

МИКРОСХЕМЫ ИНТЕГРАЛЬНЫЕ 1892ХД4Ф

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

РАЯЖ.431262.007Д17

ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ.....	5
1.1 Назначение.....	5
1.2 Функциональные параметры и возможности.....	6
2. СИСТЕМНАЯ ОРГАНИЗАЦИЯ МИКРОСХЕМЫ	7
2.1 Структурная схема.....	7
2.2 Схема синхронизации.....	9
2.3 Карта памяти	10
2.4 Регистр CLK_EN.....	20
3. АДАПТЕР СОПРЯЖЕНИЯ С ШИНОЙ ВНЕШНЕГО ПРОЦЕССОРА (МВА)..	21
3.1 Назначение.....	21
3.2 Регистры МВА	21
3.2.1 Регистры запросов прерывания	21
3.2.2 Регистры маски запросов прерывания	23
3.2.3 Регистр буферизации адреса и данных BDR.....	23
3.2.4 Регистр BUSY.....	24
3.3 Обмен данными с через адаптер МВА	24
3.3.1 Общие положения	24
3.3.2 Обмен данными с использованием признака занятости BUSY.....	25
3.3.3 Обмен данными с использованием сигнала nACK.....	26
3.4 Подключение к микропроцессору нескольких микросхем 1892ХД4Ф.....	27
4. КОНТРОЛЛЕР ШИНЫ PCI	29
4.1 Функциональные параметры и возможности.....	29
4.2 Структурная схема.....	29
4.3 Регистры.....	30
4.3.1 Конфигурационные регистры	32
4.3.2 Регистры управления обменом данных	34
4.4 Обмен данными по шине PCI в режиме Target (Slave)	39
4.5 Обмен данными с шиной PCI в режиме Master	40
4.6 Обмен данными с шиной PCI через адресное окно.....	42
4.7 Передача прерываний.....	44
4.7.1 Передача вектора прерывания из шины PCI	44
4.7.2 Передача прерываний в шину PCI.....	44
4.8 Арбитр.....	46
5. КОНТРОЛЛЕР ИНТЕРФЕЙСА SPACEWIRE (SWIC)	47
5.1 Общие положения.....	47
5.2 Блок-схема	47
5.3 Прерывания	49
5.4 Перечень регистров SWIC	50
5.4.1 Общие положения	50
5.5 Описание регистров SWIC.....	51
5.5.1 Регистр HW_VER.....	51
5.5.2 Регистр STATUS	52
5.5.3 Регистр RX_CODE	54
5.5.4 Регистр MODE_CR	54

5.5.5	Регистр TX_SPEED	56
5.5.6	Регистр TX_CODE	57
5.5.7	Регистр RX_SPEED	57
5.5.8	Регистр CNT_RX_PACK	57
5.5.9	Регистр ISR_L	58
5.5.10	Регистр ISR_H	58
5.5.11	Регистр TRUE_TIME	59
5.5.12	Регистр TOUT_CODE	59
5.5.13	Регистр ISR_tout_L	59
5.5.14	Регистр ISR_tout_H	60
5.5.15	Регистр LOG_ADDR	60
5.5.16	Регистр ACK_NONACK_MODE	60
5.5.17	Регистр ISR2_tout	61
5.5.18	Регистр ISR_HANDLER_TERM_FUNC	61
5.5.19	Регистр специальных кодов ISR_SPEC	61
5.5.20	Регистр ISR_1101	63
5.5.21	Регистр ISR_MASK_1101	63
5.5.22	Регистр INT_RESET	63
5.5.23	Регистр MODE_CR2	64
5.5.24	Регистры маски распределенных прерываний – INT_H_MASK, INT_L_MASK	65
5.5.25	Регистры маски кодов подтверждения – ACK_H_MASK, ACK_L_MASK	65
5.5.26	Регистр AUTO_SPEED_MANAGE	65
5.5.27	Регистр ISR_SOURCE_TERM_FUNC	66
5.5.28	Регистр ISR_SPEC_TERM_FUNC	66
5.5.29	Регистр ISR_L_RESET	67
5.5.30	Регистр ISR_H_RESET	67
5.6	Логика работы SWIC	67
5.6.1	Расположение данных в памяти	67
5.6.2	Схема обработки данных	68
5.6.3	Прием данных из канала SpaceWire	68
5.6.4	Передача данных в канал SpaceWire	69
5.6.5	Выравнивание границ пакетов по границам слов	71
5.6.6	Формат дескриптора пакета	72
5.6.7	Возможность передачи коммуникационного пакета	73
5.6.8	Использование симплексного режима	74
5.6.9	Маркеры времени	75
5.6.10	Коды распределенных прерываний и подтверждений	76
5.6.11	Установка скорости передачи данных	81
5.6.12	Установление соединения	82
5.6.13	Определение скорости приема данных	82
5.7	Логика работы DMA SWIC	83
5.7.1	Каналы DMA SWIC	83
5.7.2	Регистры DMA SWIC	83
5.7.3	Процедура самоинициализации	86
5.7.4	Прерывания DMA	87
5.8	Примеры работы со SWIC	87
6.	КОНТРОЛЛЕР ETHERNET MAC 10/100 (EMAC)	90
6.1	Основные характеристики EMAC	90

6.2	Структурная схема ЕМАС	91
6.3	Регистры контроллера ЕМАС.....	93
6.3.1	Регистры DMA.....	94
6.3.2	Регистры управления и состояния.....	96
6.3.3	Регистры адресов.....	104
6.4	Описание логики работы ЕМАС	106
6.4.1	Контроллер DMA	106
6.4.2	Порт управления PNY – MD_PORT	107
6.4.3	Блок передачи кадров TransmitFrame.....	109
6.4.4	Блок CALC_CRC32	125
6.4.5	Блок BACKOFF	125
6.4.6	Режим тестирования TX_FIFO	126
6.4.7	Блок приема кадров ReceiveFrame.....	126
6.4.8	Блок DADDR_CHECK.....	135
6.4.9	Блок CRC32_CHECK.....	137
6.4.10	Режим тестирования RX_FIFO.....	138
7.	ПРИНЦИПЫ КОРРЕКЦИИ ОШИБОК.....	140
8.	ЭЛЕКТРИЧЕСКИЕ ПАРАМЕТРЫ.....	143
8.1	Электропитание.....	143
8.2	Электрические параметры при приемке и поставке.....	144
8.3	Предельно-допустимые и предельные электрические режимы эксплуатации.	144
9.	ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ МИКРОСХЕМЫ.....	145
10.	ИСТОРИЯ ИЗМЕНЕНИЙ.....	151
10.1	5 сентября 2013 г	151
10.2	29 ноября 2013 г.....	151
10.3	30 декабря 2013 г	151
10.4	27 августа 2014 г.....	151
10.5	28 января 2016 г	151

1. ВВЕДЕНИЕ

1.1 Назначение

Микросхема интегральная 1892ХД4Ф (МСВ-03Р) спроектирована как однокристалльный контроллер ввода-вывода на базе IP-ядерной (IP-intellectual property) платформы «МУЛЬТИКОР», разработанной в ОАО НПЦ «ЭЛВИС».

Микросхема 1892ХД4Ф предназначена:

- для сопряжения микропроцессоров, например, сигнальных микропроцессоров семейства «Мультикор», с каналами связи SpaceWire, Ethernet и шиной PCI;
- для сопряжения устройств с интерфейсом PCI с каналами связи SpaceWire и Ethernet.

Использование микросхемы 1892ХД4Ф позволяет процессору передавать и принимать данные с использованием четырех контроллеров SWIC, взаимодействовать с блоками памяти на шине PCI. Процессор обменивается данными с микросхемой 1892ХД4Ф через интерфейс асинхронной памяти.

Использование микросхемы 1892ХД4Ф так же возможно под управлением процессора через интерфейс PCI. Интерфейс PCI функционирует в 32-х разрядном режиме, частота 33 или 66 МГц.

Микросхема 1892ХД4Ф генерирует прерывания (nINT на шине памяти и nINTA на шине PCI), причиной которых может являться установка соединения по каналам связи SpaceWire, разрыв соединения, получение маркера времени или конца пакета, завершение областей памяти, на которые настроены контроллеры SWIC, и т.д..

Микросхема 1892ХД4Ф обеспечивает взаимодействие вычислительных модулей в распределенных вычислительных системах, комплексах обработки сигналов и данных.

Микросхема 1892ХД4Ф обеспечивает взаимодействие процессора с коммуникационной сетью на основе стандарта SpaceWire, которая, как правило, строится на основе коммутаторов-маршрутизаторов. Каналы связи SpaceWire также могут быть использованы для сопряжения с источниками/приемниками высокоскоростных потоков цифровых сигналов и данных.

Таким образом, микросхема 1892ХД4Ф позволяет обеспечить взаимодействие процессоров и периферийных устройств, поддерживающих стандарт SpaceWire и PCI в широком диапазоне приложений, от передачи разнородного потока коротких пакетов с использованием технологии виртуальных каналов до непрерывного однородного потока данных.

1.2 Функциональные параметры и возможности

Микросхема 1892ХД4Ф имеет следующие функциональные параметры и возможности:

- порт для сопряжения с внешним микропроцессором:
 - шина данных – 32 разряда, шина адреса – 25 разрядов;
 - асинхронный режим работы с сигналом готовности данных;
- контроллер шины PCI:
 - соответствует спецификации Local Bus Specification. Rev. 2.2;
 - тактовая частота – от 33 до 66 МГц;
 - шина адреса и данных - 32 разряда;
 - обмен данными в режиме Master и Target (Slave);
 - встроенный арбитр на 5 запросов шины;
 - встроенные средства для организации мультипроцессорных систем;
 - обеспечивает обмен данными между шиной PCI и любой областью внутренней памяти и регистрами устройств;
- четыре контроллера канала связи SpaceWire:
 - соответствуют стандарту ECSS-E-50-12C;
 - скорость приема и передачи данных – от 5 до 300 Мбит/с;
 - дуплексный режим работы;
 - возможность программной настройки скорости передачи данных;
 - аппаратное детектирование ошибок связи: разъединение, ошибки четности;
 - возможность программной адаптивной подстройки скорости;
 - обмен данными с внутренней памятью по каналам DMA 64-х разрядными словами;
 - встроенные LVDS-приемопередатчики в соответствии со стандартом ANSI/TIA/EIA-644;
 - согласующие резисторы, встроенные в LVDS приемники сопротивлением 100+-10% Ом;
- память данных объемом 512 КБайт с защитой данных кодом Хэмминга;
- контроллер Ethernet 10/100 МГц;
- тестовый порт JTAG.

2. СИСТЕМНАЯ ОРГАНИЗАЦИЯ МИКРОСХЕМЫ

2.1 Структурная схема

Структурная схема микросхемы 1892ХД4Ф приведена на Рисунок 2.1.

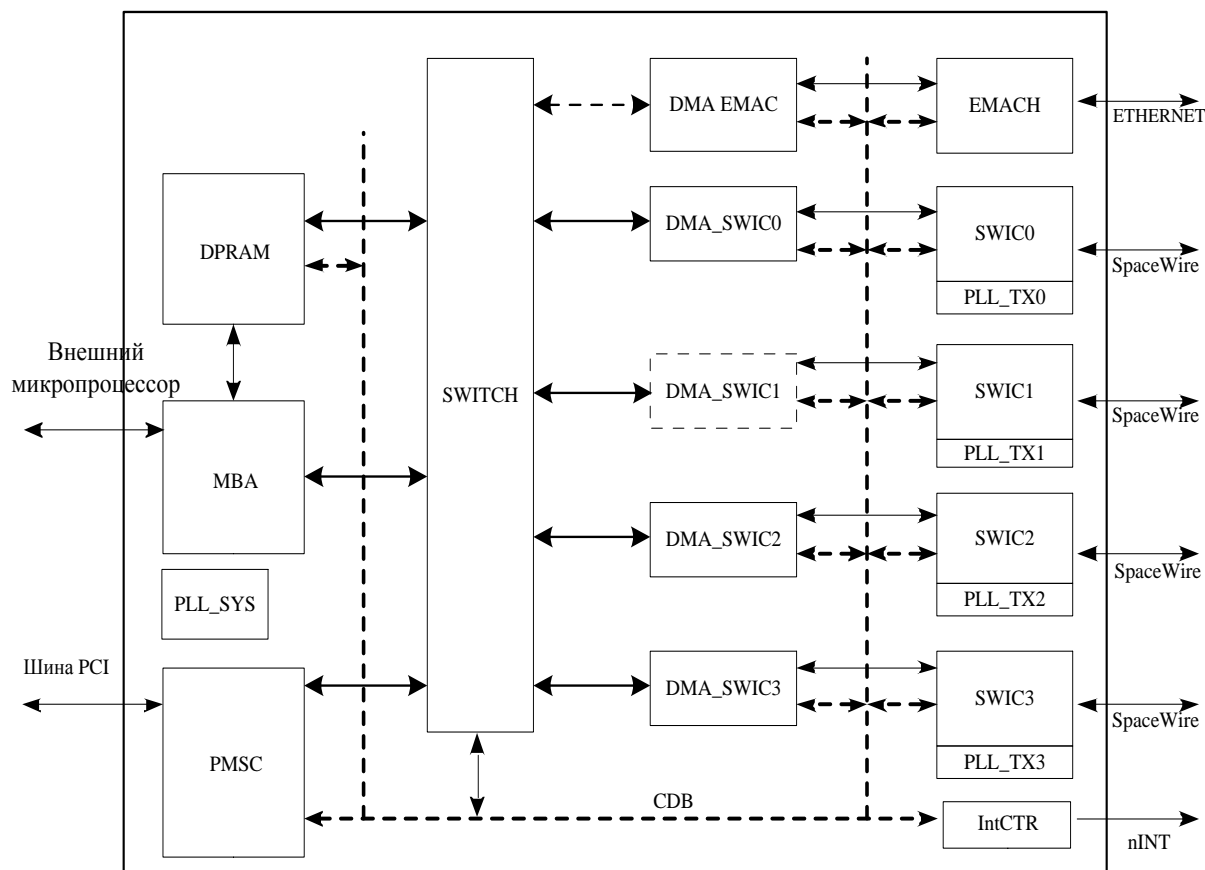


Рисунок 2.1. Структурная схема микросхема 1892ХД4Ф

Микросхема 1892ХД4Ф содержит следующие основные узлы:

- MBA - (Microprocessor Bus Adapter) - адаптер сопряжения с шинами адреса и данных внешнего микропроцессора;
- PMSC – (PCI Master/Slave Controller) - контроллер шины PCI;
- SWITCH - коммутатор данных. Используется для обмена данными DMA_SWIC <-> DPRAM, DMA_ETH <-> DPRAM и PMSC <-> DPRAM, а также для доступа к шине CDB от блоков MBA и PMSC;
- SWIC – (SpaceWire Interface Controller) контроллер SpaceWire;
- ETH – контроллер Ethernet MAC 10/100;
- DMA_SWIC - контроллер DMA блоков SWIC;
- DMA_ETH - контроллер DMA блока ETH;

- PLL_TX – программируемый множитель частоты. Формирует частоту для передачи данных контроллерами SWIC;
- DPRAM - двух портовое оперативное запоминающее устройство с защитой данных кодом Хэмминга;
- CDB – (Control Data Bus) - шина управления. Используется для обмена данными с регистрами SWIC, DMA_SWIC, ETH, DMA_ETH, PMSC;
- IntCTR - контроллер прерываний;
- PLL_SYS - множитель частоты. Формирует системную тактовую частоту 200 МГц.

Обозначения, использованные на схеме:

- SpaceWire – канал связи SpaceWire;
- ETHERNET – канал связи Ethernet MAC 10/100.

Схема потоков данных и управления внутри микросхемы 1892ХД4Ф приведена на Рисунок 2.2.

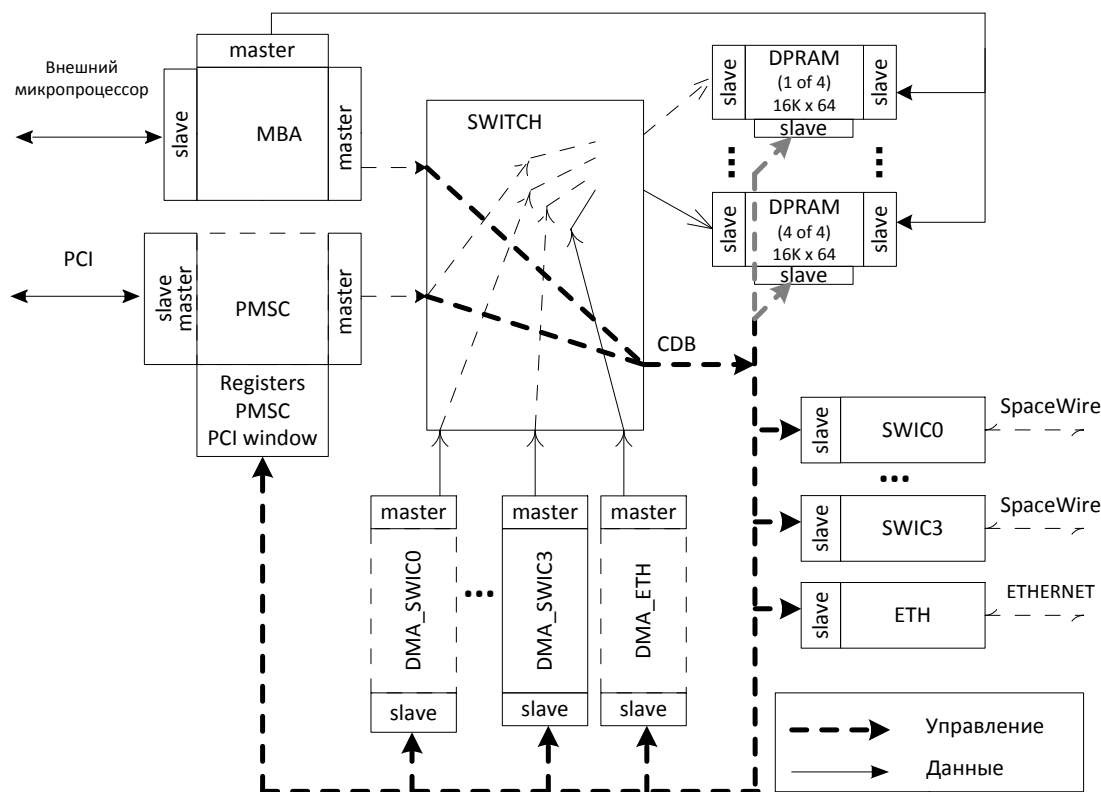


Рисунок 2.2. Схема потоков данных и управления

В микросхеме 1892ХД4Ф все инициаторы обмена (master) и исполнители (slave) подключены к коммутатору SWITCH. Устройство master не может обращаться к "себе" как устройству slave. При таких обменах данные при записи теряются, а при чтении недостоверны.

Коммутатор в соответствии картой памяти соединяет устройство master, с абонентами шины CDB или с DPRAM. Абонентами шины CDB являются регистры управления блоков SWIC, ETH, DMA_SWIC, DMA_ETH, DPRAM и PMSC. Доступ к DPRAM от внешнего микропроцессора организован через отдельный порт блока MBA.

2.2 Схема синхронизации

Схема синхронизации микросхемы 1892ХД4Ф приведена на Рисунок 2.3.

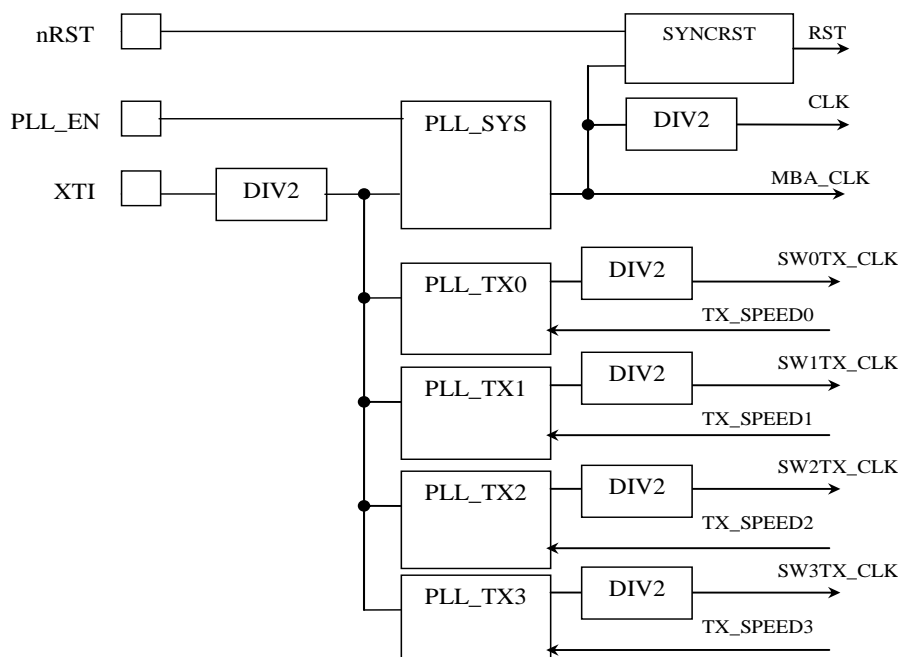


Рисунок 2.3. Схема синхронизации микросхемы 1892ХД4Ф

Синхронизация микросхемы 1892ХД4Ф осуществляется от внешнего источника синхросигнала, подключенного к входу XT1. Частота синхронизации должна быть 10+-1% МГц.

Частота с вывода XT1 через делитель на 2 поступает на умножитель частоты PLL_SYS, на выходе которого формируется системная частота MBA_CLK величиной 200 МГц. Разрешение умножения входной частоты производится высоким уровнем сигнала PLL_EN (рабочий режим). При PLL_EN=0 (тестовый режим) выходная частота PLL_SYS равна входной частоте. Частота MBA_CLK используется для тактирования MBA и DPRAM. На делителе DIV2 производится деление частоты MBA_CLK на два и формируется частота CLK, используемая для тактирования SWITCH, SWIC, DMA SWIC и PMSC, при обменах с шиной CDB и коммутатором данных.

Поделенная пополам частота со входа XT1 поступает также на входы умножителей частоты PLL_TX0 : PLL_TX3. Каждый SWIC имеет свою PLL_TX. Управление коэффициентом умножения осуществляется программно через регистры TX_SPEED.

Частота, формируемая умножителями PLL_TX, делится на 2 и используется для синхронизации передающего тракта данных, соответствующего SWIC.

Входной сигнал установки исходного состояния nRST стробируется в блоке SYNCRST частотой MBA_CLK. Длительность сигнала nRST – не менее 20 нс.

2.3 Карта памяти

Карта памяти микросхемы 1892ХД4Ф при обмене данными через адаптер MBA приведена в Таблица 2.1.

Таблица 2.1. Карта памяти микросхемы 1892ХД4Ф при обмене данными через адаптер MBA

Диапазон адресов	Название области
1C0_0000 - 1DF_FFFC	Регистры MBA
1B0_0000 - 1BF_FFFC	Регистры DMA_SWIC3
1A0_0000 - 1AF_FFFC	Регистры SWIC3
190_0000 - 19F_FFFC	Регистры DMA_SWIC2
180_0000 - 18F_FFFC	Регистры SWIC2
170_0000 - 17F_FFFC	Регистры DMA_SWIC1
160_0000 - 16F_FFFC	Регистры SWIC1
150_0000 - 15F_FFFC	Регистры DMA_SWIC0
140_0000 - 14F_FFFC	Регистры SWIC0
138_0000 - 13F_FFFC	Регистры DMA_Ethernet MAC
130_0000 - 137_FFFC	Регистры Ethernet MAC
120_0000 - 12F_FFFC	Регистры PMSC
118_0004	Регистр отключения и включения частоты CLK_EN
110_0000 - 117_FFFC	Регистры Hm_DPRAM
108_0000 - 10F_FFFC	Резерв
106_0000 - 107_FFFC	Память DPRAM3
104_0000 - 105_FFFC	Память DPRAM2
102_0000 - 103_FFFC	Память DPRAM1
100_0000 - 101_FFFC	Память DPRAM0
000_0000 - 0FF_FFFC	Окно шины PCI

Все адресное пространство микросхемы 1892ХД4Ф занимает 32 Мбайт. При обмене данными через адаптер MBA внутренний адрес микросхемы 1892ХД4Ф формируется добавлением к разрядам A[22:0] двух нулей справа. Карта памяти микросхемы 1892ХД4Ф при обмене данными через шину PCI приведена в Таблица 2.2.

Таблица 2.2. Карта памяти микросхемы 1892ХД4Ф при обмене данными через шину PCI

Диапазон адресов	Название области
01B0_0000 - 01BF_FFFC	Регистры DMA_SWIC3
01A0_0000 - 01AF_FFFC	Регистры SWIC3
0190_0000 - 019F_FFFC	Регистры DMA_SWIC2
0180_0000 - 018F_FFFC	Регистры SWIC2
0170_0000 - 017F_FFFC	Регистры DMA_SWIC1
0160_0000 - 016F_FFFC	Регистры SWIC1
0150_0000 - 015F_FFFC	Регистры DMA_SWIC0
0140_0000 - 014F_FFFC	Регистры SWIC0
0138_0000 - 013F_FFFC	Регистры DMA_Ethernet MAC

Диапазон адресов	Название области
0130_0000 - 0137_FFFC	Регистры Ethernet MAC
0120_0000 - 012F_FFFC	Регистры PCI
118_0004	Регистр отключения и включения частоты CLK_EN
0110_0000 - 011F_FFFC	Регистры Hm DPRAM
0108_0000 - 010F_FFFC	Резерв
0106_0000 - 0107_FFFC	Память DPRAM3
0104_0000 - 0105_FFFC	Память DPRAM2
0102_0000 - 0103_FFFC	Память DPRAM1
0100_0000 - 0101_FFFC	Память DPRAM0
0030_0000 - 00FF_FFFC	Резерв
0020_0000 - 002F_FFFC	Регистры PCI
0000_0000 - 001F_FFFC	Резерв

Перечень программно-доступных регистров микросхемы 1892ХД4Ф приведен в Таблица 2.3. Адреса регистров указаны для случая обмена данными через адаптер МВА. При обмене данными с микросхемой 1892ХД4Ф через шину PCI базовые адреса регистров необходимо определять по Таблица 2.2.

Таблица 2.3. Перечень программно доступных регистров микросхемы 1892ХД4Ф

Условное обозначение регистра	Назначение регистра	Адрес регистра
Регистры МВА		
QSTR0	Регистр #0 запросов прерывания	1C0_0000
MASKR0	Регистр #0 маски запросов прерывания	1C0_0004
BDR	Регистр буферных данных	1C0_0008
BUSY	Регистр признака занятости	1C0_000C
QSTR1	Регистр #1 запросов прерывания	1C0_0010
MASKR1	Регистр #1 маски запросов прерывания	1C0_0014
Регистры контроля по коду Хэмминга памяти DPRAM		
CSR0	Регистр управления и состояния DPRAM0	110_0000
AERROR0	Регистр ошибочных адресов DPRAM0	110_0004
CSR1	Регистр управления и состояния DPRAM1	110_1000
AERROR1	Регистр ошибочных адресов DPRAM1	110_1004
CSR2	Регистр управления и состояния DPRAM2	110_2000
AERROR2	Регистр ошибочных адресов DPRAM2	110_2004
CSR3	Регистр управления и состояния DPRAM3	110_3000
AERROR3	Регистр ошибочных адресов DPRAM3	110_3004
Регистры Ethernet MAC		
MAC_CONTROL	Регистр управления MAC	130_0000
MAC_ADDR_L	Регистр младшей части исходного адреса MAC	130_0004
MAC_ADDR_H	Регистр старшей части исходного адреса MAC	130_0008
DADDR_L	Регистр младшей части адреса назначения	130_000C
DADDR_H	Регистр старшей части адреса назначения	130_0010
FCS_CLIENT	Регистр контрольной суммы кадра	130_0014
TYPE	Регистр типа кадра	130_0018
IFS_COLL_MODE	Регистр IFS и режима обработки коллизии	130_001C
TX_FRAME_CONTROL	Регистр управления передачи кадра	130_0020

Условное обозначение регистра	Назначение регистра	Адрес регистра
STATUS_TX	Регистр статуса передачи кадра	130_0024
UCADDR_L	Регистр младшей части уникального адреса MAC	130_0028
UCADDR_H	Регистр старшей части уникального адреса MAC	130_002C
MCADDR_L	Регистр младшей части группового адреса	130_0030
MCADDR_H	Регистр старшей части группового адреса	130_0034
MCADDR_MASK_L	Регистр младшей части маски группового адреса	130_0038
MCADDR_MASK_H	Регистр старшей части маски группового адреса	130_003C
HASHT_L	Регистр младшей части хэш-таблицы	130_0040
HASHT_H	Регистр старшей части хэш-таблицы	130_0044
RX_FRAME_CONTROL	Регистр управления приема кадра	130_0048
RX_FR_MaxSize	Регистр максимального размера принимаемого кадра	130_004C
STATUS_RX	Регистр статуса приема кадра	130_0050
RX_FRAME_STATUS_FIFO	FIFO статусов принятых кадров	130_0054
MD_CONTROL	Регистр управления порта MD	130_0058
MD_STATUS	Регистр статуса порта MD	130_005C
MD_MODE	Регистр режима работы порта MD	130_0060
TX_TEST_CSR	Регистр управления и состояния режима тестирования TX_FIFO	130_0064
TX_FIFO	Передающее TX_FIFO	130_0068
RX_TEST_CSR	Регистр управления и состояния режима тестирования RX_FIFO	130_006C
RX_FIFO	Принимающее RX_FIFO	130_0070
Регистры DMA EMAC_CH0		
CSR_EMAC_CH0	Регистр управления и состояния. При чтении: сброс битов END и DONE	138_0000
CP_EMAC_CH0	Регистр указателя цепочки	138_0004
IR_EMAC_CH0	Регистр индекса	138_0008
RUN_EMAC_CH0	При записи: псевдорегистр управления состоянием бита RUN регистра CSR. При чтении: регистр управления и состояния CSR без сброса битов "END" и "DONE"	138_000C
Регистры DMA EMAC_CH1		
CSR_EMAC_CH1	Регистр управления и состояния. При чтении: сброс битов END и DONE	138_0040
CP_EMAC_CH1	Регистр указателя цепочки	138_0044
IR_EMAC_CH1	Регистр индекса	138_0048
RUN_EMAC_CH1	При записи: псевдорегистр управления состоянием бита RUN регистра CSR. При чтении: регистр управления и состояния CSR без сброса битов "END" и "DONE"	138_004C
Регистры PMSC		
Device/ Vendor ID	Регистр идентификации устройства. Конфигурационный регистр шины PCI.	120_0000
Status/ Command	Регистр состояния и управления. Конфигурационный регистр шины PCI.	120_0004
Class Code/ Revision ID	Регистр кода. Конфигурационный регистр шины PCI.	120_0008
Latency Timer	Регистр таймера времени передачи (MLT). Конфигурационный регистр шины PCI.	120_000C
BAR	(Base Address Register) Регистр базового адреса. Конфигурационный регистр шины PCI.	120_0010

Условное обозначение регистра	Назначение регистра	Адрес регистра
Interrupt_Line	Код прерывания. Конфигурационный регистр шины PCI.	120_003C
SEM	Регистр семафора.	120_0044
MBR	Регистр почтового ящика	120_0048
CSR_PCI	Регистр управления шины PCI	120_004C
CSR_MASTER	Регистр состояния и управления обменом с PCI в режиме Master.	120_0050
IR_MASTER	Индексный регистр адреса памяти при обмене данными с PCI в режиме Master.	120_0054
AR_PCI	Адресный регистр PCI.	120_0058
QSTR0_PCI	Регистр системных прерываний.	120_005C
MASKR0_PCI	Регистр маски системных прерываний.	120_0060
QSTR1_PCI	Регистр системных прерываний.	120_007C
MASKR1_PCI	Регистр маски системных прерываний.	120_0080
Регистры контроллера SWIC0		
HW_VER	Регистр версии контроллера	140_0000
STATUS	Регистр состояния	140_0004
RX_CODE	Регистр управляющего символа, принятого из сети (маркера времени, кода распределенного прерывания, кода подтверждения распределенного прерывания или кода CC11)	140_0008
MODE_CR	Регистр управления режимом работы	140_000C
TX_SPEED	Регистр управления скоростью передачи	140_0010
TX_CODE	Регистр управляющего символа (маркера времени, кода распределенного прерывания, кода подтверждения, кода CC11) для передачи в сеть	140_0014
RX_SPEED	Регистр измерителя скорости приема данных из канала SpaceWire	140_0018
CNT_RX_PACK	Регистр счетчика принятых пакетов ненулевой длины	140_0020
ISR_L	Регистр кодов распределенных прерываний (LSB)	140_0024
ISR_H	Регистр кодов распределенных прерываний (MSB)	140_0028
TRUE_TIME	Регистр достоверного маркера времени	140_002C
TOUT_CODE	Регистр размера таймаутов	140_0030
ISR_tout_L	Младшие разряды регистра флагов таймаутов ISR	140_0034
ISR_tout_H	Старшие разряды регистра флагов таймаутов ISR	140_0038
LOG_ADDR	Регистр логического адреса	140_003C
ACK_NONACK_MODE	Регистр управления режимом распределенных прерываний (с подтверждениями или без подтверждений)	140_0040
ISR2_tout	Регистр таймаутов кодов распределенных прерываний 2	140_0044
ISR_HANDLER_FUNC	Регистр флагов функций терминального узла обработчика	140_0048
ISR_SPEC	Регистр рассылки управляющих кодов в специальный набор портов	140_004C
ISR_1101	Регистр флагов приема управляющих кодов, назначение которых не определено в стандарте ECSS-E-50-12C	140_0050
ISR_MASK_1101	Регистр маски портов, из которых не должны приниматься управляющие коды, назначение которых не определено в текущей версии стандарта	140_0054
INT_RESET	Регистр параметров команды внешнего сброса	140_0058
MODE_CR2	Регистр режима работы 2	140_0060

Условное обозначение регистра	Назначение регистра	Адрес регистра
INT_H_MASK	Старшая половина регистра маски распределенных прерываний	140_0064
INT_L_MASK	Младшая половина регистра маски распределенных прерываний	140_0068
ACK_H_MASK	Старшая половина регистра маски кодов подтверждения	140_006C
ACK_L_MASK	Младшая половина регистра маски кодов подтверждения	140_0070
AUTO_SPEED_MANAGE	Регистр параметров автоматической установки скорости передачи	140_0074
ISR_SOURCE_TERM_FUNC	Регистр флагов функций терминального узла источника	140_0078
ISR_SPEC_TERM_FUNC	Регистр признака специальной функции для терминального узла обработчика	140_007C
ISR_L_RESET	Младшая половина регистра глобального сброса ISR	140_0080
ISR_H_RESET	Старшая половина регистра глобального сброса ISR	140_0084
Регистры контроллера SWIC1		
HW_VER	Регистр версии контроллера	160_0000
STATUS	Регистр состояния	160_0004
RX_CODE	Регистр управляющего символа, принятого из сети (маркера времени, кода распределенного прерывания, кода подтверждения распределенного прерывания или кода CC11)	160_0008
MODE_CR	Регистр управления режимом работы	160_000C
TX_SPEED	Регистр управления скоростью передачи	160_0010
TX_CODE	Регистр управляющего символа (маркера времени, кода распределенного прерывания, кода подтверждения, кода CC11) для передачи в сеть	160_0014
RX_SPEED	Регистр измерителя скорости приема данных из канала SpaceWire	160_0018
CNT_RX_PACK	Регистр счетчика принятых пакетов ненулевой длины	160_0020
ISR_L	Регистр кодов распределенных прерываний (LSB)	160_0024
ISR_H	Регистр кодов распределенных прерываний (MSB)	160_0028
TRUE_TIME	Регистр достоверного маркера времени	160_002C
TOUT_CODE	Регистр размера таймаутов	160_0030
ISR_tout_L	Младшие разряды регистра флагов таймаутов ISR	160_0034
ISR_tout_H	Старшие разряды регистра флагов таймаутов ISR	160_0038
LOG_ADDR	Регистр логического адреса	160_003C
ACK_NONACK_MODE	Регистр управления режимом распределенных прерываний (с подтверждениями или без подтверждений)	160_0040
ISR2_tout	Регистр таймаутов кодов распределенных прерываний 2	160_0044
ISR_HANDLER_TERM_FUNC	Регистр флагов функций терминального узла обработчика	160_0048
ISR_SPEC	Регистр рассылки управляющих кодов в специальный набор портов	160_004C
ISR_1101	Регистр флагов приема управляющих кодов, назначение которых не определено в текущей версии стандарта	160_0050
ISR_MASK_1101	Регистр маски портов, из которых не должны приниматься управляющие коды, назначение которых не определено в стандарте ECSS-E-50-12C	160_0054
INT_RESET	Регистр параметров команды внешнего сброса	160_0058

Условное обозначение регистра	Назначение регистра	Адрес регистра
MODE_CR2	Регистр режима работы 2	160_0060
INT_H_MASK	Старшая половина регистра маски распределенных прерываний	160_0064
INT_L_MASK	Младшая половина регистра маски распределенных прерываний	160_0068
ACK_H_MASK	Старшая половина регистра маски кодов подтверждения	160_006C
ACK_L_MASK	Младшая половина регистра маски кодов подтверждения	160_0070
AUTO_SPEED_MANAGE	Регистр параметров автоматической установки скорости передачи	160_0074
ISR_SOURCE_TERM_FUNC	Регистр флагов функций терминального узла источника	160_0078
ISR_SPEC_TERM_FUNC	Регистр признака специальной функции для терминального узла обработчика	160_007C
ISR_L_RESET	Младшая половина регистра глобального сброса ISR	160_0080
ISR_H_RESET	Старшая половина регистра глобального сброса ISR	160_0084
Регистры контроллера SWIC2		
HW_VER	Регистр версии контроллера	180_0000
STATUS	Регистр состояния	180_0004
RX_CODE	Регистр управляющего символа, принятого из сети (маркера времени, кода распределенного прерывания, кода подтверждения распределенного прерывания или кода CC11)	180_0008
MODE_CR	Регистр управления режимом работы	180_000C
TX_SPEED	Регистр управления скоростью передачи	180_0010
TX_CODE	Регистр управляющего символа (маркера времени, кода распределенного прерывания, кода подтверждения, кода CC11) для передачи в сеть	180_0014
RX_SPEED	Регистр измерителя скорости приема данных из канала SpaceWire	180_0018
CNT_RX_PACK	Регистр счетчика принятых пакетов ненулевой длины	180_0020
ISR_L	Регистр кодов распределенных прерываний (LSB)	180_0024
ISR_H	Регистр кодов распределенных прерываний (MSB)	180_0028
TRUE_TIME	Регистр достоверного маркера времени	180_002C
TOUT_CODE	Регистр размера таймаутов	180_0030
ISR_tout_L	Младшие разряды регистра флагов таймаутов ISR	180_0034
ISR_tout_H	Старшие разряды регистра флагов таймаутов ISR	180_0038
LOG_ADDR	Регистр логического адреса	180_003C
ACK_NONACK_MODE	Регистр управления режимом распределенных прерываний (с подтверждениями или без подтверждений)	180_0040
ISR2_tout	Регистр таймаутов кодов распределенных прерываний 2	180_0044
ISR_HANDLER_TERM_FUNC	Регистр флагов функций терминального узла обработчика	180_0048
ISR_SPEC	Регистр рассылки управляющих кодов в специальный набор портов	180_004C
ISR_1101	Регистр флагов приема управляющих кодов, назначение которых не определено в стандарте ECSS-E-50-12C	180_0050
ISR_MASK_1101	Регистр маски портов, из которых не должны приниматься управляющие коды, назначение которых не определено в текущей версии стандарта	180_0054

Условное обозначение регистра	Назначение регистра	Адрес регистра
INT_RESET	Регистр параметров команды внешнего сброса	180_0058
MODE_CR2	Регистр режима работы 2	180_0060
INT_H_MASK	Старшая половина регистра маски распределенных прерываний	180_0064
INT_L_MASK	Младшая половина регистра маски распределенных прерываний	180_0068
ACK_H_MASK	Старшая половина регистра маски кодов подтверждения	180_006C
ACK_L_MASK	Младшая половина регистра маски кодов подтверждения	180_0070
AUTO_SPEED_MANAGE	Регистр параметров автоматической установки скорости передачи	180_0074
ISR_SOURCE_TERM_FUNC	Регистр флагов функций терминального узла источника	180_0078
ISR_SPEC_TERM_FUNC	Регистр признака специальной функции для терминального узла обработчика	180_007C
ISR_L_RESET	Младшая половина регистра глобального сброса ISR	180_0080
ISR_H_RESET	Старшая половина регистра глобального сброса ISR	180_0084
Регистры контроллера SWIC3		
HW_VER	Регистр версии контроллера	1A0_0000
STATUS	Регистр состояния	1A0_0004
RX_CODE	Регистр управляющего символа, принятого из сети (маркера времени, кода распределенного прерывания, кода подтверждения распределенного прерывания или кода CC11)	1A0_0008
MODE_CR	Регистр управления режимом работы	1A0_000C
TX_SPEED	Регистр управления скоростью передачи	1A0_0010
TX_CODE	Регистр управляющего символа (маркера времени, кода распределенного прерывания, кода подтверждения, кода CC11) для передачи в сеть	1A0_0014
RX_SPEED	Регистр измерителя скорости приема данных из канала SpaceWire	1A0_0018
CNT_RX_PACK	Регистр счетчика принятых пакетов ненулевой длины	1A0_0020
ISR_L	Регистр кодов распределенных прерываний (LSB)	1A0_0024
ISR_H	Регистр кодов распределенных прерываний (MSB)	1A0_0028
TRUE_TIME	Регистр достоверного маркера времени	1A0_002C
TOUT_CODE	Регистр размера таймаутов	1A0_0030
ISR_tout_L	Младшие разряды регистра флагов таймаутов ISR	1A0_0034
ISR_tout_H	Старшие разряды регистра флагов таймаутов ISR	1A0_0038
LOG_ADDR	Регистр логического адреса	1A0_003C
ACK_NONACK_MODE	Регистр управления режимом распределенных прерываний (с подтверждениями или без подтверждений)	1A0_0040
ISR2_tout	Регистр таймаутов кодов распределенных прерываний 2	1A0_0044
ISR_HANDLER_TERM_FUNC	Регистр флагов функций терминального узла обработчика	1A0_0048
ISR_SPEC	Регистр рассылки управляющих кодов в специальный набор портов	1A0_004C
ISR_1101	Регистр флагов приема управляющих кодов, назначение которых не определено в текущей версии стандарта	1A0_0050
ISR_MASK_1101	Регистр маски портов, из которых не должны приниматься управляющие коды, назначение которых не определено в стандарте ECSS-E-50-12C	1A0_0054
INT_RESET	Регистр параметров команды внешнего сброса	1A0_0058

Условное обозначение регистра	Назначение регистра	Адрес регистра
MODE_CR2	Регистр режима работы 2	1A0_0060
INT_H_MASK	Старшая половина регистра маски распределенных прерываний	1A0_0064
INT_L_MASK	Младшая половина регистра маски распределенных прерываний	1A0_0068
ACK_H_MASK	Старшая половина регистра маски кодов подтверждения	1A0_006C
ACK_L_MASK	Младшая половина регистра маски кодов подтверждения	1A0_0070
AUTO_SPEED_MANAGE	Регистр параметров автоматической установки скорости передачи	1A0_0074
ISR_SOURCE_TERM_FUNC	Регистр флагов функций терминального узла источника	1A0_0078
ISR_SPEC_TERM_FUNC	Регистр признака специальной функции для терминального узла обработчика	1A0_007C
ISR_L_RESET	Младшая половина регистра глобального сброса ISR	1A0_0080
ISR_H_RESET	Старшая половина регистра глобального сброса ISR	1A0_0084
Регистры DMA SWIC0		
Канал записи в память дескрипторов принимаемых пакетов		
CSR_SWIC_RX_DES0	Регистр управления и состояния	15_0000
CP_SWIC_RX_DES0	Регистр указателя цепочки	15_0004
IR_SWIC_RX_DES0	Регистр индекса	15_0008
RUN_SWIC_RX_DES0	Псевдорегистр управления состоянием бита RUN регистра CSR	15_000C
Канал записи в память принимаемых слов данных		
CSR_SWIC_RX_DAT0	Регистр управления и состояния	15_0040
CP_SWIC_RX_DAT0	Регистр указателя цепочки	15_0044
IR_SWIC_RX_DAT0	Регистр индекса	15_0048
RUN_SWIC_RX_DAT0	Псевдорегистр управления состоянием бита RUN регистра CSR	15_004C
Канал чтения из памяти дескрипторов передаваемых пакетов		
CSR_SWIC_TX_DES0	Регистр управления и состояния	15_0080
CP_SWIC_TX_DES0	Регистр указателя цепочки	15_0084
IR_SWIC_TX_DES0	Регистр индекса	15_0088
RUN_SWIC_TX_DES0	Псевдорегистр управления состоянием бита RUN регистра CSR	15_008C
Канал чтения из памяти передаваемых слов данных		
CSR_SWIC_TX_DAT0	Регистр управления и состояния	15_00C0
CP_SWIC_TX_DAT0	Регистр указателя цепочки	15_00C4
IR_SWIC_TX_DAT0	Регистр индекса	15_00C8
RUN_SWIC_TX_DAT0	Псевдорегистр управления состоянием бита RUN регистра CSR	15_00CC
Регистры DMA SWIC1		
Канал записи в память дескрипторов принимаемых пакетов		
CSR_SWIC_RX_DES1	Регистр управления и состояния	17_0000
CP_SWIC_RX_DES1	Регистр указателя цепочки	17_0004
IR_SWIC_RX_DES1	Регистр индекса	17_0008
RUN_SWIC_RX_DES1	Псевдорегистр управления состоянием бита RUN регистра CSR	17_000C
Канал записи в память принимаемых слов данных		
CSR_SWIC_RX_DAT1	Регистр управления и состояния	17_0040
CP_SWIC_RX_DAT1	Регистр указателя цепочки	17_0044
IR_SWIC_RX_DAT1	Регистр индекса	17_0048

Условное обозначение регистра	Назначение регистра	Адрес регистра
RUN_SWIC_RX_DAT1	Псевдорегистр управления состоянием бита RUN регистра CSR	17_004C
Канал чтения из памяти дескрипторов передаваемых пакетов		
CSR_SWIC_TX_DES1	Регистр управления и состояния	17_0080
CP_SWIC_TX_DES1	Регистр указателя цепочки	17_0084
IR_SWIC_TX_DES1	Регистр индекса	17_0088
RUN_SWIC_TX_DES1	Псевдорегистр управления состоянием бита RUN регистра CSR	17_008C
Канал чтения из памяти передаваемых слов данных		
CSR_SWIC_TX_DAT1	Регистр управления и состояния	17_00C0
CP_SWIC_TX_DAT1	Регистр указателя цепочки	17_00C4
IR_SWIC_TX_DAT1	Регистр индекса	17_00C8
RUN_SWIC_TX_DAT1	Псевдорегистр управления состоянием бита RUN регистра CSR	17_00CC
Регистры DMA SWIC2		
Канал записи в память дескрипторов принимаемых пакетов		
CSR_SWIC_RX_DES2	Регистр управления и состояния	19_0000
CP_SWIC_RX_DES2	Регистр указателя цепочки	19_0004
IR_SWIC_RX_DES2	Регистр индекса	19_0008
RUN_SWIC_RX_DES2	Псевдорегистр управления состоянием бита RUN регистра CSR	19_000C
Канал записи в память принимаемых слов данных		
CSR_SWIC_RX_DAT2	Регистр управления и состояния	19_0040
CP_SWIC_RX_DAT2	Регистр указателя цепочки	19_0044
IR_SWIC_RX_DAT2	Регистр индекса	19_0048
RUN_SWIC_RX_DAT2	Псевдорегистр управления состоянием бита RUN регистра CSR	19_004C
Канал чтения из памяти дескрипторов передаваемых пакетов		
CSR_SWIC_TX_DES2	Регистр управления и состояния	19_0080
CP_SWIC_TX_DES2	Регистр указателя цепочки	19_0084
IR_SWIC_TX_DES2	Регистр индекса	19_0088
RUN_SWIC_TX_DES2	Псевдорегистр управления состоянием бита RUN регистра CSR	19_008C
Канал чтения из памяти передаваемых слов данных		
CSR_SWIC_TX_DAT2	Регистр управления и состояния	19_00C0
CP_SWIC_TX_DAT2	Регистр указателя цепочки	19_00C4
IR_SWIC_TX_DAT2	Регистр индекса	19_00C8
RUN_SWIC_TX_DAT2	Псевдорегистр управления состоянием бита RUN регистра CSR	19_00CC
Регистры DMA SWIC3		
Канал записи в память дескрипторов принимаемых пакетов		
CSR_SWIC_RX_DES3	Регистр управления и состояния	1B_0000
CP_SWIC_RX_DES3	Регистр указателя цепочки	1B_0004
IR_SWIC_RX_DES3	Регистр индекса	1B_0008
RUN_SWIC_RX_DES3	Псевдорегистр управления состоянием бита RUN регистра CSR	1B_000C
Канал записи в память принимаемых слов данных		
CSR_SWIC_RX_DAT3	Регистр управления и состояния	1B_0040

Условное обозначение регистра	Назначение регистра	Адрес регистра
CP_SWIC_RX_DAT3	Регистр указателя цепочки	1B_0044
IR_SWIC_RX_DAT3	Регистр индекса	1B_0048
RUN_SWIC_RX_DAT3	Псевдорегистр управления состоянием бита RUN регистра CSR	1B_004C
Канал чтения из памяти дескрипторов передаваемых пакетов		
CSR_SWIC_TX_DES3	Регистр управления и состояния	1B_0080
CP_SWIC_TX_DES3	Регистр указателя цепочки	1B_0084
IR_SWIC_TX_DES3	Регистр индекса	1B_0088
RUN_SWIC_TX_DES3	Псевдорегистр управления состоянием бита RUN регистра CSR	1B_008C
Канал чтения из памяти передаваемых слов данных		
CSR_SWIC_TX_DAT3	Регистр управления и состояния	1B_00C0
CP_SWIC_TX_DAT3	Регистр указателя цепочки	1B_00C4
IR_SWIC_TX_DAT3	Регистр индекса	1B_00C8
RUN_SWIC_TX_DAT3	Псевдорегистр управления состоянием бита RUN регистра CSR	1B_00CC

2.4 Регистр CLK_EN

Отключение и включение тактовой частоты устройств выполняется при помощи регистра CLK_EN, формат которого приведен в Таблица 2.4.

Таблица 2.4. Формат регистра CLK_EN

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	Не используется	-	0
27:24	CLKEN_SWIC[3:0]	Управление включением тактовой частоты SWIC[3:0], поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена	R/W	1111
23:21	-	Не используется	-	0
20	CLKEN_EMAC	Управление включением тактовой частоты EMAC, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена	R/W	1
19:17	-	Не используется	-	0
16	CLKEN_PMSC	Управление включением тактовой частоты PMSC, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена. Можно отключать только при работе через MBA	R/W	1
15:2	-	Не используется	-	0
1	CLKEN_MBA	Управление включением тактовой частоты MBA и его порта DPRAM, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена. Можно отключать только при работе через PMSC	R/W	1
0	CLKEN_SWITCH	Управление включением тактовой частоты SWITCH, поступающей от PLL_CORE: 1 – частота включена; 0 – частота выключена. Можно отключать, если используется только обмен данными с DPRAM через MBA. SWIC, PMSC и EMAC не используются	R/W	1

3. АДАПТЕР СОПРЯЖЕНИЯ С ШИНОЙ ВНЕШНЕГО ПРОЦЕССОРА (МВА)

3.1 Назначение

Адаптер сопряжения с шиной данных внешнего микропроцессора (МВА) предназначен для обмена данными с регистрами SWIC, PMSC, памятью DPRAM, а также с шиной PCI в режиме Master через адресное окно. Используя доступ к регистрам PMSC через МВА можно запускать выполнения транзакций передачи данных по шине PCI в режиме Master (см. п. 4.5).

Обмен данными через МВА выполняется 32-разрядными словами. При записи данных в память DPRAM при помощи сигналов $nWE[3:0]$ можно записать от 1 до 4 байт 32-разрядного слова.

Следует иметь в виду, что по шине $A[24:0]$ передается адрес 32-разрядного слова. Поэтому при сопряжении микросхемы 1892ХД4Ф с микропроцессорами серии «Мультикор» эту шину необходимо подключить к шине адреса микропроцессора, начиная со 2 разряда.

3.2 Регистры МВА

3.2.1 Регистры запросов прерывания

Формат регистра QSTR0 приведен в Таблица 3.1. .

Таблица 3.1. Назначение разрядов регистра QSTR0

Номер разряда	Условное обозначение	Назначение
31:30	-	Не используется
Прерывания от PMSC		
29	MASTER_INT	Завершение выполнения транзакции передачи данных с шиной PCI в режиме Master
28	MBR_INT	Признак записи данных в регистр почтового ящика MBR контроллера PMSC со стороны шины PCI. Данный бит обнуляется при чтении содержимого регистра MBR
Прерывания от DMA SWIC3		
27	SWIC3_TX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC3_TX_DATA
26	SWIC3_TX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC3_TX_DESC
25	SWIC3_RX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC3_RX_DATA
24	SWIC3_RX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC3_RX_DESC
Прерывания от DMA SWIC2		
23	SWIC2_TX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC2_TX_DATA
22	SWIC2_TX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC2_TX_DESC

Номер разряда	Условное обозначение	Назначение
21	SWIC2_RX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC2_RX_DATA
20	SWIC2_RX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC2_RX_DESC
Прерывания от DMA SWIC1		
19	SWIC1_TX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC1_TX_DATA
18	SWIC1_TX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC1_TX_DESC
17	SWIC1_RX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC1_RX_DATA
16	SWIC1_RX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC1_RX_DESC
Прерывания от DMA SWIC0		
15	SWIC0_TX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC0_TX_DATA
14	SWIC0_TX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC0_TX_DESC
13	SWIC0_RX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC0_RX_DATA
12	SWIC0_RX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC0_RX_DESC
Прерывания от SWIC3		
11	SWIC3_TIME	Получен управляющий код
10	SWIC3_ERR	Ошибка в канале приема
9	SWIC3_LINK	Установлено соединение
Прерывания от SWIC2		
8	SWIC2_TIME	Получен управляющий код
7	SWIC2_ERR	Ошибка в канале приема
6	SWIC2_LINK	Установлено соединение
Прерывания от SWIC1		
5	SWIC1_TIME	Получен управляющий код
4	SWIC1_ERR	Ошибка в канале приема
3	SWIC1_LINK	Установлено соединение
Прерывания от SWIC0		
2	SWIC0_TIME	Получен управляющий код
1	SWIC0_ERR	Ошибка в канале приема
0	SWIC0_LINK	Установлено соединение

Формат регистра QSTR1 приведен в Таблица 3.2.

Таблица 3.2. Назначение разрядов регистра QSTR1

Номер разряда	Условное обозначение	Назначение
31:9	-	Не используется
Прерывания от контроллеров памяти		
7	INT_HmDPRAM3	Прерывание по коду Хемминга от DPRAM.3
6	INT_HmDPRAM2	Прерывание по коду Хемминга от DPRAM.2
5	INT_HmDPRAM1	Прерывание по коду Хемминга от DPRAM.1
4	INT_HmDPRAM0	Прерывание по коду Хемминга от DPRAM.0
Прерывания от Ethernet MAC		
3	DMA_TX	Прерывание от DMA контроллера Ethernet MAC по завершению передачи данных
2	DMA_RX	Прерывание от DMA контроллера Ethernet MAC по завершению приема данных

Номер разряда	Условное обозначение	Назначение
1	EMAC_TX	Прерывание от контроллера Ethernet MAC по завершению попытки передачи пакета
0	EMAC_RX	Прерывание от контроллера Ethernet MAC по приему кадра или по переполнению входного FIFO

Все сигналы внутренних прерываний поступают на входы доступного только по чтению регистра QSTR. Исходное состояние QSTR – все нули (нет запросов прерывания). Все незамаскированные прерывания объединяются по «или» и поступают на внешний вывод nINT.

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 SWIC сбрасывается при записи единиц в соответствующие разряды регистра SWIC_STATUS.

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.

3.2.2 Регистры маски запросов прерывания

Каждое внутреннее прерывание микросхемы 1892XD4Ф маскируется при помощи доступного по записи и чтению регистров маски MASKR0 и MASKR1, формат которого аналогичен формату соответствующего регистра QSTR. Регистру QSTR0 соответствует MASKR0. Регистру QSTR1 соответствует MASKR1. В исходном состоянии регистры маскирования прерываний обнулены, таким образом все внутренние прерывания замаскированы.

3.2.3 Регистр буферизации адреса и данных BDR

32-разрядный регистр BDR предназначен для выполнения процедуры чтения данных из регистров SWIC, PMSC и адресного окна шины PCI.

Регистр BDR доступен по записи и чтению со стороны шины обмена данными с внешним микропроцессором. Разряды регистра BDR соответствуют разрядам шины данных D[31:0].

Исходное содержимое регистра BDR – все нули.

3.2.4 Регистр BUSY

Регистр BUSY предназначен для синхронизации обмена данными с регистрами SWIC, PMSC, а также с шиной PCI в режиме Master через адресное окно.

Формат регистра BUSY приведен в Таблица 3.3.

Таблица 3.3. Формат регистра BUSY

Номер разряда	Условное обозначение	Назначение	Доступ
31	ACK	Полярность сигнала nACK: 1 – высокий активный уровень; 0 – низкий активный уровень	RW
30:1	-	Не используется	R
0	BUSY	Признак занятости MBA при обмене данными с регистрами SWIC, PMSC или с шиной PCI в режиме Master: 0 – MBA не занят; 1 – MBA занят. Устанавливается в 1: при инициализации процедуры записи данных в регистры SWIC, PMSC или адресное окно шины PCI; при записи адреса в регистр BDR для инициализации процедуры чтения данных из регистров SWIC, PMSC или адресного окна шины PCI. Устанавливается в 0: при завершении процедуры записи данных в регистры SWIC, PMSC или адресное окно шины PCI; при завершении процедуры чтения данных из регистров SWIC, PMSC, адресного окна шины PCI и записи этих данных в регистр BDR	R

Регистр BUSY доступен только по чтению. Исходное состояние – 0.

3.3 Обмен данными через адаптер MBA

3.3.1 Общие положения

Через адаптер MBA доступны регистры SWIC, PMSC, память DPRAM, а также шина PCI в режиме Master через адресное окно.

Обмен данными с регистрами SWIC, PMSC можно выполнять двумя способами:

- с использованием признака занятости BUSY;
- с использованием сигнала nACK.

3.3.2 Обмен данными с использованием признака занятости BUSY

Обмен данными с регистрами МВА и памятью DPRAM выполняется за одну операцию записи или чтения.

Запись данных в регистры SWIC, PMSC и адресное окно шины PCI выполняется за 2 операции:

- операция чтения: опрос BUSY (ожидание BUSY=0);
- операция записи данных. Адрес операции определяется в соответствии с Таблица 2.1.

Чтение данных из регистров SWIC, PMSC и адресного окна шины PCI выполняется за 4 операции:

- операция чтения: опрос BUSY=0 (ожидание BUSY=0);
- операция записи адреса регистра (или окна шины PCI) в регистр BDR. После выполнения этой операции инициируется процедура чтения содержимого регистра (окна шины PCI). Результат выполнения процедуры записывается в регистр BDR;
- операция чтения: опрос BUSY=0 (ожидание BUSY=0);
- операция чтения содержимого регистра BDR.

Временная диаграмма операции записи данных в микросхему 1892ХД4Ф через МВА приведена на Рисунок 3.1.

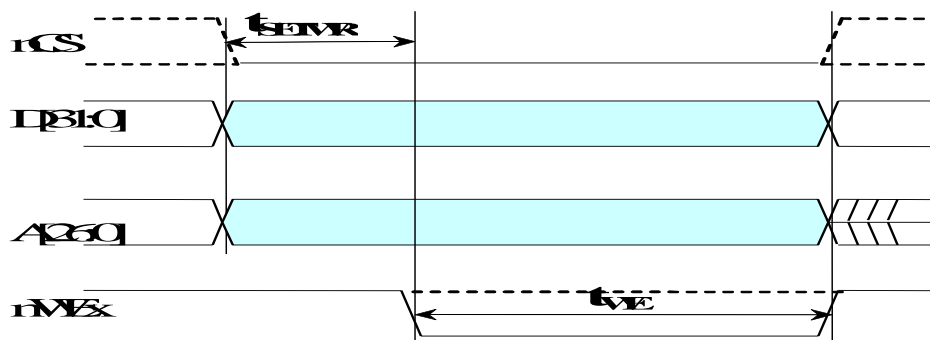


Рисунок 3.1. Временная диаграмма записи данных в микросхему 1892ХД4Ф

Временная диаграмма операции чтения данных из микросхемы 1892ХД4Ф через МВА приведена на Рисунок 3.2.

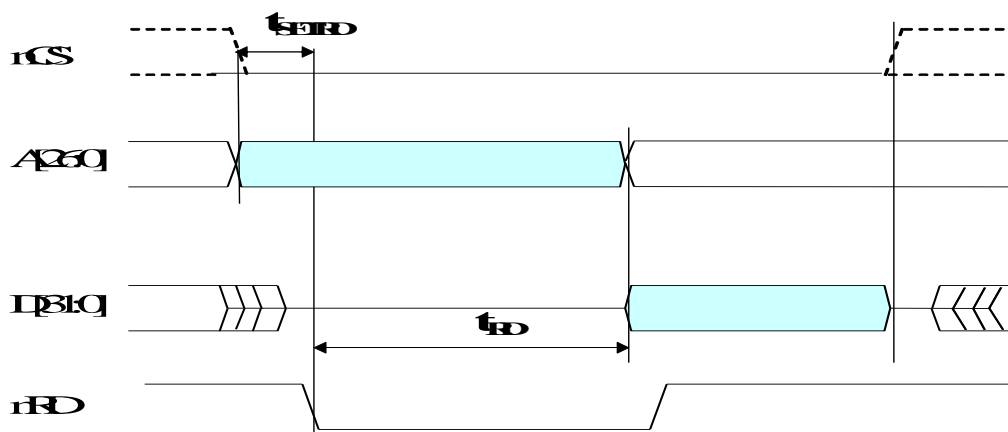


Рисунок 3.2. Временная диаграмма чтения данных из микросхемы 1892XD4Ф
Временные параметры операций записи и чтения приведены в Таблица 3.4.

Таблица 3.4. Временные параметры операций записи и чтения

Наименование параметра, единица измерения	Буквенное обозначение	Норма	
		не менее	не более
Время установки данных, адреса и сигнала выборки до сигнала разрешения записи, нс	t_{SETWR}	5	-
Время установки адреса и сигнала выборки до сигнала разрешения чтения данных, нс	t_{SETRD}	0	-
Длительность сигнала записи данных, нс	t_{WE}	10,0	-
Задержка чтения данных, нс	t_{RD}	-	14

3.3.3 Обмен данными с использованием сигнала nACK

В этом случае все обмены данными с МВА выполняется за одну операцию записи или чтения.

Временная диаграмма операции записи данных в микросхему 1892XD4Ф через МВА приведена на Рисунок 3.3.

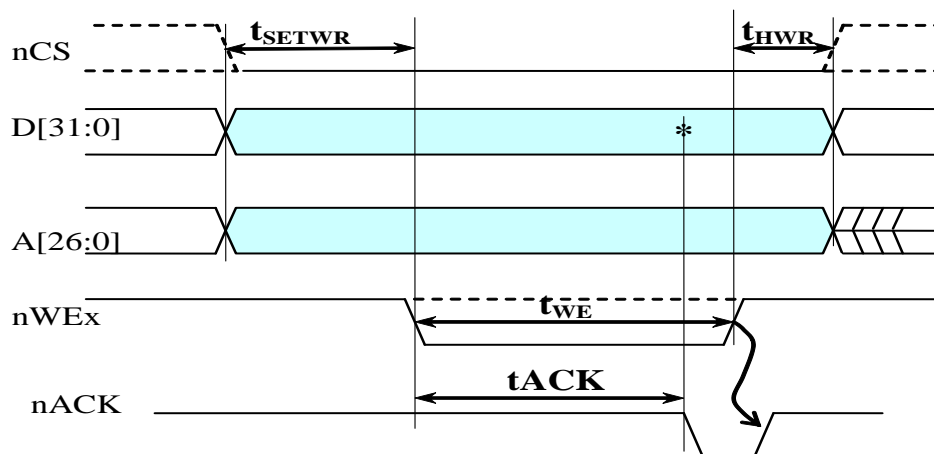


Рисунок 3.3. Временная диаграмма записи данных в микросхему 1892XD4Ф

Временная диаграмма операции чтения данных из микросхемы 1892ХД4Ф через МВА приведена на Рисунок 3.4.

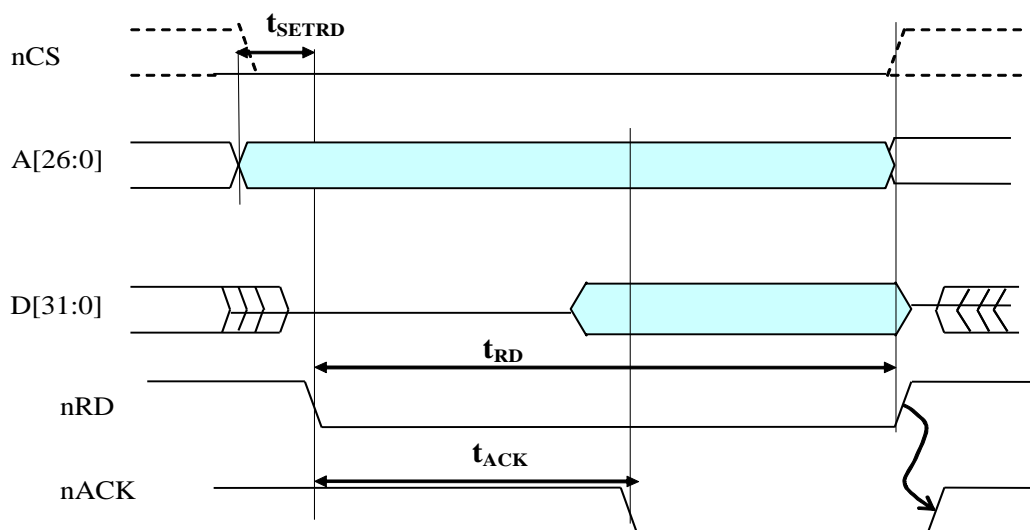


Рисунок 3.4. Временная диаграмма чтения данных из микросхемы 1892ХД4Ф

Временные параметры операций записи и чтения приведены в Таблица 3.5.

Таблица 3.5. Временные параметры операций записи и чтения

Наименование параметра, единица измерения	Буквенное обозначение	Норма	
		не менее нс	не более нс
Время установки данных, адреса и сигнала выборки до сигнала разрешения записи	t_{SETWR}	5	-
Время установки адреса и сигнала выборки до сигнала разрешения чтения данных	t_{SETRD}	0	-
Длительность сигнала разрешения записи данных	t_{WE}	20,0	-
Длительность сигнала разрешения чтения данных	t_{RD}	25	-
Задержка чтения данных	t_{ACK}	-	20
Время удержания данных, адреса и сигнала выборки после снятия сигнала разрешения записи	t_{HWR}	5	-

3.4 Подключение к микропроцессору нескольких микросхем 1892ХД4Ф

Микросхема 1892ХД4Ф имеет входы N[1:0] для задания номера микросхемы. Поэтому имеется возможность подключать к микропроцессору до четырех микросхем 1892ХД4Ф. При обращении к микросхеме два старших бита адреса A[26:25] необходимо устанавливать соответственно подключению входов N[1:0]. На Рисунок 3.5 приведена схема сравнения адреса.

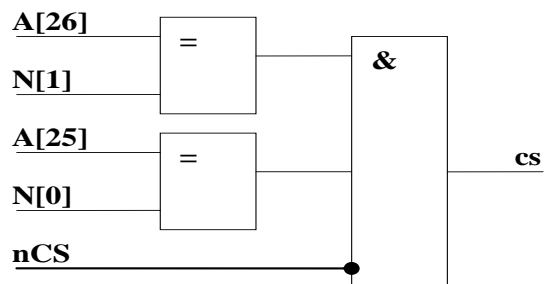


Рисунок 3.5. Дешифрация номера микросхемы

При соединении двух и более микросхем необходимо присваивать им различающиеся номера, например N[00], N[01].

4. КОНТРОЛЛЕР ШИНЫ PCI

4.1 Функциональные параметры и возможности

Контроллер шины PCI (PMSC – PCI Master-Slave controller) имеет следующие функциональные параметры и возможности:

- соответствует спецификации Local Bus Specification. Rev. 2.3;
- тактовая частота – от 33 до 66 МГц;
- шина адреса и данных - 32 разряда;
- обмен данными в режиме Master и Target (Slave);
- встроенный арбитр на 5 запросов шины;
- встроенные средства для организации мультимастерных систем;
- обеспечивает обмен данными между шиной PCI и любой областью внутренней памяти и регистрами устройств.

4.2 Структурная схема

Структурная схема PMSC приведена на Рисунок 4.1.

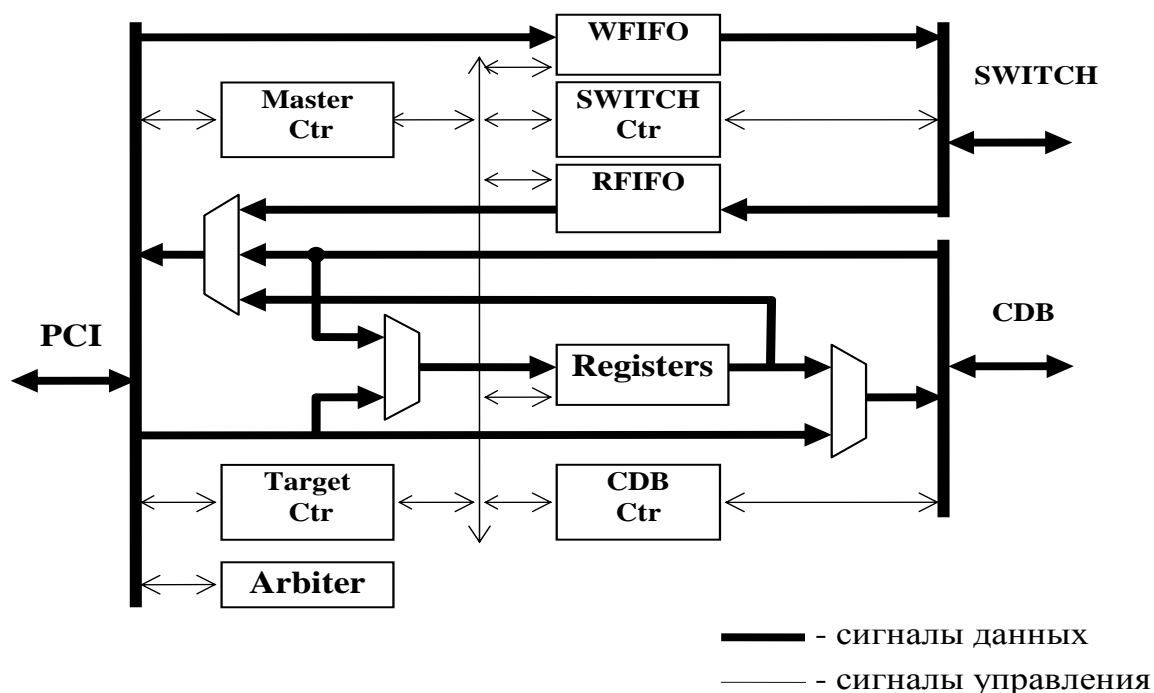


Рисунок 4.1. Структурная схема PMSC

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, BAR, Latency Timer, Interrupt Line;
 - регистры управления обменом: AR_PCI, IR_MASTER, CSR_MASTER, CSR_PCI, STATUS_MASTER, TMR_PCI, CSR_WIN;
 - регистры передачи векторов прерывания: MBR_MBA, 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 SWITCH;
- Target Ctr – блок управления передачи данных между шиной PCI и абонентами микросхемы 1892XD4Ф через коммутатор SWITCH в режиме Target (Slave);
- Master Ctr – блок управления передачи данных между шиной PCI и абонентами микросхемы 1892XD4Ф через коммутатор SWITCH в режиме Master;
- WFIFO – буфер FIFO объемом 16 32-разрядных слов для передачи данных из шины PCI в коммутатор SWITCH;
- RFIFO – буфер FIFO объемом 16 32-разрядных слов для передачи данных из коммутатора SWITCH в шину PCI;
- Arbiter – арбитр шины PCI.

4.3 Регистры

Перечень регистров PMSC, доступных со стороны шин PCI и CDB, приведен в Таблица 4.1.

Таблица 4.1. 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
BAR (Base Address Register)	Регистр базового адреса	0x10	0x00000008
Subsystem ID/ Subsystem Vendor ID	Регистр идентификации подсистемы	0x2C	0x00000002
Interrupt_Line	Код прерывания	0x3C	0x01200100
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_MBA	Регистр почтового ящика MBA	0x70	0x00000000

Смещение адреса определяется разрядами адреса 7:0 шин CDB и PCI и указано относительно базового адреса PMSC в адресном пространстве микросхемы 1892XD4Ф (см. таблицы 2.1, 2.2).

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, BAR, Interrupt_Line, IR_MASTER, AR_PCI, MASKR_PCI, MBR_MBA доступны по записи по командам Configuration Write, Memory Write.

Регистры CSR_MASTER, CSR_PCI, MBR_PCI, SEM, TMR_PCI доступны для записи 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.

4.3.1 Конфигурационные регистры

4.3.1.1 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 через MBA при инициализации PMSC.

4.3.1.2 Регистр BAR

Регистр BAR определяет базовый адрес PMSC на шине PCI в режиме Target при выполнении команд Memory Read, Memory Write.

Разряды 31-26 регистра BAR доступны по записи и чтению. Разряды 25-0 этого регистра доступны только по чтению кода 000_0008, что является индикатором пространства памяти объемом 64 Мбайт. Содержимое этого регистра определяется программно при инициализации PMSC через MBA или внешним контроллером PCI.

PMSC выполняет команды Memory Read, Memory Write при $AD[31:26]=BAR[31:26]$ в фазе адреса шины PCI. При невыполнении этого условия PMSC инициирует завершение команд Memory Read, Memory Write по условию “Master-abort” установкой высокого уровня сигнала nDEVSEL.

4.3.1.3 Регистр Status/Command

Формат регистра Status/Command приведен в Таблица 4.2.

Таблица 4.2. 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

Номер разряда	Условное обозначение	Описание	Доступ
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
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.

4.3.1.4 Регистр Latency Timer

Формат регистра Latency Timer приведен в Таблица 4.3

Таблица 4.3. 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

4.3.1.5 Регистр Interrupt Line

Формат регистра Interrupt Line приведен в Таблица 4.4

Таблица 4.4. 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

4.3.2 Регистры управления обменом данных

4.3.2.1 Регистр CSR_MASTER

Регистр CSR_MASTER предназначен для задания параметров и режимов передачи блока данных по шине PCI в режиме Master.

Запуск процедуры передачи блока данных 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 4.5.

Таблица 4.5. 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

Номер разряда	Условное обозначение	Назначение
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"
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 выходы 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

4.3.2.2 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 адрес абонента микросхемы 1892XD4Ф при выполнении транзакции на шине PCI в режиме Master.

Содержимое регистров AR_PCI и IR_MASTER в процессе обмена данными не изменяется.

4.3.2.3 Регистр STATUS_MASTER

Регистр STATUS_MASTER предназначен для контроля передачи блока данных по шине PCI в режиме Master.

Формат регистра STATUS_MASTER приведен в Таблица 4.6.

Таблица 4.6. 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 прекратиться
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.

4.3.2.4 Регистр параметров TMR_PCI

32-разрядный регистр TMR_PCI используется для хранения временных параметров выполнения транзакции на шине PCI в режиме Master.

Формат регистра TMR_PCI приведен в Таблица 4.7.

Таблица 4.7. Формат регистра TMR_PCI

Номер разряда	Условное обозначение	Назначение
31:16	WaterMark	Пороговое значение для формирования прерывания MASTER_WMARK: прерывание формируется после передачи WaterMark слов по шине PCI 0x0 – 1слово, 0x1– 2слова , ..., 0xFFFF – 65536 слов;
15:12 12: 8	- MILat	Не используется 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.

4.3.2.5 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 4.8.

Таблица 4.8. 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
15:12	-	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

Номер разряда	Условное обозначение	Назначение
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

4.3.2.6 Регистр CSR_PCI

Формат регистра CSR_PCI приведен в Таблица 4.9.

Таблица 4.9. 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
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

Номер разряда	Условное обозначение	Назначение
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 = 0xС 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 DPRAM в режиме Target: очередная порция данных в коммутатор 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.

При чтении данных из памяти DPRAM в режиме Target передача в шину PCI начинается по появлению первого слова в RFIFO.

4.4 Обмен данными по шине PCI в режиме Target (Slave)

В режиме Target по шине PCI доступны следующие ресурсы микросхемы 1892XD4Ф: регистры PMSC, регистры блоков SWIC, DMA_SWIC и память DPRAM.

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 DPRAM и регистрами SWIC выполняется по командам Memory Read, Memory Write.

Команды Memory Read Multiple, Memory Read Line исполняются как Memory Read, а Memory Write and Invalidate – как Memory Write.

Обмен данными производится 32-разрядными словами (формат DWORD) независимо от состояния выводов nCBE в фазе данных.

Передача данных в режиме Target регламентируется битами управления и параметрами регистров Status/Command и CSR_PCI и 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 данными с DPRAM и регистрами SWIC 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.

4.5 Обмен данными с шиной PCI в режиме Master

В режиме Master PMSC обеспечивает обмен данными между устройством на шине PCI и абонентами микросхемы 1892XD4Ф: регистрами блоков SWIC, DMA_SWIC и памятью DPRAM (см. таблицу 2.2).

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 1892ХД4Ф в регистр IR_MASTER;
- начальный адрес устройства на шине PCI в регистр AR_PCI;
- параметры WaterMark, MILat, MSLat, IRDLat в регистр TMR_PCI;
- параметры WC, WNM, sel_MILat, 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:
 - 2.1. В регистре CSR_MASTER: DONE = 1, Fatal Error =1, RUN=0.
 - 2.2. В регистре STATUS_MASTER: No Gnt =1, RUN=0.
 - 2.3. 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:
 - 7.1. В регистре CSR_MASTER: DONE=1, Fatal Error =1, RUN=0.
 - 7.2. 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:
 - 8.2. В регистре CSR_MASTER: DONE=1, Fatal Error=0, RUN=0.
 - 8.3. 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 программе.

В промежутках между транзакциями по шине PCI доступны для чтения и записи только регистры контроллера PMSC. При обращении к регистрам SWIC или к DPRAM PMSC как устройство “Target” завершает транзакцию требованием повтора передачи “Retry” (nSTOP=0, nTRDY=1 в начальной фазе данных).

4.6 Обмен данными с шиной PCI через адресное окно

Внешний микропроцессор через адаптер MBA может обмениваться данными с шиной PCI по операциям записи и чтения. Для этого в адресном пространстве микросхемы 1892XD4Ф выделено адресное окно размером 16 Мбайт: диапазон адресов от 000_0000 до 0FF_FFFF.

При чтении данных разряды 23:0 адреса устройства на шине PCI определяются разрядами 23:0 регистра BDR.

При записи данных разряды 23:0 адреса устройства на шине PCI определяются конкатенацией A[21:0] и двух нулей в младших разрядах.

Перед выполнением обмена данными 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).

При этом если CMD_WIN = 0xA или 0xB, на шине PCI выполняется однословная команда Configuration Read или Configuration Write,

если CMD_WIN = 0x2 или 0x3, выполняется однословная команда I/O Read или I/O Write,

иначе - команда Memory Read или Memory Write.

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: MASTER_ERROR = 1, MASTER_DONE = 0.

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 MASTER_ERROR=1, MASTER_DONE = 1;
 - если ошибка чётности не обнаружена или замаскирована (MASK_DPE = 0), то MASTER_ERROR=0, MASTER_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: MASTER_ERROR = 1, MASTER_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.

4.7 Передача прерываний

4.7.1 Передача вектора прерывания из шины PCI

Из шины PCI можно передать вектор прерывания внешнему процессору. Это 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 в регистре QSTR формируется прерывание INT_MBR. Это прерывание сбрасывается при считывании содержимого MBR из MBA. После обработки прерывания признак занятости MBR_PCI может быть сброшен записью нуля в регистр SEM из PCI по команде Memory Write или из MBA.

Конфигурационная запись в регистры MBR_PCI и SEM не выполняется.

4.7.2 Передача прерываний в шину PCI

Все сигналы прерываний от SWIC и DMA SWIC поступают на регистр QSTR_PCI. Формат регистра QSTR_PCI аналогичен регистру QSTR (за исключением разряда 28) и приведен в Таблица 4.10

Из MBA можно передать вектор прерывания в шину PCI с помощью 32-разрядного регистра почтового ящика MBR_MBA. При записи в этот регистр со стороны MBA в регистре QSTR_PCI формируется прерывание INT_MBA. Это прерывание сбрасывается при считывании содержимого MBR_MBA по шине PCI.

Таблица 4.10. Назначение разрядов регистра QSTR_PCI

Разряд регистра	Условное обозначение прерывания	Причина Прерывания
<u>Прерывания от PMSC</u>		
31	MASTER_DONE	Окончание передачи блока данных в режиме Master. Обнуляется при чтении регистра QSTR_PCI
30	MASTER_ERROR	Ошибка при передаче блока данных в режиме Master. Обнуляется при чтении регистра QSTR_PCI

Разряд регистра	Условное обозначение прерывания	Причина Прерывания
29	MASTER_WMARK	Предварительное уведомление о передаче по шине PCI в режиме Master заданного полем Water Mark регистра TMR_PCI количества слов. Обнуляется при чтении регистра QSTR_PCI
<u>Прерывание от MBA</u>		
28	INT_MBA	Признак записи данных в регистр почтового ящика MBR_MBA контроллера PMSC со стороны MBA. Обнуляется при чтении регистра MBR_MBA по шине PCI
<u>Прерывания от DMA SWIC3</u>		
27	SWIC3_TX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC3_TX_DATA
26	SWIC3_TX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC3_TX_DESC
25	SWIC3_RX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC3_RX_DATA
24	SWIC3_RX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC3_RX_DESC
<u>Прерывания от DMA SWIC2</u>		
23	SWIC2_TX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC2_TX_DATA
22	SWIC2_TX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC2_TX_DESC
21	SWIC2_RX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC2_RX_DATA
20	SWIC2_RX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC2_RX_DESC
<u>Прерывания от DMA SWIC1</u>		
19	SWIC1_TX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC1_TX_DATA
18	SWIC1_TX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC1_TX_DESC
17	SWIC1_RX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC1_RX_DATA
16	SWIC1_RX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC1_RX_DESC
<u>Прерывания от DMA SWIC0</u>		
15	SWIC0_TX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC0_TX_DATA
14	SWIC0_TX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC0_TX_DESC
13	SWIC0_RX_DATA	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC0_RX_DATA
12	SWIC0_RX_DESC	Окончание передачи блока данных или цепочки блоков данных каналом DMA SWIC0_RX_DESC
<u>Прерывания от SWIC3</u>		
11	SWIC3_TIM	Получен управляющий код
10	SWIC3_ERR	Ошибка в канале приема
9	SWIC3_LINK	Установлено соединение
<u>Прерывания от SWIC2</u>		
8	SWIC2_TIM	Получен управляющий код
7	SWIC2_ERR	Ошибка в канале приема
6	SWIC2_LINK	Установлено соединение
<u>Прерывания от SWIC1</u>		
5	SWIC1_TIM	Получен управляющий код
4	SWIC1_ERR	Ошибка в канале приема
3	SWIC1_LINK	Установлено соединение
<u>Прерывания от SWIC0</u>		

Разряд регистра	Условное обозначение прерывания	Причина Прерывания
2	SWIC0_TIM	Получен управляющий код
1	SWIC0_ERR	Ошибка в канале приема
0	SWIC0_LINK	Установлено соединение

Регистр QSTR_PCI доступен только по чтению. Исходное состояние регистра QSTR_PCI - нули (нет запросов прерывания).

Каждое прерывание, поступающее на регистр QSTR_PCI, маскируется при помощи регистра маски MASKR_PCI, формат которого аналогичен формату регистра QSTR_PCI. Регистр MASKR_PCI доступен по записи и чтению. Исходное состояние регистра MASKR_PCI - нули (все прерывания замаскированы).

Все незамаскированные прерывания регистра QSTR_PCI объединяются по схеме ИЛИ, и результат поступает на вход 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.

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 SWIC1_LINK сбрасывается при записи 1 в разряд [12] регистра статуса SWIC1_STATUS.

4.8 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 4.11 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 4.11. Приоритеты арбитра шины PCI

Обслуживаемый запрос	Распределение приоритетов очередного обмен
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]

5. КОНТРОЛЛЕР ИНТЕРФЕЙСА SPACEWIRE (SWIC)

5.1 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 микросхеме 1892ХД4Ф имеется четыре контроллера SWIC: SWIC0:SWIC3.

Основные особенности контроллера:

- контроллер разработан в соответствии с международным стандартом ECSS-E-50-12С;
- обеспечивает функционирование одного дуплексного канала связи со скоростью от 5 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;
- четыре канала DMA (два канала данных и два канала дескрипторов пакетов);
- обмен данными через DMA с памятью словами по 64 бита.

5.2 Блок-схема

Структура контроллера коммуникационного канала по стандарту SpaceWire приведена на рисунке 5.1.

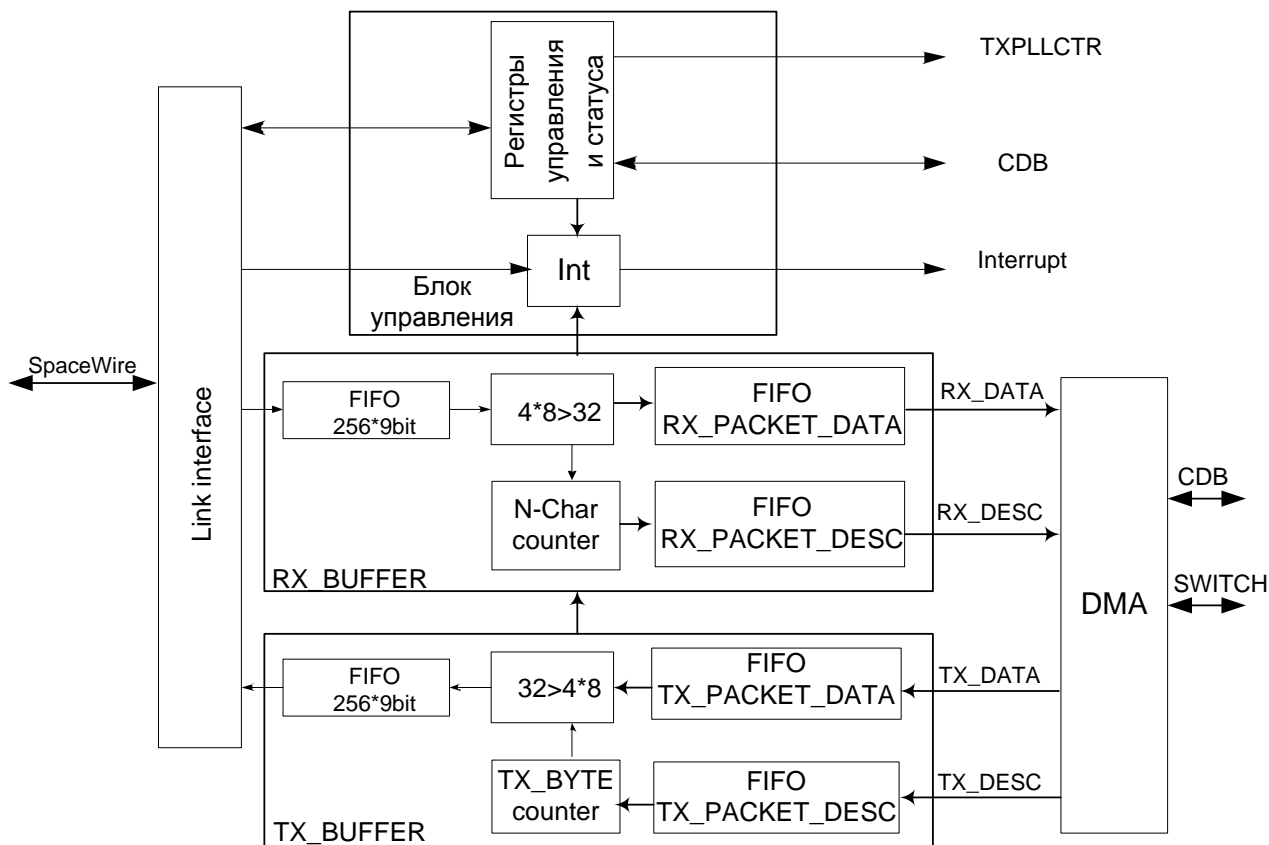


Рисунок 5.1. Структурная схема SWIC

Основой SWIC является блок Link interface (см. Figure 8-1 стандарта ECSS-E-50-12C), который через драйверы LVDS подключен к физическим линиям канала связи SpaceWire.

Управление SWIC осуществляется внешним микропроцессором или PMSC. Внешний микропроцессор (через MBA) или Master шины PCI (через PMSC) могут выполнять чтение и запись регистров контроллера для определения его состояния и настройки параметров работы.

Блок управления задает режимы работы блока Link interface. Передача управляющих кодов, кода распределенного прерывания, roll кода и контроль состояния последнего полученного извне маркера времени, производится через соответствующие регистры блока. В этом блоке находятся также регистры коэффициента скорости передачи данных и коэффициента скорости приема данных.

Блок формирования прерываний Int формирует соответствующие прерывания по состоянию блока Link interface.

Обмен данными между DPRAM и SWIC выполняется при помощи 4-канального DMA:

- канал дескрипторов передаваемых пакетов;

- канал данных передаваемых пакетов;
- канал дескрипторов принимаемых пакетов;
- канал данных принимаемых пакетов.

Описания работы блоков DMA приведено в п. 5.7.1.

К DMA SWIC подключен через буферы приема данных RX_BUFFER и передачи данных TX_BUFFER. Они обеспечивают согласование скоростей передачи данных между каналом SpaceWire и каналами DMA.

Буфер приема RX_BUFFER имеет конвейерную организацию и состоит из двух ступеней. Сначала в FIFO_256*9bit буферизируются восьмиразрядные данные, принимаемые от блока Link interface. Девятый служебный разряд несет информацию о признаке символа данных N-Char или символе конца пакета EOP. Затем в блоке преобразования формируются 32-разрядные слова данных и поступают в FIFO RX_PACKET_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_PACKET_DESC, а сам счетчик сбрасывается в 0.

В буфер передачи TX_BUFFER с помощью канала передаваемых данных DMA записываются 32-разрядные слова данных. Содержимое пакетов и их дескрипторы буферизируются в двух FIFO TX_PACKET_DATA и TX_PACKET_DESC соответственно. Данные из буфера передачи в блок Link interface выдаются побайтно через FIFO 256*9bit. Преобразование 32-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.

5.3 Прерывания

Контроллер SWIC формирует три прерывания, описание которых сведено в Таблица 5.1.

Таблица 5.1. 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.

Условное обозначение	Причина	Примечание
TIME	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].

5.4 Перечень регистров SWIC

5.4.1 Общие положения

Перечень программно-доступных регистров контроллера SWIC приведен в Таблица 5.2.

Таблица 5.2. Перечень регистров блока SWIC

Условное обозначение регистра	Название регистра	Тип доступа
HW_VER	Регистр версии контроллера	RD
STATUS	Регистр состояния	WRC/RD
RX_CODE	Регистр управляющего символа, принятого из сети (маркера времени, кода распределенного прерывания, кода подтверждения распределенного прерывания или кода CC11)	RD
MODE_CR	Регистр управления режимом работы	WR
TX_SPEED	Регистр управления скоростью передачи	WR
TX_CODE	Регистр управляющего символа (маркера времени, кода распределенного прерывания, кода подтверждения, кода CC11) для передачи в сеть	WR
RX_SPEED	Регистр измерителя скорости приема данных из канала SpaceWire	RD
CNT_RX_PACK	Регистр счетчика принятых пакетов ненулевой длины	RD/WR

Условное обозначение регистра	Название регистра	Тип доступа
ISR_L	Регистр кодов распределенных прерываний (LSB)	RD/WR
ISR_H	Регистр кодов распределенных прерываний (MSB)	RD/WR
TRUE_TIME	Регистр достоверного маркера времени	RD/WR
TOUT_CODE	Регистр размера таймаутов	RD
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
ACK_NONACK_MODE	Регистр управления режимом распределенных прерываний (с подтверждениями или без подтверждений)	RD/WR
ISR2_tout	Регистр таймаутов кодов распределенных прерываний 2	RD/WR
ISR_HANDLER_TERM_FUNC	Регистр флагов функций терминального узла обработчика	RD/WR
ISR_SPEC	Регистр рассылки управляющих кодов в специальный набор портов	RD/WR
ISR_1101	Регистр флагов приема управляющих кодов, назначение которых не определено в текущей версии стандарта	RD/WR
ISR_MASK_1101	Регистр маски портов, из которых не должны приниматься управляющие коды, назначение которых не определено в текущей версии стандарта	RD/WR
INT_RESET	Регистр параметров команды внешнего сброса	WR/RD
MODE_CR2	Регистр режима работы 2	
INT_H_MASK	Старшая половина регистра маски распределенных прерываний	WR/RD
INT_L_MASK	Младшая половина регистра маски распределенных прерываний	WR/RD
ACK_H_MASK	Старшая половина регистра маски кодов подтверждения	WR/RD
ACK_L_MASK	Младшая половина регистра маски кодов подтверждения	WR/RD
AUTO_SPEED_MANAGE	Регистр параметров автоматической установки скорости передачи	WR/RD
ISR_SOURCE_TERM_FUNC	Регистр флагов функций терминального узла источника	
ISR_SPEC_TERM_FUNC	Регистр признака специальной функции для терминального узла обработчика	
ISR_L_RESET	Младшая половина регистра глобального сброса ISR	
ISR_H_RESET	Старшая половина регистра глобального сброса ISR	

5.5 Описание регистров SWIC

5.5.1 Регистр HW_VER

Регистр номера версии SWIC. При чтении этого регистра выводится номер версии аппаратной реализации SWIC. В микросхеме 1892XD4Ф аппаратная версия SWIC – “0x0000 0005”.

Таблица 5.3. Назначение разрядов регистра HW_VER

Номер разряда	Условное обозначение	Описание
31:0	HW_VER	Номер версии SWIC

5.5.2 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 Link interface, блока приема данных из канала SpaceWire, блока передачи данных в канал SpaceWire. Назначение разрядов регистра приведено в Таблица 5.4.

Таблица 5.4. 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	LINK_STATE	Состояние Link interface. 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»

Номер разряда	Условное обозначение	Описание
11	TX_BUF_EMPTY	Буфер передачи пуст "1" – Пуст (после сигнала сброса) "0" – В буфере есть данные. Исходное состояние «1».
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 (LINK_STATE=5). Исходное состояние «0».
14	GOT_TIME	Принят маркер времени из сети "1" – Принят маркер времени "0" – Марке 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 упр. код "0" – Упр. код не принят (после сигнала сброса) Запись "1" в этот разряд сбрасывает этот разряд в "0". Исходное состояние «0».

Номер разряда	Условное обозначение	Описание
22	CC_01	Признак принятия управляющего кода C[7..6]=01 "1" – Принят упр. 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

5.5.3 Регистр RX_CODE

Регистр принятого из сети управляющего кода. Назначение разрядов регистра приведено в Таблица 5.5. Исходное состояние регистра не определено.

Таблица 5.5. 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

5.5.4 Регистр MODE_CR

Регистр режима работы. Назначение разрядов регистра приведено в Таблица 5.6.

Таблица 5.6. Назначение разрядов регистра MODE_CR

Номер разряда	Условное обозначение	Назначение
0	LinkDisabled	Установка LinkDisabled для блока Link interface
1	AutoStart	Установка Autostart для блока Link interface
2	LinkStart	Установка LinkStart для блока Link interface
3	-	Не используется
4	-	Не используется
5	LINK_RST	Сброс Link interface
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 Link interface)

Номер разряда	Условное обозначение	Назначение
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.
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 Link interface 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.

5.5.5 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).

Запись нового значения в этот регистр возможно только, если бит COEFF_10_wr (14) регистра MODE_CR (режима) установлен в 1.

Таблица 5.7. 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
17:10	TX_SPEED_10	Определяет скорость передачи данных при установке соединения (в режиме авто установки скорости). Должен быть записан код 0x02
18	PLL_TX_EN_10	Управление работой PLL_TX_SWIC: 1 – работа разрешена; 0 – работа запрещена. PLL_TX_SWIC находится в режиме пониженного энергопотребления. Состояния разрядов 8 и 18 этого регистра должны быть одинаковыми
19	LVDS_EN_10	Управление работой приемопередатчиков LVDS SWIC: 1 – работа разрешена; 0 – работа запрещена. LVDS SWIC находятся в режиме пониженного энергопотребления. Состояния разрядов 9 и 19 этого регистра должны быть одинаковыми

Номер разряда	Условное обозначение	Назначение
28:20	COEFF_10	Значение коэффициента для подсчета таймаутов установки соединения. Необходимо установить значение 0x0A
31..29	-	Резерв

5.5.6 Регистр TX_CODE

Регистр управляющего кода для передачи в канал. Сразу же после записи в этот регистр начинается передача управляющего символа в Link interface и далее в канал.

Таблица 5.8. 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

5.5.7 Регистр RX_SPEED

Назначение разрядов регистра RX_SPEED приведено в Таблица 5.9.

Таблица 5.9. Назначение разрядов регистра RX_SPEED

Номер разряда	Условное обозначение	Назначение
31:8	-	Не используется
7:0	RX_SPEED	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.

5.5.8 Регистр CNT_RX_PACK

Регистр счетчика принятых пакетов. Значение регистра увеличивается на 1 каждый раз, когда из Link interface прочитывается символ конца пакета, если ему предшествовал один или более символ данных. Исходное состояние регистра «0».

При записи, значение регистра обнуляется. Содержимое этого регистра может быть обнулено для того, чтобы начать счет пакетов заново. Рекомендуется выполнять сброс регистра каждый раз при выполнении новой настройки DMA для передачи данных в сеть.

Таблица 5.10. Назначение разрядов регистра CNT_RX_PACK

Номер разряда	Условное обозначение	Описание
31:0	CNT	Число принятых пакетов

5.5.9 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, если 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 прочесть 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 5.11. Назначение разрядов регистра ISR_L

Номер разряда	Условное обозначение	Описание
31:0	ISR_L	Младшая часть регистра ISR

5.5.10 Регистр ISR_H

В этот регистр отображается старшая [63:32] часть регистра ISR.

Таблица 5.12. Назначение разрядов регистра ISR_H

Номер разряда	Условное обозначение	Описание
31:0	ISR_H	Старшая часть регистра ISR

5.5.11 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 5.13. Исходное состояние регистра «0».

Таблица 5.13. Назначение разрядов регистра TRUE_TIME

Номер разряда	Условное обозначение	Описание
5:0	TRUE_TIME	Значение последнего правильного маркера времени
31:6	-	Не используется

5.5.12 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 5.14. Назначение разрядов регистра TOUT_CODE

Номер разряда	Условное обозначение	Описание
15..0	GLOB_COU	Значение периода глобального счетчика (задается в тактах локальной частоты)
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	

5.5.13 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 декрементируется 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 5.15. Назначение разрядов регистра ISR_tout_L

Номер разряда	Условное обозначение	Описание
31:0	ISR_tout_L	Младшая часть регистра ISR_tout

5.5.14 Регистр ISR_tout_H

В этот регистр отображается старшая (63..32) часть регистра ISR_tout.

Таблица 5.16. Назначение разрядов регистра ISR_tout_H

Номер разряда	Условное обозначение	Описание
31:0	ISR_tout_H	Старшая часть регистра ISR_tout

После сброса содержимое регистров «0».

5.5.15 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 5.17. Назначение разрядов регистра LOG_ADDR

Номер разряда	Условное обозначение	Описание
31:0	LOG_ADDR	Значение логического адреса.

5.5.16 Регистр ACK_NONACK_MODE

В данном регистре содержатся значения флагов режима для каждого кода распределенного прерывания (прерывания с номерами с 0 по 31). Если флаг режима установлен в 0 – это соответствует режиму с подтверждением, если в 1 – то режиму без подтверждений. Значение по умолчанию – 0.

Таблица 5.18. Назначение разрядов регистра ACK_NONACK_MODE

Номер разряда	Условное обозначение	Описание
31:0	MODE_FLAG S	Значения флагов режима

5.5.17 Регистр ISR2_tout

В данном регистре содержатся значения таймаутов (подсчет осуществляется на базе глобального счетчика таймаутов, задаваемого в регистре ISR_TOUTS)

Таблица 5.19. Назначение разрядов регистра ISR_TOUTS2

Номер разряда	Условное обозначение	Описание
3..0	LOC_Reset_SW_nack	Значение локального счетчика таймаутов сброса ISR в режиме без кодов подтверждений
7..4	LOC_RESET_T1_nack	Значение локального счетчика таймаутов сброса ISR в режиме терминального узла в режиме без кодов подтверждений
11..8	LOC_TG	Значение локального счетчика таймаутов TG (в режиме терминального узла используется только в режиме с подтверждениями)
15..12	LOC_TH	Значение локального счетчика таймаутов TH (в режиме терминального узла используется только в режиме с подтверждениями)
19..16	LOC_ISR_CHANGE	Значение локального счетчика таймаутов TISR_change (используется одно и то же значение и в режиме коммутатора и в режиме терминального узла)

5.5.18 Регистр ISR_HANDLER_TERM_FUNC

Регистр ISR_HANDLER_TERM_FUNC предназначен для индикации того, выполняет ли данный SWIC функции обработчика кодов прерываний.

В соответствии со значением разряда 29 регистра ISR_SPEC через этот регистр можно работать с регистром ISR_term_func_L или ISR_term_func_H. Запись 1 в соответствующий разряд регистра указывает, что для данного кода распределенного прерывания SWIC будет обработчиком.

Начальное состояние регистра – 0.

5.5.19 Регистр специальных кодов ISR_SPEC

Регистр специальных кодов (управляющих символов, назначение которых не определено в стандарте ECSS-E-50-12C) предназначен для получения информации о принятых из сети кодах, для которых не определена специальная интерпретация (коды 01xxxxxx для портов, работающих в режиме 5-и разрядных кодов распределенных прерываний и коды 11xxxxxx).

Для данных кодов используется следующая схема регистрации. Для каждого типа кодов существует регистр ISR – ISR_11 и ISR_01 соответственно (данные регистры, однако, явным образом не являются программно доступными, работа с ними осуществляется через регистры ISR_1101 и ISR_SPEC). При приеме из сети управляющего кода 11xxxxxx

соответствующий его номеру разряд регистра ISR_11 устанавливается в 1 и регистрируется номер порта, из которого он поступил. Если происходит повторное поступление кода с тем же номером из другого порта, номер порта перезаписывается. Разряды ISR_11 могут программно сбрасываться через ISR_1101.

Аналогичная схема поведения определена для кодов 01xxxxxx, поступающих из портов, для которых задан режим 5-и разрядных кодов распределенных прерываний. Они регистрируются в регистре ISR_01.

В CODE_TYPE записывается идентификатор кода (значение разрядов 7:6 кодов, информацию о которых планируется получить). Если, например, планируется получение информации о кодах 11xxxxxx, то в эти разряды необходимо записать 11. В соответствии со значением этих разрядов выполняется логическое связывание программно доступного регистра ISR_1101 с регистром ISR_11 или ISR_01 и логическое связывание программно доступного регистра ISR_MASK_1101 с регистром ISR_MASK_11 или ISR_MASK_01.

В поле CODE_HL записывается признак 0 или 1, указывающий какая половина регистра ISR_xx будет отображаться в регистр ISR_1101 (0 соответствует младшей половине, 1 – старшей половине), а так же какая половина регистра ISR_mask_xx будет отображаться в регистр ISR_mask_1101 (0 соответствует младшей половине, 1 – старшей половине), какая половина регистра ISR_term_funct будет отображаться в соответствующий регистр

В поле CODE_NUM записывается номер кода, для которого необходимо прочитать номер порта из которого этот код поступил в последний раз (если соответствующий разряд ISR_xx установлен в 0, то считается, что код не поступал)

После того, как разряды 31..23 установлены в нужные значения, из разрядов 4..0 этого регистра можно прочитать номер порта, из которого он поступил (если не поступал, то будет прочитано значение 0)

Формат регистра ISR_SPEC показан в Таблица 5.20.

Таблица 5.20. Назначение разрядов регистра ISR_SPEC

Номер разряда	Условное обозначение	Описание
4..0	LAST_PORT	Номер порта, из которого управляющий символ поступил в последний раз
22..5		Не используется
28..23	CODE_NUM	Номер символа, о котором нужно получить информацию
29	CODE_HL	Указывает, со старшей или младшей половиной регистра ISR_11, ISR_01, ISR_mask_11, ISR_mask_01, ISR_term_funct будет осуществляться работа
31..30	CODE_TYPE	Указывает тип кода: 11 или 01 (будет выполняться обращение к ISR_11 или ISR_01 при обращении к ISR_1101; будет выполняться обращение к ISR_mask_11 или ISR_mask_01 при обращении к ISR-mask_1101;)

Начальное значение регистра – 0.

5.5.20 Регистр ISR_1101

Регистр ISR_1101 предназначен для получения доступа к регистрам ISR_11 и ISR_01.

В соответствии со значением разрядов 31..29 регистра ISR_SPEC через этот регистр можно работать с регистром ISR_11 или ISR_01. если необходимо сбросить разряд ISR_xx, то в соответствующий разряд необходимо записать 1.

Начальное значение регистра – 0.

5.5.21 Регистр ISR_MASK_1101

Регистр ISR_mask_1101 предназначен для получения доступа к регистрам ISR_MASK_11 и ISR_MASK_01

Начальное значение регистра – 0.

5.5.22 Регистр INT_RESET

В данном регистре хранятся параметры для формирования сигнала сброса устройства по команде от удаленного администратора сети SpaceWire. Данная команда представляет собой последовательность из пяти кодов: распределенное прерывание 0, код подтверждения 0, распределенное прерывание 0, код подтверждения 0, распределенное прерывание 0. Эта последовательность кодов должна поступить в течении времени таймаута, задаваемого полем L_RESET_COU. Данное время таймаута подсчитывается в периодах, задаваемых полем G_RESET_INT. Период подсчитывается в мкс (основной режим) или в тактах локальной частоты Slave контроллера (отладочный режим). Режим задается полем R_MODE.

В поле W_INT задается интервал времени между обнаружением команды сброса и сбросом устройства. Данный интервал задается в тактах локальной частоты.

Формат регистра INT_RESET показан в Таблица 5.21.

Таблица 5.21. Назначение разрядов регистра INT_RESET

Номер разряда	Условное обозначение	Описание
15..0	G_RESET_INT	Значение глобального периода подсчета интервала времени
21..16	L_RESET_COU	Значение таймаута, в течении которого должно поступить 3 распределенных прерывания 0 для того, чтобы они были интерпретированы как команда сброса Slave контроллера
22	R_MODE	Режим подсчета интервала времени. Если этот разряд установлен в 0, то подсчет выполняется в тактах, если этот разряд установлен в 1, то подсчет выполняется в мкс
31..23	W_INT	Интервал времени между получением команды сброса и выполнением сброса slave контроллера.

Начальное значение регистра – 0.

5.5.23 Регистр MODE_CR2

Формат регистро MODE_CR2 приведен в Таблица 5.22.

Таблица 5.22. Назначение разрядов регистра MODE_CR2

Номер разряда	Условное обозначение	Описание
0:1	-	Не используется, должен быть установлен в 0
2	Ccode_send_mask	Маска отправки кодов распределенных прерываний и подтверждений в сеть. Если данный разряд установлен в 1, то отправка запрещена
3	Ccode_res_mask	Маска приема кодов распределенных прерываний и подтверждений из сети. Если данный разряд установлен в 1, то прием запрещен
4	Int_send_mask	Маска отправки кодов распределенных прерываний в сеть. Если данный разряд установлен в 1, то отправка запрещена
5	Int_res_mask	Маска приема кодов распределенных прерываний из сети. Если данный разряд установлен в 1, то прием запрещен
6	Ack_send_mask	Маска отправки кодов подтверждений в сеть. Если данный разряд установлен в 1, то отправка запрещена
7	Ack_res_mask	Маска приема кодов подтверждений из сети. Если данный разряд установлен в 1, то прием запрещен
10..8	Host_int_code	Кодировка для интерпретации кода распределенного прерывания, записываемого в регистр TX_CODE (в режиме 5-и биных кодов)
13..11	Host_ack_code	Кодировка для интерпретации кода подтверждения, записываемого в регистр TX_CODE (в режиме 5-и биных кодов)
14	sdIRQ_regime	Режим для внешнего порта – если данный разряд установлен в 0, то отправка осуществляется в кодировке для 6-и битных кодов, если данный разряд установлен в 1, то отправка осуществляется в кодировке для 5-и битных кодов
17..15	Spw_int_code	Кодировка для интерпретации кода распределенного прерывания, приходящего из сети (в режиме 5-и биных кодов)
20..18	SpW_ack_code	Кодировка для интерпретации кода подтверждения, приходящего из сети (в режиме 5-и биных кодов)
21	Time_send_mask	Маска отправки маркеров времени в сеть. Если данный разряд установлен в 1, то отправка запрещена
22	Time_res_mask	Маска приема маркеров времени из сети. Если данный разряд установлен в 1, то прием запрещен
23	Add_res_mask	Маска приема кодов, назначение которых не определено стандартном,сетью. Если данный разряд установлен в 1, то прием запрещен
24	err_regime	Если данный разряд имеет значение 1, то при переходе DS-макрочейки в состояние RUN автоматически сбрасываются разряды ошибок в регистре статуса и сбрасывается прерывание IRQ_1(ERR) для процессора. Если данный разряд установлен в 1, то данные действия автоматически не выполняются. Значение по умолчанию - 0

5.5.24 Регистры маски распределенных прерываний – INT_H_MASK, INT_L_MASK

Регистры INT_H_MASK[31:0], INT_L_MASK[31:0] предназначены для определения маски распределенных прерываний (определяют, при получении каких распределенных прерываний будет установлено прерывание IRQ_int). Если в *i* разряде '0' – прерывание при приходе кода подтверждения с номером *i* разрешено, если '1' – запрещено.

Начальное значение регистра – 0.

5.5.25 Регистры маски кодов подтверждения – ACK_H_MASK, ACK_L_MASK

Регистры ACK_H_MASK[31:0], ACK_L_MASK[31:0] предназначены для определения маски кодов подтверждения (определяют, при получении каких кодов подтверждения будет установлено прерывание IRQ_ack для внутреннего процессора). Если в *i* разряде '0' – прерывание при приходе кода подтверждения с номером *i* разрешено, если '1' – запрещено.

Начальное значение регистра – 0.

5.5.26 Регистр AUTO_SPEED_MANAGE

В данном регистре хранятся параметры для режима автоматической установки соединения.

Формат регистра AUTO_SPEED_MANAGE показан в Таблица 5.23.

Таблица 5.23. Назначение разрядов регистра AUTO_SPEED_MANAGE

Номер разряда	Условное обозначение	Описание
4..0	AUTO_COU	Количество неудачных переходов на заданную базовую скорость, после которого при очередном соединении переход на базовую скорость не выполняется
15..5	CONTROL_TOME	Таймаут установки соединения, отсчитывается с момента перехода порта Spacewire в состояние <code>gip</code> ("101») в тактах, если бит CONTROL_MODE установлен в 0 и микросекундах, если в 1
17..16	AUTO_COU_FLS	Признаки того, что количество неудачных переходов для портов исчерпано
30..18	-	Не используется
31	CONTROL_MODE	Режим подсчета таймаута установки соединения, 0 – в тактах, 1 – в мкс

Начальное значение регистра 0

В поле AUTO_COU задается максимальное количество неудачных переходов на заданную базовую скорость при использовании режима автоматической установки соединения.

Переход считается неудачным, если соединение разорвалось до истечения таймаута установки соединения, заданного в поле Control_time данного регистра.

В поле AUTO_COU_FLS отображаются признаки того, что количество неудачных переходов для портов исчерпано (разряд 16 соответствует 1 порту). После установки признака автомат установки соединения не прекращает попыток перейти на базовую скорость.

Для того, чтобы перезапустить автомат, необходимо записать 1 в соответствующий разряд AUTO_COU_FLS. В результате счетчик неудачных переходов на заданную базовую скорость будет сброшен и попытки перехода возобновлены.

5.5.27 Регистр ISR_SOURCE_TERM_FUNC

Регистр ISR_SOURCE_TERM_FUNC предназначен для индикации того, выполняет ли данный SWIC функции источника по отношению к каждому из кодов распределенных прерываний.

В соответствии со значением разряда 29 регистра ISR_SPEC через этот регистр можно работать с регистром ISR_source_term_funct_L или ISR_source_term_funct_H. Запись 1 в соответствующий разряд регистра указывает, что для данного кода распределенного прерывания Slave контроллера будет выполняться функции источника. Детальное описание механизма обработки кодов распределенных прерываний приведено в п. 5.6.10.

Начальное значение регистра – 0.

5.5.28 Регистр ISR_SPEC_TERM_FUNC

Регистр ISR_SPEC_TERM_FUNC предназначен для разрешения автоматической отсылки в сеть кода подтверждения по истечении таймаута Th, если данный узел является обработчиком соответствующего кода распределенного прерывания по отношению к каждому из кодов распределенных прерываний.

В соответствии со значением разряда 29 регистра ISR_SPEC через этот регистр можно работать с регистром ISR_spec_term_funct_L или ISR_spec_term_funct_H. Запись 1 в соответствующий разряд регистра в сочетании с установленным в 1 разрядом ISR_handler_term_funct указывает, что для данного кода распределенного прерывания Slave контроллера будет выполняться автоматическая отсылка кода подтверждения. Детальное описание механизма обработки кодов распределенных прерываний приведено в п. 5.6.10

Начальное значение регистра – 0.

5.5.29 Регистр ISR_L_RESET

Регистр ISR_L_RESET предназначен для поразрядного сброса младшей половины регистра ISR (ISR_L), а так же для сброса всех флагов и счетчиков, связанных со сбрасываемыми разрядами. Сбрасываются разряды, в которые осуществляется запись 1.

Детальное описание механизма обработки кодов распределенных прерываний приведено в п. 5.6.10.

Начальное значение регистра – 0.

5.5.30 Регистр ISR_H_RESET

Регистр ISR_H_RESET предназначен для поразрядного сброса старшей половины регистра ISR (ISR_H), а так же для сброса всех флагов и счетчиков, связанных со сбрасываемыми разрядами. Сбрасываются разряды, в которые осуществляется запись 1.

Детальное описание механизма обработки кодов распределенных прерываний приведено в п. 5.6.10.

Начальное значение регистра – 0.

5.6 Логика работы SWIC

5.6.1 Расположение данных в памяти.

Рассмотрим пример (см. Рисунок 5.4) представления данных в памяти DPRAM, если для данных выделен один сегмент памяти. Пусть в DPRAM 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 для записи дескрипторов. В дескрипторах указаны размеры пакетов в байтах – 0Ah, 08h и 0Vh соответственно. В дескрипторах хранится так же информация о типе конца пакета. В разряд 31 дескриптора записывается 1, что указывает на то, что дескриптор заполнен действительными данными.

5.6.2 Схема обработки данных

В данном примере пакеты могут быть обработаны в соответствии со следующей схемой. Прочитывается 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.

5.6.3 Прием данных из канала SpaceWire

Маршрут принимаемых данных и схема их обработки приведены на Рисунок 5.2.

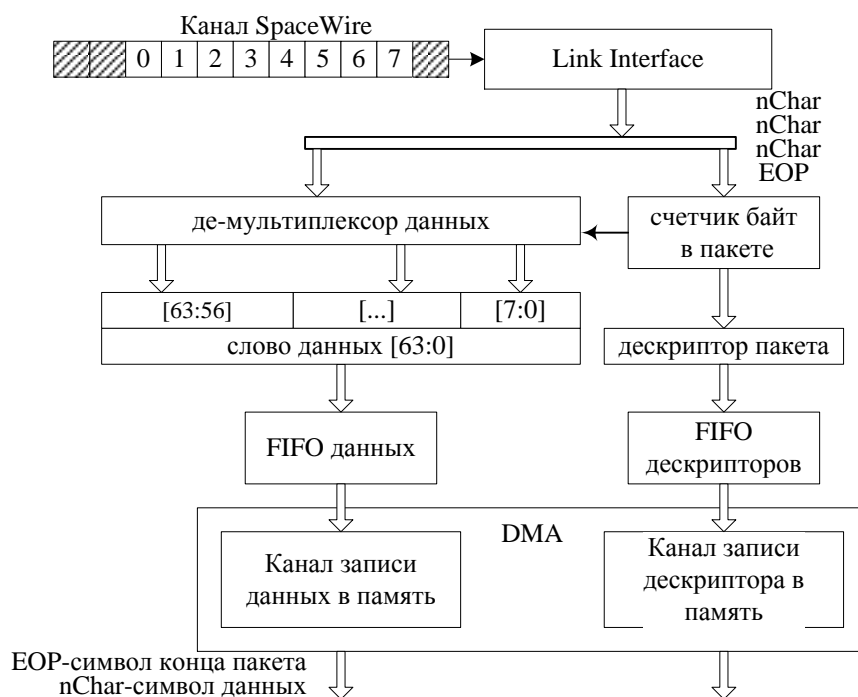


Рисунок 5.2. Схема приема данных из канала SpaceWire

Из канала SpaceWire в Link interface символы данных поступают последовательно (побитно). Link interface выделяет из последовательности приходящих символов символы данных и символы концов пакетов и передает их в блок приема. По каналу SpaceWire 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 канала SpaceWire 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.

5.6.4 Передача данных в канал SpaceWire

Процесс передачи пакетов данных из системной памяти в канал через контроллер, а также преобразование форматов данных показаны на Рисунок 5.3.

Пакеты данных загружаются из системной памяти в буфер передачи через каналы DMA чтения данных из памяти и чтения дескриптора из памяти.

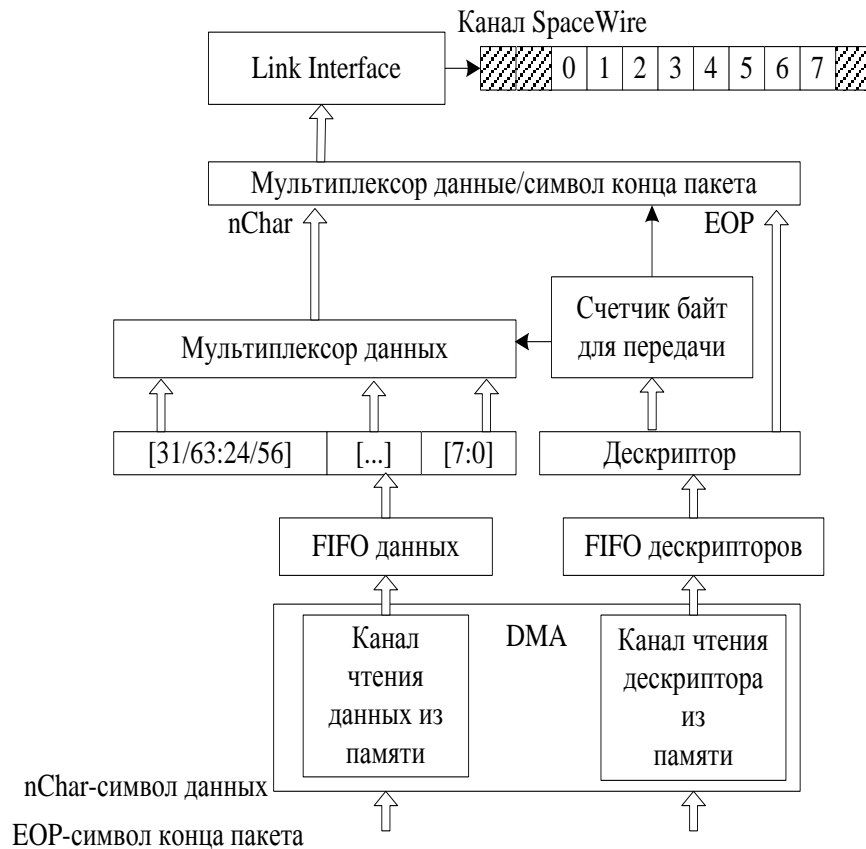


Рисунок 5.3. Передача данных из системной памяти в канал SpaceWire

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 вставляется символ конца пакета EOP. Код маркера EOP формируется 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 Link Interface 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 Link Interface. Link Interface преобразует полученные символы в соответствии с алгоритмом DS кодирования и передает их в канал. Символы передаются младшими разрядами вперед.

5.6.5 Выравнивание границ пакетов по границам слов

Рассмотрим выравнивание пакетов данных на примере Рисунок 5.4. 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).

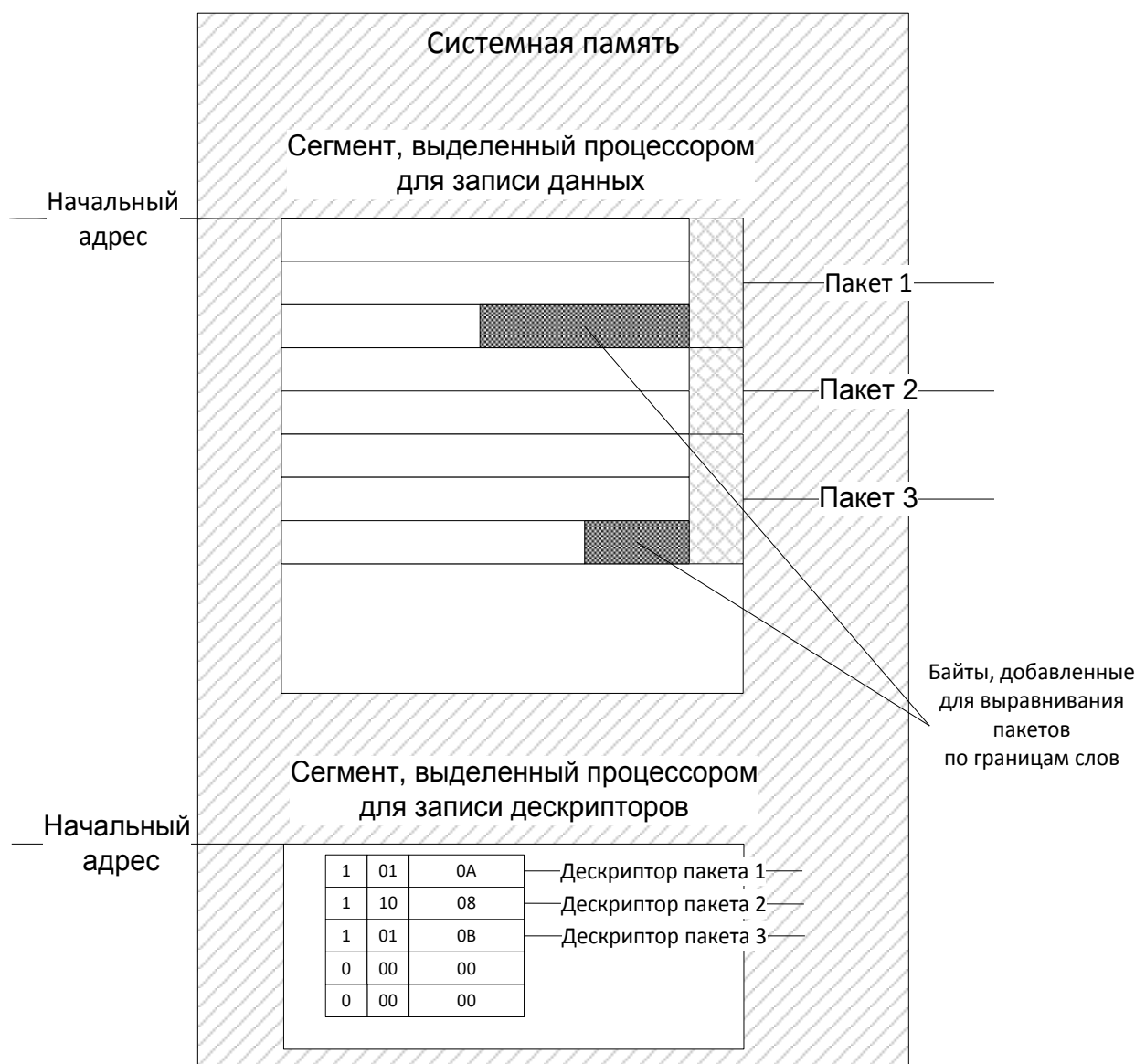


Рисунок 5.4. Представление данных в памяти DPRAM (пример)

5.6.6 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). В дескрипторах для передачи пакетов этот бит необходимо установить в единичное состояние. 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.

5.6.7 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 (EER, 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. Дескриптор 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_ADDR 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 EER.

5.6.8 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.

5.6.9 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, кодов подтверждения прерываний и кодов CC11). После записи выполняется проверка корректности маркера времени – его значение сравнивается со значением поля текущего маркера времени внутреннего регистра контроллера, отображаемого в программно-доступный регистр маркера времени (TRUE_TIME.CUR_TIME). Если значение, записанное в TX_CODE, на 1 больше (циклически) значения TRUE_TIME.CUR_TIME, то маркер времени считается корректным, в противном случае маркер времени считается не корректным. Если маркер времени признан корректным, он передается в сеть и записывается в поле последнего правильного маркера времени TRUE_TIME.TRUE_TIME. В противном случае он записывается в TRUE_TIME.CUR_TIME и более никаких действий не выполняется.

Если маркер времени признан корректным, то DS-макроячейка дожидается окончания передачи символа данных или служебного символа и начинает передачу маркера времени, после окончания передачи маркера времени продолжается передача потока данных.

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 (значение которого отображается в регистр в TRUE_TIME.CUR_TIME), 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 признан корректным, то его значение заносится в TRUE_TIME.CUR_TIME и TRUE_TIME.TRUE_TIME

Если маркер времени не является корректным, то его значение так же заносится в TRUE_TIME.CUR_TIME и соответствующие разряды регистра RX_CODE, однако, прерывание для процессора в данном случае не устанавливается и в TRUE_TIME.TRUE_TIME он не записывается. В начале работы устройства или после сброса маркер времени со значением 1 рассматривается как корректный.

Значение последнего корректного маркера времени хранится в регистре TRUE_TIME.TRUE_TUME.

Значение текущего маркера времени отображается в регистре TRUE_TIME.CUR_TUME.

5.6.10 Коды распределенных прерываний и подтверждений

Коды распределенных прерываний и подтверждений являются расширением стандарта SpaceWire.

В SWIC поддерживается два режима работы с кодами распределенных прерываний – режим 5-и разрядных кодов и режим 6-и разрядных кодов. В режиме 5-и разрядных кодов распределенных прерываний используются следующие кодировки:

100xxxxx – коды распределенных прерываний;

101xxxxx – коды подтверждений.

Младшие 5 разрядов кода – номер распределенного прерывания или подтверждения

В режиме 6-и разрядных кодов распределенных прерываний используются следующие кодировки:

01xxxxxx – коды распределенных прерываний;

10xxxxxx – коды подтверждений.

Младшие 6 разрядов кода – номер распределенного прерывания или подтверждения

При передаче коды распределенных прерываний и подтверждений имеют приоритет, следующий после маркеров времени.

Для управления режимами обработки распределенных прерываний и подтверждений используются следующие регистры:

ISR_H,L; INT_H,L_MASK; ACK_H,L_MASK, ISR_TOUT_H,L; TOUT_CODE;
ISR_TOUTS2; ISR_HANDLER_TEREM_FUNC; ISR_SOUSCE_TERM_FUNC;
ISR_SPEC_TERM_FUNC, ACK_NON_ACK_MODE, ISR_H,L_RESET; MODE_CR2;

Поля Ccode_send_mask и Ccode_res_mask в регистре MODE_CR2 позволяют замаскировать отправку и прием из сети кодов распределенных прерываний и подтверждений, а поля Int_res_mask, Int_send_mask, Ack_res_mask, Ack_send_mask позволяют замаскировать отправку и прием из сети отдельно кодов распределенных прерываний и отдельно кодов подтверждений. По умолчанию прием и отправка кодов расмаскированы.

Регистры INT_H,L_MASK; ACK_H,L_MASK позволяют замаскировать установку прерывания для процессора по факту приема из сети кодов распределенных прерываний и подтверждений. На сам прием и обработку кодов эта настройка никак не влияет.

Регистр `ACK_NON_ACK_MODE` позволяет задать режим обработки кодов прерываний – с подтверждениями или без подтверждений. По умолчанию установлен режим с подтверждениями. Если в режиме без подтверждений приходит из сети или отправляется процессором код подтверждения, то он отбрасывается без какой-либо дальнейшей обработки.

Регистр `ISR_SOURCE_TERM_FUNC` позволяет задать, для каких кодов распределенных прерываний данный терминальный узел будет источником. Если задано ненулевое значение таймаута T_g , то узел сможет отправлять в сеть только коды подтверждений, соответствующие прерываниям, для которых он является источником. Если значение $T_g=0$ то узел сможет отправлять в сеть любые коды прерываний, не зависимо от значения этого регистра.

Регистр `ISR_HANDLER_TERM_FUNC` позволяет задать, для каких кодов распределенных прерываний данный терминальный узел будет обработчиком. Если задано ненулевое значение таймаута T_h , то узел сможет отправлять в сеть только коды подтверждений, соответствующие прерываниям, для которых он является обработчиком. Если значение $T_h=0$ то узел сможет отправлять в сеть любые коды подтверждений, не зависимо от значения этого регистра.

Регистр `ISR_SPEC_TERM_FUNC` позволяет задать возможность автоматической отправки в сеть кодов подтверждений на прерывания, для которых данный узел является обработчиком. Если `ISR_handler_term_func(i) = 1` и `ISR_spec_term_func(i) = 1`, то после приема кода распределенного прерывания `INTRi` по истечении таймаута T_h в сеть автоматически будет отправлен код подтверждения `INTAi`.

Регистры `TOUT_CODE` и `ISR_TOUTS2` позволяют задать значения таймаутов для кодов распределенных прерываний и подтверждений. Для того, чтобы осуществлялся подсчет таймаута (любого) необходимо в поле `GLOB_COU` регистра `TOUT_CODE` задать ненулевое значение глобального периода подсчета таймаутов. Все значения таймаутов считаются в периодах глобального счетчика.

При приеме кода распределенного прерывания/подтверждения из сети выполняются следующие действия.

Проверяется, не замаскирован ли прием кодов данного типа из сети. Если прием замаскирован, то код отбрасывается без каких-либо дальнейших действий.

Если прием кода не замаскирован, то начинается его обработка.

При приеме кода распределенного прерывания `INTRi` проверяется значение регистра `ISR`. Если `ISR(i)=1`, то код отбрасывается без каких-либо дальнейших действий. Если `ISR(i)=0` и используется режим с подтверждениями, то проверяется состояние счетчика таймаутов изменения значения для этого разряда `ISR` (`ISR_TOUTS2.LOC_ISR_CHANGE`). Если

данный таймаут не задан, уже истек, или если $INTR_i$ принимается первый раз после сброса, то данный $INTR_i$ считается корректным, $ISR(i)$ устанавливается в значение 1 и данный код записывается в регистр RX_CODE . В противном случае данный код считается не корректным, он отбрасывается без каких-либо дальнейших действий.

Если $ISR(i)=0$ и используется режим без подтверждений, то данный код считается корректным, $ISR(i)$ устанавливается в значение 1 и данный код записывается в регистр RX_CODE .

Если для принятого корректного кода $INTR_i$ используется режим с подтверждениями и $ISR_TOUTS2.LOC_ISR_CHANGE$ не равен 0, то запускается счет таймаута изменения значения ISR . Если для принятого корректного кода $INTR_i$ используется режим с подтверждениями и задано значение $Th \neq 0$ ($ISR_TOUTS2.LOC_TH$) и узел является обработчиком этого прерывания ($ISR_handler_term_funct(i)=1$), то запускается счет таймаута Th .

Если принятый корректный код обрабатывается в режиме с подтверждениями $ACK_NON_ACK_regime(i)=0$, обрабатывается в режиме терминального узла ($ISR_handler_term_funct(i)=1$ или $ISR_source_term_funct(i)=1$) и для него задан таймаут ожидания подтверждения $TOUT_CODE.LOC_Reset_T1_ack$, то запускается счет этого таймаута.

Если принятый корректный код обрабатывается в режиме с подтверждениями $ACK_NON_ACK_regime(i)=0$, обрабатывается в режиме коммутатора ($ISR_handler_term_funct(i)=0$ и $ISR_source_term_funct(i)=0$) и для него задан таймаут ожидания подтверждения $TOUT_CODE.LOC_Reset_SW_ack$, то запускается счет этого таймаута.

Если принятый корректный код обрабатывается в режиме без подтверждений $ACK_NON_ACK_regime(i)=1$, обрабатывается в режиме терминального узла ($ISR_handler_term_funct(i)=1$ или $ISR_source_term_funct(i)=1$) и для него задан таймаут ожидания подтверждения $ISR_TOUTS2.LOC_RESET_T1_nack$ (который в этом случае должен быть задан обязательно), то запускается счет этого таймаута.

Если принятый корректный код обрабатывается в режиме без подтверждений $ACK_NON_ACK_regime(i)=1$, обрабатывается в режиме коммутатора ($ISR_handler_term_funct(i)=0$ и $ISR_source_term_funct(i)=0$) и для него задан таймаут ожидания подтверждения $ISR_TOUTS2.LOC_Reset_SW_nack$ (который в этом случае должен быть задан обязательно), то запускается счет этого таймаута.

Если был запущен счет одного из таймаутов ожидания подтверждения, и до истечения таймаута подтверждение не поступило, то соответствующий разряд $ISR(i)$ будет сброшен автоматически.

Если был запущен счет таймаута T_h и $ISR_spec_term_funct(i)=1$, то по истечении этого таймаута в сеть автоматически отправляется код $INTA_i$, разряд $ISR(i)$ сбрасывается и счет всех таймаутов сбрасывается, а так же запускается счет таймаута $ISR_TOUTS2.LOC_ISR_CHANGE$ (если он задан)

Если код распределенного прерывания отправляется в сеть встроенным RISC ядром (через регистр TX_CODE), то выполняется следующая последовательность действий.

Проверяется значение регистра ISR . Если $ISR(i)=1$, то код отбрасывается без каких-либо дальнейших действий. Если $ISR(i)=0$ и используется режим с подтверждениями, то проверяется состояние счетчика таймаутов изменения значения для этого разряда ISR ($ISR_TOUTS2.LOC_ISR_CHANGE$). Если данный таймаут не задан, уже истек, или если $INTR_i$ отправляется первый раз после сброса, то проверяется значение таймаута T_g ($ISR_TOUTS2.LOC_TG$). Если данный таймаут не задан, уже истек или если $INTR_i$ отправляется первый раз после сброса, то данный $INTR_i$ считается корректным, $ISR(i)$ устанавливается в значение 1 и данный код отправляется в сеть SpaceWire (в порт SpaceWire). В противном случае данный код считается не корректным, он отбрасывается без каких-либо дальнейших действий.

Если $ISR(i)=0$ и используется режим без подтверждений, то данный код считается корректным, $ISR(i)$ устанавливается в значение 1 и данный код отправляется в сеть SpaceWire (в порт SpaceWire).

Если для данного корректного кода $INTR_i$ используется режим с подтверждениями и $ISR_TOUTS2.LOC_ISR_CHANGE$ не равен 0, то запускается счет таймаута изменения значения ISR .

Если данный корректный код обрабатывается в режиме с подтверждениями $ACK_NON_ACK_regime(i)=0$, обрабатывается в режиме терминального узла ($ISR_handler_term_funct(i)=1$ или $ISR_source_term_funct(i)=1$) и для него задан таймаут ожидания подтверждения $TOUT_CODE.LOC_Reset_T1_ack$, то запускается счет этого таймаута.

Если данный корректный код обрабатывается в режиме с подтверждениями $ACK_NON_ACK_regime(i)=0$, обрабатывается в режиме коммутатора ($ISR_handler_term_funct(i)=0$ и $ISR_source_term_funct(i)=0$) и для него задан таймаут ожидания подтверждения $TOUT_CODE.LOC_Reset_SW_ack$, то запускается счет этого таймаута. (!!! Надо отметить, что данная ситуация является некорректной с точки зрения настроек сети, однако при ненастроенном T_g , она не блокируется)

Если данный корректный код обрабатывается в режиме без подтверждений $ACK_NON_ACK_regime(i)=1$, обрабатывается в режиме терминального узла ($ISR_handler_term_funct(i)=1$ или $ISR_source_term_funct(i)=1$) и для него задан таймаут

ожидания подтверждения `ISR_TOUTS2.LOC_RESET_T1_nack` (который в этом случае должен быть задан обязательно), то запускается счет этого таймаута.

Если данный корректный код обрабатывается в режиме без подтверждений `ACK_NON_ACK_regime(i)=1`, обрабатывается в режиме коммутатора (`ISR_handler_term_funct(i)=0` и `ISR_source_term_funct(i)=0`) и для него задан таймаут ожидания подтверждения `ISR_TOUTS2.LOC_Reset_SW_nack` (который в этом случае должен быть задан обязательно), то запускается счет этого таймаута. (!!! Надо отметить, что данная ситуация является некорректной с точки зрения настроек сети, однако при ненастроенном `Tg`, она не блокируется)

Если был запущен счет одного из таймаутов ожидания подтверждения, и до истечения таймаута подтверждение не поступило, то соответствующий разряд `ISR(i)` будет сброшен автоматически.

При приеме кода распределенного прерывания/подтверждения из сети выполняются следующие действия.

Проверяется, не замаскирован ли прием кодов данного типа из сети. Если прием замаскирован, то код отбрасывается без каких-либо дальнейших действий.

Если прием кода не замаскирован, то начинается его обработка.

При приеме кода подтверждения `INTAi` проверяется значение регистра `ISR`. Если `ISR(i)=0`, то код отбрасывается без каких-либо дальнейших действий. Если `ISR(i)=0` и используется режим с подтверждениями, то проверяется состояние счетчика таймаутов изменения значения для этого разряда `ISR` (`ISR_TOUTS2.LOC_ISR_CHANGE`). Если данный таймаут не задан, уже истек, или если `INTAi` принимается первый раз после сброса, то данный `INTAi` считается корректным, `ISR(i)` устанавливается в значение 1 и данный код записывается в регистр `RX_CODE`. В противном случае данный код считается не корректным, он отбрасывается без каких-либо дальнейших действий.

Если используется режим без подтверждений, то данный код считается не корректным и отбрасывается без каких либо действий

Если для принятого корректного кода `INTAi` `ISR_TOUTS2.LOC_ISR_CHANGE` не равен 0, то запускается счет таймаута изменения значения `ISR`. Если для принятого корректного кода `INTAi` используется режим с подтверждениями и задано значение `Tg/=0` (`ISR_TOUTS2.LOC_TG`) и узел является источником этого прерывания (`ISR_source_term_funct(i)=1`), то запускается счет таймаута `Th`.

Если код подтверждения отправляется в сеть встроенным RISC ядром (через регистр `TX_CODE`), то выполняется следующая последовательность действий.

Проверяется значение регистра ISR. Если $ISR(i)=0$, то код отбрасывается без каких-либо дальнейших действий. Если $ISR(i)=0$ и используется режим с подтверждениями, то проверяется состояние счетчика таймаутов изменения значения для этого разряда ISR (`ISR_TOUTS2.LOC_ISR_CHANGE`). Если данный таймаут не задан, уже истек, или если $INTR_i$ отправляется первый раз после сброса, то проверяется значение таймаута T_h (`ISR_TOUTS2.LOC_TH`). Если данный таймаут не задан, уже истек или если $INTA_i$ отправляется первый раз после сброса, то данный $INTA_i$ считается корректным, $ISR(i)$ устанавливается в значение 1 и данный код отправляется в сеть SpaceWire (в порт SpaceWire). В противном случае данный код считается не корректным, он отбрасывается без каких-либо дальнейших действий.

Если $ISR(i)=0$ и используется режим без подтверждений, то данный код отбрасывается автоматически без каких-либо действий.

Если для данного корректного кода $INTA_i$ `ISR_TOUTS2.LOC_ISR_CHANGE` не равен 0, то запускается счет таймаута изменения значения ISR.

В регистре `ISR_tout_L`, `ISR_tout_H` разряды, соответствующие распределенным прерываниям, для которых истек таймаут ожидания, устанавливаются в 1. Для того, чтобы сбросить значение разряда, в него необходимо записать 1.

Регистр `ISR_H,L_reset` используется для сброса разрядов регистра ISR_i всех связанных с ними служебных флагов и счетчиков. Для сброса i разряда и связанных с ним флагов и счетчиков в i разряд данного регистра необходимо записать 1.

Данный регистр рекомендуется использовать при администрировании сети, если в результате изменения конфигурации сети (либо по каким-то другим причинам) необходимо изменить размеры таймаутов, роли терминальных узлов (переназначить терминальные узлы, которые будут являться источниками и обработчиками прерываний).

5.6.11 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 10 Мбит/с. После установки соединения в эти разряды регистра могут быть записаны другие значения (соответствующие скорости передачи от 5 до 300 МГц). 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 10 Мбит/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 5 до 300М Гц. При разрыве соединения переход на коэффициент TX_SPEED_10 выполняется автоматически, при повторной установке соединения переход на TX_SPEED так же выполняется автоматически.

5.6.12 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.

5.6.13 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 ± 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 0

5.7 Логика работы DMA SWIC

5.7.1 Каналы DMA SWIC

Каждый контроллер SWIC имеет свой 4-канальный контроллер DMA. Выбор контроллера осуществляется по базовому адресу, приведенному в Таблица 5.24.

Таблица 5.24. Базовые адреса контроллеров DMA SWIC

Условное обозначение контроллера	Базовый адрес
DMA SWIC0	0x150_0000
DMA SWIC1	0x170_0000
DMA SWIC2	0x190_0000
DMA SWIC3	0x1B0_0000

Перечень каналов DMA и их взаимный приоритет приведен в Таблица 5.25. .

Таблица 5.25. Перечень каналов DMA

Условное обозначение канала	Назначение канала
Каналы DMA SWIC0	
SWIC0_RX_DESC	Канал записи в память дескрипторов принимаемых пакетов
SWIC0_RX_DATA	Канал записи в память принимаемых слов данных
SWIC0_TX_DESC	Канал чтения из памяти дескрипторов передаваемых пакетов
SWIC0_TX_DATA	Канал чтения из памяти передаваемых слов данных
Каналы DMA SWIC1	
SWIC1_RX_DESC	Канал записи в память дескрипторов принимаемых пакетов
SWIC1_RX_DATA	Канал записи в память принимаемых слов данных
SWIC1_TX_DESC	Канал чтения из памяти дескрипторов передаваемых пакетов
SWIC1_TX_DATA	Канал чтения из памяти передаваемых слов данных
Каналы DMA SWIC2	
SWIC2_RX_DESC	Канал записи в память дескрипторов принимаемых пакетов
SWIC2_RX_DATA	Канал записи в память принимаемых слов данных
SWIC2_TX_DESC	Канал чтения из памяти дескрипторов передаваемых пакетов
SWIC2_TX_DATA	Канал чтения из памяти передаваемых слов данных
Каналы DMA SWIC3	
SWIC3_RX_DESC	Канал записи в память дескрипторов принимаемых пакетов
SWIC3_RX_DATA	Канал записи в память принимаемых слов данных
SWIC3_TX_DESC	Канал чтения из памяти дескрипторов передаваемых пакетов
SWIC3_TX_DATA	Канал чтения из памяти передаваемых слов данных

Канал DMA за один цикл занятия коммутатора SWITCH передает одно 64-разрядное слово данных за один такт частоты CLK. Если после передачи слова нет запросов от более приоритетных каналов DMA, то канал начинает передавать следующее слово данных.

5.7.2 Регистры DMA SWIC

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 самоинициализации (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.

Канала DMA портов передают данные по коммутатору Switch 64-разрядными словами.

32-разрядный индексный регистр IR содержат физический адрес DPRAM. После передачи каждого слова данных к индексу IR прибавляется смещение на одно 64-х разрядное слово.

Формат регистров управления и состояния CSR каналов DMA этих портов приведен в Таблица 5.26.

Таблица 5.26. 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. 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

Номер разряда	Условное обозначение	Назначение
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	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. 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. 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 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.

5.7.3 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 SWIC размещаются в памяти в двух последовательных 64-разрядных словах, следующим образом (в порядке возрастания адресов):

```

63 _____ 0
{ IR32,    -32 };
{ CSR32, CP32 } .

```

При необходимости каналы DMA могут инициализироваться программно. Для этого необходимо загрузить все необходимые регистры индекса и смещения, а затем регистр CSR. При загрузке регистра CSR бит RUN необходимо установить в единичное состояние. Следует отметить, что бит RUN может быть использован для приостановки канала DMA. Для этого в любой момент времени в него необходимо записать 0. Для продолжения работы соответственно в бит RUN необходимо записать 1. Бит RUN может быть использован также для приостановки выполнения цепочки, если при загрузке очередных параметров он будет равен 0. Для продолжения выполнения цепочки в бит RUN необходимо записать 1. Для удобства организации обмена только с битом RUN имеется специальный регистр.

5.7.4 Прерывания DMA

Канал DMA формирует прерывание в соответствующем регистре QSTR0 (при условии, если установлены соответствующие биты в регистре MASKR0) 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.

5.8 Примеры работы со SWIC

Для передачи пакета данных в канал связи SpaceWire необходимо:

- установить соединение с соседним устройством, подключенным к каналу связи SpaceWire;
- записать в память DPRAM пакет для передачи;
- сформировать дескриптор этого пакета;
- настроить передающий канал DMA данных;
- настроить передающий канал DMA дескриптора;
- запустить настроенные каналы DMA;
- дождаться окончания работы каналов DMA.

Для приема пакета данных из канала связи SpaceWire необходимо:

- установить соединение с соседним устройством, подключенным к каналу связи SpaceWire;
- настроить приемный канал DMA дескрипторов;
- настроить приемный канал DMA данных;
- запустить настроенные каналы DMA;
- дождаться окончания работы каналов DMA;
- считать принятый пакет данных из DPRAM.

Процедура передачи пакета:

```
SWIC_TX_SPEED = 0x302 //Устанавливаем скорость соединения 10Мбит/сек.

SWIC_MODE_CR = LINK_START // Запускаем соединение

SWIC_STATUS [7:5] = 3'B101? // Чтением регистра статуса ожидаем установления
соединения

// Готовим данные в памяти

DPRAM@ 0x01000100 = 0xA0000005 // Дескриптор на передачу, 5 байт к передаче.

DPRAM@ 0x01000200 = 0xA4A3A2A1 // данные на передачу

DPRAM@ 0x01000204 = 0xA8A7A6A5 // данные на передачу

// Настраиваем канал SWIC_TX_DESC

SWIC_CSR_Ch2 = 0x00012000// Одно слово к передаче + установка прерывания

SWIC_IR_Ch2 = 0x01000100 // Адрес, где лежит дескриптор пакета для передачи

SWIC_CP_Ch2 = 0x0 // отключаем цепочечные передачи

// Настраиваем канал SWIC_TX_DATA

SWIC_CSR_Ch3 = 0x00022000// Два слова к передаче

SWIC_IR_Ch3 = 0x01000200 // Адрес, где лежит первое слово пакета для передачи

SWIC_CP_Ch3 = 0x0 // отключаем цепочечные передачи

// Запускаем каналы передачи

SWIC_RUN_Ch2 = 0x1

SWIC_RUN_Ch3 = 0x1

// Ожидание завершения работы каналов

SWIC_RUN_Ch3[0] & SWIC_RUN_Ch2[0] = 0? // Чтением регистров статуса ожидаем
завершения работы каналов DMA.

Рассмотрим подробнее процедуру приема пакета (получаем тот пакет, что передавали
ранее):

// Предположим что соединение уже установлено

// Настраиваем канал SWIC_RX_DESC
```


SWIC_CSR_Ch0 = 0x00012000 // Одно слово к передаче (из SWIC в память MCB) + установка прерывания

SWIC_IR_Ch0 = 0x01000300 // Адрес, где сохранится дескриптор принятого пакета

SWIC_CP_Ch0 = 0x0 // отключаем цепочечные передачи

// Настраиваем канал SWIC_RX_DATA

SWIC_CSR_Ch1 = 0x00022000 // Два слова к передаче (из SWIC в память MCB) + установка прерывания

SWIC_IR_Ch1 = 0x01000400 // Адрес, где сохранится первое слово принятого пакета

SWIC_CP_Ch1 = 0x0 // отключаем цепочечные передачи

// Запуск каналов передачи

SWIC_RUN_Ch0 = 0x1

SWIC_RUN_Ch1 = 0x1

// Ожидание завершения работы каналов

SWIC_RUN_Ch0[0] & SWIC_RUN_Ch1[0] = 0? // Чтением регистров статуса ожидаем завершения работы каналов DMA.

// Разбираем принятые данные

Читаем DPRAM@ 0x01000300 = 0x2000_0005 ? // Принят пакет без ошибки, длиной 5 байт.

Читаем DPRAM@ 0x01000400 = 0xA4A3_A2A1 ? // Первые 4 байта пакета. Читаем DPRAM@ 0x01000404 = 0x0000_00A5 ? // Пятый байт пакета. Старшие три байта пусты.

6. КОНТРОЛЛЕР ETHERNET MAC 10/100 (EMAC)

6.1 Основные характеристики EMAC

Контроллер Ethernet MAC 10/100 (EMAC) предназначен для использования в качестве порта Ethernet для обмена данными через приемопередатчик PHY в сети Ethernet. Контроллер EMAC поддерживает обмен данными в сети Ethernet с быстродействием 10 Мбит/с или 100 Мбит/с.

Контроллер EMAC 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 2К 64-разрядных слов или 16К байт;
- в состав контроллера входит буферное FIFO принятых данных размером 2К 64-разрядных слов или 16К байт;
- запись буферного FIFO передаваемых данных обеспечивается 64-разрядным каналом DMA на запись – DMA_ENET_Ch1 (передача данных осуществляется с точностью до байта);
- чтение буферного FIFO принятых данных обеспечивается 64-разрядным каналом DMA на чтение – DMA_ENET_Ch0 (передача данных осуществляется с точностью до байта);
- передаваемый кадр MAC целиком помещается в буферное FIFO, поэтому при возникновении коллизии повторная передача кадра будет выполняться из буферного 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 отдельное буферное FIFO статусов принятых кадров MAC размером 64 слова статуса.

6.2 Структурная схема ЕМАС

Структурная схема контроллера ЕМАС приведена на Рисунок 6.1.

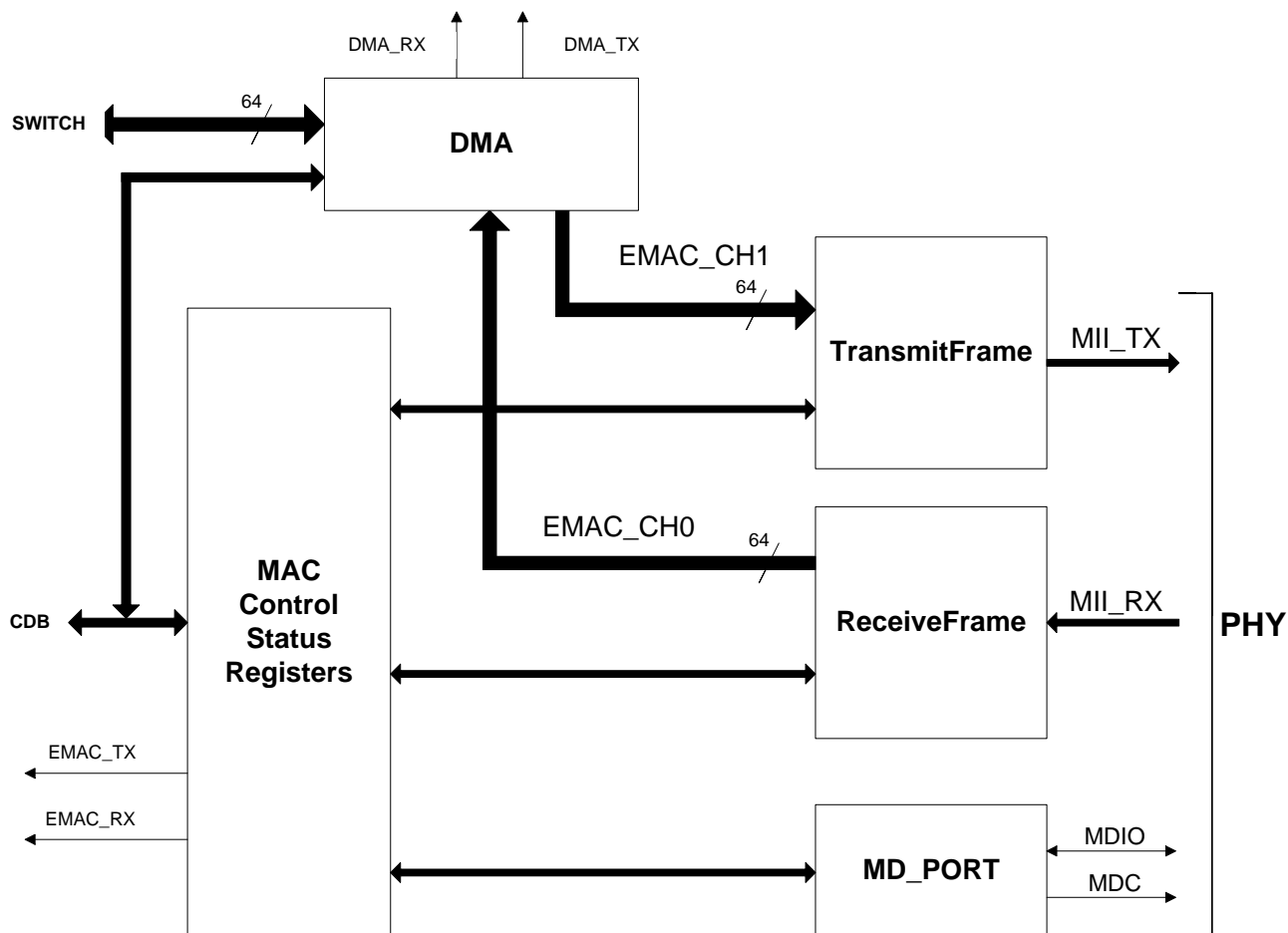


Рисунок 6.1. Структурная схема контроллера ЕМАС

Контроллер ЕМАС 10/100 включает:

- блок управления и состояния;
- контроллер DMA;
- блок передачи кадров – TransmitFrame;
- блок приема кадров – ReceiveFrame;
- порт управления PHY – MD_PORT.

Блок управления и состояния содержит регистры управления и состояния контроллера ЕМАС.

Контроллер DMA обеспечивает обмен данными по каналам EMAC_CH0 и EMAC_CH1 между буферными FIFO передаваемых/принятых данных и памятью (внешней или внутренней).

Блок передачи кадров TransmitFrame выполняет передачу кадров EMAC по шине MII. В состав блока передачи кадров входит передающее FIFO – TX_FIFO размером 16К 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 6.2 приведена структурная схема блока передачи кадров.

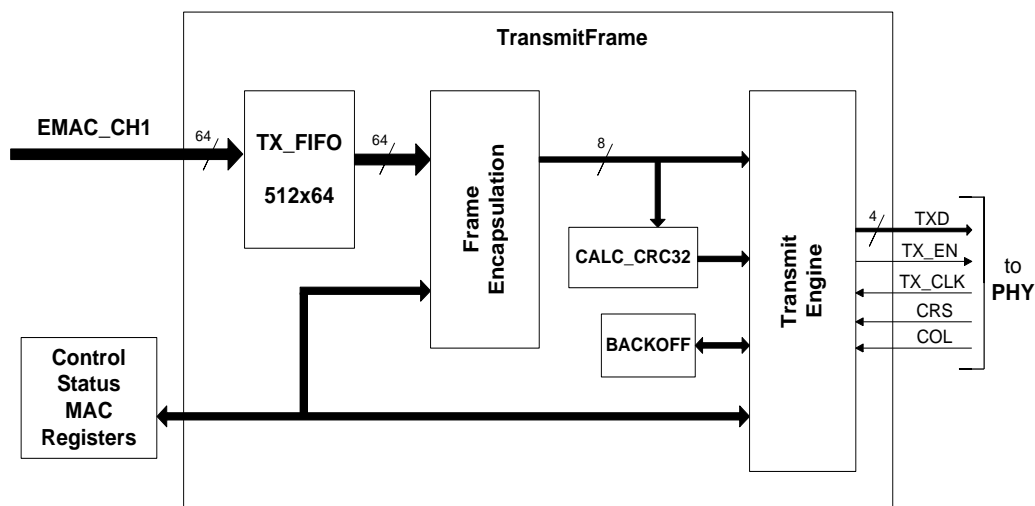


Рисунок 6.2. Структурная схема блока передачи кадров.

Блок приема кадров – ReceiveFrame – выполняет прием кадров EMAC по шине MII. В состав блока приема кадров входит принимающее FIFO – RX_FIFO размером 16К байт, блок распознавания адреса назначения принятого кадра EMAC – DADDR_CHECK, блок вычисления и проверки контрольной суммы принятого кадра – CRC32_CHECK, а также FIFO статусов принятых кадров размером 64 слова статуса.

На Рисунок 6.3 приведена структурная схема блока приема кадров.

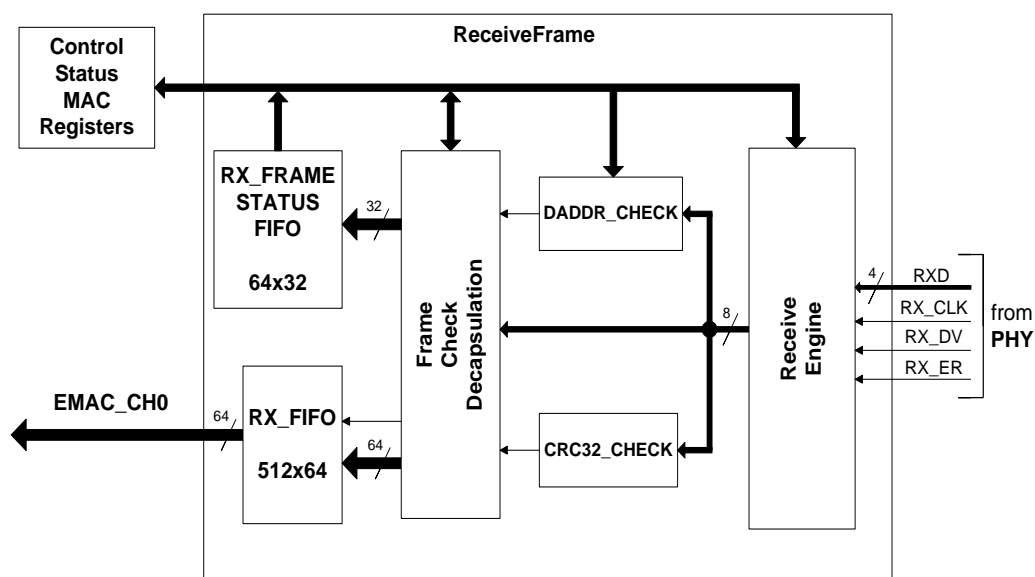


Рисунок 6.3. Структурная схема блока приема кадров.

Порт управления PHY – MD_PORT – выполняет обмен управляющими и статусными данными с приемопередатчиком PHY.

6.3 Регистры контроллера ЕМАС

В Таблица 6.1. приведен перечень программно-доступных регистров контроллера ЕМАС.

Таблица 6.1. Перечень регистров контроллера ЕМАС

Условное обозначение регистра	Название регистра	Тип доступа	Исходное состояние
CSR_EMAC_CH1[31:0]	Регистр управления и состояния канала DMA_ENET_Ch1	WR/RD	0000_0000
CP_EMAC_CH1[31:0]	Регистр указателя цепочки канала DMA_ENET_Ch1	WR/RD	не определено
IR_EMAC_CH1[31:0]	Регистр индекса канала DMA_ENET_Ch1	WR/RD	не определено
RUN_EMAC_CH1[0]	Псевдорегистр управления состоянием бита RUN регистра CSR_DMA_ENET_Ch1	WR	–
CSR_EMAC_CH0[31:0]	Регистр управления и состояния канала DMA_ENET_Ch0	WR/RD	0000_0000
CP_EMAC_CH0[31:0]	Регистр указателя цепочки канала DMA_ENET_Ch0	WR/RD	не определено
IR_EMAC_CH0[31:0]	Регистр индекса канала DMA_ENET_Ch0	WR/RD	не определено
RUN_EMAC_CH0[0]	Псевдорегистр управления состоянием бита RUN регистра CSR_DMA_ENET_Ch0	WR	–
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

Условное обозначение регистра	Название регистра	Тип доступа	Исходное состояние
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

6.3.1 Регистры DMA

6.3.1.1 Регистр управления и состояния канала DMA (CSR_EMAC_CH0/1)

Таблица 6.2. Формат регистра управления и состояния канала DMA

Номер разряда	Условное обозначение	Описание
0	RUN	Состояние работы канала DMA: 0 – состояние останова; 1 – состояние обмена данными. Доступен по чтению и записи. Значение в исходном состоянии – 0.
1	-	Резерв
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. Если в буфером FIFO осталось недостаточно места для записи, либо недостаточно слов для чтения, то значение WN будет автоматически уменьшено. Доступен по чтению и записи. Значение в исходном состоянии – 0000.
6	IPD	Запрет прерывания по запросу от порта при выключенном канале DMA (RUN=0). 0 – разрешено 1 – запрещено Доступен по чтению и записи. Значение в исходном состоянии – 0.
11:7	-	Резерв
12	CHEN	Признак разрешения самоинициализации (выполнения цепочки DMA передач). Доступен по чтению и записи. Значение в исходном состоянии – 0.

Номер разряда	Условное обозначение	Описание
13	IM	Разрешение установки признака окончания передачи блока данных: 0 – установки признака запрещено; 1 – установки признака разрешено. Доступен по чтению и записи. Значение в исходном состоянии – 0.
14	END	Признак окончания передачи блока данных. Аппаратно устанавливается в 1 после завершения передачи блока данных (при IM=1) Доступен по записи и чтению. Имеет два адреса чтения: один со сбросом бита по факту чтения, другой без сброса. Значение в исходном состоянии – 0.
15	DONE	Признак завершения передачи данных (одиночного блока либо последнего блока цепочки). Аппаратно устанавливается в 1 после завершения передачи цепочки блоков данных при CHEN=0, при этом бит RUN сбрасывается. Доступен по записи и чтению. Имеет два адреса чтения: один со сбросом бита по факту чтения другой без сброса. Состояние данного бита дублируется в соответствующий бит регистра QSTR по “или” с битом END. Значение в исходном состоянии – 0.
31:16	WC	Счетчик байт. Количество передаваемых байтов = WC + 1. . Количество передаваемых байтов должно быть кратно 8. Доступен по чтению и записи. Исходное состояние не определено.

6.3.1.2 Регистр указателя цепочки канала DMA (CP_EMAC_CH0/1)

Таблица 6.3. Формат регистра указателя цепочки канала DMA

Номер разряда	Условное обозначение	Описание
0	EN_CP	Разрешение запуска самоинициализации канала DMA. Доступен по чтению и записи. Значение в исходном состоянии – 0.
31: 1	CP[31:1]	Начальный адрес блока параметров для инициализации канала DMA (адрес должен быть выровнен по границе 64-разрядного слова). Доступен по чтению и записи. Исходное состояние не определено.

6.3.1.3 Регистр индекса канала DMA (IR_EMAC_CH0/1)

Таблица 6.4. Формат регистра индекса канала DMA

Номер разряда	Условное обозначение	Описание
31: 0	IR	Физический адрес памяти, начиная с которого выполняется обмен данными по каналу DMA. Должен быть выровнен по границе 64-разрядного слова. Доступен по чтению и записи. Исходное состояние не определено.

6.3.2 Регистры управления и состояния

6.3.2.1 Регистр управления (MAC_CONTROL)

Таблица 6.5. 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 заикливания 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	EN_ALGN_RX	Выбор режима работы с DMA блока приема кадров 0 — последовательный режим 1 — параллельный режим

Номер разряда	Условное обозначение	Описание
14	EN_ALGN_TX	Выбор режима работы с DMA блока передачи кадров 0 — последовательный режим 1 — параллельный режим

6.3.2.2 Регистр режима работы порта MD (MD_MODE)

Таблица 6.6. 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.

6.3.2.3 Регистр управления порта MD (MD_CONTROL)

Таблица 6.7. 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.

6.3.2.4 Регистр статуса порта MD (MD_STATUS)

Таблица 6.8. 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.

6.3.2.5 Регистр контрольной суммы кадра(FCS_CLIENT)

Таблица 6.9. 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.

6.3.2.6 Регистр типа кадра (TYPE)

Таблица 6.10. 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.

6.3.2.7 Регистр IFS и режима обработки коллизии (IFS_COLL_MODE)

Таблица 6.11. 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

Номер разряда	Условное обозначение	Описание
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 \square 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).

6.3.2.8 Регистр управления передачи кадра (TX_FRAME_CONTROL)

Таблица 6.12. Формат регистра управления передачи кадра

Номер разряда	Условное обозначение	Описание
11: 0	LENGTH	Если DisEncapFR = 0, то LENGTH – число байт поля <DATA> передаваемого кадра в передающем TX_FIFO. Если DisEncapFR = 1, то LENGTH – число байт передаваемого кадра в передающем TX_FIFO. Если DisEncapFR = 0 и TYPE_EN = 0, то LENGTH также задает значение поля <LENGTH/TYPE> передаваемого кадра. Доступен по чтению и записи. Значение LENGTH должно быть не нулевым. Значение в исходном состоянии – 000.
12	TYPE_EN	Если DisEncapFR = 0, то бит TYPE_EN задает в каком качестве используется поле <LENGTH/TYPE> в передаваемом кадре. Если TYPE_EN = 0, то – поле <LENGTH>; Если TYPE_EN = 1, то – поле <TYPE>. Доступен по чтению и записи. Значение в исходном состоянии – 0.
13	FCS_CLT_EN	Если FCS_CLT_EN = 0, то значение поля <FCS> вычисляет блок передачи кадров при передаче кадра; Если FCS_CLT_EN = 1, то значение поля <FCS> – уже вычисленная контрольная сумма CRC32, заданная в регистре FCS_CLIENT. Доступен по чтению и записи. Значение в исходном состоянии – 0.
14	DisEncapFR	Запрещает/разрешает режим формирования кадра в блоке передачи кадров. Если DisEncapFR = 0, то разрешен режим формирования кадра в блоке передачи кадров; Если DisEncapFR = 1, то в блок передачи кадров передается уже сформированный кадр. Доступен по чтению и записи. Значение в исходном состоянии – 0.
15	DisPAD	Запрещает/разрешает автоматическое добавление в кадр поля <PAD>, в случае когда число байт в поле <DATA> меньше 46 байт / число байт в кадре меньше 64 байт. Доступен по чтению и записи. Значение в исходном состоянии – 0.

Номер разряда	Условное обозначение	Описание
16	TX_REQ	Запрос на передачу кадра. По завершении обработки запроса на передачу бит TX_REQ автоматически сбрасывается. Доступен по чтению и записи. Во время обработки запроса на передачу кадра бит TX_REQ не доступен по записи. Значение в исходном состоянии – 0.

6.3.2.9 Регистр статуса передачи кадра (STATUS_TX)

Таблица 6.13. 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.
27:16	TXW	Число 64-разрядных слов в передающем TX_FIFO (округлено в большую сторону). TXW = 0x000 – FIFO пустое; TXW = 0x800 – FIFO полное. Доступен только по чтению. Значение в исходном состоянии – 000.

6.3.2.10 Регистр младшей части хэш-таблицы (HASHT_L)

Таблица 6.14. 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.

6.3.2.11 Регистр старшей части хэш-таблицы (HASHT_H)

Таблица 6.15. 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.

6.3.2.12 Регистр максимального размера принимаемого кадра (RX_FR_MaxSize)

Таблица 6.16. 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.

6.3.2.13 Регистр управления приема кадра (RX_FRAME_CONTROL)

Таблица 6.17. 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_FCSErr	Разрешение отбрасывания кадров с ошибкой проверки контрольной суммы. Доступен по чтению и записи. Значение в исходном состоянии – 0.

Номер разряда	Условное обозначение	Описание
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 ширококвещательным 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_MCMT	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.

6.3.2.14 Регистр статуса приема кадра (STATUS_RX)

Таблица 6.18. 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.
23:12	RXW	Число 64-разрядных слов в принимающем RX_FIFO (округлено в меньшую сторону). RXW = 0x000 – FIFO пустое; RXW = 0x800 – FIFO полное. Доступен только по чтению. Значение в исходном состоянии – 000.
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.
30	–	Резерв

Номер разряда	Условное обозначение	Описание
31	RX_FIFO_OVF_Err	Флаг переполнения принимающего RX_FIFO. Доступен по чтению и записи. Значение в исходном состоянии – 0.

6.3.2.15 FIFO статусов принятых кадров (RX_FRAME_STATUS_FIFO)

Статус принятого кадра RX_FRAME_STATUS доступен только по чтению.

Значение в исходном состоянии – 00000000.

Таблица 6.19. 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.
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	MCMT	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 широковещательного адреса назначения принятого кадра когда разрешен прием кадров с широковещательным адресом назначения.
26	ALL	Флаг распознавания адреса назначения принятого кадра, когда разрешен прием кадров с любым адресом назначения.

6.3.2.16 Регистр управления и состояния режима тестирования TX_FIFO (TX_TEST_CSR)

Таблица 6.20. 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
16:4	TM_TX_RDW	Число прочтенных 32-разрядных слов из TX_FIFO в режиме тестирования. Доступен только по чтению. Значение в исходном состоянии – 000.

6.3.2.17 Регистр управления и состояния режима тестирования RX_FIFO (RX_TEST_CSR)

Таблица 6.21. 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
16:4	TM_RX_WRW	Число записанных 32-разрядных слов в RX_FIFO в режиме тестирования. Доступен только по чтению. Значение в исходном состоянии – 000.

6.3.3 Регистры адресов

6.3.3.1 Регистр младшей части исходного адреса MAC (MAC_ADDR_L)

Таблица 6.22. 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.

6.3.3.2 Регистр старшей части исходного адреса MAC (MAC_ADDR_H)

Таблица 6.23. 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.

6.3.3.3 Регистр младшей части адреса назначения (DADDR_L)

Таблица 6.24. 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.

6.3.3.4 Регистр старшей части адреса назначения (DADDR_H)

Таблица 6.25. 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.

6.3.3.5 Регистр младшей части уникального адреса (UCADDR_L)

Таблица 6.26. Формат регистра младшей части уникального адреса

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.

6.3.3.6 Регистр старшей части уникального адреса (UCADDR_H)

Таблица 6.27. 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.

6.3.3.7 Регистр младшей части группового адреса (MCADDR_L)

Таблица 6.28. 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.

6.3.3.8 Регистр старшей части группового адреса (MCADDR_H)

Таблица 6.29. 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.

6.3.3.9 Регистр младшей части маски группового адреса (MCADDR_MASK_L)

Таблица 6.30. 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.

6.3.3.10 Регистр старшей части маски группового адреса (MCADDR_MASK_H)

Таблица 6.31. 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.

6.4 Описание логики работы EMAC

6.4.1 Контроллер DMA

Устройство DMA обеспечивает обмен данными по каналам DMA EMAC_CH0 и EMAC_CH1 между буферными FIFO передаваемых/принятых данных и памятью (внешней или внутренней).

Контроллер DMA обеспечивает по каналу EMAC_CH1 передачу данных из памяти DPRAM в передающий буфер – TX_FIFO.

Контроллер DMA обеспечивает по каналу EMAC_CH0 передачу данных из приемного буфера RX_FIFO в память DPRAM.

Для того чтобы запустить обмен по каналу DMA EMAC_CH0/1 необходимо установить в регистре CSR_EMAC_CH1/0 бит RUN=1. Бит RUN также доступен для записи по адресу псевдорегистра RUN_EMAC_CH0/1.

Перед тем как установить бит RUN необходимо проинициализировать регистры канала: IR_EMAC_CH0/1, CP_EMAC_CH0/1, CSR_EMAC_CH0/1.

При передаче данных по каналу DMA EMAC_CH1 необходимо установить бит EN_TX_DMA = 1 в регистре MAC_CONTROL, чтобы разрешить работу передающего TX_FIFO с каналом DMA.

Для запуска обмена по каналу DMA также может быть использована процедура самоинициализации. Для выполнения самоинициализации необходимо подготовить блок

параметров. Блок параметров должен располагаться в памяти DPRAM, состоять из двух последовательных 64-разрядных слов и содержать значения, которые требуется загрузить в регистры канала DMA:

	63	3231	0
1	IR_EMAC_CH0/1	-	
2	CSR_EMAC_CH0/1	CP_EMAC_CH0/1	

Начальный адрес блока параметров должен быть задан в регистре CP_EMAC_CH0/1. Для запуска процедуры самоинициализации канала необходимо установить бит 0 этого регистра.

Индексный регистр IR_EMAC_CH0/11 содержит физический адрес памяти DPRAM. Пересылка каждого слова данных осуществляется по адресу, заданному в регистре IR_EMAC_CH0/1 с его последующей инкрементацией: после передачи каждого слова данных к IR_EMAC_CH0/1 прибавляется смещение на количество переданных байт.

Взаимный приоритет каналов EMAC_CH0 и EMAC_CH1 изменяется циклически.

Канал DMA EMAC_CH1 формирует запрос на прерывание ETH_DMA_TX, а канал EMAC_CH0 - ETH_DMA_RX.

Запрос на прерывание от канала DMA возникает, если в регистре CSR_EMAC_CH биты END или DONE устанавливаются в 1.

Запрос на прерывание от канала DMA EMAC может быть сброшен чтением регистра CSR_EMAC_CH1/0 – при этом биты DONE и END автоматически обнуляются. Либо для сброса запроса на прерывание от канала DMA EMAC необходимо сбросить биты DONE и END записью в них нулевых значений.

6.4.2 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 РНУ 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 РНУ выполняет следующие операции:

- запись в регистр приемопередатчика РНУ;
- чтение регистра приемопередатчика РНУ.

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 РНУ, с которым выполняется обмен данными, задается в разрядах регистра управления порта MD_CONTROL[28:24] = PHY_ADDR.

Адрес регистра приемопередатчика РНУ, 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 РНУ 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 РНУ 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 РНУ по интерфейсу управления MD выполняется в соответствии с форматом кадра управления. Формат кадра управления представлен в Таблица 6.32.

Таблица 6.32. Формат кадра управления

Число бит	Название поля	Поле кадра управления	Значение при операции записи	Значение при операции чтения
32	Преамбула	PRE	1111...1111	1111...1111
2	Начало кадра	ST	01	01
2	Код операции	OP	01	10
5	Адрес PHY	PHYAD	PHY_ADDR	PHY_ADDR
5	Адрес регистра	REGAD	PHYREG_ADDR	PHYREG_ADDR
2	Разворот (turnaround)	TA	10	Z0
16	Данные	DATA	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.

6.4.3 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 6.4 приведен формат кадра MAC.

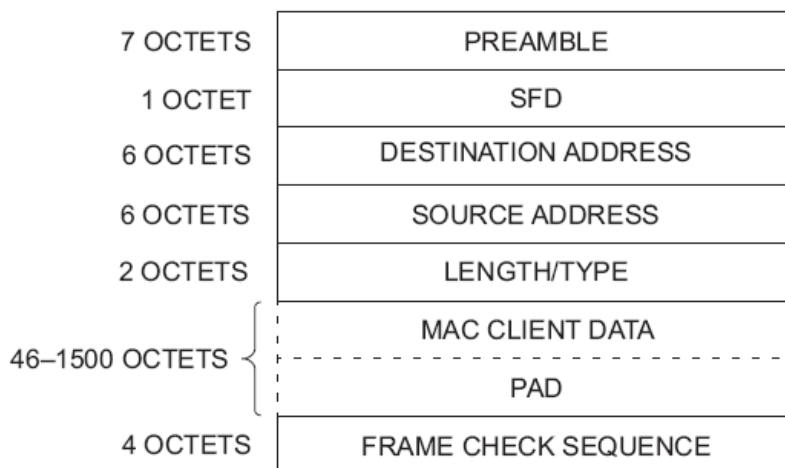


Рисунок 6.4. Формат кадра MAC

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.

6.4.3.1 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:

{MAC_ADDR_H, MAC_ADDR_L}	=>	поле <SOURCE ADDRESS>;
{DADDR_H, DADDR_L}	=>	поле <DESTINATION ADDRESS>;
TYPE	=>	поле <LENGTH/TYPE>, используемое как поле <TYPE>;
FCS_CLIENT	=>	поле <FCS> – уже вычисленная клиентом MAC контрольная сумма CRC32.

Разряды регистра TX_FRAME_CONTROL<11:0> = LENGTH => задают значение поля <LENGTH/TYPE>, используемое как поле <LENGTH>;

Содержание поля <DATA> передается по DMA-каналу на запись DMA_ENET_Ch1 в передающее FIFO – TX_FIFO – в виде последовательности 64-разрядных слов. Каждое 64-разрядное слово состоит из 8 байт поля <DATA>, начиная с байта, который должен быть передан первым, и заканчивая байтом, который должен быть передан последним:

63	5655	4847	4039	3231	2423	1615	87	0
byte _(n+7)	byte _(n+6)	byte _(n+5)	byte _(n+4)	byte _(n+3)	byte _(n+2)	byte _(n+1)	byte _(n)	
Байты передаются начиная с младшего								

В случае если последнее 64-разрядное слово поля <DATA> 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 заполняются произвольными (нулевыми) значениями.

Заполнение нулевыми значениями производится аппаратно, если работа осуществляется в последовательном режиме передачи (MAC_CONTROL[EN_ALGN_TX] == 0). В данном режиме процесс загрузки кадра в TX_FIFO и процесс передачи кадра из TX_FIFO в канал должны идти последовательно, по одному кадру. 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.

Если последовательный режим работы не устраивает, возможно его отключение (MAC_CONTROL[EN_ALGN_TX] == 1). При этом накладывает ограничение на объем передаваемых по DMA данных. Поле WCX регистра CSR_DMA_ENET_Ch* должно программироваться на количество байт кратное восьми, т. е. программист сам выравнивает поле <DATA> до полного 64-разрядного слова.

Бит регистра TX_FRAME_CONTROL<12> = TYPE_EN – задает в каком качестве используется поле <LENGTH/TYPE> в передаваемом кадре.

Если бит TYPE_EN=0, то в кадре используется поле <LENGTH> и его значение определяется разрядами TX_FRAME_CONTROL<11:0>.

Если бит TYPE_EN=1, то в кадре используется поле <TYPE> и его значение определяется значением регистра TYPE.

Независимо от значения бита TYPE_EN необходимо установить разряды регистра TX_FRAME_CONTROL<11:0> = LENGTH для задания числа байт в поле <DATA> передаваемого кадра – этот параметр используется блоком передачи кадров при передаче кадра. Значение LENGTH должно быть не нулевым.

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.

Если бит FCS_CLT_EN=1, то значение поля <FCS> – уже вычисленная клиентом MAC контрольная сумма CRC32, заданная в регистре FCS_CLIENT.

Бит регистра TX_FRAME_CONTROL<15> = Dis_PAD – запрещает/разрешает автоматическое добавление в кадр поля <PAD>, в случае когда число байт в поле <DATA> меньше 46 байт (минимальный размер поля <DATA> в соответствии со стандартом Ethernet).

Если бит Dis_PAD = 0, бит TX_FRAME_CONTROL<13> = FCS_CLT_EN = 0, а значение TX_FRAME_CONTROL<11:0> = LENGTH < 46 байт, 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.

6.4.3.2 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_ENET_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	5655	4847	4039	3231	2423	1615	87	0
byte _(n+7)	byte _(n+6)	byte _(n+5)	byte _(n+4)	byte _(n+3)	byte _(n+2)	byte _(n+1)	byte _(n)	
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.

Заполнение нулевыми значениями производится аппаратно, если работа осуществляется в последовательном режиме передачи ($MAC_CONTROL[EN_ALGN_TX] == 0$). В данном режиме процесс загрузки кадра в TX_FIFO и процесс передачи кадра из TX_FIFO в канал должны идти последовательно, по одному кадру. 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.

Если последовательный режим работы не устраивает, возможно его отключение ($MAC_CONTROL[EN_ALGN_TX] == 1$). При этом накладывает ограничение на объем передаваемых по DMA данных. Поле WCX регистра CSR_DMA_ENET_Ch* должно программироваться на количество байт кратное восьми, т.е. программист сам выравнивает поле <DATA> до полного 64-разрядного слова.

Кадр, переданный в TX_FIFO, должен быть сформирован в соответствии с форматом кадра MAC, приведенным на Рисунок 6.4 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 6.33.

Таблица 6.33. Структура кадра MAC, не включающего поле <FCS>

	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 _(LEN-1) , byte _(LEN-2) , byte _(LEN-3) , byte _(LEN-4) >			DATA<byte _(LEN-5) , byte _(LEN-6) , byte _(LEN-7) , byte _(LEN-8) >		
либо: N	0x00, DATA<byte _(LEN-1) , byte _(LEN-2) , byte _(LEN-3) >			DATA<byte _(LEN-4) , byte _(LEN-5) , byte _(LEN-6) , byte _(LEN-7) >		
либо: N	0x00, 0x00, DATA<byte _(LEN-1) , byte _(LEN-2) >			DATA<byte _(LEN-3) , byte _(LEN-4) , byte _(LEN-5) , byte _(LEN-6) >		
либо: N	0x00, 0x00, 0x00, DATA<byte _(LEN-1) >			DATA<byte _(LEN-2) , byte _(LEN-3) , byte _(LEN-4) , byte _(LEN-5) >		
либо: N	0x00, 0x00, 0x00, 0x00			DATA<byte _(LEN-1) , byte _(LEN-2) , byte _(LEN-3) , byte _(LEN-4) >		
либо: N	0x00, 0x00, 0x00, 0x00			0x00, DATA<byte _(LEN-1) , byte _(LEN-2) , byte _(LEN-3) >		
либо: N	0x00, 0x00, 0x00, 0x00			0x00, 0x00, DATA<byte _(LEN-1) , byte _(LEN-2) >		
либо: N	0x00, 0x00, 0x00, 0x00			0x00, 0x00, 0x00, DATA<byte _(LEN-1) >		

Где LEN – число байт в поле <DATA>: byte0, byte1, ..., byte_(LEN-1).

В случае, когда кадр, переданный в TX_FIFO, содержит уже вычисленное значение поля <FCS>, то кадр имеет следующую структуру, представленную в Таблица 6.34.:

Таблица 6.34. Структура кадра MAC, включающего поле <FCS>

	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 _(LEN-5) , byte _(LEN-6) , byte _(LEN-7) , byte _(LEN-8) >			DATA<byte _(LEN-9) , byte _(LEN-10) , byte _(LEN-11) , byte _(LEN-12) >		
N	FCS<31:0>			DATA<byte _(LEN-1) , byte _(LEN-2) , byte _(LEN-3) , byte _(LEN-4) >		
либо: N-1	DATA<byte _(LEN-4) , byte _(LEN-5) , byte _(LEN-6) , byte _(LEN-7) >			DATA<byte _(LEN-8) , byte _(LEN-9) , byte _(LEN-10) , byte _(LEN-11) >		
N	0x00, FCS<31:8>			FCS<7:0>, DATA<byte _(LEN-1) , byte _(LEN-2) , byte _(LEN-3) >		
либо: N-1	DATA<byte _(LEN-3) , byte _(LEN-4) , byte _(LEN-5) , byte _(LEN-6) >			DATA<byte _(LEN-7) , byte _(LEN-8) , byte _(LEN-9) , byte _(LEN-10) >		
N	0x00, 0x00, FCS<31:16>			FCS<15:0>, DATA<byte _(LEN-1) , byte _(LEN-2) >		
либо: N-1	DATA<byte _(LEN-2) , byte _(LEN-3) , byte _(LEN-4) , byte _(LEN-5) >			DATA<byte _(LEN-6) , byte _(LEN-7) , byte _(LEN-8) , byte _(LEN-9) >		
N	0x00, 0x00, 0x00, FCS<31:24>			FCS<23:0>, DATA<byte _(LEN-1) >		
либо: N-1	DATA<byte _(LEN-1) , byte _(LEN-2) , byte _(LEN-3) , byte _(LEN-4) >			DATA<byte _(LEN-5) , byte _(LEN-6) , byte _(LEN-7) , byte _(LEN-8) >		
N	0x00, 0x00, 0x00, 0x00			FCS<31:0>		
либо: N-1	FCS<7:0>, DATA<byte _(LEN-1) , byte _(LEN-2) , byte _(LEN-3) >			DATA<byte _(LEN-4) , byte _(LEN-5) , byte _(LEN-6) , byte _(LEN-7) >		
N	0x00, 0x00, 0x00, 0x00			0x00, FCS<31:8>		
либо: N-1	FCS<15:0>, DATA<byte _(LEN-1) , byte _(LEN-2) >			DATA<byte _(LEN-3) , byte _(LEN-4) , byte _(LEN-5) , byte _(LEN-6) >		
N	0x00, 0x00, 0x00, 0x00			0x00, 0x00, FCS<31:16>		
либо: N-1	FCS<23:0>, DATA<byte _(LEN-1) >			DATA<byte _(LEN-2) , byte _(LEN-3) , byte _(LEN-4) , byte _(LEN-5) >		
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, 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.

6.4.3.3 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_ENET_Ch1.

Для того чтобы разрешить работу передающего TX_FIFO с каналом DMA_ENET_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 точность 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: <DESTINATION ADDRESS>, <SOURCE ADDRESS>, <LENGTH/TYPE>, <DATA>, <FCS>.

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 <FCS>, завершает передачу кадра и затем 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 = 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<27:16> = 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 пакетов 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 6.5 приведен порядок обработки запроса на передачу кадра блоком передачи кадров.

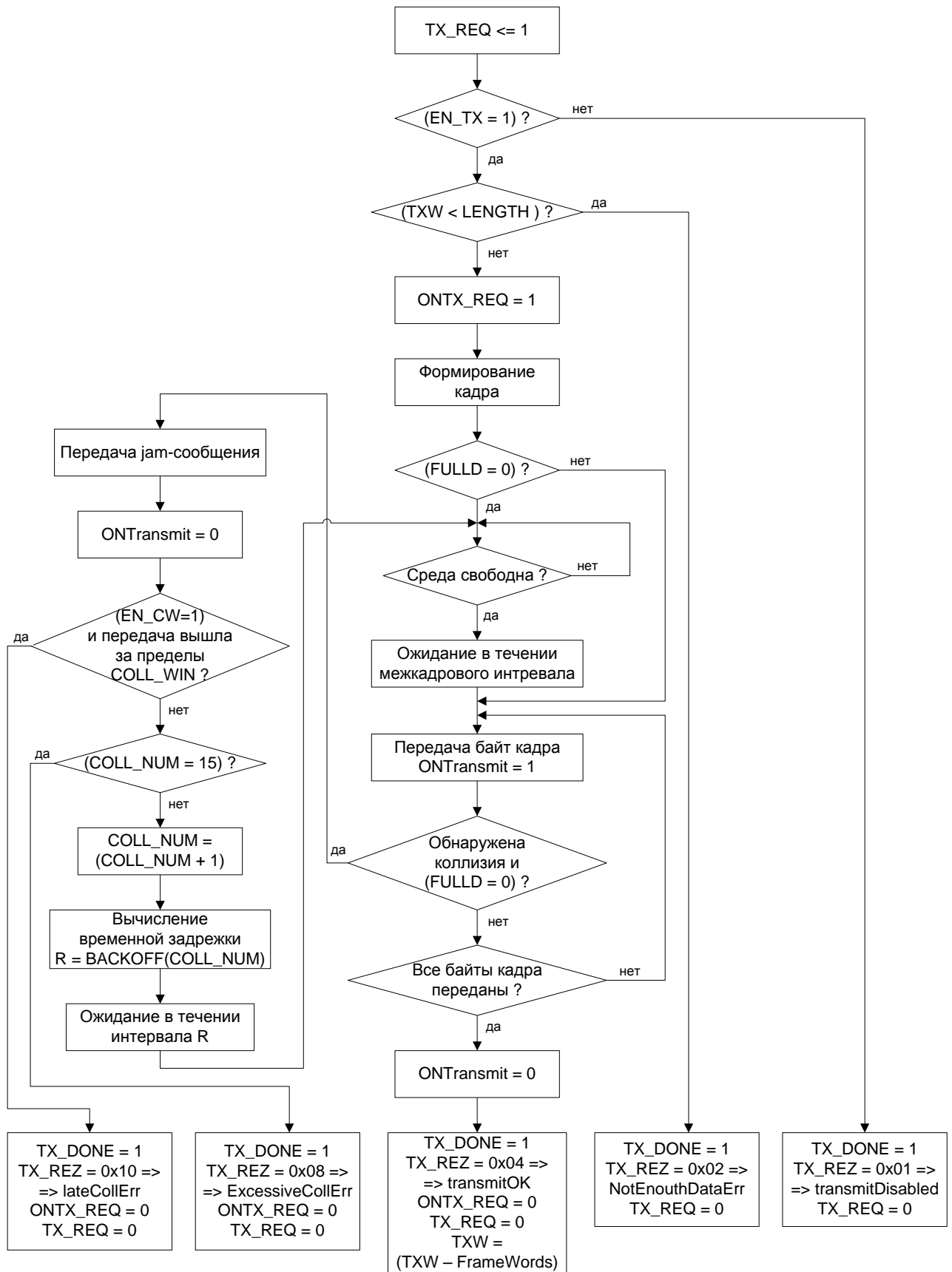


Рисунок 6.5. Порядок обработки запроса на передачу кадра

6.4.3.4 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 $IFS_COLL_MODE\langle 23:16 \rangle = COLL_WIN$. 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 $IFS_COLL_MODE\langle 4 \rangle = EN_CW = 1$. По умолчанию отслеживание окна коллизии разрешено.

В случае обнаружении коллизии во время передачи кадра, если разрешено отслеживание окна коллизии ($IFS_COLL_MODE\langle 4 \rangle = EN_CW = 1$), 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 ($IFS_COLL_MODE\langle 4 \rangle = EN_CW = 1$), 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 $STATUS_TX\langle 3 \rangle = TX_DONE = 1$.

По завершении обработки запроса на передачу кадра блок передачи кадров также сообщает результат передачи кадра в разрядах регистра статуса $STATUS_TX\langle 8:4 \rangle = TX_REZ = 0x10$ – $lateCollErr$ – ошибка поздней коллизии.

В случае, когда отслеживание окна коллизии не разрешено, то есть бит $IFS_COLL_MODE\langle 4 \rangle = EN_CW = 0$, 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 $STATUS_TX\langle 3 \rangle = TX_DONE = 1$, а также сообщает результат передачи кадра в разрядах регистра статуса $STATUS_TX\langle 8:4 \rangle = TX_REZ = 0x14$ – одновременно $transmitOK$ и $lateCollErr$ – передача кадра успешно выполнена и при этом ошибка поздней коллизии.

Когда контроллер MAC работает в дуплексном режиме (бит $MAC_CONTROL\langle 0 \rangle = FULLD = 1$), 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.

6.4.4 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>, ..., 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>, ..., CRC<30>, CRC<31>}.

6.4.5 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: $0 \leq R \leq 1$.

6.4.6 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<16:4> = TM_TX_RDW. После сброса бита разрешения режима тестирования TX_FIFO число прочтенных из TX_FIFO слов – TM_TX_RDW – обнуляется.

6.4.7 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	5655	4847	4039	3231	2423	1615	87	0
byte _(n+7)	byte _(n+6)	byte _(n+5)	byte _(n+4)	byte _(n+3)	byte _(n+2)	byte _(n+1)	byte _(n)	
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<31> = 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:

1. Если размер принятого кадра составляет меньше 18 байт, то такой кадр считается некорректным и блок приема кадров отбрасывает этот кадр.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.

7. Если в принятом кадре значение поля <LENGTH/TYPE> \leq 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.
8. 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.
9. 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>:

1. 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.
2. 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.
3. 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.
4. 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.
5. 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 – $RX_FRAME_STATUS<20> = FCS_Del = 1$.

6. 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 $STATUS_RX<23:12> = RXW$ (RXW содержит информацию о количестве данных в RX_FIFO с точностью до байта, но в регистре статуса отображена информация с точность 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_ENET_Ch0 .

Выгрузка данных из RX_FIFO , как и загрузка в TX_FIFO , может осуществляться двумя способами:

- последовательный режим ($MAC_CONTROL[EN_ALGN_RX] == 0$);
- параллельный режим ($MAC_CONTROL[EN_ALGN_RX] == 0$).

В последовательном режиме данные принятого кадра вычитываются по DMA-каналу чтения из принимающего RX_FIFO в виде последовательности 64-разрядных слов (с точностью до байта). Т.к. 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. В данном режиме необходимо очередность: чтение статуса → выгрузка кадра (строго по одному кадру).

В параллельном режиме данные принятого кадра вычитываются по DMA-каналу чтения из принимающего RX_FIFO в виде последовательности 64-разрядных слов (с точностью до восьми байт). Т.е. в случае, если длина кадра не кратна 64 бит, то кадр выкачивается из RX_FIFO вместе с дополняющими его нулевыми битами.

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 пачек 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 $-STATUS_RX<23:12> = RXW = 0$.

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 Рисунок 6.6 приведен порядок приема кадров блоком приема кадров.

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 пачек 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.

6.4.8 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 $DA<47:0> = 0xFFFFFFFF$, то такой адрес назначения является ширококвещательным. Если при этом не установлен бит запрещения приема кадров с ширококвещательным 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. 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: $DA\langle 47:0\rangle \&\{MCADDR_MASK_H\langle 15:0\rangle, MCADDR_MASK_L\langle 31:0\rangle\}$, также на значение группового адреса MAC накладывается маска: $\{MCADDR_H\langle 15:0\rangle, MCADDR_L\langle 31:0\rangle\} \&\{MCADDR_MASK_H\langle 15:0\rangle, MCADDR_MASK_L\langle 31:0\rangle\}$, а затем полученные замаскированные значения адресов сравниваются. При совпадении замаскированных адресов, адрес назначения считается распознанным и для принимаемого кадра устанавливается статусный флаг – $RX_FRAME_STATUS\langle 23\rangle = MCM = 1$.

Если принятый адрес назначения DA является групповым адресом ($DA\langle 0\rangle = 1$) и при этом установлен бит $RX_FRAME_CONTROL\langle 8\rangle = EN_MCMT = 1$, тогда по принятому 48-разрядному адресу назначения $DA\langle 47:0\rangle$ в блоке CRC32_CHECK вычисляется контрольная сумма $DA_CRC\langle 31:0\rangle$. Значение бита вычисленной контрольной суммы $DA_CRC\langle 31\rangle$ определяет младшая или старшая часть хэш-таблицы будет использоваться для распознавания адреса назначения. Если бит $DA_CRC\langle 31\rangle = 0$, то для распознавания адреса используется младшая часть хэш-таблицы, заданная в регистре HASHT_L. Если бит $DA_CRC\langle 31\rangle = 1$, то для распознавания адреса используется старшая часть хэш-таблицы, заданная в регистре HASHT_H. Значение пяти бит вычисленной контрольной суммы $DA_CRC\langle 30:26\rangle$ 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 – $RX_FRAME_STATUS\langle 24\rangle = MCMT = 1$.

На Рисунок 6.7 приведен порядок распознавания принятого адреса назначения.

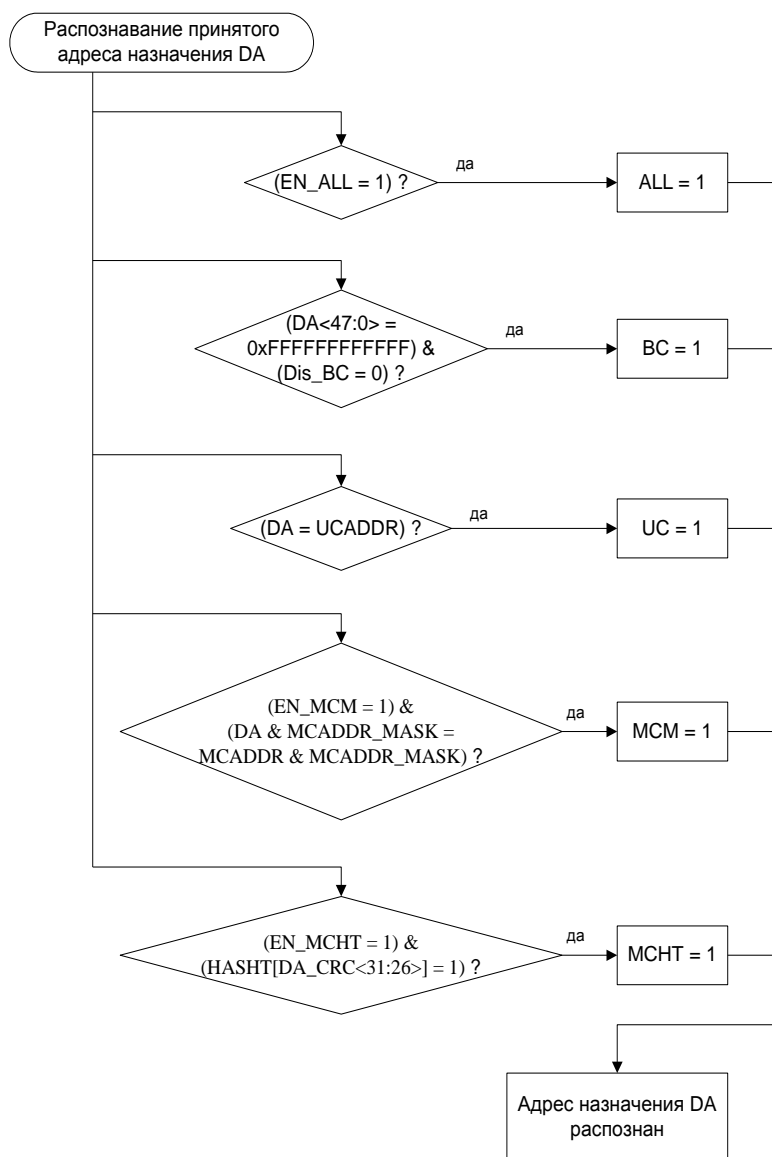


Рисунок 6.7. Порядок распознавания адреса назначения.

6.4.9 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>.

6.4.10 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<16:4> = 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`.

7. ПРИНЦИПЫ КОРРЕКЦИИ ОШИБОК

Для защиты памяти используется модифицированный код Хэмминга, то есть к контрольным разрядам по обычному коду Хэмминга добавляется общий разряд контроля четности.

Защищаемая кодом Хэмминга память DPRAM организована в виде двух отдельных блоков: основной блок для хранения данных и блок для хранения контрольных разрядов. Количество контрольных разрядов для 32-разрядных данных – 7 (см. Рисунок 7.1).

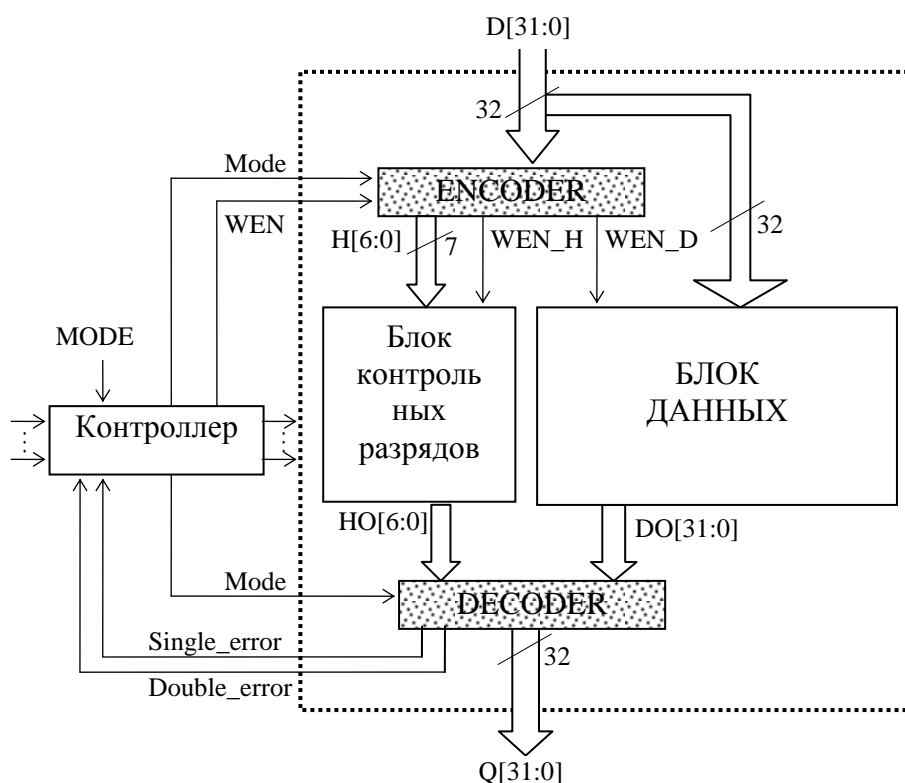


Рисунок 7.1. Структура 32-разрядного модуля памяти с коррекцией ошибок

Данные, записываемые в память, поступают на блок Encoder, который вычисляет контрольные разряды. При чтении из памяти данные поступают на блок Decoder, который анализирует контрольные разряды и определяет наличие одиночных и двойных ошибок в считанных данных либо одиночных ошибок в контрольных битах. Одиночные ошибки исправляются, двойные – фиксируются. Одновременно с достоверными данными (в случае отсутствия ошибок или коррекции одиночной ошибки) блок декодера формирует сигнал Single_Error (активный при наличии одиночной ошибки данных) или Parity_Error (активный при наличии ошибки в контрольном разряде общей четности). При обнаружении двойной ошибки, данные, не корректируются, но устанавливается в активный уровень сигнал Double_Error.

Модуль памяти имеет регистр управления и состояния CSR. Формат регистра CSR приведен в Таблица 7.1.

Таблица 7.1. Формат регистра CSR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
1:0	MODE	Режим работы памяти: 00 - режим без коррекции ошибок. Обмен данными выполняется только с блоком данных памяти; 01 - режим с коррекцией ошибок. В обмене данными участвуют блок данных и блок контрольных разрядов; 10 - режим тестирования блока контрольных разрядов; 11 - резерв.	W/R	0
2	NEMPTY	Признак наличия данных в FIFO ошибочных адресов	R	0
7:3	-	Резерв	-	0
15:8	Cnt_DERR	Счетчик двойных ошибок. При значении 255 останавливается. Прерывание сбрасывается при обнулении Cnt_DERR.	W/R	0
23:15	Num_SERR	Число одиночных ошибок данных, при котором формируется прерывание.	W/R	FF
31:24	Cnt_SERR	Счетчик одиночных ошибок. При значении 255 останавливается. Прерывание сбрасывается при $Cnt_CERR \leq Num_CERR$.	W/R	0

При отключенном режиме коррекции ошибок (MODE=0) запись осуществляется только в блок данных, содержимое блока контрольных разрядов остается неизменным. При чтении данные, считываемые из блока данных, поступают на выход напрямую в обход схемы коррекции ошибок. Сигналы Single_Error, Parity_Error и Double_Error не формируются.

Ошибки Single_Error и Parity_Error накапливаются в счетчике Cnt_SERR, а в FIFO ошибочных адресов имеют различные коды. Ошибки Double_Error накапливаются в счетчике Cnt_DERR. Прерывание формируется при $Cnt_CERR > Num_CERR$ или $Cnt_DERR > 0$. Для маскирования прерываний от одиночных ошибок Num_CERR устанавливается в состояние "FF" (т.к. Cnt_CERR не может быть больше значения "FF") при этом ошибочные адреса при возникновении Single_Error или Parity_Error в FIFO записываются.

Для целей тестирования предусматривается специальный режим (MODE=2), в котором запись данных с входной шины модуля памяти осуществляется в блок контрольных разрядов напрямую, минуя схему кодирования. Содержимое блока данных остается неизменным. При чтении из памяти на выходную шину поступают данные из блока контрольных разрядов. Старшие разряды дополняются нулями.

Основные режимы работы памяти приведены в Таблица 7.2.. Используются следующие обозначения: DI[31:0] – входная шина данных модуля, DO[31:0] – выход блока данных, H[6:0] – вход блока контрольных разрядов при 32-разрядной организации памяти, Q[31:0] – выходная шина данных модуля.

Таблица 7.2. Режимы работы памяти

MODE	Разрядность	Запись в блок данных	Запись в блок контрольных разрядов	Формирование выходной шины данных Q[31:0]
00	32	DI[31:0]	-	DO[31:0]
01	32	DI[31:0]	H[6:0]	DO[31:0] с коррекцией по H[6:0]
10	32	-	DI[6:0]	{25'h00000,HO[6:0]}
11	Резерв			

При байтовой организации памяти, запись в байтовый блок данных и соответствующий ему 7-разрядный блок контрольных разрядов производится при наличии активного сигнала разрешения записи в соответствующий байт (WEN[4]-WEN[0]). WEN[4] – запись контрольных битов. WEN[3]-WEN[0] – запись данных

Контроллер памяти формирует прерывание если:

- обнаружена двойная ошибка;
- содержимое счетчиков одиночных ошибок Cnt_SERR > Num_SERR.

Модуль памяти содержит блок FIFO ошибочных адресов AERROR, объемом 16 слов. В нем запоминаются адреса ячеек, в которых были обнаружены одиночные или двойные ошибки. FIFO доступно только по чтению. Формат слов в FIFO приведен в Таблица 7.3.

Таблица 7.3. Формат слова FIFO ошибочных адресов DPRAM

Номер разряда	Условное обозначение	Назначение
1:0	Code_ERR	Код ошибки. 00 – нет ошибки 01 – одиночная ошибка 10 - двойная ошибка 11 – ошибка в контрольном разряде общей четности
16:2	ADDR[16:2]	Адрес слова памяти, в котором произошла ошибка.
31:17	-	0

Прерывание по контролю кода Хемминга INT_HmDPRAM заведено в 30-й разряд регистров QSTR и QSTR_PCI.

8. ЭЛЕКТРИЧЕСКИЕ ПАРАМЕТРЫ

8.1 Электропитание

Номинальные значения напряжений питания:

- периферии PVDD (UCCP) – 3,3 В;
- ядра CVDD (UCCC) – 1,8 В.

Допустимые отклонения напряжений электропитания от номинальных значений – $\pm 5\%$.

Порядок включения и выключения микросхемы должен быть следующим:

- при включении микросхемы напряжение электропитания периферии UCCP необходимо подавать одновременно или после подачи напряжения электропитания ядра UCCC UCC2. Задержка между подачей напряжений электропитания должна быть не более 10 мс. Фронт нарастания напряжений электропитания до уровня 100% должен быть не более 1 мс. При этом, напряжение электропитания периферии должно достигать уровня 50% от номинального не ранее, чем напряжение электропитания ядра достигнет уровня 95% от номинального. Входные сигналы необходимо подавать одновременно или после подачи напряжения электропитания периферии;
- при выключении микросхемы необходимо одновременно отключить источники напряжения электропитания периферии и ядра. Входные сигналы необходимо снять раньше или одновременно с отключением источников напряжений электропитания.

Для фильтрации напряжений электропитания микросхемы, необходимо подключить к каждому источнику (UCCP и UCCC) не менее шести высокочастотных (low ESR) конденсаторов номиналом 0,1 мкФ типа СС 0603 Y5V 0,1 μ F Z 25V. Конденсаторы необходимо разместить по возможности равномерно по контуру корпуса микросхемы между выводами PVDD и GND, а так же CVDD и GND. При этом расстояние между контактами микросхемы и площадками подсоединения конденсаторов должно быть не более 3 мм.

8.2 Электрические параметры при приемке и поставке

Электрические параметры микросхемы приведены в Таблица 8.1.

Таблица 8.1. Электрические параметры микросхемы

Наименование параметра, единица измерения (режим измерения)	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Ток потребления периферии в статическом режиме, мА при: UCCP=3,47 В, UCCC=1,9 В, XTI=0	ICCP	–	10	от минус 60 до +85
Ток потребления ядра в статическом режиме, мА при: UCCP=3,47 В, UCCC=1,9 В, XTI=0	ICCC	–	30	
Ток потребления ядра в динамическом режиме, мА при: UCCP=3,47 В, UCCC=1,9 В	IOCCC	–	500	
Ток утечки высокого и низкого уровня на входе, мкА при: UCCP=3,47 В, UCCC=1,9 В	III	–	10	
Выходное напряжение низкого уровня, В при: IOL= 4 мА, UCCP=3,47 В	UOL	–	0,4	
Выходное напряжение высокого уровня, В при: IOH =-2,8 мА, UCCP =3,13 В	UOH	2,4	–	
Входная емкость, пФ	CI	–	30	25±10
Емкость входа/выхода, пФ	CI/O	–	30	

8.3 Предельно-допустимые и предельные электрические режимы эксплуатации

Значения предельно-допустимых и предельных электрических режимов эксплуатации микросхемы приведены в Таблица 8.2..

Таблица 8.2. Значения предельно-допустимых и предельных электрических режимов эксплуатации

Наименование параметра режима эксплуатации, единица измерения	Буквенное обозначение	Предельно допустимая норма при эксплуатации		Предельная норма	
		не менее	не более	не менее	не более
Напряжение питания периферии, В	UCCP	3,13	3,47	–	3,9
Напряжение питания ядра, В	UCCC	1,7	1,9	–	2,3
Входное напряжение высокого уровня, В	UOH	2,0	UCCP+0,2	–	UCCP+0,3
Входное напряжение низкого уровня, В	UOL	0	0,8	-0,3	–

9. ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ МИКРОСХЕМЫ

Микросхема 1892ХД4Ф имеет следующие группы выводов:

- шина обмена данными с внешним микропроцессором(68);
- шина PCI (58);
- 4 порта канала связи SpaceWire (32);
- порт Ethernet (17);
- управление (7);
- отладка и тестирование (25);
- электропитание(166).

Описание выводов приведено в Таблица 9.1.- Таблица 9.7. При описании выводов используются обозначения:

- I – вход;
- O – выход;
- IO – двунаправленный вход/выход с третьим состоянием;
- OD – выход с открытым коллектором, возможно объединение по схеме проводного ИЛИ;
- OT – выход с третьим состоянием. При низком уровне nRST находится в пассивном состоянии;
- префикс “n” (например, nWE) означает, что активный уровень сигнала - низкий. Иначе, у сигнала активный высокий уровень.

Неиспользуемые выводы типа "I" необходимо подключить к земле. Неиспользуемые выводы типа "O" необходимо оставить неподключенными. Каждый неиспользуемый вывод типа "IO" необходимо подключить к земле через резистор 10 кОм.

Таблица 9.1. Шина обмена данными с внешним микропроцессором

Название вывода	Количество	Тип	Назначение
A[24:0]	25	I	Шина адреса. Содержит адрес 32-разрядного слова в адресном пространстве микросхемы 1892ХД4Ф. При обмене данными через адаптер МВА внутренний адрес микросхемы 1892ХД4Ф формируется посредством добавления к разрядам A[22:0] двух нулей справа. Разряды A[24:23] используются для сравнения с номером N[1:0]. Следует иметь в виду, что по шине A[24:0] передается адрес 32-разрядного слова. Поэтому при сопряжении микросхемы 1892ХД4Ф с микропроцессорами серии «Мультикор» эту шину необходимо подключить к шине адреса микропроцессора начиная со 2 разряда (нулевой разряд шины A[24:0] подключить ко 2 разряду шины адреса микропроцессора и т.д.)
D[31:0]	32	IO	Шина данных.
nWE[3:0]	4	I	Сигналы записи байтов.
nRD	1	I	Сигнал чтения.
nACK	1	OD	Сигнал готовности данных.

Название вывода	Количество	Тип	Назначение
nCS[1:0]	2	I	Сигналы разрешения выборки: nCS[0] – для доступа к регистрам MBA и памяти DPRAM; nCS[1] – для доступа к регистрам SWIC, контроллеров PCI и Ethernet. Используя эти два сигнала можно работать с разным числом тактов ожидания. При этом сигнал nACK не используется. Если обмен данными выполняется с использованием сигнала nACK, то сигналы nCS[0] и nCS[1] необходимо объединить
nINT	1	O	Запрос прерывания: все незамаскированные прерывания регистра QSTR0/1 объединенные по ИЛИ
N[1:0]	2	I	Номер микросхемы. Используется при подключении нескольких микросхем 1892XD4Ф
Всего 68 выводов.			

Таблица 9.2. Управление

Название вывода	Количество	Тип	Назначение
PLL_EN	1	I	Разрешение работы PLL_SYS (см. Рисунок 2.3.): 0 – PLL_SYS отключена. Частота MBA_CLK равна частоте XTI/2. Тестовый режим работы микросхемы 1892XD4Ф. 1 – PLL_SYS включена. Частота MBA_CLK равна 200 МГц. Штатный режим работы микросхемы 1892XD4Ф
PLL_SEL[3:0]	4	I	Технологические выводы управления PLL_SYS. Должен быть установлен код 0xA
XTI	1	I	Вход опорной частоты 10 МГц +/-1%.
nRST	1	I	Сигнал установки исходного состояния
Всего 7 выводов			

Таблица 9.3. Отладка и тестирование

Название вывода	Количество	Тип	Назначение
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)
TCK_BSR	1	I	Тестовый тактовый сигнал (JTAG BSR)
TRST_BSR	1	I	Установка исходного состояния (JTAG BSR)
TMS_BSR	1	I	Выбор режима теста (JTAG BSR)
TDI_BSR	1	I	Вход данных теста (JTAG BSR)
TDO_BSR	1	O	Выход данных теста (JTAG BSR)
TEST[7:0]	8	O	Выход тестовых точек
TEST_SO[4:0]	5	O	Тестовые выходы
TEST_SI[4:0]	5	I	Тестовые входы
nTEST_SE	1	I	Вход разрешения тестирования, активный низкий уровень
nTEST_SMODE	1	I	Вход режима тестирования, активный низкий уровень
Всего 30 выводов			

Таблица 9.4. Порты каналов связи SpaceWire

Название вывода	Количество	Тип	Назначение
DINp[3:0]	4	I	Вход данных положительный
DINn[3:0]	4	I	Вход данных отрицательный
SINp[3:0]	4	I	Вход строба положительный
SINn[3:0]	4	I	Вход строба отрицательный
DOUTp[3:0]	4	O	Выход данных положительный

Название вывода	Количество во	Тип	Назначение
DOUn[3:0]	4	О	Выход данных отрицательный
SOUTp[3:0]	4	О	Выход строба положительный
SOUTn[3:0]	4	О	Выход строба отрицательный
Всего 32 вывода			

Таблица 9.5. Шина PCI

Название вывода	Количество во	Тип	Назначение
AD[31:0]	32	IO	Адрес/Данные
nCBE[3:0]	4	IO	Команда/ выбор байта
nFRAME	1	IO	Признак выполнения операции передачи данных
nIRDY	1	IO	Готовность устройства master
nTRDY	1	IO	Готовность устройства target
nSTOP	1	IO	Признак остановки передачи данных
PAR	1	IO	Дополнение до четности количества единиц на шинах AD и nCBE
nPERR	1	IO	Ошибка четности
nDEVSEL	1	IO	Подтверждения выборки
IDSEL	1	I	Выборка при доступе к конфигурационным регистрам
nREQ	1	OT	Запрос захвата шины
nGNT	1	I	Разрешение захвата шины
nINTA	1	OD	Прерывание. Все незамаскированные прерывания, поступающие на регистр QSTR_PCI, и бит CSR_PCI[0] объединяются по схеме ИЛИ, результат инвертируется и поступает на этот вывод
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 9.6. Порт Ethernet MAC

Название вывода	Количество во	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 МП
TX_EN	1	O	Признак передачи данных по интерфейсу МП
TXD[3:0]	4	O	Шина передаваемых данных по интерфейсу МП
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 МП
RX_DV	1	I	Признак наличия данных для приема по интерфейсу МП
RXD[3:0]	4	I	Шина принимаемых данных по интерфейсу МП
RX_ER	1	I	Признак обнаружения ошибки в принимаемых данных
Всего 17 выводов			

Таблица 9.7. Электропитание

Название вывода	Количество во	Назначение
CVDD	35	Напряжение электропитания ядра
PVDD	19	Напряжение электропитания входных и выходных драйверов
GND	108	Общий провод электропитания (земля)
Всего 166 выводов		

Нумерация выводов микросхемы 1892ХД4Ф в корпусе СРGA-416. приведена в Таблица 9.8.

Таблица 9.8. Нумерация выводов микросхемы 1892ХД4Ф в корпусе PGA-416

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
A1	GND	K17	GND	T17	GND
A2	GND	L10	GND	U10	GND
A13	GND	L11	GND	U11	GND
AA23	GND	L12	GND	U16	GND
AB23	GND	L13	GND	U17	GND
AC1	GND	L14	GND	Y24	GND
AC2	GND	L15	GND	Y25	GND
AC3	GND	L16	GND	Y26	GND
AC14	GND	L17	GND	A14	CVDD
AC23	GND	L24	GND	A25	CVDD
AC24	GND	L25	GND	A26	CVDD
AC25	GND	L26	GND	AC4	CVDD
AC26	GND	M11	GND	AC5	CVDD
AD1	GND	M12	GND	AC13	CVDD
AD2	GND	M13	GND	AD3	CVDD
AD14	GND	M14	GND	AD4	CVDD
AD24	GND	M15	GND	AD13	CVDD
AD25	GND	M16	GND	AE1	CVDD
AD26	GND	N11	GND	AE2	CVDD
AE14	GND	N12	GND	AE3	CVDD
AE24	GND	N13	GND	AE4	CVDD
AE25	GND	N14	GND	AE13	CVDD
AE26	GND	N15	GND	AF1	CVDD
AF3	GND	N16	GND	AF2	CVDD
AF4	GND	N26	GND	AF13	CVDD
AF14	GND	P1	GND	B14	CVDD
AF26	GND	P2	GND	B24	CVDD
B1	GND	P3	GND	B25	CVDD
B2	GND	P11	GND	C14	CVDD
B3	GND	P12	GND	C23	CVDD
B13	GND	P13	GND	C24	CVDD
B26	GND	P14	GND	C25	CVDD
C1	GND	P15	GND	D14	CVDD
C2	GND	P16	GND	D22	CVDD
C3	GND	R11	GND	D23	CVDD
C4	GND	R12	GND	K14	CVDD
C13	GND	R13	GND	K15	CVDD
C26	GND	R14	GND	M17	CVDD
D1	GND	R15	GND	N17	CVDD
D2	GND	R16	GND	P10	CVDD
D3	GND	R24	GND	R10	CVDD
D4	GND	R25	GND	U12	CVDD
D5	GND	R26	GND	U13	CVDD
D13	GND	T10	GND	A3	PVDD
G24	GND	T11	GND	AF24	PVDD
G25	GND	T12	GND	AF25	PVDD
G26	GND	T13	GND	B4	PVDD
K10	GND	T14	GND	C5	PVDD
K11	GND	T15	GND	K12	PVDD
K16	GND	T16	GND	K13	PVDD

Продолжение Таблица 9.8.

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
M10	PVDD	F4	TXD[3]	AA4	SOUTn[3]
M26	PVDD	F3	TXD[2]	AB1	SOUTp[3]
N1	PVDD	F2	TXD[1]	AB2	TDO
N2	PVDD	F1	TXD[0]	AB3	TMS
N3	PVDD	G4	TX_EN	AB4	TCK
N4	PVDD	G3	RX_CLK	AD5	TDI
N10	PVDD	G2	TX_CLK	AE5	TRST
P17	PVDD	G1	RX_DV	AF5	TRST_BSR
P26	PVDD	H4	RX_ER	AC6	TDI_BSR
R17	PVDD	H3	RXD[3]	AD6	TMS_BSR
U14	PVDD	H2	RXD[2]	AE6	TDO_BSR
U15	PVDD	H1	RXD[1]	AF6	TCK_BSR
		J4	RXD[0]	AC7	PCLK
		J3	MD	AD7	AD[31]
		J2	CRS	AE7	AD[30]
		J1	MDC	AF7	AD[29]
		K4	COL	AC8	AD[28]
		K3	DINn[0]	AD8	AD[27]
		K2	DINp[0]	AE8	AD[26]
		L4	SINp[0]	AF8	AD[25]
		L3	SINn[0]	AC9	AD[24]
		L2	DOUp[0]	AD9	AD[23]
		L1	DOUn[0]	AE9	AD[22]
		M4	SOUTn[0]	AF9	AD[21]
		M3	SOUTp[0]	AC10	AD[20]
		M2	DINn[1]	AD10	AD[19]
		M1	DINp[1]	AE10	AD[18]
		P4	SINp[1]	AF10	AD[17]
		R1	SINn[1]	AC11	AD[16]
		R2	DOUp[1]	AD11	AD[15]
		R3	DOUn[1]	AE11	AD[14]
		R4	SOUTn[1]	AF11	AD[13]
		T1	SOUTp[1]	AC12	AD[12]
		T2	DINn[2]	AD12	AD[11]
		T3	DINp[2]	AE12	AD[10]
		T4	SINp[2]	AF12	AD[9]
		U1	SINn[2]	AF15	AD[8]
		U2	DOUp[2]	AE15	AD[7]
		U3	DOUn[2]	AD15	AD[6]
		U4	SOUTn[2]	AC15	AD[5]
		V1	SOUTp[2]	AF16	AD[4]
		V3	XTI	AE16	AD[3]
		W1	PLL_EN	AD16	AD[2]
		W2	nRST	AC16	AD[1]
		Y2	DINn[3]	AF17	AD[0]
		Y3	DINp[3]	AE17	nCBE[3]
		Y4	SINp[3]	AD17	nCBE[2]
		AA1	SINn[3]	AC17	nCBE[1]
		AA2	DOUp[3]	AF18	nCBE[0]
		AA3	DOUn[3]	AE18	nGNTB[4]

Продолжение Таблица 9.8.

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
AD18	nGNTB[3]	J25	nCS0	B8	TEST_SO[4]
AC18	nGNTB[2]	J24	nWE[3]	C8	TEST_SO[3]
AF19	nGNTB[1]	J23	nWE[2]	D8	TEST_SO[2]
AE19	nGNTB[0]	H26	nWE[1]	A7	TEST+SO[1]
AD19	nGNT	H25	nWE[0]	B7	TEST_SO[0]
AC19	nREQB[4]	H24	nACK	C7	TEST_SI[4]
AF20	nREQB[3]	A22	D[31]	D7	TEST_SI[3]
AE20	nREQB[2]	D21	D[30]	A6	TEST_SI[2]
AD20	nREQB[1]	C21	D[29]	B6	TEST_SI[1]
AC20	nREQB[0]	B21	D[28]	C6	TEST_SI[0]
AF21	nDEVSEL	A21	D[27]	D6	TEST_SE
AE21	IDSEL	D20	D[26]	A5	TEST_SMODE
AD21	nIRDY	C20	D[25]		
AC21	nINTA	B20	D[24]		
AF22	nREQ	A20	D[23]		
AC22	nSTOP	D19	D[22]		
AF23	nPERR	C19	D[21]		
AE23	PAR	B19	D[20]		
AD23	nFRAME	A19	D[19]		
AB24	nTRDY	D18	D[18]		
W25	A[24]	C18	D[17]		
W26	A[23]	B18	D[16]		
V23	A[22]	A18	D[15]		
V24	A[21]	D17	D[14]		
V25	A[20]	C17	D[13]		
V26	A[19]	B17	D[12]		
U23	A[18]	A17	D[11]		
U24	A[17]	D16	D[10]		
U25	A[16]	C16	D[9]		
U26	A[15]	B16	D[8]		
T23	A[14]	A16	D[7]		
T24	A[13]	D15	D[6]		
T25	A[12]	C15	D[5]		
T26	A[11]	B15	D[4]		
R23	A[10]	A15	D[3]		
P23	A[9]	A12	D[2]		
P24	A[8]	B12	D[1]		
P25	A[7]	C12	D[0]		
N25	A[6]	A11	TEST[7]		
N24	A[5]	B11	TEST[6]		
N23	A[4]	C11	TEST[5]		
M25	A[3]	D11	TEST[4]		
M24	A[2]	A10	TEST[3]		
M23	A[1]	B10	TEST[2]		
L23	A[0]	C10	TEST[1]		
K26	nINT	D10	TEST[0]		
K25	N[1]	A9	PLL_SEL[3]		
K24	N[0]	B9	PLL_SEL[2]		
K23	nRD	C9	PLL_SEL[1]		
J26	nCS1	D9	PLL_SEL[0]		

10. ИСТОРИЯ ИЗМЕНЕНИЙ

10.1 5 сентября 2013 г

- скорректирован п. 2.2: частота MBA_CLK=200 МГц;
- уточнены табл. 2.1, 2.2, 2.3 в части регистра CLK_EN;
- описание формата регистра CLK_EN перенесено из п. 3.2 в п. 2.4;
- дополнена табл. 3.3;
- введено описание регистра RX_SPEED (п.5.5.7);
- скорректирован п. 5.6.4;
- уточнена табл. 6.2 в части разрядов 31:16;
- уточнена табл. 6.4;
- уточнена табл. 9.2;
- скорректированы пункты 5.6.6 и 5.8 в части необходимости устанавливать в единичное состояние 31 бит дескриптора для передачи данных в SWIC.

10.2 29 ноября 2013 г

- в табл. 5.7 уточнено описание поля COEFF_10.

10.3 30 декабря 2013 г

- в табл. 5.7 уточнено описание разрядов 19:10.

10.4 27 августа 2014 г

- в табл. 9.1 уточнено описание сигналов nCS[1:0];
- в разделе 5.5.1 скорректировано значение регистра HW_VER;
- в разделе 5.8 исправлено значение регистра TX_SPEED в момент установки соединения.

10.5 28 января 2016 г

- исправлен рисунок 3.5.