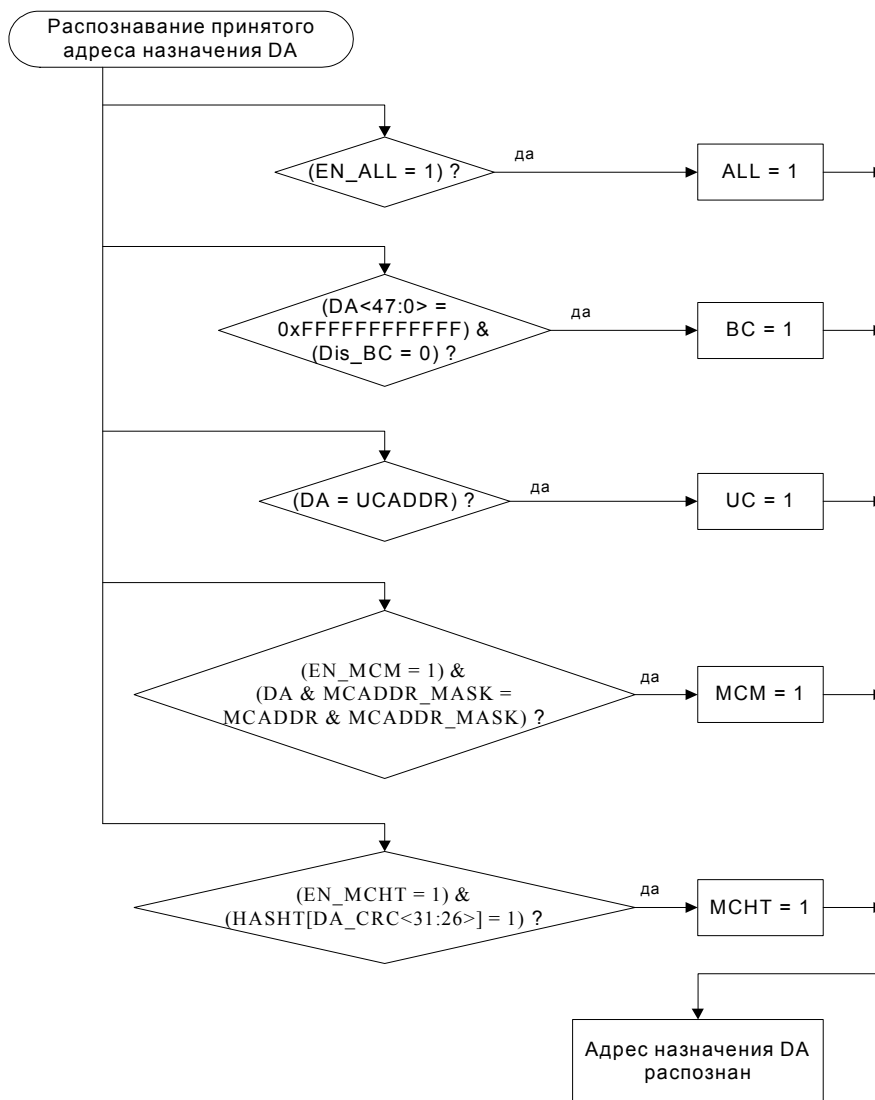


Рисунок 13.7. Порядок распознавания принятого адреса назначения

### 13.3.1.9 Блок CRC32\_CHECK

Блок CRC32\_CHECK во время приема кадра блоком приема кадров вычисляет по принимаемым байтам полей кадра контрольную сумму CRC32.

Контрольная сумма представляет собой 32-разрядное значение, которое вычисляется как функция от содержимого полей кадра: <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE>, <DATA>, <PAD>.

Алгоритм вычисления контрольной суммы CRC32 определяется полиномом:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 ;$$

После завершения приема в блоке приема кадров всех полей кадра 32-разрядное значение вычисленной контрольной суммы CRC<31:0> сравнивается со значением принятых 4 байт поля <FCS>. Если вычисленное значение контрольной суммы CRC<31:0> не совпадает с поступившим значением FCS<31:0>, тогда блок CRC32\_CHECK устанавливает флаг ошибки контрольной суммы принятого кадра.

Также блок CRC32\_CHECK после принятия в блоке приема кадров 6 байт поля <DESTINATION ADDRESS> вычисляет для блока DADDR\_CHECK контрольную сумму DA\_CRC только по байтам поля <DESTINATION ADDRESS>.

### 13.3.1.10 Режим тестирования RX\_FIFO

Для тестирования чтения данных по DMA-каналу из принимающего RX\_FIFO предусмотрен режим тестирования RX\_FIFO.

Для включения режима тестирования необходимо установить в регистре управления и состояния режима тестирования RX\_FIFO бит разрешения режима тестирования – RX\_TEST\_CSR<0> = TM\_RX\_FIFO = 1. Бит разрешения режима тестирования TM\_RX\_FIFO не доступен по записи когда разрешена работа блока приема кадров MAC\_CONTROL<4> = EN\_RX = 1 или во время приема кадра (ONReceive = 1).

При установке бита разрешения режима тестирования RX\_FIFO – TM\_RX\_FIFO = 1, автоматически устанавливается бит сброса указателей принимающего RX\_FIFO – MAC\_CONTROL<11> = CP\_RX = 1. Таким образом, после разрешения режима тестирования RX\_FIFO необходимо дождаться выполнения сброса указателей принимающего RX\_FIFO, то есть дождаться когда бит CP\_RX будет автоматически сброшен.

Когда разрешен режим тестирования, тогда RX\_FIFO становится недоступным для чтения по DMA-каналу.

Если разрешен режим тестирования, то RX\_FIFO доступно для записи по адресу RX\_FIFO. Таким образом, в режиме тестирования последовательными записями 32-разрядных слов может быть заполнено RX\_FIFO. При этом запись RX\_FIFO начинается с нулевой ячейки.

Число записанных в RX\_FIFO 32-разрядных слов отображается в разрядах регистра управления и состояния режима тестирования RX\_TEST\_CSR<14:4> = TM\_RX\_WRW. После сброса бита разрешения режима тестирования RX\_FIFO число записанных в RX\_FIFO слов – TM\_RX\_WRW – обнуляется.

При сбросе бита TM\_RX\_FIFO значение RXW обновляется в соответствии с числом записанных в тестовом режиме слов. После этого данные записанные в RX\_FIFO в тестовом режиме могут вычитаны по DMA-каналу из RX\_FIFO.

После сброса бита разрешения режима тестирования RX\_FIFO и последующего вычитывания по DMA-каналу тестовых данных, записанных в RX\_FIFO, для возможности дальнейшей корректной работы с RX\_FIFO необходимо выполнить сброс указателей принимающего RX\_FIFO. Для этого необходимо установить бит MAC\_CONTROL<11> = CP\_RX.

### 13.3.2 Регистры контроллера Ethernet MAC 10/100

В Таблица 13.4 приведен перечень программно-доступных регистров контроллера Ethernet MAC 10/100.

Таблица 13.4. Перечень регистров контроллера Ethernet MAC 10/100.

Условное обозначение регистра	Название регистра	Тип доступа	Исходное состояние
MAC_CONTROL[11:0]	Регистр управления MAC	WR/RD	0000_0000
MD_MODE[8:0]	Регистр режима работы порта MD	WR/RD	0000_0040
MD_CONTROL[31:0]	Регистр управления порта MD	WR/RD	0000_0000
MD_STATUS[31:0]	Регистр статуса порта MD	WR/RD	0000_0000
MAC_ADDR_L[31:0]	Регистр младшей части исходного адреса MAC	WR/RD	0000_0000
MAC_ADDR_H[15:0]	Регистр старшей части исходного адреса MAC	WR/RD	0000_0000
DADDR_L[31:0]	Регистр младшей части адреса назначения	WR/RD	0000_0000
DADDR_H[15:0]	Регистр старшей части адреса назначения	WR/RD	0000_0000
FCS_CLIENT[31:0]	Регистр контрольной суммы кадра	WR/RD	0000_0000
TYPE[15:0]	Регистр типа кадра	WR/RD	0000_0000
IFS_COLL_MODE[23:0]	Регистр IFS и режима обработки коллизии	WR/RD	18c3_401f
TX_FRAME_CONTROL[16:0]	Регистр управления передачи кадра	WR/RD	0000_0000
STATUS_TX[26:0]	Регистр статуса передачи кадра	WR/RD	0000_0000
UCADDR_L[31:0]	Регистр младшей части уникального адреса MAC	WR/RD	0000_0000
UCADDR_H[15:0]	Регистр старшей части уникального адреса MAC	WR/RD	0000_0000
MCADDR_L[31:0]	Регистр младшей части группового адреса	WR/RD	0000_0000
MCADDR_H[15:0]	Регистр старшей части группового адреса	WR/RD	0000_0000
MCADDR_MASK_L[31:0]	Регистр младшей части маски группового адреса	WR/RD	0000_0000
MCADDR_MASK_H[15:0]	Регистр старшей части маски группового адреса	WR/RD	0000_0000
HASHT_L[31:0]	Регистр младшей части хэш-таблицы	WR/RD	0000_0000
HASHT_H[31:0]	Регистр старшей части хэш-таблицы	WR/RD	0000_0000
RX_FR_MaxSize[11:0]	Регистр максимального размера принимаемого кадра	WR/RD	0000_05ee
RX_FRAME_CONTROL[9:0]	Регистр управления приема кадра	WR/RD	0000_0000
STATUS_RX[29:0]	Регистр статуса приема кадра	WR/RD	0000_0000
RX_FRAME_STATUS_FIFO [26:0]	FIFO статусов принятых кадров	WR/RD	0000_0000
TX_TEST_CSR[14:0]	Регистр управления и состояния режима тестирования TX_FIFO	WR/RD	0000_0000
TX_FIFO[31:0]	Передающее TX_FIFO	RD	0000_0000
RX_TEST_CSR[14:0]	Регистр управления и состояния режима тестирования RX_FIFO	WR/RD	0000_0000
RX_FIFO[31:0]	Принимающее RX_FIFO	WR	0000_0000

### 13.3.2.1 Регистр управления MAC (MAC\_CONTROL)

Таблица 13.5. Формат регистра управления MAC

Номер разряда	Условное обозначение	Описание
0	FULLD	Режим работы контроллера: FULLD=0 – полудуплексный режим, FULLD=1 – дуплексный режим. Доступен по чтению и записи. Значение в исходном состоянии – 0.
1	EN_TX_DMA	Разрешение работы передающего TX_FIFO с DMA-каналом. Доступен по чтению и записи. Значение в исходном состоянии – 0.
2	EN_TX	Разрешение работы блока передачи кадров. Доступен по чтению и записи. Значение в исходном состоянии – 0.
3	MASK_TX_DONE	Маска запроса на прерывание от блока передачи кадров. Доступен по чтению и записи. Значение в исходном состоянии – 0.
4	EN_RX	Разрешение работы блока приема кадров. Доступен по чтению и записи. Значение в исходном состоянии – 0.
5	LOOPBACK	Режим зацикливания блока приема кадров на блок передачи кадров.
6	FULLD_RX	Тестовый режим работы блока приема кадров, включение которого при работе контроллера в полудуплексном режиме (FULLD=0) позволяет блоку приема кадров принимать данные во время выполнения блоком передачи кадров передачи данных.
7	MASK_RX_DONE	Маска запроса прерывания по наличию принятых кадров в принимающем FIFO. Доступен по чтению и записи. Значение в исходном состоянии – 0.
8	MASK_RX_FIFO_OVF_ERR	Маска запроса прерывания по переполнению принимающего FIFO, либо переполнению FIFO статусов принятых кадров. Доступен по чтению и записи. Значение в исходном состоянии – 0.
9	CP_TX	Сброс указателей передающего TX_FIFO. Доступен по чтению и записи. После установки в 1 не доступен по записи, сбрасывается автоматически. Во время обработки запроса на передачу кадра не доступен по записи. Значение в исходном состоянии – 0.
10	RST_TX	Программный сброс блока передачи кадров контроллера. Доступен по чтению и записи. После установки в 1 не доступен по записи, сбрасывается автоматически. Значение в исходном состоянии – 0.
11	CP_RX	Сброс указателей принимающего RX_FIFO. Доступен по чтению и записи. После установки в 1 не доступен по записи, сбрасывается автоматически. Во время приема кадра не доступен по записи. Значение в исходном состоянии – 0.
12	RST_RX	Программный сброс блока приема кадров контроллера. Доступен по чтению и записи. После установки в 1 не доступен по записи, сбрасывается автоматически. Значение в исходном состоянии – 0.



### 13.3.2.2 Регистр режима работы порта MD (MD\_MODE)

Таблица 13.6. Формат регистра режима работы порта MD.

Номер разряда	Условное обозначение	Описание
7:0	MDC_Divider	Коэффициент деления системной частоты при формировании частоты MDC. Должен иметь четное, не нулевое значение. Доступен по чтению и записи. Значение в исходном состоянии – 0x40.
8	RST_MD	Программный сброс порта управления PHY. Доступен по чтению и записи. Автоматически сбрасывается после установки. Значение в исходном состоянии – 0.

### 13.3.2.3 Регистр управления порта MD (MD\_CONTROL)

Таблица 13.7. Формат регистра управления порта MD.

Номер разряда	Условное обозначение	Описание
15:0	WR_DT	Данные для записи в регистр PHY. Доступны по чтению и записи. Значение в исходном состоянии – 0000.
20:16	PHYREG_ADDR	Адрес регистра PHY. Доступен по чтению и записи. Значение в исходном состоянии – 00.
23:21	–	Резерв
28:24	PHY_ADDR	Адрес PHY. Доступен по чтению и записи. Значение в исходном состоянии – 00.
29	MD_MASK	Маска запроса на прерывание от порта управления PHY. Доступен по чтению и записи. Значение в исходном состоянии – 0.
31:30	MD_OP	Код выполняемой операции: MD_OP = 00 – состояние IDLE; MD_OP = 01 – операция чтения; MD_OP = 10 – операция записи; MD_OP = 11 – запрещенная комбинация. Доступен по чтению и записи. Значение в исходном состоянии – 00.

### 13.3.2.4 Регистр статуса порта MD (MD\_STATUS)

Таблица 13.8. Формат регистра статуса порта MD.

Номер разряда	Условное обозначение	Описание
15:0	RD_DT	Данные, прочтенные из регистра PHY. Доступны только по чтению. Значение в исходном состоянии – 0000.
28:16	–	Резерв
29	MD_BUSY	Признак занятости порта управления PHY – выполняется операция записи/чтения. Доступен только по чтению. Значение в исходном состоянии – 0.
31:30	MD_OP_END	Флаги завершения выполнения операции: MD_OP_END = 01 – завершилась операция чтения по порту MD; MD_OP_END = 10 – завершилась операция записи по порту MD. Доступны по чтению и записи. Значение в исходном состоянии – 00.

### 13.3.2.5 Регистр младшей части исходного адреса MAC (MAC\_ADDR\_L)

Таблица 13.9. Формат регистра младшей части исходного адреса MAC.

Номер разряда	Условное обозначение	Описание
31:0	MAC_ADDR_L	Младшая часть исходного адреса в поле <SOURCE ADDRESS> передаваемого кадра. Доступен по чтению и записи. Значение в исходном состоянии – 00000000.

### 13.3.2.6 Регистр старшей части исходного адреса MAC (MAC\_ADDR\_H)

Таблица 13.10. Формат регистра старшей части исходного адреса MAC.

Номер разряда	Условное обозначение	Описание
15:0	MAC_ADDR_H	Старшая часть исходного адреса в поле <SOURCE ADDRESS> передаваемого кадра. Доступен по чтению и записи. Значение в исходном состоянии – 0000.

### 13.3.2.7 Регистр младшей части адреса назначения (DADDR\_L)

Таблица 13.11. Формат регистра младшей части адреса назначения.

Номер разряда	Условное обозначение	Описание
31:0	DADDR_L	Младшая часть исходного адреса в поле <DESTINATION ADDRESS> передаваемого кадра. Доступен по чтению и записи. Значение в исходном состоянии – 00000000.

### 13.3.2.8 Регистр старшей части адреса назначения (DADDR\_H)

Таблица 13.12. Формат регистра старшей части адреса назначения.

Номер разряда	Условное обозначение	Описание
15:0	DADDR_H	Старшая часть исходного адреса в поле <DESTINATION ADDRESS> передаваемого кадра. Доступен по чтению и записи. Значение в исходном состоянии – 0000.

### 13.3.2.9 Регистр контрольной суммы кадра (FCS\_CLIENT)

Таблица 13.13. Формат регистра контрольной суммы кадра.

Номер разряда	Условное обозначение	Описание
31:0	FCS_CLIENT	Вычисленная клиентом MAC контрольная сумма передаваемого кадра CRC32. Доступен по чтению и записи. Значение в исходном состоянии – 00000000.

### 13.3.2.10 Регистр типа кадра (TYPE)

Таблица 13.14. Формат регистра типа кадра.

Номер разряда	Условное обозначение	Описание
15:0	TYPE	Если DisEncapFR = 0, то регистр задает значение поля <TYPE> передаваемого кадра. Доступен по чтению и записи. Значение в исходном состоянии – 0000.

### 13.3.2.11 Регистр IFS и режима обработки коллизии (IFS\_COLL\_MODE)

Таблица 13.15. Формат регистра IFS и режима обработки коллизии.

Номер разряда	Условное обозначение	Описание
3:0	ATTEMPT_NUM	Максимальное количество попыток повторных передач кадра. Доступен по чтению и записи. Значение в исходном состоянии – 0xF.
4	EN_CW	Разрешение отслеживания окна коллизии. Доступен по чтению и записи. Значение в исходном состоянии – 1.
6:5	–	Резерв
7	TM_BACKOFF	Включение тестового режима работы блока BACKOFF. Доступен по чтению и записи. Значение в исходном состоянии – 0.
15:8	COLL_WIN	Размер окна коллизии (число переданных байт). Доступен по чтению и записи. При записи значения $\leq 0xE$ (14 байт), автоматически устанавливается значение 0xF (15 байт). Значение в исходном состоянии – 0x40 (64 байта).
23:16	JAMB	Значение повторяющегося байта 32-разрядного jam-сообщения. Доступен по чтению и записи. Значение в исходном состоянии – 0xC3.
31:24	IFS	Значение межкадрового интервала – interFrameSpacing – в тактах частоты передачи TX_CLK. Доступен по чтению и записи. Значение в исходном состоянии – 0x18 (24 такта).

### 13.3.2.12 Регистр управления передачи кадра(TX\_FRAME\_CONTROL)

Таблица 13.16. Формат регистра управления передачи кадра.

Номер разряда	Условное обозначение	Описание
11: 0	LENGTH	<p>Если DisEncapFR = 0, то LENGTH – число байт поля &lt;DATA&gt; передаваемого кадра в передающем TX_FIFO.</p> <p>Если DisEncapFR = 1, то LENGTH – число байт передаваемого кадра в передающем TX_FIFO.</p> <p>Если DisEncapFR = 0 и TYPE_EN = 0, то LENGTH также задает значение поля &lt;LENGTH/TYPE&gt; передаваемого кадра.</p> <p>Доступен по чтению и записи. Значение LENGTH должно быть не нулевым.</p> <p>Значение в исходном состоянии – 000.</p>
12	TYPE_EN	<p>Если DisEncapFR = 0, то бит TYPE_EN задает в каком качестве используется поле &lt;LENGTH/TYPE&gt; в передаваемом кадре.</p> <p>Если TYPE_EN = 0, то – поле &lt;LENGTH&gt;;</p> <p>Если TYPE_EN = 1, то – поле &lt;TYPE&gt;.</p> <p>Доступен по чтению и записи.</p> <p>Значение в исходном состоянии – 0.</p>
13	FCS_CLT_EN	<p>Если FCS_CLT_EN = 0, то значение поля &lt;FCS&gt; вычисляет блок передачи кадров при передаче кадра;</p> <p>Если FCS_CLT_EN = 1, то значение поля &lt;FCS&gt; – уже вычисленная контрольная сумма CRC32, заданная в регистре FCS_CLIENT.</p> <p>Доступен по чтению и записи.</p> <p>Значение в исходном состоянии – 0.</p>
14	DisEncapFR	<p>Запрещает/разрешает режим формирования кадра в блоке передачи кадров.</p> <p>Если DisEncapFR = 0, то разрешен режим формирования кадра в блоке передачи кадров;</p> <p>Если DisEncapFR = 1, то в блок передачи кадров передается уже сформированный кадр.</p> <p>Доступен по чтению и записи.</p> <p>Значение в исходном состоянии – 0.</p>
15	DisPAD	<p>Запрещает/разрешает автоматическое добавление в кадр поля &lt;PAD&gt;, в случае когда число байт в поле &lt;DATA&gt; меньше 46 байт / число байт в кадре меньше 64 байт.</p> <p>Доступен по чтению и записи.</p> <p>Значение в исходном состоянии – 0.</p>
16	TX_REQ	<p>Запрос на передачу кадра. По завершении обработки запроса на передачу бит TX_REQ автоматически сбрасывается.</p> <p>Доступен по чтению и записи. Во время обработки запроса на передачу кадра бит TX_REQ не доступен по записи.</p> <p>Значение в исходном состоянии – 0.</p>

### 13.3.2.13 Регистр статуса передачи кадра (STATUS\_TX)

Таблица 13.17. Формат регистра статуса передачи кадра.

Номер разряда	Условное обозначение	Описание
0	ONTX_REQ	Блок передачи кадров выполняет обработку запроса на передачу кадра. Доступен только по чтению. Значение в исходном состоянии – 0.
1	ONTransmit	Блок передачи кадров выполняет передачу кадра. Доступен только по чтению. Значение в исходном состоянии – 0.
2	BUSY	Среда передачи занята – обнаружено наличие несущей. Доступен только по чтению. Значение в исходном состоянии – 0.
3	TX_DONE	Флаг завершения обработки запроса на передачу кадра. Доступен по чтению и записи. Во время обработки запроса на передачу кадра бит TX_DONE не доступен по записи. Значение в исходном состоянии – 0.
8:4	TX_REZ	Код результата передачи кадра: TX_REZ = 0x01 – transmitDisabled – передача не разрешена; TX_REZ = 0x02 – NotEnoughDataErr – в передающем TX_FIFO недостаточно данных для передачи; TX_REZ = 0x04 – transmitOK – передача кадра успешно выполнена; TX_REZ = 0x08 – ExcessiveCollErr – ошибка превышения максимального количества попыток повторных передач кадра; TX_REZ = 0x10 – lateCollErr – ошибка поздней коллизии; TX_REZ = 0x14 – transmitOK и lateCollErr – передача кадра прошла успешно и сразу по завершении передачи была обнаружена коллизия; Доступен только по чтению. Значение в исходном состоянии – 00.
10:9	–	Резерв
11	ONCOL	Наличие коллизии в среде передачи. Доступен только по чтению. Значение в исходном состоянии – 0.
15:12	COLL_NUM	Счетчик попыток повторных передач кадра. Доступен только по чтению. Значение в исходном состоянии – 0.
25:16	TXW	Число 64-разрядных слов в передающем TX_FIFO (округлено в большую сторону). TXW = 0x000 – FIFO пустое; TXW = 0x200 – FIFO полное. Доступен только по чтению. Значение в исходном состоянии – 000.

### 13.3.2.14 Регистр младшей части уникального адреса MAC (UCADDR\_L)

Таблица 13.18. Формат регистра младшей части уникального адреса MAC.

Номер разряда	Условное обозначение	Описание
31:0	UCADDR_L	Младшая часть уникального адреса MAC при приеме. Доступен по чтению и записи. Значение в исходном состоянии – 00000000.

### 13.3.2.15 Регистр старшей части уникального адреса MAC (UCADDR\_H)

Таблица 13.19. Формат регистра старшей части уникального адреса MAC.

Номер разряда	Условное обозначение	Описание
15:0	UCADDR_H	Старшая часть уникального адреса MAC при приеме. Доступен по чтению и записи. Значение в исходном состоянии – 0000.

### 13.3.2.16 Регистр младшей части группового адреса (MCADDR\_L)

Таблица 13.20. Формат регистра младшей части группового адреса.

Номер разряда	Условное обозначение	Описание
31:0	MCADDR_L	Младшая часть группового адреса при приеме. Доступен по чтению и записи. Значение в исходном состоянии – 00000001.

### 13.3.2.17 Регистр старшей части группового адреса (MCADDR\_H)

Таблица 13.21. Формат регистра старшей части группового адреса.

Номер разряда	Условное обозначение	Описание
15:0	MCADDR_H	Старшая часть группового адреса при приеме. Доступен по чтению и записи. Значение в исходном состоянии – 0000.

### 13.3.2.18 Регистр младшей части маски группового адреса (MCADDR\_MASK\_L)

Таблица 13.22. Формат регистра младшей части маски группового адреса.

Номер разряда	Условное обозначение	Описание
31:0	MCADDR_MASK_L	Младшая часть маски группового адреса при приеме. Доступен по чтению и записи. Значение в исходном состоянии – 00000000.

### 13.3.2.19 Регистр старшей части маски группового адреса (MCADDR\_MASK\_H)

Таблица 13.23. Формат регистра старшей части маски группового адреса.

Номер разряда	Условное обозначение	Описание
15:0	MCADDR_MASK_H	Старшая часть маски группового адреса при приеме. Доступен по чтению и записи. Значение в исходном состоянии – 0000.

### 13.3.2.20 Регистр младшей части хэш-таблицы (HASHT\_L)

Таблица 13.24. Формат регистра младшей части хэш-таблицы.

Номер разряда	Условное обозначение	Описание
31:0	HASHT_L	Младшая часть хэш-таблицы. Доступен по чтению и записи. Значение в исходном состоянии – 00000000.

### 13.3.2.21 Регистр старшей части хэш-таблицы (HASHT\_H)

Таблица 13.25. Формат регистра старшей части хэш-таблицы.

Номер разряда	Условное обозначение	Описание
31:0	HASHT_H	Старшая часть хэш-таблицы. Доступен по чтению и записи. Значение в исходном состоянии – 00000000.

### 13.3.2.22 Регистр максимального размера принимаемого кадра (RX\_FR\_MaxSize)

Таблица 13.26. Формат регистра максимального размера принимаемого кадра.

Номер разряда	Условное обозначение	Описание
11:0	RX_FR_MaxSize	Максимальный размер принимаемого кадра в байтах. Доступен по чтению и записи. Значение в исходном состоянии – 000.

### 13.3.2.23 Регистр управления приема кадра (RX\_FRAME\_CONTROL)

Таблица 13.27. Формат регистра управления приема кадра.

Номер разряда	Условное обозначение	Описание
0	Dis_RCV_FCS	Отключение сохранения поля <FCS> в принятом кадре. Доступен по чтению и записи. Значение в исходном состоянии – 0.
1	Dis_PAD_Del	Отключение удаления поля <PAD> в принятом кадре. Доступен по чтению и записи. Значение в исходном состоянии – 0.
2	Accept_TooShort	Разрешение приема слишком коротких кадров, размер которых меньше 64 байт. Доступен по чтению и записи. Значение в исходном состоянии – 0.
3	Discard_TooLong	Разрешение отбрасывания слишком длинных кадров, размер которых больше RX_FR_MaxSize. Доступен по чтению и записи. Значение в исходном состоянии – 0.
4	Discard_FCSCHErr	Разрешение отбрасывания кадров с ошибкой проверки контрольной суммы. Доступен по чтению и записи. Значение в исходном состоянии – 0.
5	Discard_LengthErr	Разрешение отбрасывания кадров с ошибкой длины поля данных. Доступен по чтению и записи. Значение в исходном состоянии – 0.
6	Dis_BC	Запрещение приема кадров с широковещательным адресом назначения. Доступен по чтению и записи. Значение в исходном состоянии – 0.
7	EN_MCM	Разрешение приема кадров с групповым адресом назначения, совпадающим с замаскированным групповым адресом назначения. Доступен по чтению и записи. Значение в исходном состоянии – 0.
8	EN_MCHT	Разрешение приема кадров с групповым адресом назначения, разрешенным для приема в хэш-таблице. Доступен по чтению и записи. Значение в исходном состоянии – 0.
9	EN_ALL	Разрешение приема кадров с любым адресом назначения. Доступен по чтению и записи. Значение в исходном состоянии – 0.

### 13.3.2.24 Регистр статуса приема кадра (STATUS\_RX)

Таблица 13.28. Формат регистра статуса приема кадра.

Номер разряда	Условное обозначение	Описание
0	RCV_Disabled	Прием не разрешен. Доступен только по чтению. Значение в исходном состоянии – 0.
1	ONReceive	Блок приема кадров выполняет прием кадра. Доступен только по чтению. Значение в исходном состоянии – 0.
2	–	Резерв
3	RX_DONE	Флаг наличия принятых кадров в принимающем RX_FIFO. Доступен только по чтению. Значение в исходном состоянии – 0.
10:4	NUM_RX_FR	Число принятых кадров. NUM_RX_FR = 0x00 => RX_DONE = 0 – FIFO статусов пустое; NUM_RX_FR ≠ 0x00 => RX_DONE = 1 – FIFO статусов не пустое; NUM_RX_FR = 0x40 – FIFO статусов полное. Доступен только по чтению. Значение в исходном состоянии – 00.
11	FR_STATUS_OVF_Err	Флаг переполнения FIFO статусов принятых кадров. Доступен по чтению и записи. Значение в исходном состоянии – 0.
21:12	RXW	Число 64-разрядных слов в принимающем RX_FIFO (округлено в меньшую сторону). RXW = 0x000 – FIFO пустое; RXW = 0x200 – FIFO полное. Доступен только по чтению. Значение в исходном состоянии – 000.
22	–	Резерв
23	RX_FIFO_OVF_Err	Флаг переполнения принимающего RX_FIFO. Доступен по чтению и записи. Значение в исходном состоянии – 0.
29:24	NUM_Missed_FR	Число пропущенных кадров из-за переполнения принимающего RX_FIFO или FIFO статусов принятых кадров. Доступен по чтению и записи. Значение в исходном состоянии – 00.

### 13.3.2.25 FIFO статусов принятых кадров (RX\_FRAME\_STATUS\_FIFO)

Статус принятого кадра RX\_FRAME\_STATUS доступен только по чтению.

Значение в исходном состоянии – 00000000.

Таблица 13.29. Формат слова FIFO статусов принятых кадров.

Номер разряда	Условное обозначение	Описание
11:0	RX_FR_LENGTH	Число байт в принятом кадре.
12	receiveOK	Флаг успешного принятия кадра без ошибок.
13	lengthError	Флаг ошибки длины поля данных в принятом кадре.
14	alignmentError	Флаг ошибки выравнивания в принятом кадре.
15	frameCheckError	Флаг ошибки при проверке принятого кадра.
16	frameTooLong	Флаг принятия слишком длинного кадра.
17	frameTooShort	Флаг принятия слишком короткого кадра.
18	DribbleNibble	Флаг поступления нечетного числа полубайт кадра.
19	LEN_FIELD	Флаг распознавания поля <LENGTH> в принятом кадре.
20	FCS_Del	Флаг удаления поля <FCS> в принятом кадре.
21	PAD_Del	Флаг удаления поля <PAD> в принятом кадре.



Номер разряда	Условное обозначение	Описание
22	UC	Флаг распознавания адреса назначения принятого кадра при совпадении с уникальным адресом MAC.
23	MCM	Флаг распознавания группового адреса назначения принятого кадра при совпадении с замаскированным групповым адресом назначения MAC, когда разрешен прием кадров с таким адресом назначения.
24	MCHT	Флаг распознавания группового адреса назначения принятого кадра разрешенного для приема в хэш-таблице, когда разрешен прием кадров с таким адресом назначения.
25	BC	Флаг распознавания ширококвещательного адреса назначения принятого кадра когда разрешен прием кадров с ширококвещательным адресом назначения.
26	ALL	Флаг распознавания адреса назначения принятого кадра, когда разрешен прием кадров с любым адресом назначения.

### 13.3.2.26 Регистр управления и состояния режима тестирования TX\_FIFO (TX\_TEST\_CSR)

Таблица 13.30. Формат регистра управления и состояния режима тестирования TX\_FIFO.

Номер разряда	Условное обозначение	Описание
0	TM_TX_FIFO	Разрешение режима тестирования TX_FIFO. Доступен по чтению и записи. Значение в исходном состоянии – 0.
3: 1	–	Резерв
14:4	TM_TX_RDW	Число прочтенных 32-разрядных слов из TX_FIFO в режиме тестирования. Доступен только по чтению. Значение в исходном состоянии – 000.

### 13.3.2.27 Регистр управления и состояния режима тестирования RX\_FIFO (RX\_TEST\_CSR)

Таблица 13.31. Формат регистра управления и состояния режима тестирования RX\_FIFO.

Номер разряда	Условное обозначение	Описание
0	TM_RX_FIFO	Разрешение режима тестирования RX_FIFO. Доступен по чтению и записи. Значение в исходном состоянии – 0.
3: 1	–	Резерв
14:4	TM_RX_WRW	Число записанных 32-разрядных слов в RX_FIFO в режиме тестирования. Доступен только по чтению. Значение в исходном состоянии – 000.

## 14. КОНТРОЛЛЕР ШИНЫ PCI

### 14.1 Функциональные параметры и возможности

Контроллер шины PCI (PMSC – PCI Master-Slave controller) имеет следующие функциональные параметры и возможности:

1. соответствует спецификации Local Bus Specification. Rev. 2.3;
2. тактовая частота – от 33 до 66 МГц;
3. шина адреса и данных - 32 разряда;
4. обмен данными в режиме Master и Target (Slave);
5. обмен данными между шиной PCI и любой областью памяти микропроцессора;
6. встроенный арбитр на 5 запросов шины;
7. встроенные средства для организации мультипроцессорных систем.

### 14.2 Структурная схема

Структурная схема PMSC приведена на Рисунок 14.1.

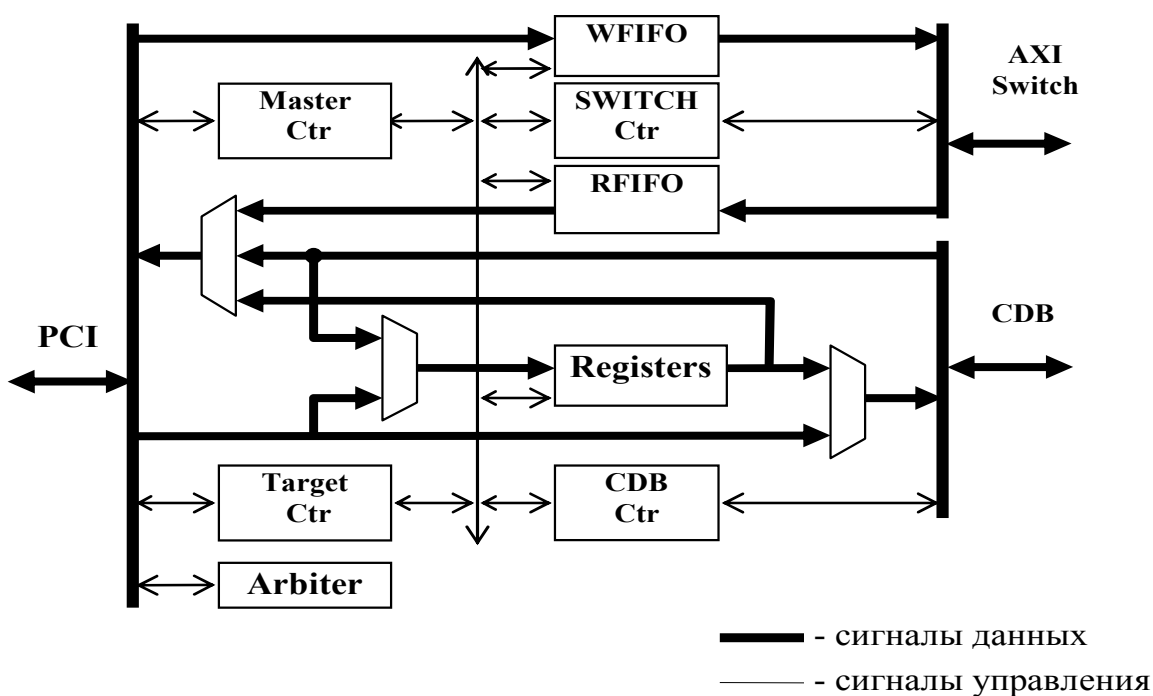


Рисунок 14.1. Структурная схема PMSC

В состав PMSC входят следующие основные узлы и компоненты:

- Registers - блок регистров, включающий:
  - конфигурационные регистры шины PCI:
    - Device ID/Vendor ID, Status/Command, Class Code/Revision ID, Subsystem ID/Subsystem Vendor ID, BAR0, BAR1, Latency Timer, Interrupt Line;
  - регистры управления обменом: AR\_PCI, IR\_MASTER, IR\_TARGET, CSR\_MASTER, CSR\_PCI, STATUS\_MASTER, TMR\_PCI, CSR\_WIN;
  - регистры передачи векторов прерывания: MBR\_CPU, MBR\_PCI и SEM;
  - регистры управления прерываниями: регистр системных прерываний QSTR\_PCI и регистр маскирования прерываний MASKR\_PCI.
- CDB Ctr - блок управления обменом данными с регистрами по шине CDB;
- SWITCH Control - блок управления обменом данными с коммутатором AXI Switch;
- Target Ctr – блок управления передачи данных между шиной PCI и памятью через коммутатор AXI Switch в режиме Target (Slave);
- Master Ctr – блок управления передачи данных между шиной PCI и памятью через коммутатор AXI Switch в режиме Master;
- WFIFO – буфер FIFO объемом 16 32-разрядных слов для передачи данных из шины PCI в коммутатор AXI Switch;
- RFIFO – буфер FIFO объемом 16 32-разрядных слов для передачи данных из коммутатора AXI Switch в шину PCI;
- Arbiter – арбитр шины PCI.

### 14.3 Регистры

Перечень регистров PMSC, доступных со стороны шин PCI и CDB приведён в Таблица 14.1.

.Таблица 14.1. Перечень программно доступных регистров PMSC

Условное обозначение регистра	Название регистра	Смещение адреса	Исходное состояние
Device ID/ Vendor ID	Регистр идентификации устройства	0x00	0x680C2001
Status/Command	Регистр состояния и управления	0x04	0x02800000
Class Code/Revision ID	Регистр кода классификации	0x08	0x07800001
Latency Timer	Регистр времени транзакции в режиме Master	0x0C	0x00000000
BAR0 (Base Address Register 0)	Регистр базового адреса 0	0x10	0x00000008
BAR1 (Base Address Register 1)	Регистр базового адреса 1	0x14	0x00000008
Subsystem ID/ Subsystem Vendor ID	Регистр идентификации подсистемы	0x2C	0x00000002
Interrupt Line	Код прерывания	0x3C	0x01200100
IR_TARGET	Регистр адреса памяти в режиме Target	0x40	0x00000000
SEM	Регистр семафора	0x44	0x00000000
MBR_PCI	Регистр почтового ящика шины PCI	0x48	0x00000000
CSR_PCI	Регистр управления и состояния шины PCI	0x4C	0x00000000
CSR_MASTER	Регистр управления режимом Master	0x50	0x00000000
IR_MASTER	Регистр адреса памяти в режиме Master	0x54	0x00000000

Условное обозначение регистра	Название регистра	Смещение адреса	Исходное состояние
AR_PCI	Регистр адреса шины PCI в режиме Master	0x58	0x00000000
QSTR_PCI	Регистр системных прерываний.	0x5C	0x00000000
MASKR_PCI	Регистр маскирования прерываний	0x60	0x00000000
STATUS_MASTER	Регистр состояния обмена с шиной PCI в режиме Master	0x64	0x00000000
TMR_PCI	Регистр параметров	0x68	0x00000000
CSR_WIN	Регистр управления обменом с PCI через адресное окно	0x6C	0x00000000
MBR_CPU	Регистр почтового ящика CPU	0x70	0x00000000

Смещение адреса определяется разрядами адреса 7:0 шин CDB и PCI и указано относительно базового адреса PMSC в адресном пространстве микропроцессора.

Обмен данными с регистрами осуществляется 32-разрядными словами. Если разряды регистра не используются, то из них считываются нули. При записи в этих разрядах необходимо указывать нули.

По шине CDB все регистры доступны по записи и чтению, кроме регистров QSTR\_PCI и STATUS\_MASTER. Эти регистры доступны только для чтения.

По шине PCI доступ к регистрам PMSC осуществляется в режиме Target по командам Configuration Read, Configuration Write в области адресов Type 0 и по командам Memory Read, Memory Write. При обмене данными с регистрами PMSC транзакция завершается сигнализацией Disconnect при передаче первого слова.

По чтению все регистры доступны по командам Configuration Read, Memory Read.

Регистры Status/Command, BAR0, BAR1, Interrupt\_Line, IR\_MASTER, IR\_TARGET, AR\_PCI, MASKR\_PCI, доступны по записи по командам Configuration Write и Memory Write.

Регистры CSR\_MASTER, CSR\_PCI, MBR\_PCI, SEM, TMR\_PCI доступны для записи только по команде Memory Write.

При описании полей и значений регистров используются обозначения:

- R – разрешено только чтение;
- RW – разрешены чтение и запись;
- RW0 – разрешены чтение и запись, при записи единицы разряд обнуляется;
- [i] – номер разряда;
- i:j – неразрывная группа разрядов, i –старший разряд группы, j –младший;
- 0x – далее следует код в шестнадцатеричной системе счисления;
- PCLK – тактовая частота шины PCI;
- AD – разряды адреса/данных шины PCI.

### 14.3.1 Конфигурационные регистры

#### 14.3.1.1 Регистры Device/Vendor ID, Class Code/Revision ID и Subsystem ID/Subsystem Vendor ID

32-разрядные регистры Device/Vendor ID, Class Code/Revision ID и Subsystem ID/Subsystem Vendor ID предназначены для хранения кодов в соответствии со спецификацией PCI. Исходное состояние регистров: Device/Vendor ID – 0x680c2001, Class Code/Revision ID – 0x07800001, Subsystem ID/Subsystem Vendor ID – 0x00000002.

По шине PCI эти регистры доступны только по чтению, их состояние можно изменить только процессор при инициализации PMSC.

#### 14.3.1.2 Регистры BAR0 и BAR1

Регистры BAR0 и BAR1 определяют базовый адрес PMSC на шине PCI в режиме Target при выполнении команд Memory Read, Memory Write:

- регистр BAR0 используется для обмена данными с регистрами PMSC и внутренней памятью (CRAM и XRAM, YRAM, PRAM любого из ядер DSP);
- регистр BAR1 используется для обмена данными с любой областью памяти.

Разряды 31:26 регистров BAR0, BAR1 доступны по записи и чтению, их содержимое устанавливается программно при инициализации PMSC (процессором или внешним контроллером PCI). Разряды 25:0 этих регистров доступны только по чтению кода 0x000\_0008, что является индикатором пространства памяти объемом 64 Мбайт.

PMSC выполняет команды Memory Read, Memory Write при  $AD[31:26]=BAR0[31:26]$  или  $AD[31:26]=BAR1[31:26]$  в фазе адреса шины PCI. При невыполнении этого условия PMSC установкой высокого уровня сигнала nDEVSEL инициирует завершение команд Memory Read, Memory Write по условию “Master abort”.

#### 14.3.1.3 Регистр Status/Command

Формат регистра Status/Command приведен в Таблица 14.2.

Таблица 14.2. Формат регистра Status/Command

Номер разряда	Условное обозначение	Описание	Доступ
31	Detected Parity Error	Признак обнаружения ошибки чётности. В режиме Target устанавливается при обнаружении ошибки чётности либо в фазе адреса, либо в фазе данных при выполнении транзакции запись. В режиме Master устанавливается при обнаружении ошибки чётности в фазе данных при выполнении транзакции чтение	RW0
30	-	Не используется.	
29	Received Master Abort	Транзакция в режиме Master завершается установкой признака Received Master Abort, если в течение пяти тактов частоты PCLK после выдачи nFRAME = 0 сохранялся высокий уровень сигнала nDEVSEL	RW0
28	Received Target Abort	Транзакция в режиме Master завершается установкой признака Received Target Abort при обнаружении в фазе передачи данных ситуации nSTOP = 0, nDEVSEL = 1	RW0

Номер разряда	Условное обозначение	Описание	Доступ
27	Signaled Target Abort	Признак Signaled Target Abort устанавливается в режиме Target, если в данном регистре бит Parity Error Response установлен в 1, в регистре CSR_PCI бит Target Parity Stop установлен в 1 и обнаружена ошибка чётности в фазе адреса. При этом PMSC завершает транзакцию установкой nSTOP = 0, nDEVSEL = 1	RW0
26:25	DEVSEL timing	Конфигурационный параметр. Определяет задержку выдачи сигнала nDEVSEL в тактах PCLK. PMSC обеспечивает задержку класса “medium”: сигнал nDEVSEL выдается через один такт после обнаружения низкого уровня сигнала nFRAME	R
24	Master Data Parity Error	Признак выдачи или обнаружения сигнала nPERR в режиме Master при условии Parity Error Response = 1	RW0
23	Fast Back-to-Back Capable	Конфигурационный параметр. Определяет способность PMSC выполнять транзакции типа “Fast Back-to-Back” если они выполняются не одним и тем же устройством	R
22:20	-	Не используется	R
19	Interrupt Status	Этот бит отражает наличие незамаскированных прерываний в PMSC. Только тогда когда в данном регистре бит Interrupt Disable установлен в 0, а бит Interrupt Status установлен в 1 будет установлен низкий уровень сигнала nINTA. Установка бита Interrupt Disable в 1 не влияет на состояние бита Interrupt Status	R
18:11	-	Не используется	R
10	Interrupt Disable	Запрещение формирования сигнала nINTA: при Interrupt Disable = 1 на выводе nINTA устанавливается пассивный высокий уровень; при Interrupt Disable = 0 и Interrupt Status = 1 на выводе nINTA устанавливается активный низкий уровень.	RW
9:7	-	Не используется	R
6	Parity Error Response	Разрешение формирование сигнала nPERR	RW
5:3	-	Не используются	R
2	Bus Master	Разрешение работы на шине PCI в режиме Master	RW
1	Memory Space	Разрешение выполнение в режиме Target команд Memory Read, Memory Write	RW
0	-	Не используется	R

Разряды 29: 28 обнуляются при запуске на шине PCI транзакции в режиме Master.

#### 14.3.1.4 Регистр Latency Timer

Формат регистра Latency Timer приведен в Таблица 14.3.

Таблица 14.3. Формат регистра Latency Timer

Номер разряда	Условное обозначение	Назначение	Доступ
31:16	-	Не используется	R
15:8	MLT	Конфигурационная переменная. Определяет время в тактах PCLK, отведенное PMSC для выполнения транзакции в режиме Master. Устанавливается при инициализации шины PCI	RW
7:0	-	Не используется	R

### 14.3.1.5 Регистр Interrupt Line

Формат регистра Interrupt Line приведен в Таблица 14.4.

Таблица 14.4. Формат регистра Interrupt Line

Номер разряда	Условное обозначение	Назначение	Доступ
31:24	Max_Lat	Определяет максимальное время между двумя передачами данных PMSC по шине PCI. Max_Lat = 0x01. Цена одного разряда - 0,25 мкс	R
23:16	Min_Gnt	Содержит минимальную величину времени, на которую PMSC занимает шину PCI при передаче данных. Min_Gnt = 0x20. Цена одного разряда - 0,25 мкс	R
15:8	Interrupt Pin	Указывает, что выход прерывания PMSC подключен к линии nINTA. Interrupt Pin = 0x01	R
7:0	Interrupt Line	Используется для реализации системных функций на PCI. Устанавливается при инициализации шины PCI	RW

## 14.3.2 Регистры управления обменом данными

### 14.3.2.1 Регистр CSR\_MASTER

Регистр CSR\_MASTER предназначен для задания параметров и запуска процедуры передачи блока данных по шине PCI в режиме Master.

Запуск процедуры передачи осуществляется записью 1 в разряд 0 регистра CSR\_MASTER. Запись в регистр CSR\_MASTER разрешена при CSR\_MASTER [0]=0.

Формат регистра CSR\_MASTER приведен в Таблица 14.5.

Таблица 14.5. Формат регистра CSR\_MASTER

Номер разряда	Условное обозначение	Назначение
31:16	WC	Размер блока данных – количество 32-разрядных слов, которое должен передать PMSC в режиме Master: 0x0 – 1 слово, 0x1 – 2 слова, ..., 0xFFFF - 65536 слов. Содержимое этого поля в процессе передачи данных не изменяется
15	DONE	Индикатор завершения процедуры передачи блока данных. Устанавливается в 0 при записи 1 в бит RUN
14	Fatal Error	Индикатор останова передачи блока данных по фатальной ошибке. Устанавливается в 0 при записи 1 в бит RUN
13	Break Done	Индикатор выполнения программного останова передачи блока данных Устанавливается в 0 при записи 1 в бит RUN
12	WINDOW	Индикатор выполнения обмена через адресное окно. Устанавливается в 1 при выполнении обмена через адресное окно. Устанавливается в 0 при записи 1 в бит RUN
11:8	WNM	Количество слов, которое должно накопиться в буфере WFIFO для передачи очередной порции данных в коммутатор SWITCH в режиме Master. Определяется из системных соображений, с точки зрения максимальной скорости передачи данных. Допустимые значения: 0, 7, 0xF. Если WC меньше или равно 0xF, необходимо определить WNM=0xF, в противном случае – выбрать WNM=7 или WNM=0. При WNM = 0,1, 2, 3 передача в коммутатор SWITCH производится пачками по 4 слова
7	sel_MILat	Разрешение изменения параметра "Master Initial Latency".
6	sel_MSLat	Разрешение изменения параметра "Master Subsequent Latency"
5	sel_IRDLat	Разрешение изменения параметра "nIRDY Latency"



Номер разряда	Условное обозначение	Назначение
4:1	CMD	Тип команды при передаче данных в режиме Master: 0010 – I/O Read; 0011 – I/O Write; 0110 – Memory Read; 0111 – Memory Write; 1010 – Configuration Read; 1011 – Configuration Write. 1100 – Memory Read Multiple; 1110 – Memory Read Line; 1111 – Memory Write and Invalidate. Эти разряды передается на выводы nCBE[3:0] в фазе адреса. В фазе данных на этих выводах устанавливается значение 0x0
0	RUN	Режим работы PMSC: 1 – выполняется процедура передачи блока данных; 0 – передача блока данных завершена или не запущена. Устанавливается в 1 при записи 1 в этот разряд. Устанавливается в 0, если блок данных передан полностью или транзакция завершилась либо по фатальной ошибке, либо по программному останову

#### 14.3.2.2 Адресные регистры обмена

32-разрядный регистр AR\_PCI предназначен для указания начального адреса на шине PCI при выполнении транзакции на шине PCI в режиме Master. При выполнении конфигурационных операций разряды AR\_PCI[1:0] определяют тип обмена (Type0 или Type1), а унитарный код в разрядах AR\_PCI[31:11] указывает IDSEL адресуемого устройства. Разряды AR\_PCI[10:2] должны быть установлены в соответствии со спецификацией Local Bus Specification Rev. 2.3 для адресуемого устройства

32-разрядный регистр IR\_MASTER хранит начальный физический адрес памяти при выполнении транзакции на шине PCI в режиме Master.

Содержимое регистров AR\_PCI и IR\_MASTER в процессе обмена данными не изменяется.

Регистр IR\_TARGET хранит начальный физический адрес памяти при выполнении команд Memory Read, Memory Write через BAR1. Содержимое регистра IR\_TARGET в процессе обмена данными не изменяется.

При обменах через BAR1 разряды 31:26 адреса шины SWITCH определяются разрядами 31:26 регистра IR\_TARGET, остальные - разрядами AD[25: 0] в фазе адреса шины PCI. Для обмена с областью памяти, подключенной к nCS[3], необходимо записать в IR\_TARGET код 0x1c000000, а для обмена с областью внутренней памяти - код 0x18000000.

При обменах через BAR0 контроллер аппаратно устанавливает в разрядах 31:26 адреса шины SWITCH базовый адрес внутренней памяти (0x06), остальные разряды начального адреса памяти определяются разрядами AD[25: 0] в фазе адреса шины PCI.



### 14.3.2.3 Регистр STATUS\_MASTER

Регистр STATUS\_MASTER предназначен контроля передачи блока данных в режиме Master. Формат регистра STATUS\_MASTER приведен в таблице 15.6.

Таблица 14.6. Формат регистра STATUS\_MASTER

Номер разряда	Условное обозначение	Назначение
31	Master Read Parity Error	Ошибка при выполнении чтения на шине PCI. Устанавливается в 1, если при установленных битах Parity Error Response регистра Status/Command и Master Parity Stop регистра CSR_PCI обнаружена ошибка чётности
30	Master Write Parity Error	Ошибка при выполнении записи на шине PCI. Устанавливается в 1, если при установленных битах Parity Error Response регистра Status/Command и Master Parity Stop регистра CSR_PCI обнаружен низкий уровень сигнала nPERR
29	Received Master Abort	Состояние признака Received Master Abort в регистре Status/Command
28	Received Target Abort	Состояние признака Received Target Abort в регистре Status/Command
27	No Trdy	Транзакция завершается установкой признака No Trdy при отсутствии сигналов nTRDY или nSTOP в течение времени "Master Initial Latency" после начала передачи
26	No Gnt	Признак отсутствия сигнала nGNT в течение 4095 тактов шины PCI после установки сигнала nREQ
25	Break Done	Состояние признака Break Done в регистре CSR_MASTER
24	Disconnect	Транзакция завершается установкой признака Disconnect при обнаружении ситуации nSTOP=0, nDEVSEL=0 в фазе передачи данных
23	Retry	Транзакция завершается установкой признака Retry при обнаружении ситуации nSTOP=0, nDEVSEL=0, nTRDY=1 в интервале времени "Master Initial Latency". Если Master Break=0, то данная транзакция будет повторяться до тех пор, пока она успешно не завершится и передача последующих данных продолжится. Если Master Break=1, то данная транзакция повторяться не будет, и передача данных прекратится
22	Timeout	Транзакция завершается установкой признака Timeout, если при высоком уровне сигнала nGNT длительность транзакции (в тактах PCLK) превысила лимит, установленный разрядами MLT регистра Latency Timer
21	IRDYout	Транзакция завершается установкой признака IRDYout при отсутствии сигнала nIRDY в течение времени "IRDY Latency" после завершения очередной фазы передачи данных
20	TRDYout	Транзакция завершается установкой признака TRDYout при отсутствии сигналов nTRDY или nSTOP в течение времени "Master Subsequent Latency" после завершения очередной фазы данных
19:17	-	Не используется
16	RUN	Состояние признака RUN в регистре CSR_MASTER
15:0	WCC	Текущий размер блока данных: перед выполнением первой транзакции $WCC = WC$ ; после передачи по шине PCI очередного слова $WCC = WCC - 1$ ; после передачи последнего слова блока данных $WCC = 0$

Разряды 31:20 определяют причину окончания транзакции передачи данных в режиме Master. Эти разряды аппаратно обнуляются при запуске транзакции.

#### 14.3.2.4 Регистр параметров TMR\_PCI

Регистр TMR\_PCI используется для хранения временных параметров выполнения транзакции на шине PCI в режиме Master.

Формат регистра TMR\_PCI приведен в Таблица 14.7.

Таблица 14.7. Формат регистра TMR\_PCI

Номер разряда	Условное обозначение	Назначение
31:16	WaterMark	Пороговое значение для формирования прерывания WMARK: прерывание формируется после передачи WaterMark слов по шине PCI 0x0 – 1 слово, 0x1– 2 слова , ..., 0xFFFF – 65536 слов;
15:12	-	Не используется
12: 8	MILat	Master Initial Latency в тактах PCLK
7:4	MSLat	Master Subsequent Latency в тактах PCLK
3: 0	IRDLat	nIRDY Latency в тактах PCLK

Если в регистре CSR\_MASTER sel\_MILat = 1 и MILat > 1, то используется MILat.

Если в регистре CSR\_MASTER sel\_MSLat =1, то используется MSLat.

Если в регистре CSR\_MASTER sel\_IRDLat =1, то используется IRDLat.

В противном случае используются значения, регламентированные в Local Bus Specification. Rev.2.3:

- Master Initial Latency = 0xF (16 тактов);
- Master Subsequent Latency = 7 (8 тактов);
- nIRDY Latency = 7(8 тактов).

Параметр nIRDY Latency определяет количество тактов ожидания (Waite State), которое может вставить PMSC при выполнении транзакции в режиме Master.

### 14.3.2.5 Регистр CSR\_WIN

Регистр CSR\_WIN предназначен для задания параметров выполнения обменов с шиной PCI через адресное окно.

Формат регистра CSR\_WIN приведен в Таблица 14.8.

Таблица 14.8. Формат регистра CSR\_WIN

Номер разряда	Условное обозначение	Назначение
31:24	AR_WIN	если SEL_ADR =1, то состояние этого поля передается на выходы AD[31:24] в фазе адреса
23:20	nCBE_WIN	если SEL_nCBE =1, то состояние этого поля передается на выходы nCBE[3:0] в фазе данных
19:17	-	Не используется
16	MASK_DPE	Разрешение формирования прерывания MASTER_ERROR по ошибке чётности: 1- разрешено; 0- запрещено
15:13	-	Не используется
12:8	MIL_WIN	Значение "Master Initial Latency" в тактах PCLK
7	SEL_MIL	Выбор параметра "Master Initial Latency": 0 – используется номинальное значение данного параметра; 1 – используется значение, определенное полем MIL_WIN этого регистра
6	SEL_ADR	Выбор адреса: 0 – на выводах AD[31:24] в фазе адреса данных устанавливается значение, определенное разрядами 31:24 регистра AR_PCI; 1 – на выводах AD[31:24] в фазе адреса данных устанавливается значение, определенное разрядами AR_WIN этого регистра
5	SEL_nCBE	Выбор nCBE: 0 – на выводах nCBE в фазе данных устанавливается нулевое значение; 1 – на выводах nCBE в фазе данных устанавливается значение, определенное разрядами nCBE_WIN этого регистра
4:1	CMD_WIN	Тип команды: 0xA или 0xB – выполняются команды типа Configuration; 0x2 или 0x3 – выполняются команды типа I/O; при остальных значениях данного поля выполняются команды Memory Write, Memory Read
0	-	Не используется

### 14.3.2.6 Регистр CSR\_PCI

Формат регистра CSR\_PCI приведен в Таблица 14.9. .

Таблица 14.9. Формат регистра CSR\_PCI

Номер разряда	Условное обозначение	Назначение
31	Master Read Parity Error	Состояние признака Master Read Parity Error в регистре STATUS_MASTER
30	Master Write Parity Error	Состояние признака Master Write Parity Error в регистре STATUS_MASTER
29	Received Master Abort	Состояние признака Received Master Abort в регистре Status/Command
28	Received Target Abort	Состояние признака Received Target Abort в регистре Status/Command

Номер разряда	Условное обозначение	Назначение
27	No Trdy	Состояние признака No Trdy в регистре STATUS_MASTER
26	No Gnt	Состояние признака No Gnt в регистре STATUS_MASTER
25	Break Done	Состояние признака Break Done в регистре CSR_MASTER
24	Master Break	Программный останов передачи блока данных: 0 – разрешение выполнения транзакций передачи данных; 1 – запрещение выполнения транзакций передачи данных
23	mst_DPErd	Устанавливается в 1, если обнаружена ошибка чётности при выполнении чтения на шине PCI в режиме Master. Обнуляется при записи 1 в разряд RUN регистра CSR_MASTER. Доступ: только чтение
22	mst_DPEwr	Устанавливается в 1, если бит Parity Error Response регистра Status/Command установлен в 1 и обнаружен низкий уровень сигнала nPERR при выполнении записи на шине PCI в режиме Master. Обнуляется при записи 1 в разряд RUN регистра CSR_MASTER. Доступ: только чтение
21	-	Не используется
20	Master Parity Stop	Разрешение прекращения передачи блока данных и формирования прерывания MASTER_ERROR при обнаружении ошибки чётности в режиме Master.
19	tgt_DPEd	Устанавливается в 1, если обнаружена ошибка чётности в фазе данных при выполнении записи на шине PCI в режиме Target. Устанавливается в 0 при чтении этого регистра. Доступ: только чтение
18	tgt_DPEa	Устанавливается в 1 в режиме Target, если в фазе адреса обнаружена ошибка чётности. Устанавливается в 0 в режиме Target, если в фазе адреса ошибка чётности не обнаружена. Устанавливается в 0 при чтении этого регистра. Доступ: только чтение
17	-	Не используется
16	Target Parity Stop	Разрешение завершения транзакции в режиме Target установкой признака Signaled Target Abort при обнаружении ошибки чётности в фазе адреса
15:12	TILat	Target Initial Latency в тактах PCLK. Номинальное значение TILat = 0xC соответствует 16 тактам шины PCI
11:8	TSLat	Target Subsequent Latency в тактах PCLK. Номинальное значение TSLat = 0x6 соответствует 8 тактам шины PCI
7	Test par	Режим формирования выходного сигнала PAR: 0 – сигнал формируется в соответствии с Local Bus Specification Rev.2.3; 1 – формируется инверсное значение сигнала. Используется для тестирования PMSC
6	Test perr	Режим формирования выходного сигнала nPERR: 0 – сигнал формируется в соответствии с Local Bus Specification Rev.2.3; 1 – в режиме Target формируется инверсное значение сигнала. Используется для тестирования PMSC
5	sel_TILat	Разрешение изменения параметра ”Target Initial Latency”: 0 – используется номинальное значение данного параметра; 1 – используется значение, определенное полем TILat этого регистра
4	sel_TSLat	Разрешение изменения параметра ”Target Subsequent Latency”: 0 – используется номинальное значение данного параметра; 1 – используется значение, определенное полем TSLat этого регистра
3:0	WNT	Количество слов, которое должно накопиться в буфере WFIFO для передачи очередной порции данных в коммутатор SWITCH в режиме Target. Определяется из системных соображений, с точки зрения максимальной скорости передачи данных. Допустимые значения: 0, 7, 0xF. Если предполагаемая длина транзакции меньше или равна 16 слов, следует определить WNT = 0xF, в противном случае – выбрать WNT = 7 или WNT = 0.

Конфигурационная запись в этот регистр игнорируется.

Значение поля WNT используется при записи данных в память в режиме Target: очередная порция данных в коммутатор AXI Switch передается при накоплении в WFIFO не менее WNT слов. При WNT = 0,1, 2, 3 передача в коммутатор SWITCH производится при накоплении в WFIFO не менее четырех слов. После окончания транзакции размер порции определяется количеством оставшихся в WFIFO слов.

При чтении данных из памяти в режиме Target передача в шину PCI начинается по появлению первого слова в RFIFO.

#### 14.4 Обмен данными по шине PCI в режиме Target (Slave)

В режиме Target PMSC обеспечивает доступ с шины PCI к регистрам PMSC и памяти микропроцессора. Обмен данными с памятью микропроцессора выполняется в режиме прямого доступа через AXI Switch.

Регистры PMSC доступны по командам Configuration Read, Configuration Write в области адресов Type 0 и по командам Memory Read, Memory Write. Все транзакции обмена данными с регистрами PMSC выполняются за три такта шины PCI и завершаются после передачи первого слова установкой требования “Disconnect” (низкий уровень сигнала nSTOP).

Обмен данными с памятью выполняется по командам Memory Read, Memory Write. Команды Memory Read Multiple, Memory Read Line исполняются как Memory Read, а Memory Writeand Invalidate – как Memory Write

Объем адресуемого пространства PMSC на шине PCI составляет 128 Мбайт. Оно разбито на два окна по 64 Мбайт каждое. Адрес ячейки памяти или регистра PMSC в окне определяется разрядами AD[25:0] в фазе адреса шины PCI.

Первое окно доступно при AD[31:26] = BAR0[31:26] в фазе адреса шины PCI. Оно отображается на область регистров PMSC при AD[23:16] = 0x2F и на внутреннюю память микропроцессора в остальных случаях. При обменах с регистрами состояние разрядов AD[25:24] и AD[15:8] в фазе адреса шины PCI безразлично. При обращении в зарезервированные области внутренней памяти или в область окна выхода на шину PCI данные при записи теряются, а при чтении недостоверны.

Второе окно доступно при AD[31:26] = BAR1[31:26] в фазе адреса шины PCI. Базовый адрес этого окна в адресном пространстве микропроцессора определяется разрядами 31:26 регистра IR\_TARGET, что позволяет адресовать любую область памяти.

При BAR1 = BAR0 обмен производится с памятью окна BAR0.

Обмен данными производится 32-разрядными словами независимо от состояния выводов nCBE в фазе данных.

Передача данных в режиме Target регламентируется битами управления и параметрами регистров Status/Command и CSR\_PCI, а также битом RUN регистра CSR\_MASTER.

Если бит Memory Space регистра Status/Command установлен в 0, то PMSC инициирует завершение команд Memory Read, Memory Write по условию “Master abort” установкой высокого уровня сигнала nDEVSEL. Состояние бита Memory Space не влияет на выполнение конфигурационных операций.

Если бит Target Parity Stop регистра CSR\_PCI установлен в 1, и обнаружена ошибка чётности в фазе адреса, то PMSC инициирует завершение обмена по условию “Target abort” установкой высокого уровня сигнала nDEVSEL и низкого уровня сигнала nSTOP.

Если бит RUN регистра CSR\_MASTER установлен в 1 (PMSC занят передачей блока данных в режиме Master), то PMSC как устройство “Target” инициирует завершение обмена с памятью по условию “Retry” установкой высокого уровня сигнала nTRDY и низкого уровня сигнала nSTOP. Состояние бита RUN не влияет на выполнение операций с регистрами PMSC.

Если значение AD[1:0] в фазе адреса (Burst Order) отлично от нуля, то PMSC инициирует завершение обмена по условию “Disconnect” установкой низкого уровня сигнала nSTOP после передачи первого слова.

## 14.5 Обмен данными с шиной PCI в режиме Master

В режиме Master PMSC обеспечивает обмен данными между памятью микропроцессора и шиной PCI. Обмен данными с памятью микропроцессора выполняется в режиме прямого доступа через AXI Switch..

В режиме Master на шине PCI могут выполняться команды: I/O Read, I/O Write, Memory Read, Memory Write, Configuration Read, Configuration Write, Memory Read Multiple, Memory Read Line, Memory Write and Invalidate. Код выполняемой команды определяется полем CMD регистра CSR\_MASTER.

Команды Memory Read Multiple и Memory Read Line выполняются как Memory Read, а команда Memory Write and Invalidate – как Memory Write. В зависимости от содержимого разрядов AR\_PCI[1:0] могут выполняться конфигурационные операции Type 0 и Type 1.

Передача данных регламентируется параметрами, установленными регистрами CSR, Latency Timer и регистрами управления обменом.

Запуск процедуры передачи осуществляется записью 1 в разряд 0 регистра CSR\_MASTER через адаптер MBA, или с шины PCI в режиме Target.

Для запуска с шины PCI необходимо предварительно установить в 1 бит Bus Master в регистре Status/Command. При запуске через адаптер MBA состояние бита Bus Master несущественно.

Перед запуском необходимо убедиться в том, что PMSC не выполняет передачу данных в режиме Master: в регистре CSR\_MASTER бит RUN установлен в 0.

Затем необходимо записать:

- начальный адрес абонента микросхемы 1892XD1Я в регистр IR\_MASTER;
- начальный адрес устройства на шине PCI в регистр AR\_PCI;
- параметры WaterMark, MlLat, MSLat, IRDLat в регистр TMR\_PCI;
- параметры WC, WNM, sel\_MlLat, sel\_MSLat, sel\_IRDLat, CMD и бит RUN=1 в регистр CSR\_MASTER.

После записи 1 в разряд RUN регистра CSR\_MASTER PMSC выполняет следующую процедуру передачи блока данных:

1. PMSC формирует запрос на шину PCI, устанавливая низкий уровень на выходе nREQ, и ожидает разрешения на занятие шины от арбитра шины (низкий уровень сигнала nGNT).
2. Если разрешение не поступило в течение 4095 тактов шины PCI, передача завершается по фатальной ошибке No Gnt:
  - в регистре CSR\_MASTER: DONE = 1, Fatal Error =1, RUN=0;
  - в регистре STATUS\_MASTER: No Gnt =1, RUN=0;
  - в регистре QSTR\_PCI: MASTER\_DONE=0, MASTER\_ERROR=1.
3. Если разрешение на занятие шины получено, PMSC снимает запрос nREQ и запускает выполнение транзакции на шине PCI установкой низкого уровня сигнала nFRAME.



4. Если после выполнения транзакции текущий размер блока данных (WCC) отличен от нуля, то PMSC продолжает передачу блока данных, формируя очередной запрос на шину PCI. Процедура передачи блока данных продолжается до тех пор, пока не будет передано последнее слово.
5. Если передача блока данных завершается по ошибке чётности (в регистре CSR\_PCI установлен бит Master Parity Stop, а в регистре STATUS\_MASTER установлен бит Master Read Parity Error или бит Master Write Parity Error), то при чтении передано с ошибкой только последнее слово, а при записи – последнее и, возможно, предпоследнее.
6. Если перед выполнением транзакции в регистре CSR\_PCI был установлен в 1 бит Master Break, то после ее выполнения передача данных прекращается. При этом, в регистре CSR\_MASTER: DONE=1, Fatal Error=0, Break Done=1, RUN=0;
7. Если транзакция завершается по фатальной ошибке (установлен в 1 один из разрядов STATUS\_MASTER [31:26]), то передача блока данных прекращается:
  - в регистре CSR\_MASTER: DONE=1, Fatal Error =1, RUN=0;
  - в регистре QSTR\_PCI: MASTER\_DONE=0, MASTER\_ERROR=1.
8. Если в результате выполнения транзакции передано последнее слово данных, то передача блока данных завершается:
  - в регистре CSR\_MASTER: DONE=1, Fatal Error=0, RUN=0;
  - в регистре QSTR\_PCI: MASTER\_DONE=1, MASTER\_ERROR=1 при mst\_DPErd =1 или mst\_DPEwr=1.

При обнаружении фатальной ошибки необходимо передать управление системной программой.

В промежутках между транзакциями по шине PCI доступны для чтения и записи только регистры контроллера PMSC. При обращении к другим абонентам микропроцессора PMSC как устройству “Target” завершает транзакцию требованием повтора передачи “Retry” (nSTOP=0, nTRDY=1 в начальной фазе данных).

## 14.6 Обмен данными с шиной PCI через адресное окно

Микропроцессор может обмениваться данными с шиной PCI по операциям записи и чтения. Для этого в адресном пространстве выделено адресное окно в диапазоне адресов от 1B000\_0000 до 1BF\_FFFF.

Перед выполнением обмена необходимо убедиться, что PMSC не выполняет передачу данных в режиме Master: в регистре CSR\_MASTER бит RUN установлен в 0. Затем необходимо записать в регистр CSR\_WIN значения параметров SEL\_ADR, AR\_WIN, SEL\_nCBE, CBE\_WIN, SEL\_MIL, MIL\_WIN, CMD\_WIN, MASK\_DPE.

При обращении в окно шины PCI в регистре CSR\_MASTER аппаратно устанавливаются нулевые значения параметров WC, WNM, sel\_MSLat, sel\_IRDLat, единичные значения признаков RUN и WINDOW, и выполняется процедура передачи блока данных (см. п. 6.5).

Если транзакция завершается по фатальной ошибке (установлен в 1 один из разрядов STATUS\_MASTER [31:26]), то

- в регистре CSR\_MASTER: DONE = 1, Fatal Error =1, Break Done =0, WINDOW=1; RUN=0;

- в регистре QSTR\_PCI: DONE =0, ERROR = 1.

Если транзакция завершается передачей данных то:

- STATUS\_MASTER [31:0] =0
- в регистре CSR\_MASTER: DONE = 1, Fatal Error =0, Break Done =0, WINDOW=1; RUN=0;
- в регистре QSTR\_PCI:  
если обнаружена ошибка чётности и MASK\_DPE =1,  
то ERROR= 1, DONE =1,  
если ошибка чётности не обнаружена или замаскирована (MASK\_DPE =0),  
то ERROR=0, DONE =0.

Если транзакция завершается по условию “Retry” (STATUS\_MASTER [23] =1) и установлен признак Master Break(CSR\_PCI[24] =1), то

- в регистре CSR\_MASTER: DONE = 1, Fatal Error =0, Break Done =1, WINDOW=1; RUN=0;
- в регистре QSTR\_PCI: ERROR = 1, DONE =1;
- в регистре STATUS\_MASTER: Break Done =1, Retry =1.

Если транзакция завершается по условию “Retry” (STATUS\_MASTER [23] =1) и признак Master Break сброшен (CSR\_PCI[24] =0), то PMSC повторяет транзакцию до тех пор, пока она не завершится передачей данных.

## 14.7 Передача прерываний

### 14.7.1 Передача вектора прерывания из шины PCI

Из шины PCI передача вектора прерывания процессору выполняется с помощью регистров почтового ящика MBR\_PCI и семафора SEM.

32-разрядный регистр MBR\_PCI предназначен для хранения вектора прерывания. Разряд 0 регистра SEM является признаком занятости MBR\_PCI по записи со стороны шины PCI: при SEM = 0 регистр он свободен, а при SEM = 1 – занят. Разряды 31:1 регистра SEM не используются.

Перед записью в регистр MBR\_PCI со стороны шины PCI следует убедиться, что он свободен. Для этого необходимо опросить состояние семафора SEM командой Memory Read. После выполнения этой команды нулевой разряд регистра SEM аппаратно устанавливается в 1, поэтому при следующем чтении MBR\_PCI будет уже занят. Этот механизм позволяет избежать конфликта при совместном использовании регистра MBR\_PCI несколькими драйверами PCI.

При записи в регистр MBR\_PCI по команде Memory Write устанавливается прерывание INT\_MBR. INT\_MBR сбрасывается при считывании содержимого MBR\_PCI по шине CDB. Его состояние доступно для чтения по шине CDB в разряде 28 регистра QSTR\_PCI. INT\_MBR устанавливается и снимается по переднему фронту системной частоты CLK.

После обработки прерывания признак занятости MBR\_PCI может быть сброшен записью нуля в регистр SEM по шине CDB или по команде Memory Write .



Конфигурационная запись в регистры MBR\_PCI и SEM не выполняется.

### 14.7.2 Передача прерываний в шину PCI

Передача вектора прерывания в шину PCI выполняется с помощью 32-разрядного регистра почтового ящика MBR\_CPU. При записи в этот регистр по шине CDB устанавливается признак прерывания INT\_CPU. Он используется для формирования прерывания nINTA.

INT\_CPU сбрасывается при считывании содержимого MBR\_CPU по шине PCI. Его состояние доступно для чтения по шине PCI в разряде 28 регистра QSTR\_PCI.

INT\_CPU устанавливается и снимается по переднему фронту частоты PCLK.

### 14.7.3 Регистры управления прерываниями

PMSC сигнализирует о наличии внутренних прерываний на линии nINTA шины PCI и на выходах INT\_PMCh, INT\_MBR.

Для организации прерываний используются регистр системных прерываний QSTR\_PCI и регистр маскирования прерываний MASKR\_PCI, форматы которых приведены в Таблица 14.10. и Таблица 14.11 соответственно.

**Таблица 14.10. Назначение разрядов регистра QSTR\_PCI**

Разряд регистра	Условное обозначение	Причина прерывания
31	DONE	Окончание передачи блока данных в режиме Master. Обнуляется при чтении регистра QSTR_PCI
30	ERROR	Ошибка при передаче блока данных в режиме Master. Обнуляется при чтении регистра QSTR_PCI
29	WMARK	Предварительное уведомление о передаче по шине PCI в режиме Master заданного полем WaterMark регистра TMR_PCI количества слов. Обнуляется при чтении регистра QSTR_PCI
28	INT_CPU/INT_MBR	При чтении по шине PCI - считывается признак INT_CPU : устанавливается при записи в регистр MBR_CPU по шине CDB, обнуляется при чтении MBR_CPU по шине PCI. При чтении по шине CDB - считывается INT_MBR: устанавливается при записи в регистр MBR_PCI по шине PCI, обнуляется при чтении MBR_PCI по шине CDB;
27:0	-	Не используются

Регистр QSTR\_PCI доступен только по чтению. Исходное состояние регистра QSTR\_PCI - нули (нет запросов прерывания).

**Таблица 14.11. Назначение разрядов регистра MASKR\_PCI**

Разряд регистра	Условное обозначение	Причина прерывания
31	M_DONE	Маска прерывания DONE шины PCI
30	M_ERROR	Маска прерывания ERROR шины PCI
29	M_WMARK	Маска прерывания WMARK шины PCI
28	M_INT_CPU	Маска прерывания INT_CPU
27:0	-	Не используются

Регистр MASKR\_PCI доступен по записи и чтению. Исходное состояние регистра - нули (все прерывания замаскированы).

Прерывания DONE, ERROR, WMARK, INT\_CPU маскируются соответствующими масками M\_\*. Результат объединения по схеме ИЛИ незамаскированных прерываний фиксируется в разряде Interrupt Status регистра Status/Command.

При Interrupt Desable = 0 (бит 10 регистра Status/Command) и Interrupt Status =1 на внешнем выводе nINTA устанавливается низкий уровень сигнала. В противном случае - пассивный уровень. Состояние вывода nINTA изменяется асинхронно относительно PCLK.

Результат объединения по схеме ИЛИ прерываний DONE, ERROR и WMARK поступает на выход системного прерывания INT\_PMCh.

Состояние вывода INT\_PMCh изменяется по переднему фронту системной частоты CLK.

## 14.8 Передача стартового адреса

В режиме PBOOT (высокий уровень одноименного входа микропроцессора) после снятия сигнала nRST процессор находится в состоянии останова. Запуск CPU и выход из режима PBOOT осуществляется записью стартового адреса в регистр QSTR\_PCI командой Memory Write. После запуска CPU начинает выполнять стартовую программу с адреса из регистра QSTR\_PCI. Предварительно, из шины PCI должны быть загружены необходимые программы и данные во внутреннюю память и/или в блок памяти, подключенный к выводу nCS[3] (или nCS[4]). После выхода из режима PBOOT регистр QSTR\_PCI аппаратно обнуляется.

## 14.9 Арбитр

Контроллер PMSC содержит арбитр шины PCI, имеющий 5 входов nREQ[4:0] запроса доступа к шине PCI и 5 выходов разрешения доступа nGNT[4:0].

В арбитра реализована одноуровневая схема приоритета доступа к шине PCI. Взаимный приоритет запросов nREQ[4:0] изменяется циклически в соответствии с Таблица 14.12. после каждого предоставления шины PCI очередному мастеру. Исходное распределение приоритетов между запросами (в порядке их убывания): nREQ[0], nREQ[1], nREQ[2], nREQ[3], nREQ[4].

Таблица 14.12. Приоритеты арбитра шины PCI

Обслуживаемый запрос	Распределение приоритетов очередного обмена
nREQ[0]	nREQ[1], nREQ[2], nREQ[3], nREQ[4], nREQ[0]
nREQ[1]	nREQ[2], nREQ[3], nREQ[4], nREQ[0], nREQ[1]
nREQ[2]	nREQ[3], nREQ[4], nREQ[0], nREQ[1], nREQ[2]
nREQ[3]	nREQ[4], nREQ[0], nREQ[1], nREQ[2], nREQ[3]
nREQ[4]	nREQ[0], nREQ[1], nREQ[2], nREQ[3], nREQ[4]

## 15. КОНТРОЛЛЕР ШИНЫ I2C

### 15.1 Назначение

Контроллер шины I2C предназначен для обмена данными по последовательной шине I2C. В состав шины I2C входят двунаправленные линии SCL, SDA, по которым передается тактовая частота синхронизации и последовательные данные.

### 15.2 Основные характеристики

Контроллер I2C имеет следующие характеристики:

- соответствует Philips I<sup>2</sup>C-bus specification version 2.1;
- поддерживает Multi-Master режим (синхронизация тактовых частот, процедура арбитража при передаче данных);
- программируемая частота обмена данными по последовательному интерфейсу;
- поддерживает 7-битный и 10-битный режим адресации.

### 15.3 Структурная схема

Структурная схема контроллера I2C приведена на Рисунок 15.1.

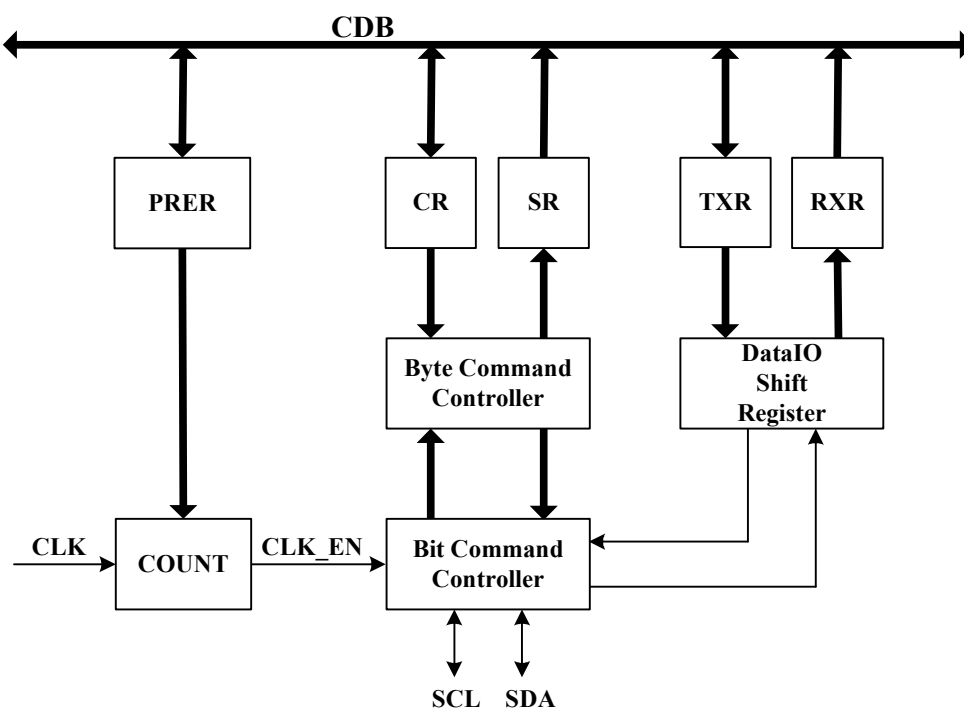


Рисунок 15.1. Структурная схема контроллера I2C.

В состав контроллера I2C входят следующие основные узлы:

- COUNT – счетчик предделителя;
- Bit Command Controller – узел, контролирующий выполнение приема/передачи бита данных;
- Byte Command Controller – узел, контролирующий выполнение приема/передачи байта данных;
- DataIO Shift Register – сдвиговый регистр передаваемых/принимаемых с линии данных.

На структурной схеме контроллера I<sup>2</sup>C использованы следующие обозначения:

- CDB – шина данных CPU;
- CLK – тактовая частота работы CPU;
- SCL, SDA – линии интерфейса I<sup>2</sup>C.

## 15.4 Регистры контроллера I2C

В Таблица 15.1 приведен перечень программно-доступных регистров контроллера I2C.

Таблица 15.1 Перечень программно-доступных регистров контроллера I2C.

Условное обозначение регистра	Название регистра	Тип доступа	Исходное Состояние	Адрес регистра HADDR<4:2>
PRER[15:0]	Регистр предделителя частоты	W/R	FFFF	000
CTR[8:0]	Регистр управления	W/R	0	001
TXR[7:0]	Регистр передачи данных	W/R	0	010
RXR[7:0]	Регистр приема данных	R	0	011
CR[7:0]	Регистр команд	W/R	0	100
SR[7:0]	Регистр состояния	R	0	101
PR_CNT[15:0]	Счетчик предделителя частоты	W/R	0000	110
SYNC[16:0]	Регистр синхронизации	W/R	10000	111

### 15.4.1 Регистр PRER

Регистр предделителя частоты PRER используется для задания частоты обмена данными по I2C интерфейсу. Порт I2C использует частоту, равную ( $5 \times F\_SCL$ ). Таким образом, значение коэффициента предделения определяется в соответствии с выражением:

$$PRER = \frac{F\_S}{5 \times F\_SCL} - 1,$$

где  $F\_S$  – системная частота,

$F\_SCL$  – требуемая частота обмена данными по I2C интерфейсу.

### 15.4.2 Регистр CTR

Формат регистра управления CTR приведен в Таблица 15.2.

Таблица 15.2 Формат регистра CTR

Номер бита	Условное обозначение	Назначение
5:0	–	Резерв
6	IEN	Разрешение прерывания от порта I2C
7	EN	Разрешение работы порта I2C: 0 – запрещение работы; 1 – разрешение работы.
8	PRST	Программный сброс
9	TM_CNT	Разрешение режима тестирования счетчика предделителя частоты. Доступен по записи только когда бит EN = 0.
10	TICK	Бит тестирования регистра счетчика PR_CNT. Доступен по записи только когда бит TM_CNT = 1.

### 15.4.3 Регистр TXR

Формат регистра TXR приведен в Таблица 15.3.

Таблица 15.3 Формат регистра TXR

Номер бита	Условное обозначение	Назначение
0	RW	При передаче байта данных этот бит задает младший разряд передаваемых данных; при передаче адреса ведомого устройства этот бит задает направление передачи данных: 1 – чтение из ведомого устройства; 0 – запись в ведомое устройство.
7:1	TXD	Передаваемые данные

### 15.4.4 Регистр RXR

Регистр RXR[7:0] содержит последний принятый байт данных.

### 15.4.5 Регистр CR

Формат регистра команд CR приведен в Таблица 15.4.

Регистр команд CR доступен по записи только при разрешении работы порта I2C, то есть когда установлен бит EN регистра управления CTR.

Биты SND, RCV, STO, STA регистра CR сбрасываются автоматически после выполнения заданной команды, либо когда порт I2C проигрывает арбитраж.

Таблица 15.4 Формат регистра команд CR

Номер бита	Условное обозначение	Назначение
0	IACK	Подтверждение прерывания. При установке этого бита сбрасывается бит IF регистра состояния SR Этот бит после установки сбрасывается автоматически
2:1	–	Резерв
3	ACK	При приеме байта данных от ведомого устройства задает, что выставит I <sup>2</sup> C порт: 0 – бит подтверждения ACK; 1 – бит неподтверждения NACK.
4	SND	Пересылка байта данных в ведомое устройство прием бита ACK/NACK от ведомого устройства
5	RCV	Прием байта данных от ведомого устройства пересылка бита ACK/NACK в ведомое устройство
6	STO	Генерация состояния STOP на линии
7	STA	Генерация состояния START (repeated START) на линии

В соответствии со спецификацией интерфейса I2C порт может инициировать передачу данных только когда шина свободна, то есть бит Busy регистра SR не установлен.

Порт I2C генерирует состояние START на линии, когда в командном регистре CR установлен бит STA и бит SND или RCV.

### 15.4.6 Регистр SR

Формат регистра состояния SR приведен в Таблица 15.5.

Таблица 15.5 Формат регистра состояния SR

Номер бита	Условное обозначение	Назначение
0	IF	Признак наличия прерывания: 0 – есть прерывание; 1 – нет прерывания. Признак устанавливается: – после завершения пересылки/приема байта данных; – когда порт I2C проигрывает арбитраж.
1	TIP	Признак выполнения передачи данных портом I2C: 1 – порт выполняет передачу данных; 0 – порт завершил передачу данных. Устанавливается в 1 с задержкой в 1 такт после установки бита RCV (SND) регистра команд CR. После выполнения команды RCV(SND) бит TIP устанавливается в 0
4:2	–	Резерв
5	AL	Признак того, что порт I2C проиграл арбитраж. Этот бит устанавливается когда: – порт I <sup>2</sup> C пытается установить высокий уровень на линии данных SDA, но на линии устанавливается низкий уровень; – порт I <sup>2</sup> C обнаруживает на линии состояние STOP, но сам порт не выполняет в данный момент команду STO. Этот бит сбрасывается при обнаружении состояния START на линии.
6	Busy	Признак того, что I2C интерфейс занят, то есть выполняется передача данных. Устанавливается при обнаружении состояния START на линии, сбрасывается при обнаружении состояния STOP на линии
7	RxACK	Принятый бит ACK/NACK от ведомого устройства после пересылки байта данных: 1 – бит неподтверждения NACK . 0 – бит подтверждения ACK;

### 15.4.7 Регистр PR\_CNT

Регистр счетчика предделителя частоты PR\_CNT используется для формирования частоты обмена данными по I2C интерфейсу в соответствии со значением регистра предделителя частоты PRER. Доступен по записи только когда разрешен режим тестирования счетчика предделителя частоты – бит TM\_CNT = 1.

### 15.4.8 Регистр SYNC

Регистр синхронизации используется для управления процессом синхронизации двух и более мастеров подключенных к шине I2C. Возможность управления процессом синхронизации повышает помехозащищенность канала SCL от внешних воздействий.

Таблица 15.6 Формат регистра SYNC

Номер бита	Условное обозначение	Назначение
15:0	WAIT_PER	Количество тактов системной частоты при высоком уровне сигнала SLC, после которых производить синхронизацию. $\text{WAIT\_PER} \leq (\text{tHSCl} / \text{tHCLK}) - 1,$ где tHCLK – период системной частоты (ns); tHSCl = 600 ns – минимальная длительность высокого уровня сигнала SCL.
16	SYNC_EN	Включение возможности синхронизации 1 – синхронизация включена; 0 – синхронизация выключена.

## 15.5 Функционирование контроллера I2C

Шина I2C подразумевает побитный обмен данными. Порт I2C выполняет следующие побитные операции:

- Генерация состояния START на линии;
- Генерация состояния repeated START на линии;
- Генерация состояния STOP на линии;
- Пересылка бита данных - send;
- Прием бита данных - receive.

Каждая побитовая операция разбивается на 5 фаз: A, B, C, D, IDLE, за исключением побитовых операций генерации состояния START/repeated START, которые выполняются за большее число фаз. Временная диаграмма выполнения побитовых операций представлена на Рисунке 15.2

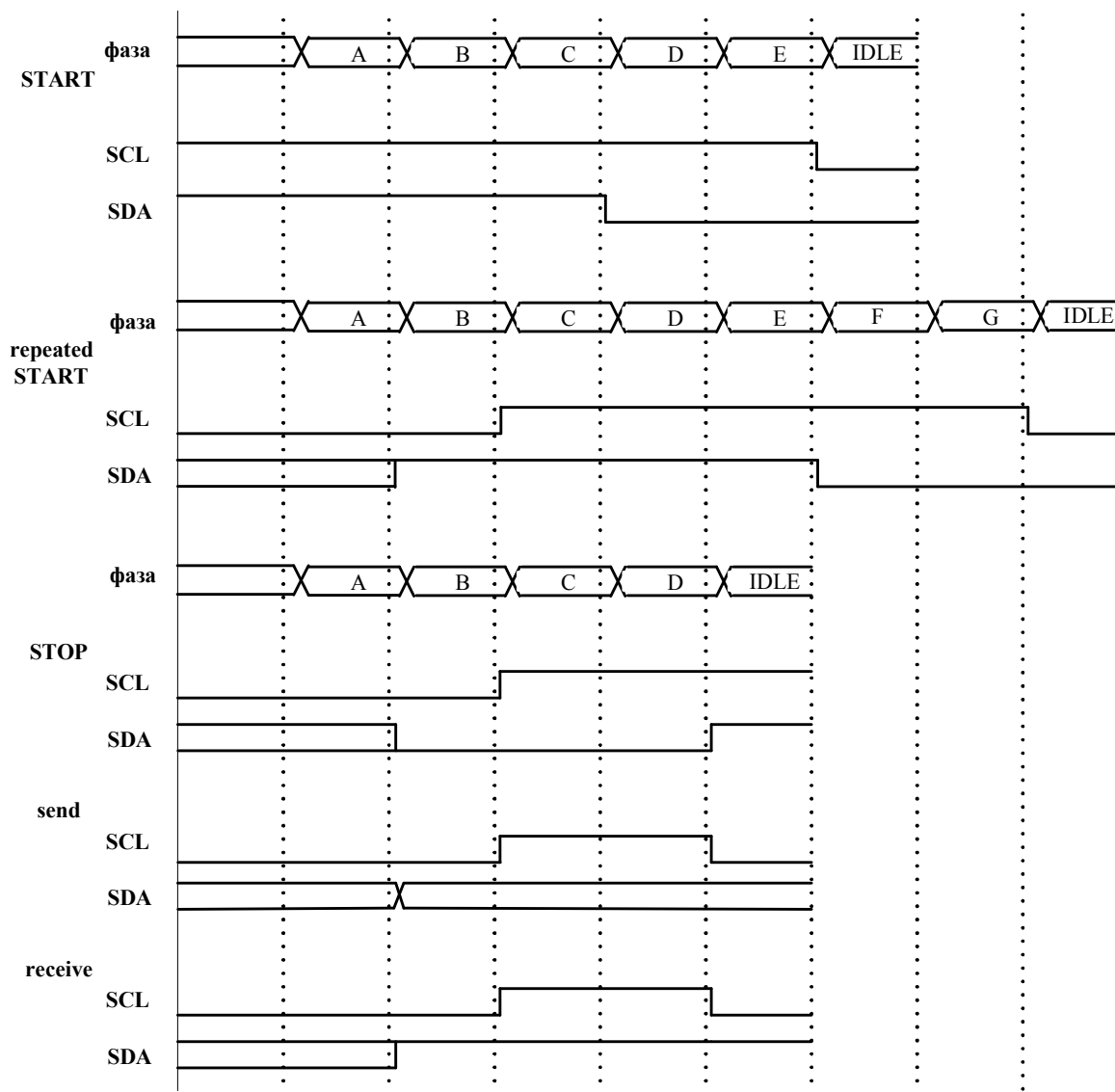


Рисунок 15.2 Временная диаграмма выполнения побитовых операций.

## 15.6 Программирование контроллера I2C

Порядок программирования при записи байта данных в ведомое устройство:

- Записать {адрес ведомого устройства , RW=0} в регистр TXR;
- Установить биты STA и SND в регистре команд CR;
- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать бит RxACK регистра состояния SR;
- Если RxACK = 1, то ведомое устройство не готово к обмену, поэтому необходимо завершить передачу данных. Для этого нужно установить бит STO в регистре команд CR;
- Если RxACK = 0, то ведомое устройство готово к обмену;
- Записать байт данных, который требуется переслать в регистр TXR;
- Установить биты SND и STO в регистре команд CR;



- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать бит RxACK регистра состояния SR;
- Если RxACK = 0, то ведомое устройство успешно приняло байт данных.

Порядок программирования при чтении двух байт данных из ведомого устройства:

- Записать {адрес ведомого устройства , RW=1} в регистр TXR;
- Установить биты STA и SND в регистре команд CR;
- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать бит RxACK регистра состояния SR;
- Если RxACK = 1, то ведомое устройство не готово к обмену, поэтому необходимо завершить передачу данных. Для этого нужно установить бит STO в регистре команд CR;
- Если RxACK = 0, то ведомое устройство готово к обмену.
- Установить бит RCV, а бит ACK сбросить в регистре команд CR для приема первого байта данных и пересылки после приема бита подтверждения;
- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать полученный байт данных из регистра RXR;
- Установить бит RCV и бит ACK в регистре команд CR для приема второго байта данных и пересылки после приема бита неподтверждения;
- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать полученный байт данных из регистра RXR;
- Установить бит STO в регистре команд CR для завершения передачи данных.

Порядок программирования при записи байта данных, а затем чтении байта данных из ведомого устройства:

- Записать {адрес ведомого устройства , RW=0} в регистр TXR;
- Установить биты STA и SND в регистре команд CR;
- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать бит RxACK регистра состояния SR;
- Если RxACK = 1, то ведомое устройство не готово к обмену, поэтому необходимо завершить передачу данных. Для этого нужно установить бит STO в регистре команд CR;
- Если RxACK = 0, то ведомое устройство готово к обмену.
- Записать байт данных, который требуется переслать в регистр TXR;
- Установить бит SND в регистре команд CR;
- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать бит RxACK регистра состояния SR;
- Если RxACK = 0, то ведомое устройство успешно приняло байт данных.
- Записать {адрес ведомого устройства , RW=1} в регистр TXR;
- Установить биты STA(repeated START) и SND в регистре команд CR;

- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать бит RxACK регистра состояния SR;
- Если RxACK = 1, то ведомое устройство не готово к обмену, поэтому необходимо завершить передачу данных. Для этого нужно установить бит STO в регистре команд CR;
- Если RxACK = 0, то ведомое устройство готово к обмену.
- Установить бит RCV и бит ACK в регистре команд CR для приема байта данных и пересылки после приема бита неподтверждения;
- Ждать, когда установится бит IF или когда сбросится бит TIP в регистре SR;
- Считать полученный байт данных из регистра RXR;
- Установить бит STO в регистре команд CR для завершения передачи данных.

## 16. ПОРТ ВВОДА ВИДЕОДАНЫХ VPIN

### 16.1 Назначение

Порт ввода видеоданных VPIN предназначен для ввода цифровых видеоданных по 8/10/12-разрядному параллельному интерфейсу. В частности, порт обеспечивает ввод информации с видеодатчиков на основе ПЗС или КМОП-матриц в формате байеровской цветовой модели (Bayer color pattern), стандарта BT.656 (ITU-R Recommendation BT.656), монохромного видео с последовательной разверткой (raw video). Порт обеспечивает простое аппаратное сопряжение с широким набором стандартных видеодатчиков, АЦП и кодеков.

### 16.2 Архитектура и функционирование порта VPIN

Структурная схема порта приведена на Рисунок 16.1.

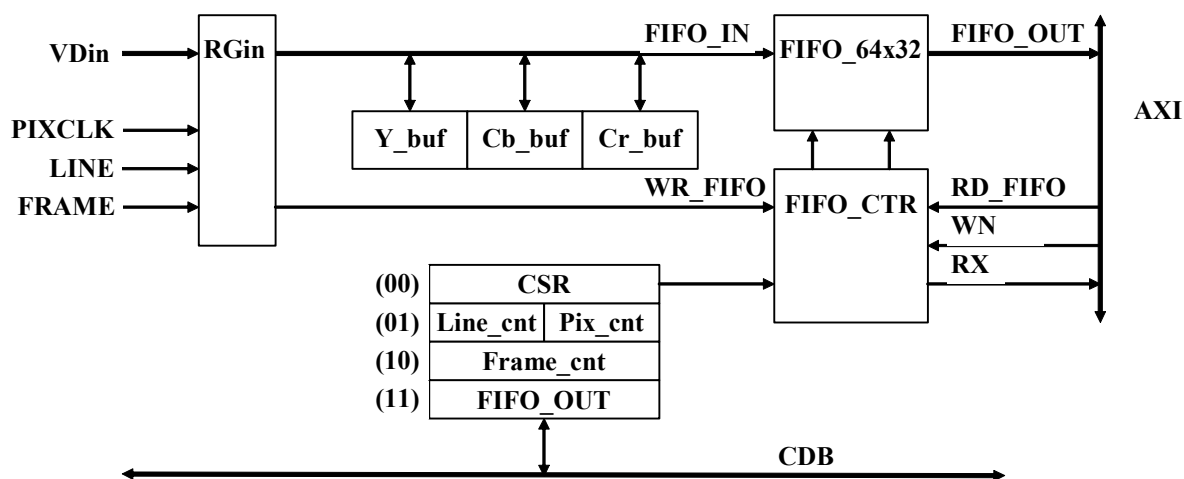


Рисунок 16.1 Структурная схема порта VPIN

В состав порта входят следующие основные блоки:

RGin – входной регистр видеоданных;

Y\_buf – буферный регистр для хранения компоненты Y входных видеоданных;

Cb\_buf – буферный регистр для хранения компоненты Cb входных видеоданных;

Cr\_buf – буферный регистр для хранения компоненты Cr входных видеоданных;

FIFO\_64x32 – буферная память типа FIFO объемом 32 слова по 64 разряда;

FIFO\_CTRL – устройство управления (контроллер) FIFO;

CSR, Line\_cnt/Pix\_cnt, Frame\_cnt, FIFO\_OUT – программно-доступные регистры порта.

Входные видеоданные VDin[9:0] поступают на вход порта в сопровождении трех сигналов синхронизации:

- синхронизации пикселей PIXCLK,
- строчной синхронизации LINE,
- кадровой синхронизации FRAME.

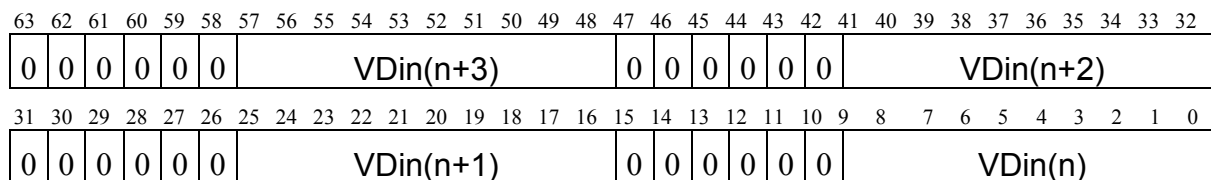
Временные диаграммы поступающих сигналов приведены на рис.19.2-19.5.

Предполагается, что входные видеоданные изменяются по положительному фронту сигнала PIXCLK. Сигналы строчной синхронизации LINE и кадровой синхронизации FRAME служат соответственно для подтверждения активной части строки (LINE=1) и активной части кадра (FRAME =1).

Входные видеоданные записываются во входной регистр RGiN по отрицательному фронту сигнала синхронизации PIXCLK при наличии одновременно активных уровней строчной и кадровой синхронизации (LINE= FRAME =1). Затем видеоданные переписываются в промежуточный регистр, работающий на системной частоте HCLK, и вся дальнейшая работа порта происходит на системной частоте.

Перед тем, как быть записанными в FIFO, данные переупорядочиваются при помощи специальных буферов цветовой компонент Y\_buf, Cb\_buf, Cr\_buf.

Переупорядочивание производится с целью объединения в одно 64-разрядное слово четырех выборок, относящихся к одной цветовой компоненте. Структура 32-разрядного слова, поступающего на вход FIFO VDin, приведена ниже.



С выхода FIFO данные могут быть прочитаны как со стороны соответствующего канала DMA по шине AMBA AXI, так и со стороны центрального процессора по шине AMBA AHB.

Чтение 64-разрядных данных из FIFO по шине AMBA AXI происходит под управлением сигналов:

RX - сигнал наличия данных в FIFO,

WN - число считываемых из FIFO слов (число слов вычисляется по формуле  $WN+1$ ),

RDY\_DMA - сигнал начала чтения данных из FIFO по шине AMBA AXI,

RD\_FIFO - сигнал завершения чтения данных из FIFO по шине AMBA AXI.

Чтение данных из FIFO по шине AMBA AHB производится 32-разрядными словами согласно протоколу данной шины, по отношению к которой порт VPIN является ведомым абонентом (Slave). Примечание: переключение указателя адреса FIFO происходит при чтении из 64-разрядной ячейки FIFO старшего 32-разрядного слова.

Кроме того, по шине AMBA AHB происходит запись-чтение программно доступных регистров порта VPIN.

Порт VPIN может выдавать центральному процессору прерывание в зависимости от состояния соответствующей маски по следующим событиям:

- FIFO заполнено,
- начало строки,
- конец строки,
- начало кадра,
- конец кадра,
- двойная ошибка при декодировании маркера по стандарту BT.656.

### 16.3 Программно-доступные регистры

По шине AMBA АНВ центральный процессор в зависимости от выбранного адреса HADDR[3:2] может произвести обращение к одному из программно-доступных регистров порта VPIN, перечень которых приводится в Таблица 16.1.

Таблица 16.1 Программно-доступные регистры порта VPIN

Условное обозначение	Число разрядов	Тип	Назначение
CSR	32	RW	Регистр управления и состояния
Line_cnt/Pix_cnt	32	R	Счетчик строк / счетчик пикселей
Frame_cnt	32	R	Счетчик кадров
FIFO_OUT	32	R	Выход FIFO

#### 16.3.1 Регистр управления и состояния (CSR)

Регистр управления и состояния CSR является 32-разрядным регистром, доступным по записи и чтению. Назначение разрядов регистра CSR приведено в Таблица 16.2.

Таблица 16.2 Назначение разрядов регистра CSR

Номер разряда	Условное обозначение	Назначение
31	CLR	Очистка порта. Установка этого бита в «1» приводит к остановке работы порта и сбросу всех указателей и счетчиков в «0».
30	RUN	RUN=0 – порт в состоянии останова; RUN=1 - порт в состоянии работы (при условии, что CLR=0).
29	Snapshot	Съемка одного кадра. Установка этого бита в «1» (при условии, что RUN=1, CLR=0) приводит к открытию порта для ввода одного ближайшего полного видеокadra, после чего порт останавливается. Перед каждым запуском данного режима необходима очистка порта (CLR=1).
28:23	INT_MSK0	Маска прерывания (разрешение прерывания при 1, запрещение – при 0): 28 – разрешено прерывание по заполнению FIFO; 27 – разрешено прерывание по началу кадра; 26 – разрешено прерывание по концу кадра; 25 – разрешено прерывание по началу строки; 24 – разрешено прерывание по концу строки; 23 – разрешено прерывание по двойной ошибке декодирования маркер
22	MRK	MRK=1 - режим декодирования маркера.
21:20	ORPM	Способ интерпретации входных видеоданных (для нечетных строк): ORPM=00 – монохроматический видеосигнал (Y); ORPM=10 – бихроматический видеосигнал (Y/C); ORPM=11 – 3-компонентный видеосигнал (Y/Cb/Cr).
19	INT_MSK1	Маска прерывания по ошибке FIFO (разрешение прерывания при 1, запрещение – при 0)
18	-	Резерв
17:16	ERPM	Способ интерпретации входных видеоданных (для четных строк): 00 – монохроматический видеосигнал (Y); 10 – бихроматический видеосигнал (Y/C); 11 – 3-компонентный видеосигнал (Y/Cb/Cr)
15	FDMA	Очистка DMA (Flush DMA) – при установке этого бита порт постоянно выдает каналу DMA сигнал готовности к передаче данных
14:4	-	Резерв
3	fifo_empty	Флаг пустого FIFO
2	fifo_err	Флаг ошибки FIFO (запись в заполненное FIFO)
1	fifo_full	Флаг заполненности FIFO
0	INT	Флаг прерывания

Начальное состояние регистра CSR=0x0.

### 16.3.2 Регистр - счетчик строк/ счетчик пикселей (*Line\_cnt/Pix\_cnt*)

Регистр-счетчик строк/ счетчик пикселей *Line\_cnt/Pix\_cnt* является 32-разрядным регистром, доступным только по чтению. Назначение разрядов регистра *Line\_cnt/Pix\_cnt* приведено в Таблица 16.3.

Таблица 16.3 Назначение разрядов регистра *Line\_cnt/Pix\_cnt*

Номер разряда	Условное обозначение	Назначение
31:28	-	0x0.
27:16	<i>Line_cnt</i>	12-разрядный счетчик строк. Автоматически обнуляется в начале каждого кадра.
15:12	-	0x0.
11:0	<i>Pix_cnt</i>	12-разрядный счетчик пикселей. Автоматически обнуляется в начале каждой строки.

Начальное состояние регистра *Line\_cnt/Pix\_cnt* = 0x0.

### 16.3.3 Регистр - счетчик кадров (*Frame\_cnt*)

Регистр-счетчик строк/ счетчик пикселей *Frame\_cnt* является 32-разрядным регистром, доступным только по чтению. Назначение разрядов регистра *Frame\_cnt* приводится в Таблица 16.4.

Таблица 16.4 Назначение разрядов регистра *Frame\_cnt*

Номер разряда	Условное обозначение	Назначение
31	F	Поле (BT.656)
30	V	V=0 - активная часть кадра (BT.656)
29	H	H=0 - активная часть строки (BT.656)
28	DBLERR	Двойная ошибка при декодировании маркера (BT.656)
27	h2FRAME	Текущее состояние сигнала FRAME
26	h2LINE	Текущее состояние сигнала LINE
25:24	-	0x0
23:0	<i>Frame_cnt</i>	24-разрядный счетчик кадров. Обнуляется при очистке порта (CLR=1).

Начальное состояние регистра *Frame\_cnt* = 0x60000000.

## 16.4 Режимы работы порта VPIN

### 16.4.1 Способы интерпретации входных видеоданных

Поля ORPM, ERPM регистра CSR определяют способ интерпретации портом входного видеопотока. Возможны три варианта:

ORPM(ERPM) = 00 – монохроматический видеосигнал (Y);

ORPM(ERPM) = 10 – бихроматический видеосигнал (Y/C);

ORPM(ERPM) = 11 – 3-компонентный видеосигнал (Y/Cb/Cr).

Временные диаграммы поступающих сигналов для трех указанных вариантов приведены на Рисунок 16.2 - Рисунок 16.4. На Рисунок 16.5 приведены временные диаграммы сигналов LINE, FRAME.

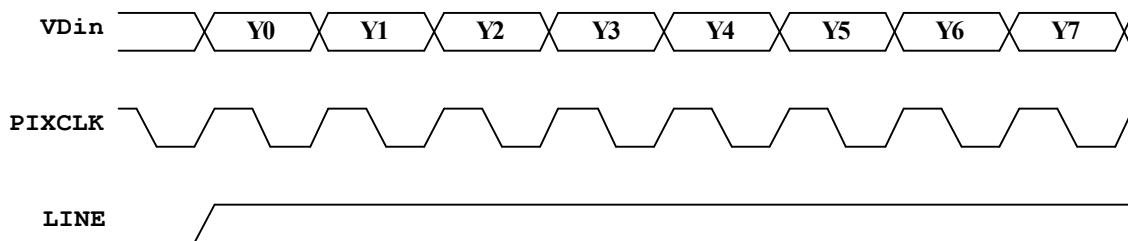


Рисунок 16.2 Временные диаграммы входных сигналов при 1-компонентных видеоданных

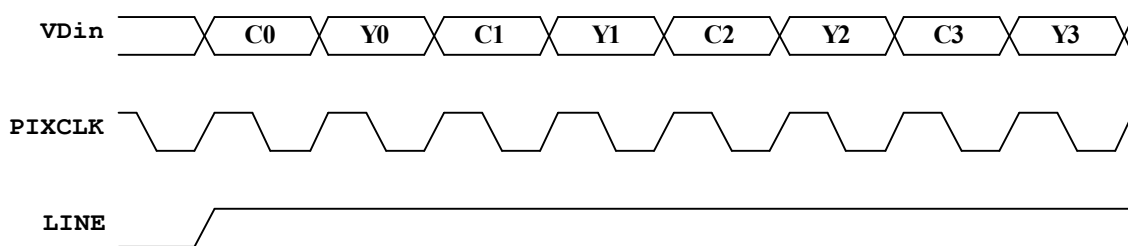


Рисунок 16.3 Временные диаграммы входных сигналов при 2-компонентных видеоданных

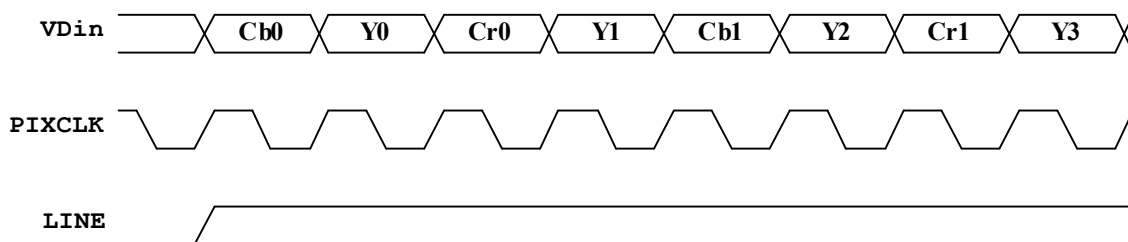


Рисунок 16.4 Временные диаграммы входных сигналов при 3-компонентных видеоданных

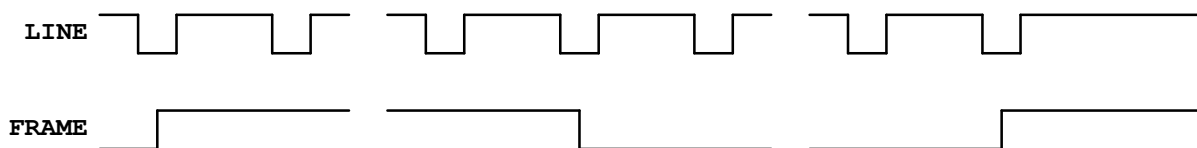


Рисунок 16.5 Временные диаграммы сигналов LINE, FRAME

### 16.4.2 Упаковка цветových компонент

Перед тем, как быть записанными в FIFO, данные переупорядочиваются при помощи специальных буферов цветových компонент Y\_buf, Cb\_buf, Cr\_buf.

Пиксели одного цвета упаковываются в одно 64-разрядное слово. Способы упаковки в зависимости от интерпретации входных видеоданных приведены на Рисунок 16.6 - Рисунок 16.8.

Номер слова	...	10	9	8	7	6	5	4	3	2	1	0
Видеоданные	...	Y43	Y39	Y35	Y31	Y27	Y23	Y19	Y15	Y11	Y7	Y3
		Y42	Y38	Y34	Y30	Y26	Y22	Y18	Y14	Y10	Y6	Y2
		Y41	Y37	Y33	Y29	Y25	Y21	Y17	Y13	Y9	Y5	Y1
		Y40	Y36	Y32	Y28	Y24	Y20	Y16	Y12	Y8	Y4	Y0

Рисунок 16.6 Структура видеопотока, поступающего на выход FIFO при ORPM(ERPM) = 00

Номер слова	...	10	9	8	7	6	5	4	3	2	1	0
Видеоданные	...	Y23	C19	Y19	C15	Y15	C11	Y11	C7	Y7	C3	Y3
		Y22	C18	Y18	C14	Y14	C10	Y10	C6	Y6	C2	Y2
		Y21	C17	Y17	C13	Y13	C9	Y9	C5	Y5	C1	Y1
		Y20	C16	Y16	C12	Y12	C8	Y8	C4	Y4	C0	Y0

Рисунок 16.7 Структура видеопотока, поступающего на выход FIFO при ORPM(ERPM) = 10

Номер слова	...	10	9	8	7	6	5	4	3	2	1	0
Видеоданные	...	Y23	C19	Y19	C15	Y15	C11	Y11	C7	Y7	C3	Y3
		Y22	C18	Y18	C14	Y14	C10	Y10	C6	Y6	C2	Y2
		Y21	C17	Y17	C13	Y13	C9	Y9	C5	Y5	C1	Y1
		Y20	C16	Y16	C12	Y12	C8	Y8	C4	Y4	C0	Y0

Рисунок 16.8 Структура видеопотока, поступающего на выход FIFO при ORPM(ERPM) = 11, PB=0



### 16.4.3 Режим съемки одного кадра (Snapshot)

Съемка одного кадра производится в следующем порядке:

- 1) Выполняется очистка порта (CLR=1);
- 2) В регистр CSR записываются биты RUN=Snapshot=1 (при CLR=0).

После выполнения съемки кадра порт останавливается (хотя биты RUN и Snapshot остаются в состоянии “1”). В счетчике кадров устанавливается значение Frame\_cnt=0x1.

### 16.4.4 Режим декодирования маркеров ВТ.656

Порт VPIN способен воспринимать и декодировать цифровой видеопоток в соответствии со стандартом ВТ.656 (ITU-R Recommendation ВТ.656). Согласно данному стандарту, строчная и кадровая синхронизация видеоданных производится при помощи встроенных в видеопоток специальных маркеров – SAV (Start Active Video) и EAV (End Active Video), обозначающих соответственно начало и конец строки.

Сигналы LINE и FRAME в данном режиме используются только для разрешения записи, но не в качестве строчной и кадровой синхронизации.

Режим декодирования маркера устанавливается битом MRK=1 регистра CSR.

Маркеры помещаются в старшем байте (разряды VDin[9:2]) входных видеоданных.

Для того, чтобы обеспечить возможность отличить маркеры от пикселей, в стандарте ВТ.656 вводится следующее ограничение: значения видеопикселей могут находиться в диапазоне от 1 до 254 (от 0x01 до 0xFE в шестнадцатеричной системе). Значения 0x00 и 0xFF используются только для кодирования маркеров.

Маркер состоит из четырех байт. Первые три байта представляют собой фиксированный префикс 0xFF 0x00 0x00, четвертый байт содержит информацию о текущем состоянии сигналов кадровой и строчной синхронизации.

Структура и назначение бит в четвертом байте маркера приведены в таблице ниже

DBin	1-й байт (0xFF)	2-й байт (0x00)	3-й байт (0x00)	4-й байт
DBin[9]	1	0	0	1
DBin[8]	1	0	0	F (поле) <sup>*)</sup>
DBin[7]	1	0	0	V (вертикальный бланк) <sup>**)</sup>
DBin[6]	1	0	0	H (горизонтальный бланк) <sup>***)</sup>
DBin[5]	1	0	0	P3 (бит защиты 3) <sup>****)</sup>
DBin[4]	1	0	0	P2 (бит защиты 2) <sup>****)</sup>
DBin[3]	1	0	0	P1 (бит защиты 1) <sup>****)</sup>
DBin[2]	1	0	0	P0 (бит защиты 0) <sup>****)</sup>

<sup>\*)</sup> F=0 для 1-го поля, F=1 для 2-го поля;

<sup>\*\*)</sup> V=0 для активной V=1 для неактивной части поля;

<sup>\*\*\*)</sup> H=0 для SAV, H=1 для EAV;

<sup>\*\*\*\*)</sup> Биты защиты P0-P3 определяются состоянием бит F, V, H;

Состояние бит защиты P0-P3 в зависимости от F, V, H приводится в таблице ниже.

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Наличие кода защиты P0-P3 позволяет исправлять одиночные ошибки при передаче F, V, H и обнаруживать двойные ошибки. Коррекция значений F, V, H производится согласно приводимой ниже таблице.

Полученные P3-P0	Полученные значения F, V, H							
	000	001	010	011	100	101	110	111
0000	000	000	000	-	000	-	-	111
0001	000	-	-	111	-	111	111	111
0010	000	-	-	011	-	101	-	-
0011	-	-	010	-	100	-	-	111
0100	000	-	-	011	-	-	110	-
0101	-	001	-	-	100	-	-	111
0110	-	011	011	011	100	-	-	011
0111	100	-	-	011	100	100	100	-
1000	000	-	-	-	-	101	110	-
1001	-	001	010	-	-	-	-	111
1010	-	101	010	-	101	101	-	101
1011	010	-	010	010	-	101	010	-
1100	-	001	110	-	110	-	110	110
1101	001	001	-	001	-	001	110	-
1110	-	-	-	011	-	101	110	-
1111	-	001	010	-	100	-	-	-

Прочерком в таблице обозначены случаи обнаружения двойных ошибок. В этих случаях в регистре Frame\_cnt устанавливается флаг двойной ошибки DBLERR и, при соответствующем состоянии маски, возникает прерывание.

## 17. ПОРТ ВЫВОДА ВИДЕОДАНЫХ VPOUT

### 17.1 Назначение

Порт вывода видеоданных VPOUT предназначен для вывода цифровых видеоданных по 16-разрядному параллельному интерфейсу. В частности, порт обеспечивает вывод видеoinформации в формате стандартов BT.656 (ITU-R Recommendation BT.656), SMPTE<sub>x</sub>, 16-разрядного RGB (5R/6G/5B) и монохромного видео с последовательной разверткой (raw video). Порт обеспечивает простое аппаратное сопряжение с широким набором стандартных видеокодек, видео-ЦАП и LCD-контроллеров.

### 17.2 Архитектура и функционирование порта VPOUT

Структурная схема порта приведена на Рисунок 17.1.

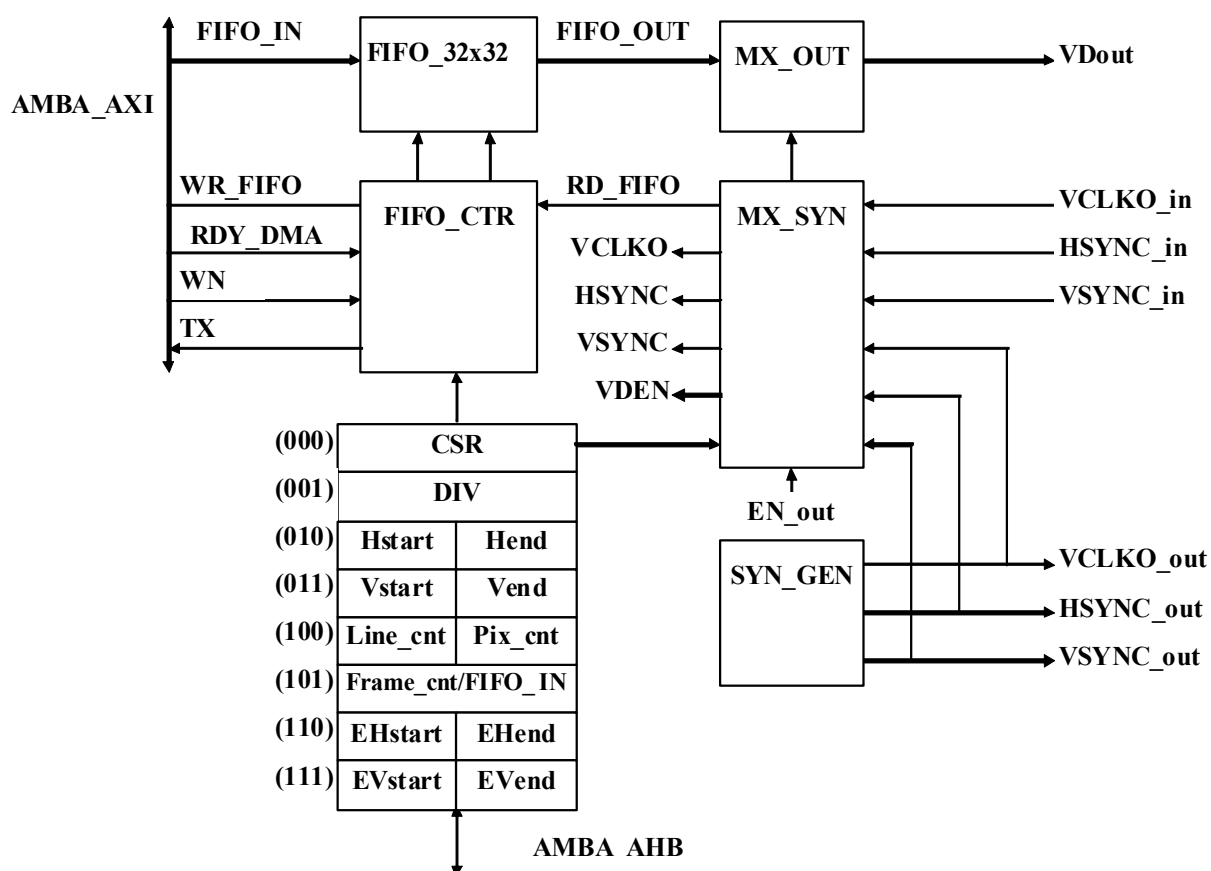


Рисунок 17.1 Структурная схема порта VPOUT.

В состав порта входят следующие основные блоки:

FIFO\_64x32 – буферная память типа FIFO объемом 32 слова по 64 разряда;

FIFO\_CTRL – устройство управления (контроллер) FIFO;

MX\_OUT – мультиплексор выходных данных;

MX\_SYN – мультиплексор сигналов синхронизации;

SYN\_GEN – генератор сигналов синхронизации;

CSR, DIV, Hstart/Hend, Vstart/Vend, Line\_cnt/Pix\_cnt, Frame\_cnt, FIFO\_IN – программно-доступные регистры порта.

Выходные видеоданные VDout[15:0] поступают на выход порта в сопровождении трех сигналов синхронизации:

- синхронизации пикселей VCLKO\_in (или VCLKO\_out),
- строчной синхронизации HSYNC\_in (или HSYNC\_out),
- кадровой синхронизации VSYNC\_in (или VSYNC\_out).

Все три сигнала синхронизации могут быть либо внешними (VCLKO\_in, HSYNC\_in, VSYNC\_in) либо внутренними, т.е. генерироваться самим портом (VCLKO\_out, HSYNC\_out, VSYNC\_out). Выбор осуществляется при помощи сигналов EN\_VCLKO, EN\_HSYNC, EN\_VSYNC:

Если EN\_VCLKO = 1 VCLKO = VCLKO\_out, иначе VCLKO = VCLKO\_in;

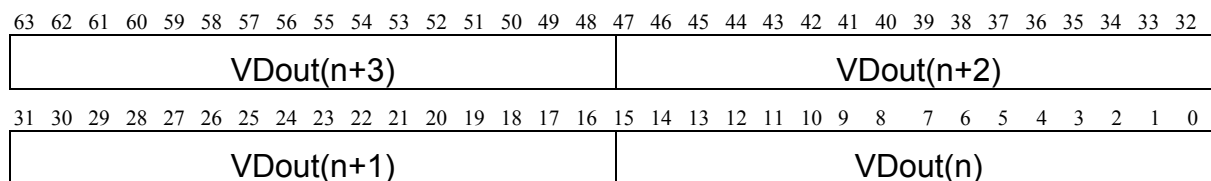
Если EN\_HSYNC = 1 HSYNC = HSYNC\_out, иначе HSYNC = HSYNC\_in;

Если EN\_VSYNC = 1 VSYNC = VSYNC\_out, иначе VSYNC = VSYNC\_in.

Временные диаграммы сигналов приведены на рис.20.2-20.3. Видеоданные изменяются по положительному фронту сигнала VCLKO. Сигналы строчной и кадровой синхронизации служат соответственно для подтверждения активной части строки и активной части кадра.

Выходные видеоданные записываются в выходной регистр **RGout** по положительному фронту сигнала пиксельной синхронизации при наличии одновременно активных (высоких) уровней строчной и кадровой синхронизации. В остальной части работа порта происходит на системной частоте HCLK.

Структура 64-разрядного слова, поступающего с выхода FIFO на выход порта VDout, приведена ниже.



Данные в FIFO могут быть записаны как со стороны соответствующего канала DMA по AXI, так и со стороны центрального процессора по шине СВИ.

Порт VPOUT может выдавать центральному процессору прерывание в зависимости от состояния соответствующей маски по следующим событиям:

- пустое FIFO,
- чтение из пустого FIFO (ошибка FIFO),
- начало строки,
- конец строки,
- начало кадра,
- конец кадра.

## 17.3 Программно-доступные регистры

Перечень программно-доступных регистров VPOUT приведен в Таблица 17.1.

Таблица 17.1

Условное обозначение	Число бит	Тип	Назначение
CSR	32	RW	Регистр управления и состояния
DIV	32	RW	Регистр периода сигнала VCLKO_out
Hstart/Hend	32	RW	Регистр начала/конца активной части строки
Vstart/Vend	32	RW	Регистр начала/конца активной части кадра
Line_cnt/Pix_cnt	32	R	Счетчик строк / счетчик пикселей
Frame_cnt (R) FIFO_IN (W)	32	R W	Счетчик кадров Адрес записи в FIFO по команде Store
EHstart/EHend	32	RW	Регистр начала/конца сигнала VDEN в строке
EVstart/EVend	32	RW	Регистр начала/конца сигнала VDEN в строке в кадре

### 17.3.1 Регистр управления и состояния (CSR)

Регистр управления и состояния CSR является 32-разрядным регистром, доступным по записи и чтению. Назначение разрядов регистра CSR приведено в Таблица 17.2.

Таблица 17.2

Номер разряда	Обозначение	Назначение
31	CLR	Очистка порта. Установка этого бита в «1» приводит к остановке работы порта и сбросу всех указателей и счетчиков в «0».
30	RUN	RUN=0 – порт в состоянии останова; RUN=1 - порт в состоянии работы (при условии, что CLR=0).
29	FEN	Разрешение переключения сигнала поля (F): FEN=0 – видеокадр состоит из одного поля, FEN=1 – видеокадр состоит из двух полей.
28:23	INT_MSK	Маска прерывания: CSR[28]=1 – разрешено прерывание по пустому FIFO; CSR[27]=1 – разрешено прерывание по началу кадра; CSR[26]=1 – разрешено прерывание по концу кадра; CSR[25]=1 – разрешено прерывание по началу строки; CSR[24]=1 – разрешено прерывание по концу строки; CSR[23]=1 – разрешено прерывание по чтению из пустого FIFO
22	EN_VCLKO	Бит выбора внутренней/внешней синхронизации VCLKO При EN_VCLKO=0 - выбор внешней синхронизации: VCLKO=VCLKO_in; При EN_VCLKO=1 - выбор внутренней синхронизации: VCLKO=VCLKO_out;
21	EN_HSYNC	Бит выбора внутренней/внешней синхронизации HSYNC При EN_HSYNC = 0 - выбор внешней синхронизации: HSYNC = HSYNC_in; При EN_HSYNC = 1 - выбор внутренней синхронизации: HSYNC = HSYNC_out;
20	EN_VSYNC	Бит выбора внутренней/внешней синхронизации VSYNC При EN_VSYNC = 0 - выбор внешней синхронизации: VSYNC = VSYNC_in; При EN_VSYNC = 1 - выбор внутренней синхронизации: VSYNC = VSYNC_out;

Номер разряда	Обозначение	Назначение
19	Snapshot	Выбор режима Snapshot. В этом режиме передается один кадр, после чего все сигналы синхронизации сбрасываются в «0».
18:17	MODE	Управление режимом формирования синхросигналов: MODE = 00: RGB, VDEN= GPIO[0], если FEN=0; MODE = 00: RGB, VDEN= F, если FEN=1; MODE = 01: RGB, VDEN=Enable Video; MODE = 10: i80, VSYNC = GPIO[1]; MODE = 11: GPIO[3:0].
16	PLR	PLR – бит, определяющий полярность VDEN: При PLR=0 VDEN имеет активный высокий уровень, при PLR=1 VDEN имеет активный низкий уровень.
15:12	GPIO	GPIO[3:0] – входы-выходы общего назначения GPIO[0] – только выход
11:3	-	Резерв
2	fifo_err	Флаг ошибки FIFO (чтение из пустого FIFO)
1	fifo_empty	Флаг пустого FIFO
0	INT	Флаг прерывания

Начальное состояние регистра CSR=0x0.

### 17.3.2 Регистр периода сигнала VCLKO\_out (DIV)

Регистр DIV является 32-разрядным регистром, доступным по записи и чтению. Назначение разрядов регистра приведено в Таблица 17.3.

Таблица 17.3

Номер разряда	Условное обозначение	Назначение
31:16	-	резерв
15:0	<b>DIV</b>	16-разрядный целочисленный коэффициент деления системной частоты HCLK для получения частоты синхронизации пикселей VCLKO_out.

Регистр DIV содержит в 16-ти младших разрядах целочисленный коэффициент деления системной частоты HCLK для получения частоты синхронизации пикселей VCLKO\_out. Период сигнала VCLKO\_out определяется формулой:

$$TVCLKO\_out = THCLK * (DIV + 1)$$

Начальное состояние регистра DIV =0x0.

### 17.3.3 Регистр начала/конца активной части строки (Hstart/Hend)

Регистр начала/конца активной части строки (Hstart/Hend) является 32-разрядным регистром, доступным по записи и чтению. Назначение разрядов регистра приведено в Таблица 17.4.

Таблица 17.4

Номер разряда	Условное обозначение	Назначение
31:16	<b>Hstart</b>	Число пикселей в неактивной части строки.
15:0	<b>Hend</b>	Число пикселей в строке.

Начальное состояние регистра Hstart=Hend =0x0.

### 17.3.4 Регистр начала/конца активной части кадра (*Vstart/Vend*)

Регистр начала/конца активной части строки (*Vstart/Vend*) является 32-разрядным регистром, доступным по записи и чтению. Назначение разрядов регистра приведено в Таблица 17.5.

Таблица 17.5

Номер разряда	Условное обозначение	Назначение
31:16	<b>Vstart</b>	Число строк в неактивной части кадра (поля).
15:0	<b>Vend</b>	Число строк в кадре (поле).

Начальное состояние регистра **Vstart=Vend=0x0**.

### 17.3.5 Регистр - счетчик строк/ счетчик пикселей (*Line\_cnt/Pix\_cnt*)

Регистр-счетчик строк/ счетчик пикселей **Line\_cnt/Pix\_cnt** является 32-разрядным регистром, доступным только по чтению. Назначение разрядов регистра **Line\_cnt/Pix\_cnt** приведено в Таблица 17.6.

Таблица 17.6

Номер разряда	Условное обозначение	Назначение
31:16	<b>Line_cnt</b>	12-разрядный счетчик строк. Автоматически обнуляется в начале каждого кадра.
15:0	<b>Pix_cnt</b>	12-разрядный счетчик пикселей. Автоматически обнуляется в начале каждой строки.

Начальное состояние регистра **Line\_cnt/Pix\_cnt=0x0**.

### 17.3.6 Регистр - счетчик кадров (*Frame\_cnt*)

Регистр-счетчик строк/ счетчик пикселей **Frame\_cnt** является 32-разрядным регистром, доступным только по чтению. Назначение разрядов регистра **Frame\_cnt** приведено в Таблица 17.7.

Таблица 17.7

Номер разряда	Условное обозначение	Назначение
31	<b>F</b>	Поле
30	<b>VSYNC</b>	Текущее состояние сигнала кадровой синхронизации
29	<b>HSYNC</b>	Текущее состояние сигнала строчной синхронизации
28:24	-	0x0
23:0	<b>Frame_cnt</b>	24-разрядный счетчик кадров. Обнуляется при очистке порта (CLR=1).

Начальное состояние регистра **Frame\_cnt=0x0**.

### 17.3.7 Регистр начала/конца сигнала *VDEN* в строке (*EHstart/EHend*)

Регистр начала/конца активной части строки (**EHstart/EHend**) является 32-разрядным регистром, доступным по записи и чтению. Назначение разрядов регистра приведено в Таблица 17.8.

Таблица 17.8

Номер разряда	Условное обозначение	Назначение
31:16	<b>Hstart</b>	Число пикселей в неактивной части строки.
15:0	<b>Hend</b>	Число пикселей в строке.

Начальное состояние регистра  $Hstart=Hend=0x0$ .

### 17.3.8 Регистр начала/конца активной части кадра (*Vstart/Vend*)

Регистр начала/конца активной части строки (*Vstart/Vend*) является 32-разрядным регистром, доступным по записи и чтению. Назначение разрядов регистра приведено в табл. 19.9.

Таблица 17.9

Номер разряда	Условное обозначение	Назначение
31:16	<b>Vstart</b>	Число строк в неактивной части кадра (поля).
15:0	<b>Vend</b>	Число строк в кадре (поле).

Начальное состояние регистра  $Vstart=Vend=0x0$ .

## 17.4 Режимы работы порта

Порт VPOUT может работать в одном из трех режимов:

- 1) Режим RGB – режим передачи видеоданных;
- 2) Режим i80 – режим передачи регистровых данных по протоколу i80;
- 3) Режим GPIO – режим работы в качестве порта ввода-вывода общего назначения.

Выбор режима работы порта задается 2-разрядным полем MODE регистра CSR.

При  $MODE=CSR[18:17]=00/01$  порт работает в режиме RGB; при  $MODE=10$  - в режиме i80, при  $MODE=11$  - в режиме GPIO.

В режимах i80 и GPIO порт работает от внутренних сигналов синхронизации, в режиме RGB может работать как от внутренних, так и от внешних сигналов.

Выбор внутренних либо внешних сигналов синхронизации VCLKO, HSYNC, VSYNC осуществляется при помощи разрядов EN\_VCLKO, EN\_HSYNC, EN\_VSYNC регистра CSR. При нулевых значениях указанных разрядов используется соответствующий внешний сигнал, при единичных значениях – внутренний. По сигналу сброса указанные разряды устанавливаются в нулевое состояние, то есть по умолчанию внешние выходы VCLKO, HSYNC, VSYNC сконфигурированы как входы.

### 17.4.1 Режим RGB

В рамках этого режима возможны три способа формирования сигнала VDEN в зависимости от значений управляющих бит MODE и FEN регистра CSR:

- 1)  $MODE = 00, FEN=0$ :  $VDEN=GPIO[0]$ .

В этом случае видеоданные сопровождаются сигналами синхронизации VCLKO, HSYNC, VSYNC в соответствии с временными диаграммами на Рисунок 17.2, Рисунок 17.3, а выход VDEN используется как программируемый выход общего назначения, состояние которого определяется битом GPIO[0] регистра CSR.

На рис.5 приведена структура видеопотока, поступающего на выход порта.

- 2)  $MODE = 00, FEN=1$ :  $VDEN= F$ .

Этот режим предназначен для передачи видеоданных с чересстрочной разверткой. В этом случае на выход VDEN подается сигнал поля F в соответствии с временной диаграммой на рисунке 18.6.



Таким образом, в зависимости от значения управляющего разряда FEN регистра CSR возможны два режима формирования сигнала кадровой синхронизации VSYNC:

FEN=0 – видеокадр состоит из одного поля,

FEN=1 – видеокадр состоит из двух полей.

В первом случае сигнал поля F (31-й разряд регистра **Frame\_cnt**) всегда остается равным нулю, во втором – переключается из “0” в “1” и обратно после окончания последней строки данного поля:

F = 0 – для первого поля,

F = 1 - для второго поля;

Второе поле отличается от первого тем, что его длительность (определяемая параметром **Vend**) и длительность его неактивной части (определяемая параметром **Vstart**) увеличены на единицу. Это объясняется структурой кадра, принятой в основных телевизионных стандартах – NTSC и PAL.

Временные диаграммы сигналов синхронизации пикселей VCLKO, строчной синхронизации HSYNC\_out, кадровой синхронизации VSYNC, F приведены на Рисунок 17.5 - Рисунок 17.6. На этих же диаграммах приведены формулы, связывающие между собой периоды сигналов синхронизации пикселей  $T_C$ , строчной синхронизации  $T_H$ , кадровой синхронизации  $T_V$  и параметры **Hstart**, **Hend**, **Vstart**, **Vend**.

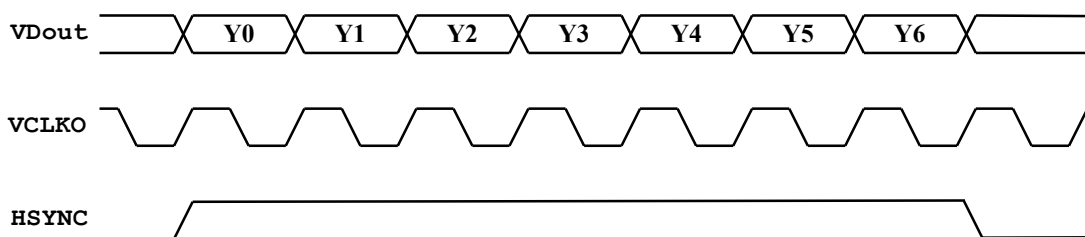


Рисунок 17.2 Временные диаграммы сигналов VDout, VCLKO, HSYNC

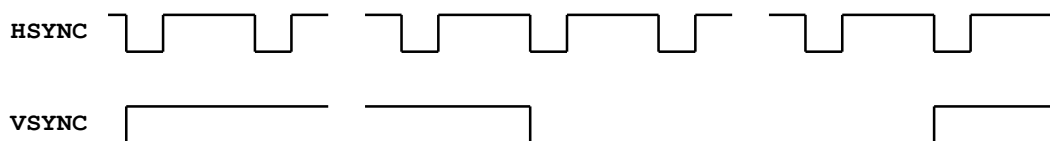


Рисунок 17.3 Временные диаграммы сигналов VSYNC, HSYNC

На Рисунок 17.4 приведена структура видеопотока, поступающего на вход и на выход FIFO.

Номер слова	...	10	9	8	7	6	5	4	3	2	1	0
Видеоданные	...	Y43	Y39	Y35	Y31	Y27	Y23	Y19	Y15	Y11	Y7	Y3
		Y42	Y38	Y34	Y30	Y26	Y22	Y18	Y14	Y10	Y6	Y2
		Y41	Y37	Y33	Y29	Y25	Y21	Y17	Y13	Y9	Y5	Y1
		Y40	Y36	Y32	Y28	Y24	Y20	Y16	Y12	Y8	Y4	Y0

Рисунок 17.4 Структура видеопотока, поступающего на выход FIFO

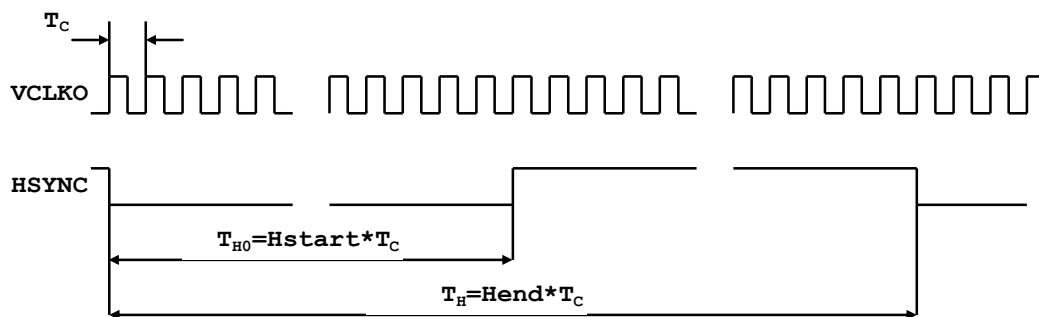


Рисунок 17.5 Временные диаграммы сигналов VCLK, HSYNC

3) MODE = 01: VDEN – сигнал подтверждения видеоданных.

В этом случае видеоданные сопровождаются сигналами синхронизации VCLKO, HSYNC, VSYNC, VDEN, причем выход VDEN используется как сигнал подтверждения видеоданных в соответствии с временными диаграммами на Рисунок 17.7.

Переключение сигнала VDEN по строке определяются параметрами **EHstart**, **EHend**, по столбцу - параметрами **EVstart**, **EVend**.

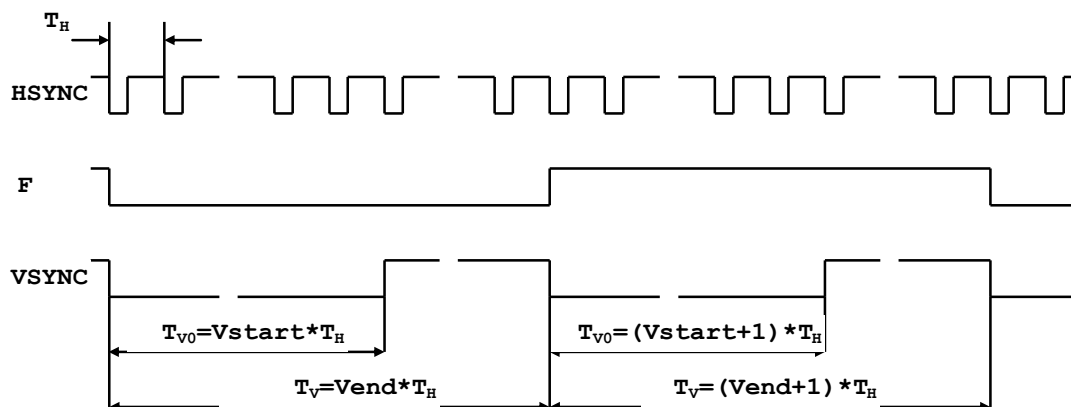


Рисунок 17.6 Временные диаграммы сигналов HSYNC, VSYNC

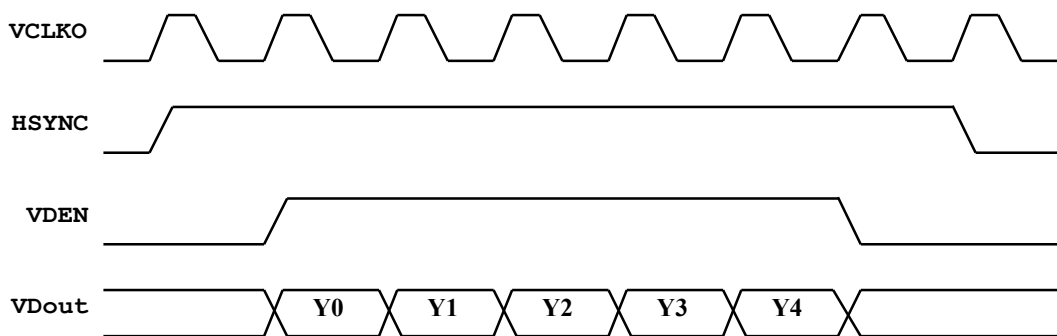


Рисунок 17.7 Временные диаграммы сигналов VDout, VCLKO, HSYNC, VDEN в режиме RGB

### 17.4.2 Режим i80

В этом режиме порт используется для однонаправленной передачи данных (только запись) по протоколу i80. При этом на выходы VCLKO, HSYNC, VSYNC, VDEN подаются следующие сигналы протокола i80 (Рисунок 17.8):

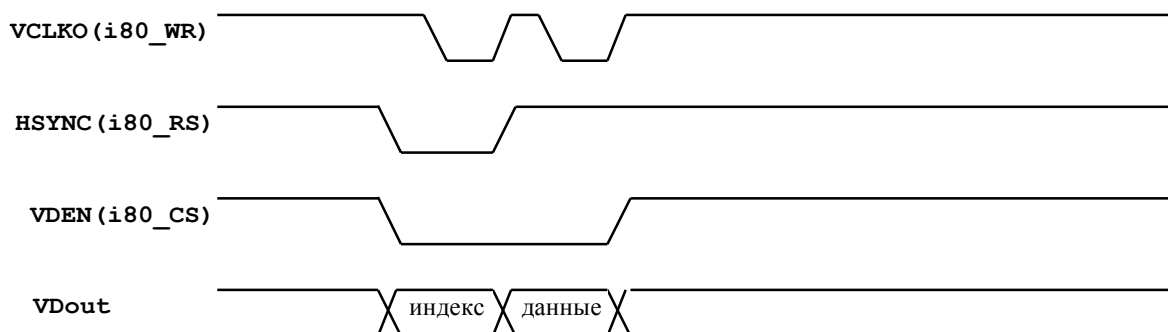


Рисунок 17.8 Временные диаграммы сигналов VDout, VCLKO, HSYNC, VDEN в режиме i80

VCLKO= i80\_WR – сигнал записи данных,

HSYNC= i80\_RS – сигнал выбора индекс/данные,

VDEN= i80\_CS – сигнал разрешения передачи.

Вывод VSYNC используется в этом режиме в качестве программируемого входа-выхода общего назначения: VSYNC= GPIO[1].

Переключение сигнала HSYNC (i80\_RS) определяются параметрами **EHstart**, **EHend**, сигнала VDEN (i80\_CS) - параметрами **EVstart**, **EVend**. Частота VCLKO (i80\_WR) задается, как и в режиме RGB, регистром **DIV**.

### 17.4.3 Режим GPIO

В этом режиме порт работает как 4-разрядный порт ввода-вывода общего назначения, причем в качестве программируемых входов-выходов общего назначения используются VCLKO, HSYNC, VSYNC, VDEN. Выводы VDout[15:0] при этом не используются.

Состояние выводов определяется битами GPIO[3:0] регистра CSR:

VCLKO= GPIO[3],

HSYNC= GPIO[2],

VSYNC= GPIO[1],

VDEN= GPIO[0],

причем вывод VDEN используется как выход, остальные – как входы-выходы.

## 18. МНОГОФУНКЦИОНАЛЬНЫЙ БУФЕРИЗИРОВАННЫЙ ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP)

### 18.1 Особенности MFBSP

Многофункциональный буферизированный последовательный порт (MFBSP) позволяет вести обмен параллельно-последовательным кодом с другими микросхемами по линковому интерфейсу (LPORT), либо обмениваться аудиоданными и управляющей информацией с внешними устройствами по последовательным интерфейсам в дуплексном режиме, с возможностью независимой настройки приёмника и передатчика. Гибкость последовательного порта позволяет организовывать передачу с широким спектром внешних устройств. Дополнительно порт позволяет организовывать обмен данными с внешними устройствами, используя входы-выходы общего назначения. На Рисунке 18.1 изображен MFBSP с двумя каналами DMA (на приём и передачу) в составе микропроцессора. По каналу DMA направления передачи осуществляется передача данных внешнему устройству, подключенному к микропроцессору через MFBSP. По каналу DMA направления приёма осуществляется приём данных из внешнего устройства, подключенного к микропроцессору через MFBSP.

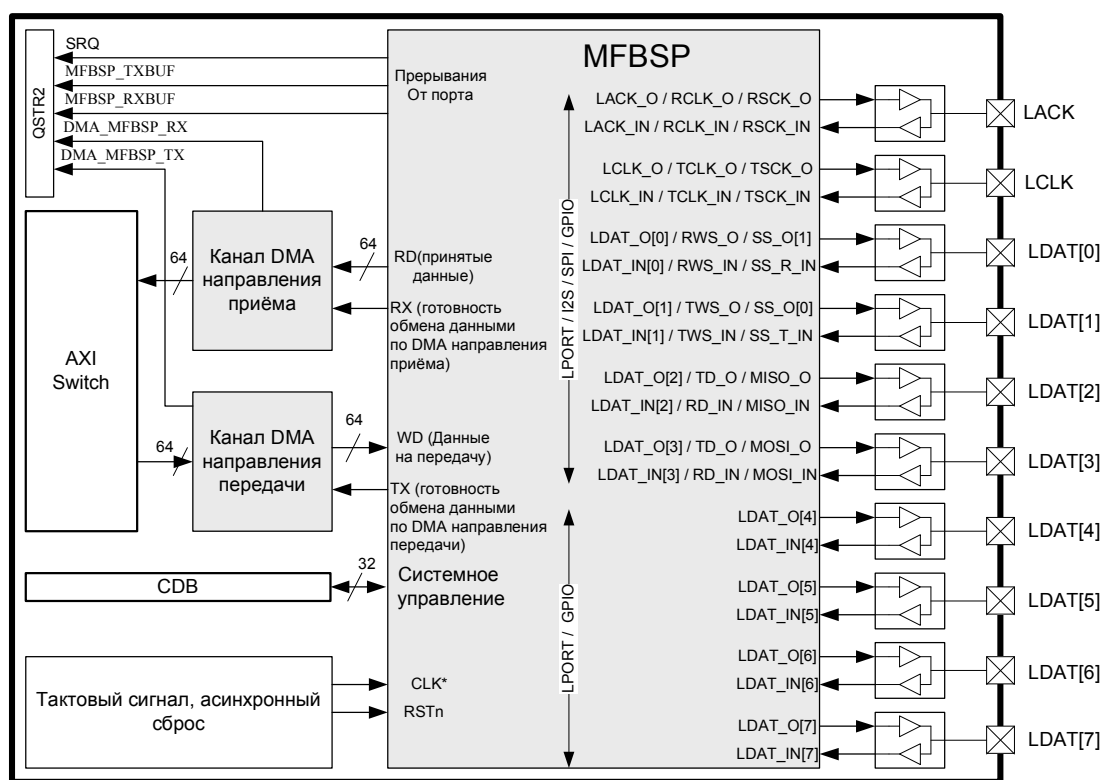


Рисунок 18.1. MFBSP в составе микропроцессора

### 18.1.1 Основные характеристики MFBSF в режиме I2S

В режиме I2S порт позволяет вести дуплексный обмен последовательными данными с внешними устройствами, используя следующие форматы передачи данных: Left-Justified, Right-Justified (при программной предобработке данных), DSP, I2S, FSB (Fast Serial Bus используемый в микросхеме CMX981);

Ограничение использования формата I2S: приемник MFBSF, в текущей версии порта, в режиме ведомого устройства не позволяет принимать слова от передатчика, если число тактов между фронтами сигнала выбора канала меньше чем  $RWORDLEN+1$  (разрядность передаваемых слов меньше, чем установленная разрядность принимаемых передатчиком слов).

#### *Приёмник и передатчик:*

Поддерживается независимая настройка передатчика и приёмника, что позволяет организовать одновременные передачу и прием последовательных данных по разным последовательным интерфейсам и на различных частотах;

Возможен перевод приёмника в зависимый от передатчика режим (когда приемник использует тактовый и контрольный сигналы передатчика), что позволяет задействовать меньшее количество выводов;

Направление любого вывода задается программно, что заметно повышает гибкость при использовании порта;

Тактовые и управляющие сигналы как приемника, так и передатчика можно формировать аппаратными средствами порта MFBSF, либо принимать их от внешнего устройства;

#### *Темп передачи данных:*

Передача данных в режиме I2S может вестись на частотах от  $CLK/2$  до  $CLK/(2*2^{10})$  (где  $CLK$  – тактовая частота, подаваемая на порт со стороны системы);

Частоту контрольного сигнала ( $TWS/RWS$ ) можно задавать в пределах от  $ICLK/2$  до  $ICLK/(2*2^{16})$ , где  $ICLK$  – рабочая частота интерфейса ( $TCLK$  для передатчика и  $RCLK$  для приемника);

#### *Приём и передача данных:*

Порт позволяет принимать и передавать слова длиной от 2-х до 32-х бит, как младшим, так и старшим битом вперед;

В режиме I2S поддерживается режим паковки/распаковки 32-х разрядного слова в два 16-ти разрядных с автоматическим определением левого/правого канала;

Специальная логика обмена позволяет обнулять или дополнять старшим разрядом избыточные биты при чтении принятых слов длиной меньше 32 в обычном режиме и длиной меньше 16 в режиме паковки;

Порт поддерживает приём и передачу данных фреймами с синхронизацией начала каждого фрейма. Число слов в одном фрейме может быть выбрано в пределах от 1 до 64;

#### *Буферы приёма и передачи:*

Используется буферизация в направлении передачи на 18 32-разрядных слов;

Используется буферизация в направлении приёма на 18 32-разрядных слов;

Доступ к буферам приёма и передачи возможен как в 32-х разрядном режиме (обмен данными непосредственно с CPU), так и в 64-х разрядном режиме с использованием каналов DMA;

Степень заполнения буфера передачи, при которой инициируется загрузка данных в порт с помощью DMA, регулируется программно;

Для каждого порта MFBSP предусмотрено два независимых канала DMA на приём и на передачу.

### **18.1.2 Основные характеристики MFBSP в режиме SPI**

В режиме SPI порт позволяет вести дуплексный обмен последовательными данными с внешними устройствами, порт поддерживает 4 формата передачи SPI (для всех сочетаний CPOL и CPHA по спецификации Motorola), при этом возможна передача данных как по стандарту Microwire (SDO, SDI), так и по стандарту Motorola (MOSI, MISO), а также по интерфейсу C-BUS (аналог SPI);

#### *Приёмник и передатчик:*

Поддерживается независимая настройка передатчика и приёмника, что позволяет организовать одновременные передачу и прием последовательных данных по разным последовательным интерфейсам и на различных частотах;

Возможен перевод приёмника в зависимый от передатчика режим (когда приемник использует тактовый и контрольный сигналы передатчика), что позволяет задействовать меньшее количество выводов;

Направление любого вывода задается программно, что заметно повышает гибкость при использовании порта;

#### *Шина выбора ведомых устройств:*

Тактовые сигналы и сигналы шины выбора ведомых устройств можно формировать аппаратными средствами порта MFBSP, программно управлять шиной выбора ведомых устройств, либо принимать тактовые сигналы и сигнал выбора ведомого от внешнего устройства;

В режиме ведущего устройства портом используется двухразрядная шина выбора ведомых устройств, что позволяет параллельно подключать до двух ведомых SPI устройств;

В режиме ведомого устройства возможен независимый выбор внешним устройством приёмника и передатчика MFBSP;

#### *Темп передачи данных:*

Передача данных в режиме SPI может вестись на частотах от  $CLK/2$  до  $CLK/(2*2^{10})$  (где CLK – тактовая частота, подаваемая на порт со стороны системы);

#### *Приём и передача данных:*

Порт позволяет принимать и передавать слова длиной от 2-х до 32-х бит, как младшим, так и старшим битом вперед;

Специальная логика обмена позволяет обнулять или дополнять старшим разрядом избыточные биты при чтении принятых слов длиной меньше 32 бит;

Порт позволяет вести обмен данными в режиме автоматического формирования сигналов выбора ведомого, с возможностью передачи от 1 до 64 слов без изменения уровня сигнала выбора ведомого;

#### *Буферы приёма и передачи:*

Используется буферизация в направлении передачи на 18 32-разрядных слов;

Используется буферизация в направлении приёма на 18 32-разрядных слов;

Доступ к буферам приёма и передачи возможен как в 32-х разрядном режиме (обмен данными непосредственно с CPU), так и в 64-х разрядном режиме с использованием каналов DMA;

Степень заполнения буфера передачи, при которой инициируется загрузка данных в порт с помощью DMA, регулируется программно;

Для каждого порта MFBSF предусмотрено два независимых канала DMA на приём и на передачу;

В данной реализации порта существует ограничение на выбор направления выводов в режиме SPI: тактовый и управляющий сигналы в режиме SPI должны быть либо оба заданы как вход, либо оба заданы как выход;

В данной реализации порта не предусмотрена возможность соединения нескольких микропроцессоров по цепочке с использованием SPI интерфейса. микропроцессор может только управлять загрузкой последовательных данных в другие ведомые устройства, соединенные по цепочке.

### **18.1.3 Основные характеристики MFBSF в режиме LPORT**

В режиме LPORT порт позволяет вести обмен с внешними устройствами по линковому интерфейсу (совместимому с ADSP21160 LINK PORT).

#### *Приёмник и передатчик:*

В режиме LPORT MFBSF может работать либо только как передатчик, либо только как приёмник (передача данных в одном направлении);

#### *Темп передачи данных:*

Передача данных по интерфейсу LPORT может вестись на частотах от CLK/32 до CLK/2 (где CLK – тактовая частота, подаваемая на порт со стороны системы). Следует отметить, что в данной микросхеме максимальная частота передачи данных – 50 МГц.

#### *Приём и передача данных:*

По параллельно-последовательному интерфейсу LPORT возможна передача данных как тетрадами, так и байтами;

#### *Буферы приёма и передачи:*

Используется буферизация в направлении передачи на 16 32-разрядных слов;

Используется буферизация в направлении приёма на 18 32-разрядных слов;

Доступ к буферам приёма и передачи возможен как в 32-х разрядном режиме (обмен данными непосредственно с CPU), так и в 64-х разрядном режиме с использованием каналов DMA;

Степень заполнения буфера передачи, при которой инициируется загрузка данных в порт с помощью DMA, регулируется программно.

В зависимости от выбранного направления порта используется либо канал DMA направления приёма, либо канал DMA направления передачи.

### 18.1.4 Основные характеристики MFBSP в режиме порта ввода-вывода общего назначения

В режиме порта ввода-вывода общего назначения все 10 выводов порта могут использоваться как вводы выводы общего назначения;

Направление каждого вывода задаётся программно;

В режиме последовательного порта(режимы SPI или I2S) 4 незадействованных в передаче последовательных данных вывода MFBSP (LDAT[7:4]) могут быть использованы в качестве вводов-выводов общего назначения.

## 18.2 Общие сведения об MFBSP

### 18.2.1 Режимы работы MFBSP

Многофункциональный порт MFBSP может быть использован как порт ввода-вывода общего назначения, как линковый порт (LPORT), либо как последовательный порт. В случае если MFBSP используется как последовательный порт, приёмник и передатчик могут настраиваться независимо. Как приёмник, так и передатчик MFBSP могут работать в режиме SPI либо в режиме I2S. Таким образом, для MFBSP существует 6 различных режимов работы, которые задаются битами LEN и SPI\_I2S\_EN регистра CSR\_MFBSP, битом TMODE регистра TCTR и битом RMODE регистра RCTR. Режимы работы MFBSP и задающие их сочетания значений управляющих бит приведены в Таблица 18.1.

Таблица 18.1. Режимы работы MFBSP

№	Значение бит, задающих режим				Режим работы MFBSP
	LEN	SPI_I2S_EN	TMODE	RMODE	
1	0	0	x	x	Порт ввода-вывода общего назначения
2	1	0	x	x	Линковый порт(LPORT)
3	0	1	0	0	Последовательный порт Передатчик – I2S Приёмник – I2S
4	0	1	0	1	Последовательный порт Передатчик – I2S Приёмник – SPI
5	0	1	1	0	Последовательный порт Передатчик – SPI Приёмник – I2S
6	0	1	1	1	Последовательный порт Передатчик – SPI Приёмник – SPI

Более подробное описание функциональных особенностей порта для режима I2S приведено в параграфе 18.3

Более подробное описание функциональных особенностей порта для режима SPI приведено в параграфе 18.4

Более подробное описание функциональных особенностей порта для режима LPORT приведено в параграфе 18.5

Более подробное описание функциональных особенностей порта для режима порта ввода-вывода общего назначения приведено в параграфе 18.6



## 18.2.2 Структурная схема многофункционального буферизированного последовательного порта

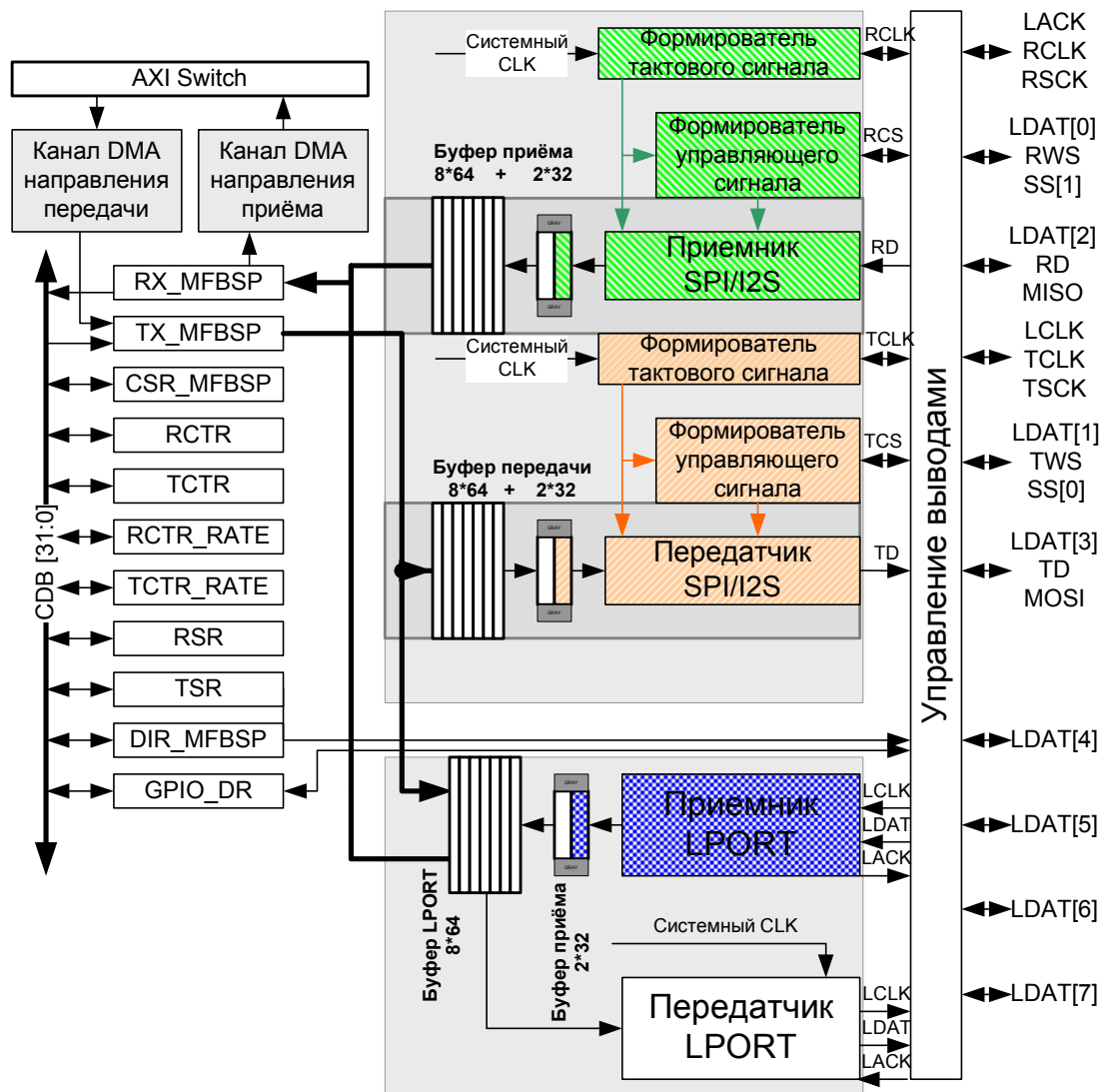


Рисунок 18.2 Структурная схема MFBSP (Защищена патентом РФ №2360282 от 27 июня 2009 года)

На Рисунок 18.1 показан MFBSP в составе микропроцессора. Порт поддерживает дуплексный обмен последовательными данными, поэтому для каждого MFBSP предусмотрено два канала DMA – один на приём и один на передачу. Каждый из внешних выводов порта двунаправленный, направление каждого вывода задается независимо.

На Рисунок 18.2 представлена более подробная структурная схема MFBSP.

В состав совмещенного контроллера входят два основных блока: **контроллер LPORT** и **контроллер SPI/I2S**. Включение **контроллера LPORT** производится установкой бита LEN, регистра CSR\_MFBSP в 1, включение **контроллера SPI/I2S** производится установкой бита SPI\_I2S\_EN, регистра CSR\_MFBSP в 1 (Таблица 18.1). Одновременная работа блоков LPORT и SPI/I2S и соответственно установка бит LEN и SPI\_I2S\_EN в 1 не допускается.

В состав **контроллера SPI/I2S** входят приёмник, передатчик, буфер приёма и буфер передачи. Приёмник и передатчик могут работать одновременно и независимо. Приёмник осуществляет синхронный приём последовательного кода с внешнего вывода схемы и запись принятых данных в буфер приёма. Передатчик осуществляет чтение данных из буфера пере-

дачи и синхронную выдачу их последовательным кодом на внешний вывод схемы. Запись передаваемых данных в буфер передачи осуществляется при записи по адресу псевдорегистра TX\_MFBSP (доступ со стороны CPU или DMA направления передачи), чтение принятых данных из буфера приёма осуществляется при чтении по адресу псевдорегистра RX\_MFBSP (доступ со стороны CPU или DMA направления приёма).

Последовательным портом при обмене данными используется только 6 выводов LCLK, LACK, LDAT[3:0]. Если порт работает в режиме SPI/I2S, выходы LDAT[4:7] могут использоваться как входы-выводы общего назначения.

В состав **контроллера LPORT** входят приёмник, передатчик и буфер LPORT. В зависимости от направления обмена данными работает либо приёмник, либо передатчик. В зависимости от направления обмена данными буфер LPORT выполняет функции либо буфера приёма, либо буфера передачи. Приёмник осуществляет синхронный приём параллельно-последовательного кода с внешних выводов схемы и запись принятых данных в буфер LPORT. Передатчик осуществляет чтение данных из буфера LPORT и синхронную выдачу их параллельно последовательным кодом на внешние выводы схемы. Запись передаваемых данных в буфер LPORT осуществляется при записи по адресу псевдорегистра TX\_MFBSP (доступ со стороны CPU или DMA направления передачи), чтение принятых данных из буфера LPORT осуществляется при чтении по адресу псевдорегистра RX\_MFBSP (доступ со стороны CPU или DMA направления приёма).

LPORT при обмене данными использует выходы LCLK, LACK, LDAT[7:0].

MFBSP использует системный тактовый сигнал CPU (CLK), при этом на MFBSP0 тактовый сигнал CLK подается постоянно, когда включен тактовый сигнал CPU, что позволяет реализовать режим начальной загрузки через MFBSP0. Для MFBSP1, MFBSP2, MFBSP3 и DMA MFBSP есть возможность программно включать и выключать подачу тактового сигнала.

Включение частоты портов происходит не моментально, поэтому чтение из регистров или запись в регистры MFBSP сразу после команды включения частоты MFBSP может привести к ошибкам. Чтобы убедиться, что обращение к регистрам происходит после фактического включения частоты необходимо прочитать регистр CLK\_EN и провести с прочитанными данными любые действия, например:

```
sw r26, CLK_EN //включение частоты
lw r26, CLK_EN //чтение состояния CLK_EN
or r26, r26 //обработка прочитанных данных
```

При отключенной частоте MFBSP чтение и запись в регистры MFBSP1-MFBSP3 не допускается.

### 18.2.3 Назначение выводов порта в различных режимах

Таблица 19.2 содержит наименования выводов порта для каждого из режимов – LPORT, SPI, I2S. Таблица 19.3 содержит информацию о назначении каждого вывода в различных режимах.

**Таблица 19.2. Обозначение выводов порта для различных режимов работы**

LPORT	I2S	SPI
LDAT[7]	-	-
LDAT[6]	-	-
LDAT[5]	-	-
LDAT[4]	-	-
LDAT[3]	TD	MOSI
LDAT[2]	RD	MISO
LDAT[1]	TWS	SS[0]
LDAT[0]	RWS	SS[1]
LCLK	TCLK	TSCK
LACK	RCLK	RSCK

**Таблица 19.3. Назначение выводов порта в различных режимах**

Наименование вывода	Режим работы порта	Направление вывода	Назначение вывода
LDAT[7:0]	LPORT	IO	Внешняя шина данных LPORT.
LCLK	LPORT	IO	Тактовый сигнал LPORT
LACK	LPORT	IO	Подтверждение готовности приема
TD	I2S	IO	Передаваемые последовательные данные
RD	I2S	IO	Принимаемые последовательные данные
TCLK	I2S	IO	Тактовый сигнал передатчика I2S
RCLK	I2S	IO	Тактовый сигнал приемника I2S
TWS	I2S	IO	Сигнал выбора канала для передаваемых данных
RWS	I2S	IO	Сигнал выбора канала для принимаемых данных
MOSI	SPI	IO	Вывод последовательных данных. Направление вывода определяется программно
MISO	SPI	IO	Вывод последовательных данных. Направление вывода определяется программно
TSCK	SPI	IO	Тактовый сигнал передатчика SPI
RSCK	SPI	IO	Тактовый сигнал приемника SPI
SS [0]	SPI	IO	<i>В режиме ведущего:</i> Сигнал выбора устройства 0. <i>В режиме ведомого:</i> сигнал выбора ведомого. Низкий уровень на входе SS[0] обозначает, что передатчику MFBSP необходимо выдавать последовательные данные (если приёмник MFBSP находится в зависимом от передатчика режиме, то активизируется и приёмник).
SS [1]	SPI	IO	<i>В режиме ведущего:</i> Если приёмник в зависимом от передатчика режиме - сигнал выбора устройства 1. Если передатчик в независимом от приёмника режиме – сигнал выбора приёмником устройства 0. <i>В режиме ведомого:</i> Сигнал выбора ведомого. Только в случае когда приёмник в независимом от передатчика режиме. Низкий уровень на входе SS[1] обозначает, что приёмнику MFBSP необходимо принимать последовательные данные.

### 18.2.4 Перечень регистров MFBSP

Таблица 18.2 содержит перечень регистров многофункционального порта.

Таблица 18.2 Перечень регистров многофункционального буферизованного порта

Условное обозначение регистра	Название регистра	Доступ
<b>TX_MFBSP</b>	Буфер передачи данных	W
<b>RX_MFBSP</b>	Буфер приёма данных	R
<b>CSR_MFBSP</b>	Регистр управления и состояния	RW
<b>DIR_MFBSP</b>	Регистр управления направлением выводов порта ввода-вывода	RW
<b>GPIO_DR</b>	Регистр данных порта ввода-вывода	RW
<b>TCTR</b>	Регистр управления передатчиком	RW
<b>RCTR</b>	Регистр управления приёмником	RW
<b>TSR</b>	Регистр состояния передатчика	RW
<b>RSR</b>	Регистр состояния приёмника	RW
<b>TCTR_RATE</b>	Регистр управления темпом передачи данных	RW
<b>RCTR_RATE</b>	Регистр управления темпом приёма данных	RW
<b>TSTART</b>	псевдорегистр <code>ten</code> – запуск/останов передатчика без изменения настроек передатчика	RW
<b>RSTART</b>	псевдорегистр <code>gen</code> – запуск/останов приемника без изменения настроек приемника	RW
<b>EMERG_MFBSP</b>	Регистр аварийного управления портом	RW
<b>IMASK_MFBSP</b>	Регистр маски прерываний от порта	RW

### 18.2.5 Каналы DMA многофункциональных портов MFBSP

Для каждого порта предусмотрено два канала DMA – один для приема данных, другой для передачи данных.

По каналу DMA направления передачи осуществляется передача данных внешнему устройству, подключенному к микропроцессору через MFBSP. По каналу DMA направления приёма осуществляется приём данных из внешнего устройства, подключенного к микропроцессору через MFBSP.

При обмене данными через MFBSP с использованием DMA максимальный размер пачки составляет 8 64-разрядных слов. Если значение бит `WN` в контрольном регистре DMA превосходит максимальный размер пачки, то `WN` автоматически корректируется (большая пачка передается в несколько этапов пачками меньшего размера).

По умолчанию при работе передатчика с DMA заполнение буфера передачи происходит до тех пор, пока буфер в состоянии принять очередную пачку, размером `WN`. Однако имеется возможность программно регулировать степень заполнения буфера передачи, путем установки бит `TBES`, регистра `TSR`. В этом случае значение выражения `TBES+1` – задает эффективный размер буфера передачи. Передача очередной пачки происходит только в случае, если при записи этой пачки в буфер передачи число 64 разрядных слов в буфере передачи не превысит `TBES+1`. При попытке передать пачку со значением `WN > TBES`, значение `WN` автоматически корректируется (большая пачка передается в несколько этапов пачками меньшего размера).

По умолчанию при работе приёмника с DMA, считывание данных из буфера приёма происходит если в буфере чтения содержится число слов большее, либо равное размеру пачки (`WN`). Степень заполнения буфера приёма, при которой начинается откачка данных с помощью DMA регулируется установкой значения `WN` соответствующего канала DMA.

### 18.2.6 Прерывания от каналов DMA MFBSPP

Бит DMA\_MFBSPP\_RX, регистра QSTR2, устанавливается, если есть прерывание от соответствующего порту канала DMA направления приёма.

Бит DMA\_MFBSPP\_TX, регистра QSTR2, устанавливается, если есть прерывание от соответствующего порту канала DMA направления передачи.

Если соответствующий канал DMA разрешен, то прерывания от канала DMA формируются по завершению передачи или приема всего блока данных.

### 18.2.7 Прерывания от MFBSPP

Бит MFBSPP\_TXBUF, регистра QSTR2, устанавливается в случае, если число 64-х разрядных слов, находящихся в буфере передачи, меньше, либо равно пороговому значению TLEV, задаваемому в регистре TSR (Рисунок 18.3). Для установки бита MFBSPP\_TXBUF также необходимо, чтобы линковый порт был включен на передачу (LEN=1 и LTRAN=1) либо включен передатчик SPI/I2S (SPI\_I2S\_EN=1, TEN=1) и разрешена установка прерывания MFBSPP\_TXBUF по условию превышения порога (TX\_LEV\_IRQ\_EN). MFBSPP\_TXBUF также устанавливается в случае, если имела место ошибка передачи TERR и разрешена установка прерывания MFBSPP\_TXBUF при ошибке передачи (чтение из пустого буфера передачи) : TX\_ERR\_IRQ\_EN.

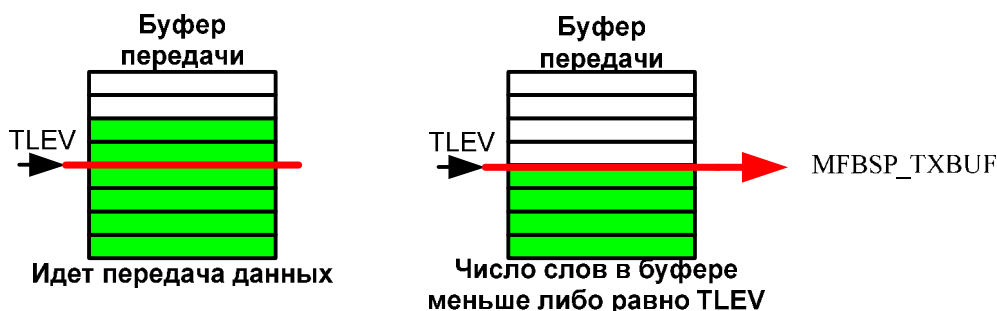
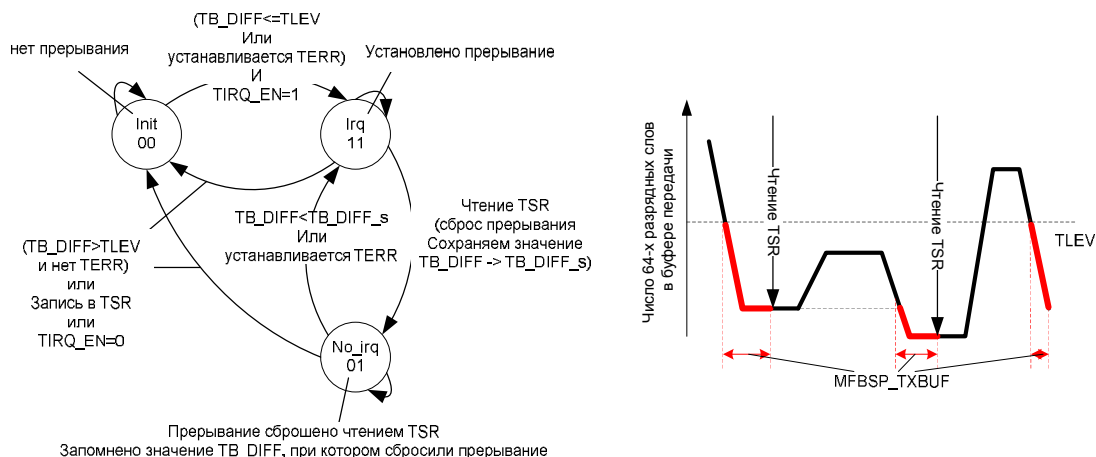


Рисунок 18.3 Назначение бит TLEV, регистра TSR

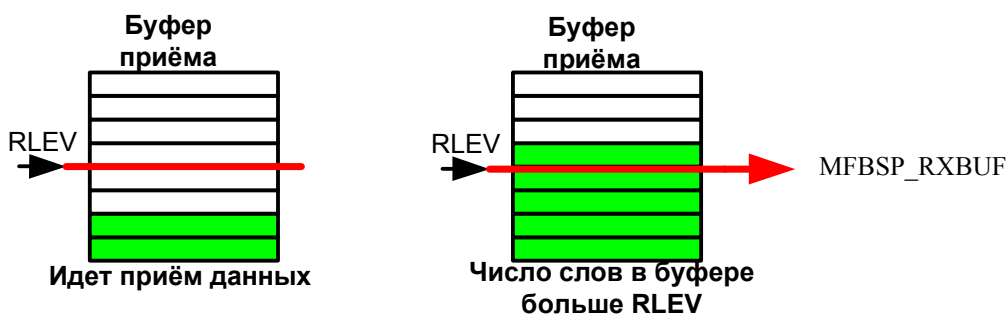
MFBSPP\_TXBUF может формироваться в двух режимах: с автоматическим сбросом при чтении регистра TSR и без автоматического сброса при чтении TSR. Данный режим определяется состоянием бита TXBUF\_R\_EN, регистра IMASK\_MFBSPP.

В случае если выбран режим без автоматического сброса прерывание установлено всегда, когда имеет место превышение уровня TLEV. В случае если установлен режим с автоматическим сбросом управление битом MFBSPP\_TXBUF осуществляется следующим образом: прерывание MFBSPP\_TXBUF автоматически сбрасывается, если число 64-х разрядных слов, находящихся в буфере передачи, становится больше порогового значения TLEV и при этом во время передачи не возникало ошибки (TERR = 0). Даже если описанное условие не выполнено, прерывание можно программно сбросить, прочитав регистр TSR. В этом случае прерывание сбросится и запомнится текущее значение слов в буфере передачи. Если число слов в буфере передачи начнет уменьшаться или произойдет ошибка передачи, то прерывание снова установится. Увеличение числа слов в буфере передачи не приведет к установке прерывания, даже, если число слов в буфере ниже порога TLEV (Рисунок 18.4).



**Рисунок 18.4** Механизм установки и сброса прерывания MFBSB\_TXBUF. На рисунке  $TIRQ\_EN = (LEN \& LTRAN \parallel TEN \& SPI\_I2S\_EN)$

Бит MFBSB\_RXBUF, регистра QSTR2, устанавливается в случае, если число 64-х разрядных слов в буфере приёма больше чем пороговое значение RLEV, задаваемое в регистре RSR (Рисунок 18.5). Для установки бита MFBSB\_RXBUF также необходимо, чтобы линковый порт был включен на приём ( $LEN=1$  и  $LTRAN=0$ ) либо включен приёмник SPI/I2S ( $SPI\_I2S\_EN=1$ ,  $REN=1$ ) и разрешена установка прерывания MFBSB\_RXBUF по условию превышения порога прерывания ( $TX\_LEV\_IRQ\_EN$ ). MFBSB\_RXBUF также устанавливается в случае, если имела место ошибка приёма RERR и разрешена установка прерывания MFBSB\_RXBUF при ошибке передачи (запись в полный буфер приёма) :  $RX\_ERR\_IRQ\_EN$ .

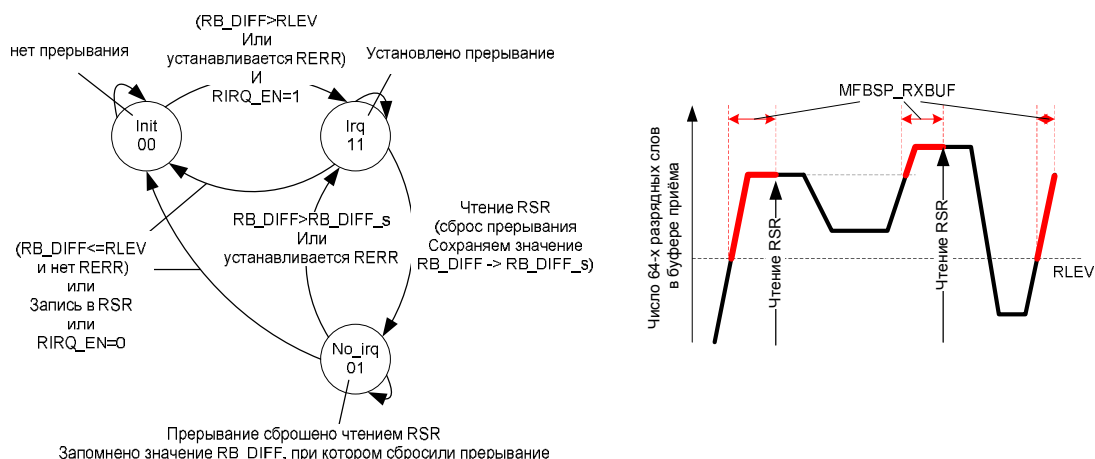


**Рисунок 18.5.** Назначение бит RLEV, регистра RSR

MFBSB\_RXBUF может формироваться в двух режимах: с автоматическим сбросом при чтении регистра RSR и без автоматического сброса при чтении RSR. Данный режим определяется состоянием бита RXBUF\_R\_EN, регистра IMASK\_MFBSB.

В случае если выбран режим без автоматического сброса прерывание установлено всегда, когда имеет место превышение уровня RLEV. В случае если установлен режим с автоматическим сбросом управление битом MFBSB\_RXBUF осуществляется следующим образом: прерывание MFBSB\_RXBUF автоматически сбрасывается, если число 64-х разрядных слов, находящихся в буфере приёма, становится меньше порогового значения RLEV и при этом во время приёма не возникало ошибки ( $RERR = 0$ ). Даже если описанное условие не выполнено, прерывание можно программно сбросить, прочитав регистр RSR. В этом случае прерывание сбросится и запомнится текущее значение слов в буфере чтения. Если число слов в буфере чтения начнет увеличиваться, то прерывание снова установится. Уменьшение числа слов в буфере чтения не приведет к установке прерывания, даже, если превышен порог RLEV (Рисунок 18.6).





**Рисунок 18.6** Механизм установки и сброса прерывания MFBSP\_RXBUF. На рисунке RIRQ\_EN = (LEN & !LTRAN || REN & SPI\_I2S\_EN)

Бит SRQ, регистра QSTR2, формируется при запросе на обслуживание, если порт MFBSP выключен (LEN=0, SPI\_I2S\_EN=0) и на выводах LACK или LCLK высокий уровень, при условии, что разрешено прерывание по запросу на обслуживание (LPT\_IRQ\_EN=1).

## 18.3 Работа MFBSP в режиме I2S

### 18.3.1 Назначение MFBSP в режиме I2S

Режим I2S буферизированного последовательного порта предназначен для организации дуплексного обмена аудиоданными с внешними устройствами последовательным кодом.

Порт в режиме I2S позволяет одновременно передавать и принимать последовательные данные. Приемник и передатчик контроллера настраиваются независимо, при этом возможен перевод приёмника в зависимое от передатчика состояние.

Порт поддерживает передачу аудиоданных в формате I2S, с поочередной передачей левого и правого каналов, а также передачу данных фреймами от 1 до 64 слов в каждом фрейме.

Поддерживается независимое задание направления каждого из выводов порта, осуществляемое установкой соответствующих бит регистра DIR\_MFBSP.

Ограничение использования формата I2S: приемник MFBSP, в текущей версии порта, в режиме ведомого устройства не позволяет принимать слова от передатчика, если число тактов между фронтами сигнала выбора канала меньше чем RWORDLEN+1 (разрядность передаваемых слов меньше, чем установленная разрядность принимаемых передатчиком слов).

### 18.3.2 Регистр управления и состояния CSR\_MFBSP (режим I2S)

Регистр CSR\_MFBSP (Таблица 18.3) используется для включения режима последовательного порта и разрешения прерываний от MFBSP.

**Таблица 18.3** Назначение разрядов регистра CSR\_MFBSP в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:15	-	Резерв	-	0
14:11	-	В режиме I2S не используется	-	0
10	-	Резерв	-	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
9	SPI_I2S_EN	Включение режима SPI/I2S: 0 – Работа в режиме LPORT 1 – Работа в режиме SPI/I2S	RW	0
8:5	-	В режиме I2S не используется	-	0
4:3	LSTAT	Состояние буфера: При LTRAN = 0 показывает состояние буфера приёма При LTRAN = 1 показывает состояние буфера передачи 00 – буфер пуст; 10 – буфер не пуст; 11 – буфер полон.	R	0
2	-	В режиме I2S не используется	-	0
1	LTRAN	Назначение бит LSTAT: 0 - LSTAT отображает состояние буфера приёма 1 - LSTAT отображает состояние буфера передачи	RW	0
0	LEN	В режиме I2S должен быть установлен в 0	RW	0

### 18.3.3 Регистр управления направлением выводов DIR\_MFBSP (режим I2S)

Регистр управления направлением выводов DIR\_MFBSP (Таблица 18.4) предназначен для индивидуальной настройки направления каждого вывода последовательного порта.

Таблица 18.4 Назначение разрядов регистра DIR\_MFBSP в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
9:6	-	Не используется в режиме I2S	-	0
5	TD_DIR	Направление вывода TD: 0 – TD – вход (при RD_DIR = 1 последовательные данные принимаются со входа TD) 1 – TD – выход (TD – является выходом для передачи последовательных данных)	RW	0
4	RD_DIR	Направление вывода RD: 0 – RD – вход (последовательные данные принимаются со входа RD) 1 – RD – выход (RD – является выходом для передачи последовательных данных)	RW	0
3	TCS_DIR	Направление вывода TWS: 0 – TWS – вход (Сигнал выбора слова TWS принимается от внешнего источника) 1 – TWS – выход (Сигнал выбора слова TWS формируется передатчиком)	RW	0
2	RCS_DIR	Направление вывода RWS: 0 – RWS – вход (Сигнал выбора слова RWS принимается от внешнего источника) 1 – RWS – выход (Сигнал выбора слова RWS формируется приёмником)	RW	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
1	TCLK_DIR	Направление вывода TCLK: 0 – TCLK – вход (тактовый сигнал TCLK принимается от внешнего источника) 1 – TCLK – выход (тактовый сигнал TCLK формируется передатчиком)	RW	0
0	RCLK_DIR	Направление вывода RCLK: 0 – RCLK – вход (тактовый сигнал RCLK принимается от внешнего источника) 1 – RCLK – выход (тактовый сигнал RCLK формируется приёмником)	RW	0

**примечание:** при RD\_DIR = 0 и TD\_DIR = 0 данные снимаются с RD, при RD\_DIR = 1 и TD\_DIR = 1 на TD и RD выдаются одинаковые данные с передатчика.

### 18.3.4 Регистр управления приёмником RCTR (режим I2S)

Таблица 18.5 Назначение разрядов регистра RCTR в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	-	Резерв	-	0
29	RCS_CONT	Включение непрерывного формирования сигнала RWS: 0 – RWS – Формируется если буфер приёма не полон. По заполнении буфера приёма формирование сигнала RWS прекращается. 1 – RWS – формируется непрерывно, если установлен бит REN	RW	0
28	RCLK_CONT	Включение непрерывного формирования сигнала RCLK: 0 – RCLK – формируется только во время приёма (пока буфер приёма не полон). Если буфер приёма полон – сигнал не формируется 1 – RCLK – формируется непрерывно, если установлен бит REN	RW	0
27	RSWAP	Порядок упаковки в 32 разрядное слово, перед записью в буфер приёма: 0 – левый канал пишется в старшие 16 разрядов 1 – левый канал пишется в младшие 16 разрядов (Используется в режиме с включенным паковщиком)	RW	0
26	RSIGN	Значение заполнителя: Если длина принимаемого слова меньше 32 при отключенном паковщике или меньше 16 при включенном паковщике, то неиспользуемые биты принятого слова заполняются При RSIGN = 0 нулями При RSIGN = 1 значением старшего разряда в принятом слове	RW	0
25	RPACK	Включение режима паковки: 0 – режим паковки выключен. Данные, принятые по каждому из каналов пишутся отдельным 32-разрядным словом в буфер приёма 1 – режим паковки включен. Данные, принятые по левому и правому каналу пакуются в 32-х разрядное слово. При этом разрядность принимаемых слов не должна превышать 16.	RW	0
24:20	RWORDLEN	Длина принимаемого слова: Число бит в принимаемом слове равно RWORDLEN + 1. RWORDLEN должно быть больше 0.	RW	5'b0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
19	RMBF	Порядок передачи бит: 0 – младшим битом вперед 1 – старшим битом вперед	RW RW	1
18	RCSNEG	Полярность управляющего сигнала приёмника: При RDSPMODE=0: RCSNEG = 0 – левый канал принимается при высоком уровне RWS RCSNEG = 1 – левый канал принимается при низком уровне RWS каждый фронт контрольного сигнала является активным и инициирует приём нового слова. При RDSPMODE=1: задаёт полярность активного фронта: RCSNEG = 0 - передний фронт активный; RCSNEG = 1 - задний фронт активный;	RW	0
17:12	RWORDCNT	Число слов во фрейме: Определяет число принимаемых в течении одного фрейма слов. Число принимаемых слов равно RWORDCNT + 1. Число бит, принимаемых в пределах одного фрейма, равно (RWORDCNT + 1)*(RWORDLEN+1) При RPACK = 1 обязательно RWORDCNT = 0	RW	0
11	RDEL	Задержка начала приёма данных на такт: 0 – захват бит принимаемого слова начинается по первому после активного фронта управляющего сигнала RWS фронту приёма такого сигнала RCLK (используется для передачи в форматах Left-Justified и Right-Justified) 1 – захват бит принимаемого слова начинается по второму после активного фронта управляющего сигнала RWS фронту приёма такого сигнала RCLK (используется для передачи в формате I2S)	RW	0
10	RNEG	Полярность тактового сигнала приёмника: Задаёт исходное состояние вывода RCLK и фронт, по которому осуществляется захват данных приёмником (фронт приёма) 0 – захват данных по заднему фронту RCLK. 1 – захват данных по переднему фронту RCLK. Исходное состояние RCLK = RNEG.	RW	0
9	RDSPMODE	Формат передачи данных: 0 – передача в формате I2S 1 – передача в формате DSP	RW	0
8:4	-	Резерв	-	0
3	RCS_CP	Дублирование сигнала TWS: 0 – выводы TWS и RWS независимы 1 – сигнал RWS, идущий на блок приёмника, дублирует TWS	RW	0
2	RCLK_CP	Дублирование TCLK: 0 – выводы TCLK и RCLK независимы 1 – сигнал RCLK, идущий на блок приёмника, дублирует TCLK	RW	0
1	RMODE	Режим работы приёмника: 0 – режим I2S 1 – режим SPI	RW	0
0	REN	Разрешение работы приёмника: 0 – приемник выключен 1 – приемник включен	RW	0

### 18.3.5 Регистр управления передатчиком TCTR (режим I2S)

Таблица 18.6 Назначение разрядов регистра TCTR в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	-	В режиме I2S не используется	-	0
29	TCS_CONT	Включение непрерывного формирования сигнала TWS: 0 – TWS – формируется только если буфер передачи не пуст. После передачи последнего слова из буфера передачи формирование сигнала TWS прекращается 1 – TWS – формируется непрерывно, если установлен бит TEN	RW	0
28	TCLK_CONT	Включение непрерывного формирования сигнала TCLK: 0 – TCLK – формируется только во время передачи. Если буфер передачи пуст – сигнал не формируется 1 – TCLK – формируется непрерывно, если установлен бит TEN	RW	0
27	TSWAP	Порядок распаковки 32-х разрядного слова: Определяет порядок распаковки из 32 разрядного слова 0 – в левый канал передаются старшие 16 разрядов 1 – в левый канал передаются младшие 16 разрядов (Используется в режиме с включенным распаковщиком)	RW	0
26	-	Резерв	-	0
25	TPACK	Включение режима распаковки: 0 – режим распаковки выключен. Каждое слово из буфера передачи используется для одной передачи по одному каналу 1 – режим распаковки включен. Слово из буфера передачи передается двумя посылками (по левому и правому каналу). При этом разрядность передаваемых слов не должна превышать 16 бит	RW	0
24:20	TWORDLEN	Длина передаваемого слова: Число бит в передаваемом слове равно TWORDLEN + 1. TWORDLEN должно быть больше 0.	RW	5'b0
19	TMBF	Порядок передачи бит: 0 – младшим битом вперед 1 – старшим битом вперед	RW	1
18	TCSNEG	Полярность управляющего сигнала передатчика: При TDSPMODE=0: TCSNEG = 0 – Левый канал передаётся с высоким уровнем TWS TCSNEG = 1 – Левый канал передаётся с низким уровнем TWS каждый фронт контрольного сигнала является активным и инициирует передачу нового слова. При TDSPMODE=1: задаёт полярность активного фронта: TCSNEG = 0 – передний фронт активный; TCSNEG = 1 – задний фронт активный;	RW	0
17:12	TWORDCNT	Число слов во фрейме: Определяет число передаваемых в течении одного фрейма слов. Число передаваемых слов равно TWORDCNT + 1. Число бит, передаваемых в пределах одного фрейма, равно (TWORDCNT + 1)*(TWORDLEN+1) При TPACK=1 обязательно TWORDCNT=0	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
11	TDEL	Задержка начала передачи данных на такт: 0 – выдача первого бита передаваемого слова начинается по первому после активного фронта управляющего сигнала TWS фронту выдачи такового сигнала TCLK (используется для передачи в форматах Left-Justified и Right-Justified) 1 – выдача первого бита передаваемого слова начинается по второму после активного фронта управляющего сигнала TWS фронту выдачи такового сигнала TCLK (используется для передачи в формате I2S)	RW	0
10	TNEG	Полярность тактового сигнала передатчика: Задает исходное состояние вывода TCLK и фронт, по которому осуществляется выдача данных передатчиком (фронт выдачи) 0 – выдача данных по переднему фронту TCLK. 1 – выдача данных по заднему фронту TCLK. Исходное состояние TCLK = TNEG.	RW	0
9	TDSPMODE	Формат передачи данных: 0 – передача в формате I2S 1 – передача в формате DSP	RW	0
8:4	-	Резерв	-	0
3	-	В режиме I2S не используется	-	0
2	TD_ZER_EN	Обнуление избыточных бит передаваемого слова: 0 – Если длина слова меньше размеров окна, отведенного под передачу слова, после передачи всех бит слова на внешней шине данных остаётся значение нулевого бита передаваемого слова. 1 – Если длина слова меньше размеров окна, отведенного под передачу слова, после передачи всех бит слова на внешнюю шину данных подаётся 0, вплоть до начала передачи следующего слова. ВНИМАНИЕ! Режим с включенным обнулением избыточных бит при передаче слова корректно функционирует только при условии, что частота последовательного порта TCLK $\leq$ CLK/4, где CLK – рабочая частота подаваемая на порт, со стороны системы.	RW	0
1	TMODE	Режим работы передатчика: 0 – режим I2S 1 – режим SPI	RW	0
0	TEN	Разрешение работы передатчика: 0 – передатчик выключен 1 – передатчик включен	RW	0

### 18.3.6 Регистр состояния приёмника RSR (режим I2S)

Таблица 18.7 Назначение разрядов регистра RSR в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	резерв	-	0
27:24	RB_DIFF	Количество принятых 64-разрядных слов в буфере приёма (макс 8).	R	0
23:19	-	Резерв	-	0
18:16	RLEV	Порог прерывания от буфера приёма: Прерывание формируется если число принятых 64-х разрядных слов больше RLEV	RW	7
15:11	-	Резерв	-	0
10	RXBUF	Результирующее прерывание MFBSP_RXBUF	R	0
9	RXBUF_D	Прерывание MFBSP_RXBUF без механизма автоматического сброса при чтении RSR	R	0
8	RXBUF_R	Прерывание MFBSP_RXBUF с механизмом автоматического сброса при чтении RSR	R	0
7	RRUN	Идёт приём: 0 – приёмник в состоянии ожидания 1 – идёт приём очередного слова	R	0
6	RERR	Ошибка передачи: 0 – приём проходил в штатном режиме 1 – была запись в полный буфер приёма (потеря данных). Флаг сбрасывается записью 0 в 6-й разряд регистра RSR.	RW	0
5	RSBF	Буфер пересинхронизации в направлении приёма полон: 0 – буфер пересинхронизации в направлении приёма не полон 1 – буфер пересинхронизации в направлении приёма полон	R	0
4	RSBE	Буфер пересинхронизации в направлении приёма пуст: 0 – буфер пересинхронизации в направлении приёма не пуст 1 – буфер пересинхронизации в направлении приёма пуст	R	1
3	RBHL	Достигнут порог прерывания в буфере приёма: 1 – число 64-х разрядных слов в буфере приёма больше чем задано в RLEV 0 – число 64-х разрядных слов в буфере приёма меньше либо равно RLEV	R	0
2	RBHF	Буфер приёма полон на половину или более: 1 – буфер приёма заполнен на половину или больше (из буфера приёма можно считать как минимум 4 слова) 0 – буфер приёма заполнен меньше чем на половину	R	0
1	RBF	Буфер приёма полон: 0 – буфер приёма не полон 1 – буфер приёма полон	R	0
0	RBE	Буфер приёма пуст: 0 – буфер приёма не пуст 1 – буфер приёма пуст	R	1

### 18.3.7 Регистр состояния передатчика TSR (режим I2S)

**Таблица 18.8 Назначение разрядов регистра TSR в режиме I2S**

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	резерв	-	0
27:24	TB_DIFF	Количество свободных 64-разрядных позиций в буфере передачи (в буфер передачи можно записать еще TB_DIFF 64-разрядных слов).	R	8
23	-	Резерв	-	0
22:20	TBES	Эффективный размер буфера передачи Актуален только для режима работы с DMA. Значение TBES+1 – определяет максимальный объем буфера передачи. Т.е. в режиме работы с DMA буфер передачи не может быть заполнен больше, чем на TBES+1 64 разрядных слов.	RW	7
19	-	Резерв	-	0
18:16	TLEV	Порог прерывания от буфера передачи: Прерывание формируется если число 64-х разрядных слов в буфере передачи меньше либо равно TLEV. В режиме передачи данных с использованием DMA определяет степень заполнения буфера передачи, при которой происходит запись в буфер очередной пачки данных	RW	0
15:11	-	Резерв	-	0
10	TXBUF	Результирующее прерывание MFBSP_TXBUF	R	0
9	TXBUF_D	Прерывание MFBSP_TXBUF без механизма автоматического сброса при чтении TSR	R	0
8	TXBUF_R	Прерывание MFBSP_TXBUF с механизмом автоматического сброса при чтении TSR	R	0
7	TRUN	Идёт передача: 0 – передатчик в состоянии ожидания 1 – идёт передача очередного слова	R	0
6	TERR	Ошибка передачи: 0 – передача проходила в штатном режиме 1 - было чтение из пустого буфера передачи (передача некорректных данных). Флаг сбрасывается записью 0 в 6-й разряд регистра TSR.	RW	0
5	TSBF	Буфер пересинхронизации в направлении передачи полон: 0 – буфер пересинхронизации в направлении передачи не полон 1 – буфер пересинхронизации в направлении передачи полон	R	0
4	TSBE	Буфер пересинхронизации в направлении передачи пуст: 0 – буфер пересинхронизации в направлении передачи не пуст 1 – буфер пересинхронизации в направлении передачи пуст	R	1
3	TBLL	Достигнут порог прерывания в буфере передачи: 1 – число 64-х разрядных слов в буфере передачи меньше либо равно TLEV 0 – число 64-х разрядных слов в буфере передачи больше TLEV	R	1
2	TBHF	Буфер передачи заполнен на половину или более: 1 – буфер передачи заполнен на половину или больше 0 – буфер передачи заполнен меньше чем на половину (в буфер передачи можно записать еще как минимум 4 слова)	R	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
1	TBF	Буфер передачи полон: 0 – буфер передачи не полон 1 – буфер передачи полон	R	0
0	TBE	Буфер передачи пуст: 0 – буфер передачи не пуст 1 – буфер передачи пуст	R	1

### 18.3.8 Регистр управления темпом приёма RCTR\_RATE (режим I2S)

Таблица 18.9 Назначение разрядов регистра RCTR\_RATE в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	RCS_RATE	Делитель частоты управляющего сигнала приёмника: Задаёт частоту управляющего сигнала приёмника, определяемую, как $RCLK/((RCS\_RATE+1)*2)$ , где RCLK – частота тактового сигнала приёмника RCS_RATE обязательно должно быть больше либо равно $(RWORDLEN+1)*(RWORDCNT+1)-1$	RW	0
15:12	-	В режиме I2S не используется	-	0
11:10	-	Резерв	-	0
9:0	RCLK_RATE	Делитель частоты приёмника:  В случае, если частота формируется самим приёмником, определяет частоту приёмника $RCLK = CLK/((RCLK\_RATE+1)*2)$ , где CLK – частота, подаваемая на порт со стороны системы.	RW	0

### 18.3.9 Регистр управления темпом передачи TCTR\_RATE (режим I2S)

Таблица 18.10 Назначение разрядов регистра TCTR\_RATE в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	TCS_RATE	Делитель частоты управляющего сигнала передатчика: Задаёт частоту управляющего сигнала передатчика, определяемую как $TCLK/((TCS\_RATE+1)*2)$ , где TCLK – частота тактового сигнала передатчика. TCS_RATE обязательно должно быть больше либо равно $(TWORDLEN+1)*(TWORDCNT+1)-1$	RW	0
15:12	-	В режиме I2S не используется	-	0
11:10	-	Резерв	-	0
9:0	TCLK_RATE	Делитель частоты передатчика:  В случае, если частота формируется самим передатчиком, определяет частоту передатчика $TCLK = CLK/((TCLK\_RATE+1)*2)$ , где CLK – частота, подаваемая на порт со стороны системы.	RW	0



### 18.3.10 Псевдорегистр TSTART (режим I2S)

Таблица 18.11 Назначение разрядов регистра TSTART в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:1	-	Резерв	-	0
0	TEN	Разрешение работы передатчика: 0 – передатчик выключен 1 – передатчик включен  Доступ к полю TEN регистра TCTR без изменения настроек TCTR	RW	0

### 18.3.11 Псевдорегистр RSTART (режим I2S)

Таблица 18.12 Назначение разрядов регистра RSTART в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:1	-	Резерв	-	0
0	REN	Разрешение работы передатчика: 0 – приемник выключен 1 – приемник включен  Доступ к полю REN регистра RCTR без изменения настроек RCTR	RW	0

### 18.3.12 Регистр аварийного управления портом EMERG\_MFBSP (режим I2S)

Таблица 18.13 Назначение разрядов регистра EMERG\_MFBSP в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:6	-	Резерв	-	0
5	RX_DBG	Программное управление признаком готовности приема данных из DMA в MFBSP: 0 – штатный режим работы. Признак готовности формируется MFBSP аппаратно; 1 – признак готовности установлен в 1. Используется для приведения DMA в исходное состояние, если: устройство подключенное к MFBSP передало в него меньший объем данных, по сравнению с тем, что указано в DMA; необходимо программно остановить прием данных в MFBSP	RW	0
4	TX_DBG	Программное управление признаком готовности передачи данных из MFBSP в DMA: 0 – штатный режим работы. Признак готовности формируется MFBSP аппаратно; 1 – признак готовности установлен в 1. Используется для приведения DMA в исходное состояние, если: устройство подключенное к MFBSP приняло из него меньший объем данных, по сравнению с тем, что указано в DMA; необходимо программно остановить передачу данных из MFBSP	RW	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
3	-	Резерв	-	0
2	RST_RXBUF	Сброс буфера приема последовательного порта и буфера пересинхронизации направления приема. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0
1	RST_TXBUF	Сброс буфера передачи последовательного порта и буфера пересинхронизации направления передачи. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0
0	RST_LPTBUF	Сброс буфера линкового порта и буфера пересинхронизации направления приёма. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0

### 18.3.13 Регистр маски прерываний от порта IMASK (режим I2S)

Таблица 18.14 Назначение разрядов регистра IMASK в режиме I2S

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:15	-	Резерв	-	0
14	RXBUF_R_EN	Разрешение автоматического сброса прерывания MFBSP_RXBUF 0 – прерывание не сбрасывается при чтении RSR 1 – прерывание сбрасывается при чтении RSR	RW	1
13	RX_LEV_IRQ_EN	Разрешение прерывания по уровню заполнения буфера приема 0 – прерывание MFBSP_RXBUF не будет устанавливаться при превышении порога RLEV 1 - прерывание MFBSP_RXBUF будет устанавливаться при превышении порога RLEV	RW	1
12	RX_ERR_IRQ_EN	Разрешение прерывания при переполнении буфера приема 0 – прерывание MFBSP_RXBUF не будет устанавливаться при переполнении буфера приема 1 - прерывание MFBSP_RXBUF будет устанавливаться при переполнении буфера приема	RW	1
11:7	-	Резерв	-	0
6	TXBUF_R_EN	Разрешение автоматического сброса прерывания MFBSP_TXBUF 0 – прерывание не сбрасывается при чтении TSR 1 – прерывание сбрасывается при чтении TSR	RW	1
5	TX_LEV_IRQ_EN	Разрешение прерывания по уровню заполнения буфера передачи 0 – прерывание MFBSP_TXBUF не будет устанавливаться, если число слов в буфере передачи меньше порога TLEV 1 - прерывание MFBSP_TXBUF будет устанавливаться, если число слов в буфере передачи меньше порога TLEV	RW	1
4	TX_ERR_IRQ_EN	Разрешение прерывания при переполнении буфера приема 0 – прерывание MFBSP_TXBUF не будет устанавливаться при чтении из пустого буфера передачи 1 - прерывание MFBSP_TXBUF будет устанавливаться при чтении из пустого буфера передачи	RW	1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
3:1	-	Резерв	-	0
0	LPT_IRQ_EN	Разрешение прерывания по запросу на обслуживание 0 – SRQ запрещено 1 – SRQ разрешено	RW	1

### 18.3.14 Структурная схема MFBSB для режима I2S

На Рисунок 18.7 представлена структурная схема MFBSB для режима I2S.

Включение режима I2S производится установкой бит LEN=0, SPI\_I2S\_EN=1, регистра CSR\_MFBSB и TMODE = 0 регистра TCTR для передатчика, RMODE = 0 регистра RCTR для приёмника.

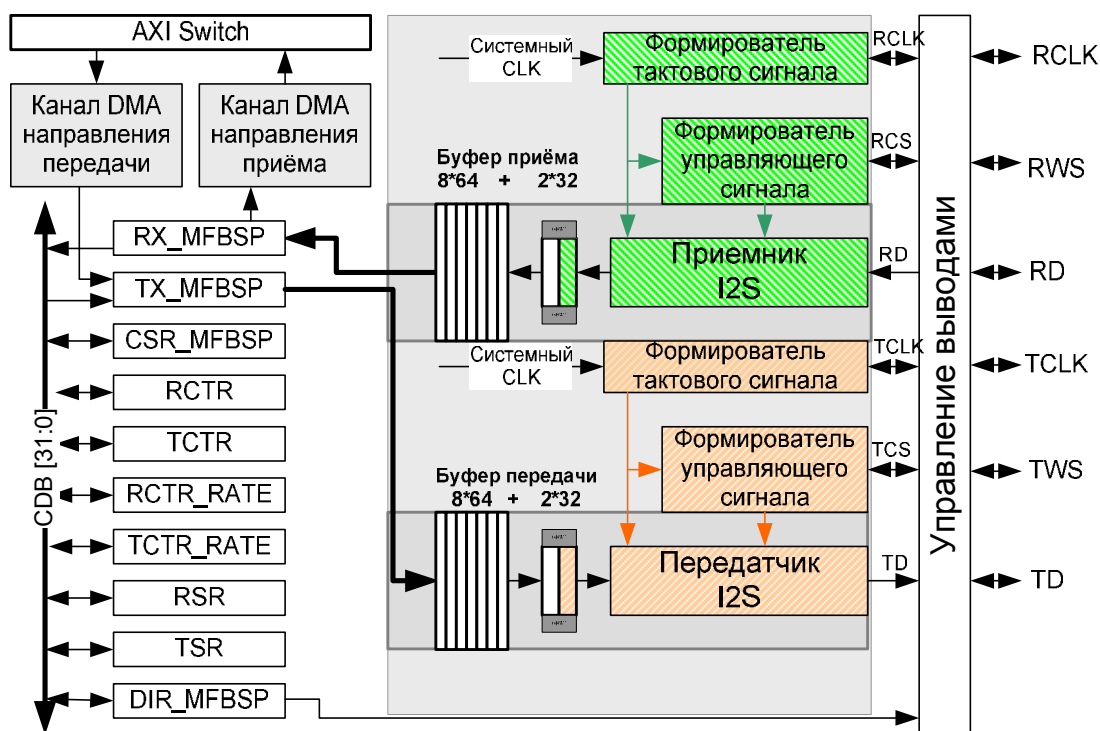


Рисунок 18.7. Структурная схема MFBSB для режима I2S

### 18.3.15 Варианты соединения порта с внешними устройствами

Программно управляя направлением выводов последовательного порта (см. описание регистра DIR\_MFBSP) можно организовать множество вариантов соединения схемы с внешними устройствами через MFBSP (Рисунок 18.8, Рисунок 18.9, Рисунок 18.10).

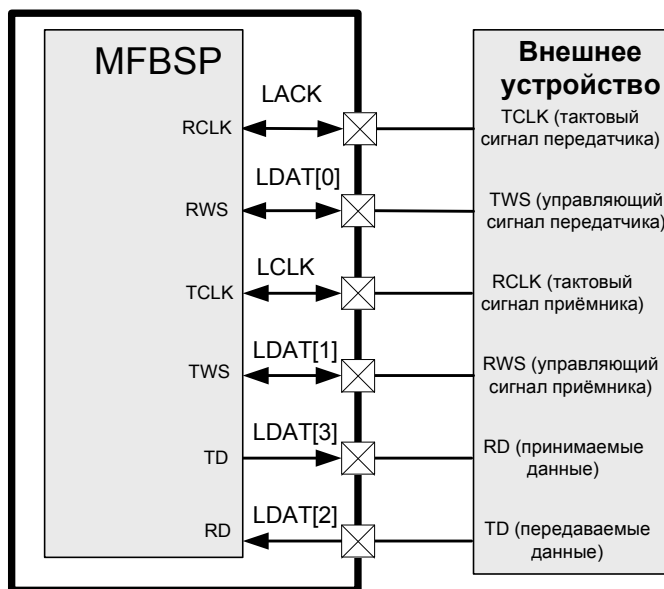


Рисунок 18.8. Соединение двух устройств по интерфейсу I2S в дуплексном режиме. Приёмник и передатчик независимые (задействовано 6 внешних выводов). Направление выводов TCLK, TWS, RCLK и RWS может быть произвольным в зависимости от требований внешнего устройства (режим №3 по Таблица 18.1)

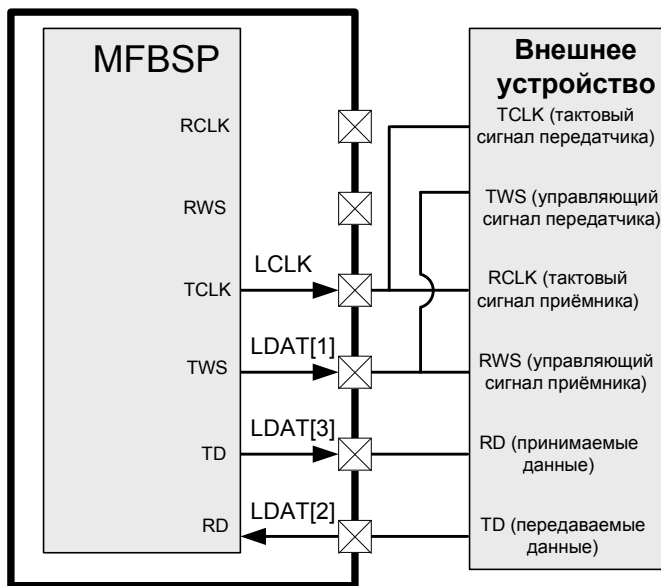


Рисунок 18.9. Соединение двух устройств по интерфейсу I2S в дуплексном режиме. Приёмник в зависимом от передатчика режиме (задействовано 4 внешних вывода) (режим №3 по Таблица 18.1)

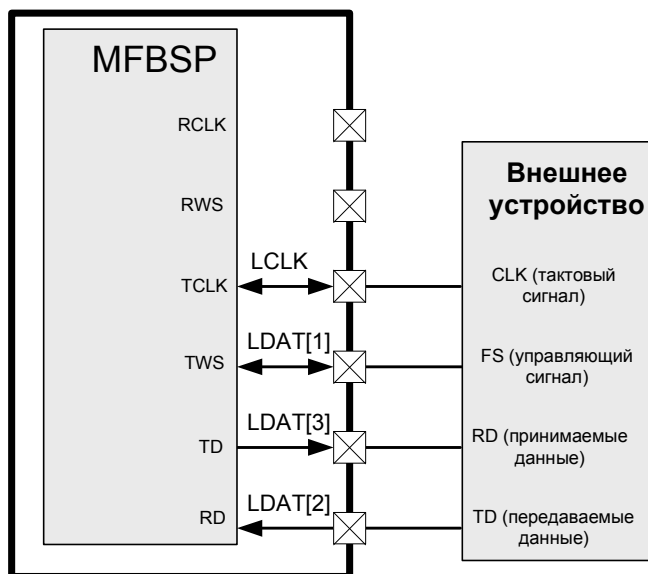


Рисунок 18.10 Соединение двух устройств по интерфейсу I2S в дуплексном режиме. Приёмник в зависимости от передатчика режиме (задействовано 4 внешних вывода). Как приёмником, так и передатчиком используются тактовый и управляющий сигналы с выводов TCLK и TWS. Направление выводов TCLK и TWS может быть произвольным в зависимости от требований внешнего устройства (режим №3 по Таблица 18.1)

### 18.3.16 Передача данных в режиме I2S

В режиме I2S возможна передача аудио данных с использованием сигнала выбора канала (бит (T/R)DSPMODE = 0). При этом программно задаётся полярность тактового сигнала, полярность управляющего сигнала и наличие задержки выдачи данных относительно фронта управляющего сигнала (см. описание регистров TCTR и RCTR). На Рисунок 18.11 представлены временные диаграммы для данного режима.

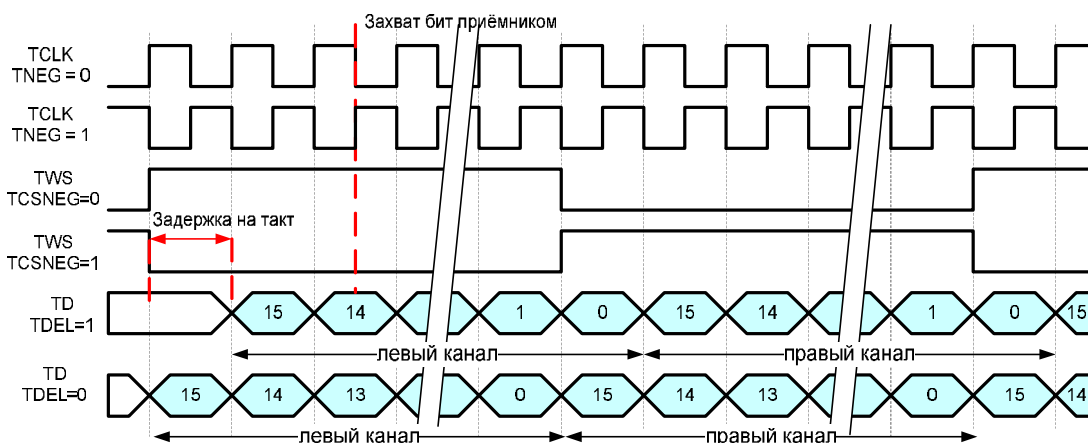
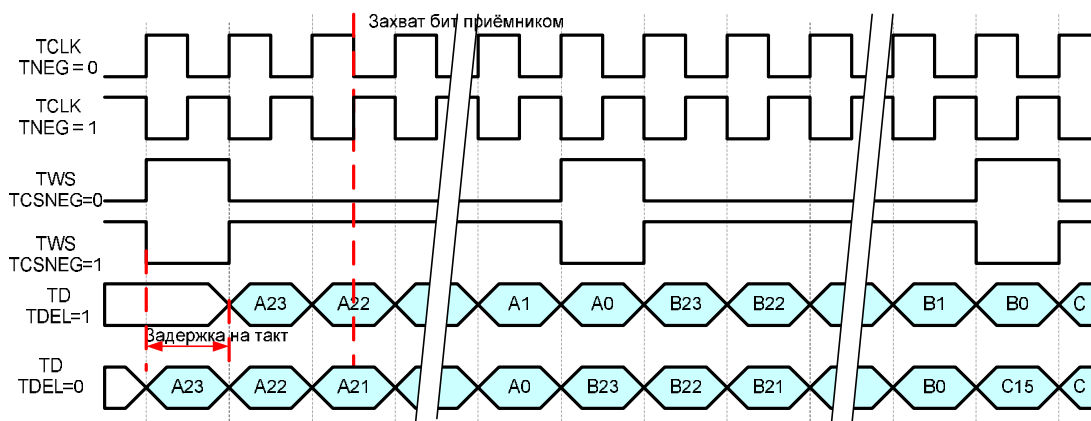


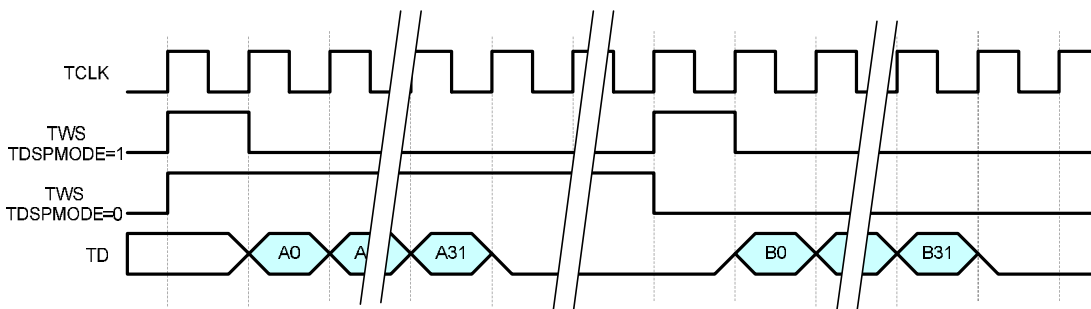
Рисунок 18.11 Передача в режиме I2S (формат I2S) TMODE = 0, TDSPMODE=0, TMBF = 1, TCS\_RATE = TWORDLEN = 15 диаграммы тактового сигнала TCLK представлены для различных значений TNEG, диаграммы управляющего сигнала TWS представлены для различных значений TCSNEG, диаграммы для последовательных данных представлены для различных значений TDEL

В режиме I2S (бит (T/R)MODE = 0) также возможна передача последовательных слов с использованием сигнала синхронизации фрейма (бит (T/R)DSPMODE = 1). При этом программно задаётся полярность тактового сигнала, полярность активного фронта управляющего сигнала и наличие задержки выдачи данных относительно фронта управляющего сигнала (Рисунок 18.12).



**Рисунок 18.12.** Передача в режиме I2S (формат DSP)  $TMODE = 0$ ,  $TDSPMODE=1$ ,  $TMBF = 1$ ,  $TCS\_RATE = TWORDLEN = 23$  диаграммы тактового сигнала TCLK представлены для различных значений TNEG, диаграммы управляющего сигнала TWS представлены для различных значений TCSNEG, диаграммы для последовательных данных представлены для различных значений TDEL

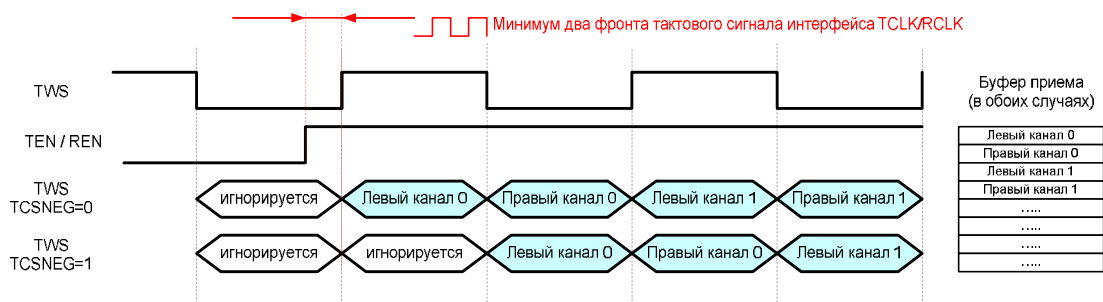
Если управляющий сигнал формируется логикой MFBS (вывод (T/R)WS – сконфигурирован как выход), то частота управляющего сигнала (либо частота импульсов синхронизации в формате DSP) может задаваться программно от  $ICLK/2$  до  $ICLK/(2*2^{16})$ , где ICLK – рабочая частота интерфейса TCLK для передатчика и RCLK для приемника (см. описание регистров TCTR\_RATE и RCTR\_RATE). Временные диаграммы для данного случая представлены на Рисунок 18.13.



**Рисунок 18.13.** Передача в режиме I2S  $TMODE = 0$ ,  $TMBF = 0$ ,  $TWORDLEN = 31$ ,  $TCS\_RATE > TWORDLEN$ ,  $TNEG = 0$ ,  $TCSNEG=0$ ,  $TDEL = 1$ . Диаграммы управляющего сигнала TWS представлены для различных значений TDSPMODE

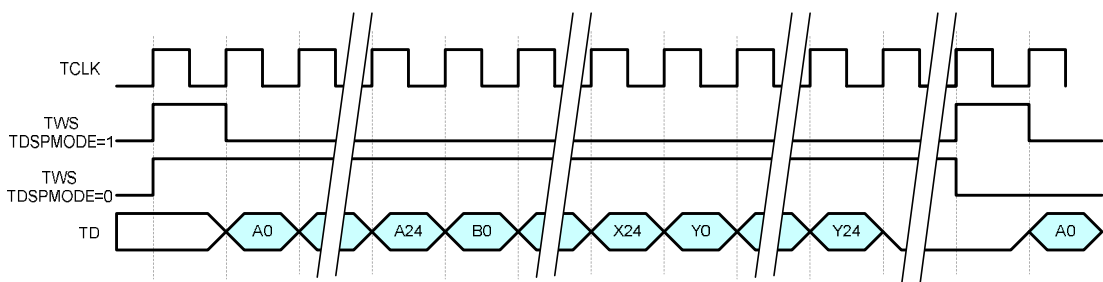
MFBS позволяет передавать от 1 до 64 слов в пределах одного фрейма (Рисунок 18.15). В этом случае с приходом сигнала синхронизации фрейма начинается передача первого слова, с передачей последнего бита первого слова из буфера передачи сразу считывается следующее слово и в следующем такте начинают передаваться биты очередного слова и так до тех пор, пока не будет передано число слов равно  $TWORDCNT+1$ . По окончании передачи последнего слова фрейма, порт ожидает очередного сигнала синхронизации фрейма. Сигнал синхронизации пришедший в момент, когда передача слов фрейма еще не закончилась игнорируется. Буфер передачи может вместить максимум 18 32-х разрядных слов, если в пределах фрейма передаётся больше 18 слов необходимо следить за тем, что бы буфер передачи был не пуст (при включенном канале DMA это происходит автоматически). Приёмник MFBS аналогичным образом может принимать от 1 до 64-х слов в пределах одного фрейма.

В режиме I2S, при  $(T/R)MODE = 0$ ,  $(T/R)DSPMODE = 0$  выполняется автоматическая синхронизация принимаемых и передаваемых данных таким образом, что первое слово переданное передатчиком будет передано в левый канал, а первое слово принятое приемником будет принято из левого канала (Рисунок 18.14).



**Рисунок 18.14.** Синхронизация передаваемых и принимаемых данных по каналам (левый/правый) в режиме I2S после включения приемника или передатчика для различных значений TCSNEG

При работе порта в режиме I2S ((T/R)MODE = 0), в случае если используется управляющий сигнал, формируемый внешним устройством (порт в режиме ведомого), то, как для передатчика, так и для приемника после первого включения порта (TEN/REN=1), перед первым фронтом сигнала выбора слова / фрейм селекта (TWS) необходима подача как минимум двух импульсов тактового сигнала, необходимых для целей синхронизации. В противном случае первый импульс управляющего сигнала может быть проигнорирован (передача начнется со следующего активного фронта управляющего сигнала).



**Рисунок 18.15.** Передача в режиме I2S TMODE = 0, TMBF = 0, TWORDLEN = 24, TWORDCNT=Y-1, TCS\_RATE+1>(TWORDLEN+1)\*(TWORDCNT+1), TNEG = 0, TCSNEG=0, TDEL = 1. Диаграммы управляющего сигнала TWS представлены для различных значений TDSPMODE

В режиме I2S (только в формате I2S (T/R)DSPMODE=0) предусмотрен режим паковщика / распаковщика. В этом режиме 32 разрядные слова из буфера передачи автоматически разбиваются на 2 16-ти разрядных слова и передаются по разным каналам. Соответственно для приёмника два принятых по разным каналам слова группируются в одно 32-х разрядное слово, которое записывается в буфер приёма. В данном режиме длина передаваемого или принимаемого слова может быть в пределах от 2 до 16 бит. Порядок выдачи разбитого слова и порядок сборки определяется битами TCSNEG, TSWAP, RCSNEG, RSWAP. Данный режим возможен только при передаче одного слова во фрейме (TWORDCNT=RWORDCNT=0).

Пример настроек для передачи по интерфейсу FSB (CMX981): TMODE = RMODE = 0, TDSPMODE = RDSPMODE = 1, TMBF = RMBF = 1, TNEG = RNEG= 1, TDEL = RDEL = 1, TCSNEG = RCSNEG = 0, TCS\_RATE >= TWORDLEN, RCS\_RATE >= RWORDLEN. Приемник при этом должен быть независим от передатчика, т.е. RCS\_CP = 0. Если шиной используется один тактовый сигнал для приема и передачи необходимо установить RCLK\_CP в 1, в этом случае приемником будет использоваться тактовый сигнал передатчика.

### 18.3.17 Формирование тактовых сигналов приёмника (RCLK) и передатчика (TCLK)

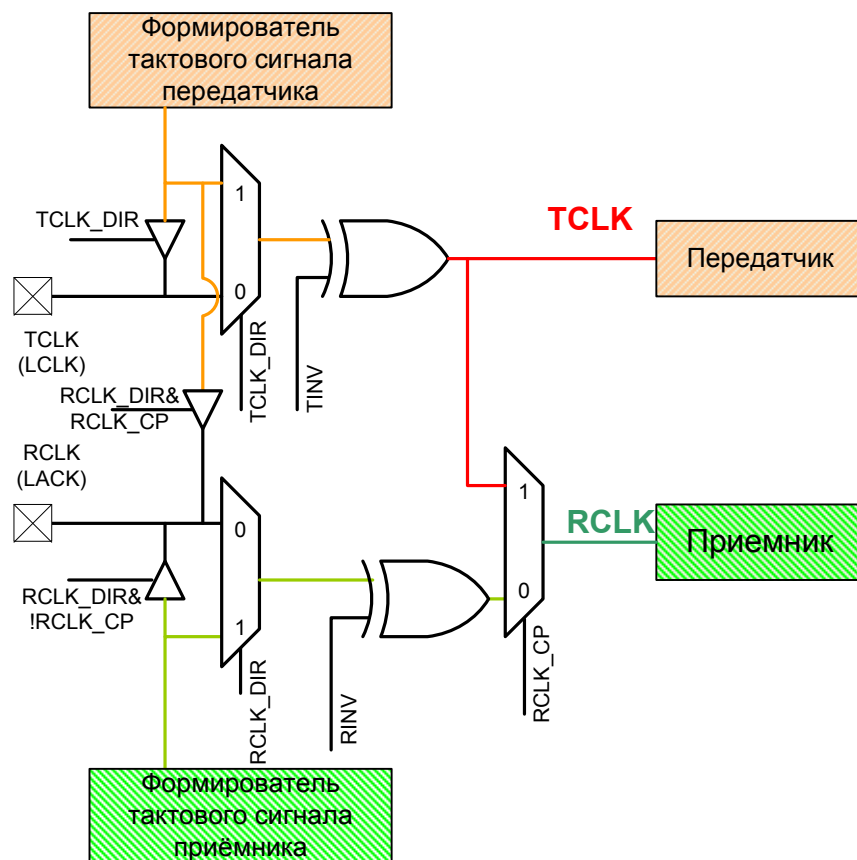


Рисунок 18.16. Схема формирования тактовых сигналов приёмника и передатчика в режиме I2S

На Рисунок 18.16 представлена схема формирования тактовых сигналов приёмника и передатчика в режиме I2S.

В зависимости от значения бита  $TCLK\_DIR$ , тактовый сигнал передатчика  $TCLK$  может как формироваться самим передатчиком, так приниматься с внешнего вывода. В зависимости от значений бит  $TMODE$ ,  $TNEG$  и  $TDEL$  тактовый сигнал либо передается передатчику без изменений, либо инвертируется.

В зависимости от значения бита  $RCLK\_DIR$ , тактовый сигнал приёмника  $RCLK$  может как формироваться самим приёмником, так приниматься с внешнего вывода. В зависимости от значений бит  $RMODE$ ,  $RNEG$  и  $RDEL$  тактовый сигнал либо передается приёмнику без изменений, либо инвертируется.

Если бит  $RCLK\_CP$  установлен в 1, то тактовый сигнал приёмника копирует тактовый сигнал передатчика. Для корректной работы устройства в этом случае настройки полярности тактового сигнала приёмника и передатчика должны совпадать ( $TNEG=RNEG$ ,  $TDEL=RDEL$ ).

При  $RCLK\_CP = 1$  тактовый сигнал передатчика передается на внешний вывод приёмника, только если передатчик сам формирует тактовый сигнал и вывод тактового сигнала приёмника сконфигурирован как выход ( $TCLK\_DIR=1$ ,  $RCLK\_DIR=1$ ).

Если биты  $RCLK\_CONT=1$  и  $RCLK\_DIR=1$  то  $RCLK$  формируется непрерывно, пока установлен бит  $REN$ . Если  $RCLK\_CONT=0$  и  $RCLK\_DIR=1$  то  $RCLK$  формируется только до мо-



мента заполнения буфера приёма. Если  $RCLK\_DIR=0$ , то  $RCLK$  принимается с внешнего вывода схемы.

Если биты  $TCLK\_CONT=1$  и  $TCLK\_DIR=1$  то  $TCLK$  формируется непрерывно, пока установлен бит  $TEN$ . Если  $TCLK\_CONT=0$  и  $TCLK\_DIR=1$  то  $TCLK$  формируется только в процессе передачи очередного слова. Если  $TCLK\_DIR=0$ , то  $TCLK$  принимается с внешнего вывода схемы.

### 18.3.18 Формирование управляющих сигналов приёмника и передатчика в режиме I2S

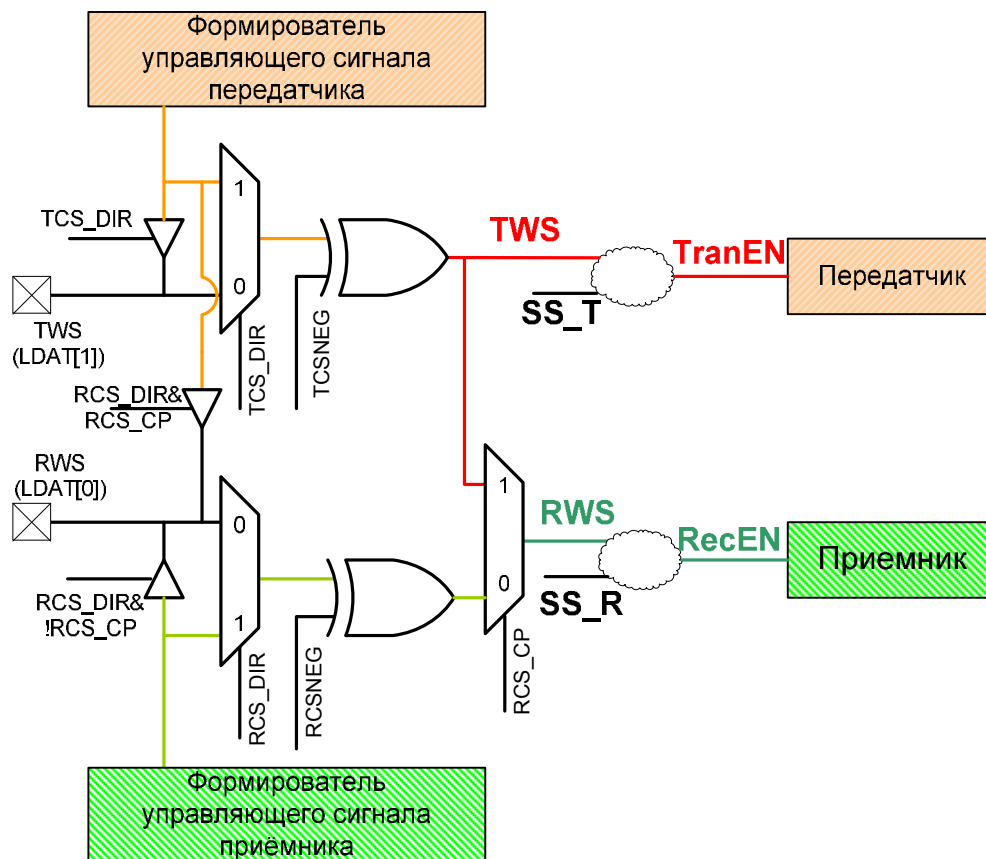


Рисунок 18.17. Схема формирования управляющих сигналов в режиме I2S

На Рисунок 18.17 представлена схема формирования управляющих сигналов в режиме I2S.

В зависимости от значения бита  $TCS\_DIR$ , задающего направление вывода  $TWS$ , управляющий сигнал передатчика  $TWS$  может как формироваться самим передатчиком, так приниматься с внешнего вывода. В зависимости от значения бита  $TCSNEG$  управляющий сигнал либо передается передатчику без изменений, либо инвертируется.

В зависимости от значения бита  $RCS\_DIR$ , задающего направление вывода  $RWS$ , управляющий сигнал приёмника  $RCLK$  может как формироваться самим приёмником, так приниматься с внешнего вывода. В зависимости от значения бита  $RCSNEG$  управляющий сигнал либо передается приёмнику без изменений, либо инвертируется.

Если бит  $RCS\_CP$  установлен в 1, то управляющий сигнал приёмника копирует управляющий сигнал передатчика. Для корректной работы устройства в этом случае настройки полярности управляющего сигнала приёмника и передатчика должны совпадать ( $TCSNEG=RCSNEG$ ).



При  $RCS\_CP = 1$  управляющий сигнал передатчика передаётся на внешний вывод приёмника, только если передатчик сам формирует управляющий сигнал и вывод управляющего сигнала приёмника сконфигурирован как выход ( $TCS\_DIR=1$ ,  $RCS\_DIR=1$ ).

Если направление вывода RWS задано как выход и  $RCS\_CONT=0$ , то управляющий сигнал RWS формируется до тех пор, пока не заполнится буфер приёма, если  $RCS\_CONT=1$  то, RWS формируется непрерывно, пока установлен бит REN. Если направление вывода задано как вход, управляющий сигнал RWS принимается от внешнего устройства. Если установлен бит  $RCS\_CP$ , RWS копирует TWS, независимо от направления вывода.

Если направление вывода TWS задано как выход и  $TCS\_CONT=0$ , то управляющий сигнал TWS формируется только во время передачи очередного слова, если  $TCS\_CONT=1$  TWS формируется непрерывно, пока установлен бит TEN. Если направление вывода задано как вход, управляющий сигнал TWS принимается от внешнего устройства.

### 18.3.19 Тракт передачи данных

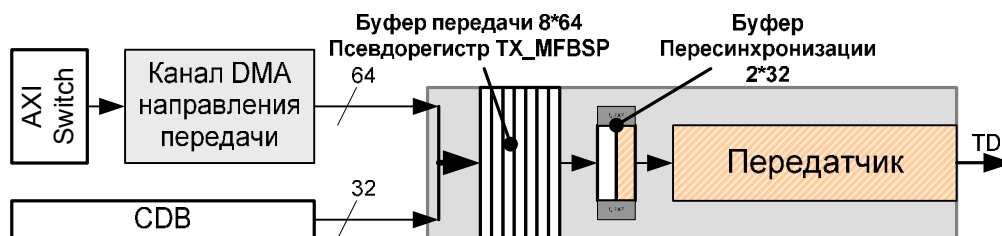


Рисунок 18.18. Тракт передачи данных для режима I2S

На Рисунок 18.18 представлен тракт передачи данных для режима I2S.

Что бы инициировать передачу данных по последовательному порту необходимо включить последовательный порт ( $SPI\_I2S\_EN=1$ ) и передатчик ( $TEN=1$ ), после чего либо начать производить запись передаваемых 32-х разрядных слов в буфер передачи по адресу псевдорегистра  $TX\_MFBSP$ , либо включить канал DMA в направления передачи для соответствующего порта (в этом случае обмен данными с портом будет вестись 64-х разрядными словами).

Данные записанные в буфер передачи автоматически перемещаются в буфер пересинхронизации направления передачи, если он не полон. Запись в буфер пересинхронизации направления передачи осуществляется на системной частоте  $CLK$ , чтение из буфера пересинхронизации осуществляется на частоте передатчика  $TCLK$ . Как только в буфере пересинхронизации оказалось хотя бы одно слово, передатчиком иницируется передача. Передатчиком производится последовательная выдача бит очередного 32-х разрядного слова до тех пор, пока число переданных бит не достигнет  $TWORDLEN+1$ , после чего производится считывание очередного слова из буфера пересинхронизации. По мере передачи слов в освобождающийся буфер пересинхронизации перемещается слово из буфера передачи. После выборки последнего слова из буфера передачи (буфер передачи пуст) в буфере пересинхронизации остаётся еще два слова. Фактическое окончание передачи можно идентифицировать по состоянию буфера пересинхронизации, либо считав бит  $TRUN$  регистра  $TSR$ .

Если управляющий сигнал формируется передатчиком, то при считывании последнего слова из буфера пересинхронизации передача останавливается. Передача продолжится только после того как в буфер пересинхронизации снова начнут поступать данные.

Если передатчик использует внешнюю частоту и внешний управляющий сигнал, в целях экономии мощности системная частота может быть установлена меньшей, чем внешняя частота передатчика, однако ее должно быть достаточно для того, что бы успеть переместить очередное слово в буфер пересинхронизации (за время передачи одного слова должно быть

хотя бы три импульса системной частоты CLK). Если внешний управляющий сигнал инициировал передачу слова при пустом буфере пересинхронизации устанавливается флаг ошибки передачи (TERR), в этом случае передаётся ошибочное слово. Если управляющий сигнал формируется самим передатчиком, системная частота может быть много меньше частоты передатчика, однако это скажется на скорости передачи данных.

Установка бита TERR в процессе передачи говорит о том, что порт произвел попытку чтения из пустого буфера передачи. Это значит, что передатчиком было передано некорректное слово.

В направлении передачи порт обладает буферизацией на 18 32-х разрядных слов. В случае передачи данных посредством DMA запись блоков данных в буфер передачи происходит до тех пор, пока буфер готов принять очередной блок, размер которого определяется битами WN, регистра CSR соответствующего канала DMA.

Степень заполнения буфера можно программно регулировать, используя биты TBES. В этом случае значение выражения  $TBES+1$  – задает эффективный размер буфера передачи. Передача очередной пачки происходит только в случае, если при записи этой пачки в буфер передачи число 64 разрядных слов в буфере передачи не превысит  $TBES+1$ . При попытке передать пачку со значением  $WN > TBES$ , значение WN автоматически корректируется (большая пачка передается в несколько этапов пачками меньшего размера).

Так, при  $WN=0$  и  $TBES=0$  очередное 64-х разрядное слово будет подкачиваться в буфер передачи только если он пуст. В этом случае по окончании работы DMA в момент прерывания от соответствующего канала передатчику останется передать ещё 4 32-х разрядных слова (два 32-х разрядных слова в буфере пересинхронизации и одно 64-х разрядное слово в буфере передачи).

Установка бита SPI\_I2S\_EN в 0 приведет к программному сбросу передатчика, и все данные находящиеся в буфере передачи будут утеряны.

### 18.3.20 Тракт приёма данных

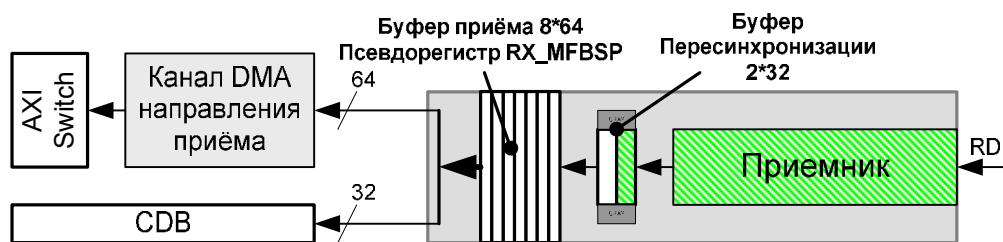


Рисунок 18.19. Тракт приёма данных в режиме I2S

На Рисунок 18.19 представлен тракт передачи данных для режима I2S.

Что бы перевести приёмник в режим готовности необходимо включить последовательный порт ( $SPI\_I2S\_EN=1$ ) и приёмник ( $REN=1$ ), после чего либо начать ожидание появления прочитанных данных в буфере приёма, либо включить канал DMA в направления приёма для соответствующего порта.

Приёмник принимает последовательные биты, поступающие с внешнего вывода до тех пор, пока число принятых бит не достигнет значения  $RWORDLEN+1$ . После этого принятое 32-х разрядное слово (если  $RWORDLEN < 31$  незадействованные биты обнуляются) перемещается в буфер пересинхронизации. Запись в буфер пересинхронизации направления приёма осуществляется на частоте приёмника RCLK, чтение из буфера пересинхронизации осуществляется на системной частоте CLK. Из буфера пересинхронизации принятое слово автоматиче-

ски перемещается в буфер приёма, если он не полон. Если в буфере приёма есть хотя бы одно 32-х разрядное слово, то принятые 32-х разрядные слова можно считывать, обращаясь по адресу псевдорегистра RX\_MFBSP. Принимать данные можно также включив соответствующий порту канал DMA направления приёма (в этом случае обмен данными с портом осуществляется 64-х разрядными словами).

Если приёмник использует внешнюю частоту, то в целях экономии мощности системная частота может быть установлена меньшей, чем внешняя частота приёмника, однако ее должно быть достаточно для того, что бы успеть переместить очередное слово из буфера пересинхронизации (за время приёма одного слова должно быть хотя бы три импульса системной частоты CLK). Если при заполненном буфере пересинхронизации приёмником был произведен приём очередного слова и инициирована попытка записи в буфер пересинхронизации устанавливается флаг ошибки приёма (RERR), а последнее принятое слово теряется.

Установка бита RERR в процессе передачи говорит о том, что порт произвел попытку записи в полный буфер приёма. Это значит, что принятое слово было потеряно.

В направлении приёма порт обладает буферизацией на 18 32-х разрядных слов. В случае приёма данных посредством DMA чтение блоков данных из буфера приёма происходит до тех пор, пока в буфере приёма достаточно слов для чтения очередного блока, размер которого определяется битами WN, регистра CSR соответствующего канала DMA. DMA обмена возможны только 64 разрядными словами, таким образом, если было принято нечетное количество 32-х разрядных слов, после окончания работы DMA необходимо прочитать оставшееся слово, обратившись к псевдорегистру RX\_MFBSP.

Установка бита SPI\_I2S\_EN в 0 приведет к программному сбросу приёмника и все данные находящиеся в буфере приёма будут утеряны.

### **18.3.21 Прерывания от последовательного порта**

Прерывание MFBSP\_RXBUF устанавливается, в случае если включен приемник (I2S\_SPI\_EN=1, REN = 1) и в буфер приёма записано количество слов большее, чем установлено уровнем прерывания RLEV, либо произошла ошибка приема (RERR = 1).

Прерывание MFBSP\_TXBUF устанавливается, в случае если включен передатчик (I2S\_SPI\_EN=1, REN = 1) и в буфере передачи осталось количество слов меньшее, либо равное чем установлено уровнем прерывания TLEV, либо произошла ошибка передачи (TERR = 1).

## **18.4 Работа MFBSB в режиме SPI**

### **18.4.1 Назначение последовательного порта в режиме SPI**

Режим SPI буферизированного последовательного порта предназначен для организации дуплексного обмена последовательными данными с внешними устройствами.

Порт в режиме SPI позволяет одновременно передавать и принимать последовательные данные. Приемник и передатчик контроллера могут настраиваются независимо, при этом возможен перевод приёмника в зависимое от передатчика состояние.

Поддерживается независимое задание направления каждого из выводов порта, осуществляемое установкой соответствующих бит регистра DIR\_MFBSP. Однако для режима SPI имеется ограничение: направление выводов тактового сигнала и сигнала выбора ведомого должно совпадать.

В режиме ведущего устройства к MFBSР параллельно может быть подключено до двух ведомых SPI устройств.

Формирование сигнала выбора ведомого возможно как в автоматическом так и в программном режиме. В автоматическом режиме после передачи каждой группы слов (число слов в группе может принимать значения от 1 до 64) сигнал выбора ведомого возвращается в высокое состояние. При программном управлении сигналами выбора ведомого данные сигналы изменяются посредством записи в контрольный регистр передатчика.

В данной реализации порта существует ограничение на выбор направления выводов в режиме SPI: тактовый и управляющий сигналы в режиме SPI должны быть либо оба заданы как вход, либо оба заданы как выход;

В данной реализации порта не предусмотрена возможность соединения нескольких микропроцессоров по цепочке с использованием SPI интерфейса. Микропроцессор может только управлять загрузкой последовательных данных в другие ведомые устройства, соединенные по цепочке.

В данной реализации порта в режиме ведомого устройства сигнал выбора ведомого предварительно пересинхронизируется на внутреннюю частоту порта, поэтому для устойчивой работы порта в режиме ведомого SPI устройства уровень сигнала SS, если необходима его установка в 1 между передачами, должен удерживаться как минимум два периода внутренней частоты CLK. Поэтому, если приемник работает в зависимом от передатчика режиме (RCS\_CP=1, RCLK\_CP=1), передатчик работает на максимальной частоте (TCLK\_RATE=0) и формирует сигнал SS в автоматическом режиме (SS\_DO=0, TCS\_DIR=1), необходимо установить значение TSS\_RATE $\geq$ 1 чтобы удерживать сигнал SS в высоком уровне как минимум два периода внутренней частоты CLK.

### 18.4.2 Регистр управления и состояния CSR\_MFBSP (режим SPI)

Регистр CSR\_MFBSP (Таблица 18.15) используется для включения режима последовательного порта и разрешения прерываний от MFBSP.

Таблица 18.15 Назначение разрядов регистра CSR\_MFBSP в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:15	-	Резерв	-	0
14:11	-	В режиме SPI не используется	-	0
10	-	Резерв	-	0
9	SPI_I2S_EN	Включение режима SPI/I2S: 0 – Работа в режиме LPORT 1 – Работа в режиме SPI/I2S	RW	0
8:5	-	В режиме I2S не используется	-	0
4:3	LSTAT	Состояние буфера: При LTRAN = 0 показывает состояние буфера приёма При LTRAN = 1 показывает состояние буфера передачи 00 – буфер пуст; 10 – буфер не пуст; 11 – буфер полон.	R	0
2	-	В режиме I2S не используется	-	0
1	LTRAN	Назначение бит LSTAT: 0 - LSTAT отображает состояние буфера приёма 1 - LSTAT отображает состояние буфера передачи	RW	0
0	LEN	В режиме SPI должен быть установлен в 0	RW	0

### 18.4.3 Регистр управления направлением выводов DIR\_MFBSP (режим SPI)

Регистр управления направлением выводов DIR\_MFBSP (Таблица 18.16) предназначен для индивидуальной настройки направления каждого вывода последовательного порта.

Таблица 18.16 Назначение разрядов регистра DIR\_MFBSP в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
9:6	-	В режиме SPI не используется	-	0
5	TD_DIR	Направление вывода MOSI: 0 – MOSI – вход (при RD_DIR = 1 последовательные данные принимаются со входа MOSI - эквивалент SDI) 1 – MOSI - выход (MOSI – является выходом для передачи последовательных данных и является эквивалентом SDO)	RW	0
4	RD_DIR	Направление вывода MISO: 0 – MISO – вход (последовательные данные принимаются со входа MISO - эквивалент SDI) 1 – MISO - выход (MISO – является выходом для передачи последовательных данных и является эквивалентом	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
		SDO)		
3	TCS_DIR	Направление вывода SS[0]: 0 – SS[0] – вход (управляющий сигнал для передатчика снимается с вывода SS[0]) 1 – SS[0] – выход, управляющий сигнал формируется передатчиком	RW	0
2	RCS_DIR	Направление вывода SS[1]: 0 – SS[1] – вход (управляющий сигнал для приёмника снимается с вывода SS[1]) 1 – SS[1] – выход, в этом случае на SS[1] в зависимости от состояния бита RCS_CP подаются управляющие сигналы, формируемые либо приёмником, либо передатчиком	RW	0
1	TCLK_DIR	Направление вывода TSCK: 0 – TSCK – вход (тактовый сигнал TSCK принимается от внешнего источника) 1 – TSCK – выход (тактовый сигнал TSCK формируется передатчиком)	RW	0
0	RCLK_DIR	Направление вывода RSCK: 0 – RSCK – вход (тактовый сигнал RSCK принимается от внешнего источника) 1 – RSCK – выход (тактовый сигнал RSCK формируется приёмником)	RW	0

**примечание:** при RD\_DIR = 0 и TD\_DIR = 0 данные снимаются с MISO, при RD\_DIR = 1 и TD\_DIR = 1 на MOSI и MISO выдаются одинаковые данные с передатчика.

#### 18.4.4 Регистр управления приёмником RCTR (режим SPI)

Таблица 18.17 Назначение разрядов регистра RCTR в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	-	Резерв	-	0
29	-	В режиме SPI не используется	-	0
28	-	В режиме SPI не используется	-	0
27	-	В режиме SPI не используется	-	0
26	RSIGN	Значение заполнителя: Если длина принимаемого слова меньше 32 при отключенном паковщике или меньше 16 при включенном паковщике, то неиспользуемые биты принятого слова заполняются При RSIGN = 0 нулями При RSIGN = 1 значением старшего разряда в принятом слове	RW	0
25	RPACK	В режиме SPI обязательно RPACK=0.	RW	0
24:20	RWORDLEN	Длина принимаемого слова: Число бит в принимаемом слове равно RWORDLEN + 1. RWORDLEN должно быть больше 0.	RW	5'b0
19	RMBF	Порядок передачи бит: 0 – младшим битом вперед 1 – старшим битом вперед	RW RW	1
18	-	В режиме SPI не используется	-	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
17:12	RWORDCNT	Число слов во фрейме: Определяет число принимаемых в течении одного фрейма слов. Число принимаемых слов равно RWORDCNT + 1. Число бит, принимаемых в пределах одного фрейма, равно (RWORDCNT + 1)*(RWORDLEN+1) Во время приёма фрейма состояние сигнала выбора ведомого не меняется.	RW	0
11	RDEL	Задержка начала приёма данных на пол такта: (Эквивалентно CPHA в спецификации Motorola). Задаёт фронт, по которому производится захват данных приёмником (фронт приёма). Ниже приведено соответствие полярности фронта приёма значениям бит RNEG, RDEL: RNEG = 0, RDEL = 0 – захват по переднему фронту RSCK RNEG = 0, RDEL = 1 – захват по заднему фронту RSCK RNEG = 1, RDEL = 0 – захват по заднему фронту RSCK RNEG = 1, RDEL = 1 – захват по переднему фронту RSCK	RW	0
10	RNEG	Полярность тактового сигнала приёмника: (эквивалентно CPOL в спецификации Motorola). Задаёт исходное состояние вывода RSCK и фронт, по которому производится захват данных приёмником (фронт приёма). Ниже приведено соответствие полярности фронта приёма значениям бит RNEG, RDEL: RNEG = 0, RDEL = 0 – захват по переднему фронту RSCK RNEG = 0, RDEL = 1 – захват по заднему фронту RSCK RNEG = 1, RDEL = 0 – захват по заднему фронту RSCK RNEG = 1, RDEL = 1 – захват по переднему фронту RSCK Исходное состояние RSCK = RNEG.	RW	0
9	-	В режиме SPI не используется	-	0
8:4	-	резерв	-	0
3	RCS_CP	Управление сигналом выбора ведомого приёмника: 0 – сигнал SS[1] принимается приёмником с внешнего вывода или формируется самим приёмником. 1 – сигнал SS[1] формируется передатчиком и является сигналом выбора ведомого устройства 1. Приёмник осуществляет приём данных синхронно с передатчиком. (в этом случае RCLK_CP должно быть так же в 1).	RW	0
2	RCLK_CP	Дублирование сигнала RSCK: 0 – RSCK формируется или принимается независимо от передатчика 1 – RSCK приёмника дублирует TSCK передатчика (в этом случае RCS_CP должно быть так же в 1).	RW	0
1	RMODE	Режим работы приёмника: 0 – режим I2S 1 – режим SPI	RW	0
0	REN	Разрешение работы приёмника: 0 – приемник выключен 1 – приемник включен	RW	0

### 18.4.5 Регистр управления передатчиком TCTR (режим SPI)

Таблица 18.18 Назначение разрядов регистра TCTR в режиме SPI

Номер разряда	Условное обозначение	назначение	Доступ	Исходное состояние
31	SS[1]	биты управления шиной Slave Select:	RW	0
30	SS[0]	Позволяют выбрать одно из двух подключенных ведомых устройств. При SS_DO = 0 установка соответствующего бита SS в 1 означает выбор ведомого устройства, с которым будет производиться обмен данными При SS_DO = 1 значения бит SS передаются на выходы SS напрямую	RW	0
29	-	В режиме SPI не используется	-	0
28	-	В режиме SPI не используется	-	0
27	-	В режиме SPI не используется	-	0
26	-	Резерв	-	0
25	TPACK	В режиме SPI обязательно TPACK=0.	RW	0
24:20	TWORDLEN	Длина передаваемого слова: Число бит в передаваемом слове равно TWORDLEN + 1. TWORDLEN должно быть больше 0.	RW	5'b0
19	TMBF	Порядок передачи бит: 0 – младшим битом вперед 1 – старшим битом вперед	RW	1
18	-	В режиме SPI не используется	-	0
17:12	TWORDCNT	Число слов во фрейме: Определяет число передаваемых в течении одного фрейма слов. Число передаваемых слов равно TWORDCNT + 1. Число бит, передаваемых в пределах одного фрейма, равно (TWORDCNT + 1)*(TWORDLEN+1) Во время передачи фрейма состояние сигнала выбора ведомого не меняется.	RW	0
11	TDEL	Задержка начала передачи данных на пол такта: (Эквивалентно CPHA в спецификации Motorola). Задаёт фронт, по которому производится выдача данных передатчиком (фронт выдачи). Ниже приведено соответствие полярности фронта выдачи значениям бит TNEG, TDEL: TNEG = 0, TDEL = 0 – выдача по заднему фронту TSCK TNEG = 0, TDEL = 1 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 0 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 1 – выдача по заднему фронту TSCK	RW	0
10	TNEG	Полярность тактового сигнала передатчика: (эквивалентно CPOL в спецификации Motorola). Задаёт исходное состояние вывода TSCK и фронт, по которому производится выдача данных передатчиком (фронт выдачи). Ниже приведено соответствие полярности фронта выдачи значениям бит TNEG, TDEL: TNEG = 0, TDEL = 0 – выдача по заднему фронту TSCK TNEG = 0, TDEL = 1 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 0 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 1 – выдача по заднему фронту TSCK Исходное состояние TSCK = TNEG.	RW	0
9	-	В режиме SPI не используется	-	0



Номер разряда	Условное обозначение	назначение	Доступ	Исходное состояние
8:4	-	резерв	-	0
3	SS_DO	управление выводами SS: 0 – управление выводами SS производится в автоматическом режиме. С началом передачи вывод SS, для которого соответствующий бит SS, регистра TCRT установлен в 1 переводится в низкое состояние, с окончанием передачи вывод SS переводится в высокое состояние. Если соответствующий выводу бит SS установлен в 0 вывод SS всегда находится в высоком состоянии. 1 – значения бит SS напрямую передаются на внешние выводы. В этом случае необходимо программное управление шиной SS в процессе передачи	RW	0
2	-	В режиме SPI не используется	-	0
1	TMODE	Режим работы передатчика: 0 – режим I2S 1 – режим SPI	RW	0
0	TEN	Разрешение работы передатчика: 0 – приемник выключен 1 – приемник включен	RW	0

#### 18.4.6 Регистр состояния приёмника RSR (режим SPI)

Таблица 18.19 Назначение разрядов регистра RSR в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	резерв	-	0
27:24	RB_DIFF	Количество принятых 64-разрядных слов в буфере приёма (max 8).	R	0
23:19	-	Резерв	-	0
18:16	RLEV	Порог прерывания от буфера приёма: Прерывание формируется если число принятых 64-х разрядных слов больше RLEV	RW	7
15:11	-	Резерв	-	0
10	RXBUF	Результирующее прерывание MFBSP_RXBUF	R	0
9	RXBUF_D	Прерывание MFBSP_RXBUF без механизма автоматического сброса при чтении RSR	R	0
8	RXBUF_R	Прерывание MFBSP_RXBUF с механизмом автоматического сброса при чтении RSR	R	0
7	RRUN	Идёт приём: 0 – приёмник в состоянии ожидания 1 – идёт приём очередного слова	R	0
6	RERR	Ошибка передачи: 0 – приём проходил в штатном режиме 1 - была запись в полный буфер приёма (потеря данных). Флаг сбрасывается записью 0 в 6-й разряд регистра RSR.	RW	0
5	RSBF	Буфер пересинхронизации в направлении приёма полон: 0 – буфер пересинхронизации в направлении приёма не полон 1 – буфер пересинхронизации в направлении приёма полон	R	0
4	RSBE	Буфер пересинхронизации в направлении приёма пуст: 0 – буфер пересинхронизации в направлении приёма не пуст 1 – буфер пересинхронизации в направлении приёма пуст	R	1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
3	RBHL	Достигнут порог прерывания в буфере приёма: 1 – число 64-х разрядных слов в буфере приёма больше чем задано в RLEV 0 – число 64-х разрядных слов в буфере приёма меньше либо равно RLEV	R	0
2	RBHF	Буфер приёма полон на половину или более: 1 – буфер приёма заполнен на половину или больше (из буфера приёма можно считать как минимум 4 слова) 0 – буфер приёма заполнен меньше чем на половину	R	0
1	RBF	Буфер приёма полон: 0 – буфер приёма не полон 1 – буфер приёма полон	R	0
0	RBE	Буфер приёма пуст: 0 – буфер приёма не пуст 1 – буфер приёма пуст	R	1

### 18.4.7 Регистр состояния передатчика TSR (режим SPI)

Таблица 18.20 Назначение разрядов регистра TSR в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	резерв	-	0
27:24	TB_DIFF	Количество свободных 64-разрядных позиций в буфере передачи (в буфер передачи можно записать еще TB_DIFF 64-разрядных слов).	R	8
23	-	Резерв	-	0
22:20	TBES	Эффективный размер буфера передачи Актуален только для режима работы с DMA. Значение TBES+1 – определяет максимальный объем буфера передачи. Т.е. в режиме работы с DMA буфер передачи не может быть заполнен больше, чем на TBES+1 64 разрядных слов.	RW	7
19	-	Резерв	-	0
18:16	TLEV	Порог прерывания от буфера передачи: Прерывание формируется если число 64-х разрядных слов в буфере передачи меньше либо равно TLEV. В режиме передачи данных с использованием DMA определяет степень заполнения буфера передачи, при которой происходит запись в буфер очередной пачки данных	RW	0
15:11	-	Резерв	-	0
10	TXBUF	Результирующее прерывание MFBSP_TXBUF	R	0
9	TXBUF_D	Прерывание MFBSP_TXBUF без механизма автоматического сброса при чтении TSR	R	0
8	TXBUF_R	Прерывание MFBSP_TXBUF с механизмом автоматического сброса при чтении TSR	R	0
7	TRUN	Идёт передача: 0 – передатчик в состоянии ожидания 1 – идёт передача очередного слова	R	0
6	TERR	Ошибка передачи: 0 – передача проходила в штатном режиме 1 - было чтение из пустого буфера передачи (передача некорректных данных). Флаг сбрасывается записью 0 в 6-й разряд регистра TSR.	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
5	TSBF	Буфер пересинхронизации в направлении передачи полон: 0 – буфер пересинхронизации в направлении передачи не полон 1 – буфер пересинхронизации в направлении передачи полон	R	0
4	TSBE	Буфер пересинхронизации в направлении передачи пуст: 0 – буфер пересинхронизации в направлении передачи не пуст 1 – буфер пересинхронизации в направлении передачи пуст	R	1
3	TBLL	Достигнут порог прерывания в буфере передачи: 1 – число 64-х разрядных слов в буфере передачи меньше либо равно TLEV 0 – число 64-х разрядных слов в буфере передачи больше TLEV	R	1
2	TBFH	Буфер передачи заполнен на половину или более: 1 – буфер передачи заполнен на половину или больше 0 – буфер передачи заполнен меньше чем на половину (в буфер передачи можно записать еще как минимум 4 слова)	R	0
1	TBF	Буфер передачи полон: 0 – буфер передачи не полон 1 – буфер передачи полон	R	0
0	TBE	Буфер передачи пуст: 0 – буфер передачи не пуст 1 – буфер передачи пуст	R	1

#### 18.4.8 Регистр управления темпом приёма RCTR\_RATE (режим SPI)

Таблица 18.21 Назначение разрядов регистра RCTR\_RATE в режиме SPI

Номер разряда	Условное обозначение	назначение	Доступ	Исходное состояние
31:16	-	В режиме SPI не используется	-	0
15:12	RSS_RATE	Если сигнал SS формируется приёмником, то задает время удержания сигнала SS в высоком уровне между передачами слов. Время удержания SS определяется как $TRCLK/2*(RSS\_RATE+1)$ , где TRCLK период тактового сигнала RCLK	RW	0
11:10	-	Резерв	-	0
9:0	RCLK_RATE	Делитель частоты приёмника:  В случае, если частота формируется самим приёмником, определяет частоту приёмника $RSCK = CLK/((RCLK\_RATE+1)*2)$ , где CLK – частота, подаваемая на порт со стороны системы.	RW	0

#### 18.4.9 Регистр управления темпом передачи TCTR\_RATE (режим SPI)

Таблица 18.22 Назначение разрядов регистра TCTR\_RATE в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:16	-	В режиме SPI не используется	-	0
15:12	TSS_RATE	Если сигнал SS формируется передатчиком, то задает время удержания сигнала SS в высоком уровне между передачами слов. Время удержания SS определяется как $TTCLK/2*(TSS\_RATE+1)$ , где TTCLK период тактового сигнала TCLK	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
11:10	-	Резерв	-	0
9:0	TCLK_RATE	Делитель частоты передатчика: В случае, если частота формируется самим передатчиком, определяет частоту передатчика $TCLK = CLK / ((TCLK\_RATE + 1) * 2)$ , где CLK – частота, подаваемая на порт со стороны системы.	RW	0

#### 18.4.10 Псевдорегистр TSTART (режим SPI)

Таблица 18.23 Назначение разрядов регистра TSTART в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:1	-	Резерв	-	0
0	TEN	Разрешение работы передатчика: 0 – передатчик выключен 1 – передатчик включен Доступ к полю TEN регистра TCTR без изменения настроек TCTR	RW	0

#### 18.4.11 Псевдорегистр RSTART (режим SPI)

Таблица 18.24 Назначение разрядов регистра RSTART в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:1	-	Резерв	-	0
0	REN	Разрешение работы передатчика: 0 – приемник выключен 1 – приемник включен Доступ к полю REN регистра RCTR без изменения настроек RCTR	RW	0

#### 18.4.12 Регистр аварийного управления портом EMERG\_MFBSP (режим SPI)

Таблица 18.25 Назначение разрядов регистра EMERG\_MFBSP в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:6	-	Резерв	-	0
5	RX_DBG	Программное управление признаком готовности приема данных из DMA в MFBSP: 0 – штатный режим работы. Признак готовности формируется MFBSP аппаратно; 1 – признак готовности установлен в 1. Используется для приведения DMA в исходное состояние, если: устройство подключенное к MFBSP передало в него меньший объем данных, по сравнению с тем, что указано в DMA; необходимо программно остановить прием данных в MFBSP	RW	0

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
4	TX_DBG	Программное управление признаком готовности передачи данных из MFBSP в DMA: 0 – штатный режим работы. Признак готовности формируется MFBSP аппаратно; 1 – признак готовности установлен в 1. Используется для приведения DMA в исходное состояние, если: устройство подключенное к MFBSP приняло из него меньший объем данных, по сравнению с тем, что указано в DMA; необходимо программно остановить передачу данных из MFBSP	RW	0
3	-	Резерв	-	0
2	RST_RXBUF	Сброс буфера приема последовательного порта и буфера пересинхронизации направления приема. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0
1	RST_TXBUF	Сброс буфера передачи последовательного порта и буфера пересинхронизации направления передачи. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0
0	RST_LPTBUF	Сброс буфера линкового порта и буфера пересинхронизации направления приёма. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0

### 18.4.13 Регистр маски прерываний от порта IMASK (режим SPI)

Таблица 18.26 Назначение разрядов регистра IMASK в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:15	-	Резерв	-	0
14	RXBUF_R_EN	Разрешение автоматического сброса прерывания MFBSP_RXBUF 0 – прерывание не сбрасывается при чтении RSR 1 – прерывание сбрасывается при чтении RSR	RW	1
13	RX_LEV_IRQ_EN	Разрешение прерывания по уровню заполнения буфера приема 0 – прерывание MFBSP_RXBUF не будет устанавливаться при превышении порога RLEV 1 - прерывание MFBSP_RXBUF будет устанавливаться при превышении порога RLEV	RW	1
12	RX_ERR_IRQ_EN	Разрешение прерывания при переполнении буфера приема 0 – прерывание MFBSP_RXBUF не будет устанавливаться при переполнении буфера приема 1 - прерывание MFBSP_RXBUF будет устанавливаться при переполнении буфера приема	RW	1
11:7	-	Резерв	-	0
6	TXBUF_R_EN	Разрешение автоматического сброса прерывания MFBSP_TXBUF 0 – прерывание не сбрасывается при чтении TSR 1 – прерывание сбрасывается при чтении TSR	RW	1

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
5	TX_LEV_IRQ_EN	Разрешение прерывания по уровню заполнения буфера передачи 0 – прерывание MFBSP_TXBUF не будет устанавливаться, если число слов в буфере передачи меньше порога TLEV 1 - прерывание MFBSP_TXBUF будет устанавливаться, если число слов в буфере передачи меньше порога TLEV	RW	1
4	TX_ERR_IRQ_EN	Разрешение прерывания при переполнении буфера приема 0 – прерывание MFBSP_TXBUF не будет устанавливаться при чтении из пустого буфера передачи 1 - прерывание MFBSP_TXBUF будет устанавливаться при чтении из пустого буфера передачи	RW	1
3:1	-	Резерв	-	0
0	LPT_IRQ_EN	Разрешение прерывания по запросу на обслуживание 0 – SRQ запрещено 1 – SRQ разрешено	RW	1

#### 18.4.14 Структурная схема MFBSP для режима SPI

На Рисунок 18.20 представлена структурная схема MFBSP для режима SPI.

Включение режима SPI производится установкой бит LEN=0, SPI\_I2S\_EN=1, TMODE = 1 (для передатчика), RMODE = 1 (для приёмника).

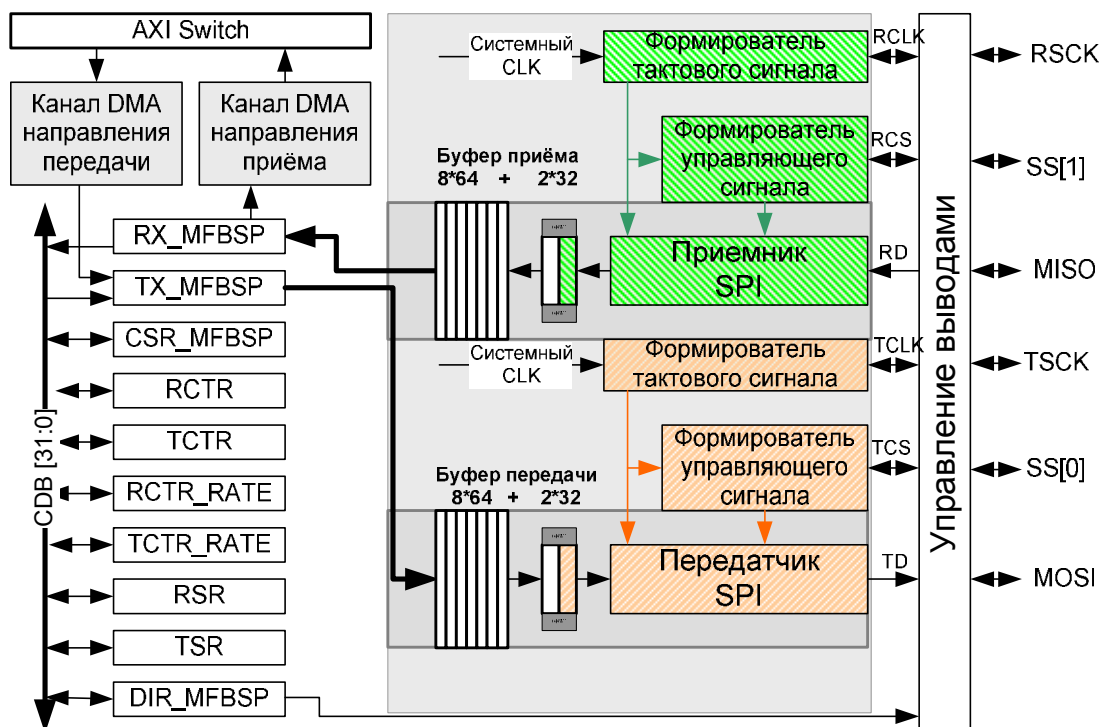


Рисунок 18.20 Структурная схема MFBSP для режима SPI

### 18.4.15 Варианты соединения порта с внешними устройствами

Программно управляя направлением выводов последовательного порта (см. описание регистра DIR\_MFBSP) можно организовать множество вариантов соединения схемы с внешними устройствами через MFBSP (Рисунок 18.21, Рисунок 18.22, Рисунок 18.23).

MFBSP позволяет подключить два ведомых SPI устройства. Выбор ведомого устройства с которым будет производиться обмен осуществляется битами SS, регистра TCTR. Если настройки двух устройств совпадают (для обоих ведомых значения TNEG, TDEL, TWORDLEN одинаковы) тогда перед для смены ведомого устройства достаточно изменить значение бит SS. Если настройки двух ведомых различаются, то для смены ведомого необходимо сначала выключить MFBSP ( $spi\_i2s\_en=0, len=0$ ), и только после этого записать новые настройки в регистры TCTR и RCTR.

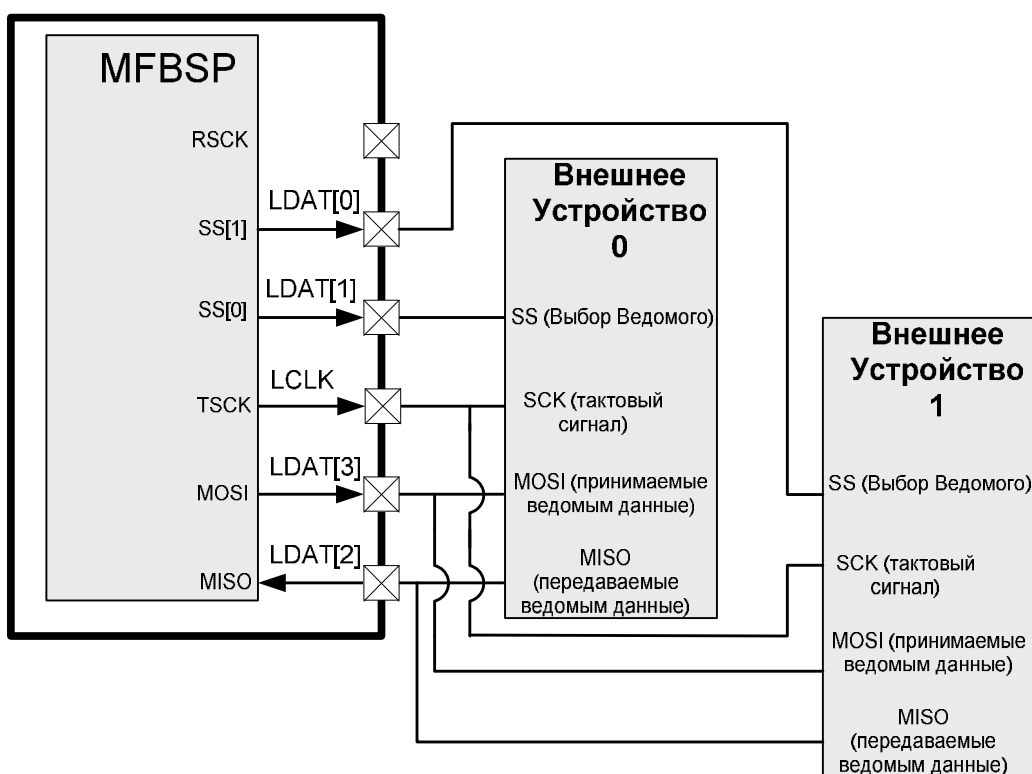


Рисунок 18.21. Подключение к MFBSP двух ведомых устройств по интерфейсу SPI. Приёмник в зависимом от передатчика режиме (режим №6 по Таблица 18.1)

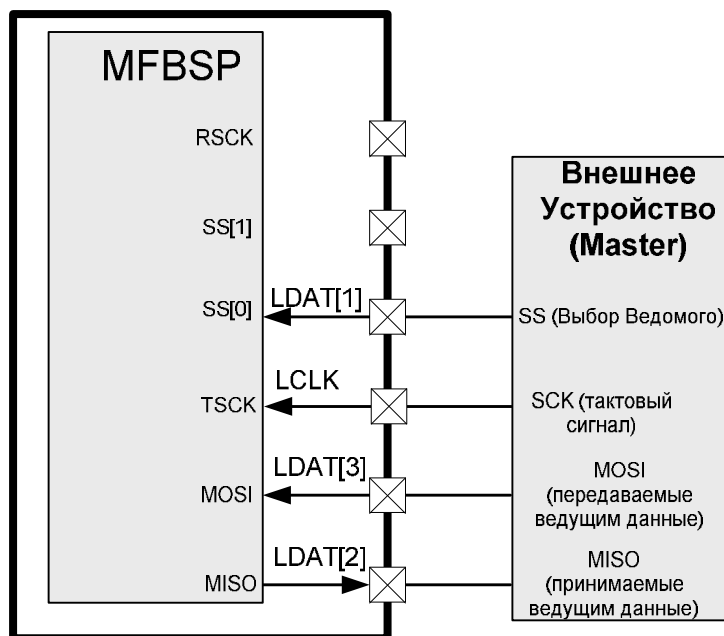


Рисунок 18.22. Подключение MFBSP по интерфейсу SPI к внешнему ведущему устройству. Приёмник в зависимом от передатчика режиме (режим №6 по Таблица 18.1)

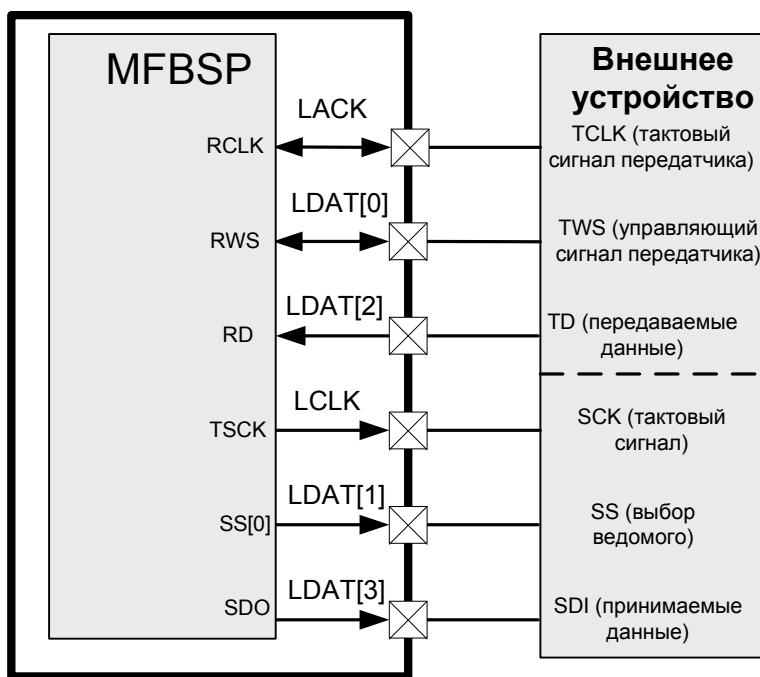
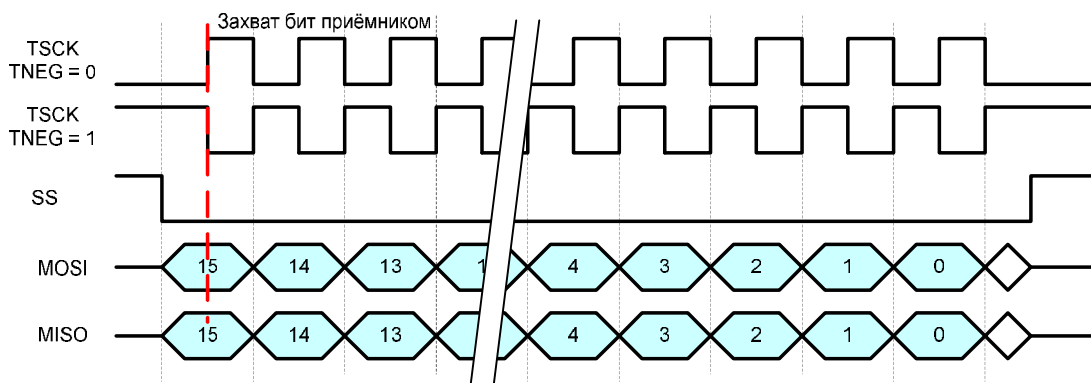


Рисунок 18.23. Организация передачи управляющих данных по интерфейсу SPI и приёма аудиоданных по интерфейсу I2S ((режим №5 по Таблица 18.1))

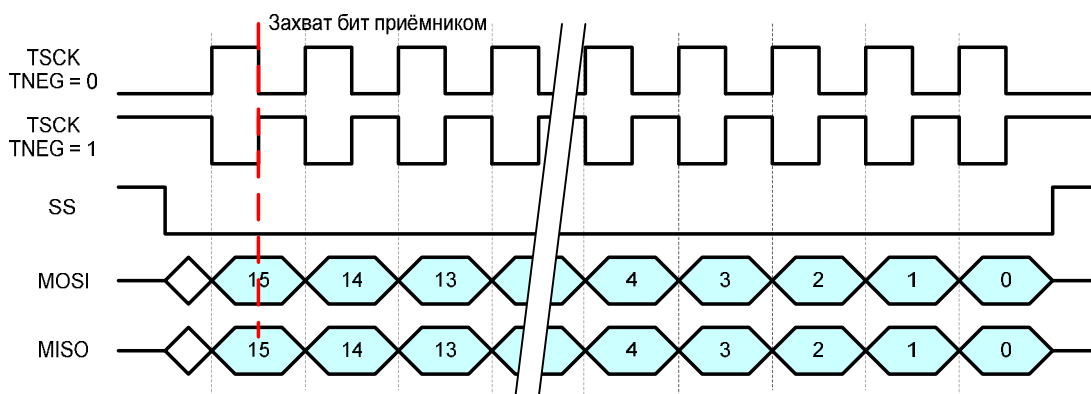
#### 18.4.16 Передача данных в режиме SPI

В режиме SPI возможна передача данных при четырёх сочетаниях бит TDEL и TNEG (Рисунок 18.24, Рисунок 18.25). При этом TNEG – задает начальное состояние вывода TCLK и полярность фронта, по которому производится чтение. TDEL задает смещение передаваемых данных на пол фазы. Значения RNEG и RDEL приёмника должны соответствовать TNEG и TDEL передатчика. После аппаратного сброса SS\_DO=0, в этом случае управление сигналом выбора ведомого производится в автоматическом режиме.



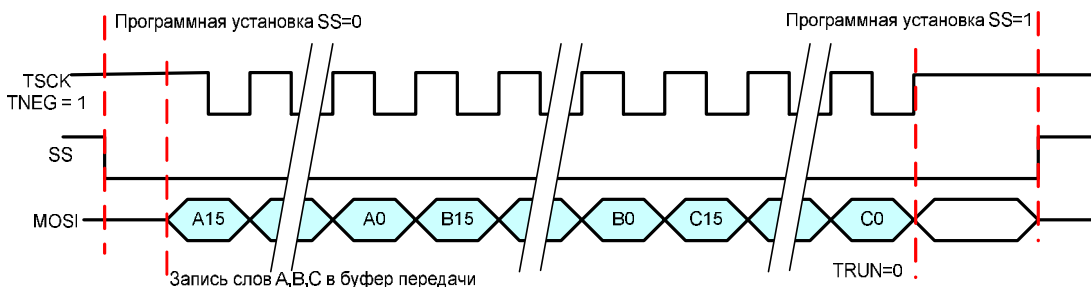


**Рисунок 18.24** Передача одного слова в режиме SPI с автоматической генерацией управляющего сигнала  $TMODE = 1$ ,  $TMBF = 1$ ,  $TDEL = 0$ ,  $SS\_DO = 0$ . Диаграммы тактового сигнала TSKC представлены для различных значений TNEG



**Рисунок 18.25** Передача одного слова в режиме SPI с автоматической генерацией управляющего сигнала  $TMODE = 1$ ,  $TMBF = 1$ ,  $TDEL = 1$ ,  $SS\_DO = 0$ . Диаграммы тактового сигнала TSKC представлены для различных значений TNEG

Что бы передать несколько слов без изменения уровня на внешнем выводе SS можно использовать программное управление внешним выводом SS, в этом случае  $SS\_DO$  необходимо установить в 1, программно установить вывод SS в 0, записать передаваемые данные в буфер передачи (или включить канал DMA на передачу), дождаться фактического окончания передачи (бит TRUN регистра TSR сбрасывается в 0), после чего программно установить вывод SS в 1 (Рисунок 18.26).



**Рисунок 18.26.** Передача трёх слов в режиме SPI с программным управлением сигналом SS,  $TMODE = 1$ ,  $TMBF = 1$ ,  $TDEL = 0$ ,  $TNEG = 0$ ,  $SS\_DO = 1$

В режиме SPI также имеется возможность программно регулировать количество слов, которое будет передано без изменения уровня сигнала SS (Рисунок 18.27). Количество слов может быть задано в пределах от 1 до 64 и определяются битом TWORDCNT. Буфер передачи

может вместить максимум 18 32-х разрядных слов, если в пределах фрейма передаётся больше 18 слов необходимо следить за тем, что бы буфер передачи был не пуст (при включенном канале DMA это происходит автоматически).

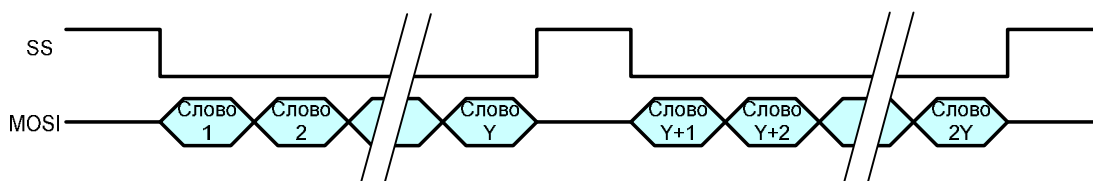


Рисунок 18.27. Передача в режиме SPI, TWORDCNT=Y-1

В режиме ведомого устройства сигнал выбора ведомого предварительно пересинхронизируется на внутреннюю частоту порта, поэтому для устойчивой работы порта в режиме ведомого SPI устройства уровень сигнала SS, если необходима его установка в 1 между передачами, должен удерживаться как минимум два периода внутренней частоты CLK.

Непосредственно к тактовому сигналу TSCCK данное ограничение не применяется, т.е. частота TSCCK может быть больше CLK.

Когда MFBSBP работает в режиме ведущего SPI устройства, время удержания сигнала SS при автоматическом формировании данного сигнала может регулироваться программно. В этом случае время между последним фронтом тактового сигнала для последней пересылки и установкой сигнала SS в 1 равно времени между установкой и сбросом сигнала SS и равно времени между сбросом сигнала SS первым фронтом тактового сигнала для новой пересылки. Это время определяется как  $TSS = (TSS\_RATE + 1) * TTCLK / 2$ , где TTCLK – период тактового сигнала, генерируемого портом для последовательной передачи данных. Если необходимо формировать сигнал SS средствами приёмника – то для этих целей используется поле RSS\_RATE (Рисунок 18.28).

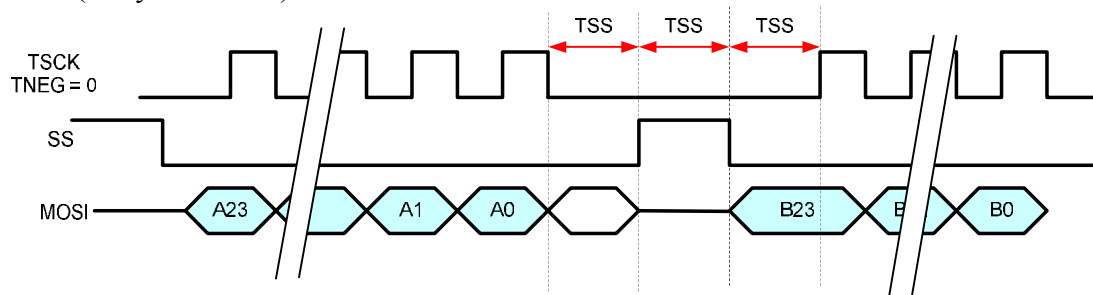


Рисунок 18.28. Управление временем удержания сигнала SS в высоком уровне между передачами, на картинке TNEG = 0, TDEL = 0, TMBF = 1, TWORDLEN = 23, TSS\_RATE = 1

#### 18.4.17 Пример чтения 8 разрядного слова по заданному адресу из ведомого устройства с интерфейсом C-BUS

Для чтения слова по указанному адресу по интерфейсу C-BUS необходима передача двух 8ми битных слов.

Для организации такого чтения необходимо записать соответствующий ведомому устройству бит SS, регистра TCTR, 1;

Перевести порт в режим SPI (LEN = 0, SPI\_I2S\_EN = 1, RMODE = 1, TMODE = 1);

Настроить приемник и передатчик: TDEL = RDEL = 0; TNEG = RNEG = 0; TWORDLEN = RWORDLEN = 5'h0F; RCLK\_CP = 1; RCS\_CP = 1, SS\_DO = 0;

Включить приемник и передатчик REN = 1, TEN = 1;

Записать в регистр TX\_MFBSP 32-х разрядное слово, содержащее во втором байте 7ми разрядный адрес и бит WR, значение младшего байта не важно.

Ожидаем до тех пор, пока в буфер приёма не будет записано принятое слово (RSR[0] сбрасывается в 0)

В прочитанном по адресу RX\_MFBSP 32-х разрядном слове, младшие 8 бит – слово, прочитанное из ведомого устройства.

На Рисунок 18.29 представлены временные диаграммы для передачи по интерфейсу CBUS.

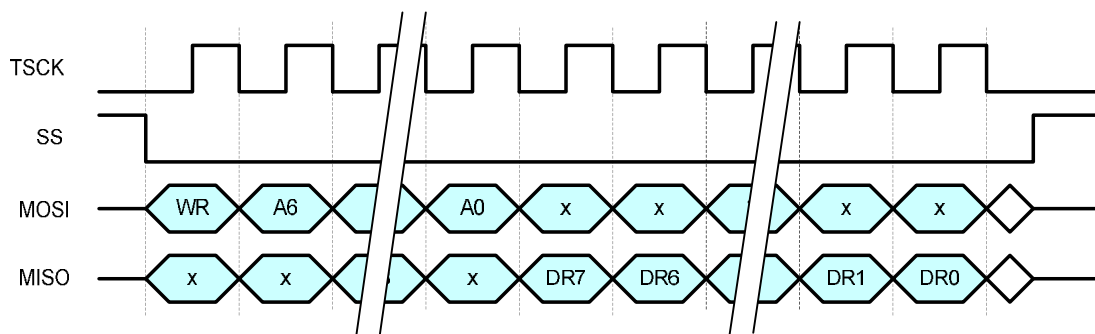


Рисунок 18.29. Пример чтения 8-ми разрядного слова из ведомого устройства (интерфейс C-BUS)

### 18.4.18 Формирование тактовых сигналов приёмника (RSCK) и передатчика (TSCK)

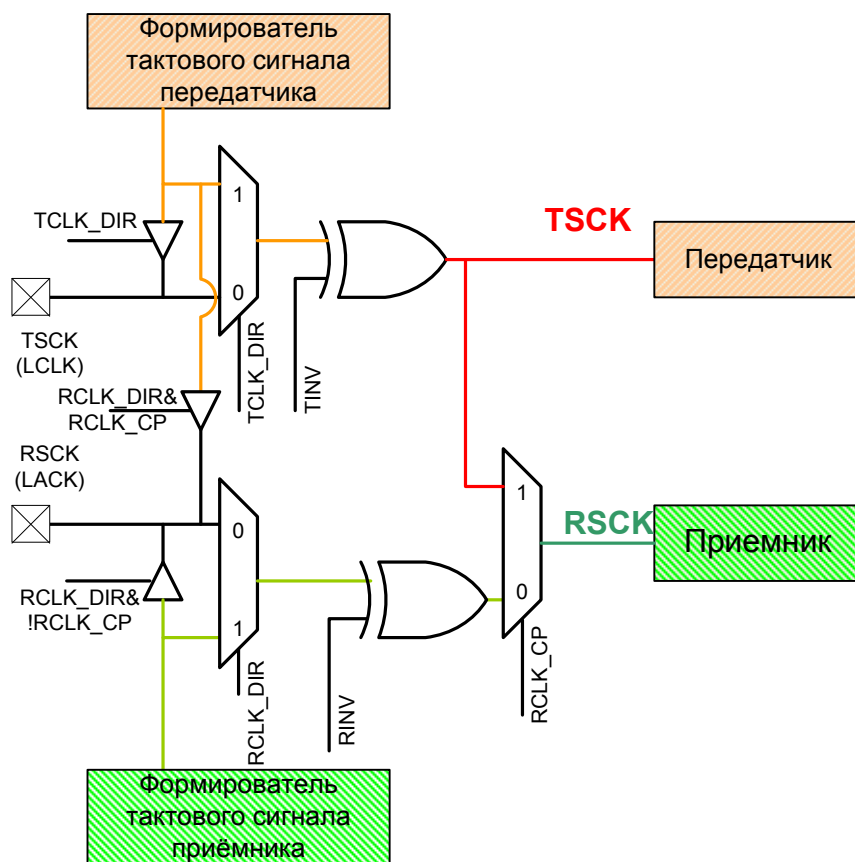


Рисунок 18.30. Схема формирования тактовых сигналов приёмника и передатчика в режиме SPI

На Рисунок 18.30 представлена схема формирования тактовых сигналов приёмника и передатчика в режиме SPI.

В зависимости от значения бита `TCLK_DIR`, тактовый сигнал передатчика `TCLK` может как формироваться самим передатчиком, так приниматься с внешнего вывода. В зависимости от значений бит `TMODE`, `TNEG` и `TDEL` тактовый сигнал либо передаётся передатчику без изменений, либо инвертируется.

В зависимости от значения бита `RCLK_DIR`, тактовый сигнал приёмника `RCLK` может как формироваться самим приёмником, так приниматься с внешнего вывода. В зависимости от значений бит `RMODE`, `RNEG` и `RDEL` тактовый сигнал либо передаётся приёмнику без изменений, либо инвертируется.

Если бит `RCLK_CP` установлен в 1, то тактовый сигнал приёмника копирует тактовый сигнал передатчика. Для корректной работы устройства в этом случае настройки полярности тактового сигнала приёмника и передатчика должны совпадать (`TNEG=RNEG`, `TDEL=RDEL`).

При `RCLK_CP = 1` тактовый сигнал передатчика передаётся на внешний вывод приёмника, только если передатчик сам формирует тактовый сигнал и вывод тактового сигнала приёмника сконфигурирован как выход (`TCLK_DIR=1`, `RCLK_DIR=1`).

### 18.4.19 Формирование управляющих сигналов приёмника и передатчика в режиме SPI

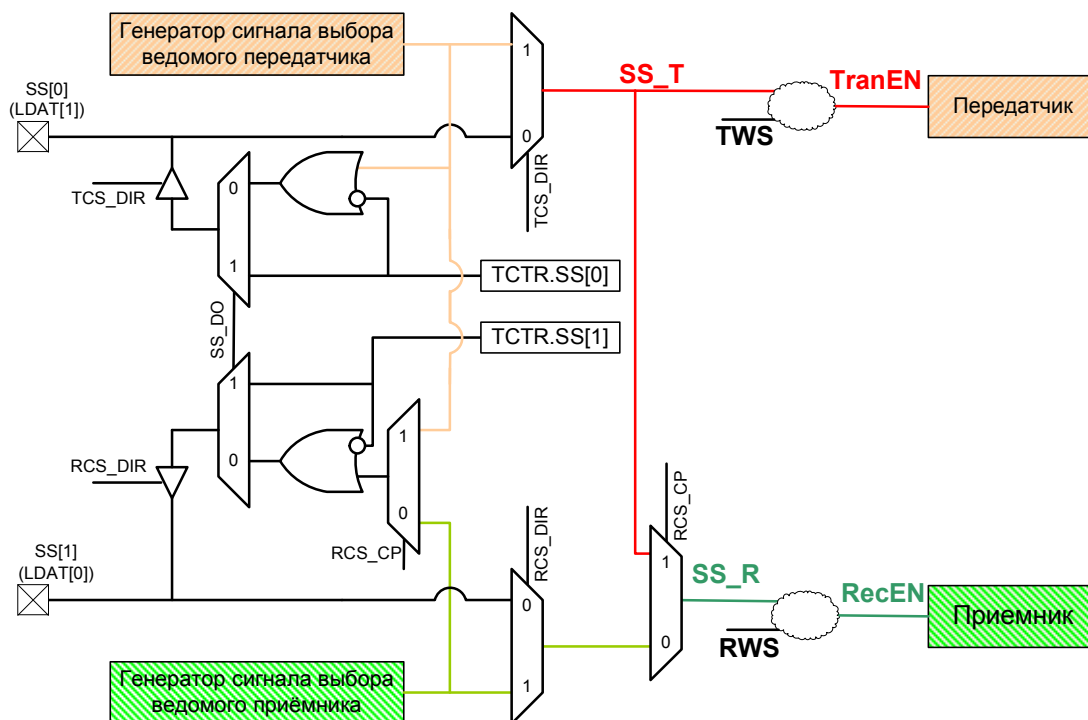


Рисунок 18.31. Схема формирования управляющих сигналов в режиме SPI

На Рисунок 18.31 представлена схема формирования управляющих сигналов в режиме SPI.

`SS` – шина выбора ведомого устройства. Низкий уровень сигнала `SS`, поданный на ведомое устройство означает, что данное устройство выбрано и с приходом тактового сигнала `SCK` должно начать обмен данными с ведущим устройством.

`MFBS` с зависимым от передатчика приёмником в режиме ведущего позволяет параллельно подключать до двух ведомых устройств по шине SPI и формировать сигналы выбора ведомого устройства как в автоматическом режиме, так и программно.

MFBSB с зависимым от передатчика приёмником может работать как ведомое SPI устройство, управляемое внешним сигналом SS[0] и внешней тактовой частотой TSCK, обеспечивая обмен данными в дуплексном режиме.

MFBSB позволяет организовать независимый приём и передачу данных по интерфейсу SPI. В этом случае SS[0] – управляющий сигнал передатчика, SS[1] – управляющий сигнал приёмника.

При TCS\_DIR = 1 передатчик SPI формирует сигнал выбора ведомого, SS[0] - выход. В автоматическом (SS\_DO=0) режиме формирования управляющего сигнала перед началом передачи очередного слова сигнал выбора ведомого переводится в низкий уровень, а по окончании передачи слова сигнал выбора ведомого снова переводится в высокий уровень. Изменение уровня на выводе SS[0] происходит только в случае, если соответствующий бит SS[0] регистра TCTR установлен в 1. Если приёмник в зависимом от передатчика режиме (RCS\_CP = 1) и SS[1] сконфигурирован как выход (RCS\_DIR=1), то вывод SS[1] используется как сигнал выбора дополнительного ведомого устройства. Изменение уровня на выводе SS[1] происходит только, в случае, если соответствующий бит SS[1] регистра TCTR установлен в 1. В случае программного управления шиной SS (SS\_DO = 1) значения бит SS[1:0] контрольного регистра TCTR передаются непосредственно на выходы SS[1:0].

Если приёмник в зависимом от передатчика режиме (RCS\_CP=1) и вывод SS[0] сконфигурирован как вход (TCS\_DIR = 0), тогда MFBSB работает в режиме дуплексного ведомого SPI устройства. Сигнал выбора ведомого принимается с внешнего вывода SS[0] и используется как приёмником, так и передатчиком.

Если приёмник работает в независимом от передатчика режиме (RCS\_CP=0), то в режиме ведущего, когда вывод SS[1] сконфигурирован как выход (RCS\_DIR=1) формируемый приёмником сигнал выбора ведомого направляется на вывод SS[1]. При автоматическом формировании управляющего сигнала (SS\_DO = 0) перед началом приёма очередного слова сигнал SS[1] автоматически переводится в низкий уровень и переводится в высокий уровень по окончании приёма каждого слова. В режиме ведущего устройства приём слов приёмником ведётся до заполнения буфера приёма. В режиме ведомого устройства, когда вывод SS[1] сконфигурирован как вход (RCS\_DIR=0) независимый приёмник (RCS\_CP=0) принимает сигнал выбора ведомого с вывода SS[1].

В режиме SPI направление выводов тактового сигнала и управляющего сигнала должно строго совпадать. Т.е. TCLK\_DIR=TCS\_DIR. В случае если приёмник работает независимо от передатчика, то RCLK\_DIR=RCS\_DIR.

#### 18.4.20 Тракт передачи данных

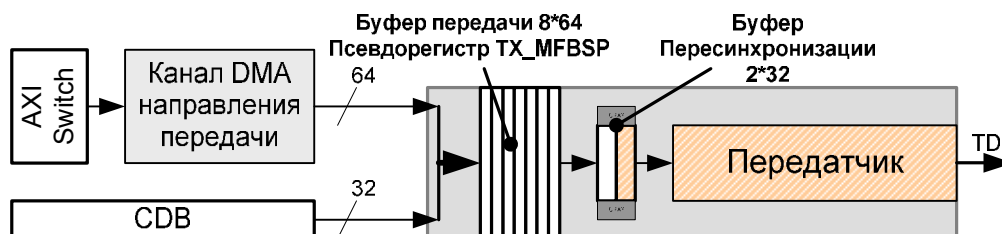


Рисунок 18.32. Тракт передачи данных в режиме SPI

На Рисунок 18.32 представлен тракт передачи данных в режиме SPI.

Что бы инициировать передачу данных по последовательному порту необходимо включить последовательный порт (SPI\_I2S\_EN=1) и передатчик (TEN=1), после чего либо начать производить запись передаваемых 32-х разрядных слов в буфер передачи по адресу псевдореги-

стра TX\_MFBSP, либо включить канал DMA в направления передачи для соответствующего порта (в этом случае обмен данными с портом будет вестись 64-х разрядными словами).

Данные записанные в буфер передачи автоматически перемещаются в буфер пересинхронизации направления передачи, если он не полон. Запись в буфер пересинхронизации направления передачи осуществляется на системной частоте CLK, чтение из буфера пересинхронизации осуществляется на частоте передатчика TCLK. Как только в буфере пересинхронизации оказалось хотя бы одно слово передатчиком инициируется передача. Передатчиком производится последовательная выдача бит очередного 32-х разрядного слова до тех пор, пока число переданных бит не достигнет TWORDLEN+1, после чего производится считывание очередного слова из буфера пересинхронизации. По мере передачи слов в освобождающийся буфер пересинхронизации перемещается слово из буфера передачи. После выборки последнего слова из буфера передачи (буфер передачи пуст) в буфере пересинхронизации остаётся еще два слова. Фактическое окончание передачи можно идентифицировать по состоянию буфера пересинхронизации, либо считав бит TRUN регистра TSR.

Если управляющий сигнал формируется передатчиком, то при считывании последнего слова из буфера пересинхронизации передача останавливается. Передача продолжится только после того как в буфер пересинхронизации снова начнут поступать данные.

Если передатчик использует внешнюю частоту и внешний управляющий сигнал, в целях экономии мощности системная частота может быть установлена меньшей, чем внешняя частота передатчика, однако ее должно быть достаточно для того, что бы успеть переместить очередное слово в буфер пересинхронизации (за время передачи одного слова должно быть хотя бы три импульса системной частоты CLK). Если внешний управляющий сигнал инициировал передачу слова при пустом буфере пересинхронизации устанавливается флаг ошибки передачи (TERR), в этом случае передается ошибочное слово. Если управляющий сигнал формируется самим передатчиком, системная частота может быть много меньше частоты передатчика, однако это скажется на скорости передачи данных.

Установка бита TERR в процессе передачи говорит о том, что порт произвел попытку чтения из пустого буфера передачи. Это значит, что передатчиком было передано некорректное слово.

В направлении передачи порт обладает буферизацией на 18 32-х разрядных слов. В случае передачи данных посредством DMA запись блоков данных в буфер передачи происходит до тех пор, пока буфер готов принять очередной блок, размер которого определяется битами WN, регистра CSR соответствующего канала DMA.

Степень заполнения буфера можно программно регулировать, используя биты TBES. В этом случае значение выражения TBES+1 – задает эффективный размер буфера передачи. Передача очередной пачки происходит только в случае, если при записи этой пачки в буфер передачи число 64 разрядных слов в буфере передачи не превысит TBES+1. При попытке передать пачку со значением  $WN > TBES$ , значение WN автоматически корректируется (большая пачка передается в несколько этапов пачками меньшего размера).

Так, при  $WN=0$  и  $TBES=0$  очередное 64-х разрядное слово будет подкачиваться в буфер передачи только если он пуст. В этом случае по окончании работы DMA в момент прерывания от соответствующего канала передатчику останется передать ещё 4 32-х разрядных слова (два 32-х разрядных слова в буфере пересинхронизации и одно 64-х разрядное слово в буфере передачи).

Установка бита SPI\_I2S\_EN в 0 приведет к программному сбросу передатчика, и все данные находящиеся в буфере передачи будут утеряны.



### 18.4.21 Тракт приёма данных

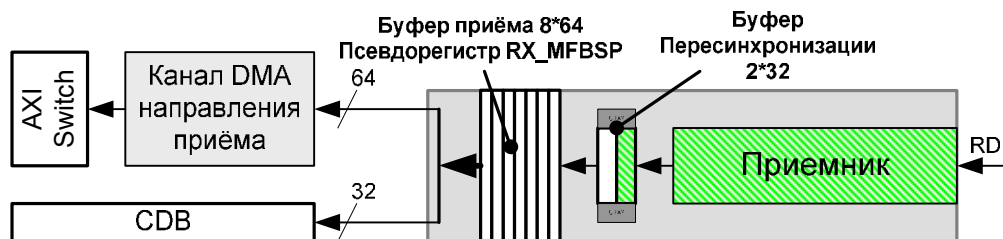


Рисунок 18.33. Тракт приёма данных в режиме SPI

На Рисунок 18.33 представлен тракт приёма данных в режиме SPI.

Что бы перевести приёмник в режим готовности необходимо включить последовательный порт ( $SPI\_I2S\_EN=1$ ) и приёмник ( $REN=1$ ), после чего либо начать ожидание появления прочитанных данных в буфере приёма, либо включить канал DMA в направления приёма для соответствующего порта.

Приёмник принимает последовательные биты, поступающие с внешнего вывода до тех пор, пока число принятых бит не достигнет значения  $RWORDLEN+1$ . После этого принятое 32-х разрядное слово (если  $RWORDLEN < 31$  незадействованные биты обнуляются) перемещается в буфер пересинхронизации. Запись в буфер пересинхронизации направления приёма осуществляется на частоте приёмника  $RCLK$ , чтение из буфера пересинхронизации осуществляется на системной частоте  $CLK$ . Из буфера пересинхронизации принятое слово автоматически перемещается в буфер приёма, если он не полон. Если в буфере приёма есть хотя бы одно 32-х разрядное слово, то принятые 32-х разрядные слова можно считывать, обращаясь по адресу псевдорегистра  $RX\_MFBS$ P. Принимать данные можно также включив соответствующий порту канал DMA направления приёма (в этом случае обмен данными с портом осуществляется 64-х разрядными словами).

Если приёмник использует внешнюю частоту, то в целях экономии мощности системная частота может быть установлена меньшей, чем внешняя частота приёмника, однако ее должно быть достаточно для того, что бы успеть переместить очередное слово из буфера пересинхронизации (за время приёма одного слова должно быть хотя бы три импульса системной частоты  $CLK$ ). Если при заполненном буфере пересинхронизации приёмником был произведен приём очередного слова и инициирована попытка записи в буфер пересинхронизации устанавливается флаг ошибки приёма ( $RERR$ ), а последнее принятое слово теряется.

Установка бита  $RERR$  в процессе передачи говорит о том, что порт произвел попытку записи в полный буфер приёма. Это значит, что принятое слово было потеряно.

В направлении приёма порт обладает буферизацией на 18 32-х разрядных слов. В случае приёма данных посредством DMA чтение блоков данных из буфера приёма происходит до тех пор, пока в буфере приёма достаточно слов для чтения очередного блока, размер которого определяется битами  $WN$ , регистра  $CSR$  соответствующего канала DMA. DMA обмена возможны только 64 разрядными словами, таким образом, если было принято нечетное количество 32-х разрядных слов, после окончания работы DMA необходимо прочитать оставшееся слово, обратившись к псевдорегистру  $RX\_MFBS$ P.

Установка бита  $SPI\_I2S\_EN$  в 0 приведет к программному сбросу приёмника и все данные находящиеся в буфере приёма будут утеряны.

### 18.4.22 Прерывания от последовательного порта

Прерывание MFBSP\_RXBUF устанавливается, в случае если включен приемник (I2S\_SPI\_EN=1, REN = 1) и в буфер приёма записано количество слов большее, чем установлено уровнем прерывания RLEV, либо произошла ошибка приема (RERR = 1).

Прерывание MFBSP\_TXBUF устанавливается, в случае если включен передатчик (I2S\_SPI\_EN=1, REN = 1) и в буфере передачи осталось количество слов меньше, либо равное чем установлено уровнем прерывания TLEV, либо произошла ошибка передачи (TERR = 1).

## 18.5 Работа MFBSP в режиме линкового порта (LPORT)

### 18.5.1 Назначение линкового порта

Линковый порт предназначен для обмена данными между различными микросхемами последовательно-параллельным кодом.

Порт может передавать 32-х разрядные слова частями по 4 бита за 8 пересылок, либо частями по 8 бит за 4 пересылки, выбор одного из этих режимов осуществляется установкой бита LDW, регистра CSR\_MFBSP.

### 18.5.2 Регистр управления и состояния CSR\_MFBSP (режим LPORT)

Таблица 18.27 Назначение разрядов регистра CSR\_MFBSP в режиме LPORT

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:17	-	Резерв	RW	0
16	-	Резерв	RW	0
15	-	Резерв	RW	0
14:11	LCLK_RATE [4:1]	Делитель частоты LPORT: $LCLK = CLK / (2 * (LCLK\_RATE + 1))$	RW	0
10	-	Резерв	RW	0
9	SPI_I2S_EN	В режиме LPORT должен быть установлен в 0	RW	0
8	SRQ_RX	Признак запроса обслуживания на прием данных	R	0
7	SRQ_TX	Признак запроса обслуживания на передачу данных	R	0
6	LDW	Разрядность внешней шины данных: 0 - 4-разряда (32-разрядное слово передается за 8 посылок); 1 - 8-разряда (32-разрядное слово передается за 4 посылки).	RW	0
5	LRERR	Ошибка приема данных: 0 – приняты все биты данных; 1 – приняты не все биты данных.	R	0
4:3	LSTAT	Состояние буфера: При LTRAN = 0 показывает состояние буфера приёма При LTRAN = 1 показывает состояние буфера передачи 00 – буфер пуст; 10 – буфер не пуст; 11 – буфер полон.	R	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
2	LCLK_RATE[0]	Делитель частоты LPORT: $LCLK = CLK / (2 * (LCLK\_RATE + 1))$	RW	0
1	LTRAN	Режим работы порта: 0 – приемник; 1 – передатчик.	RW	0
0	LEN	Разрешение работы порта: 0 – все выводы порта находятся в высокоимпедансном состоянии; 1 – порт работает в соответствии с состоянием бита LTRAN.	RW	0

Биты LSTAT, LRERR сбрасываются при LEN=0.

### 18.5.3 Регистр состояния приёмника RSR (режим LPORT)

Таблица 18.28 Назначение разрядов регистра RSR в режиме LPORT

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	резерв	-	0
27:24	RB_DIFF	Количество принятых 64-разрядных слов в буфере приёма (max 8).	R	0
23:19	-	Резерв	-	0
18:16	RLEV	Порог прерывания от буфера приёма: Прерывание формируется если число принятых 64-х разрядных слов больше RLEV	RW	7
15:11	-	Резерв	-	0
10	RXBUF	Результирующее прерывание MFBSP_RXBUF	R	0
9	RXBUF_D	Прерывание MFBSP_RXBUF без механизма автоматического сброса при чтении RSR	R	0
8	RXBUF_R	Прерывание MFBSP_RXBUF с механизмом автоматического сброса при чтении RSR	R	0
7	-	В режиме LPORT не используется	R	0
6	-	В режиме LPORT не используется	R	0
5	RSBF	Буфер пересинхронизации в направлении приёма полон: 0 – буфер пересинхронизации в направлении приёма не полон 1 – буфер пересинхронизации в направлении приёма полон	R	0
4	RSBE	Буфер пересинхронизации в направлении приёма пуст: 0 – буфер пересинхронизации в направлении приёма не пуст 1 – буфер пересинхронизации в направлении приёма пуст	R	1
3	RBHL	Достигнут порог прерывания в буфере приёма: 1 – число 64-х разрядных слов в буфере приёма больше чем задано в RLEV 0 – число 64-х разрядных слов в буфере приёма меньше либо равно RLEV	R	0
2	RBHF	Буфер приёма полон на половину или более: 1 – буфер приёма заполнен на половину или больше (из буфера приёма можно считать как минимум 4 слова) 0 – буфер приёма заполнен меньше чем на половину	R	0
1	RBF	Буфер приёма полон: 0 – буфер приёма не полон 1 – буфер приёма полон	R	0
0	RBE	Буфер приёма пуст: 0 – буфер приёма не пуст 1 – буфер приёма пуст	R	1

### 18.5.4 Регистр состояния передатчика TSR (режим LPORT)

Таблица 18.29 Назначение разрядов регистра TSR в режиме LPORT

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	резерв	-	0
27:24	TB_DIFF	Количество свободных 64-разрядных позиций в буфере передачи (в буфер передачи можно записать еще TB_DIFF 64-разрядных слов).	R	8
23	-	Резерв	-	0
22:20	TBES	Эффективный размер буфера передачи Актуален только для режима работы с DMA. Значение TBES+1 – определяет максимальный объем буфера передачи. Т.е. в режиме работы с DMA буфер передачи не может быть заполнен больше, чем на TBES+1 64 разрядных слов.	RW	7
19	-	Резерв	-	0
18:16	TLEV	Порог прерывания от буфера передачи: Прерывание формируется если число 64-х разрядных слов в буфере передачи меньше либо равно TLEV. В режиме передачи данных с использованием DMA определяет степень заполнения буфера передачи, при которой происходит запись в буфер очередной пачки данных	RW	0
15:11	-	Резерв	-	0
10	TXBUF	Результирующее прерывание MFBSP_TXBUF	R	0
9	TXBUF_D	Прерывание MFBSP_TXBUF без механизма автоматического сброса при чтении TSR	R	0
8	TXBUF_R	Прерывание MFBSP_TXBUF с механизмом автоматического сброса при чтении TSR	R	0
7	-	В режиме LPORT не используется	R	0
6	-	В режиме LPORT не используется	RW	0
5	-	В режиме LPORT не используется	R	0
4	-	В режиме LPORT не используется	R	0
3	TBLL	Достигнут порог прерывания в буфере передачи: 1 – число 64-х разрядных слов в буфере передачи меньше либо равно TLEV 0 – число 64-х разрядных слов в буфере передачи больше TLEV	R	1
2	TBNF	Буфер передачи заполнен на половину или более: 1 – буфер передачи заполнен на половину или больше 0 – буфер передачи заполнен меньше чем на половину (в буфер передачи можно записать еще как минимум 4 слова)	R	0
1	TBF	Буфер передачи полон: 0 – буфер передачи не полон 1 – буфер передачи полон	R	0
0	TBE	Буфер передачи пуст: 0 – буфер передачи не пуст 1 – буфер передачи пуст	R	1

### 18.5.5 Регистр аварийного управления портом EMERG\_MFBSP (режим LPORT)

Таблица 18.30 Назначение разрядов регистра EMERG\_MFBSP в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:6	-	Резерв	-	0
5	RX_DBG	Программное управление признаком готовности приема данных из DMA в MFBSP: 0 – штатный режим работы. Признак готовности формируется MFBSP аппаратно; 1 – признак готовности установлен в 1. Используется для приведение DMA в исходное состояние, если: устройство подключенное к MFBSP передало в него меньший объем данных, по сравнению с тем, что указано в DMA; необходимо программно остановить прием данных в MFBSP	RW	0
4	TX_DBG	Программное управление признаком готовности передачи данных из MFBSP в DMA: 0 – штатный режим работы. Признак готовности формируется MFBSP аппаратно; 1 – признак готовности установлен в 1. Используется для приведение DMA в исходное состояние, если: устройство подключенное к MFBSP приняло из него меньший объем данных, по сравнению с тем, что указано в DMA; необходимо программно остановить передачу данных из MFBSP	RW	0
3	-	Резерв	-	0
2	RST_RXBUF	Сброс буфера приема последовательного порта и буфера пересинхронизации направления приема. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0
1	RST_TXBUF	Сброс буфера передачи последовательного порта и буфера пересинхронизации направления передачи. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0
0	RST_LPTBUF	Сброс буфера линкового порта и буфера пересинхронизации направления приёма. Запись единицы в данный разряд приводит к сбросу буфера, после чего данный бит снова устанавливается в 0.	RW	0

### 18.5.6 Регистр маски прерываний от порта IMASK (режим LPORT)

Таблица 18.31 Назначение разрядов регистра IMASK в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:15	-	Резерв	-	0
14	RXBUF_R_EN	Разрешение автоматического сброса прерывания MFBSP_RXBUF 0 – прерывание не сбрасывается при чтении RSR 1 – прерывание сбрасывается при чтении RSR	RW	1
13	RX_LEV_IRQ_EN	Разрешение прерывания по уровню заполнения буфера приема 0 – прерывание MFBSP_RXBUF не будет устанавливаться при превышении порога RLEV 1 - прерывание MFBSP_RXBUF будет устанавливаться при превышении порога RLEV	RW	1
12	RX_ERR_IRQ_EN	Разрешение прерывания при переполнении буфера приема 0 – прерывание MFBSP_RXBUF не будет устанавливаться при переполнении буфера приема 1 - прерывание MFBSP_RXBUF будет устанавливаться при переполнении буфера приема	RW	1
11:7	-	Резерв	-	0
6	TXBUF_R_EN	Разрешение автоматического сброса прерывания MFBSP_TXBUF 0 – прерывание не сбрасывается при чтении TSR 1 – прерывание сбрасывается при чтении TSR	RW	1
5	TX_LEV_IRQ_EN	Разрешение прерывания по уровню заполнения буфера передачи 0 – прерывание MFBSP_TXBUF не будет устанавливаться, если число слов в буфере передачи меньше порога TLEV 1 - прерывание MFBSP_TXBUF будет устанавливаться, если число слов в буфере передачи меньше порога TLEV	RW	1
4	TX_ERR_IRQ_EN	Разрешение прерывания при переполнении буфера приема 0 – прерывание MFBSP_TXBUF не будет устанавливаться при чтении из пустого буфера передачи 1 - прерывание MFBSP_TXBUF будет устанавливаться при чтении из пустого буфера передачи	RW	1
3:1	-	Резерв	-	0
0	LPT_IRQ_EN	Разрешение прерывания по запросу на обслуживание 0 – SRQ запрещено 1 – SRQ разрешено	RW	1

### 18.5.7 Структурная схема MFBSB для режима линкового порта

На Рисунок 18.34 представлена структурная схема MFBSB для режима линкового порта. Включение линкового порта происходит при установке бита LEN в 1 и бита SPI\_I2S\_EN в 0.

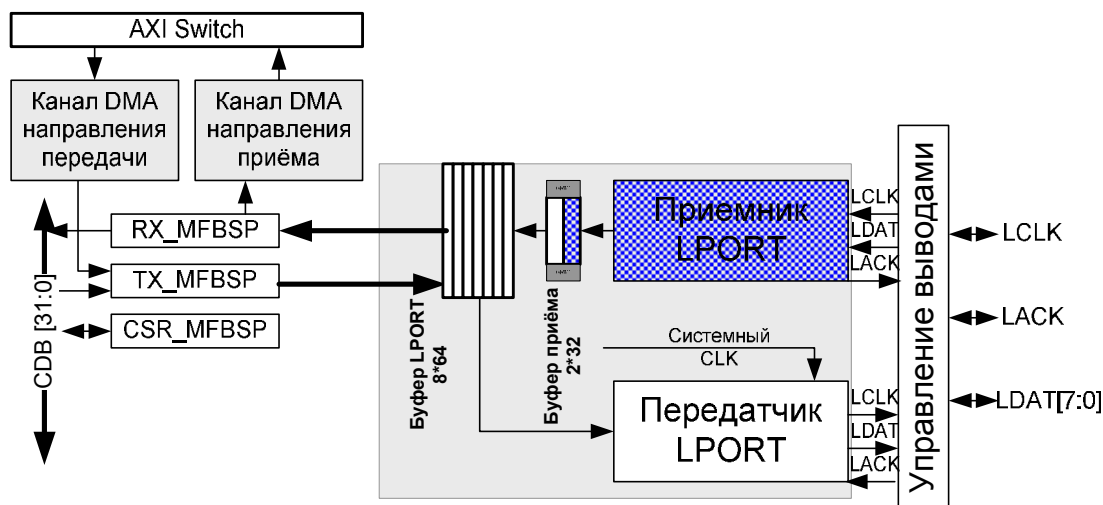


Рисунок 18.34 Структурная схема MFBSB для режима LPORT.

### 18.5.8 Соединение с внешними устройствами

На Рисунок 18.35 и Рисунок 18.36 представлены варианты соединения MFBSB с внешними устройствами в режиме линкового порта.

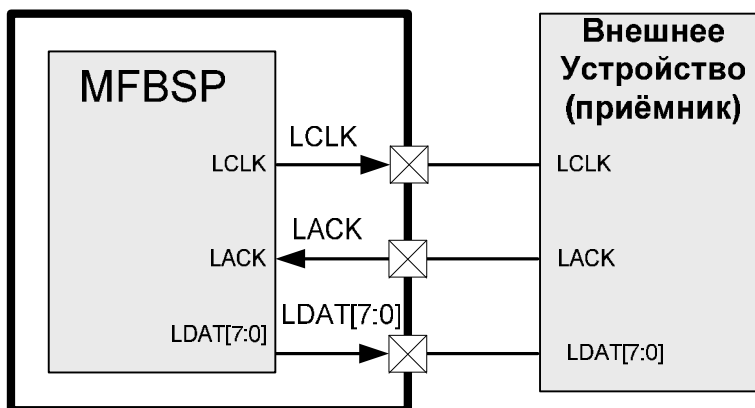


Рисунок 18.35. MFBSB в режиме передатчика LPORT (LCLK, LDAT-выходы, LACK - вход) (режим №2 по Таблица 18.1)

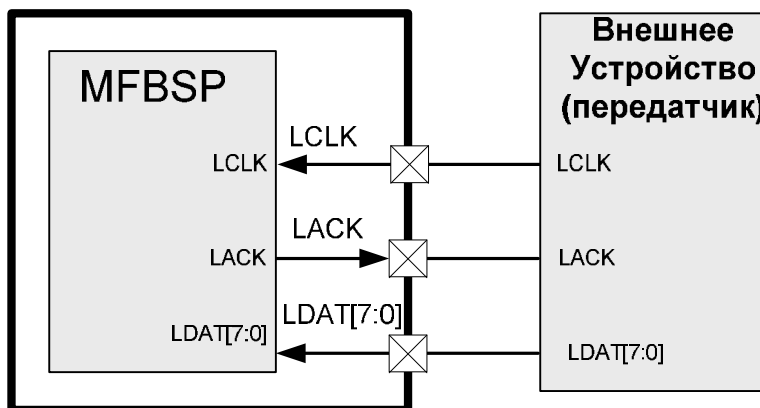


Рисунок 18.36. MFBSB в режиме приёмника LPORT (LCLK, LDAT-входы, LACK - выход) (режим №2 по Таблица 18.1)

### 18.5.9 Передача данных по линковому порту

По линковому порту передача данных происходит в одном направлении (либо передача данных, либо приём данных).

Для смены направления обмена данными по линковому порту необходимо сначала выключить порт (установить бит LEN, регистра CSR\_MFBSP в 0), затем включить порт, установив требуемое значение направления передачи данных (бит LTRAN, регистра CSR\_MFBSP)

Передача данных по линковому порту возможна для любых сочетаний частот приёмника и передатчика, скорость передачи данных будет определяться самым медленным устройством.

Для корректной передачи данных необходимо, чтобы значение бита LDW у приёмника и у передатчика совпадало.

Если для передатчика LDW=1, а для приёмника LDW=0 приёмник будет упаковывать два 32-х разрядных слова в одно 32-х разрядное слово, выкидывая из каждого байта старшие 4 бита. Установка значений LDW для передатчика LDW=0, а для приёмника LDW=1 не допускается. Временная диаграмма работы линкового порта приведена на Рисунок 18.37.

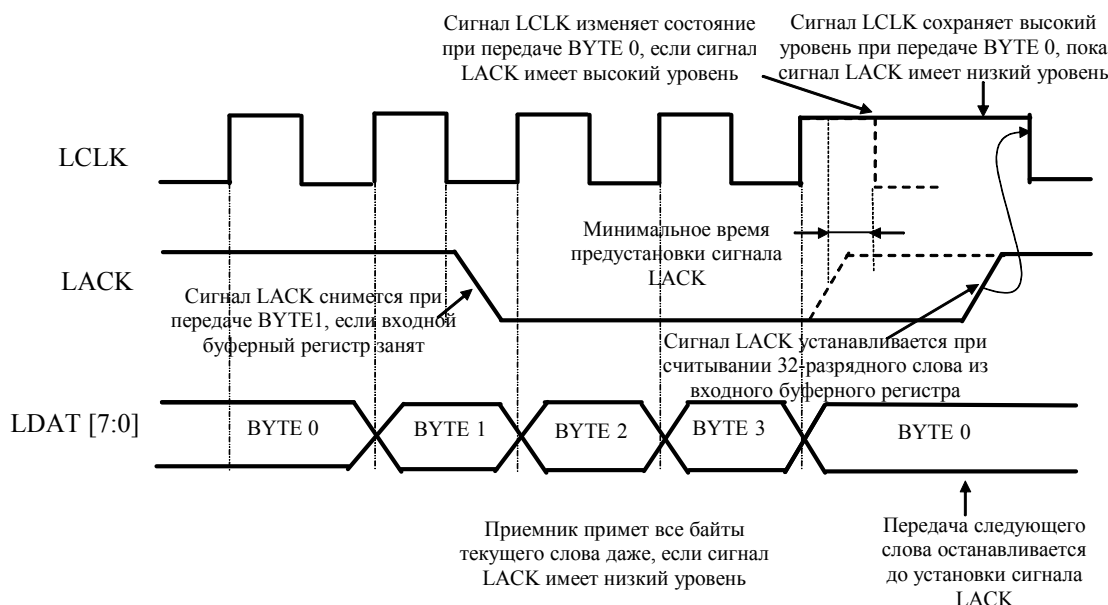


Рисунок 18.37 Временная диаграмма работы линкового порта (LDW=1)

При LDW=0 передача 32-разрядного слова выполняется за 8 посылок, а при LDW=1 - за 4 посылки. Передатчик изменяет данные LDAT по положительному фронту LCLK, а приемник защелкивает данные в буфере приёма по отрицательному фронту.

Исходное состояние сигнала LACK – высокий уровень. Сигнал LACK снимется приемником по заднему фронту LCLK при передаче BYTE1, если в буфере приёма осталось место для приёма всего одного слова. При этом приемник примет все байты текущего 32-разрядного слова даже, если сигнал LACK имеет низкий уровень. Сигнал LACK устанавливается при считывании 32-разрядного слова из входного буферного регистра.

Передатчик после выставления BYTE0 анализирует состояние сигнала LACK. Если LACK=1, то LCLK продолжает изменять свое состояние и после BYTE 0 передается BYTE 1 и так далее. Если LACK=0, то LCLK сохраняет высокий уровень при передаче BYTE 0, пока сигнал LACK имеет низкий уровень.

Если линковый порт деактивизирован ( $LEN=0$ ) сигналы LDAT, LCLK LACK являются входами. Поэтому эти сигналы необходимо привязывать к земле через резисторы 10 кОм. Если порт настроен как передатчик, LDAT и LCLK становятся выходами, а LACK – входом. Если порт настроен как приемник, LDAT и LCLK становятся входами, а LACK – выходом.

LPORT может выполнять либо только приём либо только передачу данных. Поэтому LPORT снабжен одним буфером на 8 64-х разрядных слов, используемом как в направлении приёма, так и в направлении передачи. В направлении приёма дополнительно встроено буфер на 2 32-х разрядных слова, используемый для пересинхронизации с внешней частоты LCLK на внутреннюю системную частоту.

Таким образом, LPORT обладает буферизацией в направлении передачи на 8 64-разрядных слов (16 32-разрядных слов) и буферизацией в направлении приёма на 9 64-разрядных слов (18 32-разрядных слов).

В случае передачи данных посредством DMA запись блоков данных в буфер передачи происходит до тех пор, пока буфер готов принять очередной блок, размер которого определяется битами WN, регистра CSR соответствующего канала DMA.

Степень заполнения буфера можно программно регулировать, используя биты TBES. В этом случае значение выражения  $TBES+1$  – задает эффективный размер буфера передачи. Передача очередной пачки происходит только в случае, если при записи этой пачки в буфер передачи число 64 разрядных слов в буфере передачи не превысит  $TBES+1$ . При попытке передать пачку со значением  $WN > TBES$ , значение WN автоматически корректируется (большая пачка передается в несколько этапов пачками меньшего размера).

Так, при  $WN=0$  и  $TBES=0$  очередное 64-х разрядное слово будет подкачиваться в буфер передачи только если он пуст. В этом случае по окончании работы DMA в момент прерывания от соответствующего канала передатчику останется передать ещё 4 32-х разрядных слова (два 32-х разрядных слова в буфере пересинхронизации и одно 64-х разрядное слово в буфере передачи).

Принимаемые портом данные сначала помещаются в буфер пересинхронизации и только через два такта перемещаются в буфер LPORT. При опросе контрольных регистров порта доступно состояние только буфера LPORT без учёта буфера пересинхронизации. Таким образом, после заполнения основного буфера LPORT могут быть приняты ещё два 32-х разрядных слова, которые будут перемещаться из буфера пересинхронизации в общий буфер LPORT по мере освобождения буфера LPORT.

Запись данных в буфер пересинхронизации LPORT осуществляется по внешней частоте LCLK, а перемещение данных из буфера пересинхронизации в буфер LPORT осуществляется по внутренней системной частоте CLK. Если внутренняя системная частота более чем в 4 раза меньше внешней частоты LCLK, скорости перемещения данных между двумя буферами может быть недостаточно, что будет приводить к периодическому заполнению буфера пересинхронизации. К потере данных это не приведет, поскольку в LPORT предусмотрен механизм останова передачи по заполнению буфера приёма, однако это приведёт к замедлению обмена данными по линковым портам.



### 18.5.10 Прерывания от линковых портов

Если линковый порт не активизирован ( $LEN=0$ ,  $SPI\_I2S\_EN=0$ ), он формирует прерывание по запросу обслуживания, если:

- на внешней шине выставлены данные на прием (активное состояние сигнала LCLK);
- из внешней шины поступил запрос на выдачу данных (активное состояние сигнала LACK).

Данное прерывание сбрасывается после установки  $LEN=1$ .

Если MFBSР используется в режиме линкового порта, то чтобы избежать ложной установки прерывания SRQ в случае, когда порт выключен и на выводах LACK или LCLK установлено высокоимпедансное состояние, необходимо к выводам LACK и LCLK подключить pull-down резисторы.

При  $LPT\_IRQ\_EN=0$  данное прерывание маскируется

Если включен линковый порт ( $LEN=1$ ) прерывания от MFBSР формируются в случае если в буфер приёма записано количество слов большее, чем установлено уровнем прерывания RLEV (MFBSР\_RXBUF), либо если при включенном передатчике в буфере передачи осталось количество слов меньшее, либо равное чем установлено уровнем прерывания TLEV (MFBSР\_TXBUF).

## 18.6 Работа MFBSР в режиме порта ввода-вывода общего назначения

Если многофункциональный порт выключен ( $LEN=0$ ,  $SPI\_I2S\_EN =0$ ), внешние линии LDAT[7:0], LCLK, LACK можно использовать как 10-разрядный двунаправленный порт ввода-вывода.

Если включен режим последовательного порта ( $SPI\_I2S\_EN = 1$ ), незадействованные в организации последовательной передачи данных выводы LDAT[7:4] могут быть использованы в качестве вводов-выводов общего назначения. Единственным ограничением в данной ситуации является то, что для определения режима работы последовательного порта используются биты GPIO\_DR[5:0], которые не должны меняться в процессе передачи данных по последовательному порту. Поэтому при управлении выводами общего назначения LDAT[7:4] (управляются битами DIR\_MFBSР [9:6]) запись в регистр DIR\_MFBSР необходимо проводить таким образом, что бы текущие значения бит DIR\_MFBSР [5:0] не менялись.

При работе в режиме выводов общего назначения данные с внешних выводов порта защелкиваются по положительному фронту тактового сигнала. Поэтому следует учитывать, что чтение данных с внешних выводов порта будет происходить с задержкой в 1 такт.

### 18.6.1 Регистр данных порта ввода вывода GPIO\_DR

10-разрядный регистр данных порта ввода-вывода (GPIO\_DR) предназначен для реализации гибкого интерфейса с внешними устройствами. Внешние выводы порта ввода-вывода совмещены с внешними выводами линкового порта.

Соответствие разрядов регистра GPIO\_DR и внешних линий линкового порта приведено в Таблица 18.32.



**Таблица 18.32 Назначение разрядов регистра GPIO\_DR**

Номер разряда Регистра GPIO_DR	Внешние выходы MFBSР	Значение после сброса
9:2	LDAT[7:0]	0
1	LCLK	0
0	LACK	0

### 18.6.2 Регистр управления направлением выводов DIR\_MFBSР

Настройка направления выводов порта ввода-вывода осуществляется программно при помощи 10-разрядного регистра DIR\_MFBSР. Если DIR\_MFBSР установлен в 0, то соответствующий разряд порта ввода-вывода является входом, если же разряд DIR\_MFBSР установлен в 1, то соответствующий разряд порта ввода-вывода является выходом.

**Таблица 18.33 Назначение разрядов регистра GPIO\_DR**

Номер разряда Регистра DIR_MFBSР	Внешние выходы MFBSР	Значение после сброса
9:2	Направление выводов LDAT[7:0]	0
1	Направление вывода LCLK	0
0	Направление вывода LACK	0

## 18.7 Рекомендации по аварийному выключению передатчика

В режимах SPI и I2S при TDEL = 1 выключение порта путем записи 0 в TEN, без сброса бита SPI\_I2S\_EN может привести к сбою в буфере передачи, и после очередного включения передатчика (TEN=1) данные будут передаваться некорректно. Для того, чтобы этого не происходило необходимо:

- 1) Если передатчик был выключен при TDEL=1 перед его очередным включением необходимо сбросить записью 1 в бит RST\_TXBUF
- 2) В режиме мастер выключать передатчик (если есть необходимость в дальнейшем использовать порт) вообще нет необходимости – отсутствие данных в буфере передачи автоматически останавливает дальнейшую передачу.

## 19. ПОРТ JTAG И ВСТРОЕННЫЕ СРЕДСТВА ОТЛАДКИ ПРОГРАММ

В данную микросхему встроен порт JTAG, реализованный в соответствии со стандартом IEEE 1149.1. Этот порт предназначен только для доступа к встроенным средствам отладки программ (OnCD) и не реализует Boundary Scan.

Модуль OnCD обеспечивает:

- \* выполнение остановки программы CPU по контрольным точкам (Breakpoint);
- \* выполнение заданного числа команд CPU (трассы) в реальном масштабе времени или пошаговое выполнение команд;
- \* доступ к адресуемым регистрам и памяти микросхемы.

Для подключения микросхемы к персональному компьютеру через порт JTAG необходимо использовать эмулятор JTAG, предназначенный для работы с данным микропроцессором.

## 20. ЭЛЕКТРИЧЕСКИЕ И ВРЕМЕННЫЕ ПАРАМЕТРЫ

### 20.1 Электропитание

Номинальные значения напряжений электропитания микросхемы приведены в Таблица 21.14. Допустимые отклонения напряжений электропитания микросхемы от номинального значения - не более  $\pm 5\%$ .

При включении на микросхему сначала необходимо подать напряжение электропитания ядра (CVDD)  $U_{CC2}$ , а затем напряжение электропитания периферии (PVDD)  $U_{CC1}$ . Задержка между подачей напряжений электропитания должна быть не более 10 мс. Входные сигналы подаются после подачи напряжений электропитания или одновременно с напряжением электропитания периферии  $U_{CC1}$ . Фронт нарастания напряжений электропитания должен быть не более 5 мс; При выключении микросхемы необходимо сначала снять входные сигналы, затем напряжение электропитания периферии  $U_{CC1}$ , затем, с задержкой не более 10 мс, напряжение электропитания ядра  $U_{CC2}$ .

Для фильтрации напряжений электропитания микросхемы, необходимо подключить к каждому источнику ( $U_{CC1}$  и  $U_{CC2}$ ) не менее десяти высокочастотных конденсаторов номиналом 0,1 мкФ типа СС 0603 Y5V 0,1 uF Z 25V. Конденсаторы необходимо разместить по возможности равномерно по периметру корпуса микросхемы между выводами PVDD и GND, а также CVDD и GND. При этом расстояние между контактами микросхемы и площадками подсоединения конденсаторов должно быть не более 3 мм.

### 20.2 Электрические параметры

Электрические параметры микросхемы приведены в Таблица 20.1.

Таблица 20.1. Электрические параметры микросхемы

Наименование параметров, единица измерения, режим измерения	Буквенное обозначение	Норма		Температура °С
		не менее	не более	
Ток потребления статический по цепи PVDD, мА при $U_{CC1} = 3,47$ В, $U_{CC2} = 1,26$ В, ХТИ = 0	$I_{CC1}$	-	20	от -60 до +85
Ток потребления статический по цепи CVDD, мА при $U_{CC1} = 3,47$ В, $U_{CC2} = 1,26$ В, ХТИ = 0	$I_{CC2}$	-	50	от -60 до +85
Ток потребления динамический по цепи CVDD, мА, при $U_{CC1} = 3,47$ В, $U_{CC2} = 1,26$ В и рабочей частоте 200 МГц	$I_{OCC2}$	-	2000	от -60 до +85
Ток утечки высокого и низкого уровня на входе, мкА при $U_{CC1} = 3,47$ В и $U_{CC2} = 1,26$ В	$I_{IL}$	-	2	от -60 до +85
Выходное напряжение низкого уровня, В при $I_{OL} = 4$ мА, $U_{CC1} = 3,47$ В	$U_{OL}$	-	0,4	от -60 до +85
Выходное напряжение высокого уровня, В при $I_{OH} = -4$ мА, $U_{CC1} = 3,13$ В	$U_{OH}$	2,4		от -60 до +85

Наименование параметров, единица измерения, режим измерения	Буквенное обозначение	Норма		Температура °С
		не менее	не более	
Входная емкость, пФ	C <sub>I</sub>	-	30	25 ± 10
Емкость входа/выхода, пФ	C <sub>I/O</sub>	-	30	25 ± 10

### 20.3 Динамическая потребляемая мощность

Динамическая потребляемая мощность микросхемы имеет две составляющие: потребление ядра (по цепи CVDD) и потребление выходных драйверов (по цепи PVDD).

Мощность, потребляемая ядром микросхемы по цепи CVDD, зависит от последовательности выполняемых процессорными ядрами команд, от операндов, а также от активности DMA и периферийных устройств. Максимальный ток, потребляемый ядром микросхемы, не превышает 2000 мА при внутренней частоте синхронизации 200 МГц.

Мощность, потребляемая выходными драйверами по цепи PVDD, зависит от следующих параметров:

- Число выходных драйверов (O);
- Максимальная частота, на которой выходные драйверы переключаются (F);
- Емкости нагрузки выходных драйверов (C);
- Величина напряжения электропитания выходных драйверов (U<sub>CC1</sub>).

Мощность, потребляемая выходными драйверами по цепи PVDD, определяется следующим уравнением:

$$P_{ext} = O * C * U_{CC1}^2 * F.$$

Рассмотрим для примера расчет мощности, потребляемой выходными драйверами при непрерывной записи данных в память типа SRAM (при U<sub>CC1</sub> = 3,3 В). Максимальная частота обмена данными со SRAM = CLK/4, где CLK – тактовая частота работы порта внешней памяти (например, 80 МГц). При обращении по произвольным адресам можно предположить, что с частотой CLK/4 изменяются 50% разрядов адреса. Также можно допустить, что каждый цикл изменяются 50% разрядов шины данных. Данные для расчета потребляемой мощности приведены в Таблица 20.2.

Таблица 20.2

Название драйвера	Число драйверов	Емкость нагрузки	F, МГц	U <sub>CC1</sub> <sup>2</sup>	P <sub>ext</sub> , мВт
A[31:0]	16	30	20	10,9	100
nWR[3:0]	4	30	20	10,9	25
D[63:0]	32	30	20	10,9	200
SCLK	1	30	80	10,9	25
Итого:					350

То есть, при тактовой частоте порта внешней памяти 80 МГц и C=30 пФ при непрерывной записи данных в SRAM потребление составляет 350 мВт. При чтении данных из SRAM вы-

ходные драйверы не активизируются. Поэтому, если запись данных в SRAM чередуется с чтением, то реальное энергопотребление микросхемы будет существенно меньше.

Оценим мощность, потребляемую драйверами линкового порта при передаче данных с частотой 40 МГц. Потребление по LCLK составляет 12 мВт, а потребление по данным (изменяется 50% 8-разрядных данных с частотой 20 МГц) - 24 мВт. Суммарно – 36 мВт.

## 20.4 Предельно-допустимые и предельные электрические режимы эксплуатации

Значения предельно-допустимых и предельных электрических режимов эксплуатации микросхемы приведены в Таблица 20.3.

**Таблица 20.3. Значения предельно-допустимых и предельных электрических режимов эксплуатации**

Наименование параметра, единица измерения	Буквенное обозначение	Норма			
		Предельно допустимый режим		Предельный Режим	
		не менее	не более	не менее	не более
Напряжение питания периферии, В	$U_{CC1}$	3,13	3,47	-	3,9
Напряжение питания ядра, В	$U_{CC2}$	1,14	1,26	-	1,4
Входное напряжение высокого уровня, В	$U_{IH}$	2,0	$U_{CC1}+0,2$	-	$U_{CC1}+0,3$
Входное напряжение низкого уровня, В	$U_{IL}$	0,0	0,7	-0,3	-
Напряжение, прикладываемое к выходу микросхемы в состоянии «выключено», В	$U_{OZ}$	0,0	$U_{CC1}+0,1$	-0,3	$U_{CC1}+0,3$
Емкость нагрузки каждого выхода, пФ	$C_L$	-	30	-	50

## 20.5 Временные параметры

Временные параметры при обмене данными с внешней памятью и устройствами приведены в Таблица 20.4.

Таблица 20.4. Временные параметры при обмене данными с внешней памятью и устройствами

Наименование параметра, единица измерения	Буквенное обозначение	Норма		Температу- ра °C
		не менее	не более	
Время задержки выходных сигналов A, D, nWR, nWE, nRD, nCS, SRAS, SCAS, SWE, DQM, CKE, A10, BA, nFLYBY, nOE после переднего фронта частоты SCLK, нс	$t_{DOSC}$	2	5	от -60 до +85
Время предустановки считываемых данных из асинхронной памяти перед задним фронтом частоты SCLK, нс	$t_{SDSC}$	6	-	от -60 до +85
Время удержания считываемых данных из асинхронной памяти после фронта снятия сигнала nRD, нс ( $t_{CLK}$ – период частоты CLK)	$t_{HDRD}$	0	$0,5 t_{CLK}$	от -60 до +85
Время предустановки считываемых данных из синхронной памяти перед передним фронтом частоты SCLK, нс	$t_{SDSC}$	5	-	от -60 до +85
Время удержания считываемых данных из синхронной памяти после переднего фронта частоты SCLK, нс	$t_{HDSC}$	0	$0,5 t_{CLK}$	от -60 до +85

Временная диаграмма при чтении данных из асинхронной памяти приведена на Рисунок 20.1. Считываемые данные фиксируются в микросхеме по заднему фронту частоты SCLK перед снятием сигнала nRD.

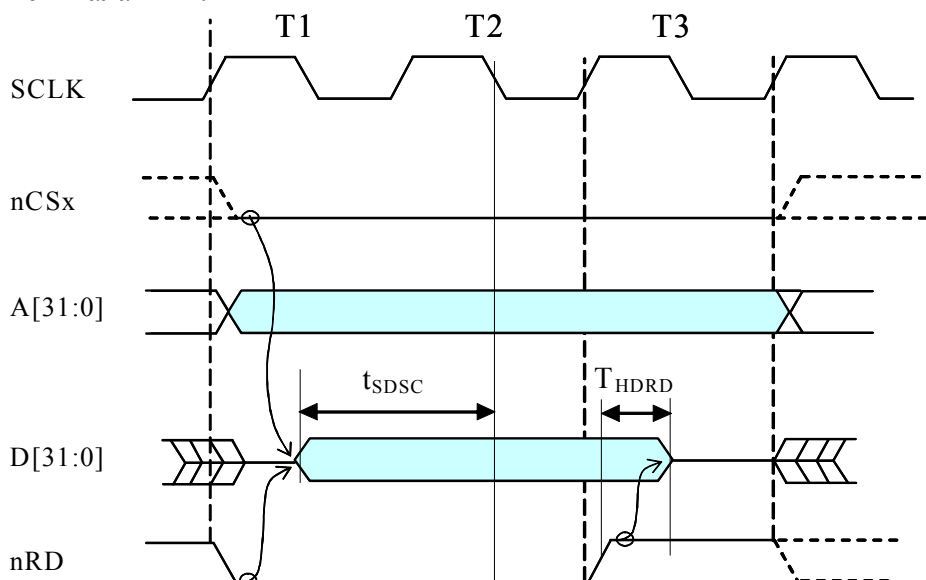


Рисунок 20.1. Чтение асинхронной памяти без дополнительных тактов ожидания.

## 20.6 Рекомендации по подключению кварцевого резонатора.

Схема подключения кварцевого резонатора к микросхеме приведена на Рисунок 20.2.

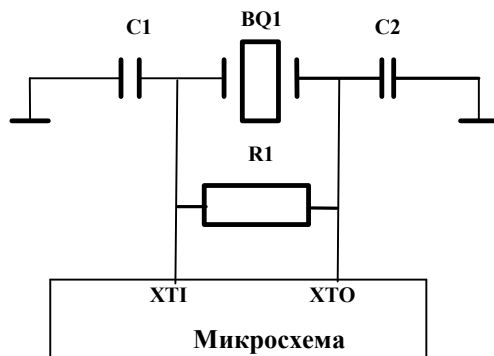


Рисунок 20.2. Схема подключения кварцевого резонатора к микросхеме

Частота кварцевого резонатора должна быть от 10 до 12 МГц. Ориентировочные величины:  $R1=1$  мОм,  $C1=C2=7$  пФ. Конкретная величина конденсаторов и резистора указывается в документации на резонатор.

## 21. ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ

Перечень сигналов микросхемы 1892ВМ7Я по группам, приведен в Таблица 21.1.

Таблица 21.1

Назначение	Число выводов
Порт внешней памяти, 64 разряда	146
2 порта DDR_PORT, 32 разряда	128
Управление	32
2 порта LP-Serial RapidIO (SRIO)	8
2 канала SpaceWire	16
Контроллер шины PCI (PMSC), 32 разряда	58
UART	2
ETHERNET	17
2 порта MFBSP	20
I2C	2
VPIN	13
VPOUT	20
Итого	478

Описание выводов микросхемы 1892ВМ7Я приведено в таблицах Таблица 21.2 - Таблица 21.14.

Таблица 21.2 Порт внешней памяти

Название вывода	Количество	Тип	Назначение
A[31:0]	32	O	Шина адреса.
D[63:0]	64	IO	Шина данных
nWRL[3:0], nWRH[3:0]	8	O	Запись байтов асинхронной памяти
nWEL, nWEH	2	O	Запись асинхронной памяти
nRD	1	O	Чтение асинхронной памяти
nWRSL[3:0] nWRSH[3:0]	8	O	Запись байтов синхронной памяти
nRDS	1	O	Чтение синхронной памяти
ACK	1	I	Готовность асинхронной памяти
nCS[4:0]	5	O	Разрешение выборки блоков внешней памяти
SRAS	1	O	Строб адреса строки
SCAS	1	O	Строб адреса колонки
SWE	1	O	Разрешение записи
DQM[7:0]	8	O	Маска выборки байта
SCLK	1	O	Тактовая частота работы
CKE	1	O	Разрешение частоты
A10	1	O	10 разряд адреса
BA[1:0]	2	O	Номер банка
nFLYBY[3:0]	4	O	Признак режима передачи DMA “Flyby”
nOE[3:0]	4	O	Разрешение чтения внешнего устройства (асинхронного)
Всего 146 выводов			



**Таблица 21.3 Порты DDR (2 штуки)**

Название вывода	Количество	Тип	Назначение
Порт 0			
A0[12:0]	13	O	Шина адреса.
DQ0[31:0]	32	IO	Шина данных
nCS0	1	O	Разрешение выборки блоков внешней памяти
RAS0	1	O	Строб адреса строки
CAS0	1	O	Строб адреса колонки
WE0	1	O	Разрешение записи
DQS0[3:0]	4	O	Строб данных
DM0[3:0]	4	O	Маска выборки байта
CK0[1:0], CK0n[1:0]	4	O	Тактовая частота
CKE0	1	O	Разрешение частоты
BA0[1:0]	2	O	Номер банка
Порт 1			
A1[12:0]	13	O	Шина адреса.
DQ1[32:0]	32	IO	Шина данных
nCS1	1	O	Разрешение выборки блоков внешней памяти
RAS1	1	O	Строб адреса строки
CAS1	1	O	Строб адреса колонки
WE1	1	O	Разрешение записи
DQS1[3:0]	4	O	Строб данных
DM1[3:0]	4	O	Маска выборки байта
CK1[1:0], CK1n[1:0]	4	O	Тактовая частота
CKE1	1	O	Разрешение частоты
BA1[1:0]	2	O	Номер банка
Всего 128 выводов			

**Таблица 21.4 Управление**

Название вывода	Количество	Тип	Назначение
nDMAR[7:0]	8	I	Запрос канала DMA. Формируется по отрицательному фронту. Минимальная длительность – не менее 1,5 периодов системной тактовой частоты CLK (частота, на которой работает CPU).
NMI	1	I	Немаскируемое прерывание. Формируется по положительному фронту сигнала
nIRQ[3:0]	4	I	Запросы прерывания. Потенциальные сигналы, активный низкий уровень

Название вывода	Количество	Тип	Назначение
BOOT[1:0]	2	I	<p>Определение источника и разрядности данных при начальной загрузке программ микропроцессора после снятия сигнала nRST:</p> <p>00 – загрузка производится из 32-разрядного блока памяти, подключенного к выводу nCS[3]. В этом случае разрядность этого блока памяти изменить нельзя;</p> <p>01 – загрузка производится из 8-разрядного блока памяти, подключенного к выводу nCS[3]. В этом случае разрядность этого блока памяти изменить нельзя;</p> <p>10 – загрузка производится из 64-разрядного блока памяти, подключенного к выводу nCS[3]. В этом случае разрядность этого блока памяти изменить нельзя;</p> <p>11 – загрузка производится из порта SPI. При этом к выводу nCS[3] может быть подключен 32-разрядный или 64-разрядный блок памяти. Его разрядность определяет бит W64 регистра CSCON3. Устанавливать BOOT=11 при PBOOT=1 запрещено</p>
PBOOT	1	I	<p>Признак режима выполнения процедуры начальной загрузки по адресу, задаваемого из шины PCI. Если микросхема 1892BM7Я используется в плате в составе персонального компьютера, обязательно должен быть установлен режим PBOOT=1, поскольку при PBOOT=0 после снятия сигнала nRST частота контроллера PMSC отключена и он не опознается на шине PCI при выполнении процедуры PLUG&amp;PLAY</p>
WDT	1	O	<p>Признак срабатывания сторожевого таймера. Этот сигнал формируется, если в программе произошел сбой. Его можно подать на системный контроллер, который будет принимать решение, что делать в данной ситуации</p>
PLL_EN	1	I	<p>Разрешение работы PLL:</p> <p>0 – системная тактовая частота микроконтроллера равна входной частоте ХТІ (см. рис. 4.1);</p> <p>1 - системная тактовая частота микроконтроллера поступает из PLL и равна входной частоте ХТІ, умноженной на коэффициент умножения/деления. (поле CLK_SEL регистра CSR).</p>
XTI, XTO	2	I, O	<p>Выходы для подключения внешнего кварцевого резонатора частотой от 10 до 12 МГц.</p> <p>На вывод ХТІ можно подать частоту от внешнего генератора, при этом вывод ХТО должен быть незадействованным.</p> <p>Стабильность частоты от внешнего генератора – не хуже +-50 ppm, скважность – от 40 до 60%, джиттер – не более 1 %</p>
RTC_XTI, RTC_XTO	2	I, O	<p>Выходы для подключения внешнего кварцевого резонатора частотой 32 КГц. На вывод RTCXTI можно подать частоту от внешнего генератора, при этом вывод RTCXTO должен быть незадействованным</p>
SRIO_CLK	1	I	<p>Сигнал тактовой частоты 125 МГц для контроллеров SRIO.</p>
HOST0, HOST1	2	I	<p>Признаки HOST для контроллеров SRIO0, SRIO1</p>

Название вывода	Количество	Тип	Назначение
nRST	1	I	Сигнал установки исходного состояния. Во время действия сигнала nRST все узлы микросхема находится в исходном (неактивном) состоянии, выходы - в неактивном состоянии, входы-выходы являются входами. При включении электропитания микросхемы сигнал nRST должен иметь низкий уровень и переключаться на высокий уровень через время не менее 1 мс после установки стабильного электропитания и стабильной тактовой частоты на входе XT1. Если необходимо установить работающую микросхему в исходное состояние, то для этого на нее необходимо подать асинхронный сигнал nRST длительностью не менее 10 тактов частоты на входе XT1. При этом, если к MPORT подключена память типа SDRAM, то до подачи сигнала nRST все операции обмена данными с SDRAM должны быть закончены. Фронт и спад сигнала nRST должен быть не более 100 нс.
TCK	1	I	Тестовый тактовый сигнал (JTAG)
TRST	1	I	Установка исходного состояния (JTAG)
TMS	1	I	Выбор режима теста (JTAG)
TDI	1	I	Вход данных теста (JTAG)
TDO	1	O	Выход данных теста (JTAG)
nDE	1	IO	Состояние DEBUG. Сигнал предназначен для отладки программного обеспечения нескольких микропроцессоров (до 8), работающих одновременно. Для этого выводы nDE у этих микросхем необходимо объединить в проводное ИЛИ. Если совместная отладка не используется, то вывод nDE должен быть незадействованным.
Всего 32 вывода			

**Таблица 21.5. Порт Ethernet MAC**

Название вывода	Количество	Тип	Назначение
MD	1	IO	Входные и выходные данные по интерфейсу MD
MDC	1	O	Тактовая частота обмена данными по интерфейсу MD
TX_CLK	1	I	Тактовая частота передачи данных по интерфейсу MII
TX_EN	1	O	Признак передачи данных по интерфейсу MII
TXD[3:0]	4	O	Шина передаваемых данных по интерфейсу MII
CRS	1	I	Сигнал наличия несущей в среде передачи
COL	1	I	Сигнал обнаружения коллизии в среде передачи
RX_CLK	1	I	Тактовая частота приема данных по интерфейсу MII
RX_DV	1	I	Признак наличия данных для приема по интерфейсу MII
RXD[3:0]	4	I	Шина принимаемых данных по интерфейсу MII
RX_ER	1	I	Признак обнаружения ошибки в принимаемых данных
Всего 17 выводов			

**Таблица 21.6. Порт ввода видеоданных**

Название вывода	Количество	Тип	Назначение
VDin[9:0]	10	I	Шина видеоданных
FRAME	1	I	Кадровая синхронизация
LINE	1	I	Строчная синхронизация
PIXCLK	1	I	Синхронизация пикселей
Всего 13 выводов			

**Таблица 21.7. Порт вывода видеоданных**

Название вывода	Количество	Тип	Назначение
VDout[15:0]	16	O	Шина видеоданных
VDEN	1	O	Признак действительности видеоданных
VSYNC	1	O	Кадровая синхронизация
HSYNC	1	O	Строчная синхронизация
VCLKO	1	O	Синхронизация пикселей
Всего 20 выводов			

**Таблица 21.8. Порты MFBSP (2 штуки)**

Название вывода	Количество	Тип	Назначение
LDAT0[7:0], LDAT1[7:0]	8	IO	Шина данных порта 0, 1
LCLK0, LCLK1	1	IO	Сигнал синхронизации порта 0, 1
LACK0, LACK1	1	IO	Признак подтверждения порта 0, 1
Всего 10*2=20 выводов			

**Таблица 21.9. Шина I2C**

Название вывода	Количество	Тип	Назначение
SCL	1	IO	Тактовая частота
SDA	1	IO	Последовательные данные
Всего 2 вывода			

**Таблица 21.10 Контроллер PCI (PMSC)**

Название вывода	Количество	Тип	Назначение
AD[31:0]	32	IO	Адрес/Данные
nC/BE[3:0]	4	IO	Команда/ выбор байта
nFRAME	1	IO	Признак выполнения операции передачи данных
nIRDY	1	IO	Готовность задатчика
nTRDY	1	IO	Готовность исполнителя
nSTOP	1	IO	Признак остановки передачи данных
PAR	1	IO	Дополнение до четности количества единиц на шинах AD и nC/BE
nPERR	1	IO	Ошибка четности
nDEVSEL	1	IO	Подтверждения выборки
IDSEL	1	I	Выборка при доступе к конфигурационным регистрам
nREQ	1	O	Запрос захвата шины
nGNT	1	I	Разрешение захвата шины
nINTA	1	O	Прерывание
PCLK	1	I	Тактовая частота работы шины PCI.
nREQB[4:0]	5	I	Запрос на использование шины PCI.
nGNTB[4:0]	5	O	Разрешение использования шины PCI.
Всего 58 выводов			

**Таблица 21.11 UART**

Название вывода	Количество	Тип	Назначение
SIN	1	I	Вход последовательных данных
SOUT	1	O	Выход последовательных данных
Всего 2 вывода			

**Таблица 21.12. Порты SpaceWire (2 штуки)**

Название вывода	Количество	Тип	Назначение
DINp0, DINp1	2	I	Вход данных положительный порта 0, 1
DINn0, DINn1	2	I	Вход данных отрицательный порта 0, 1
SINp0, SINp1	2	I	Вход строба положительный порта 0, 1
SINn0, SINn1	2	I	Вход строба отрицательный порта 0, 1
DOUtp0, DOUtp1	2	O	Выход данных положительный порта 0, 1
DOUn0, DOUn1	2	O	Выход данных отрицательный порта 0, 1
SOUtp0, SOUtp1	2	O	Выход строба положительный порта 0, 1
SOUtn0, SOUtn1	2	O	Выход строба отрицательный порта 0, 1
Всего 16 выводов			

**Таблица 21.13. Порты Serial RapidIO (2 штуки)**

Название вывода	Тип	Назначение
SRIO0		
TXP0/TXN0	O	Дифференциальный выход передачи данных порта 0
RXP0/RXN0	I	Дифференциальный вход приема данных порта 0
SRIO1		
TXP1/TXN1	O	Дифференциальный выход передачи данных порта 1
RXP1/RXN1	I	Дифференциальный вход приема данных порта 1
Всего 8 выводов		

**Таблица 21.14 Электропитание**

Название вывода	Назначение	
CVDD	40	Напряжение электропитания ядра ( $U_{CC2}$ , 1,2 В)
PVDD	45	Напряжение электропитания входных и выходных драйверов ( $U_{CC1}$ , 3,3 В)
GND	115	Земля ядра, входных и выходных драйверов
VREF0, 1	2	Относительное напряжение для приемников типа SSTL портов DDR_PORT0, DDR_PORT1 (1,25 В)
SR0_CVDD_TX	3	Напряжение электропитания передатчиков порта SRIO0 (1,2 В)
SR0_GND_TX	5	Земля передатчиков порта SRIO0
SR0_PVDD_TX	2	Напряжение электропитания элементов защиты передатчиков порта SRIO0 (3,3 В)
SR0_CVDD_RX	3	Напряжение электропитания приемников порта SRIO0 (1,2 В)
SR0_GND_RX	8	Земля приемников порта SRIO0
SR0_PVDD_RX	2	Напряжение электропитания элементов защиты приемников порта SRIO0 (3,3 В)
SR1_CVDD_TX	3	Напряжение электропитания передатчиков порта SRIO1 (1,2 В)
SR1_GND_TX	5	Земля передатчиков порта SRIO1
SR1_PVDD_TX	2	Напряжение электропитания элементов защиты передатчиков порта SRIO1 (3,3 В)
SR1_CVDD_RX	3	Напряжение электропитания приемников порта SRIO1 (1,2 В)
SR1_GND_RX	8	Земля приемников порта SRIO1
SR1_PVDD_RX	2	Напряжение электропитания элементов защиты приемников порта SRIO1 (3,3 В)
DDR_PVDD0	8	Напряжение электропитания приемопередатчиков SSTL порта DDR_PORT0 (2,5 В)
DDR_PVDD1	8	Напряжение электропитания приемопередатчиков SSTL порта DDR_PORT1 (2,5 В)
Всего 265 выводов		

Нумерация выводов микросхемы 1892BM7Я в корпусе HSBGA-765 приведена в Таблица 21.15.

Таблица 21.15. Нумерация выводов микросхемы 1892ВМ7Я в корпусе HSBGA-765

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
F14	LCLK0	AD1	RXP1[0]	T5	RTCXTI
E14	LACK0	AE1	RXN1[0]	T6	RTCXTO
A13	LDAT0[0]	AD2	RXP1[1]	M4	PLL_EN
B13	LDAT0[1]	AE2	RXN1[1]	M5	PBOOT
C13	LDAT0[2]	AD3	RXP1[2]	N6	WDT
D13	LDAT0[3]	AE3	RXN1[2]	A30	CKE
E13	LDAT0[4]	AD4	RXP1[3]	A31	BOOT[0]
F13	LDAT0[5]	AE4	RXN1[3]	A32	BOOT[1]
A12	LDAT0[6]	Y1	TXP1[0]	V1	SRIO_CLK
B12	LDAT0[7]	AA1	TXN1[0]	V2	HOST0
F11	LCLK1	Y2	TXP1[1]	V3	HOST1
B10	LACK1	AA2	TXN1[1]	P33	SCLK
C12	LDAT1[0]	Y3	TXP1[2]		
D12	LDAT1[1]	AA3	TXN1[2]	P5	nDMAR[0]
E12	LDAT1[2]	Y4	TXP1[3]	P6	nDMAR[1]
F12	LDAT1[3]	AA4	TXN1[3]	R1	nDMAR[2]
B11	LDAT1[4]			R2	nDMAR[3]
C11	LDAT1[5]	AL1	RXP0[0]	R3	nDMAR[4]
D11	LDAT1[6]	AM1	RXN0[0]	R4	nDMAR[5]
E11	LDAT1[7]	AL2	RXP0[1]	R5	nDMAR[6]
		AM2	RXN0[1]	R6	nDMAR[7]
B1	DINn0	AL3	RXP0[2]		
C1	DINp0	AM3	RXN0[2]	A14	COL
A4	SINn0	AL4	RXP0[3]	A15	TX_CLK
B4	SINp0	AM4	RXN0[3]	A16	RX_ER
A3	DOUn0	AG1	TXP0[0]	B14	CRS
B3	DOUp0	AH1	TXN0[0]	B15	TX_EN
A2	SOUTn0	AG2	TXP0[1]	B16	RX_DV
B2	SOUTp0	AH2	TXN0[1]	C14	MDC
D1	DINn1	AG3	TXP0[2]	C15	TXD[0]
E1	DINp1	AH3	TXN0[2]	C16	RXD[0]
C4	SINn1	AG4	TXP0[3]	D14	MD
D4	SINp1	AH4	TXN0[3]	D15	TXD[1]
C3	DOUn1			D16	RXD[1]
D3	DOUp1			E15	TXD[2]
C2	SOUTn1			E16	RXD[2]
D2	SOUTp1			F15	TXD[3]
				F16	RXD[3]
C10	SIN			F17	RX_CLK
D10	SOUT				
				B30	nFLYBY[0]
				B31	nFLYBY[1]
				B32	nFLYBY[2]
		M6	nDE	B33	nFLYBY[3]
		N2	TDI	C30	nOE[0]
		N3	TCK	C31	nOE[1]
D17	SDA	N4	TMS	C32	nOE[2]
E17	SCL	N5	TDO	C33	nOE[3]
		N1	TRST		
P1	nIRQ[0]				
P2	nIRQ[1]				
P3	nIRQ[2]	T1	XTI		
P4	nIRQ[3]	T2	XTO		
T3	NMI	T4	nRST		

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
R30	D[63]	AE29	D[6]	M33	A[12]
R31	D[62]	AE30	D[5]	L28	A[13]
R32	D[61]	AE31	D[4]	L29	A[14]
R33	D[60]	AE32	D[3]	L30	A[15]
T28	D[59]	AE33	D[2]	L31	A[16]
T29	D[58]	AF28	D[1]	L32	A[17]
T30	D[57]	AF29	D[0]	L33	A[18]
T31	D[56]	AF30	nWRSL[3]	K28	A[19]
T32	D[55]	AF31	nWRSL[2]	K29	A[20]
T33	D[54]	AF32	nWRSL[1]	K30	A[21]
U28	D[53]	AF33	nWRSL[0]	K31	A[22]
U29	D[52]	AL32	nWEL	K32	A[23]
U30	D[51]	AK32	nWEH	J28	A[24]
U31	D[50]	AM32	nRD	J29	A[25]
U32	D[49]	AN32	nRDS	J30	A[26]
U33	D[48]	AM31	nWRL[0]	J31	A[27]
V28	D[47]	AN31	nWRL[1]	J32	A[28]
V29	D[46]	AH32	nWRL[2]	H28	A[29]
V30	D[45]	AJ32	nWRL[3]	H29	A[30]
V31	D[44]	AH31	nWRH[0]	H30	A[31]
V32	D[43]	AJ31	nWRH[1]	H33	A10
V33	D[42]	AK31	nWRH[2]	H31	BA[0]
W28	D[41]	AL31	nWRH[3]	H32	BA[1]
W29	D[40]	AJ33	nWRSH[0]		
W30	D[39]	AK33	nWRSH[1]	E4	Vdin[0]
W31	D[38]	AL33	nWRSH[2]	F4	Vdin[1]
W32	D[37]	AM33	nWRSH[3]	F1	Vdin[2]
W33	D[36]	AH33	ACK	G2	Vdin[3]
Y28	D[35]	F31	SRAS	G3	Vdin[4]
Y29	D[34]	F32	SCAS	G4	Vdin[5]
Y30	D[33]	F33	SWE	G5	Vdin[6]
Y31	D[32]	E30	DQM[0]	G6	Vdin[7]
Y32	D[31]	E31	DQM[1]	G1	Vdin[8]
Y33	D[30]	E32	DQM[2]	H2	Vdin[9]
AA28	D[29]	E33	DQM[3]	H3	PIXCLK
AA29	D[28]	D30	DQM[4]	H4	LINE
AA30	D[27]	D31	DQM[5]	H5	FRAME
AA31	D[26]	D32	DQM[6]	H6	VDEN
AA32	D[25]	D33	DQM[7]	J2	VCLKO
AB28	D[24]	G29	nCS[0]	J3	HSYNC
AB29	D[23]	G30	nCS[1]	J4	VSYNC
AB30	D[22]	G31	nCS[2]	J5	Vdout[0]
AB31	D[21]	G32	nCS[3]	J6	Vdout[1]
AB32	D[20]	G33	nCS[4]	K2	Vdout[2]
AC28	D[19]	R29	A[0]	K3	Vdout[3]
AC29	D[18]	R28	A[1]	K4	Vdout[4]
AC30	D[17]	P32	A[2]	K5	Vdout[5]
AC31	D[16]	P31	A[3]	K6	Vdout[6]
AC32	D[15]	P30	A[4]	L1	Vdout[7]
AC33	D[14]	P29	A[5]	L2	Vdout[8]
AD28	D[13]	P28	A[6]	L3	Vdout[9]
AD29	D[12]	M28	A[7]	L4	Vdout[10]
AD30	D[11]	M29	A[8]	L5	Vdout[11]
AD31	D[10]	M30	A[9]	L6	Vdout[12]
AD32	D[9]	M31	A[10]	M1	Vdout[13]
AD33	D[8]	M32	A[11]	M2	Vdout[14]
AE28	D[7]			M3	Vdout[15]



Продолжение Таблица 21.15

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
AK25	A1[0]	AH28	nCS1	E25	A0[0]
AL25	A1[1]	AH29	WE1	D25	A0[1]
AM25	A1[2]	AJ28	RAS1	C25	A0[2]
AH26	A1[3]	AK28	CAS1	B25	A0[3]
AJ26	A1[4]	AJ29	DM1[0]	A25	A0[4]
AK26	A1[5]	AK29	DM1[1]	F26	A0[5]
AL26	A1[6]	AL29	DM1[2]	E26	A0[6]
AM26	A1[7]	AM29	DM1[3]	D26	A0[7]
AH27	A1[8]	AM30	BA1[0]	C26	A0[8]
AK27	A1[9]	AN30	BA1[1]	A26	A0[9]
AL27	A1[10]	AN29	CKE1	F27	A0[10]
AM27	A1[11]	AM24	CK1[0]	E27	A0[11]
AN27	A1[12]	AN24	CK1n[0]	D27	A0[12]
AH18	DQ1[0]	AM28	CK1[1]	C17	DQ0[0]
AJ18	DQ1[1]	AN28	CK1n[1]	B17	DQ0[1]
AK18	DQ1[2]			A17	DQ0[2]
AL18	DQ1[3]			F18	DQ0[3]
AM18	DQ1[4]	C27	nCS0	E18	DQ0[4]
AN18	DQ1[5]	C28	WE0	D18	DQ0[5]
AH19	DQ1[6]	F28	RAS0	C18	DQ0[6]
AJ19	DQ1[7]	E28	CAS0	B18	DQ0[7]
AM19	DQ1[8]	B28	DM0[0]	E19	DQ0[8]
AN19	DQ1[9]	A28	DM0[1]	D19	DQ0[9]
AH20	DQ1[10]	F29	DM0[2]	C19	DQ0[10]
AJ20	DQ1[11]	E29	DM0[3]	B19	DQ0[11]
AK20	DQ1[12]	B29	BA0[0]	A19	DQ0[12]
AL20	DQ1[13]	A29	BA0[1]	F20	DQ0[13]
AM20	DQ1[14]	D29	CKE0	E20	DQ0[14]
AN20	DQ1[15]	B24	CK0[0]	D20	DQ0[15]
AK21	DQ1[16]	A24	CK0n[0]	F21	DQ0[16]
AL21	DQ1[17]	B27	CK0[1]	E21	DQ0[17]
AM21	DQ1[18]	A27	CK0n[1]	D21	DQ0[18]
AN21	DQ1[19]			C21	DQ0[19]
AH22	DQ1[20]			B21	DQ0[20]
AJ22	DQ1[21]			F22	DQ0[21]
AK22	DQ1[22]			E22	DQ0[22]
AL22	DQ1[23]			D22	DQ0[23]
AJ23	DQ1[24]			F23	DQ0[24]
AK23	DQ1[25]			E23	DQ0[25]
AL23	DQ1[26]			D23	DQ0[26]
AM23	DQ1[27]			C23	DQ0[27]
AN23	DQ1[28]			B23	DQ0[28]
AH24	DQ1[29]			A23	DQ0[29]
AJ24	DQ1[30]			F24	DQ0[30]
AK24	DQ1[31]			E24	DQ0[31]
AH21	DQS1[1]			A18	DQS0[0]
AK19	DQS1[0]			C20	DQS0[1]
AM22	DQS1[2]			C22	DQS0[2]
AL24	DQS1[3]			D24	DQS0[3]
AH23	VREF1			A22	VREF0

Продолжение Таблица 21.15

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
AH11	PCLK	AH10	nGNTB[0]	A10	GND
AK8	nFRAME	AJ10	nGNTB[1]	A20	GND
AJ9	nCBE[0]	AK10	nGNTB[2]	A5	GND
AK9	nCBE[1]	AL10	nGNTB[3]	AB12	GND
AL9	nCBE[2]	AM10	nGNTB[4]	AB16	GND
AM9	nCBE[3]	AN7	nREQB[0]	AB17	GND
AM8	PAR	AM7	nREQB[1]	AB22	GND
AN10	nGNT	AL7	nREQB[2]	AB33	GND
AJ11	nREQ	AK7	nREQB[3]	AB5	GND
AH7	nPERR	AJ7	nREQB[4]	AG28	GND
AH8	nSTOP			AG29	GND
AH9	nDEVSEL			AG30	GND
AJ8	nTRDY			AG31	GND
AK11	IDSEL			AG32	GND
AL8	nIRDY			AG33	GND
AM11	AD[0]			AH17	GND
AN11	AD[1]			AH30	GND
AH12	AD[2]			AJ17	GND
AJ12	AD[3]			AK30	GND
AK12	AD[4]			AK17	GND
AL12	AD[5]			AJ30	GND
AM12	AD[6]			AL17	GND
AN12	AD[7]			AN15	GND
AH13	AD[8]			AN2	GND
AJ13	AD[9]			AN25	GND
AK13	AD[10]			AN9	GND
AL13	AD[11]			M12	GND
AM13	AD[12]			M16	GND
AN13	AD[13]			M17	GND
AH14	AD[14]			M22	GND
AJ14	AD[15]			N28	GND
AK14	AD[16]			N29	GND
AL14	AD[17]			N30	GND
AM14	AD[18]	H1	GND	N31	GND
AN14	AD[19]	F7	GND	N32	GND
AH15	AD[20]	F8	GND	N33	GND
AJ15	AD[21]	E7	GND	K1	GND
AK15	AD[22]	E8	GND	K33	GND
AL15	AD[23]	D6	GND	U3	GND
AM15	AD[24]	D7	GND	U4	GND
AH16	AD[25]	C6	GND	U5	GND
AJ16	AD[26]	C7	GND	U6	GND
AK16	AD[27]	C8	GND	P14	GND
AL16	AD[28]	C9	GND	P15	GND
AM16	AD[29]	E10	GND	P16	GND
AM17	AD[30]	F9	GND	P17	GND
AN17	AD[31]	E9	GND	P18	GND
AL11	nINTA	F10	GND	P19	GND
		D8	GND	P20	GND
		D9	GND	B5	GND

Продолжение Таблица 21.15

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
C5	GND	AB13	PVDD	AA20	CVDD
D5	GND	AB14	PVDD	AA22	CVDD
E5	GND	AB15	PVDD	A21	CVDD
V14	GND	A11	PVDD	AA12	CVDD
V15	GND	AA13	PVDD	AA33	CVDD
V16	GND	AA14	PVDD	AB18	CVDD
V17	GND	AA15	PVDD	AB19	CVDD
V18	GND	AA16	PVDD	AB20	CVDD
V19	GND	AB21	PVDD	AA17	CVDD
G28	GND	AJ5	PVDD	AA18	CVDD
V20	GND	AJ6	PVDD	AA19	CVDD
R14	GND	AK6	PVDD	AB6	CVDD
R15	GND	AN16	PVDD	AC6	CVDD
R16	GND	AN4	PVDD	AN26	CVDD
R17	GND	AN5	PVDD	AN3	CVDD
R18	GND	AN6	PVDD	AN8	CVDD
R19	GND	N13	PVDD	N17	CVDD
R20	GND	N14	PVDD	N18	CVDD
W14	GND	N15	PVDD	N19	CVDD
W15	GND	N16	PVDD	M18	CVDD
W16	GND	M13	PVDD	M19	CVDD
W17	GND	M14	PVDD	M20	CVDD
W18	GND	M15	PVDD	N12	CVDD
W19	GND	M21	PVDD	E6	CVDD
W20	GND	U13	PVDD	N20	CVDD
T14	GND	U21	PVDD	N22	CVDD
T15	GND	F6	PVDD	V12	CVDD
T16	GND	J1	PVDD	P12	CVDD
T17	GND	J33	PVDD	V22	CVDD
T18	GND	N21	PVDD	W12	CVDD
T19	GND	V13	PVDD	P22	CVDD
U1	GND	P13	PVDD	R12	CVDD
T20	GND	V21	PVDD	R22	CVDD
U14	GND	V4	PVDD	T12	CVDD
U15	GND	V5	PVDD	W22	CVDD
U16	GND	V6	PVDD	Y12	CVDD
U17	GND	P21	PVDD	Y22	CVDD
U18	GND	R13	PVDD	U12	CVDD
U19	GND	W13	PVDD	U22	CVDD
U2	GND	R21	PVDD	T22	CVDD
U20	GND	T13	PVDD		
Y14	GND	W21	PVDD		
Y15	GND	Y13	PVDD		
Y16	GND	Y21	PVDD		
Y17	GND	T21	PVDD		
Y18	GND	AA21	PVDD		
Y19	GND				
Y20	GND				
F5	GND				

Продолжение Таблица 21.15

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
AC4	SR1_GND_RX			AH25	DDR_PVDD1
AC5	SR1_CVDD_RX			AJ21	DDR_PVDD1
AC3	SR1_CVDD_RX			AJ25	DDR_PVDD1
AC1	SR1_CVDD_RX			AJ27	DDR_PVDD1
AC2	SR1_GND_RX			AL19	DDR_PVDD1
AB3	SR1_GND_RX			AL28	DDR_PVDD1
AB4	SR1_GND_RX			AL30	DDR_PVDD1
AB2	SR1_GND_RX			AN22	DDR_PVDD1
AE5	SR1_GND_RX				
AE6	SR1_PVDD_RX			B20	DDR_PVDD0
Y5	SR1_GND_TX			B22	DDR_PVDD0
Y6	SR1_PVDD_TX			B26	DDR_PVDD0
W1	SR1_CVDD_TX			C24	DDR_PVDD0
W2	SR1_GND_TX			F25	DDR_PVDD0
W3	SR1_CVDD_TX			F19	DDR_PVDD0
W4	SR1_GND_TX			D28	DDR_PVDD0
W5	SR1_CVDD_TX			C29	DDR_PVDD0
W6	SR1_GND_TX				
AD5	SR1_GND_RX				
AD6	SR1_PVDD_RX				
AA5	SR1_GND_TX				
AA6	SR1_PVDD_TX				
AB1	SR1_GND_RX				
AL5	SR0_GND_RX				
AL6	SR0_PVDD_RX				
AM5	SR0_GND_RX				
AM6	SR0_PVDD_RX				
AJ1	SR0_GND_RX				
AJ4	SR0_GND_RX				
AF1	SR0_CVDD_TX				
AF2	SR0_GND_TX				
AF3	SR0_CVDD_TX				
AF4	SR0_GND_TX				
AF5	SR0_CVDD_TX				
AF6	SR0_GND_TX				
AG5	SR0_GND_TX				
AG6	SR0_PVDD_TX				
AH5	SR0_GND_TX				
AH6	SR0_PVDD_TX				
AK2	SR0_GND_RX				
AK1	SR0_CVDD_RX				
AK3	SR0_CVDD_RX				
AK4	SR0_GND_RX				
AK5	SR0_CVDD_RX				
AJ2	SR0_GND_RX				
AJ3	SR0_GND_RX				

## 22. ИСТОРИЯ ИЗМЕНЕНИЙ

### 22.1 20 декабря 2012 г

- Уточнен п. 21.3

### 22.2 31 июля 2013 г

- Из п. 1.3 удалены каналы DMA GSWIC.
- Из табл. 2.50 удалены сигналы прерывания GSWIC.
- Из табл. 4.6 удалены сигналы прерывания GSWIC.
- Из раздела 7 удалены каналы DMA GSWIC.
- Дополнен раздел 15 в части выполнения на шине PCI процедуры RETRY в режиме Master.
- Из табл. 22.16 удалены сигналы gSW\_RXP и gSW\_RXN.

### 22.3 08 апреля 2015 г

- Уточнены пункты 1,2, 3.1 в части пиковой производительности DSP.