

МИКРОСХЕМА ИНТЕГРАЛЬНАЯ 1892ВК016

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

РАЯЖ.431295.002Д17





ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ	10
1.1 Назначение	10
1.2 Функциональные параметры и возможности	
1.3 Структурная схема	
1.4 Инструментальное программное обеспечение	13
1.5 Операционная система для микросхемы 1892ВК016	14
2. СИСТЕМНАЯ ОРГАНИЗАЦИЯ МИКРОСХЕМЫ	15
2.1 Карта памяти	15
2.2 Система синхронизации	
2.2.1 Входы синхронизации и умножители частоты	
2.2.2 Управление работой PLL	
2.2.3 Отключение и включение тактовой частоты	
2.3 Контроллер прерываний	29
2.4 Системные регистры	
2.5 Процедура начальной загрузки	
3. ПРОЦЕССОРЫ СРИ0, СРИ1	38
3.1 Основные характеристики	38
3.2 Блок-схема	
3.3 Составляющие логические блоки	
3.3.1 Устройство исполнения	
3.3.2 Устройство умножения/деления (MDU)	
3.3.3 Системный управляющий сопроцессор	
3.3.4 Устройство управления памятью (MMU)	
3.3.5 Контроллер кэш	
3.3.6 Устройство шинного интерфейса (BIU – Bus Interface Unit)	40
3.3.7 OnCD контроллер	
3.4 Конвейер	41
3.4.1 Стадии конвейера	41
3.4.2 Операции умножения и деления	43
3.4.3 Задержка выполнения команд перехода (Jump, Branch)	43
3.4.4 Обходные пути передачи данных (Data bypass)	44
3.4.5 Задержка загрузки данных	45
3.5 Устройство управления памятью (MMU)	
3.5.1 Введение	
3.5.2 Режимы работы	
3.5.3 Буфер быстрого преобразования адреса (TLB)	
3.5.4 Преобразование виртуального адреса в физический в режиме TLB	
3.6 Исключения	
3.6.1 Условия исключений	
3.6.2 Приоритеты исключений	
3.6.3 Расположение векторов исключений	62
3.6.4 Обработка общих исключений	
3.6.5 Исключения	
3.6.6 Алгоритмы обработки исключений	
3.7 Регистры СРО	
3.7.1 Назначение	75



3.7.2 Обзор регистров СРО	
3.7.3 Регистры СРО	
3.8 Кэш	94
4. ИНТЕРВАЛЬНЫЙ ТАЙМЕР	95
4.1 Назначение	95
4.2 Структурная схема IT	
4.3 Описание регистров интервального таймера	
4.4 Программирование ІТ	
5. КОНТРОЛЛЕР ПРЯМОГО ДОСТУПА В ПАМЯТЬ (DMA)	98
5.1 Перечень каналов DMA	98
5.2 Организация обмена данными в микросхеме	
5.3 Каналы DMA типа память-память	
5.4 Каналы DMA SWIC	103
5.5 Процедура самоинициализации	
5.6 Прерывания DMA	
6. ПОРТ ВНЕШНЕЙ ПАМЯТИ ТИПА DDR SDRAM (DDR_PORT)	108
6.1 Общие положения	108
6.2 Регистры DDR PORT	
6.2.1 Регистр конфигурации DDRAM	
6.2.2 Регистр параметров DDR_TMR	
6.2.3 Регистр состояний и управления SDRCSR	
6.2.4 Регистр режимов DDR_MOD	
6.2.5 Perистр DDR_EXT	115
6.2.6 Регистр DDR_ERR	116
7. УНИВЕРСАЛЬНЫЙ АСИНХРОННЫЙ ПОРТ (UART)	118
7.1 Общие положения	118
7.2 Регистры UART	119
7.2.1 Общие положения	119
7.2.2 Регистр LCR	119
7.2.3 Регистр FCR	120
7.2.4 Регистр LSR	121
7.2.5 Регистр IER	
7.2.6 Регистр IIR	
7.2.7 Регистр MCR	
7.2.8 Программируемый генератор скорости обмена	
7.3 Работа с FIFO по прерыванию	
7.4 Работа с FIFO по опросу	126
8. МОДУЛЬ SPINLOCK	127
9. МОДУЛЬ ОБМЕНА СООБЩЕНИЯМИ MAILBOX	129
9.1 Назначение	
9.2 Характеристики	
9.3 Регистры MAILBOX	
9.3.1 Регистр SOFT_RESET	
9.3.2 Регистр MESSAGE m	
9.3.3 Perucтр FIFO_STATUS_m	



9.3.4 Регистры SET_IRQ_WRITE0, SET_IRQ_WRITE1	
9.3.5 Регистры SET_IRQ_READ0, SET_IRQ_READ1	131
9.3.6 Регистры CLR_IRQ_WRITE0, CLR_IRQ_WRITE1	132
9.3.7 Регистры CLR_IRQ_READ0, CLR_IRQ_READ0	132
9.4 Методы организации межпроцессорного обмена данными	132
9.4.1 Обмен данными по прерыванию MAILBOX_READ	132
9.4.2 Обмен данными по прерыванию MAILBOX_WRITE	133
9.4.3 Обмен данными по опросу	133
10. КОНТРОЛЛЕР ИНТЕРФЕЙСА SPACEWIRE (SWIC)	134
10.1 Общие положения	
10.2 Блок схема	
10.3 Прерывания	
10.4 Перечень регистров SWIC	
10.4.1 Общие положения	
10.5 Описание регистров SWIC	
10.5.1 Perистр HW_VER	
10.5.2 Perucтр STATUS	
10.5.3 Perucтр RX_CODE	
10.5.4 Peгистр MODE_CR	
10.5.5 Peгистр TX_SPEED	
10.5.6 Регистр RX_SPEED	
10.5.7 Регистр ТХ_CODE	
10.5.8 Регистр CNT_RX_PACK	
10.5.9 Регистр CNT_RX0_PACK	146
10.5.10 Регистр ISR_L	146
10.5.11 Регистр ISR_H	147
10.5.12 Perистр TRUE_TIME	147
10.5.13 Perucтр TOUT CODE	147
10.5.14 Perистр ISR tout L	148
10.5.15 Perистр ISR tout H.	148
10.5.16 Регистр LOG ADDR	148
10.6 Работа со SWIC. Пакеты данных, дескрипторы пакетов	
10.6.1 Расположение данных в памяти	
10.6.2 Схема обработки данных процессором	
10.6.3 Прием данных из канала SpaceWire.	
10.6.4 Передача данных в канал SpaceWire	
10.6.5 Выравнивание границ пакетов по границам слов	
10.6.6 Формат дескриптора пакета	
10.6.7 Возможность передачи коммуникационного пакета	
10.6.8 Использование симплексного режима	
10.6.9 Маркеры времени	
10.6.10 Коды распределенных прерываний	157
10.6.11 Коды подтверждения распределенных прерываний	
10.6.12 Установка скорости передачи данных	
10.6.13 Установка екорости переда и данных	
10.6.14 Определение скорости приема данных	
11. KOMMYTATOP SPFR	
11.1 Функциональные параметры и возможности	
11.2 Типы информационных потоков	160



11.2.1 Введение	60
11.2.2 Потоки пакетов транспортных протоколов с частичной обработкой в	
SpFR1	
11.2.3 Потоки пакетов SpW без обработки в коммутаторе SpFR	
11.2.4 Потоки пакетов в пространство конфигурации коммутатора SpFR 1	
11.3 Перечень программно-доступных регистров коммутатора SpFR	
11.4 Описание программно-доступных регистров	
11.4.1 Регистр ID_VER 1	
11.4.2 Perucтр ID SWITCH	
11.4.3 Perucтр DMA_CHANNELS_MODE 1	69
11.4.4 Perucтр DMA_CHANNELS_WORK_EN 1	
11.4.5 Perucтр DMA_RMAP_CONFIGURATION 1	
11.4.6 Perucтр DMA REGIONS 12 STATUS	
11.4.7 Perucтр DMA REGIONS 34 STATUS	74
11.4.8 Регистр DMA_PACK_STATUS	74
11.4.9 Perucтр DMA IRQ MASK 1	75
11.4.10 Регистр DMA_IRQ_MASK21	76
11.4.11 Регистр DMA_IRQ_MASK3 1	76
11.4.12 Регистр DMA START ADDR R1	
11.4.13 Регистр DMA AREA SIZE R11	
11.4.14 Регистр DMA START ADDR R2	
11.4.15 Регистр DMA AREA SIZE R2	
11.4.16 Регистр DMA_START_ADDR_R31	
11.4.17 Регистр DMA AREA SIZE R3	
11.4.18 Регистр DMA START ADDR R4	
11.4.19 Регистр DMA_AREA_SIZE_R4	
11.4.20 Регистр FIFO ANS	
11.4.21 Perucтр FIFO_STATUS	80
11.4.22 Регистр REJ STATUS	81
11.4.23 Perucтр RMAP_REC_PACK_COUNTERS_1 1	
11.4.24 Perистр RMAP_REC_PACK_COUNTERS_21	
11.4.25 Регистр ID NET 1	
11.4.26 Регистр MODE_R 1	85
11.4.27 Peгистр MODE_R1 1	
11.4.28 Регистр STATE_R1	86
11.4.29 Perucтр STATE_R_CONNECT	.87
11.4.30 Perucтр STATE_R_ERROR	
11.4.31 Регистр IRQ_MASK1	88
11.4.32 Регистр GiGA_SPW_STATUS	90
11.4.33 Регистр GIGA SPW MODE	
11.4.34 Регистр GIGA_SPW_TRANSMISSION_PARAMETERS 1	.91
11.4.35 Регистр GIGA_SPW_CREDIT_PARAMETERS 1	.92
11.4.36 Регистр РМА_STATUS	
11.4.37 Регистр РМА_MODE 1	93
11.4.38 Peгистр SpF_PORT_MODE_VC_PARAMS 1	
11.4.39 Регистр SpF_PORT_MODE_VC_TSLOTS_L	
11.4.40 Peгистр SpF_PORT_MODE_VC_TSLOTS_H 1	95
11.4.41 Peгистр SpF_PORT_MODE_VC_FR_LL 1	
11.4.42 Peгистр SpF_PORT_MODE_LANE_CONSTANTS 1	
11.4.43 Peгистр SpF_PORT_STATUS_VC11	



	19/
11.4.45 Peгистр SpF_PORT_CUR_CREDIT_VALUE	198
11.4.46 Регистр SpF_PORT_STATUS_LL_EL	198
11.4.47 Регистр ADG	199
11.4.48 Peгистр CHANNEL_ROUTING	199
11.4.49 Peгистр CHANNEL_REGISTERS	200
11.4.50 Регистр DRT_TOUTS	200
11.4.51 Регистр DRT_TOUTS_VALIDS	201
11.4.52 Peгистр DRT_TOUTS_FLS	201
11.4.53 Регистр DA_TOUTS	201
11.4.54 Регистр DA_SPF_TOUTS	202
11.4.55 Регистр DA_SPF_TOUTS2	
11.4.56 Регистр DA_SPF_TOUTS_FLS	203
11.4.57 Регистр INCORRECT_VC_INFO	204
11.4.58 Регистр SPEC_VC_ARB_PARAMETERS	
11.4.59 Регистр SPEC_ARB	205
11.4.60 Регистр CONTROL_CONNECTION	
11.5 Описание логики работы контроллеров транспортного уровня, DMA комм	иутатора
206	
11.6 Конфигурирование областей памяти для виртуальных каналов, предназна	
для обращений в пространство конфигурации или команд SpW	
11.7 Конфигурирование областей памяти для виртуальных каналов, предназна	
для обработки пакетов Транспортных протоколов	
11.8 Перечень прерываний, формируемых коммутатором SpFR	218
12. МНОГОФУНКЦИОНАЛЬНЫЙ БУФЕРИЗИРОВАНННЫЙ	
	226
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP)	
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP)	226
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP	226 226
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро	226 226 эванного
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP)	226 226 эванного 227
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах	226 226 эванного 227 228
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP	226 226 ованного 227 228 229
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP	226 226 ованного 227 228 229 230
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP	226 226 ованного 227 228 229 230 232
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2 Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI	226 226 ованного 227 228 229 230 232
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2 Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI)	226 226 ованного 227 228 229 230 232 232 233
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2 Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI)	226 226 ованного 227 228 229 230 232 232 233 234
12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2 Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI)	226 226 ованного 227 228 229 230 232 232 233 234 235
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2 Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI) 12.2.5 Регистр управления передатчиком TCTR (режим SPI)	226 226 ованного 227 228 229 230 232 232 233 234 235 237
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2.1 Назначение последовательного порта в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI) 12.2.5 Регистр управления передатчиком TCTR (режим SPI) 12.2.6 Регистр состояния приёмника RSR (режим SPI)	226 226 ованного 227 228 229 230 232 232 233 234 235 237 239
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2 Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI) 12.2.5 Регистр управления передатчиком TCTR (режим SPI) 12.2.6 Регистр состояния приёмника RSR (режим SPI) 12.2.7 Регистр состояния передатчика TSR (режим SPI)	226 226 ованного 227 228 229 230 232 232 233 234 235 237 239 240
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2 Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI) 12.2.5 Регистр управления приёмником TCTR (режим SPI) 12.2.6 Регистр состояния приёмника RSR (режим SPI) 12.2.7 Регистр состояния приёмника RSR (режим SPI) 12.2.8 Структурная схема MFBSP для режима SPI	226 226 ованного 227 228 229 230 232 232 233 234 235 237 239 240
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2. Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI) 12.2.5 Регистр управления передатчиком TCTR (режим SPI) 12.2.6 Регистр состояния приёмника RSR (режим SPI) 12.2.7 Регистр состояния передатчика TSR (режим SPI) 12.2.8 Структурная схема MFBSP для режима SPI 12.2.9 Соединение порта с внешним устройством	226 226 ованного 227 228 229 230 232 232 233 234 235 237 239 240 241
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2. Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI) 12.2.5 Регистр управления передатчиком TCTR (режим SPI) 12.2.6 Регистр состояния приёмника RSR (режим SPI) 12.2.7 Регистр состояния передатчика TSR (режим SPI) 12.2.8 Структурная схема MFBSP для режима SPI 12.2.9 Соединение порта с внешним устройством 12.2.10 Передача данных в режиме SPI	226 226 ованного 227 228 229 230 232 233 234 235 237 239 240 241 242
12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2 Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI) 12.2.5 Регистр управления приёмником TCTR (режим SPI) 12.2.6 Регистр состояния приёмника RSR (режим SPI) 12.2.7 Регистр состояния приёмника RSR (режим SPI) 12.2.8 Структурная схема MFBSP для режима SPI 12.2.9 Соединение порта с внешним устройством 12.2.10 Передача данных в режиме SPI 12.2.11 Пример чтения 8 разрядного слова по заданному адресу из в	226 226 ованного 227 228 229 230 232 233 234 235 237 239 240 241 242 eдомого
ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP) 12.1 Общие сведения об MFBSP 12.1.1 Режимы работы MFBSP 12.1.2 Структурная схема многофункционального буферизиро последовательного порта 12.1.3 Назначение выводов порта в различных режимах 12.1.4 Перечень регистров MFBSP 12.1.5 Прерывания от MFBSP 12.2. Работа MFBSP в режиме SPI 12.2.1 Назначение последовательного порта в режиме SPI 12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI) 12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI) 12.2.4 Регистр управления приёмником RCTR (режим SPI) 12.2.5 Регистр управления передатчиком TCTR (режим SPI) 12.2.6 Регистр состояния приёмника RSR (режим SPI) 12.2.7 Регистр состояния передатчика TSR (режим SPI) 12.2.8 Структурная схема MFBSP для режима SPI 12.2.9 Соединение порта с внешним устройством 12.2.10 Передача данных в режиме SPI	226 226 ованного 227 228 229 230 232 232 233 234 235 237 239 240 241 242 242 едомого 244



12.2.13 Формирование управляющих сигналов приемника и передатчик	•
12.2.14 Тракт передачи данных	
12.2.15 Тракт приёма данных	
12.2.16 Прерывания от последовательного порта	
12.3 Работа MFBSP в режиме порта ввода-вывода общего назначения	
12.3.1 Регистр данных порта ввода вывода GPIO_DR	
12.3.2 Регистр управления направлением выводов DIR_MFBSP	252
13. NAND FLASH CONTROLLER	254
13.1 Общие положения	254
13.1.1 Функциональные параметры и возможности	254
13.1.2 Cостав NFC	255
13.1.3 Термины и определения	255
13.2 Регистры NFC	
13.2.1 Перечень регистров NFC	
13.2.2 Регистр РАСКЕТ	258
13.2.3 Регистр MEMADDR1	
13.2.4 Регистр MEMADDR2	
13.2.5 Регистр COMMAND.	
13.2.6 Регистр PROGRAM	
13.2.7 Perucтр INTERRUPT_STATUS_EN	
13.2.8 Регистр INTERRUPT_SIGNAL_EN	267
13.2.9 Регистр INTERRUPT_STATUS	
13.2.10 Регистр ID1	
13.2.11 Регистр ID2	
13.2.12 Perucтр FLASH_STATUS	
13.2.13 Peructp TIMING	
13.2.14 Perистр BUFFER_DATA	
13.2.15 Peructp ECC	
13.2.16 Peructp ECC_ERROR_CNT	
13.2.17 Peructp ECC_SPARE_CMD.	
13.2.18 Perистры ERROR_CNT_nBIT	
13.2.19 Peructp DMA_ADDR	
13.2.20 Peructp DMA_BUFFER_BOUNDARY	
13.2.21 Peructp WRITE_PROTECT	
13.3 Описание выполнения операций с флэш-памятью	
13.3.1 Схема выполнения команды «Без доступа к FIFO»	
13.3.2 Схема выполнения команды «С чтением из FIFO»	
13.3.3 Схема выполнения команды «С записью в FIFO»	
13.3.4 Команда Reset	
13.3.6 Команда Read ID	
13.3.7 Команда Read Parameter Page	
13.3.9 Команда Page Program с ECC	
13.3.10 Команда Раде Program для spare области	
13.3.11 Команда Page Program для spare области	
13.3.12 Команда Read c ECC	
13.3.13 Команда Read для spare области	
10.0.10 Itomaniqu Itoma gain upuro uvauerin	



	13.3.14 Команда Block Erase	288
	13.3.15 Команда Multi-plane Page Program	288
	13.3.16 Команды Multi-plane Copyback	289
	13.3.17 Команда Multi-plane Block Erase	290
	13.3.18 Команда Multi-plane Read	291
	13.3.19 Команда Multi-plane Read с ECC	292
	13.3.20 Команда Page Cache Program (2 interleaved адреса)	294
	13.3.21 Команда Page Cache Program	
	13.3.22 Команда Read Cashe Sequential	
	13.3.23 Команда Read Cashe Random	
	13.3.24 Выбор LUN/die	
	13.3.25 Команда Read Status Enhanced	
	13.3.26 Page Program LUN0, Page Program LUN1	
	13.3.27 Read LUN0, Read LUN1	
	13.3.28 Block Erase LUN0, Block Erase LUN1	
	13.3.29 Page Program LUN0, Read LUN1	
	13.3.30 Small Data Move	
	13.3.31 Small Data Move, Page Program	
	13.3.32 Small Data Move, Copyback Program	
	13.3.33 Small Data Move, Change Read Column Enhanced	
	13.3.34 Команда Change Row Address	
	13.3.35 Команда Change Row Address	
	13.3.36 Свойство Page Register Clear Enhancement	
	13.3.37 Команда Set Features	
	13.3.38 Команда Get Features	
	13.3.39 Переключение временных режимов	
	13.3.40 Page Program (режим MDMA)	
	13.3.41 Read (режим MDMA)	
	.4 Обнаружение и инициализация флэш-памяти	
	.5 Настройка тактирования NFC	
	.6 Прерывания	
	.7 Подключение NAND флэш-памяти	
	13.7.1 Последовательный способ работы	
	13.7.2 Параллельный способ работы	
14.	ПРИНЦИПЫ КОРРЕКЦИИ ОШИБОК	311
15.	ПОРТ ЈТАС И ВСТРОЕННЫЕ СРЕДСТВА ОТЛАДКИ ПРОГРАММ	315
16.	ЭЛЕКТРИЧЕСКИЕ И ВРЕМЕННЫЕ ПАРАМЕТРЫ	316
16	.1 Электропитание	316
16	.2 Электрические параметры	317
	.3 Динамическая потребляемая мощность	
17.	ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ	
18.	ИСТОРИЯ ИЗМЕНЕНИЙ	330
12	.1 25 мая 2017 г	
	.2 20 ноября 2017 г	
	.3 29 января 2018 г	
	.4 15 марта 2018 г	
10	. 1 10 Mapta 2010 1	551



18.5 18 декабря 2018 г	3.	3
18.6 16 сентября 2019 г	3.	3



1. ВВЕДЕНИЕ

1.1 Назначение

Микросхема интегральная контроллера сетевого твердотельного накопителя информации 1892ВК016 (МСТ-04) спроектирована как однокристальная «система на кристалле» на базе IP-ядерной (IP-intellectual property) платформы «МУЛЬТИКОР», разработанной в АО НПЦ «ЭЛВИС».

Микросхема 1892ВК016 предназначена для применения в бортовых твердотельных накопителях информации объемом до 1 терабайта.

1.2 Функциональные параметры и возможности

Микросхема 1892ВК016 имеет следующие функциональные параметры и возможности:

- два центральных процессора (CPU0. CPU1):
 - архитектура MIPS32;
 - 32-х битные шины передачи адреса и данных;
 - кэш команд объемом 32 Кбайт;
 - кэш данных объемом 32 Кбайт;
 - архитектура привилегированных ресурсов в стиле ядра R4000:
 - о регистры Count/Compare для прерываний реального времени;
 - о отдельный вектор обработки исключений по прерываниям;
 - программируемое устройство управления памятью:
 - два режима работы с TLB (Translation Lookaside Buffer) и FM (Fixed Mapped);
 - о 16 строк в режиме TLB;
 - устройство умножения и деления;
 - JTAG IEEE 1149.1, встроенные средства отладки программ;
 - оперативная память центрального процессора (CRAM) объемом 128 Кбайт;
 - 1 внешний запрос прерывания;
- порт внешней памяти типа DDR (DDR PORT):
 - шина данных 32 разряда;
 - пиковая пропускная способность 1600 Мбайт/с;
 - программное конфигурирование типа блоков памяти и их объема;
 - перевод DDR SDRAM в режим энергосбережения;
- периферийные устройства:
 - два дуплексных порта по стандарту SpaceWire (ECSS-E-50-12C) с пропускной способностью от 2 до 300 Мбод каждый (SWIC0, SWIC1).
 Порты подключены к коммутатору АХІ через 8-канальный контроллер DMA;



- 4-портовый мультипротокольный коммутатор SpaceFibre/GigaSpaceWire (SpaceWire-RUS). Пропускная способность каждого порта от 5 Мбод до 1,25 Гбод. Поддерживает протокол RMAP. Коммутатор подключен к коммутатору АХІ через два многоканальных контроллера DMA;
- 8 портов NFC со встроенным каналом DMA для подключения ONFI 2 NAND Flash. Каждый порт обеспечивает подключение до 16 8-разрядных микросхем NAND Flash;
- многофункциональный буферизированный последовательный порт (MFBSP) работающий в режимах контроллера шины SPI и порта ввода-вывода общего назначения GPIO[2:0];
- 8-канальный контроллер прямого доступа (DMA) типа память-память;
- контроллер прерываний;
- универсальный асинхронный порт (UART) типа 16550;
- два универсальных 32-разрядных таймера (IRT0, IRT1), интервальные/реального времени с двумя источниками входной частоты: CLK, XTI;
- дополнительные возможности и особенности:
 - умножители/делители входной частоты на основе узлов фазовой автоподстройки частоты (PLL);
 - коррекция ошибок внутренней и внешней памяти: исправление однократных ошибок и обнаружение двукратных ошибок по коду Хэмминга;
 - встроенные средства отладки программ (OnCD) с портом JTAG в соответствии со стандартом IEEE 1149.1;
 - режимы энергосбережения;
 - поддержка операционной системы Linux;
 - керамический корпус типа CPGA-720.



1.3 Структурная схема

Структурная схема микросхемы 1892ВК016 приведена на рисунке 1.1.

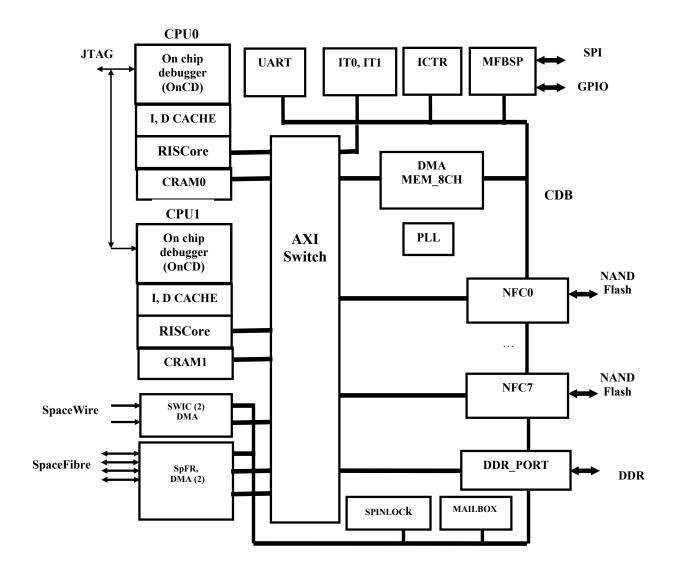


Рисунок 1.1. Структурная схема микропроцессора 1892ВК016

В состав микросхемы 1892ВК016 входят следующие основные узлы:

- CPU0, CPU1 центральные процессоры 0, 1 на основе RISC-ядер;
- I, D CACHE кэш команд и кэш данных CPU;
- CRAM0, CRAM1 оперативная память процессоров;
- CDB шина данных CPU;
- DMA MEM CH 8-канальный контроллер прямого доступа типа память-память;
- DDR PORT порт внешней памяти типа DDR;
- NFC0:NFC7 контроллер внешней памяти типа NAND Flash (NAND FLASH CONTROLLER);



- DMA контроллер прямого доступа в память;
- SPINLOCK модуль низкоуровневого взаимоисключающего примитива синхронизации;
- MAILBOX модуль обмена сообщениями;
- OnCD встроенные средства отладки программ;
- AXI Switch коммутатор AXI;
- PLL умножители частоты на основе PLL;
- SWIC0, SWIC1 контроллеры интерфейса SpaceWire;
- SpFR (SpaceFibre Router) 4-портовый мультипротокольный коммутатор SpaceFibre/GigaSpaceWire (SpaceWire-RUS). Поддерживает протокол RMAP. Подключен к коммутатору АХІ через два многоканальных контроллера DMA;
- MFBSP многофункциональный буферизированный последовательный порт работающий в режимах контроллера шины SPI и GPIO[2:0];
- ICTR контроллер прерываний;
- UART универсальный асинхронный порт;
- IRT0, IRT1 универсальные таймеры, интервальные/реального времени;
- JTAG отладочный порт.

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

Маster выполняется параллельно и без конфликтов.

Исполнительными устройствами являются блоки внутренней памяти CRAM или внешняя память, доступная через DDR_PORT. Задатчиками могут быть CPU, каналы DMA SWIC, SpFR, NFC, каналы DMA типа память-память.

1.4 Инструментальное программное обеспечение

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

Интегрированная среда проектирования включает:

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



1.5 Операционная система для микросхемы 1892ВК016

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

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

.



2. СИСТЕМНАЯ ОРГАНИЗАЦИЯ МИКРОСХЕМЫ

2.1 Карта памяти

Карта физической памяти микросхемы 1892ВК016 приведена в Таблица 2.1. Здесь и далее, если это не оговорено специально, коды адреса и данных указаны в шестнадцатеричной системе счисления. Объемы областей памяти указаны с учетом ее дальнейшего расширения.

Таблица 2.1. Карта физической памяти

Диапазон адресов	Название области
FFFF FFFF	
_	Не используется
1C00_0000	_
1BFF FFFF	Внутренняя память
_	
1800 0000	
17FF FFFF	
_	Внешняя память (не более 128 Мбайт)
0000_0000	, ,

Внешняя память доступна через порт внешней памяти типа DDR.

Следует отметить, что виртуальный базовый адрес исключений $0x8000_0000$ будет размещен во внешней памяти по физическому адресу $0x0000_0000$. Виртуальный базовый адрес исключений $0xBFC0_0000$ будет размещен во внешней памяти по физическому адресу, который зависит от объема внешней памяти следующим образом: 32 Мбайт $-0x01C0_0000$, 64 Мбайт $-0x03C0_0000$, 128 Мбайт $-0x07C0_0000$.

Для CPU0, CPU1 все адресное пространство памяти микросхемы является 32-разрядным. Память CRAM0, CRAM1, а также внешняя память, могут адресоваться с точностью до байта.

Карта внутренней памяти микросхемы 1892ВК016 приведена в Таблица 2.2.

Таблица 2.2. Карта внутренней памяти

Диапазон адресов	Название области
1BFF_FFFF	
	Резерв
1830_0200	
1830_00FF	
	ПЗУ начального загрузчика
1830_0000	
182F_FFFF	
	Регистры устройств микросхемы
182F_0000	



Диапазон адресов	Название области
182E FFFF	
_	Резерв
1842_0000	
1841 FFFF	
_	Память CRAM1
1840_0000	
183F FFFF	
_	Резерв
1802_0000	
1801_FFFF	
_	Память CRAM0
1800_0000	

Доступ к памяти CRAM0, CRAM1 выполняется следующим образом:

- CPU0 обращаясь по адресу 0x1840.0000 попадает через коммутатор АХІ в CRAM1;
- CPU0 обращаясь по адресу 0x1800.0000 попадает напрямую в CRAM0;
- СРU1 обращаясь по адресу 0x1840.0000 попадает напрямую в CRAM1;
- CPU1 обращаясь по адресу 0x1800.0000 попадает через коммутатор АХІ в CRAM0;
- остальные мастера, обращаясь по адресу 0x1800.0000 попадают через коммутатор AXI в CRAM0;
- остальные мастера, обращаясь по адресу 0x1840.0000 попадают через коммутатор AXI в CRAM1.

Перечень программно доступных регистров приведен в Таблица 2.3.



Таблица 2.3

Условное		Адрес
обозначение	Название регистра	регистра
регистра	ATM CIT	
Регистры DMA M	·	
CSR_MEM_CH	Регистр управления и состояния (по чтению сброс	182F_0000
0	битов «END» и «DONE»)	
CP_MEM_CH0	Регистр указателя цепочки	182F_0004
IR0_MEM_CH0	Регистр индекса 0	182F_0008
IR1 MEM CH0	Регистр индекса 1	182F 000C
OR MEM CH0	Регистр смещений	182F 0010
Y MEM CH00	Регистр параметров направления У при двухмерной	182F 0014
	адресации	_
RUN MEM CH	На запись: Псевдорегистр управления состоянием	182F 0018
0	бита RUN регистра CSR MEM CH0	1021_0010
	На чтение: Регистр управления и состояния канала	
	МЕМ СН0 без сброса битов «END» и «DONE»	
	WIEWI_CITO OCS COPOCA ONTOB (CITOD) II (DOTVE)	
CCD MEM CH	Dorwood vanopayyya v 20072 (-2 5	192E 0090
CSR_MEM_CH	Регистр управления и состояния (по чтению сброс	182F_0080
l CD MEM CHI	битов «END» и «DONE»)	100E 0004
CP_MEM_CH1	Регистр указателя цепочки	182F_0084
IR0_MEM_CH1	Регистр индекса 0	182F_0088
IR1_MEM_CH1	Регистр индекса 1	182F_008C
OR_MEM_CH1	Регистр смещений	182F_0090
Y_MEM_CH1	Регистр параметров направления У при двухмерной	182F_0094
	адресации канала	
RUN MEM CH	На запись: Псевдорегистр управления состоянием	182F 0098
1	бита RUN регистра CSR MEM CH1	_
	На чтение: Регистр управления и состояния канала	
	MEM CH1 без сброса битов «END» и «DONE»	
CSR MEM CH	Регистр управления и состояния (по чтению сброс	182F 0100
2	битов «END» и «DONE»)	
CP MEM CH2	Регистр указателя цепочки	182F 0104
IRO MEM CH2	Регистр индекса 0	182F 0108
IR1 MEM CH2	Регистр индекса 1	182F 010C
	<u> </u>	
OR_MEM_CH2	Регистр смещений канала МЕМ_СН2	182F_0110
Y_MEM_CH2	Регистр параметров направления У при двухмерной	182F_0114
DIDI MEM CH	адресации канала МЕМ_СН2	100E 0110
RUN_MEM_CH	На запись: Псевдорегистр управления состоянием	182F_0118
2	бита RUN регистра CSR_MEM_CH2	
	На чтение: Регистр управления и состояния канала	
	MEM_CH2 без сброса битов «END» и «DONE»	
CSR_MEM_CH	Регистр управления и состояния	182F_0180
3	(по чтению сброс битов «END» и «DONE»)	
CP_MEM_CH3	Регистр указателя цепочки	182F_0184
IR0_MEM_CH3	Регистр индекса 0	182F_0188
IR1_MEM_CH3	Регистр индекса 1	182F_018C
OR MEM CH3	Регистр смещений	182F 0190
Y MEM CH3	Регистр параметров направления У при двухмерной	182F 0194
	адресации	
	· u - · · ·	



Условное обозначение регистра	Название регистра	Адрес регистра				
RUN_MEM_CH	На запись: Псевдорегистр управления состоянием	182F_0198				
3	бита RUN регистра CSR_MEM_CH3 На чтение: Регистр управления и состояния канала					
Darwarni DMA M	MEM_CH3 без сброса битов «END» и «DONE» Регистры DMA MEM CH					
CSR MEM CH	Регистр управления и состояния (по чтению сброс	182F 0200				
4	битов «END» и «DONE»)	1821_0200				
CP MEM CH4	Регистр указателя цепочки	182F 0204				
IR0 MEM CH4	Регистр индекса 0	182F 0208				
IR1 MEM CH4	Регистр индекса 1	182F 020C				
OR MEM CH4	Регистр смещений	182F 0210				
Y_MEM_CH4	Регистр параметров направления Y при двухмерной адресации	182F_0214				
RUN_MEM_CH 4	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH4 На чтение: Регистр управления и состояния канала MEM_CH0 без сброса битов «END» и «DONE»	182F_0218				
CSR_MEM_CH 5	Регистр управления и состояния (по чтению сброс битов «END» и «DONE»)	182F_0280				
CP MEM CH5	Регистр указателя цепочки	182F 0284				
IR0 MEM CH5	Регистр индекса 0	182F 0288				
IR1_MEM_CH5	Регистр индекса 1	182F_028C				
OR_MEM_CH5	Регистр смещений	182F_0290				
Y_MEM_CH5	MEM_CH5 Регистр параметров направления Y при двухмерной адресации канала					
RUN_MEM_CH 5	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH5 На чтение: Регистр управления и состояния канала MEM_CH1 без сброса битов «END» и «DONE»	182F_0298				
CSR_MEM_CH	Регистр управления и состояния (по чтению сброс битов «END» и «DONE»)	182F_0300				
CP MEM CH6	Регистр указателя цепочки	182F 0304				
IR0_MEM_CH6	Регистр индекса 0	182F_0308				
IR1_MEM_CH6	Регистр индекса 1	182F_030C				
OR_MEM_CH6	Регистр смещений канала МЕМ_СН6	182F_0310				
Y_MEM_CH6	Регистр параметров направления Y при двухмерной адресации канала MEM CH2	182F_0314				
RUN_MEM_CH	На запись: Псевдорегистр управления состоянием	182F_0318				
6	бита RUN регистра CSR_MEM_CH02					
	На чтение: Регистр управления и состояния канала MEM_CH2 без сброса битов «END» и «DONE»					
CSR MEM CH	Регистр управления и состояния	182F 0380				
7	(по чтению сброс битов «END» и «DONE»)	1021_0300				
CP MEM CH7	Регистр указателя цепочки	182F 0384				
IR0 MEM CH7	Регистр индекса 0	182F 0388				
IR1 MEM CH7						
OR_MEM_CH7 Регистр смещений		182F_038C 182F_0390				



Условное обозначение регистра	Название регистра	Адрес регистра		
Y_MEM_CH7	Регистр параметров направления Y при двухмерной адресации	182F_0394		
RUN_MEM_C7	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR_MEM_CH7	182F_0398		
	На чтение: Регистр управления и состояния канала МЕМ_СНЗ без сброса битов «END» и «DONE»			
Регистры DMA SV				
	мять дескрипторов принимаемых пакетов			
CSR_SWIC_RX_E ES0	Регистр управления и состояния (по чтению сброс битов «END» и «DONE»)	182F_B000		
CP_ SWIC RX DES0	Регистр указателя цепочки	182F_B004		
IR_ SWIC RX DES0	Регистр индекса	182F_B008		
RUN_ SWIC RX DES0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR	182F_B00C		
SWIE_IGI_BESS	На чтение: Регистр управления и состояния без сброса битов «END» и «DONE»			
Канал записи в па	мять принимаемых слов данных			
CSR	Регистр управления и состояния	182F B040		
SWIC_RX_DAT0	(по чтению сброс битов «END» и «DONE»)	_		
CP_ SWIC RX DAT0	Регистр указателя цепочки	182F_B044		
IR_ SWIC RX DAT0	Регистр индекса	182F_B048		
RUN_ SWIC_RX_DAT0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR	182F_B04C		
	На чтение: Регистр управления и состояния без сброса битов «END» и «DONE»			
	амяти дескрипторов передаваемых пакетов			
CSR_ SWIC_TX_DES0	Регистр управления и состояния (по чтению сброс битов «END» и «DONE»)	182F_B080		
CP_ SWIC_TX_DES0	Регистр указателя цепочки	182F_B084		
IR_ SWIC TX DES0	Регистр индекса	182F_B088		
RUN_ SWIC_TX_DES0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR	182F_B08C		
	На чтение: Регистр управления и состояния без сброса битов «END» и «DONE»			
Канал чтения из памяти передаваемых слов данных				
CSR_	Регистр управления и состояния	182F_B0C0		
SWIC_TX_DAT0	(по чтению сброс битов «END» и «DONE»)	100F D0C4		
CP_ SWIC_TX_DAT0	Регистр указателя цепочки	182F_B0C4		
IR_ SWIC_TX_DAT0	Регистр индекса	182F_B0C8		
RUN_ SWIC_TX_DAT0	На запись: Псевдорегистр управления состоянием бита RUN регистра CSR На чтение: Регистр управления и состояния без	182F_B0CC		
	сброса битов «END» и «DONE»			



Условное обозначение регистра	Название регистра	Адрес регистра		
Регистры DMA SW	IC1			
Канал записи в пам	ять дескрипторов принимаемых пакетов			
CSR SWIC RX D	Регистр управления и состояния	182F B100		
ES1	(по чтению сброс битов «END» и «DONE»)	_		
СР	Регистр указателя цепочки	182F_B104		
SWIC RX DES1		_		
IR	Регистр индекса	182F B108		
SWIC RX DES1				
RUN	На запись: Псевдорегистр управления состоянием	182F B10C		
SWIC RX DES1	бита RUN регистра CSR	1021_2100		
SWIE_IGI_BEST	На чтение: Регистр управления и состояния без			
	сброса битов «END» и «DONE»			
Канал записи в пам	ять принимаемых слов данных			
CSR	Регистр управления и состояния	182F B140		
SWIC RX DAT1	(по чтению сброс битов «END» и «DONE»)	1021_D170		
CP CP	Регистр указателя цепочки	182F B144		
SWIC RX DAT1	тегистр указателя цепочки	1021 _D144		
	Darwarm www.maraa	102E D140		
IR_	Регистр индекса	182F_B148		
SWIC_RX_DAT1	П	102F D14C		
RUN_	На запись: Псевдорегистр управления состоянием	182F_B14C		
SWIC_RX_DAT1	бита RUN регистра CSR_			
	На чтение: Регистр управления и состояния без			
TC	сброса битов «END» и «DONE»			
	ияти дескрипторов передаваемых пакетов	1000 0100		
CSR_	Регистр управления и состояния	182F_B180		
SWIC_TX_DES1	(по чтению сброс битов «END» и «DONE»)			
CP_	Регистр указателя цепочки	182F_B184		
SWIC_TX_DES1				
IR_	Регистр индекса	182F_B188		
SWIC_TX_DES1				
RUN_	На запись: Псевдорегистр управления состоянием	182F_B18C		
SWIC_TX_DES1	бита RUN регистра CSR_			
	На чтение: Регистр управления и состояния без			
	сброса битов «END» и «DONE»			
Канал чтения из пам	ияти передаваемых слов данных			
CSR_	Регистр управления и состояния	182F_B1C0		
SWIC_TX_DAT1	(по чтению сброс битов «END» и «DONE»)			
CP_	Регистр указателя цепочки	182F_B1C4		
SWIC_TX_DAT1		_		
IR	Регистр индекса	182F B1C8		
SWIC TX DAT1	•	_		
RUN	На запись: Псевдорегистр управления состоянием	182F B1CC		
SWIC TX DAT1	бита RUN регистра CSR	_		
	На чтение: Регистр управления и состояния без			
	сброса битов «END» и «DONE»			
	•			
Регистры контроллера SWIC0				
HW VER0	Регистр аппаратной версии контроллера	182F A000		
STATUS0	Регистр состояния	182F A004		
RX CODE0	Регистр принятого управляющего символа	182F A008		
KA_CODE0	т стистр припятого управляющего символа	1021 _AUU0		



Условное обозначение регистра	Название регистра	Адрес регистра
MODE_CR0	Регистр управления режимом работы	182F_A00C
TX_SPEED0	Регистр управления скоростью передачи	182F_A010
TX_CODE0	Регистр передаваемого управляющего символа	182F_A014
RX_SPEED0	Регистр измерителя скорости приема	182F_A018
CNT_RX0_PACK	Регистр счетчика принятых пакетов нулевой	182F_A01C
0	длины	1000 1000
CNT_RX_PACK0	Регистр счетчика принятых пакетов ненулевой длины	182F_A020
ISR_L0	Регистр кодов распределенных прерываний (младшая часть)	182F_A024
ISR_H0	Регистр кодов распределенных прерываний (старшая часть)	182F_A028
TRUE TIME0	Регистр достоверного маркера времени	182F A02C
TOUT_CODE0	Регистр размера таймаутов	182F_A030
ISR_tout_L0	Младшие разряды регистра флагов таймаутов ISR	182F_A034
ISR tout H0	Старшие разряды регистра флагов таймаутов ISR	182F A038
LOG ADDR0	Регистр логического адреса	182F A03C
		_
Регистры контролл	epa SWIC1	
HW_VER1	Регистр аппаратной версии контроллера	182F_A800
STATUS1	Регистр состояния	182F_A804
RX_CODE1	Регистр принятого управляющего символа	182F_ A808
MODE_CR1	Регистр управления режимом работы	182F_A80C
TX_SPEED1	Регистр управления скоростью передачи	182F_A810
TX_CODE1	Регистр передаваемого управляющего символа	182F_A814
RX_SPEED1	Регистр измерителя скорости приема	182F_A818
CNT_RX0_PACK	Регистр счетчика принятых пакетов нулевой длины	182F_A81C
CNT_RX_PACK1	Регистр счетчика принятых пакетов ненулевой длины	182F_A820
ISR_L1	Регистр кодов распределенных прерываний (младшая часть)	182F_A824
ISR_H1	Регистр кодов распределенных прерываний (старшая часть)	182F_A828
TRUE TIME1	Регистр достоверного маркера времени	182F A82C
TOUT CODE1	Регистр размера таймаутов	182F A830
ISR tout L1	Младшие разряды регистра флагов таймаутов ISR	182F A834
ISR tout H1	Старшие разряды регистра флагов таймаутов ISR	182F A838
LOG ADDR1	Регистр логического адреса	182F A83C
Регистры контролл	· · · · · · · · · · · · · · · · · · ·	<u>-</u>
PACKET	Регистр настройки пакетов	182F C000
MEMADDR1	Регистр адреса 1	182F C004
MEMADDR2	Регистр адреса 2	182F_C008
COMMAND	Регистр команды	182F_C00C
PROGRAM	Регистр программы	182F_C010
INTERRUPT_STA TUS EN	Регистр разрешения статуса прерываний	182F_C014
INTERRUPT_SIG NAL_EN	Регистр разрешения сигнала прерываний	182F_C018



Условное обозначение регистра	Название регистра	Адрес регистра			
INTERRUPT_STA TUS	Регистр статуса прерываний	182F_C01C			
ID1	Регистр ID1	182F C020			
ID2	Регистр ID2	182F C024			
FLASH STATUS	Регистр статуса флэш-памяти	182F C028			
TIMING	Регистр временных параметров	182F C02C			
BUFFER DATA	Регистр данных буфера	182F C030			
ECC ECC	Регистр ЕСС	182F C034			
ECC ERROR CN	Счётчик ошибок ЕСС	182F_C034			
T		_			
ECC_SPARE_CM D	Регистр команд ЕСС	182F_C03C			
ERROR_CNT_1BI T	Счётчик 1-битных ошибок	182F_C040			
ERROR_CNT_2BI T	Счётчик 2-битных ошибок	182F_C044			
ERROR_CNT_3BI T	Счётчик 3-битных ошибок	182F_C048			
ERROR_CNT_4BI T	Счётчик 4-битных ошибок	182F_C04C			
DMA ADDR	Регистр адреса DMA	182F C050			
DMA_BUFFER_B OUNDARY	Регистр границы буфера DMA	182F_C054			
RESERVED	Зарезервировано	182F C058			
ERROR_COUNT_ 5BIT	Счётчик 5-битных ошибок	182F_C05C			
ERROR_COUNT_ 6BIT	Счётчик 6-битных ошибок	182F_C060			
ERROR_COUNT_ 7BIT	Счётчик 7-битных ошибок	182F_C064			
ERROR_COUNT_ 8BIT	Счётчик 8-битных ошибок	182F_C068			
WRITE_PROTEC T	Регистр управления защитой памяти	182F_C06C			
Регистры контролле	epa NFC1				
	Базовый адрес	182F_C800			
Регистры контролле	epa NFC2	<u></u>			
	Базовый адрес	182F_D000			
Регистры контролле	•	-			
•	Базовый адрес	182F D800			
Регистры контролле	*	_			
•	Базовый адрес	182F E000			
Регистры контролле	•	-			
•	Базовый адрес	182F E800			
Регистры контролле	Регистры контроллера NFC5				
•	Базовый адрес	182F F000			
Регистры контролле	•	<u> </u>			
į į į	Базовый адрес	182F F800			
Регистры MFBSP	· 2	_			
TX_MFBSP	Буфер передачи данных	182F_3800			



Условное обозначение регистра	Название регистра	Адрес регистра
RX_MFBSP	Буфер приёма данных	182F_3800
CSR_MFBSP	Регистр управления и состояния	182F_3804
DIR_MFBSP	Регистр управления направлением выводов порта ввода-вывода	182F_3808
GPIO_DR	Регистр данных порта ввода-вывода	182F_380C
TCTR	Регистр управления передатчиком	182F_3810
RCTR	Регистр управления приёмником	182F_3814
TSR	Регистр состояния передатчика	182F_3818
RSR	Регистр состояния приёмника	182F_381C
Регистры UART		
RBR	Приемный буферный регистр	182F 3000
THR	Передающий буферный регистр	182F 3000
IER	Регистр разрешения прерываний	182F 3004
IIR	Регистр идентификации прерывания	182F 3008
FCR	Регистр управления FIFO	182F 3008
LCR	Регистр управления линией	182F 300C
MCR	Регистр управления модемом	182F 3010
LSR	Регистр состояния линии	182F 3014
SPR	Регистр Scratch Pad	182F 301C
DLL	Регистр делителя младший	182F 3000
DLM	Регистр делителя старший	182F 3004
SCLR	Регистр предделителя (scaler)	182F_3014
Регистры ITR1		
ITCSR	Регистр управления	182F_2000
ITPERIOD	Регистр периода работы таймера	182F_2004
ITCOUNT	Регистр счетчика	182F_2008
ITSCALE	Регистр предделителя	182F_200C
Регистры IRT0		
ITCSR	Регистр управления	182F_2020
ITPERIOD	Регистр периода работы таймера	182F_2024
ITCOUNT	Регистр счетчика	182F_2028
ITSCALE	Регистр предделителя	182F_202C
Регистры DDR PC	DRT	
DDR_CON	Регистр конфигурации	182F_1014
DDR_TMR	Регистр параметров	182F_1018
DDR_CSR	Регистр управления и состояния	182F_101C
DDR_MOD	Регистр режимов	182F_1020
DDR_EXT	Регистр управления режимами контроля памяти 18	
DDR_ERR	Регистр ошибок памяти	182F_1028
Системные регист	ОЫ	
CR PLL Регистр управления PLL 182F 4000		182F 4000
CLK_EN	Регистр управления отключением частоты от устройств	182F_4004
CSR0	Регистр управления и состояния 0	182F_4008



Условное обозначение регистра	Название регистра	Адрес регистра
CSR1	Регистр управления и состояния 1	182F_4048
BOOT	Регистр управления начальной загрузкой	182F_4034
MASKR00	Регистр маски прерываний из регистра QSTR0 для CPU0	182F-4010
MASKR10	Регистр маски прерываний из регистра QSTR0 для CPU1	182F-4050
QSTR0	Регистр запросов прерываний QSTR0	182F-4014 182F-4054
MASKR01	Регистр маски прерываний из регистра QSTR1 для CPU0	182F-4018
MASKR11	Регистр маски прерываний из регистра QSTR1 для CPU1	182F-4058
QSTR1	Регистр запросов прерываний QSTR1	182F-401C
		182F-405C
MASKR02	Регистр маски прерываний из регистра QSTR2 для CPU0	182F-4020
MASKR12	Регистр маски прерываний из регистра QSTR2 для CPU1	182F-4060
QSTR2	Регистр запросов прерываний QSTR2	182F-4024
		182F-4064
MASKR03	Регистр маски прерываний из регистра QSTR3 для CPU0	182F-4028
MASKR13	Регистр маски прерываний из регистра QSTR3 для CPU1	182F-4068
QSTR3	Регистр запросов прерываний QSTR3	182F-402C 182F-406C
MASKR04	Регистр маски прерываний из регистра QSTR4 для CPU0	182F-4038
MASKR14	Регистр маски прерываний из регистра QSTR4 для CPU1	182F-4078
QSTR4	Регистр запросов прерываний QSTR4	182F-403C 182F-407C
IRQM	Регистр управления режимом приема внешних прерываний nIRQ[3:0]	182F-4030
Регистры модуля SI	PINI OCK	
тегистры модули от	Базовый адрес	182F_6000
Регистры модуля М	AUBOX	
гегистры модуля ічі	Базовый адрес	182F_7000
Регистры коммутат		
	Базовый адрес	182F_8000
Регистры контроля	Xэммингом памяти CRAM0, ICACHE0, DCACHE0	CPU0
CSR CRAM00A Регистр управления и состояния CRAM0A 182F 4400		
AERROR_CRAM0 0A	FIFO ошибочных адресов CRAM0A	182F_4404
CSR_CRAM00B	Регистр управления и состояния CRAM0B	182F_4408



Условное обозначение регистра	Название регистра	Адрес регистра
AERROR_CRAM0 0B	FIFO ошибочных адресов CRAM0B	182F_440C
CSR_CRAM01A	Регистр управления и состояния CRAM1A	182F_4410
AERROR_CRAM0 1A	FIFO ошибочных адресов CRAM1A	182F_4414
CSR_CRAM01B	Регистр управления и состояния CRAM1B	182F_4418
AERROR_CRAM0 1B	FIFO ошибочных адресов CRAM1B	182F_441C
CSR_CRAM02A	Регистр управления и состояния CRAM2A	182F_4420
AERROR_CRAM0 2A	FIFO ошибочных адресов CRAM2A	182F_4424
CSR_CRAM02B	Регистр управления и состояния CRAM2B	182F_4428
AERROR_CRAM0 2B	FIFO ошибочных адресов CRAM2B	182F_442C
CSR_CRAM03A	Регистр управления и состояния CRAM3A	182F_4430
AERROR_CRAM0 3A	FIFO ошибочных адресов CRAM3A	182F_4434
CSR_CRAM03B	Регистр управления и состояния CRAM3B	182F_4438
AERROR_CRAM0 3B	FIFO ошибочных адресов CRAM3B	182F_443C
CSR_ICACHE0	Регистр управления и состояния ІСАСНЕ	182F_4800
AERROR_ICACH E0	FIFO ошибочных адресов ICACHE	182F_4804
CSR_DCACHE0	Регистр управления и состояния DCACHE	182F_4C00
AERROR_DCACH E0	FIFO ошибочных адресов DCACHE	182F_4C04
Регистры контроля	Хэммингом памяти CRAM1, ICACHE1, DCACHE	1 CPU1
CSR CRAM10A	Регистр управления и состояния CRAM0A	182F 5400
AERROR_CRAM1 0A	FIFO ошибочных адресов CRAM0A	182F_5404
CSR CRAM10B	Регистр управления и состояния CRAM0B	182F 5408
AERROR_CRAM1 0B	FIFO ошибочных адресов CRAM0B	182F_540C
CSR_CRAM11A	Регистр управления и состояния CRAM1A	182F_5410
AERROR_CRAM1 1A	FIFO ошибочных адресов CRAM1A	182F_5414
CSR_CRAM11B	Регистр управления и состояния CRAM1B	182F_5418
AERROR_CRAM1 1B	FIFO ошибочных адресов CRAM1B	182F_541C
CSR_CRAM12A	Регистр управления и состояния CRAM2A	182F_5420
AERROR_CRAM1 2A	FIFO ошибочных адресов CRAM2A	182F_5424
CSR_CRAM12B	Регистр управления и состояния CRAM2B	182F_5428
AERROR_CRAM1 2B	FIFO ошибочных адресов CRAM2B	182F_542C
CSR_CRAM13A	Регистр управления и состояния CRAM3A	182F_5430
AERROR_CRAM1 3A	FIFO ошибочных адресов CRAM3A	182F_5434
CSR_CRAM13B	Регистр управления и состояния CRAM3B	182F_5438



Условное обозначение регистра	Название регистра	Адрес регистра
AERROR_CRAM1 3B	FIFO ошибочных адресов CRAM3B	182F_543C
CSR_ICACHE1	Регистр управления и состояния ІСАСНЕ	182F_5800
AERROR_ICACH E1	FIFO ошибочных адресов ICACHE	182F_5804
CSR_DCACHE1	Регистр управления и состояния DCACHE	182F_5C00
AERROR_DCACH E1	FIFO ошибочных адресов DCACHE	182F_5C04

2.2 Система синхронизации

2.2.1 Входы синхронизации и умножители частоты

Микросхема 1892ВК016 имеет следующие входы синхронизации:

- XTI частота 10 МГц для синхронизации всех умножителей частоты микросхемы;
- XTI125 частоты 125 МГц для работы портов SpaceFibre/GigaSpaceWire.

Для синхронизации работы узлов микросхемы 1892ВК016 используются умножители частоты на основе схемы фазовой автоподстройки частоты (PLL). Имеется следующие умножители частоты:

- PLL_CORE формирует тактовую частоту работы CPU0, CPU1, UART, IT, коммутатора AXI, системной части всех устройств микросхемы;
- PLL DDR формирует тактовую частоту для работы внешней памяти типа DDR;
- PLL_NAND формирует тактовую частоту для работы памяти типа NAND Flash;
- PLL_TX_SW0, PLL_TX_SW1 формирует тактовую частоту для передачи последовательного кода из портов SpaceWire в одноименную сеть.

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

2.2.2 Управление работой PLL

Управление работой PLL осуществляется при помощи регистра CR_PLL, формат которого приведен в Таблица 2.4.



Таблица 2.4. Формат регистра CR_PLL

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	-	Не используется	R	0
30:24	CLK_SEL_DDR[6:0]	Коэффициент умножения/деления входной частоты PLL_DDR (частота XTI, деленная на 2): 00 – 1/16; 01 – 1 02 – 2; 03 – 3;	R/W	0
		1E – 30 (максимально допустимый коэффициент умножения). Частота PLL_DDR в 2 раза больше чем тактовая частота СК, подаваемая на внешнюю память типа DDR		
23	PLL_NAND_EN	Выбор источника тактовой частоты для работы NAND Flash: 1 – PLL_NAND; 0 – вход XTI	R/W	0
22:16	CLK_SEL_NAND[6:0]	Коэффициент умножения/деления входной частоты PLL_NAND (частота XTI, деленная на 2): 00 – 1/16; 01 – 1 02 – 2; 03 – 3; 7F – 127	R/W	0
15:8	-	Не используется	R	0
7	PLL_CORE_EN	Выбор источника тактовой частоты для работы CPU, UART, IT, WDT, MFBSP, коммутатора AXI, системной части всех устройств микросхемы: 1 – PLL_CORE; 0 – вход XTI	R/W	0
6:0	CLK_SEL_CORE[6:0]	Коэффициент умножения/деления входной частоты PLL_CORE (частота XTI, деленная на 2): 00 – 1/16; 01 – 1 02 – 2; 03 – 3; 7F – 127	R/W	1



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

2.2.3 Отключение и включение тактовой частоты

В данной микросхеме имеется два режима энергосбережения:

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

Уменьшение тактовой частоты устройств выполняется при записи необходимого кода в поле CLK_SEL регистра CR_PLL. При этом значение тактовой частоты изменится через время не более чем 2 мс.

Отключение тактовой частоты от устройств выполняется при помощи регистра CLK_EN, формат которого приведен в Таблица 2.5.

Таблица 2.5. Формат регистра CLK_EN

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:29	-	Не используется	-	0
28	CLKEN_SpFR	Управление включением тактовых частот коммутатора SpFR, поступающих от PLL_CORE и входа	R/W	0
		микросхемы XTI125: 1 – частота включена; 0 – частота выключена		
27:26	-	Не используется	-	0
25:24	CLKEN_SWIC[1:0]	Управление включением тактовой частоты SWIC[1:0], поступающей от PLL_CORE: 1 — частота включена; 0 — частота выключена	R/W	0
23:16	CLKEN_NFC[7:0]	Управление включением тактовых частот NFC[7:0] поступающих от PLL_CORE и PLL_NAND: 1 — частота включена; 0 — частота выключена.	R/W	0
15:13	-	Не используется	-	0
12	CLKEN_DMA	Управление включением тактовой частоты DMA MEM_CH поступающей от PLL_CORE: 1 — частота включена; 0 — частота выключена.	R/W	0
11:7	-	Не используется	-	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
6	CLKEN_DDR	Управление включением тактовой частоты DDR_PORT поступающей от PLL_CORE: 1 — частота включена; 0 — частота выключена. В этом случае программный доступ к регистрам DDR PORT запрещен	R/W	0
5:2	-	Не используется	-	0
1	CLKEN_CPU1	Управление включением тактовой частоты для CPU1, поступающей от PLL_CORE: 1 — частота включена; 0 — частота выключена.	R/W	0
0	CLKEN_CORE	Признак наличия тактовой частоты для CPU0, UART, IT, WDT, MFBSP, коммутатора AXI, поступающей от PLL_CORE, а также признак включения PLL_DDR. Обязательное требование: при записи данных в регистр CLK_EN, в этом разряде должна быть всегда «1»	R	1

При CLKEN CORE = 1:

- частота от PLL_CORE всегда поступает на CPU0, UART, IRT0, IRT1, WDT, MFBSP, коммутатор AXI Switch;
- частота от PLL_CORE поступающая на DMA, SWIC, SpFR, NFC, DDR_PORT может быть отключена, при помощи соответствующего разряда регистра CLK EN.

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

2.3 Контроллер прерываний

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

Каждый разряд регистров QSTR содержит запрос прерывания от внутренних узлов микросхемы и от внешнего сигнала прерывания nIRQ вне зависимости от состояния соответствующих разрядов регистров MASKR:

- 0 нет запроса;
- 1 есть запрос.

Сигналы внутренних прерываний формируются в соответствующих устройствах при выполнении определенных условий. В процессе обслуживания прерывания необходимо проанализировать состояние устройства для определения причины его возникновения. Сброс прерывания осуществляется в момент исключения причины возникновения данного прерывания.



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

Исходное состояние регистров QSTR – нули.

Каждое внутреннее прерывание можно замаскировать. Для этого имеются 4 регистра маски MASKR0, MASK1, MASK2, MASK3 и MASK4 форматы которых аналогичны форматам соответствующих регистров QSTR0, QSTR1, QSTR2, QSTR3 и QSTR4. Исходное состояние регистров маски – нули (все прерывания запрещены). Регистры маски доступны по записи и чтению.

Форматы регистров QSTR приведены в Таблица 2.6 – Таблица 2.10.

Таблица 2.6. Формат регистра QSTR0

Номер Разряда	Условное обозначение прерывания	Название прерывания		
31	SW_TX_DAT_CH1	Прерывание от канала DMA SW_TX_DAT_CH1		
30	SW TX DES CH1	Прерывание от канала DMA SW TX DES CH1		
29	SW RX DAT CH1	Прерывание от канала DMA SW RX DAT CH1		
28	SW RX DES CH1	Прерывание от канала DMA SW RX DES CH1		
27	-	Не используется		
26	SW TIME1	Прерывание SWIC1 – получен маркер		
	_	времени/распределенное прерывание		
25	SW_ERR1	Прерывание SWIC1 – ошибка в канале		
24	SW_LINK1	Прерывание SWIC1 – установлено соединение		
23	SW_TX_DAT_CH0	Прерывание от канала DMA SW_TX_DAT_CH0		
22	SW_TX_DES_CH0	Прерывание от канала DMA SW_TX_DES_CH0		
21	SW_RX_DAT_CH0	Прерывание от канала DMA SW_RX_DAT_CH0		
20	SW_RX_DES_CH0	Прерывание от канала DMA SW_RX_DES_CH0		
19	-	Не используется		
18	SW_TIME0	Прерывание SWIC0 – получен маркер		
		времени/распределенное прерывание		
17	SW_ERR0	Прерывание SWIC0 – ошибка в канале		
16	SW_LINK0	Прерывание SWIC0- установлено соединение		
15	MAILBOX_WRITE1	Прерывание MAILBOX WRITE1		
14	MAILBOX_WRITE0	Прерывание MAILBOX WRITE0		
13	MAILBOX_READ1	Прерывание MAILBOX READ1		
12	MAILBOX_READ0	Прерывание MAILBOX READ0		
11	SRQ0	Запрос обслуживания от порта MFBSP0.		
		Формируется, если порт выключен (LEN=0,		
		SPI_I2S_EN=0), а на выводах LACK или LCLK		
		присутствует сигнал высокого уровня		
10	MFBSP_RXBUF	Формируется, если порт MFBSP включен на прием		
		данных (в одном из режимов), а число 64-разрядных слов		
		в буфере приёма больше чем RLEV (RLEV		
		устанавливается в регистре состояния приёмника RSR)		



Номер Разряда	Условное обозначение прерывания	Название прерывания	
9	MFBSP_TXBUF	Формируется, если порт MFBSP включен на передачу данных (в одном из режимов), а число 64-разрядных слов, находящихся в буфере передачи меньше, либо равно TLEV (TLEV устанавливается в регистре состояния передатчика TSR)	
8	UART	Прерывание от UART	
7	-	Не используется	
6	IRT1	Прерывание от таймера IRT1	
5	IRT0	Прерывание от таймера IRT0	
4:1	-	Не используется	
0	IRQ	Внешнее прерывание nIRQ	

Таблица 2.7. Формат регистра QSTR1

Номер Разряда	Условное обозначение прерывания	Название прерывания	
31:30	-	Не используется	
29	INT_HmDCACHE1	Прерывание по контролю кода Хемминга от DCACHE CPU1	
28	INT_HmICACHE1	Прерывание по контролю кода Хемминга от ICACHE CPU1	
27:25	-	Не используется	
24	INT_HmCRAM1	Прерывание по контролю кода Хемминга от CRAM1	
23	INT_HmDDR	Прерывание по контролю кода Хемминга от DDR	
22	-	Не используется	
21	INT_HmDCACHE0	Прерывание по контролю кода Хемминга от DCACHE CPU0	
20	INT_HmICACHE0	Прерывание по контролю кода Хемминга от ICACHE CPU0	
19:17	-	Не используется	
16	INT_HmCRAM0	Прерывание по контролю кода Хемминга от CRAM0	
15:8	-	Не используется	
7	MEM_CH7	Прерывание от канала DMA MEM_CH7	
0	MEM_CH0	Прерывание от канала DMA MEM_CH0	



Таблица 2.8. Формат регистра QSTR2

	Tuomingu 2:00 + Opinut pernetpu Qo 11c2				
Номер Разряда	Условное обозначение	Название прерывания			
31:24	прерывания NAND PORT[7:0]	Прерывания от портов NAND PORT			
	г коммутатора SpFR				
23:20	-	Не используется			
19	IRQ_REC_RMAP_4	Прерывание по приему пакета, являющегося RMAP-			
10	IDO DEG DIGIDA	командой чтения или записи по каналу DMA4			
18	IRQ_REC_RMAP_3	Прерывание по приему пакета, являющегося RMAP- командой чтения или записи по каналу DMA3			
17	IRQ_REC_RMAP_2	Прерывание по приему пакета, являющегося RMAP-командой чтения или записи по каналу DMA2			
16	IRQ_REC_RMAP_1	Прерывание по приему пакета, являющегося RMAP- командой чтения или записи по каналу DMA1			
15	IRQ RMAP REJ	Прерывание по факту получения RMAP-команды,			
		которую пришлось отбросить из-за превышения допустимого количества одновременно обрабатываемых команд			
14	IRQ REG4 FINISH	Прерывание по завершению области памяти DMA4			
13	IRQ REG3 FINISH	Прерывание по завершению области памяти DMA3			
12	IRQ REG2 FINISH	Прерывание по завершению области памяти DMA2			
11	IRQ REG1 FINISH	Прерывание по завершению области памяти DMA1			
10	IRQ FIFO ERR	Прерывание по факту ошибки при записи дескриптора в			
		FIFO ответов на команды чтения или записи –			
		идентификатор команды не соответствует			
		идентификатору ни одной из отложенных команд,			
		дублетный идентификатор команды (запись в			
		заполненное FIFO)			
9	IRQ_REC_PACKET	Прерывание по факту приема пакета, не являющегося RMAP-командой.			
8	IRQ_REC_TOUT	Прерывание по факту истечения таймаута ожидания			
		приема очередного символа из сети (пакет начал			
		приниматься и почему-то прекратилась передача без			
		разрыва соединения) у одного или нескольких портов			
7	IRQ_SEND_TOUT	Прерывание по факту истечения таймаута отправки			
		очередного символа в сеть (в режиме Giga), по факту			
		истечения счетчика количества неудачных попыток			
		отправки фрейма (в режиме SpFi) у одного или			
	IDO ADD TOUT	нескольких портов			
6	IRQ_ARB_TOUT	Прерывание по факту истечения таймаута арбитража у одного или нескольких портов			
5	IRQ_BR_SPF	Прерывание по факту приема Broadcast кода стандарта SpFi			
4	IRQ_C01_C11	Прерывание по факту приема управляющего кода,			
3	IDO INT ACK	назначение которого не определено стандартом Прерывание по факту приема кода распределенного			
3	IRQ_INT_ACK	прерывание по факту приема кода распределенного прерывания/подтверждения			
2	IRQ TCODE	Прерывания по факту приема маркера времени			
1	IRQ_ICODE IRQ DISCONNECT	Прерывание по факту приема маркера времени Прерывание по факту разрыва соединения по одному			
		или нескольким SpFm портам			
0	IRQ_CONNECT	Прерывание по факту установки соединения по одному или нескольким SpFm портам			



Таблица 2.9. Формат регистра QSTR3

Номер Разряда	Условное обозначение прерывания	Название прерывания
31:24	RB_NAND_PORT3[7:0]	Прерывания по изменению состояния сигналов RB NAND_PORT3
23:16	RB_NAND_PORT2[7:0]	Прерывания по изменению состояния сигналов RB NAND_PORT2
15:8	RB_NAND_PORT1[7:0]	Прерывания по изменению состояния сигналов RB NAND_PORT1
7:0	RB_NAND_PORT0[7:0]	Прерывания по изменению состояния сигналов RB NAND_PORT0

Таблица 2.10. Формат регистра QSTR4

Номер Разряда	Условное обозначение прерывания	Название прерывания	
31:24	RB_NFC7[7:0]	Прерывания по изменению состояния сигналов RB NFC7	
23:16	RB_NFC6[7:0]	Прерывания по изменению состояния сигналов RB NFC26	
15:8	RB_NFC5[7:0]	Прерывания по изменению состояния сигналов RB NFC5	
7:0	RB_NFC4[7:0]	Прерывания по изменению состояния сигналов RB NFC4	

В регистрах QSTR3, QSTR4 запоминается изменение состояния сигналов RB из 0 в 1. Эти регистры доступны по чтению и записи. Обнуление запомненных прерываний осуществляется посредством записи 1 в соответствующий разряд.

Для управления режимом приема внешних прерываний nIRQ[3:0] имеется регистр IRQM, формат которого приведен в Таблица 2.11.

Таблица 2.11. Формат регистра IRQM

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:12	-	Не используется	-	0
11:8	IRQ_MODE	Режим приема внешнего прерывания nIRQ: 0 – потенциальные сигналы, активный низкий уровень; 1 – прерывание формируется при переходе состояния входного сигналы с высокого уровня на низкий уровень. Прерывание запоминается на регистре. Регистр обнуляется при помощи разрядов IRQ NULL	R/W	0
7:1	-	Не используется	-	0
0	IRQ_NULL	Обнуление запомненного прерывания при IRQ_MODE = 1. Прерывание nIRQ обнуляются при записи 1 в этот разряд	R/W1	0



2.4 Системные регистры

Формат регистров управления и состояния CSR0, CSR1 приведен в Таблица 2.12, Таблица 2.13.

Таблица 2.12. Формат регистра CSR0

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:15	-	Не используется	R	0
14	FLUSH_D0	При записи 1 в данный разряд кэш данных CPU0 останавливается в исходное состояние, то есть ее содержимое девалидируется. Эта процедура может использоваться для обеспечения когерентности кэш при работе DMA.	W	0
13	-	Не используется	R	0
12	FLUSH_I0	При записи 1 в данный разряд кэш команд CPU0 останавливается в исходное состояние, то есть ее содержимое девалидируется. Эта процедура может использоваться для обеспечения когерентности кэш при работе DMA.	W	0
11	TST_CACHE0	Режим работы кэш программ и кэш данных CPU0: 0 — нормальный режим; 1 — режим тестирования. Используется только при технологическом тестировании кэш программ. Пользователям устанавливать этот режим запрещено	R/W	0
10:2	-	Не используется	R	0
1	TR_CRAM0	Режим размещения векторов прерываний при BEV = 0 (регистр Status) CPU0: 0 — вектора прерываний размещаются во внешней памяти (базовый адрес 0x80000000); 1 — вектора прерываний размещаются во внутренней памяти CRAM0 (базовый адрес 0xB8000000)	RW	0
0	FM0	Режим преобразования виртуальных адресов CPU0 в физические адреса: 0 – с использованием TLB; 1 – Fixed Mapped (FM).	RW	1



Таблица 2.13. Формат регистра CSR1

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:15	-	Не используется	R	0
14	FLUSH_D1	При записи 1 в данный разряд кэш данных CPU1 останавливается в исходное состояние, то есть ее содержимое девалидируется. Эта процедура может использоваться для обеспечения когерентности кэш при работе DMA.	W	0
13	-	Не используется	R	0
12	FLUSH_I1	При записи 1 в данный разряд кэш команд CPU1 останавливается в исходное состояние, то есть ее содержимое девалидируется. Эта процедура может использоваться для обеспечения когерентности кэш при работе DMA.	W	0
11	TST_CACHE1	Режим работы кэш программ и кэш данных CPU1: 0 — нормальный режим; 1 — режим тестирования. Используется только при технологическом тестировании кэш программ. Пользователям устанавливать этот режим запрещено	R/W	0
10:2	-	Не используется	R	0
1	TR_CRAM1	Режим размещения векторов прерываний при BEV = 0 (регистр Status) CPU1: 0 — вектора прерываний размещаются во внешней памяти (базовый адрес 0x80000000); 1 — вектора прерываний размещаются во внутренней памяти CRAM1 (базовый адрес 0xB8400000)	RW	0
0	FM1	Режим преобразования виртуальных адресов CPU1 в физические адреса: 0 – с использованием TLB; 1 – Fixed Mapped (FM).	RW	1

Формат регистра ВООТ приведен в Таблица 2.14.

Таблица 2.14. Формат регистра ВООТ

Номер разряда	Условное Обозначение	Назначение	Доступ	Исходное состояние
31:1	-	Не используется	R	0
0	BOOT	Режим начальной загрузки: 0 — начальная загрузка не выполняется. CPU0 стартует с адреса 0x1800.0000 (CRAM0). Тестовый режим работы; 1 — начальная загрузка выполняется	R/W	1



2.5 Процедура начальной загрузки

По сигналу nRST (низкий уровень) все устройства микросхемы устанавливаются в исходное состояние. После его снятия (высокий уровень) частота от CPU1 отключена, а CPU0 начинает выполнять программу из ПЗУ начальной загрузки. Физический базовый адрес этого Π 3У -0х1830 $_{-}$ 000, объем - 256 байт.

При помощи программы начальной загрузки CPU0 по SPI загружает из внешней памяти типа NOR Flash в память CRAM0 программу объемом 64 32-разрядных слова, начиная с адреса 0x1800_0000. Затем CPU0 переходит на ее выполнение. Эта программа может выполнять, например, настройку регистров DDR_PORT, инициализацию DDR, а также загрузку основной программы из NOR Flash в CRAM0, CRAM1 или DDR и передачу управление ей. После этого при необходимости может быть включена частота CPU1 посредством записи «1» в разряд 1 регистра CLK_EN. После включения частоты CPU1 он всегда стартует с адреса 0x1840 0000.

Следует отметить, что основная программа или ее часть может быть загружена также через коммутатор SpFR или SWIC.

Возможен вариант работы с микросхемой 1892ВК016 без использования ПЗУ начальной загрузки. Он может быть, например, использован при работе микросхемы 1892ВК016 в составе отладочной платы. В этом случае при помощи отладочных средств по JTAG, не снимая сигнала nRST, микросхема 1892ВК016 переводится в режим отладки. Затем сигнал nRST снимается и осуществляется загрузка программ в CRAM0, CRAM1 или DDR. После чего CPU0 стартует с заданного адреса. Необходимо иметь в виду, что загрузка программ в DDR возможна только после настройки регистров DDR PORT и инициализации DDR.

Вывод CS микросхемы NOR Flash подключать к выводу CS шины SPI микросхемы 1892ВК016 (вывод AJ23) необходимо через внешнюю логику, обеспечивающую размыкание соединения при подаче сигнала nRST. Вариант схемы подключения вывода CS микросхемы NOR Flash к выводу CS шины SPI микросхемы 1892ВК016 приведен на Рисунок 2.1.



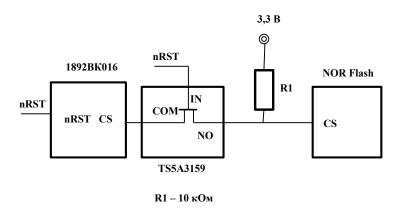


Рисунок 2.1. Схема подключения вывода CS микросхемы Flash к выводу CS (вывод AJ23) шины SPI микросхемы 1892ВК016

В качестве микросхемы Flash можно, например, использовать микросхему NX25P40.



3. ПРОЦЕССОРЫ СРИО, СРИ1

3.1 Основные характеристики

- архитектура MIPS32;
- 32-х битные пути передачи адреса и данных;
- кэш команд объемом 16 Кбайт;
- кэш данных объемом 16 Кбайт;
- архитектура привилегированных ресурсов в стиле ядра R4000:
 - регистры Count/Compare для прерываний реального времени;
 - отдельный вектор обработки исключений по прерываниям;
- программируемое устройство управления памятью:
 - два режима работы с TLB и Fixed Mapped (FM);
 - 16 строк в режиме TLB;
 - в режиме FM адресные пространства отображаются с использованием битов регистров;
- устройство умножения и деления;
- поддержка отладки JTAG.

3.2 Блок-схема

Блок схема процессоров СРИО, СРИ1 приведена на Рисунок 3.1.

Ядро содержит следующие узлы:

- устройство исполнения (Execution Core);
- устройство целочисленного умножения и деления (MDU);
- системный управляющий сопроцессор (СР0);
- сопроцессор арифметики в формате с плавающей точкой (FPU);
- устройство управления памятью (MMU Memory Management Unit);
- контроллер кэш (Cache Controller);
- устройство шинного интерфейса (BIU);
- кэш команд (Instruction Cache);
- кэш данных (Data Cache);
- преобразователь виртуального адреса в физический адрес (TLB/FM);
- средства отладки программ (OnCD On Chip Debugger) с JTAG портом.



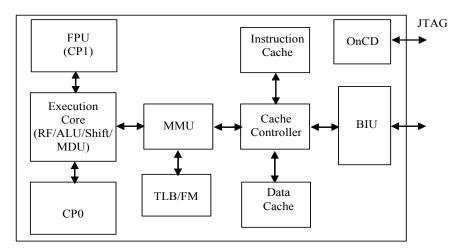


Рисунок 3.1. Блок схема процессоров CPU0, CPU1

3.3 Составляющие логические блоки

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

3.3.1 Устройство исполнения

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

В состав устройства исполнения входят:

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



3.3.2 Устройство умножения/деления (MDU)

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

3.3.3 Системный управляющий сопроцессор

Сопроцессор отвечает за преобразование виртуального адреса в физический, протоколы кэш, систему управления исключениями, выбор режима функционирования (Kernel/User) и за разрешение/запрещение прерываний. Конфигурационная информация доступна посредством чтения регистров СР0 (см. раздел 2.7 «Регистры СР0»).

3.3.4 Устройство управления памятью (MMU)

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

3.3.5 Контроллер кэш

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

3.3.6 Устройство шинного интерфейса (BIU – Bus Interface Unit)

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



3.3.7 OnCD контроллер

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

3.4 Конвейер

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

В этой главе содержатся следующие разделы:

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

3.4.1 Стадии конвейера

Конвейер содержит пять стадий:

выборка команды (стадия I- Instruction);
дешифрация команды (стадия D - Data);
исполнение команды (стадия E - Execution);
выборка из памяти (стадия М - Memory);
обратная запись (стадия W – Write Back).

На Рисунок 3.2 показаны операции, выполняемые RISC-ядром на каждом этапе конвейера.



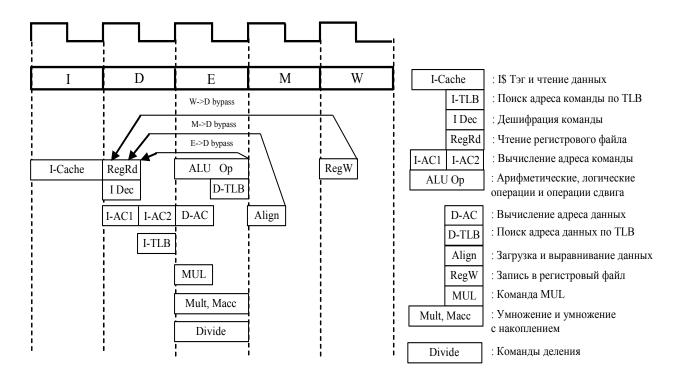


Рисунок 3.2

3.4.1.1 Стадия I: выборка команды

На этой стадии команда выбирается из командного кэш.

3.4.1.2 Стадия D: дешифрация команды

На этой стадии:

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



3.4.1.3 Стадия Е: исполнение

На этой стадии:

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

3.4.1.4 Стадия М: выборка из памяти

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

3.4.1.5 Стадия W: обратная запись

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

3.4.2 Операции умножения и деления

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

3.4.3 Задержка выполнения команд перехода (Jump, Branch)

Конвейер осуществляет выполнение команд перехода с задержкой в один такт. Однотактная задержка является результатом функционирования логики, ответственной за принятие решения о переходе на стадии D конвейера. Эта задержка позволяет использовать адрес перехода, вычисленный на предыдущей стадии, для доступа к команде на следующей D-стадии. Слот задержки перехода (branch delay slot) позволяет отказаться от остановок конвейера при переходе. Вычисление адреса и проверка условия перехода выполняются одновременно на стадии D. Итоговое значение PC (счетчика команд) используется для выборки очередной команды на стадии I, которая является второй командой после перехода. На Рисунок 3.3показан слот задержки перехода.



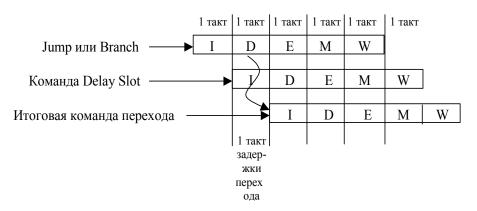


Рисунок 3.3. Слот задержки перехода

3.4.4 Обходные пути передачи данных (Data bypass)

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

Мультиплексоры обходных путей передачи данных для обоих операндов располагаются между регистровым файлом и ALU (Рисунок 3.4). Они позволяют передавать данные с выхода стадий E, M и W конвейера прямо на стадию D, если один из регистров источника (source) декодируемой команды совпадает с регистром назначения (target) одной из предшествующих команд. Входы мультиплексоров подключены к обходным путям $M \rightarrow D$ и $E \rightarrow D$, а также $W \rightarrow D$.

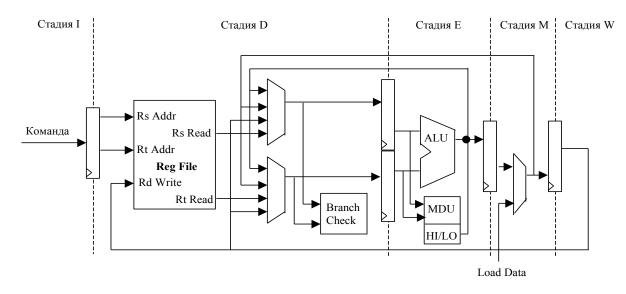


Рисунок 3.4



На Рисунок 3.5 показаны обходные пути передачи данных для команды Add1, за которой следует команда Sub2 и затем снова Add3. Поскольку команда Sub2 в качестве одного из операндов использует результат операции Add1, используется обходной путь $E \rightarrow D$. Следующая команда Add3 использует результаты обеих предшествующих операций: Add1 и Sub2. Так как данные команды Add1 в это время находятся на стадии M, используется обходной путь $M \rightarrow D$. Кроме того, вновь используется обходной путь $E \rightarrow D$ для передачи результата операции Sub2 команде Add3.

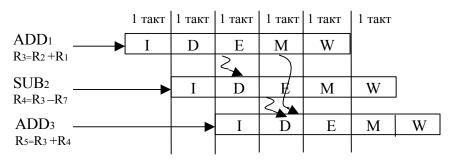


Рисунок 3.5

3.4.5 Задержка загрузки данных

Данные, выбираемые командами загрузки (Load), становятся доступными на конвейере только после выравнивания на стадии М. При этом данные, являющиеся исходными операндами, должны предоставляться командам для обработки уже на стадии D. Поэтому, если сразу за командой загрузки следует команда, для которой один из регистров исходных операндов совпадает с регистром, в который производится загрузка данных, это вызывает приостановку в работе конвейера на стадии D. Эта приостановка осуществляется аппаратной вставкой команды NOP. Во время этой задержки часть конвейера, которая находится дальше стадии D, продолжает продвигаться. Если же команда, использующая загружаемые данные, следует за командой загрузки не сразу, а через одну или через две, то для обеспечения бесперебойной работы конвейера используется один из обходных путей передачи данных: М—D или W—D (Рисунок 3.6).

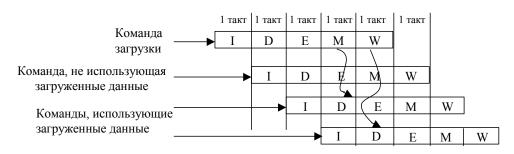


Рисунок 3.6



3.5 Устройство управления памятью (MMU)

3.5.1 Введение

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

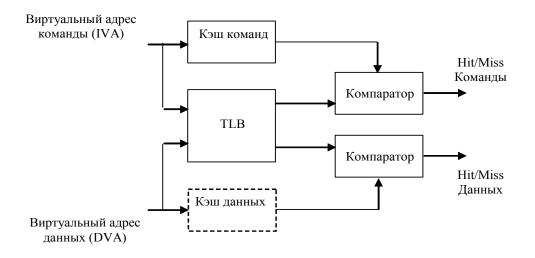
MMU может выполнять преобразование адресов в двух режимах: в режиме TLB и в режиме FM. Режим преобразования определяется битом FM регистра CSR.

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

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

На Рисунок 3.7 показано, взаимодействие MMU с процедурой доступа к кэш в режиме TLB, а на Рисунок 3.8 – в режиме FM.





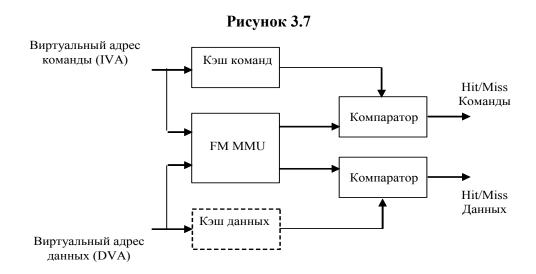


Рисунок 3.8

3.5.2 Режимы работы

Процессорное ядро поддерживает два режима работы:

- режим User (непривилегированный режим);
- режим Kernel (привилегированный режим).

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

Преобразования, выполняемые ММU, зависят от режима работы процессора.



3.5.2.1 Виртуальные сегменты памяти

Виртуальные сегменты памяти, на которые делится адресное пространство, различаются в зависимости от режима работы процессора. На Рисунок 3.9 показана сегментация для 4 Гбайт (2^{32} байт) виртуального адресного пространства, адресуемого 32-разрядным виртуальным адресом для обоих режимов работы.

Ядро входит в режим Kernel после аппаратного сброса или когда происходит исключение. В режиме Kernel программное обеспечение имеет доступ к полному адресному пространству и ко всем регистрам CP0. В режиме User доступ ограничен подмножеством виртуального адресного пространства (0x0000_0000 - 0x7FFF_FFFF) и запрещен доступ к функциям CP0. В режиме User недоступны виртуальные адреса 0x8000_0000 - 0xFFFF FFFF и обращение к ним вызывает исключение.

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

Рисунок 3.9. Карта виртуальной памяти для режимов User и Kernel

Каждый из сегментов, показанных на Рисунок 3.9, является либо отображаемым (mapped), либо неотображаемым (unmapped). Различие объясняется в следующих двух разделах.

3.5.2.1.1 Неотображаемые сегменты

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



Для неотображаемых сегментов преобразование виртуального адреса в физический является фиксированным.

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

3.5.2.1.2 Отображаемые сегменты

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

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

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

3.5.2.2 Режим User

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

На Рисунок 3.10 показано размещение виртуального адресного пространства режима User.

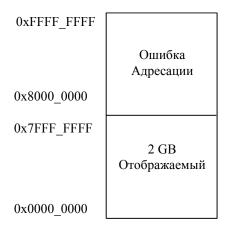


Рисунок 3.10

Сегмент потребителя начинается с адреса 0x0000_0000 и заканчивается адресом 0x7FFF_FFFF. Обращения по всем остальным адресам вызывают прерывания по ошибке адресации.



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

- UM = 1;
- EXL = 0;
- ERL = 0.

В Таблица 3.1 приводятся характеристики сегмента useg режима User.

Таблица 3.1

Адрес	Регистр Состояния Имя		Диапазон	Размер сегмента			
пдрес	EXL	ERL	UM	сегмента	адресов	тазмер сегмента	
					0x0000_0000	2GB	
A(31)=0	0	0	1	useg	\rightarrow	(2 ³¹ байт)	
					0x7FFF FFFF		

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

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

В режиме FM, область виртуальных адресов $0x0000_0000-0x7FFF_FFFF$ преобразуется в область физических адресов $0x4000_0000-0xBFFF_FFFF$. Кэшируемость задается полем KU регистра Config CP0.

3.5.2.3 Режим Kernel

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

- UM = 0;
- ERL = 1;
- EXL = 1.

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



Виртуальное адресное пространство режима Kernel разделено на области в соответствии со значением старших битов виртуального адреса, как показано на Рисунок 3.11. Кроме того, в Таблица 3.2 содержатся характеристики сегментов режима Kernel.

0xFFFF_FFFF 0xE000_0000	Kernel virtual address spase Mapped, 512 MB	kseg3
0xDFFF_FFFF 0xC000_0000	Kernel virtual address spase Mapped, 512 MB	kseg2
0xBFFF_FFFF 0xA000 0000	Kernel virtual address spase Unmapped, Uncached, 512 MB	kseg1
0x9FFF_FFFF 0x8000_0000	Kernel virtual address spase Unmapped, 512 MB	kseg0
0x7FFF_FFFF	Mapped, 2048 MB	kuseg
0x0000_0000		

Рисунок 3.11

Таблица 3.2

Адрес	Peгистр Состояния EXL ERL UM	Имя сегмента	Диапазон адресов	Размер сегмента
A(31)=0	UM = 0	kuseg	0x0000_0000 → 0x7FFF_FFFF	2 GB (2 ³¹)
A(31:29)=100 ₂	или	kseg0	0x8000_0000 → 0x9FFF FFFF	512 MB (2 ²⁹)
A(31:29)=101 ₂	EXL=1 или	kseg1	0xA000_0000 → 0xBFFF FFFF	512 MB (2 ²⁹)
A(31:29)=110 ₂	ERL=1	kseg2	0xC000_0000 → 0xDFFF_FFFF	512 MB (2 ²⁹)
A(31:29)=111 ₂		kseg3	0xE000_0000 → 0xFFFF_FFFF	512 MB (2 ²⁹)



3.5.2.3.1 Режим Kernel, Пространство пользователя (kuseg)

Если старший значащий бит виртуального адреса A[31]=0, то выбирается виртуальное адресное пространство kuseg объемом 2 Гбайт, отображенное на адреса $0x0000_0000 - 0x7FFF$ FFFF.

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

При ERL=0 в режиме FM, область виртуальных адресов 0x0000_0000-0x7FFF_FFFF преобразуется в область физических адресов 0x4000_0000-0xBFFF_FFFF. Кэшируемость задается полем KU регистра Config CP0.

При ERL = 1 в режимах TLB и FM, область адресов пользователя становится неотображаемым и некэшируемым адресным пространством. Виртуальный адрес kuseg соответствует тому же физическому адресу и не включает поле ASID. То есть, область виртуальных адресов kuseg соответствует области физических адресов $0x0000_0000-0x7FFF$ FFFF.

3.5.2.3.2 Режим Kernel, пространство 0 режима Kernel (kseg0)

Если в режиме Kernel три старших бита виртуального адреса равны 100_2 , выбирается виртуальное адресное пространство kseg0. Это область размером 2^{29} байт (512 MB), которая расположена внутри границ, определяемых адресами $0x8000_0000$ и 0x9FFF FFFF.

Вне зависимости от состояния бита ERL и режима работы ссылки к kseg0 не отображаются, а физический адрес получается вычитанием 0x8000_0000 из виртуального адреса. Кэшируемость сегмента kseg0 определяется значением поля K0 регистра Config CP0.

3.5.2.3.3 Режим Kernel, пространство 1 режима Kernel (kseg1)

Если в режиме Kernel три старших бита виртуального адреса равны 101_2 , выбирается виртуальное адресное пространство kseg1. Это область размером 2^{29} байт (512 MB), которая расположена внутри границ, определяемых адресами $0xA000_0000$ и $0xBFFF_FFF$.

Вне зависимости от состояния бита ERL и режима работы ссылки к kseg1 не отображаются, а физический адрес получается вычитанием 0xA000_0000 из виртуального адреса.



3.5.2.3.4 Режим Kernel, пространство 2 режима Kernel (kseg2)

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

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

В режиме FM вне зависимости от состояния бита ERL это виртуальное пространство зафиксировано в физических адресах 0xC000_0000 - 0xDFFF_FFFF и его кэшируемость определяется полем K23 Регистра Config CP0.

3.5.2.3.5 Режим Kernel, пространство 3 режима Kernel (kseg3)

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

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

В режиме FM вне зависимости от состояния бита ERL это виртуальное пространство зафиксировано в физических адресах 0xE000_0000 - 0xFFFF_FFFF и его кэшируемость определяется полем K23 регистра Config.

3.5.3 Буфер быстрого преобразования адреса (TLB)

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

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

Буфер TLB организован в виде страничных пар для минимизации общего количества хранящейся информации. Каждая строка тэговой порции соответствует двум физическим



строкам данных — строке четных страниц и строке нечетных страниц. Самый старший разряд виртуального адреса, не участвующий в сравнении тэгов, определяет какая строка из двух строк данных используется. Поскольку размер страницы может варьироваться для каждой пары страниц, определение адресных разрядов, участвующих в сравнении и разряда, задающего четность страницы, должно осуществляться динамически при поиске по TLB.

На Рисунок 3.12 показано содержание одной из 16 двойных строк TLB.

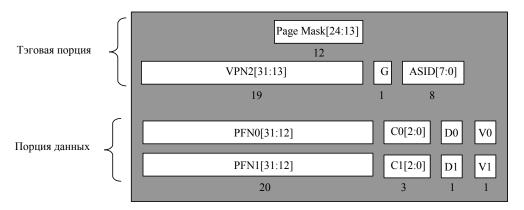


Рисунок 3.12

Описание полей строки TLB приведены в Таблица 3.3.

Таблица 3.3

Название поля	Описание			
	Значение маски размера страницы. Определяет размер страницы маскировкой соответствующих разрядов VPN2, и тем самым исключением их из рассмотрения. Также используется для задания адресного разряда, определяющего четность страницы (PFN0-PFN1). См. следующую таблицу:			
	Page Mask[11:]	Размер страницы	Бит определения четности	
	0000_0000_0000	4 КБ	VAddr[12]	
	0000_ 000_0011	16 КБ	VAddr[14]	
	0000_0000_1111	64 КБ	VAddr[16]	
	0000_0011_1111	256 КБ	VAddr[18]	
7 7 1 5 4 4 4 3 7	0000_1111_1111	1 МБ	VAddr 20]	
Page Mask[24:13]	0011 1 _ 1	4 МБ	VAddr[22]	
	1111 111 1111	16 МБ	VAddr[2	
	Поскольку каждая г значение, физическа Page Mask, содержа значение всегда пре	пара битов этого полая строка в TLB сод щую только 6 бит. Собразуется в 12-биту, что при кэшируе	мых ссылках, страницы	



Название поля		Описание	
VPN2[31:13]	Виртуальный номер страницы, поделенный на 2. Данное поле содержит старшие разряды виртуального номера страницы. Виртуальный номер разделен на 2 потому, что он соответствует паре страниц TLB. Разряды 31:25 всегда участвуют в сравнении. Участие в сравнении разрядов 24:13 зависит от размера страницы, задаваемого полем Page Mask.		
G	Бит глобальности. Если он установлен, данная строка является глобальной для всех процессов и подпроцессов, и таким образом, поле ASID исключается из рассмотрения.		
ASID[7:0]	Идентификатор ад	ресного пространства. Определяет процесс или орым ассоциируется данная строка TLB.	
PFN0[31:12], PFN1[31:12]		о кадра. Задает старшие разряды физического щ размером более 4 Кбайт используется го поля.	
C0[2:0], C1[2:0]	кэшируемости и ог	держит закодированное значение атрибута пределяет должна ли страница помещаться в кэш ируется следующим образом: Атрибуты когерентности При записи преобразуется в код 011 При записи преобразуется в код 011 Некэшируемая страница Кэшируемая страница При записи преобразуется в код 011	
D0, D1	«Dirty» (Грязная страница) — бит разрешения записи. Показывает, что в страницу была сделана запись и/или разрешена запись в данную страницу. Если этот бит установлен, разрешены операции сохранения в данной странице. Если не установлен, сохранения в		
V0, V1	данной странице будут вызывать исключения модификации. Бит валидности. Показывает, что данная строка TLB и, соответственно, отображение виртуальной страницы, действительны. Если этот бит установлен, то обращения к данной странице разрешены. Если не установлен, то обращения к странице будут вызывать исключения TLB (TLB invalid).		

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

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



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

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

3.5.4 Преобразование виртуального адреса в физический в режиме TLB.

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

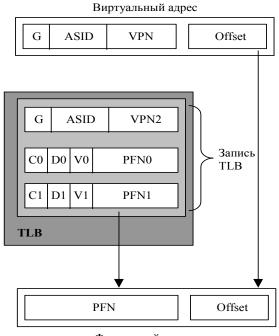
- установлен бит глобальности (G) для четных и нечетных страниц в строке TLB;
- поле ASID виртуального адреса совпадает с полем ASID строки TLB.

Это соответствие называется попаданием TLB. Если не имеется ни одного соответствия, возникает исключение промаха TLB и программному обеспечению дается возможность пополнить TLB из расположенной в памяти таблицы страниц виртуальных /физических адресов. На Рисунок 3.13 показана логика преобразования виртуального адреса в физический.

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



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



Физический адрес

Рисунок 3.13

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

На Рисунок 3.14 показана блок-схема процесса преобразования адреса. В верхней части рисунка показан виртуальный адрес для страницы размером 4 Кбайт. Ширина поля смещения определяется размером страницы.

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



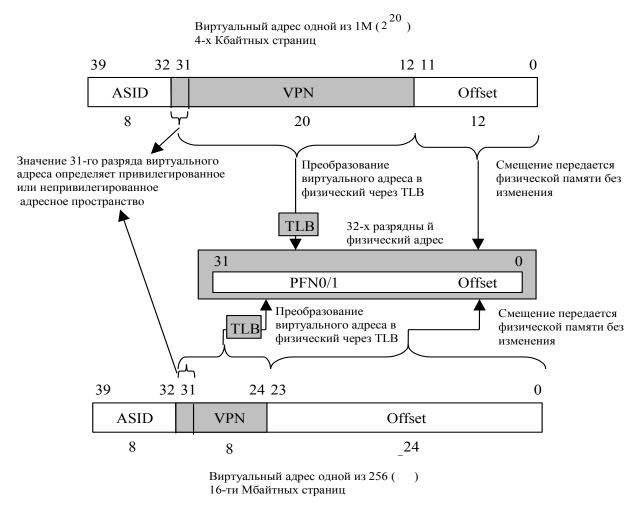


Рисунок 3.14

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

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

Если соответствие не найдено, возникает исключение TLB Refill, и программное обеспечение пополняет TLB из таблицы страниц, находящейся в памяти. На Рисунок 3.15 показан алгоритм преобразования и условия возникновения исключений TLB.

Программное обеспечение может делать записи в конкретные строки TLB или использовать аппаратный механизм записи в случайно выбранные строки. Регистр Random определяет, в какую строку будет сделана запись командой TLBWR. Этот регистр



декрементируется на каждом такте продвижения конвейера, возвращаясь к максимальному значению после достижения величины, равной значению регистра Wired. Таким образом, строки TLB, чей номер меньше значения регистра Wired, не затрагиваются командой TLBWR, что позволяет зарезервировать TLB-отображения первостепенной важности.

В режиме TLB также реализован механизм сравнения при записи с целью предотвращения возникновения нескольких соответствий (множественных попаданий). Работает он следующим образом. При выполнении операции записи в TLB, поле VPN2 сравнивается с одноименными полями всех строк TLB. Если будет найдено соответствие, возникнет аппаратно обрабатываемое исключение, которое установит бит TS регистра Status CP0 и прервет эту операцию. Подробно исключения описаны в п. 0. В каждой строке TLB имеется скрытый бит, обнуляемый при аппаратном сбросе. Устанавливается этот бит при записи в данную строку, разрешая просмотр этой строки при поисках соответствий. Поэтому непроинициализированные строки не вызывают неадекватные преобразования адресов.

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

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

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

3.5.4.2 Размеры страниц и алгоритм замещения

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



Второй механизм управляет замещением, когда возникает промах при просмотре TLB. Для выбора строки TLB, в которую будет записано новое отображение, в процессорном ядре предусмотрен алгоритм случайного замещения. Но существует также способ программно предотвратить случайное замещение зарезервированных отображений, количество которых определяется значением регистра Wired CP0. (см. также п. 3.7.3.6).

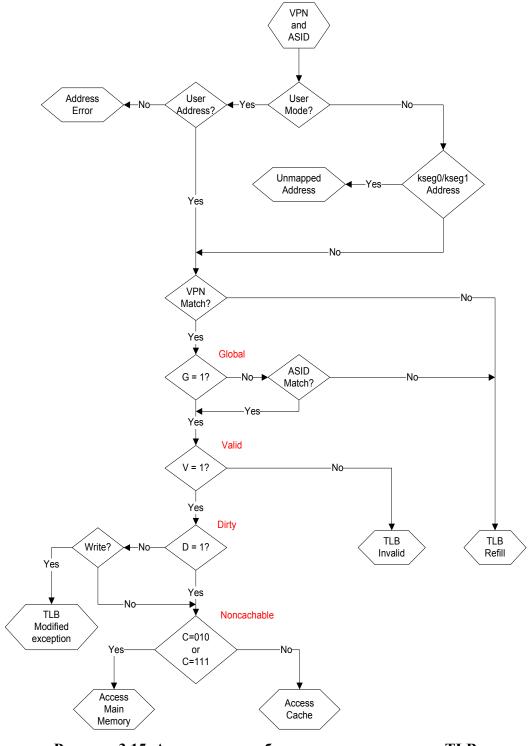


Рисунок 3.15. Алгоритм преобразования адреса через TLB



3.6 Исключения

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

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

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

3.6.1 Условия исключений

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

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

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



3.6.2 Приоритеты исключений

В Таблица 3.4 перечислены все возможные исключения со своими относительными приоритетами от высшего к низшему. Некоторые из этих исключений могут случаться одновременно, в этом случае вызывается исключение с наивысшим приоритетом.

Таблица 3.4

Исключение	Описание
Reset	Аппаратный сброс
NMI	Внешнее немаскируемое прерывание и прерывание от таймера WDT (см.
	табл. 7.2).
TLB_Ri,	Промах TLB при выборке команды,
TLB_Ii	Попадание в запрещенную страницу TLB (V=0) при выборке команды
AdELi	Ошибка выравнивания адреса при выборке команды;
	Ссылка на адрес режима Kernel при работе в режиме User при выборке команды
MCheck	Запись в TLB, создающая конфликт с существующей строкой TLB
Sys	Выполнение команды SYSCALL
Bp	Выполнение команды BREAK
CpU	Выполнение команды сопроцессора в режиме User
RI	Выполнение зарезервированной команды
Ov	Переполнение в арифметической команде
Tr	Выполнение trap (когда условие trap истинно)
AdELd	Ошибка выравнивания адреса при загрузке данных;
	Ссылка на адрес режима Kernel при работе в режиме User при загрузке
AdES	данных
	Ошибка выравнивания адреса при сохранении данных;
	Попытка сохранения по адресу Kernel в режиме User
TLB_Rd,	Промах TLB при загрузке данных;
TLB_Id	Попадание в запрещенную страницу TLB (V=0) при загрузке данных
TLB_M	Сохранение в TLB-странице с D=0
Interrupt	Установка немаскируемых HW или SW - прерываний

3.6.3 Расположение векторов исключений

Векторы исключений аппаратного сброса и NMI всегда находятся по адресу 0xBFC_0000. Адреса всех других исключений являются комбинациями векторных смещений и базового адреса. В Таблица 3.5 приведены базовые адреса как функции исключения и состояния бита BEV Регистра Status. В Таблица 3.6 приведены смещения от базового адреса как функции исключения. В Таблица 3.7 эти две таблицы сведены в одну таблицу, содержащую все возможные адреса векторов исключений как функции состояний, влияющих на выбор этих векторов.



Таблица 3.5

Исключение	Statusbev		
TERMO TERME	0	1	
Reset	0xBFC0_0000		
Остальные исключения для	0х8000_0000 при	0xBFC0_0200	
CPU0	TR_CRAM0=0 регистра CSR0		
	(Таблица 2.12);		
	0хВ800_0000 при		
	TR_CRAM0=1 регистра CSR0		
Остальные исключения для	0х8000_0000 при	0xBFC0_0200	
CPU1	TR_CRAM1=0 регистра CSR1		
	(Таблица 2.13);		
	0хВ840_0000 при		
	TR_CRAM1=1 регистра CSR1		

Следует отметить, что виртуальный базовый адрес исключений $0x8000_0000$ будет размещен во внешней памяти по физическому адресу $0x0000_0000$. Виртуальный базовый адрес исключений $0xBFC0_0000$ будет размещен во внешней памяти по физическому адресу, который зависит от объема внешней памяти следующим образом: 32 Мбайт – $0x01C0_0000$, 64 Мбайт – $0x03C0_0000$, 128 Мбайт – $0x07C0_0000$.

Таблица 3.6. Базовые адреса векторов исключений

Исключение	Смещение вектора
TLB Refill, EXL = 0	0x000
Reset	0x000
Исключения общего характера (General	0x180
Exeptions)	
Interrupt, Cause _{IV} = 1	0x200

Таблица 3.7. Векторы исключений

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



3.6.4 Обработка общих исключений

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

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

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



```
Operation:
if StatusEXL == 0 then
if InstructionInBranchDelaySlot then
EPC <= PC - 4
CauseBD <= 1
else
EPC <= PC
CauseBD <= 0
if (ExceptionType == TLBRefill) then
vectorOffset <= 0x000</pre>
elseif (ExceptionType == Interrupt) and
(CauseIV == 1) then
vectorOffset <= 0x200
else
vectorOffset <= 0x180
endif
vectorOffset <= 0x180</pre>
endif
CauseCE <= FaultingCoprocessorNumber</pre>
CauseExcCode <= ExceptionType</pre>
StatusEXL <= 1
if (StatusBEV == 1) then
PC <= 0xBFC0 0200 + vectorOffset
PC \le 0x8000 0000 + vectorOffset
Endif
```

3.6.5 Исключения

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

3.6.5.1 Исключение по аппаратному сбросу (Reset Exception)

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

- регистр Random устанавливается в значение, равное количеству строк TLB 1;
- регистр Wired устанавливается в 0;
- регистр Config устанавливается в свое начальное состояние (boot state);
- поля BEV, TS, NMI и ERL Регистра Status устанавливаются в заданные значения;
- в PC загружается значение 0xBFC0 0000 (виртуальный адрес).



```
Вектор исключения:
Reset (0xBFC0_0000)
Operation:
Random <= TLBEntries - 1
Wired <= 0
Config <= ConfigurationState
Status<sub>BEV</sub> <= 1
Status<sub>TS</sub> <= 0
Status<sub>NMI</sub> <= 0
Status<sub>ERL</sub> <= 1
PC <= 0xBFC0_0000
```

3.6.5.2 Исключение по немаскируемому прерыванию (Non Maskable Interrupt – NMI Exception)

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

- поля BEV, TS, NMI и ERL регистра Status принимают заданные значения;
- в регистр ErrorEPC загружается значение PC 4, если прерывание произошло на фоне команды в слоте задержки перехода. В противном случае в регистр ErrorEPC загружается значение PC;
- в PC загружается значение 0xBFC0 0000.

```
Вектор исключения:
Reset (0xBFC0_0000)
Operation:
StatusBEV <= 1
StatusTS <= 0
StatusNMI <= 1

StatusERL <= 1
if InstructionInBranchDelaySlot then
ErrorEPC <= PC - 4
else
ErrorEPC <= PC
endif
PC <= 0xBFC0 0000
```

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

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



Значение поля ExcCode регистра Cause:

TLBL: Произошла ссылка по загрузке данных или выборке команды

TLBS: Произошла ссылка по сохранению данных

Дополнительно сохраняемые состояния:

Таблина 3.8

Состояние регистра	Значение
BadVAddr	ошибочный адрес
Context	поле BadVPN2 содержит VA _{31:13} ошибочного адреса
EntryHi	поле VPN2 содержит VA _{31:13} ошибочного адреса; поле
	ASID содержит ASID отсутствующей ссылки

Вектор исключения:

Вектор TLB Refill (смещение 0х000)

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

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

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

Значение поля ExcCode регистра Cause:

TLBL: Произошла ссылка по загрузке данных или выборке команды

TLBS: Произошла ссылка по сохранению данных

Дополнительно сохраняемые состояния:

Таблица 3.9

Состояние регистра	Значение
BadVAddr	ошибочный адрес
Context	поле BadVPN2 содержит VA31:13 ошибочного адреса
EntryHi	поле VPN2 содержит VA _{31:13} ошибочного адреса; поле
	ASID содержит ASID отсутствующей ссылки

Вектор исключения:

Общий Вектор исключения (смещение 0х180)



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

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

- выбрать команду, загрузить или сохранить слово данных, если они не выровнены в границах слова;
- загрузить или сохранить половину слова, если оно не выровнено в границах половины слова;
- обратиться по адресу пространства Kernel при работе в режиме User.

Значение поля ExcCode регистра Cause:

ADEL: Произошла ссылка по загрузке данных или выборке команды

ADES: Произошла ссылка по сохранению данных

Дополнительно сохраняемые состояния:

Таблица 3.10

Состояние регистра	Значение
BadVAddr	ошибочный адрес

Вектор исключения:

Общий Вектор исключения (смещение 0х180)

3.6.5.6 Исключение по аппаратному контролю (Mcheck – Machine Check Exception)

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

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

Значение поля ExcCode регистра Cause:

Mcheck

Дополнительно сохраняемые состояния:

Нет



Вектор исключения:

Общий Вектор исключения (смещение 0х180)

3.6.5.7 Исключение исполнения – системный вызов (System Call Exception)

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

исполнении команды SYSCALL.

Значение поля ExcCode регистра Cause:

Sys

Дополнительно сохраняемые состояния:

Нет

Вектор исключения:

Общий Вектор исключения (смещение 0х180)

3.6.5.8 Исключение исполнения — Breakpoint (Execution Exception – Breakpoint)

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

Значение поля ExcCode регистра Cause:

Bp

Дополнительно сохраняемые состояния:

Нет

Вектор исключения:

Общий Вектор исключения (смещение 0х180)

CpU

Нет

Вектор исключения:



3.6.5.9 Исключение исполнения — зарезервированная команда (Execution Exception – Reserved Instruction)

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

кодом операции или полем функции.
Значение поля ExcCode регистра Cause:
RI
Дополнительно сохраняемые состояния:
Нет
Вектор исключения:
Общий Вектор исключения (смещение 0х180)
3.6.5.10 Исключение исполнения — недоступен сопроцессор (Execution Exception – Coprocessor Unusable)
Исключение недоступности сопроцессора является одним из шести исключений исполнения. Все такие исключения имеют одинаковый приоритет. Исключение недоступности сопроцессора вызывается при попытке исполнения команды сопроцессора CP0 в режиме User.
Значение поля ExcCode регистра Cause:

Общий Вектор исключения (смещение 0х180)

Дополнительно сохраняемые состояния:

Вектор исключения:

Общий Вектор исключения (смещение 0х180)



3.6.5.11 Исключение исполнения — целочисленное переполнение (Execution Exception – Integer Overflow)

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

The same of the sa
Значение поля ExcCode регистра Cause:
Ov
Дополнительно сохраняемые состояния:
Нет
Вектор исключения:
Общий Вектор исключения (смещение 0х180)
3.6.5.12 Исключение исполнения — Trap (Execution Exception – Trap)
Исключение Trap является одним из шести исключений исполнения. Все такие исключения имеют одинаковый приоритет. Исключение Trap вызывается, если условие команды trap истинно (TRUE).
Значение поля ExcCode регистра Cause:
Tr
Дополнительно сохраняемые состояния:

3.6.5.13 Исключение сохранения в запрещенной области (TLB Modified Exception)

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

Найденная строка TLB действительна, но страница запрещена для записи.



Значение поля ExcCode регистра Cause:

Mod

Дополнительно сохраняемые состояния:

Таблина 3.11

Состояние регистра	Значение
BadVAddr	Ошибочный адрес
Context	Поля BadVPN2 содержат VA31:13 ошибочного адреса
EntryHi	Поле VPN2 содержит VA _{31:13} ошибочного адреса;
	поле ASID содержит ASID отсутствующей ссылки

Вектор исключения:

Общий Вектор исключения (смещение 0х180)

3.6.5.14 Исключение прерывания (Interrupt Exception)

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

Значение поля ExcCode регистра Cause:

Int

Дополнительно сохраняемые состояния:

Таблица 3.12

Состояние регистра	Значение
Cause _{IP}	Указывает код прерывания

Вектор исключения:

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

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

3.6.6 Алгоритмы обработки исключений

В этом разделе приведены алгоритмы обработки следующих исключений:

- общие исключения;
- исключения пропуска при поиске по TLB;
- исключения Reset и NMI.

Исключения аппаратно обрабатываются, а затем программно обслуживаются.



Алгоритмы обработки исключений приведены на Рисунок 3.16, Рисунок 3.17, Рисунок 3.18.

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

замаскированы битами IE и IM Комментарий EntryHi и Context устанавливаются EntryHi <= VPN2, ASID только для исключений TLB- Invalid, Context <= VPN2 Modified, Refill и для исключений VCED/ Set Cause EXCCode,CE I. Не устанавливаются в случае Bus BadVA <= VA Error =1 **EXL** =0 Instr. in Yes No Br. Dly. Slot? EPC <= (PC - 4) EPC <= PC Cause.BD <= 1 Cause.BD <= 0 EXL <= 1 Процессор переводится в режим Kernel и прерывания запрещаются =0 (normal) =1 (bootstrap) Status.BEV PC <= 0x8000 0000 + 180 PC <= 0xBFC0_0200 + 180 (unmapped, cached) (unmapped, uncached) Программное обслуживание

Рисунок 3.16. Обработка общих исключений

исключений



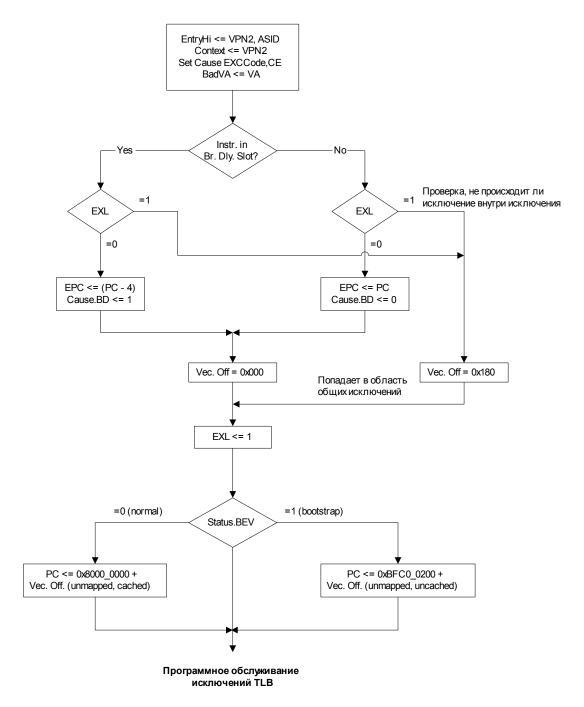


Рисунок 3.17. Обработка исключений TLB Refill и TLB Invalid



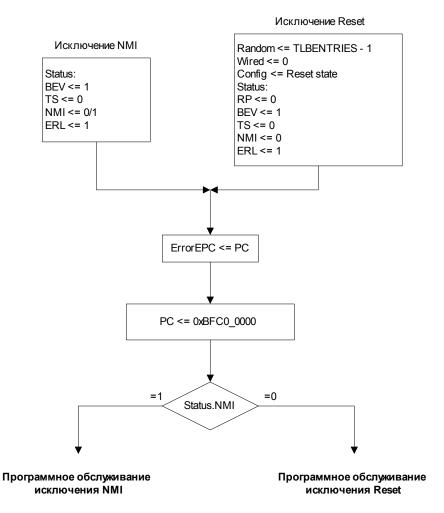


Рисунок 3.18. Обработка исключений Reset и NMI

3.7 Регистры СР0

3.7.1 Назначение

Системный Управляющий Сопроцессор (СР0) обеспечивает регистровый интерфейс с процессорным ядром MIPS32 и поддерживает управление памятью, преобразование адреса, обработку исключений и другие привилегированные операции. Каждому регистру СР0 соответствует определяющий его уникальный номер; этот номер называется номером регистра. Например, регистру PageMask соответствует 5-й номер регистра.

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



3.7.2 Обзор регистров СР0

В Таблица 3.13 приведены все регистры СРО в порядке возрастания нумерации. В разделе 5.3 каждый из этих регистров описан отдельно.

Таблица 3.13. Регистры СР0

1Random¹Случайным от TLB (режим²2EntryLo0¹Младшая час четными ном3EntryLo1¹Младшая час нечетными нечетными нечетными нечетными нечетными нечетными нечетными нетрежим TLB)5PageMask¹Управление нетрежим TLB (режим TLB)6Wired¹Управление нетрежим тЕВ (режим TLB)7ReservedРезерв8BadVAddr²Содержит ададресацией и дадресацией и дадре	ть строки TLB для виртуальных страниц с ерами (режим TLB) сть строки TLB для виртуальных страниц с
2EntryLo0¹Младшая час четными ном3EntryLo1¹Младшая час нечетными телем (режим ТLВ)5PageMask¹Управление пететрок TLB (режим ТLВ)6Wired¹Управление нетотрок TLB (режим ТLВ)7ReservedРезерв8BadVAddr²Содержит ададерсацией и даресацией и	TLB) ть строки TLB для виртуальных страниц с верами (режим TLB) сть строки TLB для виртуальных страниц с
Четными ном3EntryLo1¹Младшая час нечетными нечетными нечетными нечетными не тLB)4Context²Указатель на TLB)5PageMask¹Управление петрежим TLB (режим TLB)6Wired¹Управление петрок TLB (режим TLB)7ReservedРезерв8BadVAddr²Содержит адаресацией и даресацией и да	ерами (режим TLB) сть строки TLB для виртуальных страниц с
4Context2Указатель на TLB)5PageMask1Управление и (режим TLB)6Wired1Управление и строк TLB (р7ReservedPeзерв8BadVAddr2Содержит адрадресацией и адресацией и даресацией и даресацие	1 1 1
4Context2Указатель на TLB)5PageMask1Управление и (режим TLB)6Wired1Управление и строк TLB (р7ReservedРезерв8BadVAddr2Содержит ад адресацией и адресацией и от и о	омерами (режим TLB)
6 Wired¹ Управление в строк TLB (р 7 Reserved Резерв 8 BadVAddr² Содержит адуадресацией и адресацией и адресацией и адресацией и по в процем в про	строку в таблице страниц памяти (режим
строк TLB (р 7 Reserved Резерв 8 BadVAddr² Содержит адјадресацией и адресацией и адресацией и адресацией и адресацией и адресацией и адресацией и по	переменным размером страниц строк TLB
8 BadVAddr² Содержит адрадресацией и 9 Count² Счетчик прогование и 10 EntryHi¹ Старшая часта и 11 Compare² Управление и 12 Status² Состояние и 13 Cause² Причина послание и 14 EPC² Значение сче исключения 15 PRId Идентификал 15 EBase Номер проце 16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	количеством закрепленных "привязанных" ежим TLB)
9 Count² Счетчик проп 10 EntryHi¹ Старшая част 11 Compare² Управление п 12 Status² Состояние и п 13 Cause² Причина послиключения 14 EPC² Значение сченисключения 15 PRId Идентификал 15 EBase Номер проце 16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	·
9 Count² Счетчик проп 10 EntryHi¹ Старшая част 11 Compare² Управление п 12 Status² Состояние и п 13 Cause² Причина послисключения 14 EPC² Значение счей исключения 15 PRId Идентификал 15 EBase Номер проце 16 Config/Config¹ Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	рес, вызвавший последнее связанное с сключение
10 EntryHi¹ Старшая част 11 Compare² Управление п 12 Status² Состояние и 13 Cause² Причина пос. 14 EPC² Значение сче исключения 15 PRId Идентификац 15 EBase Номер проце 16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	цессорных циклов
11 Compare² Управление и 12 Status² Состояние и 13 Cause² Причина пос. 14 EPC² Значение сче исключения 15 PRId Идентификал 15 EBase Номер проце 16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	гь строки TLB (режим TLB)
12 Status² Состояние и 13 Cause² Причина пос. 14 EPC² Значение сче исключения 15 PRId Идентификал 15 EBase Номер проце 16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	трерыванием таймера
14 EPC² Значение сче исключения 15 PRId Идентификац 15 EBase Номер проце 16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	управление процессором
15PRIdИдентифика15EBaseНомер проце16Config/Config1Конфигураци17LLAddrЗагрузка адре18-19Не реализованы	леднего исключения
15 EBase Номер проце 16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	тчика команд во время последнего
15 EBase Номер проце 16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	ция и ревизия процессора
16 Config/Config1 Конфигураци 17 LLAddr Загрузка адре 18-19 Не реализованы	ссора в многопроцессорной системе
18-19 Не реализованы	ионный регистр
18-19 Не реализованы	еса сопряжения
20.22 Deserved Deserved	•
20-22 Reserved Резерв	
23-24 Не реализованы	
25-27 Reserved Резерв	
28-29 Не реализованы	
30 ErrorEPC ² Значение сче	тчика команд при последней ошибке
31 Не реализован	i iina nomang nph nooneghen omnoke

¹Регистры, используемые при управлении памятью.

²Регистры, используемые при обработке исключений.



3.7.3 Регистры СР0

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

Все поля описанных регистров характеризуются свойствами записи / чтения, а также значением после аппаратного сброса. Свойства записи / чтения охарактеризованы в Таблица 3.14.

Таблица 3.14

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

3.7.3.1 Регистр Index (Регистр 0 CP0, Select 0)

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

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



Формат регистра Index

3	1 30	4	3	0
P	0		Index	

Таблица 3.15. Описание полей регистра Index

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

3.7.3.2 Peructp Random (Peructp CP0 1, Select 0)

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

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

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

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

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

Формат регистра Random

	31 4	3 0)
(0	Random	

Таблица 3.16. Описание полей регистра Random

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



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

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

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

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

Формат регистров EntryLo0, EntryLo1

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

Таблица 3.17. Описание полей регистров EntryLo0 и EntryLo1

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

В Таблица 3.18. приведена кодировка для поля С регистров EntryLo0 и EntryLo1 и полей K0, K23 и KU регистра Config.



Таблица 3.18. Атрибуты когерентности кэш

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

3.7.3.4 Peructp Context (Peructp 4 CP0, Select 0)

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

При возникновении исключения TLB (TLB Refill, TLB Invalid, или TLB Modified) биты $VA_{31:13}$ виртуального адреса записываются в поле BadVPN2 регистра Context. Поле PTEBase записывается и используется операционной системой.

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

Формат регистра Context

31	23 22	4 3	0
PTEBase	BadVPN2		

Таблица 3.19. Описание полей регистра Context

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



3.7.3.5 Peructp PageMask (Peructp 5 CP0, Select 0)

Регистр PageMask доступен для чтения и записи, и используется для чтения TLB и записи в TLB. Он содержит маску сравнения, которая устанавливает переменную размера страниц для каждой строки TLB, как показано в Таблица 3.21.

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

Формат регистра PageMask

31	25 24	13 12	0
0	Mask	0	

Таблица 3.20. Описание полей регистра PageMask

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

Таблица 3.21. Таблица возможных значений поля Mask регистра PageMask

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

3.7.3.6 Peructp Wired (Peructp 6 CP0, Select 0)

Регистр Wired доступен для чтения и записи. Этот регистр определяет границу между случайными и "привязанными" строками TLB, как показано на Рисунок 3.19. Ширина поля Wired определяется так же, как для описанного выше регистра Index. «Привязанные» строки зафиксированы, то есть они не являются удаляемыми и не могут быть перезаписаны командой TLBWI

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



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

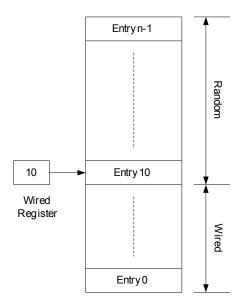


Рисунок 3.19. «Привязанные» и случайные строки TLB

Формат регистра Wired

31	4	3	0
0		Wired	

Таблица 3.22. Описание полей регистра Wired

Поля		Onuconno	Чтение/	Начальное
Имя	Биты	Описание	запись	состояние
0	31:4	При чтении возвращается нуль	0	0
Wired	3:0	Граница между «привязанными» и	R/W	0
		случайными строками TLB.		

3.7.3.7 Peructp BadVAddr (Peructp 8 CP0, Select 0)

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

- ошибка адреса (AdEL или AdES);
- TLB Refill;
- TLB Invalid;
- TLB Modified.

Формат регистра BadVAddr

31	0
BadVAddr	



Таблица 3.23. Описание полей регистра BadVAddr

Поля		0	Чтение/	Начальное
Имя	Биты	Описание	запись	состояние
BadVAddr	31:0	Виртуальный адрес, вызвавший	R	Не определено
		исключение		

3.7.3.8 Peгистр Count (Peгистр 9 CP0, Select 0)

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

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

Формат регистра Count

31	0
Count	

Таблица 3.24. Описание полей регистра Count

Поля		0	Чтение/	Начальное
Имя	Биты	Описание	запись	состояние
Count	31:0	Счетчик	R/W	Не определено

3.7.3.9 Peructp EntryHi (Peructp 10 CP0, Select 0)

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

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

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

Формат регистра EntryHi

31			0
VPN2	0	ASID	



Таблица 3.25. Описание полей регистра EntryHi

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

3.7.3.10 Peructp Compare (Peructp 11 CP0, Select 0)

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

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

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

Формат регистра Compare

31	0	
Compare		

Таблица 3.26. Описание полей регистра Compare

Поля		Ownsouns	Чтение/	Начальное
Имя	Биты	Описание	запись	состояние
Compare	31:0	Период счета таймера	R/W	Не определено



3.7.3.11 Peructp Status (Peructp 12 CP0, Select 0)

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

Разрешение прерываний: прерывания разрешаются, когда истинны все следующие условия:

- IE = 1;
- EXL = 0;
- ERL = 0.

Если эти условия выполнены, прерывания разрешаются установкой битов IM.

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

- режим User: UM = 1, EXL = 0, and ERL = 0;
- режим Kernel: UM = 0 или EXL = 1 или ERL = 1.

Формат Status регистра

31	28	27	26	23	22	21	20	19	18	16 15	8 7	5	4	3	2	1	0
CU3	-CU0	0	0		BEV	TS	0	NMI	0	IM7-IN	M0 ()	UM	0	ERL	EXL	IE

Таблица 3.27. Описание полей регистра Status

Поля		Описание	Чтение/	Начальное состояние	
Имя	Биты	Описание	запись	Пачальное состояние	
CU3-CU0	31:28	Не используются	R/W	Не определено	
-	27	Не используется	0	0	
-	26:23	При чтении возвращается нуль	0	0	
		Управление размещением векторов			
BEV	22	исключения:	R/W	1	
		0: Нормальный			
		1: Начальная загрузка			
		TLВ-закрытие системы. Этот бит			
		устанавливается, если при выполнении			
TS	21	команд TLBWI или TLBWR образуется	R/W	0	
		команда, которая приводит к условию			
		закрытия, если оно разрешено.			
		Программа может записывать в этот			
		разряд только 0, чтобы очистить его, и			
		не может вызвать переход этого бита из			
		0 в 1.			



Пол		Описание	Чтение/	Начальное состояние
Имя	Биты	Указывает, что вход в вектор	запись	
NMI	19	исключения начальной установки был осуществлен по причине возникновения NMI. 0: Не NMI (Аппаратный сброс) 1: NMI Программное обеспечение может записывать в этот бит только 0, чтобы	R/W	1 для NMI, иначе 0
	18:16	очистить его, и не может записать 1.	0	0
IM[7:0]	15:8	При чтении возвращается нуль Маска прерываний: управление разрешением внешних, внутренних и программных прерываний. Прерывание принимается в случае, если установлен бит IE регистра Status и установлены соответствующие биты как в поле IM[7:0] регистра Status, так и в поле IP[7:0] регистра Cause. 0: Запрос на прерывание не разрешен. 1: Запрос на прерывание разрешен.	R/W	Не определено
-	7:5	При чтении возвращается нуль	0	0
UM	4	Указывает на то, что процессор работает в непривилегированном режиме (User): 0: Процессор работает в привилегированном режиме (Kernel) 1: Процессор работает в непривилегированном режиме (User) Замечание: процессор может также находиться в режиме Kernel, если установлены биты EXL или ERL. Это условие не влияет на состояние бита UM.	R/W	Не определено
-	3	При чтении возвращается нуль	0	0
ERL	2	Уровень ошибки. Устанавливается процессором при возникновении исключений Reset и NMI. 0: Нормальный уровень 1: Уровень ошибки Когда бит ERL установлен: Процессор находится в режиме Kernel. Прерывания запрещены. Команда ERET использует адрес возврата, содержащийся в ErrorEPC вместо EPC. киѕед используется как неотображаемая и некэшируемая область. Это позволяет иметь доступ к главной памяти при ошибках кэш. Поведение процессора не определено, если бит ERL установлен при выполнении кода из useg/kuseg.	R/W	1



Пол	я	Описание	Чтение/	Начальное состояние
Имя	Биты	Officating	запись	11a 1a.1biioc coctonine
EXL	1	Уровень Исключения. Устанавливается процессором при возникновении любого исключения, кроме Reset и NMI. 0: Нормальный уровень 1: Уровень исключения Когда бит EXL установлен: Процессор переходит в привилегированный режим (Kernel). Прерывания запрещены. Исключения TLB Refill используют общий вектор исключения вместо вектора TLB Refill. Если происходит другое исключение, EPC не модифицируется.	R/W	Не определено
IE	0	Разрешение Прерывания. 0: Отключает прерывания 1: Разрешает прерываниям	R/W	Не определено

3.7.3.12 Peructp Cause (Peructp 13 CP0, Select 0)

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

Формат регистра Cause

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



Таблица 3.28. Описание полей регистра Cause

Пол Имя	ія Биты	Описание	Чтение/ запись	Начальное состояние
BD	31	Указывает на то, что последнее исключение произошло в слоте задержки перехода: 0: Не в слоте задержки 1: В слоте задержки Замечание: бит ВD не модифицируется на новом исключении, если установлен бит EXL.	R	Не определено
0	30:24	При чтении возвращается нуль	0	0
IV	23	Указывает, какой вектор используется для обслуживания исключений прерывания — общий или специальный вектор прерываний: 0: Используется общий вектор исключения (0х180) 1: Используется специальный вектор прерываний (0х200)	R/W	Не определено
0	22:16	При чтении возвращается нуль	0	0
IP[7:2]	15:10	Указывает, какое прерывание установлено: 15 — СОМРАКЕ; 14 - прерывания регистра QSTR4, объединенные по ИЛИ; 13 - прерывания регистра QSTR3, объединенные по ИЛИ; 12 - прерывания регистра QSTR2, объединенные по ИЛИ; 11 - прерывания регистра QSTR1, объединенные по ИЛИ; 11 - прерывания регистра QSTR1, объединенные по ИЛИ; 10 - прерывания регистра QSTR0, объединенные по ИЛИ	R	Не определено
IP[1:0]	9:8	Управляет запросами программных прерываний (посредством записи «1» в данные разряды): 9: Запрос программного прерывания 1; 8: Запрос программного прерывания 0	R/W	Не определено
ID	7	Прерывание от встроенных средств отладки программ (OnCD).	R/W	0
Exc Code	6:2	Код исключения — см. Таблица 3.29		
0	1:0	При чтении возвращается нуль	0	0

Таблица 3.29. Описание поля Exc Code регистра Cause

Значение Exc Code Мнемоника		Описание			
0	Int	Прерывание			
1	Mod	TLB-исключение модификации			
2	TLBL	TLB-исключение (загрузка или вызов команды)			
3	TLBS	TLB-исключение (сохранение)			



Значение Exc Code	Мнемоника	Описание
4	AdEL	Прерывание по ошибке адресации (загрузка или вызов команды)
5	AdES	Прерывание по ошибке адресации (сохранение)
6-7	-	Не используются
8	Sys	Системное исключение
9	Вр	Исключение Breakpoint
10	RI	Исключение зарезервированной команды
11	CpU	Исключение недоступности сопроцессора
12	Ov	Исключение целочисленного переполнения
13	Tr	Исключение Тгар
14	-	Не используются
15	FPE	Исключение от сопроцессора арифметики в формате с плавающей точкой (FPU)
16-23	-	Не используются
24	MCheck	Аппаратный контроль
25-31	-	Не используются

3.7.3.13 Регистр ЕРС (Регистр 14 СР0, Select 0)

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

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

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

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

Формат регистра ЕРС

31	0
EPC	

Таблица 3.30. Описание полей регистра ЕРС

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



3.7.3.14 Peructp PRId (Peructp 15 CP0, Select 0)

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

Формат регистра PRId

31	24 23	16 15	8 7	0
R	Company ID	Processor ID	Revision	

Таблица 3.31. Описание полей регистра PRId

Поля		Описание	Чтение/	Начальное
Имя	Биты	Описание	запись	состояние
R		При чтении возвращается нуль	R	0
Company ID	23:16	Идентификация компании, которая проектировала или изготовляла процессор.	R	0x0A
Processor ID	15:8	Идентификация типа процессора.	R	0x20
Revision	7:0	Номер версии процессора. Позволяет программам различать разные версии одного типа процессора.	R	0x01

3.7.3.15 Peructp EBase (Peructp 15 CP0, Select 1)

Регистр EBase содержит номер процессора в многопроцессорной системе. Это 32-разрядный регистр, доступный только для чтения.

Формат регистра EBase

31	30	10 9	0
1	0	Num	

Таблица 3.32. Описание полей регистра EBase

Поля		Описание	Чтение/	Начальное
Имя	Биты		запись	состояние
1	31	При чтении возвращается единица	R	1
-	30:10	Не используется	R	0x0
Num	9:0	Номер процессора. Позволяет программам различать разные процессоры в многопроцессорной системе	R	CPU0 – 0x000, CPU1 – 0x001



3.7.3.16 Регистр Config (Регистр 16 CP0, Select 0)

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

Формат регистра Config

31	30	28	27	25	24	21	20	19	18	17	16	15	14	13	12	10	9	7	6	3	2 0	
M	K23	5	KU		0		MDU	R	MN	1	BM	BE	AT		AR		MT		0		K0	

Таблица 3.33. Описание полей регистра Config

По Имя	оля Биты	Описание	Чтение/	Начальное
М	31	Этот бит аппаратно устанавливается в высокий уровень, указывая на наличие регистра Config1	запись R	1
K23	30:28	Это поле управляет кэшируемостью адресных сегментов kseg2 и kseg3 в режиме FM. В режиме TLB не используется. См. табл.2.33	FM:R/ W TLB:R	FM:010 TLB:000
KU	27:25	Это поле управляет кэшируемостью адресных сегментов kuseg и useg в режиме FM. В режиме TLB не используется. См. табл.2.33	FM:R/ W TLB:R	FM:010 TLB:000
0	24:21	Не используются	0	0
MDU	20	Тип MDU: итеративный умножитель и делитель	R	1
R	19	При чтении возвращается нуль	0	0
MM	18:17	Режим No Merging для 32 bit collapsing write buffer	R	0
BM	16	Тип передачи Burst: последовательный	R	0
BE	15	Режим endian: Little endian	R	0
AT	14:13	Тип архитектуры, реализованной процессором: MIPS32	R	0
AR	12:10	Номер версии: 1	R	0
MT	9:7	Тип MMU: 1: Стандартный TLB (FM = 0) 3: Фиксированное отображение (FM = 1) 0, 2, 4-7: зарезервированы	R	TLB: 01 FM: 11
R	6:3	При чтении возвращается нуль	0	0
K0	2:0	Алгоритм когерентности для кseg0, см. Таблица 3.18.	R/W	010



Таблица 3.34. Атрибуты когерентности кэш

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

3.7.3.17 Peructp Config1 (Peructp 16 CP0, Select 1)

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

Формат регистра Config1

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

Таблица 3.35. Описание полей Config1 регистра

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

3.7.3.18 Perистр LLAddr – Load Linked Address (Регистр 17 CP0, Select 0)

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

Формат LLAddr регистра

31	28 27	0
0	Paddr[31:4]	



Таблица 3.36. Описание полей LLAddr регистра

Поля	I	Ownsowns	Чтение/	Начальное
Имя	Биты	Описание	запись	состояние
0	31:28	При чтении возвращается нуль	0	0
Paddr[31:4]	27:0	Физический адрес последней	R	Не определено
		команды LL		_

3.7.3.19 Регистр ErrorEPC (Регистр 30 CP0, Select 0)

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

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

Этот адрес может быть:

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

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

Формат регистра ErrorEPC

31	0
ErrorEPC	

Таблица 3.37. Описание полей регистра ErrorEPC

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

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



3.8 Кэш

CPU имеет кэш команд и кэш данных типа direct mapped объемом по 32 Кбайт. Кэш данных работает по протоколу write-through.

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

На Рисунок 3.20 представлен формат каждой строки массивов тэгов и данных. Тэговая строка содержит 20 старших бита физического адреса (биты [31:12]) и бит валидности.

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

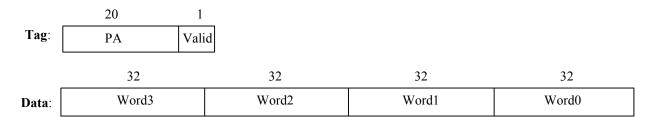


Рисунок 3.20. Формат массива кэш

Кэш имеет только два атрибута кэшируемости. Область может быть либо кэшируемой, либо некэшируемой (см. Таблица 3.34).



4. ИНТЕРВАЛЬНЫЙ ТАЙМЕР

4.1 Назначение

Интервальный таймер (IT) предназначен для выработки периодических прерываний на основе деления тактовой частоты CPU либо внешней тактовой частоты XTI

. Основные характеристики таймера:

- число разрядов делителя 32;
- число разрядов предделителя 8;
- программное управление стартом и остановкой таймера;
- доступ ко всем регистрам обеспечивается в любой момент времени.

В микросхеме 1892ВК016 имеется два интервальных таймера IRT0, IRT1.

4.2 Структурная схема IT

Структурная схема IT представлена на Рисунок 4.1.

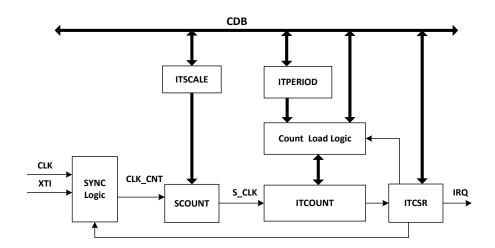


Рисунок 4.1. Структурная схема IT

В состав таймера входят следующие основные узлы:

- ITCSR регистр управления и состояния;
- ITCOUNT счетчик основного делителя;
- ITPERIOD регистр периода основного делителя;
- ITSCALE регистр предделителя;
- SCOUNT счетчик предделителя;
- SYNC Logic логика синхронизации частот;
- Count Load Logic логика загрузки счетчика основного делителя.

На структурной схеме интервального таймера использованы следующие обозначения:



- CDB шина данных CPU;
- CLK тактовая частота работы CPU;
- XTI внешняя тактовая частота;
- RTCXTI внешняя тактовая частота;
- CLK CNT выходная частота логики синхронизации;
- S CLK выходная частота предделителя;
- IRQ запрос на прерывание от таймера реального времени.

На вход интервального таймера поступает тактовая частота CPU и внешняя тактовая частота XTI. Для правильной работы таймера должно выполняться соотношение: $f_{XTI} \le \frac{f_{CLK}}{4}$, где f_{XTI} и f_{CLK} значения частот XTI и CLK соответственно.

4.3 Описание регистров интервального таймера

В Таблица 4.1 приведен перечень программно-доступных регистров IT.

Таблица 4.1. Перечень регистров IT

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

Формат регистра ITCSR приведен в Таблица 4.2.

Таблица 4.2. Формат регистра ITCSR

Номер разряда	Условное обозначение	Описание	
0	EN	Разрешение работы таймера: 0 — запрещение работы (неактивное состояние таймера); 1 — разрешение работы (активное состояние таймера)	
1	INT	Признак срабатывания таймера. Состояние данного разряда транслируется в биты IRT0 или IRT1 регистра QSTR0. Сбрасывается при записи нуля в этот разряд	
2	TICK	Бит тестирования регистра счетчика ITCOUNT и регистра предделителя IRTSCALE. При записи 1 в бит TICK декрементируется значение счетчика IRTCOUNT и предделителя ITSCALE. Поле доступно только по записи	



Номер разряда	Условное обозначение	Описание
4:3	CLK_SEL	Задает тактовую частоту от которой работает интервальный таймер: 00 – CLK – тактовая частота CPU; 01 – XTI – внешняя тактовая частота; 10 – не используется

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

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

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

4.4 Программирование IT

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

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

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

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

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



5. КОНТРОЛЛЕР ПРЯМОГО ДОСТУПА В ПАМЯТЬ (DMA)

5.1 Перечень каналов DMA

Контроллер DMA микросхемы 1892ВК016 имеет 28 каналов. Перечень каналов приведен в Таблица 5.1.

Таблица 5.1. Перечень каналов DMA

Условное обозначение канала	Назначение канала
SpFR_CH0 – SpFRCH4	Обмен данными между SpFR и памятью микросхкмы
CW TV DAT CHO	Hamayaya yayyay ya yaygay a SWICO
SW_TX_DAT_CH0	Передача данных из памяти в SWIC0
SW_TX_DES_CH0	Передача дескрипторов из памяти в SWIC0
SW_RX_DAT_CH0	Передача данных из SWIC0 в память
SW_RX_DES_CH0	Передача дескрипторов из SWIC0 в память
SW_TX_DAT_CH1	Передача данных из памяти в SWIC1
SW_TX_DES_CH1	Передача дескрипторов из памяти в SWIC1
SW_RX_DAT_CH1	Передача данных из SWIC1 в память
SW_RX_DES_CH1	Передача дескрипторов из SWIC1 в память
MEM_CH00 -	Обмен данными типа память-память
MEM_CH07	

Памятью могут быть CRAM0, CRAM1 и внешняя память DDR, доступная через DDR PORT.

Если при работе DMA изменяется программный код в памяти, то когерентность кэш CPU0, CPU1 аппаратно не обеспечивается. В этом случае для обеспечения когерентности используется бит FLUSH в системном регистре CSR.

Каналы DMA коммутатора SpFR описаны в разделе 11.

5.2 Организация обмена данными в микросхеме

Для передачи данных в микросхеме 1892ВК016 имеются: шина CDB (CPU Data Bus) и коммутатор AXI Switch (см. Рисунок 1.1).

CPU без конфликтов с DMA обменивается данными с памятью CRAM, с системными регистрами (CSR, MASKR, QSTR и т.д.), а также с регистрами устройств IRT0, IRT1, DMA, DDR_PORT, MFBSP, UART, SWIC, SpFR.

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



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

5.3 Каналы DMA типа память-память

В микросхеме 1892ВК016 8-канальный DMA МЕМ_СН, который обеспечивает обмен данными между двумя областями любых блоков памяти (внутренней или внешней).

Для управления работой каждого канала МЕМ_СН имеются следующие регистры:

- регистр управления и состояния CSR MEM CH;
- регистры индекса (физический адрес памяти) IR0, IR1;
- регистры смещения OR, Y;
- регистр начального физического адреса блока параметров DMA передачи для выполнения процедуры самоинициализации (СР);
- псевдорегистр управления состоянием бита RUN регистра CSR (RUN_MEM_CH).

Исходное состояние регистров CSR_MEM_CH: разряды 15:0 – нули, а состояние разрядов 31:16 не определено. Исходное состояние остальных регистров не определено.

Формат регистров CSR_MEM_CH этих каналов приведен в Таблица 5.2.

Таблица 5.2. Формат регистра управления и состояния каналов МЕМ СН

Номер разряда	Условное обозначение	Назначение
0	RUN	Состояние работы канала DMA:
		0 – состояние останова;
		1 – состояние обмена данными.
		Устанавливается в 1 при записи 1 в этот
		разряд.
		Устанавливается в 0:
		при записи 0 в этот разряд и после
		окончания передачи данных, оставшихся в
		канале;
		при завершении передачи блока данных.
		Состояние этого бита определяется в
		процессе выполнения процедуры
		самоинициализации
1	DIR	Направление обмена данными:
		0 – память по IR $0 = $ память по IR 1 ;
		1 – память по IR1 => память по IR0.
5:2	WN	Пакет данных, который передается по
		коммутатору AXI Switch за одно
		предоставление прямого доступа:
		0 – 1 слово;
		F – 16 слов



Номер разряда	Условное обозначение	Назначение
6	EN64	Формат передаваемых данных по коммутатору AXI Switch: 0 — 32 разряда; 1 — 64 разряда. При передаче 32-разрядными словами: WCX — число 32-разрядных слов; адрес в IR0, IR1 должен быть выровнен по границе 32-разрядного слова. При передаче 64-разрядными словами: WCX — число 64-разрядных слов; адрес в IR0, IR1 должен быть выровнен по границе 64-разрядного слова
7:8	-	Не используется
9	2D	Режим модификации адреса регистра IR1: 0 – одномерный режим; 1 – двухмерный режим.
10:11	-	Не используется
12	CHEN	Разрешение выполнения очередной процедуры самоинициализации: 0 — выполнение очередной процедуры самоинициализации запрещено; 1 — выполнение очередной процедуры самоинициализации разрешено. Используется только при обмене цепочкой блоков данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
13	IM	Маска разрешение установки признака END: 0 — установки признака запрещено; 1 — установки признака разрешено. Используется только при обмене цепочкой блоков данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
14	END	Признак завершения передачи блока данных. Устанавливается в 1 при завершении передачи блока данных (при IM=1). Устанавливается в 0 при чтении содержимого этого регистра. Доступен по записи и чтению
15	DONE	Признак завершения передачи блока данных. Устанавливается в 1 при завершении передачи блока данных при СНЕN=0 (СНЕN=1 может быть только при использовании процедуры самоинициализации). Устанавливается в 0 при чтении содержимого этого регистра



Номер разряда	Условное обозначение	Назначение
31:16	WCX	Число слов данных, которые должен передать канал DMA при одномерной адресации (блок данных). Число слов в строке при двухмерной адресации. Количество передаваемых слов = WCX + 1. Содержимое этого поля уменьшается на длину пакета данных, переданного каналом DMA

Все разряды регистра CSR MEM СН доступны по записи и чтению.

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

32-разрядные регистры индекса IR0, IR1 содержат начальные физические адреса источника и приемника данных (или, наоборот, в зависимости от содержимого разряда DIR регистра CSR_MEM_CH) памяти микросхемы. В зависимости от содержимого разряда EN64 адреса в этих регистрах должны быть выровнены по границе 32 или 64-разрядного слова.

Формат регистра смещения OR приведен в Таблица 5.3.

Таблица 5.3. Формат регистра индекса и смещения каналов МЕМ СН

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

При модификации индексного регистра смещение, задаваемое полями OR0, OR1, рассматривается как число со знаком в диапазоне от –32768 до +32767 слов данных (32 или 64-разрядных). Алгоритм модификации адреса:

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



```
модификация адреса для 32-х разрядного обмена: IR1 = IR1 + \{\{14\{OR1[15]\}\},OR1,00\};
```

Канал MEM_CH обеспечивают передачу двумерных массивов (матриц W[m;n]). При этом, память (внутренняя или внешняя) адресуется в двухмерном режиме. Для этого имеется 32-разрядный регистр Y, формат которого приведен в Таблица 5.4.

Таблица 5.4. Формат регистра У

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

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

Содержимое счетчика WCX сохраняется в буферном регистре;

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

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

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



```
for (y = 0; y \le WCY; y++) {
                                                    for ( x = 0; x < WCX;
x++ ) { пересылка по адресу IR1
                           для 64-x разрядного обмена : IR1 = IR1 +
{{13{OR1[15]}},OR1,3'h0};
                           для 32-x разрядного обмена : IR1 = IR1 +
{{14{OR1[15]}},OR1,2'h0}
                          };
                                                    пересылка по адресу
IR1
                                                   для 64-х разрядного
обмена : IR1 = IR1 + \{\{13\{ORY[15]\}\},ORY,3'h0\};
                                                   для 32-х разрядного
обмена : IR1 = IR1 + \{\{14\{ORY[15]\}\},ORY,2'h0\};
                                                   };
//общее кол-во пересылок (WCX=1) * (WCY+1)
```

5.4 Каналы DMA SWIC

Для обслуживания портов SWIC имеются следующие каналы DMA:

```
SWIC_TX_DES_CH0, SWIC_TX_DAT_CH0, SWIC_RX_DES_CH0, SWIC_RX_DAT_CH0, SWIC_TX_DES_CH1, SWIC_TX_DAT_CH1, SWIC_RX_DES_CH1, SWIC_RX_DAT_CH1.
```

Для управления работой каналы DMA портов содержат следующие регистры:

- регистр управления и состояния (CSR);
- регистр индекса (физический адрес памяти) (IR);
- регистр начального адреса блока параметров DMA передачи для 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.

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

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

Памятью могут быть CRAM, внешняя память, доступная через DDR PORT.



Формат регистров управления и состояния CSR каналов DMA этих портов приведен в Таблица 5.5.

Таблица 5.5. Формат регистров управления и состояния DMA портов

Номер разряда	Условное обозначение	Назначение
0	RUN	Состояние работы канала DMA: 0 – состояние останова; 1 – состояние обмена данными. Устанавливается в 1 при записи 1 в этот разряд. Устанавливается в 0: при завершении передачи блока данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
5:2	- WN	Не используется Число слов данных (пачка), которое передается за одно предоставление прямого доступа: 0 – 1 слово, F – 16 слов. Посредством этого параметра можно плавно изменять приоритет каналов DMA относительно других устройств и относительно друг друга
11:6	-	Не используется
12	CHEN	Разрешение выполнения очередной процедуры самоинициализации: 0 — выполнение очередной процедуры самоинициализации запрещено; 1 — выполнение очередной процедуры самоинициализации разрешено. Используется только при обмене цепочкой блоков данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
13	IM	Маска разрешение установки признака END: 0 — установки признака запрещено; 1 — установки признака разрешено. Используется только при обмене цепочкой блоков данных. Состояние этого бита определяется в процессе выполнения процедуры самоинициализации
14	END	Признак завершения передачи блока данных. Устанавливается в 1 при завершении передачи блока данных (при IM=1). Устанавливается в 0 при чтении содержимого этого регистра.



Номер разряда	Условное обозначение	Назначение
15	DONE	Признак завершения передачи блока данных. Устанавливается в 1 при завершении передачи блока данных при СНЕN=0 (СНЕN=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

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.



5.5 Процедура самоинициализации

Все каналы DMA могут выполнять процедуру самоинициализации (выполнение цепочки передач DMA).

Для выполнения самоинициализации в каналах DMA имеется 32-разрядный регистр СР, в котором хранится физический начальный адрес блока параметров очередного DMA обмена. Младшие три разряда регистра СР игнорируются (адреса выровнены по границе 64-разрядного слова). Младший (нулевой разряд) регистра СР 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 СР записать адрес первого блока параметров DMA передачи. При этом 0 разряд записываемых данных должен содержать 1 (признак пуска самоинициализации). В результате этого, соответствующий канал загрузит в свои регистры параметры DMA передачи и начнет обмен данными.

После окончания передачи блока данных бит END в регистре CSR устанавливается в единичное состояние, если бит IM = 1 - выдается прерывание. По окончании передачи блока данных также проверяется состояние бита CHEN. Если он равен 1, то будет загружен следующий блок параметров DMA передачи и т.д. В противном случае цепочка DMA обменов закончится и в регистре CSR бит DONE установится в единичное состояние и выдается прерывание.

Параметры для самоинициализации каналов DMA MEM_CH размещаются в памяти в трех последовательных 64-разрядных словах, следующим образом (в порядке возрастания адресов):

```
63______0
{ IR1<sub>32</sub>, IR0<sub>32</sub> };
{{WCY<sub>16</sub>,ORY<sub>16</sub>},{OR1<sub>16</sub>,OR0<sub>16</sub> }};
{ CSR<sub>32</sub>, CP<sub>32</sub> }.
```

Параметры для самоинициализации каналов DMA портов размещаются в памяти в двух последовательных 64-разрядных словах, следующим образом (в порядке возрастания адресов):



При необходимости каналы DMA могут инициализироваться программно. Для этого CPU должен 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.

5.6 Прерывания DMA

Канал DMA формирует прерывание в соответствующем регистре QSTR (при условии, если установлены соответствующие биты в регистре MASKR и в поле IM[12:10] регистра STATUS CPU) при единичном состоянии битов DONE или END.

Обнуление битов DONE и END (и снятие соответствующего прерывания) выполняется посредством чтения содержимого регистра CSR или записью в эти биты нулей.



6. ПОРТ ВНЕШНЕЙ ПАМЯТИ ТИПА DDR SDRAM (DDR_PORT)

6.1 Общие положения

Внешний интерфейс порта обеспечивает подключение памяти типа DDR SDRAM, соответствующей стандарту JESD79C, с параметрами:

- организация x8, x16, x32;
- количество банков 4;
- разрядность адреса строки не более13;
- разрядность адреса столбца 8, 9, 10, 11, 12;
- задержка данных при чтении 2, 2.5 и 3 такта.

Контроллер имеет следующие основные характеристики:

- шина данных 32 разряда;
- шина адреса 13 разрядов;
- шина адреса банка— 2 разряда;
- количество стробов DQS-5;
- количество стробов DM-5;
- количество каналов выдачи частоты синхронизации

$$CK-2$$
; $CKn-2$;

- программируемая установка параметров памяти;
- программная и аппаратная подстройка «окна» приема данных;
- аппаратный контроль открытия страниц;
- защита памяти модифицированным кодом Хэмминга.

6.2 Регистры DDR PORT

Перечень регистров приведен в Таблица 6.1.

Таблица 6.1. Регистры контроллера

Условное обозначение	Название регистра	
DDR_CON	Регистр конфигурации	
DDR_TMR	Регистр параметров	
DDR_CSR	Регистр управления и состояния	
DDR_MOD	Регистр режимов	
DDR_EXT	Регистр управления режимами контроля памяти	
DDR_ERR	Регистр ошибок памяти	



При описании полей и значений регистров используются обозначения:

- R только чтение;
- RW чтение и запись;
- RW1 Чтение, пуск операции;
- і: ј неразрывная группа разрядов, і –старший разряд группы, ј –младший;
- [i] номер разряда;
- 0х далее следует шестнадцатеричный код.

Термины и обозначения временных параметров и команд управления DDRAM соответствуют стандарту JESD79C.

6.2.1 Регистр конфигурации DDRAM

Perucтр DDR_CON предназначен для программирования конфигурационных параметров синхронной памяти типа DDR.

Формат регистра DDR CON приведен в Таблица 6.2.

Таблица 6.2. Формат регистра DDR_CON

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:30	-	Резерв	R	0
29:16	tRFR	Период авторегенерации DDRAM в тактах частоты СК	RW	0
15:13	-	Резерв	R	0
12	tWTR	Внутренняя задержка DDRAM между командами WRITE и READ в тактах частоты CK: 0 – 1 такт CK; 1 – 2 такта CK	RW	0
11:9	-	Резерв.	R	0
8	DS	Мощность выходов микросхем DDRAM, подключенных к контроллеру (Drive Strength): 0 — нормальная STTL class II; 1 — пониженная (~54% от нормальной)	RW	0
7	-	Резерв	R	0
6:4	CL	Задержка данных при чтении (CAS latency): 010 – 2 такта СК; 011 – 3 такта СК; 110 – 2.5 такта СК; 000:001, 100:101, 111 – резерв	RW	0
3	-	Резерв	R	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
2:0	PS	Размер страницы микросхем DDRAM,		
		подключенных к контроллеру:		
		000 - 512;		
		001 - 1024;		
		010 - 2048;	RW	0
		011 – 4096;		
		100 - 256;		
		101: 111 – резерв.		
		Число банков DDRAM – 4		

Преобразование физического адреса в адрес 32 - разрядной памяти DDRAM при различных значениях параметра PS представлено в Таблица 6.3 - Таблица 6.5. Разряды физического адреса в таблицах обозначены строчными буквами «а».

Таблица 6.3. Отображение адреса строки

PS		Адрес DDRAM											
18	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
100	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12
000	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13
001	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14
010	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15
011	a28	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16

Таблица 6.4. Отображение адреса столбца

PS		Адрес DDRAM											
10	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
100	0	0	0	0	0	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2
000	0	0	0	0	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2
001	0	0	0	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2
010	a13	a12	0	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2
011	a13	a12	0	a11	a 10	a 9	a 8	a 7	a 6	a 5	a 4	a 3	a 2

Таблица 6.5. Отображение адреса банка

PS	Адрес банка DDRAM				
rs	BA1	BA0			
100	a11	a10			
000	a12	a11			
001	a13	a12			
010	a14	a13			
011	a15	a14			

Период авторегенерации должен определяться индивидуально для используемой конфигурации DDRAM. Например, при тактовой частоте CK 200 М Γ ц для обеспечения 8 192 цикловой регенерации за 64 мс необходимо в поле tRFR записать код 0x61A, что соответствует 7,81 мкс на строку. При tRFR = 0 режим авторегенерации отключен.



6.2.2 Peructp параметров DDR_TMR

Регистр DDR_TMR предназначен для задания интервалов (в тактах частоты CK) между различными командами DDR.

Формат регистра DDR TMR приведен в Таблица 6.6.

Таблица 6.6. Формат регистра DDR_TMR

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:27	-	Резерв.	R	0
27:24	tRC	Минимальный интервал между командами ACTIVE	RW	0
23:20	tRFC	Минимальный интервал между командами AUTO REFRESH.	RW	0
19:16	tRAS	Минимальная задержка между командами ACTIVE и PRECHARGE	RW	0
15:14	-	Резерв	R	0
13:12	tRTW	Дополнительная задержка команды READ после WRITE	RW	0
11:10	-	Резерв.	R	0
9:8	tRCD	Минимальная задержка между командами ACTIVE и READ / WRITE	RW	0
7:6	-	Резерв	R	0
5:4	tRP	Минимальный период команд PRECHARGE	RW	0
3:2	-	Резерв	R	0
1:0	tWR	Минимальная задержка между записью данных и командой PRECHARGE (Write recovery)	RW	0

Значения 0, 1,..., n параметра в таблице соответствуют интервалу в 1, 2,..., n+1 тактов. Например, значение 0xF параметра tRFC задает интервал 16 тактов между командами AUTO REFRESH, а значение 0 – интервал в один такт.

При вычислении параметров в соответствии с рабочей частотой и со спецификацией используемой памяти, полученные значения необходимо округлять до ближайшего меньшего целого. Например, если в спецификации указано время tRCD = 20 ns, то при частоте CK 133 МГц (период 7.5ns) минимальный интервал в 2.7 такта нужно округлить до 2 и в поле tRCD регистра dRCD регистра dRCD записать код dRCD ох2.

6.2.3 Регистр состояний и управления SDRCSR

Perucтр DDR_CSR предназначен для запуска команд изменения режимов DDRAM или контроллера и индикации их исполнения.

Формат регистра DDR_CSR приведен в Таблица 6.7.



Таблица 6.7. Формат регистра DDR_CSR

Номер	Условное обозначение	Назначение	Доступ	Исходное состояние
разряда 31:9	-	Резерв.	R	О
8	_	Не используется	R	0
7	_	Резерв.	R	0
6	APPLY	При записи 1 в данный разряд контроллер	RW1	0
		выполняет перепись содержимого регистров DDR_TMR, DDR_CON, DDR_MOD, DDR_EXT в одноименные исполнительные регистры		
5	EYEW	Запись 1 в данный разряд запускает команду контроллера "подстройка частоты приема данных". При чтении - признак окончания команды "подстройка частоты приема данных": Устанавливается в 1 после завершения команды; сбрасывается при записи любой команды. Данная команда также запускается автоматически при выполнении команд инициализации и выхода из режима саморегенерации при сброшенном бите TMODE регистра DDR_MOD. При этом бит EYEW не устанавливается, а биты INIT и EXIT	RW1	0
4	EXIT	устанавливается только после завершения подстройки частоты Запись 1 в данный разряд запускает команду выхода DDRAM из режимов саморегенерации и пониженного потребления.	RW1	0
		При чтении - признак выполнения команды выхода DDRAM из режимов саморегенерации и пониженного потребления: устанавливается в 1 после завершения команды; сбрасывается при записи любой команды		
3	PWDN	Запись 1 в данный разряд запускает команду перевода DDRAM в режим пониженного потребления. При чтении - признак окончания данной команды: устанавливается в 1 после завершения команды; сбрасывается записью команды EXIT	RW1	0
2	SREF	Запись 1 в данный разряд запускает команду перевода DDRAM в режим саморегенерации. При чтении - признак окончания данной команды: устанавливается в 1 после завершения команды; сбрасывается записью команды EXIT	RW1	0
1	AREF	При записи 1 в данный разряд контроллер выполняет команду регенерации DDRAM. При чтении - признак окончания данной команды: устанавливается в 1 после завершения команды; сбрасывается при записи любой команды	RW1	0



Номер	Условное	Назначение	Доступ	Исходное
разряда	обозначение			состояние
0	INIT	Запись 1 в данный разряд запускает команду инициализации DDRAM с параметрами: Вust Length — 2; Burst Type — sequential; CAS Latency — поле CL регистра DDR_CON; Drive Strength — поле DS регистра DDR_CON. При чтении - признак окончания команды инициализации: устанавливается в 1 после завершения данной команды; сбрасывается при записи любой команды	RW1	0

Команды кодируются унитарным кодом в разрядах 8 - 0. Запись других кодов игнорируются. Выражение "Запись 1 в данный разряд" в столбце «Назначение» означает корректную запись унитарного кода с единицей в данном разряде.

При запуске любой команды изменения режимов DDR_PORT ожидает завершения текущей операции и выполняет необходимую последовательность команд DDR.

Выполнение команд INIT, EYEW и EXIT зависит от состояния поля TMODE регистра DDR_MOD.

По команде INIT после специфицированной последовательности команд инициализации,

DDR PORT дополнительно выполняет

при TMODE = 0:

- команду LOAD MODE REGISTER с битом DLL=0;
- пауза 200 тактов СК;
- команду EYEW и устанавливает индикатор INIT,
- · при TMODE = 1:
 - команду LOAD MODE REGISTER с битом DLL=0;
 - пауза tMRD тактов СК и устанавливает индикатор INIT.

До выполнения начальной инициализации необходимо записать все параметры в регистры DDR_CON, DDR_TMR и DDR_MOD и затем выполнить команду APPLY.

DDR_PORT не контролирует паузу 200 мкс между установкой стабильного питания и командой INIT.

По команде EYEW при TMODE = 0 контроллер выполняет:

- запуск аппаратуры подстройки частоты приема данных;
- циклическое чтение памяти и ожидает (~30-40 тактов) окончания подстройки;
- завершает цикл чтения и устанавливает индикатор EYEW.

По команде EYEW при TMODE = 1 контроллер выполняет запуск аппаратуры подстройки



частоты приема данных и устанавливает индикатор EYEW. В этом режиме аппаратура

аппаратура подстройки выполняет сдвиг окна приема данных для каналов, заданных полем SEL регистра DDR MOD на величину 1/34 периода частоты СК.

В режиме TMODE = 0 аппаратура подстройки автоматически центрирует окно приема данных в середину стробов DQS.

Для инициализации работы DDR_PORT и внешней памяти DDR необходимо сначала выполнить последовательно команды INIT, EYEW, EYEW. Затем в регистре DDR_MOD установить разряды TMODE=1, SEL=0x1F и выполнить последовательно команды APPLY, INIT, EYEW, EYEW.

По команде PWDN контроллер выполняет:

- PRECHARGE;
- пауза tRP, AUTO REFRESH;
- пауза 1 такт СК;
- сброс СКЕ;
- пауза tRFC, установка индикатора PWDN.

После выполнения данной команды память находится в режиме precharge power-down.

Аналогично выполняется команда SREF. Отличие в том, что сброс CKE происходит одновременно с AUTO REFRESH и устанавливается индикатор SREF.

После выполнения команд PWDN и SREF контроллер находится в состоянии останова до выполнения команды EXIT. В этом состоянии DDR_PORT не контролирует выполнение интервала tREFC.

По команде EXIT контроллер устанавливает CKE и после паузы tXSNR выполняет AREF. tXSNR = tRFC + 2.

При выходе из PWDN или из SREF при TMODE =1 команда на этом завершается установкой индикатора EXIT, а при выходе из SREF при TMODE =0, контроллер ждет 200 тактов СК, выполняет команду EYEW и устанавливает индикатор EXIT.

Контроллер игнорирует команду EXIT при сброшенных индикаторах PWDN и SREF.

По команде AREF контроллер выполняет:

- PRECHARGE;
- пауза tRP, AUTO REFRESH;
- пауза tRFC, установка индикатора AREF.



По команде APPLY контроллер выполняет перепись содержимого регистров DDR_TMR, DDR_CON, DDR_MOD, DDR_EXT в соответствующие исполнительные регистры.

6.2.4 Регистр режимов DDR MOD

Perucтр DDR_MOD предназначен для управления аппаратурой настройки окна приема данных и задания режимов выполнения специальных команд.

Формат регистра DDR_MOD приведен в Таблица 6.8.

Таблица 6.8. Формат регистра DDR_MOD

Номер разряда	Условное обозначение параметра	Назначение	Доступ	Исходное состояние
31:16	tEYE	Период автоподстройки частоты приема данных в циклах tRFR	RW	0
15	TMODE	Режим выполнения специальных команд: 0- автоматический; 1- пошаговый.	RW	0
14:5	-	Резерв.	R	0
4:0	SEL	Выбор канала приема данных в пошаговом режиме: SEL[i] =1 –сдвиг окна для строба DQS[i] разрешен, i= 0,1,2,3,4; SEL[i] =0 – сдвиг окна для строба DQS[i] запрещен, i= 0,1,2,3,4. SEL[4] соответствует каналу приема данных блока контрольных разрядов	RW	0

При TMODE = 0 контроллер аппаратно выполняет команду EYEW через каждые tRFR * tEYE тактов частоты CK.

При tEYE =0 или tRFR =0 режим автоподстройки частоты приема данных отключен.

6.2.5 Peгистр DDR EXT

Perucтр DDR_EXT предназначен для управления режимами контроля и коррекции памяти модифицированным кодом Хэмминга.

Формат регистра приведен в Таблица 6.9.



Таблица 6.9. Формат регистра DDR_EXT

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояие
31:24	Cnt_SERR	Счетчик одиночных ошибок. При значении 0xFF счетчик останавливается	WR	0
23:16	Num SERR	Допустимый порог одиночных ошибок	WR	0xFF
15:8	Cnt_DERR	Счетчик двойных ошибок. При значении 0xFF счетчик останавливается	WR	0
7:5	-	Резерв	R	0
4	-	Резерв	WR	1
3	-	Резерв	WR	0
2	NEMPTY	Признак наличия данных в FIFO ошибочных адресов. Обнуляется при записи в регистр DDR_ERR		0
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 и блок данных, и блок контрольных разрядов; 10 - режим тестирования блока контрольных разрядов. Обмен данными выполняется только с блоком контрольных разрядов; 11 - резерв	WR	0

В режиме MODE = 01 байтовая запись выполняется операцией "чтение-модификациязапись". При выполнении операции "чтение-модификация-запись" ошибки фазы чтения исправляются и фиксируются в FIFO ошибочных адресов.

В режиме MODE = 01 при Cnt_DERR > 0 или Cnt_SERR > Num_SERR формируется прерывание INT_Hm MPORT поступающее на одноименный вход регистра QSTR_Hm. Прерывание сбрасывается по следующим условиям:

- при записи Cnt_DERR = 0 и Cnt_SERR =0;
- при записи Cnt DERR = 0, если Cnt SERR ≤ Num SERR;
- при записи Cnt SERR = 0 или Num SERR = 255, если Cnt DERR = 0.

6.2.6 Peгистр DDR_ERR

Perucтр DDR_ERR предназначен для фиксации и локализации ошибок фазы чтения в режиме MODE = 01.

Регистр доступен для чтения при установленном признаке NEMPTY регистра DDR_EXT. При чтении считывается очередное слово из FIFO ошибочных адресов.

При NEMPTY = 0 состояние регистра не определено.

При записи в этот регистр любого кода признак NEMPTY устанавливается в 0.



Формат регистра приведен в Таблица 6.10.

Таблица 6.10. Формат регистра DDR_ERR

Номер разряда	Условное обозначение	Назначение
1:0	Code_ERR	Код ошибки:
		01 – одиночная ошибка;
		10 – двойная ошибка;
		11 – ошибка в контрольном разряде общей четности
31:2	ADDR_ERR	Разряды 31:2 физического адреса памяти, при чтении из которой
		обнаружена ошибка



7. УНИВЕРСАЛЬНЫЙ АСИНХРОННЫЙ ПОРТ (UART)

7.1 Общие положения

Универсальный асинхронный порт (далее UART) имеет следующие характеристики:

- по архитектуре совместим с UART 16550;
- частота приема и передачи данных от 50 до 1 Мбод;
- FIFO для приема и передачи данных имеют объем по 16 байт;
- полностью программируемые параметры последовательного интерфейса: длина символа от 5 до 8 бит; генерация и обнаружение бита четности; генерация стопового бита длиной 1, 1.5 или 2 бита;
- диагностический режим внутренней петли;
- эмуляция символьных ошибок.

Структурная схема порта UART приведена на Рисунок 7.1.

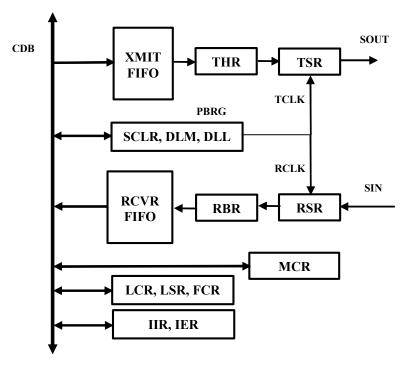


Рисунок 7.1. Структурная схема UART

Передаваемые данные записываются в регистр THR, а затем аппаратно переписываются в передающий сдвигающий регистр (TSR), если он пуст. После этого в регистр THR могут быть записаны следующие данные.

После приема данных в приемный сдвигающий регистр (RSR) данные переписываются в регистр RBR, если он не занят.



7.2 Регистры UART

7.2.1 Общие положения

Перечень регистров UART приведен в Таблица 7.1.

Таблица 7.1. Перечень регистров UART

Условное обозначение регистра	Название регистра	Смещение	Доступ (R-чтение, W-запись)
RBR	Приемный буферный регистр	0 (DLAB=0)	R
THR	Передающий буферный регистр	0 (DLAB=0)	W
IER	Регистр разрешения прерываний	1 (DLAB=0)	R/W
IIR	Регистр идентификации прерывания	2	R
FCR	Регистр управления FIFO	2	W
LCR	Регистр управления линией	3	R/W
MCR	Регистр управления модемом	4	R/W
LSR	Регистр состояния линии	5	R
SPR	Регистр Scratch Pad	7	R/W
DLL	Регистр делителя младший	0 (DLAB=1)	R/W
DLM	Регистр делителя старший	1 (DLAB=1)	R/W
SCLR	Регистр предделителя (scaler)	5	W

7.2.2 Регистр LCR

Формат регистра LCR приведен в Таблица 7.2.

Таблица 7.2. Формат регистра LCR

Номер бита	Условное обозначение	Назначение				
1:0	WLS	Количество бит данных в передаваемом символе:				
	(Word Length	00 -5 бит,				
	Select)	01 -6 бит,				
		10 -7 бит,				
		11 -8 бит.				
2	STB	Количество стоп-бит:				
	(Number Stop	0 - 1 стоп-бит,				
	Bits)	1 - 2 стоп-бита (для 5-битного символа стоп-бит имеет длину 1,5				
	,	бита).				
		Приемник анализирует только первый стоп бит.				
3	PEN	Разрешение генерации (передатчик) или проверки (приемник)				
	(Parity Enable)	контрольного бита:				
		1 – контрольный бит (паритет или постоянный) разрешен,				
		0 – запрещен.				



Номер бита	Условное обозначение	Назначение
4	EPS	Выбор типа контроля (при PEN=1):
	(Even Parity	0 – нечетность,
	Select)	1 – четность.
5	STP	Принудительное формирование бита паритета:
	(Stick Parity)	0 – контрольный бит генерируется в соответствии с паритетом
		выводимого символа,
		1 – постоянное значение контрольного бита: при EPS=1 - нулевое,
		при EPS=0 – единичное.
6	SBC	Формирование обрыва линии:
	(Set Break	0 – нормальная работа;
	Control)	1 – на выходе SOUT устанавливается низкий уровень (Spacing
		level). Это влияет только на выход SOUT, а не на логику передачи
		символа.
7	DLAB	Управление доступом к регистрам:
	(Divisor Latch	0 – разрешен доступ к регистрам RBR, THR, IER;
	Access bit)	1 – разрешен доступ к регистрам DLL, DLM

Исходное состояние регистра LCR – нули.

Бит SBC используется как признак «Внимание» для приемного терминала, подключенному к выходу UART. Для того чтобы не было передано ошибочного символа при использовании бита SBC, необходимо выполнять следующую последовательность действий:

- загрузить в регистр THR все нули по признаку THRE=1;
- установить SBC=1 по следующему THRE=1;
- дождаться ТЕМТ=1.

Для восстановления нормальной передачи необходимо установить SBC=0.

7.2.3 Регистр FCR

Формат регистра FCR приведен в Таблица 7.3.

Таблица 7.3. Формат регистра FCR

Номер бита	Условное обозначение	Назначение					
0	FEWO	Разрешение работы XMIT и RCVR FIFO:					
	(FIFO Enable)	0 – символьный режим;					
		1 – режим FIFO.					
		При изменении состояния этого бита, данные из FIFO, не					
		удаляются. Запись в биты RFR, TFR, RFTL выполняется, если					
		FEWO=1.					
1	RFR	Установка RCVR FIFO в исходное состояние.					
	(Receiver	Регистр RSR не обнуляется.					
	FIFO Reset)	После записи 1 в этот бит он автоматически сбрасывается.					
2	TFR	Установка XMIT FIFO в исходное состояние.					
	(Transmitter	Регистр TSR не обнуляется.					
	FIFO Reset)	После записи 1 в этот бит он автоматически сбрасывается.					



Номер бита	Условное обозначение	Назначение
5:3	-	Резерв
7:6	RFTL (RCVR FIFO Trigger Level)	Порог заполнения RCVR FIFO (в байтах), при котором формируется прерывание: 00 – 1; 01 – 4; 10 – 8; 11 – 14.

Исходное состояние регистра FCR – нули.

7.2.4 Регистр LSR

Формат регистра LSR приведен в Таблица 7.4.

Таблица 7.4. Формат регистра LSR

Номер бита	Условное обозначение	Назначение
0	RDR (Receiver Data	Готовность данных. Устанавливается после приема символа данных и передачи его в
	Ready)	регистр RBR или FIFO.
		Сбрасывается после чтения регистра RBR (в символьном режиме) или чтения всего содержимого RCVR FIFO (в режиме FIFO)
1	OE	Ошибка переполнения.
	(Overrun	Устанавливается, если содержимое регистра RBR не было
	Error)	прочитано, в сдвигающий регистр принят следующий символ и
		начат прием очередного символа. При этом новый символ
		записывается в сдвигающий регистр вместо старого.
		В режиме FIFO устанавливается, если после перехода порогового
		(trigger) уровня FIFO заполнено до конца, во входной сдвигающий
		регистр полностью принят следующий символ и начат прием
		очередного символа. При этом в FIFO ничего не передается. Бит сбрасывается при чтении содержимого регистра LSR.
2	PE	Ошибка контрольного бита (паритета или фиксированного).
	(Parity Error)	В режиме FIFO этот бит показывает на ошибку в символе,
		находящемся наверху FIFO.
		Бит сбрасывается при чтении содержимого регистра LSR.
3	FE	Ошибка кадра.
	(Framing	Устанавливается, если стоп-бит равен нулю (Spacing level).
	Error)	В режиме FIFO этот бит показывает на ошибку в символе,
		находящемся наверху FIFO.
		После этой ошибки UART пересинхронизируется.
		Бит сбрасывается при чтении содержимого регистра LSR



Номер бита	Условное обозначение	Назначение
4	BI	Обрыв линии.
	(Break	Устанавливается, если вход приема данных находится в состоянии
	Interrupt)	0 (Spacing level) не менее чем время передачи всего символа.
		В режиме FIFO этот бит показывает на ошибку в символе, находящемся наверху FIFO.
		При возникновении этой ситуации, в FIFO загружается только
		один нулевой символ. Прием следующих символов разрешается
		после того, как вход приема данных перейдет в единичное
		состояние (Marking state) и будет принят действительный стартовый бит.
		Бит сбрасывается при чтении содержимого регистра LSR
5	THRE (Transmitter	Передающий буферный регистр пуст. Показывает, что UART готов принять следующий символ для передачи.
	Holding Register	Устанавливается, когда содержимое регистра THR передается в передающий сдвигающий регистр. Одновременно с этим
	Empty)	генерируется прерывание THREI, если оно разрешено. Бит
		сбрасывается при записи символа в регистр THR.
		В режиме FIFO этот бит устанавливается, когда XMIT FIFO пусто, и сбрасывается, если в XMIT FIFO записывается хотя бы один
		символ
6	TEMT	Передатчик пуст.
	(Transmitter	Устанавливается, если регистры THR и TSR пусты. Имеет нулевое
	Empty)	состояние, если хотя бы один из регистров THR и TSR не пуст.
		В режиме FIFO этот бит устанавливается, если нет символов ни в XMIT FIFO, ни в регистре TSR.
7	EIRF	Наличие хотя бы одного признака ошибки в FIFO.
	(Error in	В символьном режиме этот бит всегда равен нулю.
	RCVR FIFO)	Бит сбрасывается при чтении содержимого регистра LSR, если в FIFO нет больше признаков ошибок

Исходное состояние бит THRE, TEMT – 1, остальных – 0.

Установка бит OE, PE, FE, BI приводит к формированию прерыванию по состоянию входа приема данных (Receiver Line Status Interrupt), если это прерывание разрешено.

7.2.5 Регистр IER

Формат регистра IER приведен в Таблица 7.5. Исходное состояние регистра IER – нули.

Таблица 7.5. Формат регистра IER

Номер бита	Условное обозначение	Назначение			
0	ERBI	Разрешение прерывания по наличию принятых данных (RDAI), а также по таймауту (CTI)			
1	ETBEI	Разрешение прерывания по отсутствию данных в регистре THR (THREI)			
2	ERLSI	Разрешение прерывания по статусу приема данных (RLSI)			
3	EMSI	Разрешение прерывания по статусу модема (MSI)			
7:4	-	Резерв			



7.2.6 Регистр IIR

Формат регистра IIR приведен в Таблица 7.6.

Таблица 7.6. Формат регистра IIR

Номер бита	Условное обозначение	Назначение		
0	IP	Признак наличия прерывания:		
	(Interrupt	0 – есть прерывание;		
	Pending)	1 – нет прерывания		
3:1	IID[2:0]	Код идентификации прерывания в соответствии с Таблица 7.7		
5:4	-	Резерв		
7:6	FE	Признак разрешения работы RCVR и XMIT FIFO		

Исходное состояние бита IP - 1, остальных - 0.

Таблица 7.7. Идентификация прерываний

Код поля ID[2:0]	Уровень приоритета (1 – наивысший)	Тип прерывания	Причина прерывания	Условие сброса прерывания
011	1	Статус приема данных (RLSI – Receiver Line Status Interrupt)	OE - Overrun Error; PE - Parity Error; FE - Framing Error; BI - Break Interrupt.	Чтение содержимого регистра LSR. Чтение из FIFO символа, по которому сформировано это прерывание. Обнуление FIFO.
010	2	Наличие принятых данных (RDAI – Received Data Available Interrupt)	Наличие данных в регистре RBR или достижение заданного порога FIFO	Чтение содержимого регистра RBR. Считывание данных из FIFO до уровня ниже порогового.
110	2	Таймаут (СТІ – Character Timeout Interrupt)	С момента приема последнего символа в RCVR FIFO прошло время, равное длительности передачи 4-х символов и не было ни чтения FIFO, ни приема очередного символа.	Чтение содержимого регистра RBR. Прием очередного символа. Сброс FIFO.



Код поля ID[2:0]	Уровень приоритета (1 – наивысший)	Тип прерывания	Причина прерывания	Условие сброса прерывания
001	3	Perистр THR пуст (THREI – Transmitter Holding Register Empty Interrupt)	Регистр THR пуст	Чтение содержимого регистра IIR, если источником прерывания является это условие. Запись символа в регистр THR
000	4	Статус модема (MSI – Modem Status Interrupt)	Изменение состояния сигналов на входах порта nCTS, nDSR, nRI, nDCD	Чтение содержимого регистра MSR.

7.2.7 Регистр MCR

Формат регистра MCR приведен в Таблица 7.8.

Таблица 7.8. Формат регистра MCR

Номер бита	Условное Обозначение	Назначение				
0:3	-	Не используется				
4	LOOP	Режим петли.				
		Используется для тестирования UART. При установке этого бита				
		в 1 выполняется следующее:				
		На выходе SOUT UART устанавливается высокий уровень;				
		Bxoд SIN UART отключается от внешнего вывода;				
		Выход регистра TSR подключается к входу регистра RSR;				
		В режиме петли передаваемые данные немедленно принимаются.				
		В режиме петли все прерывания формируются как обычно.				
7:5	-	Не используется				

Исходное состояние регистра MCR – нули.

7.2.8 Программируемый генератор скорости обмена

В UART имеется программируемый генератор скорости обмена данными (PBRG – Programmable Baud Rate Generator). Он состоит из 8-разрядного предделителя и 16-разрядного основного делителя частоты. На вход предделителя поступает системная тактовая частота СLK, на которой работает CPU, UART и другие устройства. Выходная частота предделителя поступает на вход основного делителя. Выходная частота генератора PBRG в 16 раз больше частоты обмена последовательными данными.



Значение частоты на выходе предделителя равно CLK/(SCLR + 1). Коэффициент деления основного делителя задается 16-разрядным регистром, который является конкатенацией регистров DLM и DLL.

Период частот передачи и приема (TCLK и RCLK) UART вычисляется по формуле:

 $CLK/(SCLR + 1) / ((конкатенация содержимого регистров DLM и DLL) *16). Минимальная величина, которая может быть записана в регистры {DLM, DLL}, равна 1.$

Исходное состояние регистров DLL, DLM, SCLR – нули.

7.3 Работа с FIFO по прерыванию

Если установлен режим работы с FIFO (EFWO=1 в регистре FCR) и разрешены прерывания по приему (бит ERI=1 в регистре IER), то в процессе приема:

- формируется прерывание, если число символов в RCVR FIFO достигло запрограммируемого порога. Это прерывание сбрасывается, если при чтении из FIFO число символов оставшихся в нем, станет меньше запрограммируемого порога;
- одновременно с этим в регистре IIR устанавливается индикатор наличия принятых данных RDAI. Индикатор обнуляется, при чтении из FIFO до снижения запрограммируемого порога;
- может возникнуть прерывание по статусу приема данных (RLSI), приоритет которого выше, чем RDA;
- бит RDR в регистре LSR устанавливается в момент передачи символа из регистра RSR в RCVR FIFO. Этот бит обнуляется при считывании из FIFO всех символов данных.

Если установлен режим работы с FIFO (EFWO=1 в регистре FCR) и разрешены прерывания по приему (ERI=1 в регистре IER), то генерируется прерывание по таймауту, если с момента приема последнего символа в RCVR FIFO прошло время, равное длительности передачи 4-х символов и за это время не было:

- ни чтения RCVR FIFO;
- ни приема в RCVR FIFO очередного символа.

При 12-битном символе и скорости передачи 300 бод, прерывание по этой причине возникнет через 160 мс.

При возникновении прерывания по таймауту оно обнуляется при считывании символа из RCVR FIFO. При этом обнуляется и таймер, генерирующий данное прерывание. Если прерывание по таймауту не возникло, то таймер таймаута обнуляется при приеме нового символа или при считывании символа из RCVR FIFO.



Если установлен режим работы с FIFO (EFWO=1 в регистре FCR) и разрешены прерывания по передаче данных (бит ETI=1 в регистре IER), то генерируется прерывание по передаче следующим образом:

- формируется прерывание THREI, если XMIT FIFO пусто. Это прерывание обнуляется, как только выполняется запись символа в регистр THR (при приеме данного прерывания в XMIT FIFO можно записать от 1 до 16 символов);
- индикатор TEMT в регистре LSR установится в единичное состояние через время равное длительности одного символа минус последний стоп бит, после установки THRE=1. Первое прерывание по передаче (если оно разрешено) формируется немедленно после установки FEWO=1.

7.4 Работа с FIFO по опросу

Если установлен режим работы с FIFO (EFWO=1 в регистре FCR) и запрещены прерывания, то обмен данными выполняется по опросу, а управление FIFO приема и передачи (RCVR, XMIT) выполняется раздельно.

В этом режиме опрос состояния RCVR и XMIT FIFO осуществляется программно, посредством считывания содержим\ого регистра LSR:

- бит RDR=1, пока есть данные в RCVR FIFO;
- биты ОЕ, РЕ, FE, BI указывают на ошибки. Эти ошибки обрабатываются так же, как и при работе по прерыванию;
- бит THRE=1, если XMIT FIFO пусто;
- бит TEMT=1, если в XMIT FIFO и TSR нет данных.

При работе по опросу нет индикации таймаута и факта достижения порога RCVR FIFO. Однако оба RCVR и XMIT FIFO могут хранить символы данных.



8. МОДУЛЬ SPINLOCK

Модуль SPINLOCK представляет собой набор из 32-х низкоуровневых взаимоисключающих примитив синхронизации. При их использовании необходимо выполнять цикл в ожидании получения блокировки.

Структурная схема модуля SPINLOCK приведена на Рисунок 8.1.

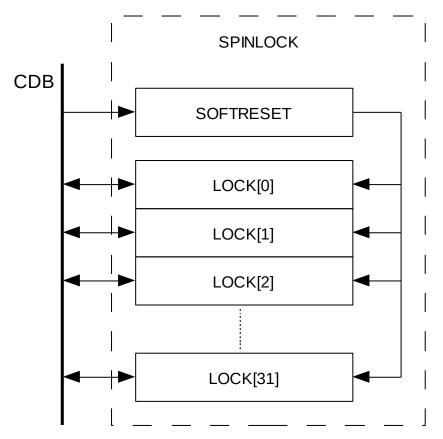


Рисунок 8.1. Структурная схема модуля SPINLOCK

Перечень регистров модуля SPINLOCK приведен в Таблица 8.1.

Таблица 8.1. Перечень регистров SPINLOCK

Условное обозначение регистра	Название регистра	Тип доступа	Исходное состояние	Смещение
SOFTRESET	Регистр программного ресета	R	0x0	0x0
LOCK[N], где N от 0 до 31	Регистр блокировки	W/R	0x0	0x800+(0x4* N)

Все регистры модуля имеют один нулевой разряд.

Алгоритм работы модуля SPINLOCK приведен на Рисунок 8.2.



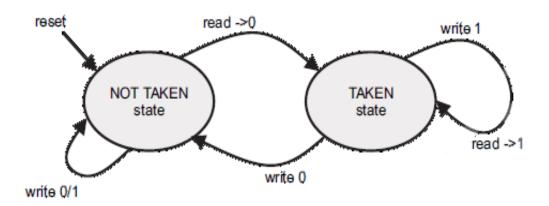


Рисунок 8.2. Алгоритм работы модуля SPINLOCK

Каждый регистр LOCK[N] имеет два состояния:

- NOT TAKEN (0);
- TAKEN (1).

При этом переход NOT TAKEN -> TAKEN происходит при чтении регистра, находящегося в состоянии NOT TAKEN. А переход TAKEN -> NOT TAKEN происходит при записи 0 в регистр, находящийся в состоянии TAKEN.

Запись «1» в регистр SOFTRESET сбрасывает все регистры LOCK[N] в состояние NOT TAKEN.



9. МОДУЛЬ ОБМЕНА СООБЩЕНИЯМИ MAILBOX

9.1 Назначение

Модуль MAILBOX предназначен для обмена данными между CPU0 и CPU1.

9.2 Характеристики

Модуль MAILBOX имеет следующие характеристики:

- 8 буферов FIFO_m (m изменяется от 0 до 7) объемом 4 32-разрядных слова;
- поддерживает режимы работы по прерыванию и по опросу;
- формирует четыре прерывания: MAILBOX_WRITE0, MAILBOX_WRITE1 прерывания при записи данных в MAILBOX, MAILBOX_READ0, MAILBOX READ1 прерывания при чтении данных в MAILBOX;
- установка и сброс прерываний осуществляется через отдельные регистры.

9.3 Регистры MAILBOX

В Таблица 9.1 приведен перечень программно-доступных регистров блока Mailbox.

Таблица 9.1. Перечень программно-доступных регистров Mailbox

Условное обозначение регистра	Название регистра	Тип доступа	Исходное состояние	Смещение
SOFT_RESET	Регистр программного сброса	W1	0x0	0x0
MESSAGE_m	Регистр обмена	W/R	0x0	m=0: 0x100 m=1: 0x104 m=2: 0x108 m=3: 0x10C m=4: 0x110 m=5: 0x114 m=6: 0x118 m=7: 0x11C
FIFO_STATUS_m	Регистр статуса FIFO	R	0x2	m=0: 0x200 m=1: 0x204 m=2: 0x208 m=3: 0x20C m=4: 0x210 m=5: 0x214 m=6: 0x218 m=7: 0x21C
SET_IRQ_WRITE0	Регистр установки прерывания IRQ_WRITE0	W1/R	0x0	0x400
SET_IRQ_WRITE1	Регистр установки прерывания IRQ_WRITE1	W1/R	0x0	0x404
SET_IRQ_READ0	Регистр установки прерывания IRQ_READ0	W1/R	0x0	0x440



Условное обозначение регистра	Название регистра	Тип доступа	Исходное состояние	Смещение
SET_IRQ_READ1	Регистр установки прерывания IRQ_READ1	W1/R	0x0	0x444
CLR_IRQ_WRITE0	Регистр сброса прерывания IRQ_WRITE0	W1/R	0x0	0x480
CLR_IRQ_WRITE1	Регистр сброса прерывания IRQ_WRITE1	W1/R	0x0	0x484
CLR_IRQ_READ0	Регистр сброса прерывания IRQ_READ0	W1/R	0x0	0x4C0
CLR_IRQ_READ1	Регистр сброса прерывания IRQ_READ1	W1/R	0x0	0x4C4

9.3.1 Peructp SOFT_RESET

Регистр установки исходного состояния модуля MAILBOX.

Формат регистра SOFT_RESET приведен в Таблица 9.2.

Таблица 9.2. Формат регистра SOFT RESET

Номер бита	Условное обозначение	Назначение
0	SOFT_RESET	В момент записи «1» приводятся в исходное состояние
		регистры FIFO_STATUS_m и сбрасывают все прерывания
31:1	-	Не используется

9.3.2 Регистр MESSAGE_m

Регистр для обмена данными с FIFO_*m*.

Формат регистра MESSAGE *т* приведен в Таблица 9.3.

Таблица 9.3. Формат регистра MESSAGE m

Номер бита	Условное обозначение	Назначение
31:0	MESSAGE	Регистр для обмена данными с FIFO_m (чтение и запись)

9.3.3 Регистр FIFO_STATUS_m

Регистр статуса FIFO_*m*.

Формат регистра FIFO_STATUS_*m* приведен в Таблица 9.4.



Таблица 9.4. Формат регистра FIFO_STATUS_m

Номер бита	Условное обозначение	Назначение
0	FULL	Регистр, указывающий на то, что FIFO полон.
1	EMPTY	Регистр, указывающий на то, что FIFO пуст.
3:2	-	Не используется
7:4	STATUS	Регистр, указывающий на число слов данных, содержащихся в FIFO
31:8	-	Не используется

9.3.4 Регистры SET_IRQ_WRITE0, SET_IRQ_WRITE1

Регистры SET_IRQ_WRITE0 и SET_IRQ_WRITE1 предназначены для формирования прерываний MAILBOX_WRITE0 и MAILBOX_WRITE1 соответственно. Эти прерывания формируются сложением всех бит соответствующего регистра по «ИЛИ». Каждому из 8 бит этих регистров соответствует буфер FIFO_m, в который необходимо записать данные.

Формат регистров SET_IRQ_WRITE0 и SET_IRQ_WRITE1 приведен в Таблица 9.5.

Таблица 9.5. Формат регистров SET IRQ WRITE0 и SET IRQ WRITE1

Номер бита	Условное обозначение	Назначение
7:0	SET_IRQ_WRIT	Каждый бит, доступен только по записи «1» (формирование
	E	прерывания).
		При чтении каждого бита: «0» - нет прерывания, «1» - есть
		прерывание
31:8	-	Не используется

9.3.5 Регистры SET_IRQ_READ0, SET_IRQ_READ1

Регистры SET_IRQ_READ0 и SET_IRQ_READ1 предназначены для формирования прерываний MAILBOX_READ0 и MAILBOX_READ1 соответственно. Эти прерывания формируются сложением всех бит соответствующего регистра по «ИЛИ». Каждому из 8 бит этих регистров соответствует буфер FIFO_m, из которого необходимо прочитать данные.

Формат регистров SET IRQ READ0 и SET IRQ READ1 приведен в Таблица 9.6.

Таблица 9.6. Формат регистров SET IRQ READ0 и SET IRQ READ1

Номер бита	Условное обозначение	Назначение
7:0	SET_IRQ_READ	Каждый бит, доступен только по записи «1» (формирование
		прерывания).
		При чтении каждого бита: «0» - нет прерывания, «1» - есть
		прерывание
31:8	-	Не используется



9.3.6 Регистры CLR_IRQ_WRITE0, CLR_IRQ_WRITE1

Peruстры CLR_IRQ_WRITE0 и CLR_IRQ_WRITE1 предназначены для сброса прерываний в регистрах SET_IRQ_WRITE0 и SET_IRQ_WRITE1 соответственно. Запись «1» в любой из 8-и бит этих регистров, обнуляет значение, установленное в соответствующем бите регистров SET_IRQ_WRITE0 и SET_IRQ_WRITE1.

Формат регистров CLR IRQ WRITE0 и CLR IRQ WRITE1 приведен в Таблица 9.7.

Таблица 9.7. Формат регистров CLR_IRQ_WRITE0 и CLR_IRQ_WRITE1

Номер бита	Условное обозначение	Назначение
7:0	CLR_IRQ_WRIT	Каждый бит, доступен только по записи «1» (сброс
	E	прерывания).
		При чтении каждого бита: «0» - нет прерывания, «1» - есть
		прерывание
31:8	-	Не используется

9.3.7 Регистры CLR_IRQ_READ0, CLR_IRQ_READ0

Регистры CLR_IRQ_READ0 и CLR_IRQ_READ1 предназначены для сброса прерываний в регистрах SET_IRQ_READ0 и SET_IRQ_READ1 соответственно. Запись «1» в любой из 8-и бит этих регистров, обнуляет значение, установленное в соответствующем бите регистров SET_IRQ_READ0 и SET_IRQ_READ1.

Формат регистров CLR IRQ READ0 и CLR IRQ READ1 приведен в Таблица 9.8.

Таблица 9.8. Формат регистров CLR IRQ 0 и CLR IRQ WRITE1

Номер бита	Условное обозначение	Назначение
7:0	CLR_IRQ_READ	Каждый бит, доступен только по записи «1» (сброс прерывания).
		При чтении каждого бита: «0» - нет прерывания, «1» - есть прерывание
31:8	-	Не используется

9.4 Методы организации межпроцессорного обмена данными

9.4.1 Обмен данными по прерыванию MAILBOX_READ

Если CPU0 имеет данные на передачу в CPU1, то CPU0 необходимо записать данные в пустой буфер FIFO_mчерез регистр MESSAGE_m и записать «1» в бит SET_IRQ_READ0[m], что приведет к запросу прерывания на чтение CPU1 из буфера FIFO m.



СРU1, получив прерывание, необходимо прочитать регистр SET_IRQ_READ0, для определения номера буфера FIFO $_m$, содержащего данные, считать данные из буфера FIFO $_m$ через регистр MESSAGE $_m$ а после этого сбросить прерывание, записав «1» в бит CLR IRQ READ0[$_m$].

9.4.2 Обмен данными по прерыванию MAILBOX_WRITE

Если СРU1 необходимо запросить данные от СРU0, то ему необходимо выбрать пустой буфер FIFO $_m$. После этого, СРU1 необходимо установить в «1» бит SET_IRQ_WRITE1[m], что вызовет запрос прерывания MAILBOX_WRITE1 на запись данных от СРU0 в буфер FIFO $_m$. После этого необходимо дождаться пока СРU0 сбросит это прерывание, и считать данные из буфера FIFO $_m$ через регистр MESSAGE $_m$.

Соответственно, CPU0, получив прерывание MAILBOX_WRITE1, необходимо прочитать регистр SET_IRQ_WRITE1, для определения номера буфера FIFO_m. Затем записать данные в буфер FIFO_m через регистр MESSAGE_m и сбросить прерывание, записав «1» в бит CLR_IRQ_WRITE1[m].

9.4.3 Обмен данными по опросу

Если CPU0 необходимо периодически передавать данные в CPU1 в выбранный буфер FIFO $_m$, то CPU0 необходимо записывать данные через регистр MESSAGE $_m$, проверяя содержимое регистра FIFO STATUS $_m$ на наличие свободного места в буфере FIFO $_m$.

Соответвенно, CPU1 должно периодически опрашивать регистр FIFO_STATUS $_m$ на предмет наличия данных в буфере FIFO $_m$, и читать их через регистр MESSAGE $_m$.



10. КОНТРОЛЛЕР ИНТЕРФЕЙСА SPACEWIRE (SWIC)

10.1 Общие положения

Контроллер интерфейса SpaceWire (далее по тексту SWIC – SpaceWire Interface Controller) предназначен для обеспечения аппаратной поддержки функций внутрисистемных коммуникаций с использованием протокола SpaceWire.

В микропроцессоре имеется два контроллера: SWIC0 и SWIC1.

Основные особенности контроллера:

- контроллер разработан в соответствии с международным стандартом ECSS-E-50-12C;
- обеспечивает функционирование одного дуплексного канала связи со скоростью от 2 до 300 Мбит/с в каждую сторону;
- реализация контроллера охватывает уровни стека протоколов SpaceWire, от сигнального до сетевого (частично) уровня;
- аппаратное детектирование ошибок связи: рассоединение, ошибки четности;
- встроенные LVDS приемопередатчики в соответствии со стандартом стандарта ANSI/TIA/EIA-644(LVDS);
- встроенные в приемник LVDS резисторы-терминаторы;
- четыре канала DMA (два канала данных и два канала дескрипторов пакетов);
- обмен данными через DMA с памятью словами по 64 бита.

10.2 Блок схема

Структура контроллера коммуникационного канала по стандарту SpaceWire приведена на Рисунок 10.1. Основой контроллера является Link Interface (см. Figure 8-1 стандарта ECSS-E-50-12C), реализующий функции кодера/декодера SpaceWire. Кодер/декодер SpaceWire через драйверы LVDS подключен к физическим линиям связи.

Контроллер канала SW взаимодействует с центральным процессором через шину AHB (работа с программно-доступными регистрами контроллера) и FIFO-подобный интерфейс с DMA (прием/передача пакетов данных). Для взаимодействия с внутренней памятью использованы блоки DMA, поддерживающие интерфейс буферов. На шине CDB SWIC представлен интерфейсом ведомого устройства.



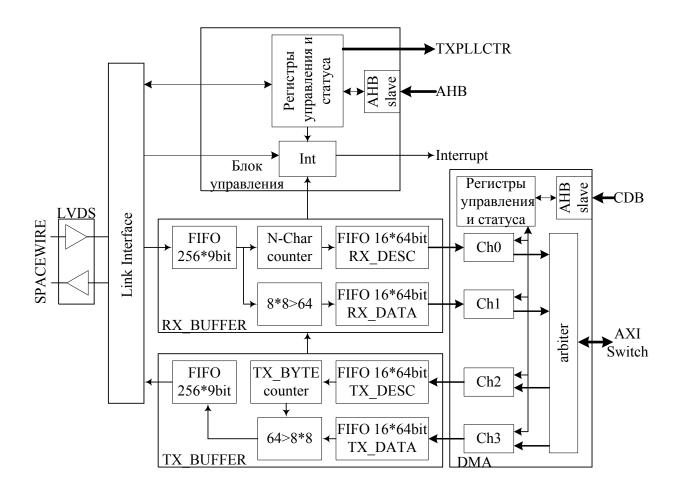


Рисунок 10.1. Структурная схема SWIC

Блок управления по командам центрального процессора задает режимы работы приемопередатчика SpaceWire (Link Interface). В этом блоке содержатся программно управляемый регистр, содержащий коэффициент скорости передачи данных, и доступный программному обеспечению на чтение регистр, в который записывается коэффициент скорости приема данных. Передача управляющих кодов; контроль состояние последнего полученного извне маркера времени, кода распределенного прерывания и кода подтверждения производится через соответствующие регистры блока управления.

Блок формирования прерываний INT формирует необходимые прерывания по состоянию Link Interface.

Буфер приема RX_BUFFER имеет конвейерную организацию и состоит из двух ступеней. Сначала в FIFO_256*9bit буферизируются восьмиразрядные данные, принимаемые от Link Interface. Девятый служебный разряд несет информацию о признаке символа данных N-Char или символе конца пакета EOP. Затем в блоке преобразования формируются 64-разрядные слова данных и поступают в FIFO RX_ DATA. Дескриптор пакета формируется в счетчике N-Char_counter. При поступлении символа данных N-Char счетчик увеличивается на 1, при поступлении символа конца пакета значение счетчика переписывается в выходной буфер RX_DESC, а сам счетчик сбрасывается в 0.



В буфер передачи ТХ_ВUFFER с помощью канала передаваемых данных DMA записываются 64-разрядные слова данных. Содержимое пакетов и их дескрипторы буферизируются в двух FIFO ТХ_DATA и ТХ_DESC соответственно. Данные из буфера передачи в Link Interface поступают побайтно через FIFO 256*9bit. Преобразование 64-хразрядных слов в байты осуществляется в блоке преобразования под управлением счетчика ТХ_ВYTE 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 ЕОР, а в счетчик заносится размер следующего передаваемого пакета из следующего дескриптора.

Буферы приема-передачи предназначены для согласования скоростей передачи данных между коммутатором AXI Switch и каналом SpaceWire.

10.3 Прерывания

Контроллер SWIC формирует три прерывания, описание которых сведено в Таблица 10.1.

Таблица 10.1. 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
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[76]=01 (при включенном режиме 5-и разрядных распределенных прерываний), (CC_01); принят управляющий код C[76]=11 (CC_11); истекло время ожидания таймаута приема кода распределенного прерывания (регистр ISR_tout)



10.4 Перечень регистров SWIC

10.4.1 Общие положения

Перечень программно-доступных регистров контроллера SWIC приведен в Таблица 10.2.

Таблица 10.2. Перечень регистров блока SWIC

Условное обозначение	Описание	Тип доступа	Адрес
HW VER	Номер версии контроллера	RD	0x00
STATUS	Регистр состояния	WR/RD	0x04
RX_CODE	Регистр управляющего символа, принятого из сети (маркера времени, кода распределенного прерывания, кода подтверждения распределенного прерывания или кода СС11)	RD	0x08
MODE_CR	Регистр режима работы	WR	0x0C
TX_SPEED	Регистр коэффициента скорости передачи	WR	0x10
TX_CODE	Регистр управляющего символа (маркера времени, кода распределенного прерывания, кода подтверждения, кода СС11) для передачи в сеть	WR	0x14
RX_SPEED	Регистр скорости приема данных в канале SpaceWire	RD	0x18
	Не используется		
CNT_RX_PACK	Регистр счетчика принятых пакетов ненулевой длины	RD/WR	0x20
ISR L	Младшие разряды регистра ISR	RD/WR	0x24
ISR H	Старшие разряды регистра ISR	RD/WR	0x28
TRUE_TIME	Регистр, содержащий значение последнего правильного маркера времени и последнего принятого из сети маркера времени	RD	0x2C
TOUT CODE	Регистр размера таймаутов	RD/WR	0x30
ISR tout L	Младшие разряды регистра флагов таймаутов ISR	RD/WR	0x34
ISR tout H	Старшие разряды регистра флагов таймаутов ISR	RD/WR	0x38
LOG ADDR	Регистр логического адреса	RD/WR	0x3C
ack_nonack_regime	Регистр управления режимом распределенных прерываний (с подтверждениями или без подтверждений)	RD/WR	0x40
isr touts2	Регистр таймаутов кодов распределенных прерываний 2	RD/WR	0x44
ISR_handler_TERM_FUN CT	Регистр флагов функций терминального узла –обработчика	RD/WR	0x48
ISR SPEC	Регистр рассылки управляющих кодов в специальный набор портов	RD/WR	0x4C
ISR_1101	Регистр флагов приема управляющих кодов, назначение которых не определено в текущей версии стандарта	RD/WR	0x50
ISR_MACK_1101	Регистр маски портов, из которых не должны приниматься управляющие коды, назначение которых не определено в текущей версии стандарта	RD/WR	0x54
INT_RESET	Регистр параметров команды внешнего сброса	WR/RD	0x58
STATUS2	Регистр состояния		0x5C
MODE_CR2	Регистр режима работы 2		0x60
Int_H_mack	Старшая половина регистра маски распределенных прерываний	WR/RD	0x64
Int_L_mack	Младшая половина регистра маски распределенных прерываний	WR/RD	0x68
Ack_H_mack	Старшая половина регистра маски кодов подтверждения	WR/RD	0x6C
Ackl_L_mack	Младшая половина регистра маски кодов подтверждения	WR/RD	0x70
AUTO_SPEED_MANAG E	Регистр параметров автоматической установки скорости передачи	WR/RD	0x74



Условное обозначение	Описание	Тип доступа	Адрес
ISR_source_TERM_FUN CT	Регистр флагов функций терминального узла –источника		0x78
ISR_spec_term_funct	Регистр признака специальной функции для терминального узла – обработчика		0x7C
ISR_L_reset	Младшая половина регистра глобального сброса ISR		0x80
ISR_H_reset	Старшая половина регистра глобального сброса ISR		0x84

10.5 Описание регистров SWIC

10.5.1 Peгистр HW_VER

Регистр номера версии SWIC. При чтении этого регистра выводится номер версии аппаратной реализации SWIC. В 1892ВК016 аппаратная версия SWIC – «0х0000 0005».

Таблица 10.3. Назначение разрядов регистра HW VER

Номер разряда	Условное обозначение	Описание
31:0	HW VER	Номер версии SWIC

10.5.2 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 10.4.

Таблица 10.4. Назначение разрядов регистра STATUS

Номер разряда	Условное обозначение	Описание
0	DC_ERR	Признак ошибки рассоединения (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».



Л
И
М
М
И
_
И
га
poca).
,
poca).
по
K,
,



Номер разряда	Условное обозначение	Описание
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 ТІМЕ посредством записи 1 в этот разряд. Исходное состояние «0».
15	GOT_INT	Принят код распределенного прерывания из сети
	331_111	«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 TIME посредством
		записи 1 в этот разряд.
16	GOT ACK	Исходное состояние «0». Принят код подтверждения из сети
10	GO1_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 ТІМЕ 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	LINK	Признак прерывания LINK (соединение установлено).
		Устанавливается, если оно не замаскировано в регистре
		режима. Сбрасывается при сбросе соответствующего
		прерывания.
10	EDD	Исходное состояние «0».
19	ERR	Признак прерывания ERR (обнаружена ошибка в канале). Устанавливается, если оно не замаскировано в регистре
		режима. Сбрасывается при сбросе соответствующего
		прерывания. Исходное состояние «0».
20	TIME	Признак прерывания TIME (получен управляющий код).
		Устанавливается, еслили оно не замаскировано в регистре
		режима. Сбрасывается при сбросе соответствующего
		прерывания.
		Исходное состояние «0».



Номер разряда	Условное обозначение	Описание
21	CC_11	Признак принятия управляющего кода С[76]=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».
22	CC_01	Признак принятия управляющего кода С[76]=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	-	Не используются
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

10.5.3 Регистр RX_CODE

Регистр принятого из сети управляющего кода. Назначение разрядов регистра приведено в Таблица 10.5. Исходное состояние регистра не определено.

Таблица 10.5. 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 С[76]=11 принятого из сети последним

10.5.4 Регистр MODE_CR

Регистр режима работы. Назначение разрядов регистра приведено в Таблица 10.6.

Таблица 10.6. 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



Номер разряда	Условное обозначение	Назначение
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	LINK_Loopback	Loopback (перед Link Interface)
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 10 МГц, 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-битные коды
		распределенных прерываний, если в 1 – то используются 5-
1.7		битные коды распределенных прерываний
17	I DIV MACV	Не используется
18	LINK_MASK	Маска прерывания LINK: 1 - значение прерывания LINK отображается в регистрах
		STATUS и QSTR2;
		0 – не отображается
19	ERR MASK	Маска прерывания ERR:
	LKK_WASK	1 - значение прерывания LINK отображается в регистрах
		STATUS и QSTR2;
		0 – не отображается
20	TIME MASK	Маска прерывания TIME:
	_	1 - значение прерывания LINK отображается в регистрах
		STATUS и QSTR2;
		0 – не отображается
21	CTR	Если этот бит установлен в 1, то установка соединения
		выполняется без ожидания таймаутов (используется в
		отладочном режиме)
22	TCode_mask	Маска формирования прерывания TIME при получении тайм-
		кода (маркера времени):
		0 – прерывание не устанавливается;
22	D.IT. 1	1 – прерывание устанавливается
23	INT_mask	Маска формирования прерывания TIME при получении кода
		распределенного прерывания или кода подтверждения:
		0 – прерывание не устанавливается;
		1 – прерывание устанавливается



Номер разряда	Условное обозначение	Назначение
24	CC_11_mask	Маска формирования прерывания ТІМЕ управляющего кода
		C[76]=11:
		0 – прерывание не устанавливается;
25	CC 011-	1 – прерывание устанавливается
25	CC_01_mask	Маска формирования прерывания ТІМЕ при получении
		управляющего кода C[76]=01 (dIRQ_regime=1):
		0 – прерывание не устанавливается;
26	DIT 441-	1 – прерывание устанавливается
26	INT_tout_mask	Маска формирования прерывания ТІМЕ по факту таймаута
		получения кода подтверждения:
		0 – прерывание не устанавливается;
28:27	DIT tout allow	1 – прерывание устанавливается
28:27	INT_tout_allow	Разрешение контроля таймаутов получения кодов
		подтверждения:
		00 – контроль таймаутов запрещен 01 – выполняется контроль таймаутов и установка флагов
		от – выполняется контроль таимаутов и установка флагов истечения таймаутов
		10 – выполняется контроль таймаутов, установка флагов
		истечения таймаутов и отправка кода подтверждения в сеть
29	LVDS mode	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.

10.5.5 Регистр ТХ_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 МГп.

В разряды 28:20 этого регистра записывается значение коэффициента для подсчета таймаутов установки соединения (6,4 мкс и 12,8 мкс). Значение данного коэффициента зависит от тактовой частоты работы СРU, на которой осуществляется подсчет таймаутов.



Значение после сброса для этого регистра — 0x0A, что соответствует тактовой частоте CPU $100~M\Gamma$ ц.

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

Таблица 10.7. Назначение разрядов регистра ТХ SPEED

Таблица 10.7. Назначение разрядов регистра ТХ_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	
28:20	COEFF_10	Значение коэффициента для подсчета таймаутов установки соединения. Содержимое этого поля должно быть целым от деления тактовой частоты работы CPU (CLK) на 10: 0x05 - при CLK = 50 или 55 МГц; 0x0A - при CLK = 100 или 105 МГц; 0x14 - при CLK = 200 или 205 МГц и т.д.	
3129	-	Резерв	



10.5.6 Peгистр RX_SPEED

Назначение разрядов регистра RX_SPEED приведено в Таблица 10.8.

Таблица 10.8. 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*8*FCLK/1024) Мбит/с, где FCLK - тактовая частота работы CPU (CLK). Например, если RX_SPEED=128 (десятичное), а FCLK=100 МГц, то скорость приема данных равна 100 Мбит/с		

Исходное состояние регистра – нули.

10.5.7 Регистр ТХ СОDE

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 10.9. Назначение разрядов регистра ТХ 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	

10.5.8 Регистр CNT_RX_PACK

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

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

Таблица 10.10. Назначение разрядов регистра CNT RX PACK

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



10.5.9 Регистр CNT_RX0_PACK

Регистр счетчика принятых пустых пакетов. Значение регистра увеличивается на 1 каждый раз, когда из DS макроячейки прочитывается символ конца пакета, если ему не предшествовал хотя бы один символ данных. Исходное состояние регистра «0».

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

Таблица 10.11. Назначение разрядов регистра CNT_RX0_PACK

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

10.5.10 Регистр ISR_L

В этот регистр отображается младшая (31..0) часть регистра ISR Регистр ISR содержит информацию о принятых и отправленных кодах распределенных прерываний и подтверждения. Если из сети получено распределенное прерывание, то бит регистра ISR, соответствующий номеру распределенного прерывания устанавливается в 1 (если он уже не был установлен в 1). Аналогично, если в регистр TX_CODE осуществляется запись кода распределенного прерывания, соответствующий бит регистра ISR устанавливается в 1.

Если из сети получен код подтверждения, то бит регистра ISR, соответствующий номеру кода подтверждения устанавливается в 0 (если он уже не был установлен в 0). Аналогично, 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 приходит из сети очень часто, быстрее, чем процессор может среагировать на очередное прерывание и прочитать код. Если даже в регистре RX_CODE код распределенного прерывания или код подтверждения будет перезаписан следующим, информация о нем не будет утрачена – она сохранится в регистре ISR.

Существует возможность программного сброса отдельных битов ISR. Для этого необходимо записать в соответствующие биты 1. (Если в бит записывается значение 0, то его значение не меняется)

Таблица 10.12. Назначение разрядов регистра ISR L

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



10.5.11 Регистр ISR_H

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

Таблица 10.13. Назначение разрядов регистра ISR_H

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

10.5.12 Peгистр 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 10.14. Исходное состояние регистра «0».

Таблица 10.14. Назначение разрядов регистра TRUE TIME

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

10.5.13 Регистр TOUT CODE

В этот регистр записываются значение периода для глобального счетчика таймаутов (в количестве тактов локальной частоты) и максимальные значения локальных счетчиков таймаутов ожидания кодов подтверждения распределенных прерываний. Отдельный локальный счетчик таймаутов соответствует каждому разряду ISR. Если в SWIC поступает код распределенного прерывания, то запускается соответствующий ему счетчик локальных таймаутов. Он декрементируется каждый раз при завершении очередного периода счета глобального счетчика таймаутов.

Таблица 10.15. Назначение разрядов регистра TOUT CODE

Номер разряда	Условное обозначение	Описание
150	GLOB_COU	Значение периода глобального счетчика (задается в тактах локальной частоты)
2016	LOC_COU1	Значение таймаута ожидания кода подтверждения (на код прерывания, отправленный процессором через SWIC)
2521	LOC_COU2	Значение таймаута ожидания кода подтверждения (на код прерывания, принятый из сети)
31:26	Не используется	



10.5.14 Peгистр 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 10.16. Назначение разрядов регистра ISR tout L

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

10.5.15 Регистр ISR tout H

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

Таблица 10.17. 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».

10.5.16 Регистр LOG_ADDR

В этом регистре хранится значение логического адреса, добавляемого к пакету по умолчанию, если установлен соответствующий режим. Длина логического адреса может быть от одного до 4 байтов, она определяется значением дескриптора пакета.

Таблица 10.18. Назначение разрядов регистра LOG ADDR

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



10.6 Работа со SWIC. Пакеты данных, дескрипторы пакетов

В этой главе описывается формирование пакетов данных в памяти для передачи в канал, формат пакетов данных, дескрипторов, передача данных из памяти в канал SpaceWire, прием данных из канала SpaceWire в память, интерпретирование принятых данных, системные сообщения.

10.6.1 Расположение данных в памяти

Рассмотрим пример (см. Рисунок 10.4) представления данных в системной памяти, если для данных выделен один сегмент памяти. Пусть в системную память из канала SpaceWire было записано 3 пакета. Первый пакет имеет размер 10 байт и заканчивается символом ЕОР. Второй пакет имеет размер 8 байт и заканчивается символом ЕЕР. Третий пакет имеет размер 11 байт и заканчивается символом ЕОР. Собственно, пакеты хранятся в сегменте памяти, выделенном процессором для записи данных. Для выравнивания по границам 64-х разрядных слов, первый и третий пакеты дополнены двумя и одним байтом соответственно.

Дескрипторы хранятся в сегменте памяти, выделенном процессором для записи дескрипторов. В дескрипторах указаны размеры пакетов в байтах — 0Ah, 08h и 0Bh соответственно. В дескрипторах хранится так же информация о типе конца пакета. В разряд 31 дескриптора записывается 1, что указывает процессору на то, что дескриптор заполнен действительными данными.

10.6.2 Схема обработки данных процессором

В данном примере пакеты могут быть обработаны процессором в соответствии со следующей схемой. Процессор прочитывает первое слово из блока, выделенного для дескрипторов – первый дескриптор. По дескриптору он определяет тип конца пакета, в соответствии с этим решает, как его обрабатывать. По дескриптору он определяет действительный размер пакета и извлекает данные, относящиеся к пакету 1. Для того чтобы вычислить начальный адрес второго пакета к начальному адресу блока данных добавляется размер первого пакета и выполняется округление до границы ближайшего слова. После того, как первый пакет полностью обработан, процессор прочитывает дескриптор второго пакета. Обработка остальных пакетов выполняется аналогично. Процесс обработки очереди пакетов заканчивается, когда 31 разряд очередного дескриптора равен 0.



10.6.3 Прием данных из канала SpaceWire.

Маршрут принимаемых данных и схема их обработки приведены на Рисунок 10.2.

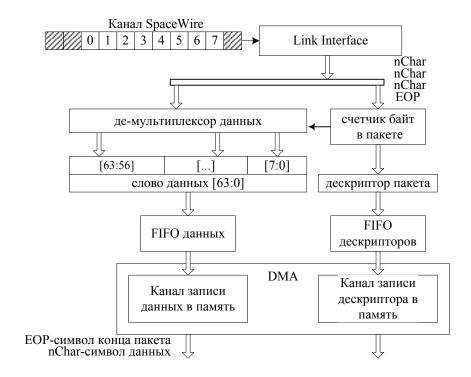


Рисунок 10.2. Схема приема данных из канала SpaceWire

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 Link Interface 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.

10.6.4 Передача данных в канал SpaceWire

Процесс передачи пакетов данных из системной памяти в канал через контроллер, а также преобразование форматов данных показаны на Рисунок 10.3.

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

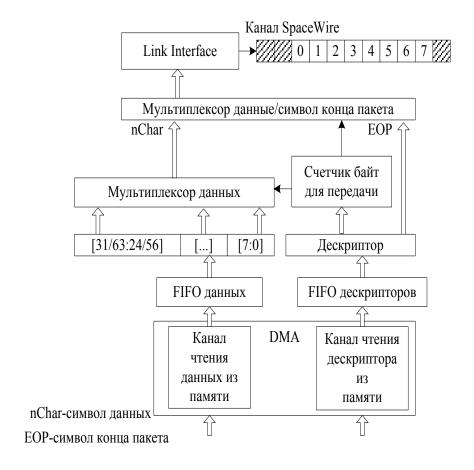


Рисунок 10.3. Передача данных из системной памяти в канал SpaceWire

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 вставляются символы концов пакетов ЕОР или ЕЕР. Если в канал SpaceWire передаются пакеты,



стенерированные в данном узле, то предполагается, что они всегда должны заканчиваться символом ЕОР. Однако пакеты могут проходить через данный процессорный модуль транзитом. В этом случае они могут заканчиваться символом ЕЕР. Коды маркеров ЕОР или ЕЕР формируются контроллером аппаратно, на основании кодов дескриптора пакета на передачу (разряды 29:30 дескриптора пакета). Сами дескрипторы пакетов на передачу в сеть из основной памяти формируются программно.

Распаковка 64-разрядного слова в последовательность из 8 байт при передаче из контроллера выполняется по правилу, согласованному с правилом упаковки байтов при приеме данных из канала в контроллер.

Блок передачи вначале передает в 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.

10.6.5 Выравнивание границ пакетов по границам слов

Рассмотрим выравнивание пакетов данных на примере Рисунок 10.4. 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 — ЕОР, или признак завершения пакета с ошибкой — ЕЕР).



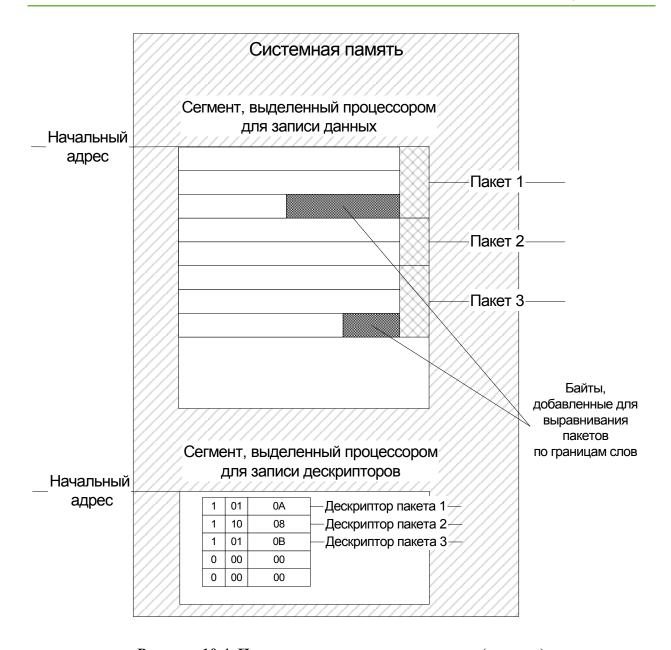


Рисунок 10.4. Представление данных в памяти (пример)

10.6.6 Формат дескриптора пакета

Дескриптор пакета имеет следующую структуру:

- [63:32] не используются;
- [31] признак заполнения дескриптора действительными данными. Бит учитывается только при приёме пакетов (позволяет процессору идентифицировать конец очереди дескрипторов в памяти). При передаче пакетов этот бит не учитывается (DMA вычитывает всю область дескрипторов, заданную процессором). До запуска приёма, все 31-е биты дескрипторов области приёма должны быть обнулены программно; DMA не обнуляет 31-е биты не принятых дескрипторов, DMA только записывает '1' в 31-е биты принятых дескрипторов;



- [30:29] тип конца пакета:
 - 00 передавать данные пакета из памяти и не вставлять конец пакета;
 - 01 EOP;
 - -10 EEP;
 - 11 передавать данные пакета из памяти 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.

10.6.7 Возможность передачи коммуникационного пакета

Дескриптор пакета в битах [30:29] содержит информацию о типе передаваемого пакета. Пакет может иметь нормальное окончание (ЕОР, код 01), ошибочное окончание (ЕЕР, 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 раздельно формировать данные для коммуникационного пакета и данные основного пакета. При этом оба пакета располагаются друг за другом, каждому соответствует свой дескриптор, и данные пакетов выровнены по длине 64-разрядных слов.

Для тех случаев, когда программист предпочитает иметь заранее сформированный коммуникационный пакет, который бы вставлялся перед передаваемым пакетом данных из памяти, предусмотрен режим передачи коммуникационного пакета из регистра LOG_ADDR. В этом случае нет необходимости формировать коммуникационный пакет для каждого пакета данных. Программисту следует записать в регистр LOG_ADDRS данные коммуникационного пакета (максимум - 4 байта) и сформировать для него дескриптор по описанной выше схеме (в области дескрипторов на передачу, перед дескрипторами данных, для которых требуется вставка коммуникационного пакета LOG_ADDR). Всегда, встречая дескриптор с кодом 11, SWIC передает число байт, указанное в этом дескрипторе, из регистра LOG_ADDR, а не из памяти. После передачи данных из этого регистра в канал не высылается 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 ЕОР или ЕЕР.

10.6.8 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.

10.6.9 Маркеры времени

Маркеры времени - системная функция стандарта SpaceWire. Они предназначены для синхронизации системных часов взаимодействующих систем.

При передаче данных маркеры времени имеют наивысший приоритет. Маркер времени записывается в регистр TX_CODE. Этот же регистр используется и для передачи в сеть кодов распределенных прерываний и кодов подтверждения. После записи Link Interface дожидается окончания передачи символа данных или служебного символа и начинает передачу маркера времени, после окончания передачи маркера времени продолжается передача потока данных. Для того, чтобы не произошло утраты управляющего символа в результате перезаписи его в регистре TX_CODE следующим управляющим символом до



передачи в сеть необходимо программно отслеживать значение бита [17] (FL_CONTROL) регистра состояния. Если этот бит установлен в 0, то SWIC готов к передаче следующего управляющего символа. Если в момент записи в регистр TX_CODE нового значения этот бит был установлен в 1, то существует вероятность того, что предыдущий управляющий код не будет передан в сеть.

В канале приема маркер времени выделяется из потока данных и при безошибочном приеме заносится в регистр RX_CODE (разряды 7 - 0) с выставлением соответствующего прерывания, если маркер времени является корректным. Корректным признается маркер времени на 1 больше, чем предыдущий, если предыдущий маркер времени имел значение меньше 63. Если предыдущий маркер времени имел значение 63, то следующий корректный маркер времени должен иметь значение 0. Если маркер времени не является корректным, то его значение так же заносится в соответствующие разряды регистра RX_CODE, однако, прерывание для процессора в данном случае не устанавливается. В начале работы устройства или после сброса маркер времени со значением 1 рассматривается как корректный.

10.6.10 Коды распределенных прерываний

Коды распределенных прерываний являются расширением стандарта SpaceWire. Механизм передачи кодов распределенных прерываний в сеть аналогичен механизму передачи маркеров времени.

При приеме кода распределенного прерывания из сети выполняются следующие действия.

Если соответствующий коду распределенного прерывания разряд регистра ISR установлен в 1, то данное прерывание игнорируется (никаких действий не выполняется). Если соответствующий разряд регистра установлен в 0, то в него записывается 1 и код распределенного прерывания записывается в разряды [15:8] регистра RX_CODE. В этом случае устанавливается прерывание.

10.6.11 Коды подтверждения распределенных прерываний

Коды подтверждения распределенных прерываний являются расширением стандарта SpaceWire. Механизм передачи кодов подтверждения в сеть аналогичен механизму передачи маркеров времени.

При приеме кода подтверждения прерывания из сети выполняются следующие действия. Если соответствующий коду подтверждения разряд регистра ISR установлен в 0, то данный код игнорируется (никаких действий не выполняется). Если соответствующий разряд регистра установлен в 1, то в него записывается 0 и код записывается в разряды [23:16] регистра RX_CODE. В этом случае устанавливается прерывание.



10.6.12 Установка скорости передачи данных

Управление скоростью передачи осуществляется посредством регистра ТХ 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 2 до 400 МГц, в соответствии со стандартом SpaceWire). Если происходит разрыв соединения, то в этот регистр снова необходимо записать коэффициент, соответствующий 10 Мбит/с.

Если установлен режим автоматического контроля скорости, то до установки соединения на TX_PLL подается коэффициент TX_SPEED_10 из разрядов 19:10 регистра TX_SPEED. Он должен соответствовать 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 2 до 400 МГц. При разрыве соединения переход на коэффициент TX_SPEED_10 выполняется автоматически, при повторной установке соединения переход на TX_SPEED так же выполнятся автоматически.

10.6.13 Установление соединения

Для разрешения процесса установки соединения необходимо записать лог <0> в разряд LinkDisabled и <1> в разряд LinkStart регистра режима работы MODE_CR — для запуска канала, WORK_TYPE = <1>.

Критерием успешного установления соединения является прохождение прерывания INT LINK и отсутствие прерывания INT ERR.

После обнаружения прерывания INT_LINK, необходимо считать регистр STATUS и проверить биты DC_ERR, P_ERR, ESC_ERR, CREDIT_ERR на равенство «0». Бит CONNECTED должен быть равен «1». При выполнении этих условий - соединение с удаленной системой установлено.

Для активации функции автоматического восстановления соединения после обрыва связи дополнительно в разряд AutoStart записывается «1». В этом случае после рассоединения из-за ошибок будет выставлено прерывание INT_ERR, и система будет производить повторное установление соединения. Однако следует учитывать, что повторное соединение на скорости выше 10 Мбит/с не предусмотрено стандартом SpaceWire,



вследствие этого при обнаружении рассоединения необходимо снова установить скорость передачи равной 10 Мбит/с.

10.6.14 Определение скорости приема данных

Оценка скорости приема выполняется при разрешенной работе канала и установленном соединении. Скорость приема данных отображается в регистре RX_SPEED[9:0]. После установления соединения скорость должна составлять 10 ± 1 Мбит/с при этом регистр RX_SPEED[9:0] будет равен 0x00000000 ±1 M3P. Разряды регистра с 8 по 31 не используются и при чтении содержат 0.



11. KOMMYTATOP SPFR

11.1 Функциональные параметры и возможности

Коммутатор SpFR имеет следующие функциональные параметры и возможности:

- имеет 4 мультипротокольных порта SpaceFibre/GigaSpaceWire (SpaceWire-RUS);
- пропускная способность каждого порта от 5 Мбод до 1,25 Гбод;
- обеспечивает дуплексный режим приема и передачи данных;
- аппаратное детектирование ошибок связи: рассоединение, ошибки четности;
- каждый порт имеет два виртуальных канала VC0, VC1;
- коммутатор SpFR подключен к коммутатору AXI при помощи двухпортового четырехканального контроллера DMA;
- поддерживает протокол RMAP;
- команды RMAP записи используются только с подтверждением;
- контроллер RMAP выполняет до 32 отложенных команд чтения и записи. Максимальное количество команд (в пределах 32) программируется. Если в контроллере RMAP находится на выполнении максимальное количество команд, то очередная команда не выполняется и в SpF выдается статус ошибки.

11.2 Типы информационных потоков

11.2.1 Введение

Коммутатор SpFR поддерживает следующие основные типы информационных потоков:

- потоки RMAP пакетов с программно-аппаратной обработкой пакетов;
- потоки SpW пакетов, с программной обработкой пакетов. Никакой аппаратной обработки пакетов не производится. Программно для них могут поддерживаться различные протоколы транспортного и прикладного уровня;
- потоки RMAP команд обращений в пространство конфигурации блока SpFR от удаленного сетевого администратора.

Интерпретация типов потоков обращений осуществляется в привязке к логическим номерам виртуальных каналов портов SpFR, из которых они поступают.

Все пакеты, поступающие из виртуального канала VC0, автоматически интерпретируются как RMAP команды обращения в пространство конфигурации.

Для виртуального канала VC1 имеется возможность программной настройки режима интерпретации — либо как поток пакетов RMAP, либо как поток SpW пакетов, не интерпретируемых блоком SpFR и обрабатываемых чисто программно.



11.2.2 Потоки пакетов транспортных протоколов с частичной обработкой в блоке SpFR

Коммутатор SpFR обеспечивает запись принимаемых пакетов RMAP команд в память микросхемы. При этом коммутатор SpFR аппаратно производится частичная проверка и обработка принимаемой RMAP команды.

Обработка содержимого RMAP команд для обращения к NAND Flash осуществляется из памяти с использованием алгоритмов, реализованных программно-аппаратно за пределами коммутатора SpFR.

По завершении выполнения очередной команды обращения к NAND Flash с использованием этих алгоритмов, статусная информация о выполнении команды записывается в коммутатор SpFR, а прочитанные данные – в память. В соответствии с этой статусной информацией SpFR аппаратно выполняет формирование пакетов RMAP ответов и отправку их в сеть SpaceFibre/GigaSpaceWire.

Потоки пакетов транспортных протоколов в память могут поступать из одного или нескольких портов коммутатора (в соответствии с программными настройками). Для каждого из потоков выделяются отдельные области оперативной памяти для работы с ним. Это обеспечивает возможность параллельной записи/чтения данных, соответствующих нескольким потокам.

11.2.3 Потоки пакетов SpW без обработки в коммутаторе SpFR

Потоки пакетов SpW, поступающие в коммутатор SpFR из сети, записываются им в память микросхемы. Они могут соответствовать различным транспортным или прикладным протоколам.

Дальнейшая интерпретация содержимого пакетов, их обработка осуществляется в памяти программно вне коммутатора SpFR.

Если в соответствии с полученной в пакете командой требуется отправка ответа, то ответные данные формируются программно в памяти микросхемы. Далее коммутатором SpFR эти данные прочитываются из памяти, из них формируется пакет SpW и отправляется в сеть.

Потоки пакетов SpW могут поступать в память из одного или нескольких портов коммутатора (в соответствии с программными настройками). Для каждого из потоков выделяются отдельные области памяти для работы с ним. Это обеспечивает возможность параллельной записи/чтения данных, соответствующих нескольким потокам.



11.2.4 Потоки пакетов в пространство конфигурации коммутатора SpFR

Потоки пакетов в пространство конфигурации представляют собой потоки RMAP команд обращений к регистрам и таблице маршрутизации коммутатора SpFR. Данные потоки обрабатываются полностью аппаратно в коммутаторе SpFR.

Режим полностью аппаратной обработки RMAP команд обращений в пространство конфигурации блока SpFR позволяет обеспечить быстрое выполнение этих команд, что может быть очень существенно для организации процесса удаленного администрирования.



11.3 Перечень программно-доступных регистров коммутатора SpFR

Перечень программно-доступных регистров DMA коммутатора приведен в Таблица 11.1.

Таблица 11.1. Перечень программно-доступных регистров коммутатора SpFR

Условное обозначение	Количество	Описание	Тип доступа	Смещение	Исходное состояние
ID VER	1	Регистр версии	RD	0x0	0x00000009
ID_SWITCH	1	Регистр идентификатора SpFR4 (MCT-04)	RD WR	0x4	0x00000000
DMA_CHANNEL S_MODE	1	Регистр режимов функционирования каналов DMA	RD WR	0x8	0x00000001
DMA_CHANNEL S_WORK_EN	1	Регистр разрешения работы каналов DMA	RD WR	0xc	0x00000000
DMA_RMAP_CO NFIGURATION	1	Регистр конфигурирования RMAP DMA	RD WR	0x10	0x00000000
DMA_REGIONS_ 12 STATUS	1	Регистр состояния областей памяти 1, 2 каналов DMA	RD	0x14	0x00000000
DMA_REGIONS_ 34_STATUS	1	Регистр состояния областей памяти 3, 4 каналов DMA	RD	0x18	0x00000000
DMA_PACK_STA TUS	1	Регистр статуса приема пакетов каналов DMA	RD WR	0x1c	0x00000000
DMA_IRQ_MASK	1	Регистр маски прерываний от DMA 1	RD WR	0x20	0x000003FF
DMA_IRQ_MASK 2	1	Регистр маски прерываний от DMA 2	RD WR	0x24	0x00007FFF
DMA_IRQ_MASK	1	Регистр маски прерываний от DMA 2	RD WR	0x28	0x000001FF
DMA_START_AD DR_R1	5	Регистр начального адреса области памяти R1 (регион 1)	RD WR	0x2c	0x00000000
DMA_AREA_SIZ E_R1	5	Регистр размера области памяти R1 (регион 1)	RD WR	0x40	0x00000000
DMA_START_AD DR_R2	5	Регистр начального адреса области памяти R2 (регион 2)	RD WR	0x54	0x00000000
DMA_AREA_SIZ E_R2	5	Регистр размера области памяти R2 (регион 2)	RD WR	0x68	0x00000000
DMA_START_AD DR_R3	5	Регистр начального адреса области памяти R3 (регион 3)	RD WR	0x7c	0x00000000
DMA_AREA_SIZ E_R3	5	Регистр размера области памяти R3 (регион 3)	RD WR	0x90	0x00000000
DMA_START_AD DR R4	5	Регистр начального адреса области памяти R4 (регион 4)	RD WR	0xa4	0x00000000
DMA_AREA_SIZ E_R4	5	Регистр размера области памяти R4 (регион 4)	RD WR	0xb8	0x00000000
FIFO_ANS	1	Регистр FIFO декскрипторов ответов RMAP команд чтения и записи	RD WR	0xcc	0x00000000
FIFO_STATUS	1	Регистр статуса FIFO	RD WR	0xd0	0x00000000



Условное обозначение	Количество	Описание	Тип доступа	Смещение	Исходное состояние
REJ_STATUS	1	Регистр признаков отброшенных команд	RD WR	0xd4	0x00000000
RMAP_REC_PAC K_COUNTERS_1	1	Регистр счетчиков принятых RMAP команд 1	RD WR	0xd8	0x00000000
RMAP_REC_PAC K_COUNTERS_2	1	Регистр счетчиков принятых RMAP команд 2. В данной версии не используется!!!	RD WR	0xdc	0x00000000
ID_NET	1	Регистр идентификации сетевых линков SpFR4 (MCT-04)	RD WR	0xe0	0x00000000
MODE_R	1	Регистр режима блока коммутатора SpFR4 (MCT-04) 1	RD WR	0xe4	0x0005900D
MODE_R1	1	Регистр режима блока коммутатора SpFR4 (MCT-04) 2 (продолжение)	RD WR	0xe8	0x00004000
STATE_R	1	Регистр состояния блока коммутатора SpFR4 (MCT-04)	RD WR	0xec	0x00000000
STATE_R_CONN ECT	1	Регистр состояния блока коммутатора SpFR4 (MCT-04) - информация о наличии соединения	RD WR	0xf0	0x00000000
STATE_R_ERROR	1	Регистр состояния блока коммутатора SpFR4 (MCT-04) - информация о разрывах соединения	RD WR	0xf4	0x0000000F
IRQ_MASK	1	Регистр глобальных масок прерываний	RD WR	0xf8	0x00000000
GIGA_SPW_STAT US	4	Регистр состояния мультипротокольного порта в режиме gigaSpW	RD WR	0xfc	0x00000000
GIGA_SPW_MOD E	4	Регистр режима работы мультипротокольного порта в режиме gigaSpW	RD WR	0x10c	0x00B20D0D
GIGA_SPW_TRA NSMISSION_PAR AMETERS	4	Регистр параметров передачи порта gigaSpW	RD WR	0x11c	0x00000108
GIGA_SPW_CRE DIT_PARAMETE RS	4	Регистр параметров кредитования порта gigaSpW	RD WR	0x12c	0x10003808
PMA_STATUS	4	Регистр состояния РМА - используется и в режиме порта SpF и в режиме gigaSpW	RD	0x14c	0x00000000
PMA_MODE	4	Регистр режима работы РМА - используется и в режиме порта SpF и в режиме gigaSpW	RD WR	0x15c	0x01C02E00
SpF_PORT_MODE _VC_PARAMS	8	Регистр режима порта SpF VC_PARAMS	RD WR	0x16c	0x00080A
SpF_PORT_MODE _VC_TSLOTS_L	8	Регистр флагов таймслотов (младшая часть), эти таймслоты используются и для конфигурационного порта	RD WR	0x18c	0xFFFFFFF



Условное обозначение	Количество	Описание	Тип доступа	Смещение	Исходное состояние
SpF_PORT_MODE _VC_TSLOTS_H	8	Регистр флагов таймслотов (старшая часть), эти таймслоты используются и для конфигурационного порта	RD WR	0x1ac	0xFFFFFFF
SpF_PORT_MODE _VC_FR_LL	4	Регистр режима порта SpF VC framing lane	RD WR	0x1cc	0x03FC0000
SpF_PORT_MODE _LANE_CONSTA NTS	4	SpF_PORT_MODE_LANE_CON STANTS	RD WR	0x1dc	0x40000F00
SpF_PORT_STAT US_VC1	4	Регистр состояния порта SpF Virtual channels 1	RD	0x1ec	0x00000000
SpF_PORT_STAT US_VC2	4	Регистр состояния порта SpF Virtual channels 2 and retry	RD WR	0x1fc	0x00000000
SpF_PORT_CUR_ CREDIT_VALUE	4	Регистр состояния - значение cur_credit_counter порта SpF	RD	0x20c	0x00000000
SpF_PORT_STAT US_LL_EL	4	Регистр состояния порта SpF Lane Encoding	RD WR	0x21c	0x00000004
ADG	6	Регистр адаптивной групповой маршрутизации порта	RD WR	0x22c	0x000000
CHANNEL_ROUT ING	1	Регистр режима маршрутизации	RD WR	0x244	0x00000000
CHANNEL_REGI STERS	6	Регистры номеров портов для режима маршрутизации каналов.	RD WR	0x248	0x00000000
DRT_TOUTS	1	Регистр таймаутов ожидания приема передачи Nchar (для мультипротокольных портов в режиме gigaSpW)	RD WR	0x260	0x00000000
DRT_TOUTS_VA LIDS	1	Регистр разрешения подсчета таймаутов ожидания приема передачи Nchar (для портов SpW и мультипротокольных портов в режиме gigaSpW)	RD WR	0x264	0x00000000
DRT_TOUTS_FLS	1	Регистр флагов срабатывания таймаутов ожидания приема передачи Nchar (для портов SpW и мультипротокольных портов в режиме gigaSpW)	RD WR	0x268	0x00000000
DA_TOUTS	1	Регистр таймаутов арбитража	RD WR	0x26c	0x00000000
DA_SPF_TOUTS	13	Регистры таймаутов арбитража и разрешения подсчета таймаутов для виртуальных каналов (SpF) или портов SpW, gigaSpW	RD WR	0x270	0x00000000
DA_SPF_TOUTS2	13	Регистры 2 (продолжение) таймаутов арбитража и разрешения подсчета таймаутов для виртуальных каналов (SpF) или портов SpW, gigaSpW	RD WR	0x2a4	0x00000000



Условное обозначение	Количество	Описание	Тип доступа	Смещение	Исходное состояние
DA_SPF_TOUTS_ FLS	13	Регистры флагов таймаутов арбитража и разрешения подсчета таймаутов для виртуальных каналов (SpF) или портов SpW, gigaSpW; флагов переполнения счетчиков кредитования	RD WR	0x2d8	0x00000800
INCORRECT_VC_INFO	6	Регистры информации о пакетах, которые поступили в VC с некорректными номерами	RD WR	0x30c	0x00000000
SPEC_VC_ARB_P ARAMETERS	1	Регистр специальных параметров арбитража для виртуальных каналов	RD WR	0x324	0x00000000
SPEC_ARB	1	Регистр специальных условий арбитража	RD WR	0x328	0x00000000
CONTROL_CONN ECTION	1	Регистр контроля соединения (для мультипротокольных портов в режиме gigaSpW)	RD WR	0x32c	0x0008008A
IRQ_CONNECT	1	Регистр контроля соединения (для мультипротокольных портов в режиме SpF)	RD WR	0x330	0x00004001
STATE_CONNEC TION	1	Регистр флагов состояния соединения (для мультипротокольных портов в режиме gigaSpW)	RD WR	0x334	0x00000000
STATE_CONNEC TION_SpF	1	Регистр флагов состояния соединения (для мультипротокольных портов в режиме SpF)	RD WR	0x338	0x00000000
INT_SEPARATE_ SEND_MASK	4	Регистр маски отправки отдельных кодов распределенных прерываний мультипротокольного порта	RD WR	0x33c	0x00000000
INT_SEPARATE_ REC_MASK	4	Регистр маски приема отдельных кодов распределенных прерываний мультипротокольного порта	RD WR	0x34c	0x00000000
ACK_SEPARATE _SEND_MASK		Регистр маски отправки отдельных кодов подтверждения мультипротокольного порта	RD WR	0x35c	0x00000000
ACK_SEPARATE _REC_MASK	4	Регистр маски приема отдельных кодов подтверждения мультипротокольного порта	RD WR	0x36c	0x00000000
CCODE_OUT	. I	Регистр управляющего кода для отправки в сеть	RD WR	0x37c	0x00000000
CUR_TIME	1	Регистр маркера времени из сети	RD WR	0x380	0x00000000
ISR_L	1	Младшая половина регистра ISR ISR_L	RD WR	0x384	0x00000000
ISR_H	1	Старшая половина регистра ISR ISR_H	RD WR	0x388	0x00000000



Условное обозначение	Количество	Описание	Тип доступа	Смещение	Исходное состояние
INTR_IRQ_MASK _L	1	Регистр маски прерывания для встроенного процессора при приеме кода распределенного прерывания (младшая половина)	RD WR	0x38c	0x00000000
INTR_IRQ_MASK _H	1	Регистр маски прерывания для встроенного процессора при приеме кода распределенного прерывания (старшая половина)	RD WR	0x390	0x00000000
INTA_IRQ_MASK _L	1	Регистр маски прерывания для встроенного процессора при приеме кода подтверждения (младшая половина)	RD WR	0x394	0x00000000
INTA_IRQ_MASK _H	1	Регистр маски прерывания для встроенного процессора при приеме кода подтверждения (старшая половина)	RD WR	0x398	0x00000000
CCODES_MASK_ TIME	1	Регистр маски портов для ТІМЕ кодов	RD WR	0x39c	0x00000000
CCODES_MASK_ REQ_ACK	1	Регистр маски портов для кодов распределенных прерываний и подтверждений.	RD WR	0x3a0	0x00000000
CCODES_MASK_ REQ	1	Регистр маски портов для кодов распределенных прерываний.	RD WR	0x3a4	0x00000000
CCODES_MASK_ ACK	1	Регистр маски портов для кодов распределенных и подтверждений.	RD WR	0x3a8	0x00000000
CCODES_MASK_ ADD	1	Регистр входных портов для управляющих кодов, назначение которых не определено стандартом.	RD WR	0x3ac	0x00000000
DIST_INTS_TOUT S1		Регистр таймаутов распределенных прерываний 1.	RD WR	0x3b0	0x00000000
DIST_INTS_TOUT S2	1	Регистр таймаутов распределенных прерываний 2.	RD WR	0x3b4	0x00000000
ACK_NON_ACK_ REGIME		Регистр флагов режима обработки распределенных прерываний (с подтверждением / без подтверждения)	RD WR	0x3b8	0x00000000
CCODES_SPEC_R EGIME	1	Регистр специального режима работы с управляющими кодами	RD WR	0x3bc	0x00000000
SPEC_ISR_REGI ME	1	Регистр флагов автоматической отправки подтверждения	RD WR	0x3c0	0x00000000
INTER_HANDLE R_TERM_FUNCT		Флаги-признаки обработчика для распределенных прерываний	RD WR	0x3c4	0x00000000
ISR_SOURCE_TE RM_FUNCT	1	Флаги-признаки источника для распределенных прерываний	RD WR	0x3c8	0x00000000
ISR_TOUTS_FLS_ L	1	Младшая половина регистра флагов таймаутов	RD WR	0x3cc	0x00000000
ISR_TOUTS_FLS_ H	1	Старшая половина регистра флагов таймаутов	RD WR	0x3d0	0x00000000



Условное обозначение	Количество	Описание	Тип доступа	Смещение	Исходное состояние
ISR_1101	1	Значение регистра ISR для управляющих кодов, назначение которых не определено стандартом	RD WR	0x3d4	0x00000000
EXTERNAL_RES ET_PARAMETER S	1	Регистр параметров удаленного сброса	RD WR	0x3d8	0x00000000
CCODES_INTERV AL	1	Регистр интервала времени между последовательными отправками управляющих кодов - используется и в режиме SpF и в режиме gigaSpW	RD WR	0x3dc	0x00000000
BROADCAST_M ODE	1	Регистр режима передачи пакетов BROADCAST	RD WR	0x3e0	0x00000001
BROADCAST_NU MS	1	Регистр номеров виртуальных каналов BROADCAST, обрабатываемых данным блоком	RD WR	0x3e4	0x00000000
BROADCAST_DA TA_OUT_L	1	Регистр младшей части поля данных отправляемого пакета BROADCAST	RD WR	0x3e8	0x00000000
BROADCAST_DA TA_OUT_H	1	Регистр старшей части поля данных отправляемого пакета BROADCAST	RD WR	0x3ec	0x00000000
BROADCAST_DA TA_OUT_PARAM ETERS	1	Регистр параметров отправляемого пакета BROADCAST	RD WR	0x3f0	0x00000000
BROADCAST_DA TA_OUT_PARAM ETERS FLS	1	Регистр флагов пакета BROADCAST	RD WR	0x3f4	0x00000000
BROADCAST_DA TA_IN_L	1	Регистр младшей части поля данных принятого пакета BROADCAST	RD	0x3f8	0x00000000
BROADCAST_DA TA_IN_H	1	Регистр старшей части поля данных принятого пакета BROADCAST	RD	0x3fc	0x00000000
BROADCAST_DA TA_IN_PARAME TERS	1	Регистр параметров принятых пакетов BROADCAST	RD WR	0x400	0x00000000
BROADCAST_IN_ STATUS1	1	Регистр флагов принятых корректных пакетов BROADCAST 1	RD WR	0x404	0x00000000
BROADCAST_IN_ STATUS_ALL1	1	Регистр флагов принятых пакетов BROADCAST (корректных и не корректных) 1	RD WR	0x408	0x00000000
TIME_SLOTS_PA RAMETERS	1	Регистр параметров таймслотов для класса сервиса Scheduling	RD WR	0x40c	0x00000000
TIME_SLOTS_SE TTINGS	1	Регистр настройки таймслота	RD WR	0x410	0x00000000



В Таблица 11.1 в графе «Смещение» указано смещение адреса регистра относительно базового.

11.4 Описание программно-доступных регистров

11.4.1 Регистр ID VER

Назначение разрядов регистра версии ID_VER приведено в Таблица 11.2.

Таблица 11.2. Назначение разрядов регистра ID VER

Номер разряд		Описание	Тип доступа	Исходное состояние
31:0	ID_SWITCH	Поле версии коммутатора	RD	0x00000009

11.4.2 Регистр ID_SWITCH

Назначение разрядов регистра идентификатора сетевого коммутатора ID_SWITCH приведено в Таблица 11.3.

Таблица 11.3. Назначение разрядов регистра ID SWITCH

Номер	р Условное	Описание	Тип	Исходное
разряд	ца обозначение	Officating	доступа	состояние
31:0	ID_SWITCH	Поле идентификатора сетевого коммутатора	RD WR	0x00000000

11.4.3 Регистр DMA_CHANNELS_MODE

Назначение разрядов регистра DMA CHANNELS MODE приведено в Таблица 11.4.

Таблица 11.4. Назначение разрядов регистра DMA CHANNELS MODE

Номер разряда	Условное обозначение	Описание	Тип доступа	Исходное состояние
0	CHANNEL_0_M ODE	Поле режима функционирования канала 0 (обращения в пространство конфигурации SpFR) DMA. Если данный разряд установлен в 0, то обращения в пространство конфигурации обрабатываются программно, если в 1 - то обращения в пространство конфигурации обрабатываются аппаратно, встроенным RMAP контроллером		0x1
4:1	CHANNELS_M ODE	Поле режима функционирования остальных каналов DMA. І разряд данного поля соответствует І+1 каналу DMA. Если данный разряд установлен в 0, то канал функционирует в режиме обработки RMAP пакетов, если в 1 - то канал функционирует в режиме обработки SpW пакетов	RD WR	0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Исходное состояние
8:5	CHANNELS_TR ANSPORT_MO	Поле режима Транспортного протокола для остальных каналов DMA. І разряд данного поля соответствует І+1 каналу DMA. Если данный разряд установлен в 0, то канал функционирует в режиме обработки RMAP пакетов, если в 1 - зарезервировано для дальнейших применений. Должен быть установлен в «0»	RD WR	0x0
12:9	DDR	Поле разрешения поддержки расширенной RMAP адресации для остальных каналов DMA. I разряд данного поля соответствует I+1 каналу DMA. Если данный разряд установлен в 0, то канал не поддерживает расширенную RMAP адресацию, если в 1 - поддерживает	RD WR	0x0
14:13	PAGE_SIZE	Поле размера страницы NAND Flash. Если данное поле имеет значение 0, то размер страницы -2 КБайт, 1 - 4 Кбайт, 2 - 8 Кбайт	RD WR	0x0
19:15	RMAP_MAX_SI ZE	Поле максимально допустимого размера поля данных RMAP пакета. Максимально допустимый размер поля данных RMAP пакета определяется на базе значения, заданного в этом поле следующим образом: MAX_SIZE (MBytes) = (Value + 1) *32 . Например, если данное поле установлено в значение 0, то MAX_SIZE = 32 Кбайт, если в значение 1 - 64 Кбайт, 2 - 96 КБайт и т.д. Максимально допустимое значение 1024 Кбайт. Максимально допустимый размер поля данных пакета должен быть кратен размеру страницы NAND Flash	RD WR	0x00
31:20	-	Не используется	RD	0x000

11.4.4 Регистр DMA_CHANNELS_WORK_EN

Назначение разрядов регистра DMA_CHANNELS_WORK_EN приведено в Таблица 11.5.



Таблица 11.5. Назначение разрядов регистра DMA_CHANNELS_WORK_EN

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
4:0	CHANNELS_P1 _WORK_EN	Разрешение функционирования 1 контроллера DMA. І разряд данного поля соответствует І каналу DMA. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 0, то установка данного разряда в 1 соответствует разрешению RMAP команд обращения к flash. Одновременно с этим должен быть установлен в 1 и соответствующий разряд DMA_CHANNELS_WORK_EN.CHANNELS_P2_W ORK_EN. Для того, чтобы осуществлялась обработка этих команд, кроме этого должен быть настроен регион R1 и R4 данного канала DMA - значение DMA_AREA_SIZE_R1.DMA_AREA_SIZE_R1 и DMA_AREA_SIZE_R4.DMA_AREA_SIZE_R4 должно быть отлично от 0. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 1, то установка данного разряда в 1 соответствует разрешению передачи данных, формируемых процессорным ядром в сеть. Для того, чтобы осуществлялась передача данных, кроме этого должны быть настроены регионы R1 и R2 данного канала DMA - значения DMA_AREA_SIZE_R1.DMA_AREA_SIZE_R1 и DMA_AREA_SIZE_R2.DMA_AREA_SIZE_R2 должны быть отличны от 0.	RD WR	0x00



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
	CHANNELS_P2 _WORK_EN	Разрешение функционирования части 2 канала DMA. І разряд данного поля соответствует І каналу DMA. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 0, то установка данного разряда в 1 соответствует разрешению RMAP команд обращения flash. Одновременно с этим должен быть установлен в 1 и соответствующий разряд DMA_CHANNELS_WORK_EN.CHANNELS_P1_W ORK_EN. Для того, чтобы осуществлялась обработка этих команд, кроме этого должны быть настроены регионы R1 и R4 данного канала DMA - значения DMA_AREA_SIZE_R1.DMA_AREA_SIZE_R1 и DMA_AREA_SIZE_R4.DMA_AREA_SIZE_R4 должны быть отличны от 0. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 1, то установка данного разряда в 1 соответствует разрешению приема данных для процессорного ядра из сети. Для того, чтобы осуществлялась передача данных, кроме этого должны быть настроены регионы R3 и R4 данного канала DMA - значения DMA_AREA_SIZE_R3.DMA_AREA_SIZE_R3 и DMA_AREA_SIZE_R4.DMA_AREA_SIZE_R3 и DMA_AREA_SIZE_R4.DMA_AREA_SIZE_R4 должны быть отличны от 0.	RD WR	
1 / 111	PENDING_RM AP	Максимально допустимое количество отложенных команд чтения и записи. Максимально допустимое количество отложенных команд определяется по следующей формуле: (PENDING_RMAP + 1)*4. Данное количество является суммарным по всем типам команд (запись, чтение) и по всем каналам DMA.	RD WR	0x0
13	RMAP_SIZE_T EST_MODE	Включение тестового режима для принимаемых RMAP команд. Если данное поле установлено в значение 0, то допустимая длина поля данных RMAP команд определяется в соответствии со значениями DMA_CHANNELS_MODE.PAGE_SIZE и DMA_CHANNELS_MODE.RMAP_MAX_SIZE. Если данное поле установлено в значение 1, то допустимой считается длина поля данных кратная 8 байтам и >0. Данный режим предназначен для использования в тестовых целях.	RD WR	OXO
31:14	-	Не используется	RD	0x00000

DMA_RMAP_CONFIGURATION



11.4.5 Регистр DMA RMAP_CONFIGURATION

Назначение разрядов регистра DMA_RMAP_CONFIGURATION приведено в Таблица 11.6.

Таблица 11.6. Назначение разрядов регистра DMA_RMAP_CONFIGURATION

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
7:0	DEST_KEY	Значение dest key (гтар). При приеме очередного RMAP пакета данное значение сравнивается со значением соответствующего поля в заголовке и в случае несовпадения пакет отбрасывается.	RD WR	0x00
15:8	TARGET_LA1	Значение target la 1 (rmap). При приеме очередного RMAP пакета данное значение, а так же значения DMA_RMAP_CONFIGURATION.TARGET_LA2 и DMA_RMAP_CONFIGURATION.TARGET_LA3 сравниваются со значением соответствующего поля в заголовке и в случае несовпадения ни с одним из этих значений пакет отбрасывается.	RD WR	0x00
23:16	TARGET_LA2	Значение target la 2 (rmap).	RD WR	0x00
31:24	TARGET_LA3	Значение target la 3 (rmap).	RD WR	0x00

11.4.6 Регистр DMA REGIONS 12 STATUS

Назначение разрядов регистра DMA_REGIONS_12_STATUS приведено в Таблица 11.7.

Таблица 11.7. Назначение разрядов регистра DMA_REGIONS_12_STATUS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
4:0	REGIONS_1_ST ATUS	Поле состояния областей памяти 1 каналов DMA. І разряд данного поля соответствует І каналу DMA. Если данный разряд установлен в 1, то закончилась область памяти, выделенная для региона 1 соответствующего канала DMA. При установке данного разряда в 1 формируется прерывание для CPU (если оно не замаскировано).	RD	0x000
9:5	REGIONS_2_ST ATUS	Поле состояния областей памяти 2 каналов DMA. І разряд данного поля соответствует І каналу DMA. Если данный разряд установлен в 1, то закончилась область памяти, выделенная для региона 2 соответствующего канала DMA. При установке данного разряда в 1 формируется прерывание для CPU (если оно не замаскировано).	RD	0x000
31:10	-	Не используется	RD	0x0000



11.4.7 Регистр DMA REGIONS 34 STATUS

Назначение разрядов регистра DMA_REGIONS_34_STATUS приведено в Таблица 11.8.

Таблица 11.8. Назначение разрядов регистра DMA REGIONS 34 STATUS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
4:0	REGIONS_3_ST ATUS	Поле состояния областей памяти 3 каналов DMA. І разряд данного поля соответствует І каналу DMA. Если данный разряд установлен в 1, то закончилась область памяти, выделенная для региона 3 соответствующего канала DMA. При установке данного разряда в 1 формируется прерывание для CPU (если оно не замаскировано).	RD	0x000
4.7	REGIONS_4_ST ATUS	Поле состояния областей памяти 4 каналов DMA. І разряд данного поля соответствует І каналу DMA. Если данный разряд установлен в 1, то закончилась область памяти, выделенная для региона 4 соответствующего канала DMA. При установке данного разряда в 1 формируется прерывание для CPU (если оно не замаскировано).	RD	0x000
31:10	-	Не используется	RD	0x0000

11.4.8 Регистр DMA_PACK_STATUS

Назначение разрядов регистра DMA_PACK_STATUS приведено в Таблица 11.9.

Таблица 11.9. Назначение разрядов регистра DMA PACK STATUS

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение	Officatine	доступа	значение
4:0	PACK_REC_FL	Поле флагов приема пакетов, не являющихся RMAP командами. Запись 1 в это поле приводит к его сбросу. І разряд данного поля соответствует І каналу DMA. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 0, то данный разряд не используется. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 1, то установка данного разряда в 1 соответствует приему из сети пакета данных. Если данный разряд устанавливается в 1, то устанавливается прерывание для процессора (если оно не замаскировано).	RD WR	0x00



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
9:5	PACK_RMAP_R _REC_FL	Поле флагов приема пакетов, являющихся RMAP командами чтения. Запись 1 в это поле приводит к его сбросу. І разряд данного поля соответствует І каналу DMA. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 0, то установка данного разряда в 1 соответствует получению RMAP команды чтения flash. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 1, то данный разряд не используется. Если данный разряд устанавливается в 1, то устанавливается прерывание для процессора (если оно не замаскировано).	RD WR	0x00
14:10	PACK_RMAP_ W_REC_FL	Поле флагов приема пакетов, являющихся RMAP командами записи. Запись 1 в это поле приводит к его сбросу. І разряд данного поля соответствует І каналу DMA. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 0, то установка данного разряда в 1 соответствует получению RMAP команды записи flash. Если DMA_CHANNELS_MODE.CHANNELS_MODE для этого канала установлено в значение 1, то данный разряд не используется. Если данный разряд устанавливается в 1, то устанавливается прерывание для процессора (если оно не замаскировано).	RD WR	0x00
31:15	-	Не используется	RD	0x00000

11.4.9 Регистр DMA_IRQ_MASK 1

Назначение разрядов регистра DMA_IRQ_MASK1 приведено в Таблица 11.10.

Таблица 11.10. Назначение разрядов регистра DMA_IRQ_MASK1

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
4:0	REC_PACKET_ MASK	Маска прерываний по приему пакета, не являющегося RMAP командой, каналов DMA. I разряд данного поля соответствует I каналу DMA. Если I разряд установлен в 1 - прерывание замаскировано. В данной версии разряд 0 не используется, должен быть установлен в 1.	RD WR	0x1F
9:5	REC_PACKET_ RMAP_MASK	Маска прерываний по приему пакета, являющегося RMAP командой чтения или записи, каналов DMA. І разряд данного поля соответствует І каналу DMA. Если І разряд установлен в 1 - прерывание замаскировано. В данной версии разряд 0 не используется, должен быть установлен в 1.	RD WR	0x1F



Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:10	-	Не используется	RD	0x000000

11.4.10 Peructp DMA_IRQ_MASK2

Назначение разрядов регистра DMA IRQ MASK2 приведено в Таблица 11.11.

Таблица 11.11. Назначение разрядов регистра DMA_IRQ_MASK2

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
4:0	REGION_1_FINI SH_MASK	Маска прерываний по завершении области памяти 1 каналов DMA. І разряд данного поля соответствует І каналу DMA. Если І разряд установлен в 1 - прерывание замаскировано.	RD WR	0x1F
9:5	REGION_2_FINI SH_MASK	Маска прерываний по завершении области памяти 2 каналов DMA. І разряд данного поля соответствует І каналу DMA. Если І разряд установлен в 1 - прерывание замаскировано.	RD WR	0x1F
14:10	REGION_3_FINI SH_MASK	Маска прерываний по завершении области памяти 3 каналов DMA. І разряд данного поля соответствует І каналу DMA. Если І разряд установлен в 1 - прерывание замаскировано.	RD WR	0x1F
31:11	-	Не используется	RD	0x000000

11.4.11 Регистр DMA_IRQ_MASK3

Назначение разрядов регистра DMA_IRQ_MASK3 приведено в Таблица 11.12.

Таблица 11.12. Назначение разрядов регистра DMA_IRQ_MASK3

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
4:0	SH_MASK	Маска прерываний по завершении области памяти 4 каналов DMA. І разряд данного поля соответствует І каналу DMA. Если І разряд установлен в 1 - прерывание замаскировано.	RD WR	0x1F
5		Маска прерывания по ошибке в дескрипторах ответов на RMAP команды чтения и записи. Если разряд установлен в 1 - прерывание замаскировано.	RD WR	0x1
6	ANS_MASK	Маска прерываний по ошибочной попытке записи в полное FIFO дескрипторов ответов на RMAP команды чтения и записи. Если разряд установлен в 1 - прерывание замаскировано.	RD WR	0x1
7	READ_REJ_MA SK	Маска прерываний по факту отбрасывания RMAP команды чтения. Если разряд установлен в 1 - прерывание замаскировано.	RD WR	0x1
8	WRITE_REJ_M ASK	Маска прерываний по факту отбрасывания RMAP команды записи. Если разряд установлен в 1 - прерывание замаскировано.	RD WR	0x1



Номер разряда	Условное обозначение	Описание		Начальное значение
31:9 -		Не используется	RD	0x000000

11.4.12 Регистр DMA START ADDR R1

Назначение разрядов регистра DMA START ADDR R1 приведено в Таблица 11.13.

Таблица 11.13. Назначение разрядов регистра DMA_START_ADDR_R1

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	DMA_START_A DDR_R1	Поле начального адреса области памяти. Если для текущего VC используется режим обработки RMAP команд к FLASH памяти, то данный регион соответствует области дескрипторов RMAP команд чтения и записи данных. Если для текущего VC используется режим команд от БКУ (или обращений в пространство конфигурации), то данный регион соответствует области данных на передачу в сеть. Значение, записываемое в данное поле, должно быть кратным 8 байтам		0x00000000

11.4.13 Регистр DMA AREA SIZE R1

Назначение разрядов регистра DMA_AREA_SIZE_R1 приведено в Таблица 11.14.

Таблица 11.14. Назначение разрядов регистра DMA_AREA_SIZE_R1

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	DMA_AREA_SI ZE_R1	Поле размера области памяти. Если для текущего VC используется режим обработки RMAP команд к FLASH памяти, то данный регион соответствует области дескрипторов RMAP команд чтения и записи данных. Если для текущего VC используется режим команд от БКУ (или обращений в пространство конфигурации), то данный регион соответствует области данных на передачу в сеть. Если данный регион соответствует области дескрипторов RMAP команд и для данного канала используется режим Extended address, то размер области должен быть кратен 16 байтам. В остальных случаях размер области должен быть кратен 8 байтам	RD WR	0x00000000



11.4.14 Регистр DMA_START_ADDR_R2

Назначение разрядов регистра DMA START ADDR R2 приведено в Таблица 11.15.

Таблица 11.15. Назначение разрядов регистра DMA_START_ADDR_R2

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	DMA_START_A DDR_R2	Поле начального адреса области памяти. Если для текущего VC используется режим обработки RMAP команд к FLASH памяти, то данный регион не используется. Если для текущего VC используется режим команд от БКУ (или обращений в пространство конфигурации), то данный регион соответствует области дескрипторов на передачу в сеть. Значение, записываемое в данное поле, должно быть кратным 8 байтам		0x00000000

11.4.15 Peructp DMA_AREA_SIZE_R2

Назначение разрядов регистра DMA_AREA_SIZE_R2 приведено в Таблица 11.16.

Таблица 11.16. Назначение разрядов регистра DMA_AREA_SIZE_R2

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
	DMA_AREA_SI ZE_R2	Поле размера области памяти. Если для текущего VC используется режим обработки RMAP команд к FLASH памяти, то данный регион не используется. Если для текущего VC используется режим команд от БКУ (или обращений в пространство конфигурации), то данный регион соответствует области дескрипторов на передачу в сеть. Размер области должен быть кратен 8 байтам	RD WR	0x00000000

11.4.16 Регистр DMA START ADDR R3

Назначение разрядов регистра DMA START ADDR R3 приведено в Таблица 11.17.



Таблица 11.17. Назначение разрядов регистра DMA START ADDR R3

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	DMA_START_A DDR_R3	Поле начального адреса области памяти. Если для текущего VC используется режим обработки RMAP команд к FLASH памяти, то данный регион не используется. Если для текущего VC используется режим команд от БКУ (или обращений в пространство конфигурации), то данный регион соответствует области данных на прием из сети. Значение, записываемое в данное поле, должно быть кратным 8 байтам		0x00000000

11.4.17 Регистр DMA AREA SIZE R3

Назначение разрядов регистра DMA_AREA_SIZE_R3 приведено в Таблица 11.18.

Таблица 11.18. Назначение разрядов регистра DMA AREA SIZE R3

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	ZL_K3	Поле размера области памяти. Если для текущего VC используется режим обработки RMAP команд к FLASH памяти, то данный регион не используется. Если для текущего VC используется режим команд от БКУ (или обращений в пространство конфигурации), то данный регион соответствует области данных на прием из сети. Размер области должен быть кратен 8 байтам	RD WR	0x00000000

11.4.18 Регистр DMA_START_ADDR_R4

Назначение разрядов регистра DMA_START_ADDR_R4 приведено в Таблица 11.19.

Таблица 11.19. Назначение разрядов регистра DMA START ADDR R4

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	DMA_START_A DDR_R4	Поле начального адреса области памяти. Если для текущего VC используется режим обработки RMAP команд к FLASH памяти, то данный регион соответствует области данных, поступающих в RMAP командах записи данных. Если для текущего VC используется режим команд от БКУ (или обращений в пространство конфигурации), то данный регион соответствует области дескрипторов на прием из сети. Значение, записываемое в данное поле, должно быть кратным 8 байтам		0x00000000



11.4.19 Регистр DMA_AREA_SIZE_R4

Назначение разрядов регистра DMA_AREA_SIZE_R4 приведено в Таблица 11.20.

Таблица 11.20. Назначение разрядов регистра DMA_AREA_SIZE_R4

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0		Поле размера области памяти. Если для текущего VC используется режим обработки RMAP команд к FLASH памяти, то данный регион соответствует области данных, поступающих в RMAP командах записи данных. Если для текущего VC используется режим команд от БКУ (или обращений в пространство конфигурации), то данный регион соответствует области дескрипторов на прием из сети. Размер области должен быть кратен 8 байтам	RD WR	0x00000000

11.4.20 Peгистр FIFO_ANS

Назначение разрядов регистра FIFO_READ_ANS приведено в Таблица 11.21.

Таблица 11.21. Назначение разрядов регистра FIFO READ ANS

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	FIFO_ANS	Поле декскриптора RMAP команд чтения и записи Данное поле обеспечивает доступ к FIFO дескрипторов ответов на RMAP команды чтения и записи. При записи в этот регистр осуществляется запись очередного слова дескриптора в FIFO	WR	0x00000000

11.4.21 Peructp FIFO_STATUS

Назначение разрядов регистра FIFO STATUS приведено в Таблица 11.22.

Таблица 11.22. Назначение разрядов регистра FIFO STATUS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
0	ANS_EMPTY	Поле флагов empty FIFO ответов на команды чтения и записи RMAP. Если установлен в 1, FIFO ответов на команды чтения и записи пусто.	RD	0x0
1	ANS_FULL	Поле флагов full FIFO ответов на команды чтения и записи RMAP. Если разряд установлен в 1, FIFO ответов на команды чтения и записи полно.	RD	0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
2	ANS_NEXIST	Поле флага записи дескриптора ответа на команды чтения и записи RMAP с несуществующим идентификатором. Запись 1 в это поле приводит к его сбросу. Если в ответном дескрипторе указывается несуществующий идентификатор команды, то данный флаг взводится в 1, сброс флага осуществляется путем записи 1 в этот разряд.	RD WR	0x0
7:3	ANS_NEXIST_V ALUE	Поле значения последного записанного идентификатора, который не существует в списке отложенных команд чтения и записи RMAP.	RD	0x00
8	ANS_DOUBLE	Поле флага записи дескриптора ответа на команды чтения и записи RMAP с дублетным идентификатором. Запись 1 в это поле приводит к его сбросу. Если в ответном дескрипторе указывается дублетный (уже присутствующий в данном FIFO) идентификатор команды, то данный флаг взводится в 1, сброс флага осуществляется путем записи 1 в этот разряд.	RD WR	0x0
13:9	ANS_DOUBLE_ VALUE	Поле значения последнего записанного идентификатора, который дублетный в списке отложенных команд чтения и записи RMAP.	RD	0x00
31:14	-	Не используется	RD	0x00000

11.4.22 Регистр REJ_STATUS

Назначение разрядов регистра REJ_STATUS приведено в Таблица 11.23.

Таблица 11.23. Назначение разрядов регистра REJ_STATUS

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
3:0	READ_REJ_FL	Поле признака отбрасывания команды чтения RMAP. Запись 1 в это поле приводит к его сбросу. І разряд данного поля соответствует І каналу DMA. Данный флаг устанавливается в 1, если в момент, когда на обработке находится максимально допустимое количество RMAP команд, из сети поступает очередная RMAP команда чтения.	RD WR	0x0
7:4	WRITE_REJ_FL	Поле признака отбрасывания команды записи RMAP. Запись 1 в это поле приводит к его сбросу. І разряд данного поля соответствует І каналу DMA. Данный флаг устанавливается в 1, если в момент, когда на обработке находится максимально допустимое количество RMAP команд, из сети поступает очередная RMAP команда записи.	RD WR	0x0
31:8	-	Не используется	RD	0x000000



11.4.23 Peгистр RMAP_REC_PACK_COUNTERS_1

Назначение разрядов регистра RMAP_REC_PACK_COUNTERS_1 приведено в Таблица 11.24.

Таблица 11.24. Назначение разрядов регистра RMAP_REC_PACK_COUNTERS_1

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
5:0		Счётчик принятых по каналу 1 RMAP команд. В данный регистр отображается количество принятых по каналу 1 корректных RMAP пакетов команд чтения и записи. При приеме очередной команды (для команд чтения по факту записи дескриптора команды в память, для команд записи по факту записи дескриптора и поля данных команды в память) значение данного счетчика инкрементируется. Сброс счетчика осуществляется каждый раз в момент его чтения. Если момент чтения совпадает с моментом инкрементирования счетчика, то его значение устанавливается в 1	RD WR	0x00
7:6	-	Не используется	RD	0x0
13:8		Счётчик принятых по каналу 2 RMAP команд. В данный регистр отображается количество принятых по каналу 2 корректных RMAP пакетов команд чтения и записи. При приеме очередной команды (для команд чтения по факту записи дескриптора команды в память, для команд записи по факту записи дескриптора и поля данных команды в память) значение данного счетчика инкрементируется. Сброс счетчика осуществляется каждый раз в момент его чтения. Если момент чтения совпадает с моментом инкрементирования счетчика, то его значение устанавливается в 1	RD WR	0x00
15:14	-	Не используется	RD	0x0
21:16		Счётчик принятых по каналу 3 RMAP команд. В данный регистр отображается количество принятых по каналу 3 корректных RMAP пакетов команд чтения и записи. При приеме очередной команды (для команд чтения по факту записи дескриптора команды в память, для команд записи по факту записи дескриптора и поля данных команды в память) значение данного счетчика инкрементируется. Сброс счетчика осуществляется каждый раз в момент его чтения. Если момент чтения совпадает с моментом инкрементирования счетчика, то его значение устанавливается в 1.	RD WR	0x00
23:22		Не используется	RD	0b0 0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
	RMAP_REC_PA	Счетчик принятых по каналу 4 RMAP команд. В данный регистр отображается количество принятых по каналу 4 корректных RMAP пакетов команд чтения и записи. При приеме очередной команды (для команд чтения по факту записи дескриптора команды в память, для команд записи по факту записи дескриптора и поля данных команды в память) значение данного счетчика инкрементируется. Сброс счетчика осуществляется каждый раз в момент его чтения. Если момент чтения совпадает с моментом инкрементирования счетчика, то его значение устанавливается в 1	RD WR	0x00
31:30	-	Не используется	RD	0x00

11.4.24 Perucтр RMAP_REC_PACK_COUNTERS_2

Назначение разрядов регистра RMAP_REC_PACK_COUNTERS_2 приведено в Таблица 11.25.

Таблица 11.25. Назначение разрядов регистра RMAP_REC_PACK_COUNTERS_2

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
		Счетчик принятых по каналу 5 RMAP команд. В данный регистр отображается количество принятых по каналу 1 корректных RMAP пакетов команд чтения и записи. При приеме очередной команды (для команд чтения по факту записи дескриптора команды в память, для команд записи по факту записи дескриптора и поля данных команды в память) значение данного счетчика инкрементируется. Сброс счетчика осуществляется каждый раз в момент его чтения. Если момент чтения совпадает с моментом инкрементирования счетчика, то его значение устанавливается в 1	RD WR	0x00
7:6	-	Не используется	RD	0x0
		Счетчик принятых по каналу 6 RMAP команд. В данный регистр отображается количество принятых по каналу 2 корректных RMAP пакетов команд чтения и записи. При приеме очередной команды (для команд чтения по факту записи дескриптора команды в память, для команд записи по факту записи дескриптора и поля данных команды в память) значение данного счетчика инкрементируется. Сброс счетчика осуществляется каждый раз в момент его чтения. Если момент чтения совпадает с моментом инкрементирования счетчика, то его значение устанавливается в 1	RD WR	0x00



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
15:14	-	Не используется	RD	0x0
		Счетчик принятых по каналу 7 RMAP команд. В данный регистр отображается количество принятых по каналу 3 корректных RMAP пакетов команд чтения и записи. При приеме очередной команды (для команд чтения по факту записи дескриптора команды в память, для команд записи по факту записи дескриптора и поля данных команды в память) значение данного счетчика инкрементируется. Сброс счетчика осуществляется каждый раз в момент его чтения. Если момент чтения совпадает с моментом инкрементирования счетчика, то его значение устанавливается в 1	RD WR	0x00
23:22		Не используется	RD	0b0 0x0
		Счетчик принятых по каналу 8 RMAP команд. В данный регистр отображается количество принятых по каналу 4 корректных RMAP пакетов команд чтения и записи. При приеме очередной команды (для команд чтения по факту записи дескриптора команды в память, для команд записи по факту записи дескриптора и поля данных команды в память) значение данного счетчика инкрементируется. Сброс счетчика осуществляется каждый раз в момент его чтения. Если момент чтения совпадает с моментом инкрементирования счетчика, то его значение устанавливается в 1	RD WR	0x00
31:30	-	Не используется	RD	0x00

11.4.25 Регистр ID_NET

Назначение разрядов регистра ID_NET приведено в Таблица 11.26.

Таблица 11.26. Назначение разрядов регистра ID_NET

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
5:0	ID_NET	Поле идентификации сетевых линков. і разряд соответствует і порту, если і разряд установлен в 1, то пакеты, рассылаемые широковещательно, в этот порт рассылаться не будут.	RD WR	0x00
31:6	-	Не используется	RD	0x0000000



11.4.26 Регистр МОDE_R

Назначение разрядов регистра МОДЕ_ В приведено в Таблица 11.27.

Таблица 11.27. Назначение разрядов регистра MODE_R

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
8:0	_	Значение коэффициента локальной частоты. Коэффициент локальной частоты должен быть задан равным [локальная частота / 10], например, если локальная частота = 125Мгц, то коэффициент локальной частоты должен быть установлен в значение 12. Локальная частота, это частота, подаваемая на блок SpFR4.	RD WR	0x00D
9	ORK_EN	Разрешение работы Broadcast контроллера Если в данное поле записать значение 1, то Broadcast контроллер будет переведен в состояние сброса, в котором будет находиться до тех пор, пока в данное поле не будет записано значение 0	RD WR	0x0
10	AXI_ENABLED	Разрешение работы блоков сопряжения с интерфейсом AXI_master. В данной версии не используется!!! При установке разряда в 0 - работа блоков сопряжения с AXI_master запрещена. При установке 1 в этот разряд работа разрешена. Разряд оказывает влияние на все каналы ДМА.	RD WR	0x0
11	DMA_CLR_FIF O	Очистка буферов блоков сопряжения с интерфейсом AXI_master. В данной версии не используется!!! Если этот разряд установлен в 0, то все работает в штатном режиме. При установке разряда в 1 выполняется сброс FIFO: указатели всех FIFO устанавливаются в 0, что эквивалентно их опустошению, все блоки выставляют фальшивый сигнал готовности блоку DMA. При этом ДМА соответствующими каналами, может вычитывать бесконечное число недостоверных данных, или записывать бесконечное число данных, которые будут теряться. Данные из/в SpFrx12 не обрабатываются. Разряд оказывает влияние на все каналы ДМА.	RD WR	0x0
12	RISC_D_INT_A CK_REGIMES	Режим обработки кодов распределенных прерываний при работе (приеме/отправке) со стороны встроенного процессорного ядра. Если данное поле установлено в значение '0', то используются 6-и битные коды распределенных прерываний, если данное поле установлено в значение '1', то используются 5-и битные коды распределенных прерываний.	RD WR	0x1
15:13	ES	Кодировка INT при работе (приеме/отправке) со стороны встроенного процессорного ядра.	RD WR	0x4
18:16		Кодировка АСК при работе (приеме/отправке) со стороны встроенного процессорного ядра.	RD WR	0x5



Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:19 -		Не используется	RD	0x0000

11.4.27 Регистр **MODE_R1**

Назначение разрядов регистра MODE R1 приведено в Таблица 11.28.

Таблица 11.28. Назначение разрядов регистра MODE_R1

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
12:0	NGTH	Значение длительности окна смены приоритетов при арбитраже. Длительность окна смены приоритетов задается в количестве фаз запросов. Продолжительность одной фазы запроса составляет 3 такта локальной частоты. Период окна смены приоритетов начинает отсчитываться с момента очередной смены приоритетов портов. Если порт, имеющий в текущий момент времени наивысший приоритет, до истечения периода окна смены приоритетов не выставил запрос на передачу пакета, то по истечении периода окна смены приоритетов происходит очередная смена приоритетов портов. Если этот порт выставил запрос на передачу пакета, то смена приоритетов будет осуществлена после предоставления ему гранта.	RD WR	0x0000
13	NDEL_DROPPE D_PACKETS	флаг удаления оборванных пакетов. В данной версии устройства не используется.	RD WR	0x0
14	data_switch_sync _reset	сброс коммутатора данных. Если в данное поле записать значение 0, то на 1 такт будет установлен сигнал синхронного сброса коммутатора данных.	RD WR	0x1
15	DISABLE_ADA PTIVE_PATH_A DDR	отключение адаптивной маршрутизации для путевых адресов. Если в данное поле записать значение 1, то для путевых адресов будет отключена адаптивная групповая маршрутизация (задаваемая регистрами адаптивной маршрутизации).	RD WR	0x0
31:16	-	Не используется	RD	0x0000

11.4.28 Регистр STATE_R

Назначение разрядов регистра STATE R приведено в Таблица 11.29.

Таблица 11.29. Назначение разрядов регистра STATE_R

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
0	GOT_TIME	Признак получения маркера времени (GOT_TIME) Запись 1 в это поле приводит к его сбросу.	RD WR	0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
1	GOT_INT	Признак получения кода распределенного прерывания (GOT_INT) Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
2	GOT_ACK	Признак получения подтверждения (GOT_ACK) Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
3	GOT_C01_COD E	Признак получения кода C01 (GOT_C01_CODE) Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
4	GOT_C11_COD E	Признак получения кода C11 (GOT_C11_CODE) Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
5	INT_RST	Признак получения команды сброса (External reset) от удаленного сетевого администратора. Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
6	GOT_BR	Из сети принят Broadcast SpF. Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
12:7	ARB_TOUT_AD DR_0	последнее значение адреса пакета, для которого истек таймаут арбитража для конф порта.	RD	0x00
31:13	-	Не используется	RD	0x00000

11.4.29 Регистр STATE_R_CONNECT

Назначение разрядов регистра STATE_R_CONNECT приведено в Таблица 11.30.

Таблица 11.30. Назначение разрядов регистра STATE_R_CONNECT

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
3:0	SpW_connected	Признаки соединения по мультипротокольным портам в режиме gigaSpW и SpF. і разряд данного поля установлен в 1, если в текущий момент времени по і внешнему порту установлено соединение	RD	0x0
7:4	DODT CONNEC	Флаги установки соединения по мультипротокольным портам в режиме gigaSpW и SpF. Запись 1 в это поле приводит к его сбросу. і разряд данного поля устанавливается в 1, если по і внешнему порту произошла установка соединения	KD WK	0x0
31:8	-	Не используется	RD	0x000000



11.4.30 Регистр STATE_R_ERROR

Назначение разрядов регистра STATE_R_ERROR приведено в Таблица 11.31.

Таблица 11.31. Назначение разрядов регистра STATE_R_ERROR

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
3:0	SpW_errored	Признаки разрыва соединения по мультипротокольным портам в режиме gigaSpW и SpF. і разряд данного поля установлен в 1, если в текущий момент времени по і порту отсутствует соединение	RD	0xF
7:4	PORT_ERRORE D	Флаги разрыва соединения по мультипротокольным портам в режиме gigaSpW и SpF. Запись 1 в это поле приводит к его сбросу. і разряд данного поля устанавливается в 1, если по і внешнему порту произошел разрыв соединения	KD WK	0x0
31:8	-	Не используется	RD	0x000000

11.4.31 Регистр IRQ MASK

Назначение разрядов регистра IRQ_MASK приведено в Таблица 11.32.

Таблица 11.32. Назначение разрядов регистра IRQ_MASK

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
0		Маска прерывания по факту установки соединения IRQ_CONNECT_MASK. Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
1	CT MASK	Маска прерывания по факту разрыва соединения IRQ_DISCONNECT_MASK. Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
2	IRQ_IA_CODE_	Маска прерывания по коду распределенного прерывания и подтверждения IRQ_CCODE_GLOBAL_MASK. Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
3	IRQ_TIME_MAS K	Маска установки прерывания по управляющему коду при приходе маркера времени IRQ_TIME_MASK. Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
4	IRQ_INT_MASK	Маска установки прерывания по управляющему коду при приходе кода распределенного прерывания IRQ_INT_MASK. Если данный разряд установлен в 0 и RISC_IRQ_MASK.IRQ_IA_CODE_GLOBAL_MA SK='0' то при приходе кода распределенного прерывания прерывание по управляющему коду не будет установлено	RD WR	0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
5	IRQ_ACK_MAS K	Маска установки прерывания по управляющему коду при приходе кода подтверждения IRQ_ACK_MASK Если данный разряд установлен в 0 и RISC_IRQ_MASK.IRQ_IA_CODE_GLOBAL_MA SK='0' то при приходе кода подтверждения прерывание по управляющему коду не будет установлено	RD WR	0x0
6	IRQ_C01_MASK	Маска установки прерывания по управляющему коду при приходе C01 IRQ_C01_MASK. Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
7	IRQ_C11_MASK	Маска установки прерывания по управляющему коду при приходе C11 IRQ_C11_MASK. Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
8	IRQ_INT_RST_ MASK	Маска установки прерывания по приходу команды сброса от удаленного администратора IRQ_INT_RST_MASK Если данный разряд установлен в 0 и RISC_IRQ_MASK.IRQ_IA_CODE_GLOBAL_MA SK='0' то при приходе команды сброса от удаленного администратора прерывание по управляющему коду не будет установлено	RD WR	0x0
9	IRQ_ISR_TOUT S_MASK	Маска установки прерывания по истечении таймаутов Th, Tg, ISR_change IRQ_ISR_TOUTS_MASK Если данный разряд установлен в 0 и RISC_IRQ_MASK.IRQ_IA_CODE_GLOBAL_MA SK='0' то по истечении таймаутов Th, Tg, ISR_change прерывание по управляющему коду не будет установлено	RD WR	0x0
10	IRQ_BR_MASK	Маска установки прерывания по Broadcast SpF IRQ_BR_MASK Если данный разряд установлен в 0 и XXXXXXXXXXXIRQ_CCODE_GLOBAL_MASK= '0', то при приходе Broadcast SpF прерывание по управляющему коду не будет установлено	RD WR	0x0
11	IRQ_TARB_MA SK	Маска прерывания по факту истечения таймаута арбитража Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
12	IRQ_TRES_MAS K	Маска прерывания по факту истечения таймаута ожидания очередного символа при приеме Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
13	IRQ_TSEND_M ASK	Маска прерывания по факту истечения таймаута ожидания очередного символа при передаче Если данный разряд установлен в 0, то прерывание замаскировано	RD WR	0x0
31:14	-	Не используется	RD	0x00000



11.4.32 Регистр GiGA_SPW_STATUS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
0	DC_ERR	Признак ошибки рассоединения. Запись 1 в это поле приводит к его сбросу.	RD WR	x0
1	P_ERR	Признак ошибки паритета. Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
2	ESC_ERR	Признак ошибки escape последовательности. Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
3	CREDIT_ERR	Признак ошибки кредитования. Запись 1 в это поле приводит к его сбросу.	RD WR	0x0
6:4	BDS_STATE	Текущее состояние порта SpaceWire	RD	0x0
31:7	-	Не используется	RD	0x0000000

Назначение разрядов регистра GIGA SPW STATUS приведено в Таблица 11.33.

Таблица 11.33. Назначение разрядов регистра GIGA_SPW_STATUS

11.4.33 Регистр GIGA_SPW_MODE

Назначение разрядов регистра Giga_SPW_MODE приведено в Таблица 11.34.

Таблица 11.34. Назначение разрядов регистра GIGA GIGA MODE

Номер	Условное обозначение	Описание	Тип	Начальное
разряда 0	LINKDISABLED	LinkDisabled	доступа RD WR	значение 0х1
1	AUTOSTART	AutoStart	RDWR	0x0
2	LINKSTART	LinkStart	RD WR	0x1
3	BDS_RESET	Синхронный сброс порта SpW	RD WR	0x1
4	CODEC_LOOPB ACK	CODEC LOOPBACK	RD WR	0x0
5	LVDS_LOOPBA CK	LVDS (PMA) LOOPBACK	RD WR	0x0
6	DY SINGLE	RX SINGLE. В данной версии не используется!!! Если данный разряд установлен в 1, то включен режим RX_SINGLE, если в 0 - то стандартный режим приема данных.	RD WR	0x0
7	TX_SINGLE	TX SINGLE. В данной версии не используется!!! Если данный разряд установлен в 1, то включен режим TX_SINGLE, если в 0 - то стандартный режим передачи данных.	RD WR	0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
16:8	KOEFF_10_LOC AL	Коэффициент локальной частоты. КОЕFF_10_LOCAL * 64 определяет количество таков локальной частоты, равное ResetTime (Временной интервал нахождения в состоянии Reset). КОЕFF_10_LOCAL * 64 должен быть в два раза больше чем время определения ошибки рассоединения, которое равно времени приема DISCONNECT_COUNTER*64 символов. В данной реализации временной интервал установки соединения (wait_conn_time) равен ResetTime*2.	KD WK	0x00D
17	D_INT_ACK_RE GIMES	Режим обработки кодов распределенных прерываний и подтверждений для данного порта. Если данный разряд установлен в 0 - режим 6-и битных кодов, 1 - режим 5-и битных кодов.	RD WR	0x1
20:18	INT_CODES	Кодировка кодов распределенных прерываний в 5-и битном режиме.	RD WR	0x4
23:21	ACK_CODES	Кодировка кодов подтверждений в 5-и битном режиме.	RD WR	0x5
31:24	-	Не используется	RD	0x00

11.4.34 Peructp GIGA SPW_TRANSMISSION_PARAMETERS

Назначение разрядов регистра GIGA_SPW_TRANSMISSION_PARAMETERS приведено в Таблица 11.35.

Таблица 11.35. Назначение разрядов регистра GIGA SPW TRANSMISSION PARAMETERS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
5:0	KOEFF_COMM A	Частота отправки символа СОММА Символ СОММА отправляется в сеть 1 раз в KOEFF_COMMA*8 символов.	RD WR	0x08
9:6	DISCONNECT_ COUNTER	Период определения ошибки рассоединения. DISCONNECT_COUNTER*64 задает количество символов пришедших от сети (поступивших от РМА в цифровую часть) среди которых должен быть хотя бы один символ СОММА. Если среди заданного количества символов нет ни одного символа СОММА, то определяется ошибка рассоединения. DISCONNECT_COUNTER и КОЕFF_COMMA должны быть заданы таким образом, чтобы DISCONNECT_COUNTER*64 был в 2 (и более) раза больше, чем (КОЕFF_COMMA*8).	RD WR	0x4
31:10	-	Не используется	RD	0x000000



11.4.35 Peructp GIGA_SPW_CREDIT_PARAMETERS

Назначение разрядов регистра GIGA_SPW_CREDIT_PARAMETERS приведено в Таблица 11.36.

Таблица 11.36. Назначение разрядов регистра GIGA SPW CREDIT PARAMETERS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
7:0	CREDIT_NUMB ER	Размер кредита, соответствующего одному FCT Количество информационных символов, которое кредитуется одним FCT. При использовании режима управления потоком SpaceWire параметр FCT_COUNT должен быть равен 8. При использовании режима управления потоком GigaSpace-Wire параметр FCT_COUNT должен быть равен 32.	RD WR	0x08
	MBER_FCT	Максимальное значение счетчика кредитования При использовании режима управления потоком SpaceWire параметр MAX_FCT_COUNT должен быть равен 56. При использовании режима управления потоком GigaSpace-Wire параметр MAX_FCT_COUNT должен быть равен 512.	RD WR	0x0038
31:24	TX_SIZE_BUFF ER	Размер буфера кредитования	RD WR	0x10

11.4.36 Регистр РМА_STATUS

Назначение разрядов регистра PMA_STATUS приведено в Таблица 11.37.

Таблица 11.37. Назначение разрядов регистра PMA STATUS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
	ALIGN_STATE	ALIGN_STATE	RD	0x0
5:4	LOCK	LOCK	RD	0x0
6	ALIGN_ERROR	ALIGN_ERROR	RD	0x0
7	OVR	OVR	RD	0x0
8	UNR	UNR	RD	0x0
31:9	-	Не используется	RD	0x000000



11.4.37 Регистр РМА_MODE

Назначение разрядов регистра РМА_МОДЕ приведено в Таблица 11.38.

Таблица 11.38. Назначение разрядов регистра РМА_МОДЕ

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
0	PWDn_TX	PWDn_TX	RD WR	0x0
1	PWDn_RX	PWDn_RX	RD WR	0x0
2	COMMAEN	COMMAEN	RD WR	0x0
3	COMPARE_EN	COMPARE_EN	RD WR	0x0
10:4	PMA_RX_SPEE D	Частота работы РМА_RX: 0x1 - 5 Мбод, 0x2 - 10 Мбод; 0x3 - 15 Мбод; 0x19 - 125 Мбод 0x20 - 312,5 Мбод; 0x40 - 625 Мбод; 0x60 - 1250 Мбод. Поля РМА_RX_SPEED и РМА_TX_SPEED должны иметь одинаковое содержимое	RD WR	0x60
12:11	ALIGN MODE	ALIGN MODE	RD WR	0x1
13	EN PMA RX	EN PMA RX. Разрешение работы PMA на прием	RD WR	0x1
15:14	CDR_MODE	CDR_MODE	RD WR	0x0
16	LB_EN_RX	LB_EN_RX. Разрешение режима Loopback PMA RX	RD WR	0x0
23:17	PMA_TX_SPEE D	Частота работы РМА_ТХ: 0x1 - 5 Мбод, 0x2 - 10 Мбод; 0x3 - 15 Мбод; 0x19 - 125 Мбод 0x20 - 312,5 Мбод; 0x40 - 625 Мбод; 0x60 - 1250 Мбод. Поля РМА_RX_SPEED и РМА_ТХ_SPEED должны иметь одинаковое содержимое	RD WR	0x60
24	EN_PMA_TX	EN_PMA_TX. Разрешение работы РМА на передачу	RD WR	0x1
25	LB_EN_TX	LB_EN_TX. Разрешение режима Loopback PMA TX	RD WR	0x0
31:26	-	Не используется	RD	0x00



11.4.38 Peructp SpF_PORT_MODE_VC_PARAMS

Назначение разрядов регистра SpF_PORT_MODE_VC_PARAMS приведено в Таблица 11.39.

Таблица 11.39. Назначение разрядов регистра SpF_PORT_MODE_VC_PARAMS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
7:0	VC_LNUM	Логический номер виртуального канала.	RD WR	0x
15:8	VC_THROUGHP UT	Доля пропускной способности для данного виртуального канала. Доля пропускной способности задается в процентах. Допустимые значения от 1 до 100.	RD WR	0x0A
18:16	VC_PRIORITY	Уровень приоритета виртуального канала.	RD WR	0x0
19	VC_WORK_EN	Разрешение работы виртуального канала. Если данное поле установлено в значение '0', то работа данного виртуального канала запрещена. Установка логического номера виртуального канала должна осуществляться до разрешения его работы или одновременно с разрешением работы.	RD WR	0x1
20	CREDIT_OVER FLOW	VC: переполнение счетчика кредитов. Данный флаг устанавливается в 1, если при получении очередного FCT инкремент счетчика кредитов не возможен, т.к. он достиг своего максимально возможного значения.	RD	0x0
21	DATA_OVERFL OW	VC: переполнение приемного буфера данных. Данный разряд устанавливается в 1, если прием очередного фрейма данных не возможен, т.к. закончилось место в буфере, т.е. произошла ошибка кредитования.	RD	0x0
31:22	-	Не используется	RD	0x000

11.4.39 Peructp SpF_PORT_MODE_VC_TSLOTS_L

Назначение разрядов регистра SpF_PORT_MODE_VC_TSLOTS_L приведено в Таблица 11.40.

Таблица 11.40. Назначение разрядов регистра SpF PORT MODE VC TSLOTS L

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	VC_TSLOTS_L	Флаги таймслотов от 0 до 31. і разряд этого поля соответствует таймслоту с номером і. Если і разряд установлен в 1, то передача данных в соответствующем таймслоте разрешена. Данный регистр используется для SpF порта и для конфигурационного порта.	RD WR	0xFFFFFFFF



11.4.40 Peructp SpF_PORT_MODE_VC_TSLOTS_H

Назначение разрядов регистра SpF_PORT_MODE_VC_TSLOTS_Н приведено в Таблица 11.41.

Таблица 11.41. Назначение разрядов регистра SpF_PORT_MODE_VC_TSLOTS_H

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
31:0	VC_TSLOTS_H	Флаги таймслотов от 32 до 63. і разряд этого поля соответствует таймслоту с номером і+32. Если і разряд установлен в 1, то передача данных в соответствующем таймслоте разрешена. Данный регистр используется для SpF порта и для конфигурационного порта.	RD WR	0xFFFFFFFF

11.4.41 Peructp SpF_PORT_MODE_VC_FR_LL

Назначение разрядов регистра SpF_PORT_MODE_VC_FR_LL приведено в Таблица 11.42.

Таблица 11.42. Назначение разрядов регистра SpF PORT MODE VC FR LL

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
0	SpFm_PORT_M ODE	SpFm port режим функционирования : 0 - SpF, 1 - gigaSpW	RD WR	0x0
1	SpF_PORT_SYN C_RESET	Синхронный сброс SpF порта.	RD WR	0x0
4:2		Порядковый номер виртуального канала, для которого будет отображаться текущее значение credit_value	RD WR	0x0
5	SCRAMBLER_E N	Если данный разряд установлен в 1, то скрамблинг разрешен	RD WR	0x0
6	DESCRAMBLE R_EN	Если данный разряд установлен в 1, то дескрамблинг разрешен	RD WR	0x0
7	LANESTART	Если данный разряд установлен в 1, то SpF порт выполняет установку соединения в режиме LANE_START.	RD WR	0x0
8	LAUTOSTART	lane: AUTOSTART Если данный разряд установлен в 1, то SpF порт выполняет установку соединения в режиме AUTOSTART	RD WR	0x0
9	NO_SIGNAL	Нет сигнала в линии	RD WR	0x0
10	REMOTE_FLUS H_REQ	Запись 1 в данный разряд приводит к отправке команды REMOTE FLUSH REQ - запроса на сброс соседнего абонента	RD WR	0x0
11	SCRAMBLER_R EQ	Если к моменту начала установки соединения на уровне lane в этот разряд будет записана 1, то в ходе установки соединения в соседний абонент будет отправлен запрос на включение скрамблинга	RD WR	0x0
16:12	OTHER_CAPAB ILITY	lane: OTHER_CAPABILITY	RD WR	0x00



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
17	NUMBER_VALI D_WORDS	lane: NUMBER_VALID_WORDS	RD WR	0x0
25:18	RX_ERR_MAX	lane: RX_ERR_MAX.	RD WR	0xFF
26		Запись 1 в данный разряд приводит к выполнению команды flush для данного порта	RD WR	0x0
27	FL_CRC_MODE _TRANS	Режим подсчета CRC16 для отправляемых фреймов. Если данный разряд установлен в 0, то подсчет CRC отправляемых фреймов осуществляется в соответствии с алгоритмом SpF стандарта, если данный разряд установлен в 1 - то по классической схеме.	RD WR	0x0
28	FL_CRC_MODE _REC	Режим подсчета CRC16 для принимаемых фреймов. Если данный разряд установлен в 0, то подсчет CRC принимаемых фреймов осуществляется в соответствии с алгоритмом SpF стандарта, если данный разряд установлен в 1 - то по классической схеме.	RD WR	0x0
31:29	-	Не используется	RD	0x0

11.4.42 Peгистр SpF_PORT_MODE_LANE_CONSTANTS

Назначение разрядов регистра SpF_PORT_MODE_LANE_CONSTANTS приведено в Таблица 11.43.

Таблица 11.43. Назначение разрядов регистра SpF PORT MODE LANE CONSTANTS

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение	Описание	доступа	значение
15:0	CONST_TIMER	CONST_TIMER	RD WR	0x0F00
31:16	NUM_DO_DEC REASE	NUM_DO_DECREASE	RD WR	0x4000

11.4.43 Peructp SpF_PORT_STATUS_VC1

Назначение разрядов регистра SpF_PORT_STATUS_VC1 приведено в Таблица 11.44.

Таблица 11.44. Назначение разрядов регистра SpF_PORT_STATUS_VC1

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
0	FL_REC_VC_N UM_ERR	VC: флаг ошибки номера виртуального канала для фреймов данных. Данный разряд устанавливается в 1, если был принят фрейм данных с некорректным значением поля номера виртуального канала. (Номером, который не обрабатывается в данном SpFmIC4).	RD	0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
8:1	FL_REC_VC_N	VC: значение некорректного номера виртуального канала для фреймов данных. В данное поле записывается значение последнего принятого некорректного номера виртуального канала для фреймов данных.	RD	0x00
9	FL_FCT_VC_NU M_ERR	VC: флаг ошибки номера виртуального канала для фреймов FCT. Данный разряд устанавливается в 1, если был принят фрейм FCT с некорректным значением поля номера виртуального канала. (Номером, который не обрабатывается в данном SpFmIC4).	RD	0x0
17:10	FL_FCT_VC_ER R_VALUE	VC: значение некорректного номера виртуального канала для фреймов FCT. В данное поле записывается значение последнего принятого некорректного номера виртуального канала для фреймов FCT.	RD	0x00
31:18	-	Не используется	RD	0x0000

11.4.44 Peructp SpF_PORT_STATUS_VC2

Назначение разрядов регистра SpF_PORT_STATUS_VC2 приведено в Таблица 11.45.

Таблица 11.45. Назначение разрядов регистра SpF_PORT_STATUS_VC2

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
3:0	RETRY_COUNT	Количество отправленных команд RETRY. В данном поле указывается количество отправленных на текущий момент времени команд RETRY - количество стартов повторных передач групп фреймов, на которые получен NACK	RD WR	0x0
4	RETRY_COUNT ER_TRANSMIT _OVERFLOW	Признак переполнения счетчика количества отправленных команд RETR	RD	0x0
8:5	RETRY_COUNT ER_RECEIVE	Количество принятых команд RETRY	RD WR	0x0
9	RETRY_COUNT ER_RECEIVE_O VERFLOW		RD	0x0
10	CRC_8_ERROR	Флаг ошибки CRC8 в одном из принятых фреймов. Запись 1 в это поле приводит к его сбросу. Данный флаг устанавливается, если принят фрейм с ошибкой CRC	RD WR	0x0
11	CRC_16_ERROR	Флаг ошибки CRC16 в одном из принятых фреймов. Запись 1 в это поле приводит к его сбросу. Данный флаг устанавливается, если принят фрейм с ошибкой CRC16	RD WR	0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
12		Буфер данных на передачу пуст. Данный флаг устанавливается, если буфер данных на передачу уровня RETRY пуст, т.е. нет отправленных фреймов данных, на которые не получено подтверждение об успешном приеме ACK	RD	0x0
13	DTV	Буфер FCT на передачу пуст. Данный флаг устанавливается, если буфер FCT на передачу уровня RETRY пуст, т.е. нет отправленных фреймов FCT, на которые не получено подтверждение об успешном приеме ACK	RD	0x0
14	RETRY_B_BEM	Буфер BROADCAST на передачу пуст. Данный флаг устанавливается, если буфер BROADCAST на передачу уровня RETRY пуст, т.е. нет отправленных фреймов BROADCAST, на которые не получено подтверждение об успешном приеме ACK	RD	0x0
31:15	-	Не используется	RD	0x00000

11.4.45 Peructp SpF_PORT_CUR_CREDIT_VALUE

Назначение разрядов регистра SpF_PORT_CUR_CREDIT_VALUE приведено в Таблица 11.46.

Таблица 11.46. Назначение разрядов регистра SpF PORT CUR CREDIT VALUE

Номер разряд		Описание	Тип доступа	Начальное значение
31:0	SpF_PORT_CUR _CREDIT_VAL UE	Текущее значение счетчика кредитов. В данное поле отображается текущее значение счетчика кредитов для виртуального канала, порядковый номер которого указан в SpF_PORT_MODE_VC_FR_LL.CUR_CREDIT_V ALUE_NUM	RD	0x00000000

11.4.46 Peгистр SpF_PORT_STATUS_LL_EL

Назначение разрядов регистра SpF_PORT_STATUS_LL_EL приведено в таблице 11.47.

Таблица 11.47. Назначение разрядов регистра SpF PORT STATUS LL EL

Номер	Условное обозначение	Описание	Тип	Начальное значение
разряда			доступа	
1:0	STATE_EL	Состояние машины состояний уровня encoding	RD	0x0
5:2	STATE_LL	Состояние машины состояний уровня lane	RD	0x1
6	LOSS_OF_SIGN AL	Признак потери сигнала	RD	0x0
		Признак переполнения счетчика ошибок при приеме. Запись 1 в это поле приводит к его сбросу	RD WR	0x0
8	STANDBY	Запись 1 в это поле приводит к его сбросу	RD WR	0x0



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
17:9	LOS	Запись 1 в это поле приводит к его сбросу	RD WR	0x000
18	TIMEOUT	Запись 1 в это поле приводит к его сбросу	RD WR	0x0
19	RX_POLARITY	lane : RX_POLARITY	RD	0x0
20	FLUSH_FL	Запись 1 в это поле приводит к его сбросу. Данный флаг устанавливается в 1, если от соседнего абонента получена команда REMOTE_FLUSH.	RD WR	0x0
21	LANE_START_ FL	lane : LANE_START_FL	RD	0x0
22	DESCRAMBLE R_FL	Флаг запроса дескрамблинга. Данное поле устанавливается в 1, если от соседнего абонента в ходе установки соединения на уровне lane получен запрос на выполнение дескрамблинга	RD	0x0
31:23	-	Не используется	RD	0x000

11.4.47 Регистр ADG

Назначение разрядов регистра ADG приведено в таблице 11.48.

Таблица 11.48. Назначение разрядов регистра ADG

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
5:0	ADG	Вектор адаптивной групповой маршрутизации	RD WR	0x
31:6	-	Не используется	RD	0x0000000

11.4.48 Регистр CHANNEL_ROUTING

Назначение разрядов регистра CHANNEL_ROUTING приведено в таблице 11.49.

Таблица 11.49. Назначение разрядов регистра CHANNEL ROUTING

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение		доступа	значение
2.11	CHANNEL_RO UTING	Вектор режима маршрутизации каналов. если в і разряде данного поля установлена 1, то для і порта включен режим маршрутизации каналов: осуществляется передача пакетов в заданный в регистре порт не зависимо от заголовка пакета. В данном устройстве для порта 0 (конфигурационного) данный разряд должен быть установлен в 0!	RD WR	0x00



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
	CHANNEL_RO UTING_L	Если в і разряде данного поля установлена 1, то для і порта включен режим маршрутизации каналов для логических и региональнологических адресов: осуществляется передача пакетов в заданный в регистре порт если заголовок пакета не является путевым адресом В данном устройстве для порта 0 (конфигурационного) данный разряд должен быть установлен в 0!	RD WR	0x00
31:12	-	Не используется	RD	0x00000

11.4.49 Peгистр CHANNEL_REGISTERS

Назначение разрядов регистра CHANNEL_REGISTERS приведено в таблице 11.50.

Таблица 11.50. Назначение разрядов регистра CHANNEL REGISTERS

Номер разряда		Описание	Тип доступа	Начальное значение
113.0	ISTERS	Вектор номеров портов разряды 5:0 используются для задания номера выходного порта, разряды 13:6 - для задания номера VC в выходном порту	RD WR	0x0000
31:14	-	В данной версии не используется	RD	0x00000

11.4.50 Регистр DRT_TOUTS

Назначение разрядов регистра DRT TOUTS приведено в таблице 11.51.

Таблица 11.51. Назначение разрядов регистра DRT TOUTS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
	DRT_GTOUT	Значение глобального периода подсчета таймаутов ожидания приема/передачи. На базе значения, заданного в этом поле, осуществляется подсчет таймаутов ожидания приема/передачи Nchar	RD WR	0x0000
23:16	DRT_LTOUT	Значение локального периода подсчета таймаутов ожидания приема/передачи Это значение используется в счетчиках таймаутов ожидания приема/передачи Nchar. Эти счетчики инкрементируются при достижении счетчиком глобального подсчета таймаутов значения, заданного в DRT_TOUTS.DRT_GTOUT	RD WR	0x00
31:24	-	Не используется	RD	0x00



11.4.51 Регистр DRT_TOUTS_VALIDS

Назначение разрядов регистра DRT TOUTS VALIDS приведено в таблице 11.52.

Таблица 11.52. Назначение разрядов регистра DRT TOUTS VALIDS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
0		Режим подсчета глобального периода подсчета таймаутов ожидания приема/передачи 0 - в тактах локальной частоты работы устройства, 1 - в мкс.	RD WR	0x0
6:1	DK_IOUI_VAL	Разрешение подсчета таймаутов ожидания приема. Если і разряд установлен в 1, то для і порта разрешен подсчет таймаутов приема.	RD WR	0x00
12:7	DI_IOUI_VAL	Разрешение подсчета таймаутов ожидания передачи Если і разряд установлен в 1, то для і порта разрешен подсчет таймаутов передачи.	RD WR	0x00
31:13	-	Не используется	RD	0x00000

11.4.52 Peгистр DRT_TOUTS_FLS

Назначение разрядов регистра DRT TOUTS FLS приведено в таблице 11.53.

Таблица 11.53. Назначение разрядов регистра DRT_TOUTS_FLS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
5:0	DR_TOUT_FL	Признаки истечения таймаутов приема Запись 1 в это поле приводит к его сбросу. і разряд соответствует і порту.	RD WR	0x00
11:6	DT_TOUT_FL	Признаки истечения таймаутов передачи Запись 1 в это поле приводит к его сбросу. і разряд соответствует і порту.	RD WR	0x00
31:12	-	Не используется	RD	0x00000

11.4.53 Регистр DA_TOUTS

Назначение разрядов регистра DA_TOUTS приведено в таблице 11.54.

Таблица 11.54. Назначение разрядов регистра DA TOUTS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
9:0	DA_TOUT	Значение таймаута арбитража пакетов данных Подсчет таймаута арбитража осуществляется в количестве неуспешных фаз запроса выходных портов. Если по истечении заданного в DA_TOUTS.DA_TOUT количества фаз запроса выходных портов требуемые выходные порты еще не были предоставлены, то пакет стирается	RD WR	0x000
31:10	-	Не используется	RD	0x000000



11.4.54 Peructp DA_SPF_TOUTS

Назначение разрядов регистра DA_SPF_TOUTS приведено в таблице 11.55.

Таблица 11.55. Назначение разрядов регистра DA_SPF_TOUTS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
	DA_TOUT_VAL ID	Разрешение счета таймаута арбитража для пакетов, поступающих из данного виртуального канала или порта SpW, gigaSpW	RD WR	0x0
8:1		Значение таймаута арбитража виртуального канала. Подсчет таймаута арбитража виртуального канала осуществляется в количестве неуспешных фаз запроса выходных портов. Если по истечении заданного количества фаз запроса выходных портов требуемые выходные порты еще не были предоставлены, то происходит переход к обслуживанию других виртуальных каналов этого порта. Арбитраж для этого виртуального канала прекращается и будет возобновлен позднее, по истечении таймаута DA SPF TOUTS.ARB COU TOUT	RD WR	0x00
9	ARB_REQ_TOU	Разрешение счета таймаута арбитража для пакетов ,поступающих из данного виртуального канала (для портов SpW, gigaSpW это поле должно быть установлено в 0)	RD WR	0x0
17:10	ARB_COU_TOU T	Значение таймаута возобновления арбитража виртуального канала. Подсчет таймаута возобновления арбитража виртуального канала осуществляется в количестве фаз запроса выходных портов. По истечении данного таймаута арбитраж для данного виртуального канала может быть возобновлен	RD WR	0x00
25:18	IN_INT_QUANTI	Значение таймаута передачи данных по виртуальному каналу. После того, как по каналу будет передано заданное количество символов, передача может быть прервана, если во входном порту готовы данные для передачи в более приоритетных виртуальных каналах	RD WR	0x00
26		Разрешение прерывания передачи данных по виртуальному каналу (для портов SpW, gigaSpW это поле должно быть установлено в 0)	RD WR	0x0
27	BUF_MODE	Режим буферизации. если данный разряд установлен в значение 0 - используется режим коммутации без буферизации, если данный разряд установлен в значение 1 - то режим коммутации с буферизацией		0x0
31:28	-	Не используется	RD	0x0



11.4.55 Регистр DA_SPF_TOUTS2

Назначение разрядов регистра DA_SPF_TOUTS2 приведено в таблице 11.56.

Таблица 11.56. Назначение разрядов регистра DA_SPF_TOUTS2

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
9:0	REC_TRAN_TO UT	Значение таймаута передачи данных по виртуальному каналу. Если передача данных от входного порта к выходным портам не происходила в течение времени этого таймаута (вследствие неготовности очередных данных во входном порту или неготовности выходных портов принять очередные данные), то по истечении таймаута работа канала между входным портом и выходными будет временно остановлена, в дальнейшем она может быть возобновлена.	RD WR	0x000
10	REC_TRAN_TO UT_VALIDS	Разрешение счета таймаута передачи данных по виртуальному каналу (для портов SpW, gigaSpW это поле должно быть установлено в 0)	RD WR	0x0
	REC_TRAN_PE NDING_TOUT	Значение таймаута на который приостанавливаются попытки передать данные по виртуальному каналу. Если передача данных от входного порта к выходным портам была прервана по истечении таймаута DA_SPF_TOUTS2.REC_TRAN_TOUT то по истечении данного таймаута этот виртуальный сможет возобновить попытки передачи данных.	RD WR	0x000
31:21	-	Не используется	RD	0x000

11.4.56 Peгистр DA_SPF_TOUTS_FLS

Назначение разрядов регистра DA_SPF_TOUTS_FLS приведено в таблице 11.57.

Таблица 11.57. Назначение разрядов регистра DA SPF TOUTS FLS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
0	ARB_IOUI_FL	Поле флагов истечения таймаутов арбитража. Запись 1 в это поле приводит к его сбросу. Флаг устанавливается, если для пакета, передаваемого по данному SpW, gigaSpW порту или виртуальному каналу SpF порта истек таймаут арбитража	RD WR	0x0
8:1		Последнее значение адреса пакета, для которого истек таймаут арбитража. В данное поле записывается адрес последнего пакета, для которого истек таймаут арбитража	RD	0x00



Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
	REC_TRAN_TO UT_FL	Поле флагов истечения таймаутов передачи данных по виртуальному каналу. Запись 1 в это поле приводит к его сбросу. Флаг устанавливается, если истек таймаут передачи данных по данному виртуальному каналу.	RD WR	0x0
10	BUF_FULL	Признак того, что буфер виртуального канала (порта) полон	RD	0x0
11	BUF_EMPTY	Признак того, что буфер виртуального канала (порта) пуст	RD	0x1
31:12	-	Не используется	RD	0x00000

11.4.57 Peructp INCORRECT_VC_INFO

Назначение разрядов регистра INCORRECT_VC_INFO приведено в таблице 11.58.

Таблица 11.58. Назначение разрядов регистра INCORRECT_VC_INFO

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
	INCORRECT_V C_FL	Поле флага получения пакета, адресованного в VC с некорректным логическим номером. Запись 1 в это поле приводит к его сбросу. Флаг устанавливается, если поступил пакет, адресованный в порт, в котором отсутствует виртуальный канал с требуемым номером.	RD WR	0x0
x · i	INCORRECT_V C_NUM	Поле номера виртуального канала. В данное поле записывается номер некорректного виртуального канала.	RD	0x00
	INCORRECT_V C_PACKET_HE ADER	Адрес пакета, адресованного в виртуальный канал с некорректным номером. В данное поле записывается адрес пакета.	RD	0x00
31:17	-	Не используется	RD	0x0000



11.4.58 Peructp SPEC_VC_ARB_PARAMETERS

Назначение разрядов регистра SPEC_VC_ARB_PARAMETERS приведено в таблице 11.59.

Таблица 11.59. Назначение разрядов регистра SPEC_VC_ARB_PARAMETERS

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
3:0	VC_NUM_BY_R T	Флаги разрешения определения номера виртуального канала по таблице маршрутизации. Устанавливаются только для мультипротокольных портов в режиме SpF. I разряд данного поля соответствует SpF порту с номером I+2 (номера 0 и 1 соответствуют внутренним портам маршрутизатора). Если разряд установлен в 0, то пакет, приходящий из данного порта в выходном порту (выходных портах) отправляется в VC с тем же логическим номером из которого он поступил. Если данный разряд установлен в 1, то в выходном порту (выходных портах) пакет отправляется в VC с логическим номером, указанным в таблице маршрутизации для его адреса.	RD WR	0x0
	INCORRECT_V C_MULTICAST	Разрешение широковещательной рассылки пакета, если в для части выходных портов VC является не корректным. Если данное поле установлено в 1, то пакет, для которого в части выходных портов есть требуемый VC, а в части он отсутствует, будет рассылаться в те выходные порты, для которых VC существует.		0x0
31:5	-	Не используется	RD	0x0000000

11.4.59 Регистр SPEC_ARB

Назначение разрядов регистра SPEC ARB приведено в таблице 11.60.

Таблица 11.60. Назначение разрядов регистра SPEC ARB

Номер	Условное	Описание	Тип	Начальное
разряда	обозначение	Januarine		значение
5:0	SPEC_ARB	Поле специальных условий арбитража	RD WR	0x00
31:6	-	Не используется	RD	0x0000000



11.4.60 Peructp CONTROL_CONNECTION

Назначение разрядов регистра CONTROL CONNECTION приведено в таблице 11.61.

Таблица 11.61. Назначение разрядов регистра CONTROL_CONNECTION

Номер разряда	Условное обозначение	Описание	Тип доступа	Начальное значение
5:0	CONN_HOLD_T OUT	Таймаут нахождения в состоянии run Данный таймаут начинает отсчитываться после перехода порта в состояние run и установки базовой скорости. Если до истечения этого таймаута не произошел разрыв соединения, то считается, что соединение установилось (позволяет отследить ситуации дребезга - когда происходит переход в состояние run, но через небольшой период времени соединение разрывается).	RD WR	0x0A
6		Режим отсчета таймаута CONTROL_CONNECTION.CONN_HOLD_TOUT. Если данное поле установлено в значение '0', то отсчет таймаута нахождения в состоянии run осуществляется в тактах, если в '1' - то в мкс	RD WR	0x0
17:7	UNCONNECT_T OUT	Таймаут отсутствия соединения (для портов SpW и мультипротокольных портов в режиме gigaSpW) Данный таймаут начинает отсчитываться, если происходит разрыв соединения. Если происходит восстановление соединения (и, если CONTROL_CONNECTION.CONN_HOLD_TOUT >0 оно сохраняется в течении заданного времени), то счет этого таймаута сбрасывается.	RD WR	0x001
18	UNCONNECT_T OUT_REGIME	Режим отсчета таймаута CONTROL_CONNECTION.UNCONNECT_TOUT. Если данное поле установлено в значение '0', то отсчет таймаута нахождения в состоянии run осуществляется в тактах, если в '1' - то в мкс	RD WR	0x0
19		соединения. Если данное поле установлено в значение '1', то по истечении таймаута отсутствия соединения пакеты будут стираться.	RD WR	0x1
31:20	-	Не используется	RD	0x000

11.5 Описание логики работы контроллеров транспортного уровня, DMA коммутатора

Коммутатор SpFR подключен к AXI через две 64-х разрядные точки подключения AXI-master.

Управление обменом данных по ним осуществляется блоками DMA, входящими в состав Коммутатора SpFR.

Расчет количества каналов DMA



В каждом SpFi порту должен быть виртуальный канал с номером 0 для поддержки обращений в пространство конфигурации. Остальные виртуальные каналы могут быть использованы для поддержки любых других обращений (команд БКУ или пакетов Транспортных протоколов). В соответствии с этим необходимое количество каналов DMA может быть определено по следующей формуле:

$$1 + \sum_{i=1}^{Np} (NVC_i - 1)$$

Np – количество внешних портов SpFR

NVCi – количество виртуальных каналов в *i*-ом внешнем порту SpFR

Если в двух портах 4 VC и в двух 2 VC, то суммарное количество поддерживаемых виртуальных каналов:

$$1 + 3*2 + 1*2 = 9 \text{ VC}$$

В соответствии с этим в блоках DMA суммарно поддерживается 9 независимых каналов.

Канал DMA с номером 0 предназначен для поддержки обращений в пространство конфигурации.

Остальные каналы DMA обеспечивают возможность параллельного выполнения транзакций записи /чтения в соответствии с пакетами, приходящими из остальных виртуальных каналов SpFi портов (и пакетами, отправляемыми в остальные виртуальные каналы SpFi портов).

Для сокращения количества настроек, и количество регистров настроек соответственно и упрощения процесса программирования в этом проекте используется прямое отображение виртуальных каналов портов SpFi на каналы DMA.

Каждый виртуальный канал порта SpFi имеет физический номер – порядковый номер в рамках массива виртуальных каналов данного порта. Например, если в порту поддерживается 4 VC, то они будут иметь физические номера от 0 до 3 соответственно.

В соответствии со стандартом SpFi для каждого виртуального канала может быть задан логический номер в диапазоне от 0 до 255. В рамках одного порта SpFi все виртуальные каналы должны иметь уникальные логические номера. Логический номер может совпадать или не совпадать с физическим номером. Для упрощения процесса настроек в данном проекте логический номер 0 в каждом порту будет присвоен виртуальному каналу с физическим номером 0. Виртуальным каналам с остальными физическими номерами могут быть присвоены логические номера в диапазоне от 1 до 255.



Отображение виртуальных каналов портов SpFi на каналы DMA осуществляется по следующим правилам.

Между каналами DMA и виртуальными каналами SpFi портов существует взаимнооднозначное соответствие по порядку следования их номеров.

Первый канал DMA соответствует VC с физическим номером 1 в SpFi порту с номером 1. Второй канал DMA соответствует VC с физическим номером 2 в SpFi порту с номером 1, и т.д. Это отображение иллюстрируется на Рисунок 11.1.

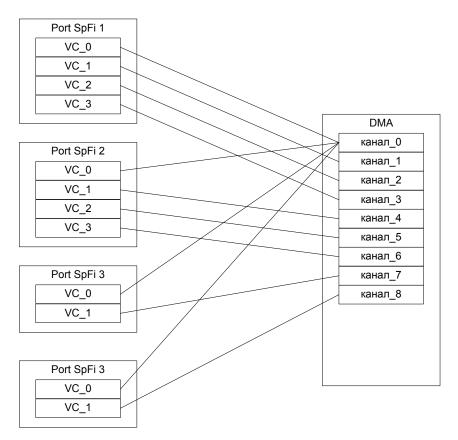


Рисунок 11.1. Отображение виртуальных каналов портов SpFi на каналы DMA

Прием пакетов из сети

Прием пакетов из сети и запись их в память по соответствующему VC возможен при наличии настройки и наличии разрешения работы соответствующего канала DMA на прием. Формат записи пакетов в память (пакет Транспортных протоколов или «обычный» SpW пакет) определяется настройками соответствующего канала DMA. При приеме очередного пакета из сети или при заполнении одной из областей памяти, выделенных для канала DMA, блок SpFR устанавливает сигнал прерывания для процессора.

Передача пакетов в сеть



Выборка данных из памяти и передача пакетов в сеть осуществляется по тем VC, для которых имеются настройки соответствующих каналов DMA и имеется разрешение работы этих каналов DMA.

Способ интерпретации выбираемых из памяти данных (формат формируемых пакетов RMAP для flash или «обычный» SpW пакет) определяется настройками соответствующего канала DMA.

Если канал DMA настроен на обработку RMAP команд для flash, то процессор должен выставлять разрешение работы после того, как процессором завершена обработка команды (например, обмен данными с flash) и в области памяти, соответствующие этому каналу DMA, записаны данные (при успешном выполнении команды чтения) и статус выполнения команды. (При выполнении команды записи и при неуспешном выполнении команды чтения записывается только статус выполнения команды.)

Если канал DMA настроен на обработку «обычных» пакетов SpW, то процессор должен выставлять разрешение работы после того, как в соответствующие этому каналу области памяти записаны содержимое и дескриптор пакета.

При приеме пакетов из сети контроллер транспортного уровня разделяет поток пакетовобращений в пространство конфигурации, поток пакетов-обращений в память и поток «обычных» пакетов SpW. Далее, в соответствии с типом потока и номером виртуального канала, из которого поступил пакет (фрагмент пакета), DMA определяет адрес памяти, по которому он должен быть записан и передает его в коммутатор АХІ.

При передаче пакетов в сеть контроллер транспортного уровня в соответствии с имеющимися настройками областей памяти и виртуальных каналов определяет очередность выборки данных из различных областей памяти. Обрабатываются только потоки от виртуальных каналов, для которых выполнены настройки соответствующих каналов DMA и дано разрешение их работы. Среди каналов DMA очередность работы определяется в соответствии с приоритетами VC, определенной для них пропускной способностью, и перечнем таймслотов, в которых для них разрешена работа.

Настройки областей памяти.

Для каждого виртуального канала могут быть настроены 4 области памяти.

Назначение этих областей памяти различно в зависимости от режима функционирования данного канала DMA.

Если канал DMA предназначен для обращений в пространство конфигурации и команд SpW, то области имеют следующее назначение:

- область данных на передачу в сеть;
- область дескрипторов на передачу пакета в сеть;



- область данных на прием из сети;
- область дескрипторов на прием пакетов из сети.

Если канал DMA предназначен для обработки пакетов Транспортных протоколов, то области памяти имеют следующее назначение:

- область дескрипторов RMAP команд чтения данных;
- область дескрипторов RMAP команд записи данных;
- область данных, поступающих в RMAP командах записи данных.

11.6 Конфигурирование областей памяти для виртуальных каналов, предназначенных для обращений в пространство конфигурации или команд SpW

Если данный виртуальный канал предназначен для обработки обращений в пространство конфигурации или команд SpW, то области памяти имеют следующее назначение:

- область данных на передачу в сеть;
- область дескрипторов на передачу в сеть;
- область данных на прием из сети;
- область дескрипторов на прием из сети.

Потенциально по такой схеме могут быть записаны и RMAP команды и пакеты SpW любого другого формата. Для RMAP пакетов, которые будут записываться в эти области памяти, можно выравнивать поля данных, по словам памяти (если длина RMAP заголовка оказалась не кратной 8 байтам, то поле данных все равно начинать записывать в новое слово памяти).

При завершении одной из выделенных областей памяти, при приеме пакета, при завершении передачи пакета DMA формирует прерывание для процессора.

Формат младшего слова дескриптора пакета (передаваемого и принятого) приведен в Таблина 11.62.

Таблица 11.62. Формат младшего слова дескриптора пакета (передаваемого и принятого)

Номер разряда	Описание
31	Признак заполнения дескриптора действительными данными. Учитывается только при приёме пакетов (позволяет процессору идентифицировать конец очереди дескрипторов в памяти). При передаче пакетов этот разряд не учитывается (DMA вычитывает всю область дескрипторов, заданную процессором). До запуска приёма, все 31-е разряды дескрипторов области приёма должны быть обнулены программно; DMA не обнуляет 31-е разряды не принятых дескрипторов, DMA только записывает «1» в 31-е разряды принятых дескрипторов
30:29	Тип конца пакета (01 – EOP; 10 – EEP)
28:26	Не используются



Номер разряда	Описание
25:0	Размер пакета в байтах

Формат старшего слова дескриптора передаваемого пакета показан в Таблица 11.63.

Таблица 11.63. Формат старшего слова дескриптора передаваемого пакета

Номер разряда	Описание
63:45	Не используются
44:37	Не используются
36:32	Маска портов, в которые надо передать пакет, для того, чтобы обеспечить возможность широковещательной рассылки маска задается в унарной кодировке – каждый разряд маски соответствует одному порту

Формат старшего слова дескриптора принятого пакета показан в Таблица 11.64.

Таблица 11.64. Формат старшего слова дескриптора принятого пакета

Номер разряда	Описание
63:48	Не используются
47:40	Номер виртуального канала
34:32	Номер порта, из которого принят пакет

Пример расположения пакетов и дескрипторов представлен на Рисунок 11.2. Пусть в системной памяти находится 3 пакета принятых из сети. Первый пакет принят из порта SpF 2, имеет размер 10 байт и заканчивается символом ЕОР. Второй пакет принят из порта SpF 3, имеет размер 8 байт и заканчивается символом ЕЕР. Третий пакет принят из порта SpF 1, имеет размер 11 байт и заканчивается символом ЕОР. Данные из этих пакетов хранятся в области памяти, выделенной процессором для приема данных из сети. Первый и третий пакет дополнены 6-ю и 5-ю байтами соответственно, для выравнивания по границам 64-разрядных слов.

Дескрипторы пакетов хранятся в области дескрипторов пакетов, принятых из сети. В дескрипторе первого пакета указан номер порта, из которого он принят -2, признак действительности дескриптора =1, тип конца пакета =0b'01-EOP, размер пакета 0x00000A. В дескрипторе второго пакета указан номер порта, из которого он принят -3, признак действительности дескриптора =1, тип конца пакета =0b'10-EEP, размер пакета 0x000008. В дескрипторе третьего пакета указан номер порта, из которого он принят -1, признак действительности дескриптора =1, тип конца пакета =0b'01-EOP, размер пакета 0x00000B.

Пусть от процессора должно быть отправлено 3 пакета. Пусть первый из них должен быть отправлен в порт SpF 2, имеет длину А. Пусть второй из них должен быть отправлен в порт SpF 1,2 (широковешательно), имеет длину В. Пусть третий должен быть отправлен в порт SpF 2,3 (широковешательно), имеет длину 8.



Поля данных пакетов расположены в области данных на передачу в сеть. Для каждого из пакетов выполнено выравнивание по границам слов памяти. Дескрипторы пакетов расположены в области дескрипторов на передачу в сеть.

В дескрипторе первого пакета указано 0x4 - в унарном коде это соответствует передаче в SpF 2, признак действительности дескриптора установлен в 1, тип конца пакета =0b'01 – EOP, размер пакета 0x00000A.

В дескрипторе второго пакета указано 0x6 - в унарном коде это соответствует передаче в SpF 1,2, признак действительности дескриптора установлен в 1, тип конца пакета =0b'01 – EOP, размер пакета 0x00000B.

В дескрипторе третьего пакета указано 0xC - в унарном коде это соответствует передаче в SpF 2,3, признак действительности дескриптора установлен в 1, тип конца пакета =0b'01 – EOP, размер пакета 0x000008.



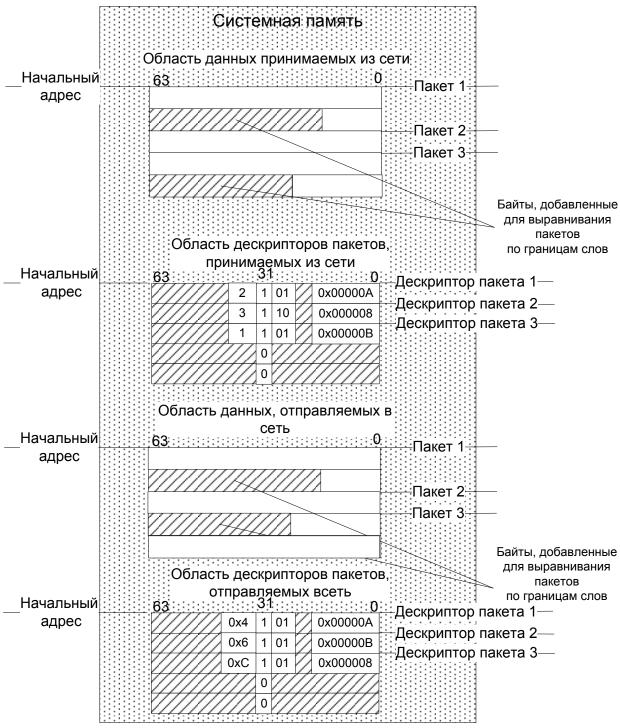


Рисунок 11.2. Расположение пакетов и дескрипторов (для обработки обращений в пространство конфигурации или команд SpW)



11.7 Конфигурирование областей памяти для виртуальных каналов, предназначенных для обработки пакетов Транспортных протоколов

Если данный виртуальный канал предназначен для обработки пакетов Транспортных протоколов памяти, то области памяти имеют следующее назначение:

- область дескрипторов RMAP команд чтения данных;
- область дескрипторов RMAP команд записи данных;
- область данных, поступающих в RMAP командах записи данных.

При завершении одной из выделенных областей памяти, при приеме очередной RMAP команды, при завершении передачи пакета, соответствующего ответу на очередную RMAP команду, DMA формирует прерывание для процессора.

При приеме RMAP команды чтения дескриптор команды записывается в соответствующую область памяти данного виртуального канала.

После того, как команда чтения выполнена, дескриптор ответа помещается процессором в FIFO, программно доступное через регистр FIFO дескрипторов ответов чтения (FIFO_READ_ANS). В дескрипторе ответа содержится начальный адрес области, в которой расположены прочитанные данные. После отправки в сеть ответного пакета формируется прерывание для процессора. Данное FIFO является общим для всех виртуальных каналов. Номер виртуального канала и номер выходного порта для отправки пакета-ответа на команду чтения определяется автоматически по уникальному идентификатору команды, расположенному в дескрипторе. Эта информация хранится в служебном FIFO вместе с адресом, по которому должен быть отправлен ответ и другими атрибутами команды, необходимыми для формирования ответа.

При приеме RMAP команды записи дескриптор команды записывается в соответствующую область памяти данного виртуального канала, данные, поступившие в команде, записываются в соответствующую область данных виртуального канала.

После выполнения RMAP команды записи дескриптор со статусной информацией помещается процессором в FIFO, программно доступное через регистр FIFO дескрипторов статуса выполнения команд записи (FIFO_WRITE_ANS). Данное FIFO является общим для всех виртуальных каналов. Номер виртуального канала и номер выходного порта для отправки пакета-ответа на команду записи определяется автоматически по уникальному идентификатору команды, расположенному в дескрипторе. Эта информация хранится в служебном FIFO вместе с адресом, по которому должен быть отправлен ответ и другими атрибутами команды, необходимыми для формирования ответа.

Размер всех FIFO равен 16 дескрипторам (максимально допустимое количество отложенных команд).



Формат дескриптора RMAP команды чтения показан в Таблица 11.65, Таблица 11.66.

Таблица 11.65. Формат дескриптора RMAP команды чтения (первое слово)

Номер разряда	Описание
63:32	Адрес обращения (берется из RMAP команды)
31	Признак активности дескриптора (признак заполнения дескриптора действительными данными).
30	Не используется
29:26	Идентификатор команды чтения (нумерация выполняется циклически)
23:0	Размер области данных, которая должна быть прочитана (берется из RMAP команды)

Таблица 11.66. Формат дескриптора RMAP команды чтения (второе слово)

Номер разряда	Описание
63:8	Не используется
7:0	Extended address (берется из RMAP команды)

В зависимости от режима (с использованием Extended Address или без него) дескриптор может быть либо со вторым словом, либо без него (см. регистр DMA_CHANNELS_MODE).

Формат дескриптора RMAP команды записи показан в Таблица 11.67, Таблица 11.68.

Таблица 11.67. Формат дескриптора RMAP команды записи (первое слово)

Номер разряда	Описание
63:32	Адрес обращения (берется из RMAP команды)
31	Признак активности дескриптора (признак заполнения дескриптора действительными данными).
20	
30	Не используется
29:26	Идентификатор команды записи
23:0	Размер области данных, которая должна быть записана (берется из RMAP
	команды)

Таблица 11.68. Формат дескриптора RMAP команды записи (второе слово)

Номер бита	Описание
63:8	Не используется
7:0	Extended address (берется из RMAP команды)

Формат дескриптора данных, прочитанных по команде RMAP, дескриптора информации о статусе выполнения команды записи RMAP показан в Таблица 11.69.

Таблица 11.69. Формат дескриптора данных, прочитанных по команде RMAP, дескриптора о статусе выполнения команды записи RMAP

Номер разряда	Описание
63:32	Начальный адрес блока данных, прочитанных по команде чтения (для
	дескрипторов ответов на команду записи не используется)
31	Тип команды:
	0 – команда чтения;
	1 – команда записи.



Номер разряда	Описание
30	Статус выполнения команды (записи или чтения):
	0 – выполнена успешно;
	1 – не выполнена
29:26	Идентификатор команды чтения / команды записи
23:0	Размер области данных, которая прочитана (только для дескрипторов,
	соответствующих данным, прочитанным по командам чтения)

Пример расположения пакетов и дескрипторов представлен на Рисунок 11.3.

Пусть в системной памяти находится 2 RMAP команды чтения и 2 RMAP команды записи. Первая RMAP команда чтения имеет идентификатор 5 и требует прочтения 0х4 байтов данных. Вторая RMAP команда чтения имеет идентификатор 6 и требует прочтения 0х8 байтов данных. Первая RMAP команда записи имеет идентификатор 2 и требует записи 8 байтов данных. Эти байты имеют значение 0х0000000000000057 (они расположены в области данных, поступающих в RMAP командах записи данных). Вторая RMAP команда имеет идентификатор 3 и требует записи 4 байтов данных. В данном случае для выравнивания в области данных, поступающих в RMAP командах записи данных, добавлены байты, поскольку размер поля данных не кратен 8-и байтам.

Предположим, что обе команды чтения были выполнены успешно, команда записи 2 (с идентификатором 3) выполнена успешно, а команду записи 1 (с идентификатором 2) выполнить не удалось.

Причем сначала была выполнено обращение во flash память, соответствующее команде чтения 1, потом команде записи 2, потом команде записи 1, потом команде чтения 2.

В результате в FIFO_READ_ANS была записана следующая последовательность дескрипторов

- дескриптор команды чтения 1: 0 признак команды чтения, 0 статус успешного выполнения, 5 идентификатор команды чтения, 0х4 длина поля данных, ADDR_1 начальный адрес поля данных, прочитанных по этой команде;
- дескриптор команды чтения 2: 0 признак команды чтения, 0 статус успешного выполнения, 6 идентификатор команды чтения, 0х8 длина поля данных, ADDR 2 начальный адрес поля данных, прочитанных по этой команде.

В некотором месте памяти (ADDR_1) были расположены данные, прочитанные по команде чтения 1:

0х32, добавлено 4 байта данных для выравнивания

В некотором месте памяти (ADDR_2) были расположены данные, прочитанные по команде чтения 2:

0x57, 0x30



B FIFO WRITE ANS была записана следующая последовательность дескрипторов:

- дескриптор команды записи 2: 1 признак команды записи, 0 статус успешного выполнения, 3 идентификатор команды записи;
- дескриптор команды записи 1: 1 признак команды записи, 1 статус не выполнения, 2 идентификатор команды записи.

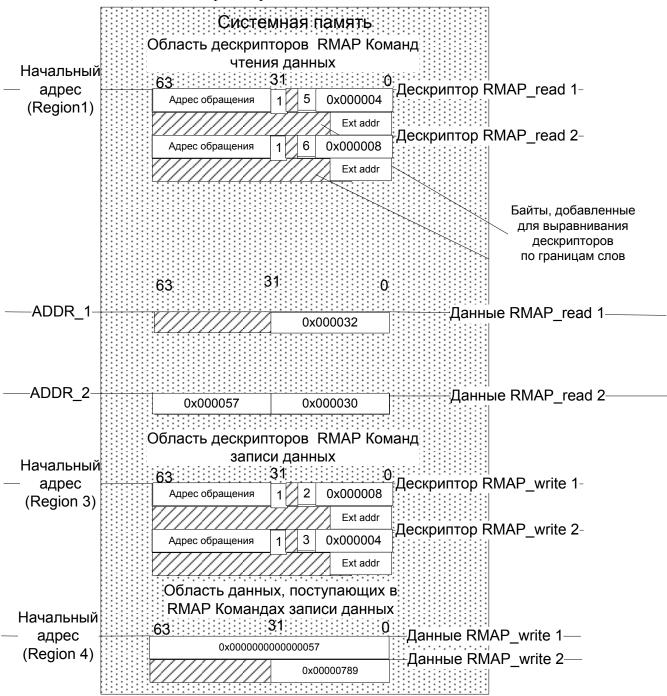


Рисунок 11.3. Расположение пакетов и дескрипторов (для обработки RMAP команд к flash памяти, то области памяти имеют следующее назначение)



11.8 Перечень прерываний, формируемых коммутатором SpFR

Перечень прерываний, формируемых коммутатором SpFR, приведен в Таблица 11.70.

Таблица 11.70. Перечень прерываний, формируемых SpFR

Номер	Название	Описание
0	IRQ_CONNECT	Прерывание по установке соединения в каком-либо из портов SpFR. Для определения, по какому именно порту произошла установка соединения, необходимо прочитать регистр STATE_R_CONNECT. В поле PORT_CONNECTED находятся флаги установки соединения. Флаг PORT_CONNECTED(I) устанавливается в 1, если в порту I произошла установка соединения. Для сброса соответствующего флага необходимо записать 1 в I разряд. При формировании прерывания IRQ_CONNECT флаги PORT_CONNECTED(I) объединяются по логике «ИЛИ». Для маскирования прерывания IRQ_CONNECT имеется поле IRQ_CONNECT_MASK в регистре CPU_IRQ_MASK
1	IRQ_DISCONNECT	Прерывание по разрыву соединения, в каком либо из портов SpFR. Для определения, по какому именно порту произошел разрыв соединения, необходимо прочитать регистр STATE_R_ERROR. В поле PORT_ERRORED находятся флаги разрыва соединения. Флаг PORT_ERRORED(I) устанавливается в 1, если в порту I произошел разрыв соединения и при этом для порта не запрещена установка соединения (для режима gigaSpW SPW_GIGA_MODE.LINKDISABLED = 0, для режима SpFi (SpF_PORT_MODE_VC_FR_LL.LANESTART=1) (SpF_PORT_MODE_VC_FR_LL.LANESTART=1) (SpF_PORT_MODE_VC_FR_LL.LAUTOSTART=1)). Для сброса флага необходимо записать 1 в I разряд. При формировании прерывания IRQ_DISCONNECT флаги PORT_ERRORED объединяются по логике «ИЛИ». Для маскирования прерывания IRQ_DISCONNECT имеется поле IRQ_DISCONNECT_MASK в регистре CPU_IRQ_MASK
2	IRQ_TCODE	Прерывание по приему корректного маркера времени из любого порта SpFR. Маркер времени считается корректным, если его значение на 1 больше предыдущего принятого маркера времени (по модулю 64). Значение маркера времени может быть прочитано из поля TRUE_TIME регистра CUR_TIME. Для того, чтобы сбросить данное прерывание, необходимо записать 1 в разряд GOT_TIME регистра STATE_R. Для того, чтобы замаскировать данное прерывание необходимо записать 0 в разряд IRQ_TIME_MASK регистра CPU_IRQ_MASK. По умолчанию данное прерывание замаскировано



Номер	Название	Описание		
3	IRQ_INT_ACK	Прерывание по факту:		
		приема кода распределенного прерывания;		
		приема кода подтверждения;		
		приема команды сброса от удаленного администратора;		
		истечения одного из таймаутов, заданных для распределенных		
		прерываний.		
		Для того, чтобы определить какая из этих причин привела к		
		установке прерывания необходимо прочитать содержимое регистра STATE_R: если STATE_R.GOT_INT = 1 – был принят код		
		распределенного прерывания, если STATE_R.GOT_ ACK = 1 – был принят код подтверждения, если STATE R.INT RST=1 была		
		принята команда сброса от удаленного администратора. Если ни		
		один из этих флагов не установлен, значит, причиной прерывания		
		является истечение таймаутов.		
		Для того, чтобы сбросить прерывание по причине, не связанной с		
		таймаутами, необходимо записать «1» в соответствующий разряд		
		регистра STATE_R.		
		Для того, чтобы определить, номер прерывания, для которого		
		сработал таймаут, необходимо прочитать регистры		
		ISR_TOUTS_FLS_L, ISR_TOUTS_FLS Для сброса прерывания по		
		факту истечения таймаута необходимо записать «1» в		
		установленные в 1 разряды этого регистра.		
		Для того, чтобы определить, какой был принят код		
		распределенного прерывания или подтверждения необходимо		
		выполнить чтение регистра ISR_L, ISR_H и сравнить его со		
		значением, полученным в результате предыдущего чтения. Для маскирования прерывания необходимо в регистре		
		СРU IRQ MASK установить в «0» разряд:		
		IRQ_IA_CODE_GLOBAL_MASK для маскирования прерывание по		
		всем заданным причинам;		
		IRQ INT MASK для маскирования прерывания, по факту приема		
		кода распределенного прерывания;		
		IRQ_ACK_MASK для маскирования прерывания по факту приема		
		кода подтверждения;		
		IRQ INT RST MASK для маскирования прерывания по приему		
		команды сброса от удаленного администратора;		
		IRQ_ISR_TOUTS_MASK для маскирования прерывания по факту		
		истечения одного из таймаутов.		
		По умолчанию все причины данного прерывания замаскированы		
4	IRQ_C01_C11	Прерывание по приему управляющего кода, назначение которого		
		не определено стандартом (С01 или С11) из любого порта SpFR.		
		Если из сети принят код C01, то в регистре STATE_R		
		устанавливается. GOT_C01_CODE=1, если код С11, то		
		устанавливается GOT_C11_CODE=1.		
		Для того, чтобы сбросить данное прерывание, необходимо записать		
		1 в разряд. GOT_C01_CODE или GOT_C11_CODE.		
		Для того, чтобы замаскировать прерывание по приему кодов C01		
		или C11 необходимо в регистре CPU_IRQ_MASK записать «0» в		
		разряд IRQ_C01_MASK или IRQ_C11_MASK. По умолчанию данное прерывание замаскировано		
	<u> </u>	тто умолчанию данное прерывание замаскировано		



Номер	Название	Описание
5	IRQ_BR_SPF	Прерывание по приему корректного пакета Broadcast из любого порта SpFR. Пакет Broadcast считается корректным, если его значение на 1 больше предыдущего принятого пакета Broadcast (по модулю 64) с тем же номером виртуального канала. Для того, чтобы определить с каким номером виртуального канала был принят пакет Broadcast, необходимо прочитать регистр BROADCAST_IN_STATUS1. В этом регистре разряды с 0 до 7 соответствуют виртуальным каналам от 0 до 7 соответственно. разряды с 8 до 11 соответствует виртуальному каналу, заданному в полях. BROADCAST_NUM0, BROADCAST_NUM1, BROADCAST_NUM2, BROADCAST_NUM3 регистра BROADCAST_NUMS. Запись 1 в соответствующий разряд приводит к его сбросу, что в дальнейшем позволяет отследить факт прихода корректного пакета Broadcast из виртуального канала с тем же номером. Для того, чтобы прочитать параметры, соответствующие последнему принятому пакету Broadcast по конкретному номеру виртуального канала необходимо в поле BR_NUM_TO_READ регистра BROADCAST_DATA_IN_PARAMETERS записать номер виртуального канала. Значения параметров принятого пакета Broadcast могут быть прочитаны из регистров ВROADCAST_DATA_IN_PARAMETERS.BR_SEQ_IN, BROADCAST_DATA_IN_PARAMETERS.BR_TYPE_IN и BROADCAST_DATA_IN_PARAMETERS.BR_TYPE_IN
6	IRQ_ARB_TOUT	Прерывание по истечению таймаута арбитраж передачи пакета в сеть. Прерывание является информативным, используется для системного администрирования. Пакет, для которого истек таймаут, стирается в контроллере входа порта автоматически. Для того, чтобы определить в каком входном порту истек данный таймаут, необходимо прочитать поле ARB_TOUT_FL регистра DA_SPF_TOUTS_FLS. І разряд этого поля соответствует І порту. Запись 1 в этот разряд приводит к сбросу флага и позволяет отслеживать ситуации повторного истечения таймаута для других пакетов. Для сброса данного прерывания необходимо сбросить все флаги ARB_TOUT_FL в регистре DA_SPF_TOUTS_FLS. Для того, чтобы замаскировать данное прерывание необходимо записать 0 в разряд IRQ_TARB_MASK регистра CPU_IRQ_MASK По умолчанию данное прерывание замаскировано



Номер	Название	Описание
7	IRQ_SEND_TOUT	Прерывание по истечению таймаута отправки очередного символа в сеть (в режиме GigaSpaceWire) или по факту переполнения счетчика неудачных попыток отправки фрейма (в режиме SpaceFibre). Данное прерывание позволяет отследить ситуацию, когда в сети при установленном соединении невозможно передавать данные. Это может быть следствием каких-либо сбоев в управлении процессом передачи пакетов в соседнем узле. При возникновении этого прерывания рекомендуется выполнить сброс соединения в порту, для которого возникло это прерывание. Для того, чтобы определить для какого именно порта истек таймаут, необходимо прочитать поле DT_TOUT_FL регистра DRT_TOUTS_FLS. І разряд данного поля соответствует І порту. Запись 1 в І разряд приводит к его сбросу. Для того, чтобы сбросить данное прерывание, необходимо записать 1 во все установленные разряды DT_TOUT_FL.
8	IRQ_REC_TOUT	Прерывание по таймауту ожидания приема очередного символа из сети (пакет начал приниматься и почему-то прекратилась передача без разрыва соединения). Данное прерывание позволяет отследить ситуацию, когда в сети при установленном соединении невозможно передавать данные. Это может быть следствием каких-либо сбоев в управлении процессом передачи пакетов в соседнем узле. При возникновении этого прерывания рекомендуется выполнить сброс соединения в порту, для которого возникло это прерывание. Для того, чтобы определить для какого именно порта истек таймаут, необходимо прочитать поле DR_TOUT_FL регистра DRT_TOUTS_FLS І разряд данного поля соответствует І порту. Запись 1 в І разряд приводит к его сбросу. Для того, чтобы сбросить данное прерывание, необходимо записать 1 во все установленные разряды DR_TOUT_FL. Для того, чтобы замаскировать данное прерывание необходимо записать 0 в разряд TRES_MASK регистра CPU_IRQ_MASK По умолчанию данное прерывание замаскировано



Номер	Название	Описание
9	IRQ_REC_PACKET	Прерывание по приему пакета, не являющегося RMAP командой.
		Данное прерывание устанавливается после того, как поле данных и
		дескриптор пакета записываются в соответствующие области
		памяти
		Для того, чтобы определить в какую область памяти (по какому
		каналу DMA) поступил пакет, необходимо прочитать поле PACK REC FL регистра DMA PACK STATUS. І разряд данного
		поля соответствует I каналу DMA. При поступлении очередного
		пакета соответствующий разряд данного поля устанавливается в 1.
		Для того, чтобы в дальнейшем можно было отслеживать
		поступление пакетов по этому каналу DMA необходимо сбросить
		данный разряд посредством записи в него «1».
		Для того, чтобы сбросить данное прерывание необходимо сбросить
		все разряды поля PACK_REC_FL.
		Маскирование данного прерывания осуществляется для каждого
		канала DMA в отдельности. Для того, чтобы замаскировать данное
		прерывание для канала DMA с номером I, необходимо в разряд
		REC_PACKET_MASK(i) регистра DMA_IRQ_MASK1 записать
		значение 0. По умолчанию прерывания от всех каналов DMA
10	IRQ FIFO ERR	замаскированы Прерывание по ошибке при записи дескриптора в FIFO ответов на
10	IKQ_FIFO_EKK	команды чтения и записи – идентификатор команды не
		соответствует идентификатору ни одной из отложенных команд,
		дублетный идентификатор команды (запись в заполненное FIFO).
		Для того, чтобы установить причину прерывания, необходимо
		прочитать регистр FIFO STATUS. Если произошла запись
		дескриптора RMAP команды чтения или записи с несуществующим
		идентификатором, то разряд ANS_NEXIST будет установлен в 1.
		Для его сброса необходимо записать 1 в этот разряд. Если
		произошла запись дескриптора RMAP команды записи с
		дублетным идентификатором, то разряд ANS_DOUBLE будет
		установлен в 1. Для его сброса необходимо записать 1 в этот
		разряд.
		Для маскирования данного прерывания необходимо в регистре FIFO STATUS разряд ANS MASK установить в «0»
		THO_DIATOS Pashad VIAS_IMVOK ACIAHORATE R ((A))



Номер	Название	Описание
11	IRQ_REG1_FINISH	Прерывание по завершению региона 1.
		Для того, чтобы определить для какого именно канала DMA
		завершился регион 1, необходимо прочитать в регистре
		DMA_REGIONS_12_STATUS поле REGIONS_1_STATUS. I разряд
		этого поля соответствует I каналу DMA. Если он установлен в 1, то
		регион завершился. (Если данный канал DMA работает в режиме
		обработки команд от БКУ, то это означает, что все данные из
		области данных на передачу переданы в сеть; если в режиме
		обработки RMAP команд – то область дескрипторов RMAP команд чтения и записи полностью заполнена принятыми дескрипторами.)
		Для того, чтобы сбросить данное прерывание необходимо
		выполнить повторную настройку всех завершившихся для всех
		каналов DMA, либо запретить работу каналов DMA, если их
		дальнейшее использование не планируется.
		Маскирование данного прерывания осуществляется для каждого
		канала DMA в отдельности. Для того, чтобы замаскировать данное
		прерывание для канала DMA с номером I, необходимо в регистре
		DMA_IRQ_MASK2 в рле REGION_1_FINISH_MASK (i) записать
		«0».
		По умолчанию прерывания от всех каналов DMA замаскированы
12	IRQ_REG2_FINISH	Прерывание по завершению региона 2.
		Для того, чтобы определить для какого именно канала DMA
		завершился регион 1, необходимо прочитать в регистре
		DMA_REGIONS_12_STATUS поле REGIONS_2_STATUS. I разряд этого поля соответствует I каналу DMA. Если он установлен в 1, то
		регион завершился. (Если данный канал DMA работает в режиме
		обработки команд от БКУ, то это означает, что все данные из
		области данных на передачу переданы в сеть; если в режиме
		обработки RMAP команд – то область дескрипторов RMAP команд
		чтения и записи полностью заполнена принятыми дескрипторами.)
		Для того, чтобы сбросить данное прерывание необходимо
		выполнить повторную настройку всех завершившихся для всех
		каналов DMA, либо запретить работу каналов DMA, если их
		дальнейшее использование не планируется.
		Маскирование данного прерывания осуществляется для каждого
		канала DMA в отдельности. Для того, чтобы замаскировать данное
		прерывание для канала DMA с номером I, необходимо в регистре
		DMA_IRQ_MASK2 в рле REGION_2_FINISH_MASK (i) записать «0».
		«о». По умолчанию прерывания от всех каналов DMA замаскированы
		по умолчанию прерывания от всех каналов DIVIA замаскированы



Номер	Название	Описание
13	IRQ_REG3_FINISH	Прерывание по завершению региона 3.
		Для того, чтобы определить для какого именно канала DMA
		завершился регион 1, необходимо прочитать в регистре
		DMA_REGIONS_34_STATUS поле REGIONS_3_STATUS. I разряд
		этого поля соответствует I каналу DMA. Если он установлен в 1, то
		регион завершился. (Если данный канал DMA работает в режиме
		обработки команд от БКУ, то это означает, что все данные из
		области данных на передачу переданы в сеть; если в режиме
		обработки RMAP команд – то область дескрипторов RMAP команд
		чтения и записи полностью заполнена принятыми дескрипторами.)
		Для того, чтобы сбросить данное прерывание необходимо
		выполнить повторную настройку всех завершившихся для всех
		каналов DMA, либо запретить работу каналов DMA, если их
		дальнейшее использование не планируется.
		Маскирование данного прерывания осуществляется для каждого
		канала DMA в отдельности. Для того, чтобы замаскировать данное
		прерывание для канала DMA с номером I, необходимо в регистре
		DMA_IRQ_MASK2 в рле REGION_3_FINISH_MASK (i) записать
		«0».
		По умолчанию прерывания от всех каналов DMA замаскированы
14	IRQ_REG4_FINISH	Прерывание по завершению региона 4.
		Для того, чтобы определить для какого именно канала DMA
		завершился регион 1, необходимо прочитать в регистре
		DMA_REGIONS_34_STATUS поле REGIONS_4_STATUS. I разряд
		этого поля соответствует I каналу DMA. Если он установлен в 1, то
		регион завершился. (Если данный канал DMA работает в режиме
		обработки команд от БКУ, то это означает, что все данные из
		области данных на передачу переданы в сеть; если в режиме обработки RMAP команд – то область дескрипторов RMAP команд
		чтения и записи полностью заполнена принятыми дескрипторами.)
		Для того, чтобы сбросить данное прерывание необходимо
		выполнить повторную настройку всех завершившихся для всех
		каналов DMA, либо запретить работу каналов DMA, если их
		дальнейшее использование не планируется.
		Маскирование данного прерывания осуществляется для каждого
		канала DMA в отдельности. Для того, чтобы замаскировать данное
		прерывание для канала DMA с номером I, необходимо в регистре
		DMA IRQ MASK2 в рле REGION 4 FINISH MASK (i) записать
		«0».
		По умолчанию прерывания от всех каналов DMA замаскированы
	I	



Номер	Название	Описание
15	IRQ_RMAP_REJ	Прерывание по получению RMAP команды, которую пришлось
		отбросить из-за превышения допустимого количества
		одновременно обрабатываемых команд. Используется в
		информативных целях для системного администрирования.
		Для того, чтобы определить, по какому именно каналу произошло
		отбрасывание команды, необходимо прочитать регистр
		REJ_STATUS.
		В поле READ_REJ_FL отображаются флаги – признаки
		отбрасывания RMAP команд чтения. І разряд соответствует І
		каналу DMA – отображенному на него вириального канала. Для
		того, чтобы иметь возможность повторного отслеживания этой
		ситуации необходимо сбросить флаг, путем записи 1 в
		соответствующий разряд.
		В поле WRITE_REJ_FL отображаются флаги – признаки
		отбрасывания RMAP команд записи. І разряд соответствует І
		каналу DMA – отображенному на него вириального канала. Для
		того, чтобы иметь возможность повторного отслеживания этой
		ситуации необходимо сбросить флаг, путем записи 1 в
		соответствующий разряд.
		Для того, чтобы сбросить данное прерывание, необходимо сбросить
		все флаги в регистре REJ_STATUS.
		Маскирование данного прерывания осуществляется отдельно для
		команд чтения и команд записи. Для маскирования по
		отбрасыванию команд чтения необходимо в регистре REJ_STATUS
		разряд READ_REJ_MASK установить в «0», для маскирования по
		отбрасыванию команд записи необходимо в регистре REJ_STATUS
1.6	TDO DEC DIALBEA	разряд WRITE_REJ_MASK установить в «0»
16	IRQ_REC_RMAP[3:	Прерывания по приему пакета, являющегося RMAP командой
	0]	чтения или записи для каждого канала DMA. Данные прерывания
		устанавливаются после того, как поле данных и дескриптор пакета
		записываются в соответствующие области памяти.
		Для того, чтобы сбросить данное прерывание необходимо прочитать регистр RMAP_REC_PACK_COUNTERS. Чтение этих
		регистров приводит к сбросу счетчиков принятых пакетов. Одновременно сбрасываются все счетчики, отображаемые в один
		регистр. Если в момент чтения происходит регистрация приема
		очередного пакета, то соответствующий счетчик устанавливается в
		значение 1 и прерывание сброшено не будет.
		Маскирование данного прерывания осуществляется для каждого
		канала DMA в отдельности. Для того, чтобы замаскировать данное
		прерывание для канала DMA с номером I, необходимо в регистре
		DMA IRQ MASK1 в соответствующий разряд поля
		REC PACKET RMAP MASK записать «0». По умолчанию
		прерывания от всех каналов DMA замаскированы
	I.	



12. МНОГОФУНКЦИОНАЛЬНЫЙ БУФЕРИЗИРОВАНННЫЙ ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ (MFBSP)

12.1 Общие сведения об MFBSP

12.1.1 Режимы работы MFBSP

Многофункциональный порт MFBSP может быть использован как порт ввода-вывода общего назначения, как линковый порт (LPORT), либо как последовательный порт. В случае если MFBSP используется как последовательный порт, приёмник и передатчик могут настраиваться независимо. Как приёмник, так и передатчик MFBSP могут работать в режиме SPI либо в режиме I2S. Таким образом, для MFBSP существует 6 различных режимов работы, которые задаются битами LEN и SPI_I2S_EN регистра CSR_MFBSP, битом TMODE регистра TCTR и битом RMODE регистра RCTR. Режимы работы MFBSP и задающие их сочетания значений управляющих бит приведены в Таблица 12.1.

Таблица 12.1. Режимы работы MFBSP

Значение бит, задающих режим				Downey poforty MEDCD
LEN	SPI_I2S_EN	TMODE	RMODE	Режим работы MFBSP
0	0	X	X	Порт ввода-вывода общего назначения (GPIO)
1	0	X	X	Линковый порт(LPORT)
0	1	0	0	Последовательный порт Передатчик – I2S Приёмник – I2S
0	1	0	1	Последовательный порт Передатчик – I2S Приёмник – SPI
0	1	1	0	Последовательный порт Передатчик – SPI Приёмник – I2S
0	1	1	1	Последовательный порт Передатчик – SPI Приёмник – SPI

В данной микросхеме MFBSP может работать только в режимах SPI и GPIO. Установка других режимов работы запрещена.

Более подробное описание функциональных особенностей порта для режима SPI приведено в разделе 12.2.

Более подробное описание функциональных особенностей порта для режима порта вводавывода общего назначения приведено в разделе 12.3.



12.1.2 Структурная схема многофункционального буферизированного последовательного порта

На Рисунок 12.1 представлена структурная схема MFBSP.

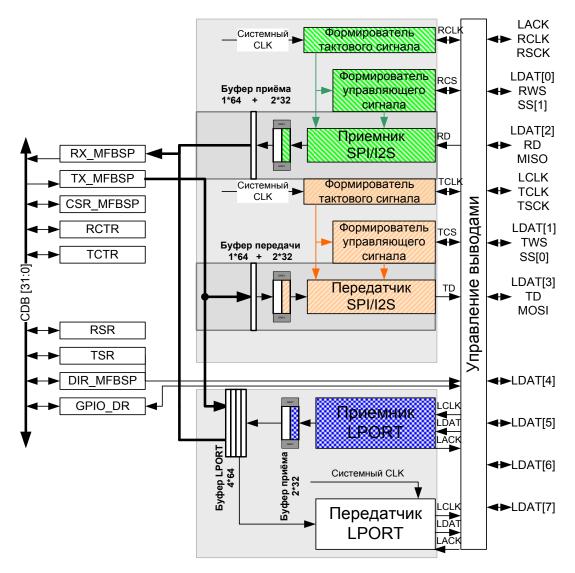


Рисунок 12.1. Структурная схема MFBSP (Защищена патентом РФ №2360282 от 27 июня 2009 года)

В состав MFBSP входят два основных блока: контроллер LPORT и контроллер SPI/I2S. Включение контроллера LPORT производится установкой бита LEN, регистра CSR_MFBSP в 1, включение контроллера SPI_I2S производится установкой бита SPI_I2S_EN, регистра CSR_MFBSP в 1 (Таблица 12.1). Одновременная работа блоков LPORT и SPI/I2S и соответственно установка бит LEN и SPI_I2S_EN в 1 не допускается.

В состав контроллера SPI/I2S входят приёмник, передатчик, буфер приёма и буфер передачи. Приёмник и передатчик могут работать одновременно и независимо. Приёмник осуществляет синхронный приём последовательного кода с внешнего вывода схемы и запись принятых данных в буфер приёма. Передатчик осуществляет чтение данных из



буфера передачи и синхронную выдачу их последовательным кодом на внешний вывод схемы. Запись передаваемых данных в буфер передачи осуществляется при записи по адресу псевдорегистра TX_MFBSP, чтение принятых данных из буфера приёма осуществляется при чтении по адресу псевдорегистра RX MFBSP.

Последовательным портом при обмене данными используется только 6 выводов LCLK, LACK, LDAT[3:0]. Если порт работает в режиме SPI/I2S, выводы LDAT[4:7] могут использоваться как вводы-выводы общего назначения.

MFBPS использует тактовый сигнал CLK, при этом на MFBSP тактовый сигнал CLK подается постоянно, когда включен тактовый сигнал CPU, что позволяет реализовать режим начальной загрузки через MFBSP.

12.1.3 Назначение выводов порта в различных режимах

Таблица 12.2 содержит наименования выводов порта для каждого из режимов – LPORT, SPI, I2S. Таблица 12.3 содержит информацию о назначении каждого вывода в различных режимах.

Таблица 12.2. Обозначение выводов порта для различных режимов работы

LPORT	I2S	SPI
LDAT[7]	-	-
LDAT[6]	-	-
LDAT[5]	-	-
LDAT[4]	-	-
LDAT[3]	TD	MOSI
LDAT[2]	RD	MISO
LDAT[1]	TWS	SS[0]
LDAT[0]	RWS	SS[1]
LCLK	TCLK	TSCK
LACK	RCLK	RSCK

Таблица 12.3. Назначение выводов порта в различных режимах

Наименование вывода	Режим работы порта	Направление вывода	Назначение вывода
LDAT[7:0]	LPORT	IO	Внешняя шина данных LPORT.
LCLK	LPORT	IO	Тактовый сигнал LPORT
LACK	LPORT	IO	Подтверждение готовности приема
TD	I2S	IO	Передаваемые последовательные данные
RD	I2S	IO	Принимаемые последовательные данные
TCLK	I2S	IO	Тактовый сигнал передатчика I2S
RCLK	I2S	IO	Тактовый сигнал приемника I2S
TWS	I2S	IO	Сигнал выбора канала для передаваемых данных
RWS	I2S	Ю	Сигнал выбора канала для принимаемых данных



Наименование вывода	Режим работы порта	Направление вывода	Назначение вывода
MOSI	SPI	IO	Вывод последовательных данных. Направление вывода определяется программно
MISO	SPI	Ю	Вывод последовательных данных. Направление вывода определяется программно
TSCK	SPI	IO	Тактовый сигнал передатчика SPI
RSCK	SPI	IO	Тактовый сигнал приемника SPI
SS [0]	SPI	IO	В режиме ведущего: Сигнал выбора устройства 0. В режиме ведомого: сигнал выбора ведомого. Низкий уровень на входе SS[0] обозначает, что передатчику MFBSP необходимо выдавать последовательные данные (если приёмник MFBSP находится в зависимом от передатчика режиме, то активизируется и приёмник).
SS [1]	SPI	Ю	В режиме ведущего: Если приёмник в зависимом от передатчика режиме - сигнал выбора устройства 1. Если передатчик в независимом от приёмника режиме — сигнал выбора приёмником устройства 0. В режиме ведомого: Сигнал выбора ведомого. Только в случае, когда приёмник в независимом от передатчика режиме. Низкий уровень на входе SS[1] обозначает, что приёмнику MFBSP необходимо принимать последовательные данные.

В данной микросхеме внешним выводам SCK, SO, SI, CS (шина SPI) соответствуют выводы порта TSCK, MOSI, MISO, SS[0] соответственно.

12.1.4 Перечень регистров MFBSP

Таблица 12.4 содержит перечень регистров многофункционального порта.

Таблица 12.4. Перечень регистров многофункционального буферизованного порта

Условное обозначение регистра	Внутренний адрес	Название регистра
TX_MFBSP	0	Буфер передачи данных
RX_MFBSP	0	Буфер приёма данных
CSR_MFBSP	1	Регистр управления и состояния



Условное обозначение регистра	Внутренний адрес	Название регистра
DIR_MFBSP	2	Регистр управления направлением выводов порта
		ввода-вывода
GPIO_DR	3	Регистр данных порта ввода-вывода
TCTR	4	Регистр управления передатчиком
RCTR	5	Регистр управления приёмником
TSR	6	Регистр состояния передатчика
RSR	7	Регистр состояния приёмника

12.1.5 Прерывания от MFBSP

Бит MFBSP_TXBUF, регистра QSTR0, устанавливается в случае, если число 64-разрядных слов, находящихся в буфере передачи, меньше, либо равно пороговому значению TLEV, задаваемому в регистре TSR (Рисунок 12.2). Для установки бита MFBSP_TXBUF также необходимо, чтобы линковый порт был включен на передачу (LEN=1 и LTRAN=1) либо включен передатчик SPI/I2S (SPI_I2S_EN=1, TEN=1) и разрешено прерывание MFBSP TXBUF (MFBSP TXBUF IRQ EN).

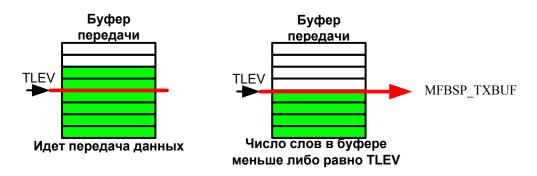
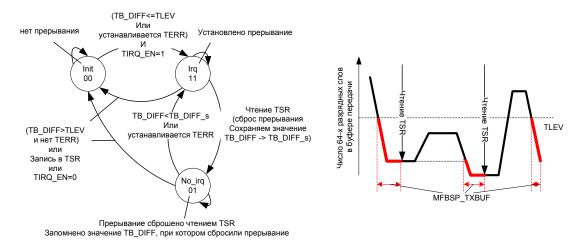


Рисунок 12.2. Назначение бит TLEV, регистра TSR

Прерывание MFBSP_TXBUF автоматически сбрасывается, если число 64-разрядных слов, находящихся в буфере передачи, становится больше порогового значения TLEV и при этом во время передачи не возникало ошибки (TERR = 0). Даже если описанное условие не выполнено, прерывание можно программно сбросить, прочитав регистр TSR. В этом случае прерывание сбросится и запомнится текущее значение слов в буфере передачи. Если число слов в буфере передачи начнет уменьшаться или произойдет ошибка передачи, то прерывание снова установится. Увеличение числа слов в буфере передачи не приведет к установке прерывания, даже, если число слов в буфере ниже порога TLEV (Рисунок 12.3).





Pисунок 12.3. Механизм установки и сброса прерывания MFBSP_TXBUF. На рисунке TIRQ EN = (LEN & LTRAN || TEN & SPI 12S EN)

Бит MFBSP_RXBUF, регистра QSTR0, устанавливается в случае, если число 64-разрядных слов в буфере приёма больше чем пороговое значение RLEV, задаваемое в регистре RSR (Рисунок 12.4). Для установки бита MFBSP_RXBUF также необходимо, чтобы линковый порт был включен на приём (LEN=1 и LTRAN=0) либо включен приёмник SPI/I2S (SPI_I2S_EN=1, REN=1) и разрешено прерывание MFBSP_RXBUF (MFBSP_RXBUF IRQ_EN).

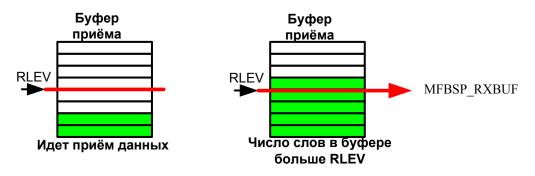


Рисунок 12.4. Назначение бит RLEV, регистра RSR

Прерывание MFBSP_RXBUF автоматически сбрасывается, если число 64-разрядных слов, находящихся в буфере приёма, становится меньше порогового значения RLEV и при этом во время приёма не возникало ошибки (RERR = 0). Даже если описанное условие не выполнено, прерывание можно программно сбросить, прочитав регистр RSR. В этом случае прерывание сбросится и запомнится текущее значение слов в буфере чтения. Если число слов в буфере чтения начнет увеличиваться, то прерывание снова установится. Уменьшение числа слов в буфере чтения не приведет к установке прерывания, даже, если превышен порог RLEV (Рисунок 12.5).



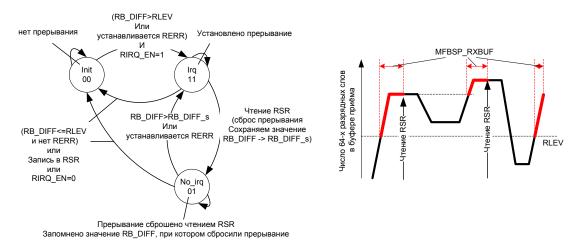


Рисунок 12.5. Механизм установки и сброса прерывания MFBSP_RXBUF. На рисунке RIRQ EN = (LEN & !LTRAN || REN & SPI 12S EN)

Бит SRQ, регистра QSTR2, формируется при запросе на обслуживание, если порт MFBSP выключен (LEN=0, SPI_I2S_EN=0) и на выводах LACK или LCLK высокий уровень, при условии, что разрешено прерывание по запросу на обслуживание (LPT_IRQ_EN=1).

12.2 Работа MFBSP в режиме SPI

12.2.1 Назначение последовательного порта в режиме SPI

Режим SPI буферизированного последовательного порта предназначен для организации дуплексного обмена последовательными данными с внешними устройствами.

Порт в режиме SPI позволяет одновременно передавать и принимать последовательные данные. Приемник и передатчик контроллера можно настраивать независимо, при этом возможен перевод приёмника в зависимое от передатчика состояние.

Поддерживается независимое задание направления выводов последовательных данных порта, осуществляемое установкой соответствующих бит регистра DIR MFBSP.

В режиме ведущего устройства к MFBSP параллельно может быть подключено до двух ведомых SPI устройств.

Формирование сигнала выбора ведомого возможно как в автоматическом так и в программном режиме. В автоматическом режиме после передачи каждого слова сигнал выбора ведомого возвращается в высокое состояние. При программном управлении сигналами выбора ведомого данные сигналы изменяются посредством записи в контрольный регистр передатчика.

В данной реализации порта существует ограничение на выбор направления выводов в режиме SPI: тактовый и управляющий сигналы в режиме SPI должны быть либо оба заданы как вход, либо оба заданы как выход;



В данной реализации порта не предусмотрена возможность соединения нескольких микропроцессоров по цепочке с использованием SPI интерфейса. Микропроцессор может только управлять загрузкой последовательных данных в другие ведомые устройства, соединенные по цепочке.

В данной реализации порта в режиме ведомого устройства сигнал выбора ведомого предварительно пересинхронизируется на внутреннюю частоту порта, поэтому для устойчивой работы порта в режиме ведомого SPI устройства уровень сигнала SS, если необходима его установка в 1 между передачами, должен удерживаться как минимум два периода внутренней частоты CLK. Поэтому, если приемник работает в зависимом от передатчика режиме (RCS_CP=1, RCLK_CP=1), передатчик работает на максимальной частоте (TCLK_RATE=0) и формирует сигнал SS в автоматическом режиме (SS_DO=0, TCS_DIR=1), необходимо установить значение TSS_RATE>=1 чтобы удерживать сигнал SS в высоком уровне как минимум два периода внутренней частоты CLK.

12.2.2 Регистр управления и состояния CSR_MFBSP (режим SPI)

Регистр CSR_MFBSP (Таблица 12.5) используется для включения режима последовательного порта и разрешения прерываний от MFBSP.

Таблица 12.5. Назначение разрядов регистра CSR MFBSP в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:17	-	Не используется	R	0
16	MFBSP_TXBUF_IRQ_E N	Разрешение прерывания MFBSP_TXBUF: 0 – прерывание запрещено; 1- прерывание разрешено.	RW	1
15	MFBSP_RXBUF_IRQ_E N	Разрешение прерывания МFBSP_RXBUF: 0 – прерывание запрещено; 1- прерывание разрешено.	RW	1
14:10	-	Не используется	-	0
9	SPI_I2S_EN	Включение режима SPI/I2S: 0 – Работа в режиме LPORT 1 – Работа в режиме SPI/I2S	RW	0
8:5	-	Не используется	-	0
4:3	LSTAT	Состояние буфера: При LTRAN = 0 показывает состояние буфера приёма При LTRAN = 1 показывает состояние буфера передачи 00 — буфер пуст; 10 — буфер не пуст; 11 — буфер полон.	R	0
2	-	Не используется	-	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
1	LTRAN	Назначение бит LSTAT: 0 - SPI работает в направлении приёма, LSTAT отображает состояние буфера приёма 1 - SPI работает в направлении	RW	0
		передачи, LSTAT отображает состояние буфера передачи		
0	LEN	В режиме SPI должен быть установлен в 0	RW	0

12.2.3 Регистр управления направлением выводов DIR_MFBSP (режим SPI)

Регистр управления направлением выводов DIR_MFBSP (Таблица 12.6) предназначен для индивидуальной настройки направления каждого вывода последовательного порта.

Таблица 12.6. Назначение разрядов регистра DIR_MFBSP в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
9:6	-	Не используется	-	0
5	TD_DIR	Направление вывода MOSI:	RW	0
		0 – MOSI – вход (при RD_DIR = 1		
		последовательные данные принимаются со входа		
		MOSI - эквивалент SDI)		
		1 – MOSI - выход (MOSI – является выходом для		
		передачи последовательных данных и является		
		эквивалентом SDO)		
4	RD_DIR	Направление вывода MISO:	RW	0
		0 – MISO – вход (последовательные данные		
		принимаются со входа MISO - эквивалент SDI)		
		1 – MISO - выход (MISO – является выходом для		
		передачи последовательных данных и является эквивалентом SDO)		
3	TCS DIR		RW	0
3	ICS_DIK	Направление вывода SS[0]: 0 – SS[0] – вход (управляющий сигнал для	KW	0
		передатчика снимается с вывода SS[0])		
		1 – SS[0] - выход, управляющий сигнал		
		формируется передатчиком		
2	RCS DIR	Направление вывода SS[1]:	RW	0
	Res_bit	0 – SS[1] – вход (управляющий сигнал для	1000	
		приёмника снимается с вывода SS[1])		
		1 – SS[1] - выход, в этом случае на SS[1] в		
		зависимости от состояния бита RCS СР подаются		
		управляющие сигналы, формируемые либо		
		приемником, либо передатчиком		
1	TCLK_DIR	Направление вывода TSCK:	RW	0
		0 – TSCK – вход (тактовый сигнал TSCK		
		принимается от внешнего источника)		
		1 – TSCK – выход (тактовый сигнал TSCK		
		формируется передатчиком)		



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
0	RCLK_DIR	Направление вывода RSCK: 0 – RSCK – вход (тактовый сигнал RSCK	RW	0
		принимается от внешнего источника) 1 – RSCK – выход (тактовый сигнал RSCK		
		формируется приёмником)		

При RD_DIR = 0 и TD_DIR = 0 данные снимаются с MISO, при RD_DIR = 1 и TD_DIR = 1 на MOSI и MISO выдаются одинаковые данные с передатчика.

12.2.4 Регистр управления приёмником RCTR (режим SPI)

Таблица 12.7. Назначение разрядов регистра RCTR в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:27	-	Не используется	-	0
26	RSIGN	Значение заполнителя: Если длина принимаемого слова меньше 32 при отключенном паковщике или меньше 16 при включенном паковщике, то неиспользуемые биты принятого слова заполняются При RSIGN = 0 нулями При RSIGN = 1 значением старшего разряда в принятом слове	RW	0
25	RPACK	В режиме SPI обязательно RPACK=0.	RW	0
24:20	RWORDLE N	Длина принимаемого слова: Число бит в принимаемом слове равно RWORDLEN + 1. RWORDLEN должно быть больше 0.	RW	5'b0
19	RMBF	Порядок передачи бит: 0 – младшим битом вперед 1 – старшим битом вперед	RW RW	1
18:12	-	Не используется	_	0
11	RDEL	Задержка начала приёма данных на пол такта: (Эквивалентно СРНА в спецификации Motorola). Задает фронт, по которому производится захват данных приёмником (фронт приёма). Ниже приведено соответствие полярности фронта приёма значениям бит RNEG, RDEL: RNEG = 0, RDEL = 0 – захват по переднему фронту RSCK RNEG = 0, RDEL = 1 – захват по заднему фронту RSCK RNEG = 1, RDEL = 0 – захват по заднему фронту RSCK RNEG = 1, RDEL = 1 – захват по переднему фронту RSCK RNEG = 1, RDEL = 1 – захват по переднему фронту RSCK	RW	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
10	RNEG	Полярность тактового сигнала приёмника: (эквивалентно CPOL в спецификации Motorola). Задает исходное состояние вывода RSCK и фронт, по которому производится захват данных приёмником (фронт приёма). Ниже приведено соответствие полярности фронта приёма значениям бит RNEG, RDEL: RNEG = 0, RDEL = 0 – захват по переднему фронту RSCK RNEG = 0, RDEL = 1 – захват по заднему фронту RSCK RNEG = 1, RDEL = 0 – захват по заднему фронту RSCK RNEG = 1, RDEL = 1 – захват по переднему фронту RSCK RNEG = 1, RDEL = 1 – захват по переднему фронту RSCK исходное состояние RSCK = RNEG.	RW	0
9	-	Не используется	-	0
8:4	RCLK_RAT E [4:0]	Делитель частоты приёмника: В случае, если частота формируется самим приёмником, определяет частоту приёмника RSCK= CLK/((RCLK_RATE+1)*2), где CLK – частота, подаваемая на порт со стороны системы.	RW	0
3	RCS_CP	Управление сигналом выбора ведомого приёмника: 0 — сигнал SS[1] принимается приёмником с внешнего вывода или формируется самим приёмником. 1 - сигнал SS[1] формируется передатчиком и является сигналом выбора ведомого устройства 1. Приёмник осуществляет приём данных синхронно с передатчиком, (в этом случае RCLK_CP должно быть так же в 1).	RW	0
2	RCLK_CP	Дублирование сигнала RSCK: 0 — RSCK формируется или принимается независимо от передатчика 1 — RSCK приёмника дублирует TSCK передатчика (в этом случае RCS_CP должно быть так же в 1).	RW	0
1	RMODE	Режим работы приёмника: 0 – режим I2S 1 – режим SPI	RW	0
0	REN	Разрешение работы приёмника: 0 – приемник выключен 1 – приемник включен	RW	0



12.2.5 Регистр управления передатчиком TCTR (режим SPI)

Таблица 12.8. Назначение разрядов регистра TCTR в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31	SS[1]	При SS_DO = 1 значение бита SS передаются на вывод LDAT[0]	-	0
30	SS[0]	биты управления шиной Slave Select: Позволяет активировать подключенное ведомое устройство. При SS_DO = 0 установка соответствующего бита SS в 1 означает выбор ведомого устройства, с которым будет производиться обмен данными При SS_DO = 1 значения бит SS передаются на выводы SS напрямую	RW	0
29:26	-	Не используется	-	0
25	TPACK	В режиме SPI обязательно TPACK=0.	RW	0
24:20	TWORDLE N	Длина передаваемого слова: Число бит в передаваемом слове равно TWORDLEN + 1. TWORDLEN должно быть больше 0.	RW	5'b0
19	TMBF	Порядок передачи бит: 0 – младшим битом вперед 1 – старшим битом вперед	RW	1
18:12	-	Не используется	-	0
11	TDEL	Задержка начала передачи данных на пол такта: (Эквивалентно СРНА в спецификации Motorola). Задает фронт, по которому производится выдача данных передатчиком (фронт выдачи). Ниже приведено соответствие полярности фронта выдачи значениям бит TNEG, TDEL: TNEG = 0, TDEL = 0 – выдача по заднему фронту TSCK TNEG = 0, TDEL = 1 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 0 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 0 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 1 – выдача по заднему фронту TSCK	RW	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
10	TNEG	Полярность тактового сигнала передатчика: (эквивалентно CPOL в спецификации Motorola). Задает исходное состояние вывода TSCK и фронт, по которому производится выдача данных передатчиком (фронт выдачи). Ниже приведено соответствие полярности фронта выдачи значениям бит TNEG, TDEL: TNEG = 0, TDEL = 0 – выдача по заднему фронту TSCK TNEG = 0, TDEL = 1 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 0 – выдача по переднему фронту TSCK TNEG = 1, TDEL = 1 – выдача по заднему фронту TSCK TNEG = 1, TDEL = 1 – выдача по заднему фронту TSCK Ucxодное состояние TSCK = TNEG.	RW	0
9	_	Не используется	_	0
8:4	TCLK_RAT E	Делитель частоты передатчика: В случае, если частота формируется самим передатчиком, определяет частоту передатчика ТSCK = CLK/((TCLK_RATE+1)*2), где CLK — частота, подаваемая на порт со стороны системы.	RW	0
3	SS_DO	управление выводами SS: 0 — управление выводами SS производится в автоматическом режиме. С началом передачи вывод SS, для которого соответствующий бит SS, регистра TCRT установлен в 1 переводится в низкое состояние, с окончанием передачи вывод SS переводится в высокое состояние. Если соответствующий выводу бит SS установлен в 0 вывод SS всегда находится в высоком состоянии. 1 — значения бит SS напрямую передаются на внешние выводы. В этом случае необходимо программное управление шиной SS в процессе передачи	RW	0
2	-	В режиме SPI не используется	-	0
1	TMODE	Режим работы передатчика: 0 – режим I2S 1 – режим SPI	RW	0
0	TEN	Разрешение работы передатчика: 0 – приемник выключен 1 – приемник включен	RW	0



12.2.6 Регистр состояния приёмника RSR (режим SPI)

Таблица 12.9. Назначение разрядов регистра RSR в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	Не используется	-	0
27:24	RSS_RATE	Если сигнал SS формируется приёмником, то задает время удержания сигнала SS в высоком уровне между передачами слов. Время удержания SS определяется как TRCLK/2*(RSS_RATE+1), где TRCLK период тактового сигнала RCLK	RW	0
23	-	Не используется	-	0
22:20	RB_DIFF	Количество принятых 64-разрядных слов в буфере приёма (мах 8).	R	0
19:17	-	Не используется	-	0
16:12	RCLK_RAT E [9:5]	Делитель частоты приёмника: В случае, если частота формируется самим приёмником, определяет частоту приёмника RCLK = CLK/((RCLK_RATE+1)*2), где CLK – частота, подаваемая на порт со стороны системы.	RW	0
11	-	Не используется	-	
10	RRUN	Идёт приём: 0 — приёмник в состоянии ожидания 1 — идёт приём очередного слова	R	0
9	RERR	Ошибка передачи: 0 — приём проходил в штатном режиме 1 - была запись в полный буфер приёма (потеря данных). Флаг сбрасывается записью 0 в 6-й разряд регистра RSR.	RW	0
8	RSBF	Буфер пересинхронизации в направлении приёма полон: 0 — буфер пересинхронизации в направлении приёма не полон 1 — буфер пересинхронизации в направлении приёма полон	R	0
7	RSBE	Буфер пересинхронизации в направлении приёма пуст: 0 — буфер пересинхронизации в направлении приёма не пуст 1 — буфер пересинхронизации в направлении приёма пуст	R	1
6:4	RLEV	Порог прерывания от буфера приёма: Прерывание формируется если число принятых 64-х разрядных слов больше RLEV	RW	7
3	RBHL	Достигнут порог прерывания в буфере приёма: 1 — число 64-разрядных слов в буфере приёма больше чем задано в RLEV 0 — число 64-разрядных слов в буфере приёма меньше либо равно RLEV	R	0
2	RBHF	Буфер приёма полон наполовину	R	0



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
1	RBF	Буфер приёма полон:	R	0
		0 – буфер приёма не полон		
		1 – буфер приёма полон		
0	RBE	Буфер приёма пуст:	R	1
		0 – буфер приёма не пуст		
		1 – буфер приёма пуст		

12.2.7 Регистр состояния передатчика TSR (режим SPI)

Таблица 12.10. Назначение разрядов регистра TSR в режиме SPI

Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
31:28	-	Не используется	-	0
27:24	TSS_RATE	Если сигнал SS формируется передатчиком, то задает время удержания сигнала SS в высоком уровне между передачами слов. Время удержания SS определяется как TTCLK/2*(TSS_RATE+1), где TTCLK период тактового сигнала TCLK	RW	0
23	-	Не используется	-	0
22:20	TB_DIFF	Количество свободных 64-разрядных позиций в буфере передачи (в буфер передачи можно записать еще TB_DIFF 64-разрядных слов).	R	8
19:17	-	Не используется	-	0
16:12	TCLK_RAT E [9:5]	Делитель частоты передатчика: В случае, если частота формируется самим передатчиком, определяет частоту передатчика ТСLK = CLK/((TCLK_RATE+1)*2), где CLK – частота, подаваемая на порт со стороны системы.	RW	0
11	-	Не используется	-	0
10	TRUN	Идёт передача: 0 – передатчик в состоянии ожидания 1 – идёт передача очередного слова	R	0
9	TERR	Ошибка передачи: 0 — передача проходила в штатном режиме 1 - было чтение из пустого буфера передачи (передача некорректных данных). Флаг сбрасывается записью 0 в 6-й разряд регистра TSR.	RW	0
8	TSBF	Буфер пересинхронизации в направлении передачи полон: 0 — буфер пересинхронизации в направлении передачи не полон 1 — буфер пересинхронизации в направлении передачи полон	R	0
7	TSBE	Буфер пересинхронизации в направлении передачи пуст: 0 — буфер пересинхронизации в направлении передачи не пуст 1 — буфер пересинхронизации в направлении передачи пуст	R	1



Номер разряда	Условное обозначение	Назначение	Доступ	Исходное состояние
6:4	TLEV	Порог прерывания от буфера передачи:	R	0
		Прерывание формируется если число 64-х		
		разрядных слов в буфере передачи меньше либо равно TLEV.		
		В режиме передачи данных с использованием		
		DMA определяет степень заполнения буфера		
		передачи, при которой происходит запись в буфер		
		очередной пачки данных		
3	TBLL	Достигнут порог прерывания в буфере передачи:	R	1
		1 – число 64-х разрядных слов в буфере передачи		
		меньше либо равно TLEV		
		0 – число 64-х разрядных слов в буфере передачи		
	TDIIE	больше TLEV	D	0
2	TBHF	Буфер передачи заполнен наполовину	R	0
1	TBF	Буфер передачи полон:	R	0
		0 – буфер передачи не полон		
		1 – буфер передачи полон		
0	TBE	Буфер передачи пуст:	R	1 1
		0 – буфер передачи не пуст		
		1 – буфер передачи пуст		

12.2.8 Структурная схема MFBSP для режима SPI

На Рисунок 12.6 представлена структурная схема MFBSP для режима SPI.

Включение режима SPI производится установкой бит LEN=0, SPI_I2S_EN=1, я передатчика), RMODE = 1 (для приёмника).

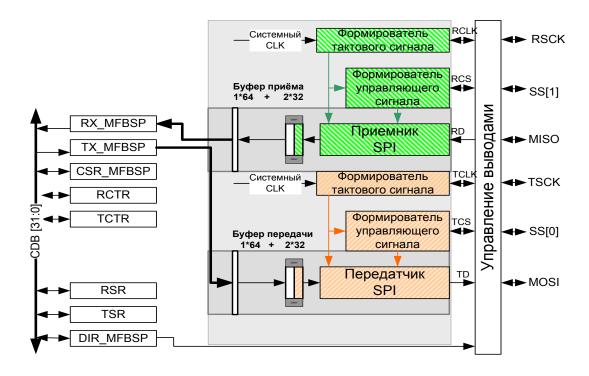


Рисунок 12.6. Структурная схема MFBSP для режима SPI



12.2.9 Соединение порта с внешним устройством

Подключение к MFBSP ведомого устройства по интерфейсу SPI приведено на Рисунок 12.7.

MFBSP позволяет подключить одно ведомое SPI устройство. Активация ведомого устройства с которым будет производиться обмен осуществляется битом SS[0], регистра TCTR.



Рисунок 12.7. Подключение к MFBSP ведомого устройства по интерфейсу SPI

12.2.10 Передача данных в режиме SPI

В режиме SPI возможна передача данных при четырёх сочетаниях бит TDEL и TNEG (Рисунок 12.8, Рисунок 12.9). При этом TNEG — задает начальное состояние вывода TCLK и полярность фронта, по которому производится чтение. TDEL задает смещение передаваемых данных на пол фазы. Значения RNEG и RDEL приёмника должны соответствовать TNEG и TDEL передатчика. После аппаратного сброса SS_DO=0, в этом случае управление сигналом выбора ведомого производится в автоматическом режиме.



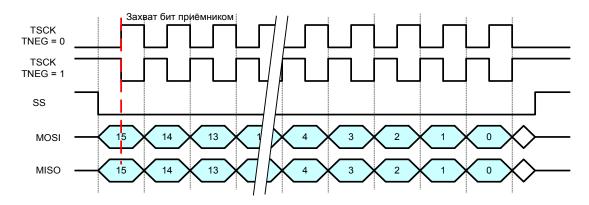


Рисунок 12.8. Передача одного слова в режиме SPI с автоматической генерацией управляющего сигнала TMODE = 1, TMBF = 1, TDEL = 0, SS_DO = 0. Диаграммы тактового сигнала TSCK представлены для различных значений TNEG

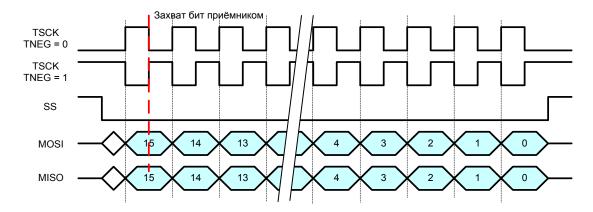
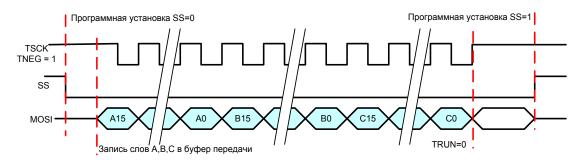


Рисунок 12.9. Передача одного слова в режиме SPI с автоматической генерацией управляющего сигнала TMODE = 1, TMBF = 1, TDEL = 1, SS_DO = 0. Диаграммы тактового сигнала TSCK представлены для различных значений TNEG

Чтобы передать несколько слов без изменения уровня на внешнем выводе SS можно использовать программное управление внешним выводом SS, в этом случае SS_DO необходимо установить в 1, программно установить вывод SS в 0, записать передаваемые данные в буфер передачи (или включить канал DMA на передачу), дождаться фактического окончания передачи (бит TRUN регистра TSR сбрасывается в 0), после чего программно установить вывод SS в 1 (Рисунок 12.10).



Pucyhok 12.10. Передача трёх слов в режиме SPI с программным управлением сигналом SS, TMODE = 1, TMBF = 1, TDEL = 0, TNEG = 0, SS DO = 1



В режиме ведомого устройства сигнал выбора ведомого предварительно пересинхронизируется на внутреннюю частоту порта, поэтому для устойчивой работы порта в режиме ведомого SPI устройства уровень сигнала SS, если необходима его установка в 1 между передачами, должен удерживаться как минимум два периода внутренней частоты CLK.

Непосредственно к тактовому сигналу TSCK данное ограничение не применяется, т.е. частота TSCK может быть больше CLK.

Когда MFBSP работает в режиме ведущего SPI устройства, время удержания сигнала SS при автоматическом формировании данного сигнала может регулироваться программно. В этом случае время между последним фронтом тактового сигнала для последней пересылки и установкой сигнала SS в 1 равно времени между установкой и сбросом сигнала SS и равно времени между сбросом сигнала SS первым фронтом тактового сигнала для новой пересылки. Это время определятся как **TSS** (TSS RATE+1)*TTCLK/2, где TTCLK – период тактового сигнала, генерируемого портом для последовательной передачи данных (Рисунок 12.11). Если необходимо формировать сигнал SS средствами приёмника – то для этих целей используется поле RSS RATE.

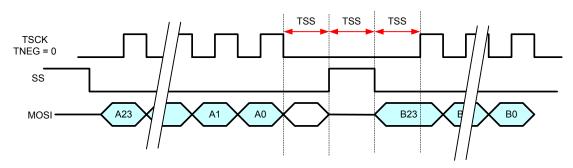


Рисунок 12.11. Управление временем удержания сигнала SS в высоком уровне между передачами, на картинке TNEG = 0, TDEL = 0, TMBF = 1, TWORDLEN = 23, TSS RATE = 1

12.2.11 Пример чтения 8 разрядного слова по заданному адресу из ведомого устройства с интерфейсом C-BUS

Для чтения слова по указанному адресу по интерфейсу C-BUS необходима передача двух 8ми битных слов.

Для организации такого чтения необходимо записать соответствующий ведомому устройству бит SS, регистра TCTR, 1;

Перевести порт в режим SPI (LEN = 0, SPI_I2S_EN = 1, RMODE = 1, TMODE = 1);

Hастроить приемник и передатчик: TDEL = RDEL = 0; TNEG = RNEG = 0; TWORDLEN = RWORDLEN = 5'h0F; RCLK_CP = 1; RCS_CP = 1, SS_DO = 0;



Включить приемник и передатчик REN = 1, TEN = 1;

Записать в регистр LTX 32-х разрядное слово, содержащее во втором байте 7-разрядный адрес и бит WR, значение младшего байта не важно.

Ожидаем до тех пор, пока в буфер приёма не будет записано приятое слово (RSR[0] сбрасывается в 0)

В прочитанном по адресу LRX 32-х разрядном слове, младшие 8 бит — слово, прочитанное из ведомого устройства.

На Рисунок 12.12 представлены временные диаграммы для передачи по интерфейсу CBUS.

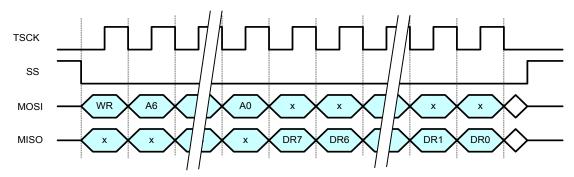


Рисунок 12.12. Пример чтения 8-разрядного слова из ведомого устройства (интерфейс C-BUS)



12.2.12 Формирование тактовых сигналов приёмника (RSCK) и передатчика (TSCK)

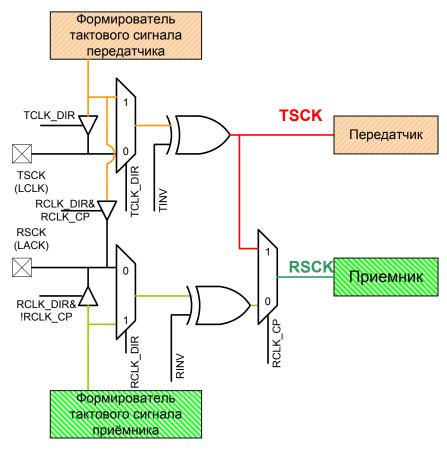


Рисунок 12.13. Схема формирования тактовых сигналов приёмника и передатчика в режиме SPI

На Рисунок 12.13 представлена схема формирования тактовых сигналов приёмника и передатчика в режиме SPI.

В зависимости от значения бита TCLK_DIR, тактовый сигнал передатчика TSCK может как формироваться самим передатчиком, так приниматься с внешнего вывода. В зависимости от значений бит TMODE, TNEG и TDEL тактовый сигнал либо передаётся передатчику без изменений, либо инвертируется.

В зависимости от значения бита RCLK_DIR, тактовый сигнал приёмника RSCK может как формироваться самим приёмником, так приниматься с внешнего вывода. В зависимости от значений бит RMODE, RNEG и RDEL тактовый сигнал либо передаётся приёмнику без изменений, либо инвертируется.

Если бит RCLK_CP установлен в 1, то тактовый сигнал приёмника копирует тактовый сигнал передатчика. Для корректной работы устройства в этом случае настройки полярности тактового сигнала приёмника и передатчика должны совпадать (TNEG=RNEG, TDEL=RDEL).



При RCLK_CP = 1 тактовый сигнал передатчика передаётся на внешний вывод приёмника, только если передатчик сам формирует тактовый сигнал и вывод тактового сигнала приёмника сконфигурирован как выход (TCLK DIR=1, RCLK DIR=1).

12.2.13 Формирование управляющих сигналов приёмника и передатчика в режиме SPI

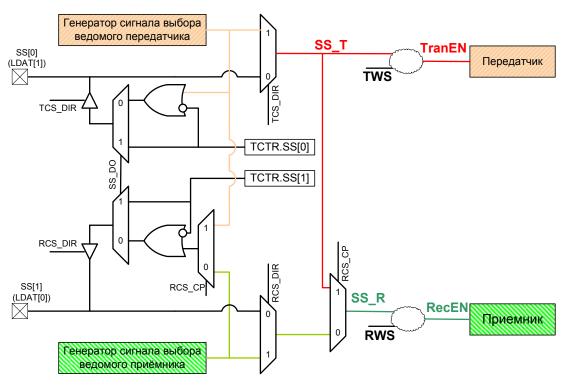


Рисунок 12.14. Схема формирования управляющих сигналов в режиме SPI

На Рисунок 12.14 представлена схема формирования управляющих сигналов в режиме SPI.

SS — шина выбора ведомого устройства. Низкий уровень сигнала SS, поданный на ведомое устройство означает, что данное устройство выбрано и с приходом тактового сигнала SCK должно начать обмен данными с ведущим устройством.

MFBSP с зависимым от передатчика приёмником в режиме ведущего позволяет параллельно подключать до двух ведомых устройств по шине SPI и формировать сигналы выбора ведомого устройства как в автоматическом режиме, так и программно.

MFBSP с зависимым от передатчика приёмником может работать как ведомое SPI устройство, управляемое внешним сигналом SS[0] и внешней тактовой частотой TSCK, обеспечивая обмен данными в дуплексном режиме.



MFBSP позволяет организовать независимый приём и передачу данных по интерфейсу SPI. В этом случае SS[0] — управляющий сигнал передатчика, SS[1] — управляющий сигнал приёмника.

При TCS_DIR = 1 передатчик SPI формирует сигнал выбора ведомого, SS[0] - выход. В автоматическом (SS_DO=0) режиме формирования управляющего сигнала перед началом передачи очередного слова сигнал выбора ведомого переводится в низкий уровень, а по окончании передачи слова сигнал выбора ведомого снова переводится в высокий уровень. Изменение уровня на выводе SS[0] происходит только в случае, если соответствующий бит SS[0] регистра TCTR установлен в 1. Если приёмник в зависимом от передатчика режиме (RCS_CP = 1) и SS[1] сконфигурирован как выход (RCS_DIR=1), то вывод SS[1] используется как сигнал выбора дополнительного ведомого устройства. Изменение уровня на выводе SS[1] происходит только, в случае, если соответствующий бит SS[1] регистра TCTR установлен в 1. В случае программного управления шиной SS (SS_DO = 1) значения бит SS[1:0] контрольного регистра TCTR передаются непосредственно на выводы SS[1:0].

Если приёмник в зависимом от передатчика режиме (RCS_CP=1) и вывод SS[0] сконфигурирован как вход (TCS_DIR = 0), тогда MFBSP работает в режиме дуплексного ведомого SPI устройства. Сигнал выбора ведомого принимается с внешнего вывода SS[0] и используется как приёмником, так и передатчиком.

Если приёмник работает в независимом от передатчика режиме (RCS_CP=0), то в режиме ведущего, когда вывод SS[1] сконфигурирован как выход (RCS_DIR=1) формируемый приёмником сигнал выбора ведомого направляется на вывод SS[1]. При автоматическом формировании управляющего сигнала (SS_DO = 0) перед началом приёма очередного слова сигнал SS[1] автоматически переводится в низкий уровень и переводится в высокий уровень по окончании приёма каждого слова. В режиме ведущего устройства приём слов приёмником ведется до заполнения буфера приёма. В режиме ведомого устройства, когда вывод SS[1] сконфигурирован как вход (RCS_DIR=0) независимый приёмник (RCS_CP=0) принимает сигнал выбора ведомого с вывода SS[1].

В режиме SPI направление выводов тактового сигнала и управляющего сигнала должно строго совпадать. Т.е. TCLK_DIR=TCS_DIR. В случае если приёмник работает независимо от передатчика, то RCLK_DIR=RCS_DIR.



12.2.14 Тракт передачи данных

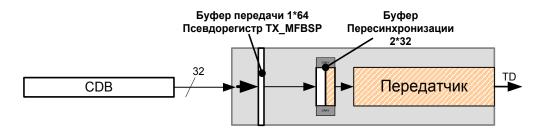


Рисунок 12.15. Тракт передачи данных для режима SPI

На Рисунок 12.15. Тракт передачи данных для режима SPIпредставлен тракт передачи данных для режима SPI.

Чтобы инициировать передачу данных по последовательному порту, необходимо включить последовательный порт (SPI_I2S_EN=1) и передатчик (TEN=1), после чего-либо начать производить запись передаваемых 32-х разрядных слов в буфер передачи по адресу псевдорегистра ТХ MFBSP.

Данные записанные в буфер передачи автоматически перемещаются буфер пересинхронизации направления передачи, если он не полон. буфер Запись пересинхронизации направления передачи осуществляется на системной частоте СLK, чтение из буфера пересинхронизации осуществляется на частоте передатчика ТСЬК. Как только в буфере пересинхронизации оказалось хотя бы одно слово, передатчиком инициируется передача. Передатчиком производится последовательная выдача бит очередного 32-х разрядного слова до тех пор, пока число переданных бит не достигнет TWORDLEN+1, после чего производится считывание очередного слова из буфера пересинхронизации. По мере передачи слов В освобождающийся буфер пересинхронизации перемещается слово из буфера передачи. После выборки последнего слова из буфера передачи (буфер передачи пуст) в буфере пересинхронизации остаётся еще два слова. Фактическое окончание передачи можно идентифицировать по состоянию буфера пересинхронизации, либо считав бит TRUN регистра TSR.

Если управляющий сигнал формируется передатчиком, то при считывании последнего слова из буфера пересинхронизации передача останавливается. Передача продолжится только после того как в буфер пересинхронизации снова начнут поступать данные.

Если передатчик использует внешнюю частоту и внешний управляющий сигнал, в целях экономии мощности системная частота может быть установлена меньшей, чем внешняя частота передатчика, однако ее должно быть достаточно для того, чтобы успеть переместить очередное слово в буфер пересинхронизации (за время передачи одного слова должно быть хотя бы три импульса системной частоты ССК). Если внешний сигнал управляющий инициировал передачу слова при пустом буфере пересинхронизации, устанавливается флаг ошибки передачи (TERR), в этом случае ошибочное Если управляющий сигнал передаётся слово. формируется самим



передатчиком, системная частота может быть много меньше частоты передатчика, однако это скажется на скорости передачи данных.

Установка бита TERR в процессе передачи говорит о том, что порт произвел попытку чтения из пустого буфера передачи. Это значит, что передатчиком было передано некорректное слово, кроме того могло быть нарушено состояние указателей в буфере передачи. Продолжать передачу в таком состоянии порта нельзя. В этом случае необходимо произвести выключение порта - установить бит i2s_spi_en в 0, что приведет к сбросу состояния всех буферов порта, после чего программно сбросить бит TERR записью в регистр TSR. После чего можно снова включить порт и продолжать передачу.

В направлении передачи порт обладает буферизацией на 4 32-разрядных слова.

Установка бита SPI_I2S_EN в 0 приведет к программному сбросу передатчика, и все данные находящиеся в буфере передачи будут утеряны.

12.2.15 Тракт приёма данных

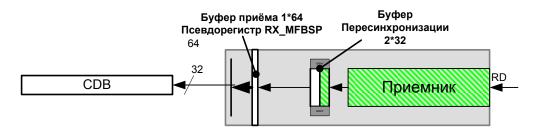


Рисунок 12.16. Тракт приёма данных в режиме SPI

На Рисунок 12.16 представлен тракт передачи данных для режима SPI.

Что бы перевести приёмник в режим готовности необходимо включить последовательный порт (SPI_I2S_EN=1) и приёмник (REN=1), после чего-либо начать ожидание появления прочитанных данных в буфере приёма.

Приёмник принимает последовательные биты, поступающие с внешнего вывода до тех пор, пока число принятых бит не достигнет значения RWORDLEN+1. После этого принятое 32-х разрядное слово (если RWORDLEN<31 незадействованные биты буфер обнуляются) перемещается В пересинхронизации. Запись буфер пересинхронизации направления приёма осуществляется на частоте приёмника RCLK, чтение из буфера пересинхронизации осуществляется на системной частоте CLK. Из буфера пересинхронизации принятое слово автоматически перемещается в буфер приёма, если он не полон. Если в буфере приёма есть хотя бы одно 32-х разрядное слово, то принятые 32-х разрядные слова можно считывать, обращаясь по адресу псевдорегистра RX MFBSP.



Если приёмник использует внешнюю частоту, то в целях экономии мощности системная частота может быть установлена меньшей, чем внешняя частота приёмника, однако ее должно быть достаточно для того, чтобы успеть переместить очередное слово из буфера пересинхронизации (за время приёма одного слова должно быть хотя бы три импульса системной частоты СLK). Если при заполненном буфере пересинхронизации приёмником был произведен приём очередного слова и инициирована попытка записи в буфер пересинхронизации, устанавливается флаг ошибки приёма (RERR), а последнее принятое слово теряется.

Установка бита RERR в процессе передачи говорит о том, что порт произвел попытку записи в полный буфер приёма. Это значит, что принятое слово было потеряно, кроме того могло быть нарушено состояние указателей в буфере приёма. Продолжать приём в таком состоянии порта нельзя. В этом случае необходимо произвести выключение порта - установить бит i2s_spi_en в 0, что приведет к сбросу состояния всех буферов порта, после чего программно сбросить бит RERR записью в регистр RSR. После чего можно снова включить порт и продолжать приём.

В направлении приёма порт обладает буферизацией на 4 32-х разрядных слова.

Установка бита SPI_I2S_EN в 0 приведет к программному сбросу приёмника и все данные находящиеся в буфере приёма будут утеряны.

12.2.16 Прерывания от последовательного порта

Прерывание MFBSP_RXBUF устанавливается, в случае если включен приемник (I2S_SPI_EN=1, REN = 1) и в буфер приёма записано количество слов большее, чем установлено уровнем прерывания RLEV, либо произошла ошибка приема (RERR = 1).

Прерывание MFBSP_TXBUF устанавливается, в случае если включен передатчик $(12S_SPI_EN=1, REN=1)$ и в буфере передачи осталось количество слов меньшее, либо равное чем установлено уровнем прерывания TLEV, либо произошла ошибка передачи (TERR=1).

12.3 Работа MFBSP в режиме порта ввода-вывода общего назначения

Если многофункциональный порт выключен (LEN=0, SPI_I2S_EN =0), внешние лини LDAT[7:0], LCLK, LACK можно использовать как 10-разрядный двунаправленный порт ввода-вывода. В данной микросхеме к внешним выводам GPIO[2:0] подключены только линии LDAT[7:5] соответственно.

Если включен режим последовательного порта (SPI_I2S_EN = 1), незадействованные в организации последовательной передачи данных выводы LDAT[7:4] могут быть



использованы в качестве вводов-выводов общего назначения. Единственным ограничением в данной ситуации является то, что для определения режима работы последовательного порта используются биты LDIR[5:0], которые не должны меняться в процессе передачи данных по последовательному порту. Поэтому при управлении выводами общего назначения LDAT[7:4] (управляются битами DIR_MFBSP [9:6]) запись в регистр DIR_MFBSP необходимо проводить таким образом, чтобы текущие значения бит DIR MFBSP [5:0] не менялись.

При работе в режиме выводов общего назначения данные с внешних выводов порта защелкиваются по положительному фронту тактового сигнала. Поэтому следует учитывать, что чтение данных с внешних выводов порта будет происходить с задержкой в 1 такт.

12.3.1 Регистр данных порта ввода вывода GPIO_DR

10-разрядный регистр данных порта ввода-вывода (GPIO_DR) предназначен для реализации гибкого интерфейса с внешними устройствами. Внешние выводы порта ввода-вывода совмещены с внешними выводами линкового порта.

Соответствие разрядов регистра GPIO_DR и внешних линий линкового порта приведено в Таблица 12.11.

Таблица 12.11. Назначение разрядов регистра GPIO DR

Номер разряда Регистра GPIO_DR	Внешние выводы MFBSP	Значение после сброса
9:2	LDAT[7:0]	0
1	LCLK	0
0	LACK	0

12.3.2 Регистр управления направлением выводов DIR_MFBSP

Настройка направления выводов порта ввода-вывода осуществляется программно при помощи 10-разрядного регистра DIR_MFBSP. Если DIR_MFBSP установлен в 0, то соответствующий разряд порта ввода-вывода является входом, если же разряд DIR_MFBSP установлен в 1, то соответствующий разряд порта ввода-вывода является выходом.

Таблица 12.12. Назначение разрядов регистра DIR MFBSP

Номер разряда Регистра DIR_MFBSP	Внешние выводы MFBSP	Значение после сброса
9:2	Направление выводов	0
	LDAT[7:0]	



Номер разряда Регистра DIR_MFBSP	Внешние выводы MFBSP	Значение после сброса
1	Направление вывода LCLK	0
0	Направление вывода LACK	0



13. NAND FLASH CONTROLLER

13.1 Общие положения

13.1.1 Функциональные параметры и возможности

NAND FLASH CONTROLLER (NFC) имеет следующие функциональные параметры и возможности:

- Cooтветствует стандарту Open Nand Flash Interface Working Group (ONFI) 1.0, 2.0, 2.1 и 2.2;
- Обеспечивает подключение до 8 блоков памяти типа Nand Flash объемом до 128 Гбит каждый;
- Обеспечивает подключение Nand Flash фирм: Micron, Samsung, ST-Micro и других;
- Обеспечивает все обязательные команды Nand Flash, а также некоторые опциональные команды;
- Полный доступ к области «Spare»;
- Обеспечивает скорость передачи данных от 40 Мбайт/с до 200 Мбайт/с;
- Обеспечивает следующие операции интерливинга:
- Page Program Interleaving;
- Copyback Program Interleaving;
- Block Erase Interleaving;
- Read Interleaving;
- Cache Interleaving.
- Обеспечивает Multi LUN/Die операции;
- Обеспечивает изменение адреса строки (Row);
- Обеспечивает Reset LUN;
- Поддерживает размер страницы (Page Size): 512 байт, 2 Кбайт, 4 Кбайт, 8 Кбайт;
- Обеспечивает программирование временных параметров Nand Flash;
- Обеспечивает 4 и 5 адресных циклов;
- Обеспечивает подключение 8-битных асинхронных и синхронных Nand Flash;
- Обеспечивает параллельное подключение двух 8-битных асинхронных Nand Flash;
- Коды исправления ошибок:
- код Хамминга исправление однократных ошибок и обнаружение двукратных ошибок;
- код ВСН исправление 4, 8, 12 или 16 ошибок в секторе размером 512 бит.
- Формирование 8 сигналов преерывания по переходу входных смгналов микросхемы RB[7:0] из 0 в 1.



13.1.2 Состав NFC

NFC имеет следующие интерфейсы и компоненты:

• Nand Flash Интерфейс. Nand Flash Интерфейс устанавливает сигналы управления памяти и обеспечивает обмен в соответствии со стандартами ONFI 1.0, 2.0, 2.1, 2.2. Поддерживается подключение до 16 8-разрядных микросхем памяти (Target). Каждая микросхема память может иметь объём до 128 Гбайт.

Поддерживается как синхронный, так и асинхронный интерфейс для всех временных режимов (0-5).

- 1. Шина CDB Slave, 32 разряда. Через эту шину осуществляется программный доступ CPU к управляющим регистрам NFC.
- 2. AXI Master, 32 разряда. NFC может работать как ведущее устройство на системной шине в режиме MDMA (Master DMA). При операции записи в флэш-память AXI мастер считывает данные из системной памяти и сохраняет их в FIFO. При операции чтения из флэш-памяти AXI мастер считывает данные из FIFO и сохраняет в системную память.
- 3. FIFO данных объемом 4 Кбайт. При операции чтения данные, полученные из флэш-памяти, помещаются в FIFO. А затем считываются из FIFO либо процессором, либо блоком DMA. При операции записи во флэш-память данные предварительно загружаются в FIFO либо процессором, либо блоком DMA. Затем данные из FIFO записываются во флэш-память.
- 4. ECC (Error-Correcting Code). Блок ECC обеспечивает обнаружение и коррекцию ошибок: код Хэмминга и код BCH. Код Хэмминга позволяет исправлять одиночные ошибки и обнаруживать двойные. Код BCH (код Боуза-Чоудхури) поддерживает коррекцию до 32 разрядов.
- 5. Регистры. Процессор управляет конфигурацией и работой NFC через регистры. Конфигурирование включает в себя установку временных параметров, настройку адреса, размера пересылки, режима синхронизации и т.д. Через регистры производится запуск выполнения команд, чтение/запись данных из FIFO данных.

13.1.3 Термины и определения

Страница (Page) - наименьшая адресуемая единица для операций чтения и программирования. Страница состоит из множества байтов или слов. (см. ONFI)

Стриница параметров (Parameter Page). Структура данных во флэш-памяти, которая описывает организацию флэш-памяти, поддерживаемые свойства, временные режимы и другие поведенческие параметры. Страница параметров считывается из флэш-памяти с помощью команды Read Parameter Page. (см. ONFI)

Блок (Block). Блок состоит из множества страниц и является наименьшей адресуемой единицей для операции Erase. (см. ONFI)



Регистр страницы (Page Register). Регистр страницы используется для временного хранения данных до того, как они будут записаны в страницу в массиве флэш-памяти или после их чтения из массива флэш-памяти. (см. ONFI)

LUN - наименьший элемент, который может независимо выполнять команды и сообщать о своём состоянии. LUN содержит как минимум один регистр страницы и массив флэшпамяти. (см. ONFI)

Целевое устройство (Target) - независимый компонент флэш-памяти с собственным сигналом СЕ#. Целевое устройство может состоять из нескольких LUN. (см. ONFI)

Массив флэш (Flash Array). Массив запоминающих ячеек флэш-памяти.

Адрес флэш. Адрес состоит из адреса строки (Row Address) и адреса столбца (Column Address). (см. ONFI)

Адрес строки (Row Address) - адрес LUN, блока и страницы к которой производится обращение. (см. ONFI)

Адрес столбца (Column Address). Адрес столбца определяет байт или слово внутри страницы, к которой производится обращение. (см. ONFI)

Interleaved/Multiplane onepaции. При interleaved/multi-plane операциях множество команд одного типа может одновременно выполнятся над различными блоками одного LUN. (см. ONFI)

NFC поддерживает следующие multi-plane операции:

- Page Program;
- Copyback Program;
- Block Erase;
- Read:
- Page Program.

Пакем - порция данных, которую считывает или записывает NFC во флэш-память за одну пересылку. Количество байт внутри пакета и общее количество пакетов настраивается программно.

Spare bytes/область памяти. Дополнительная резервная область страницы. В данной области размещаются контрольные данные, позволяющие обнаруживать и корректировать ошибки в основной области.



13.2 Регистры NFC

13.2.1 Перечень регистров NFC

Перечень регистров NFC приведен в Таблица 13.1.

Таблица 13.1. Перечень регистров NFC

Обозначение	Смещение	Доступ	Название
PACKET	0x0	RW	Регистр настройки пакетов
MEMADDR1	0x4	RW	Регистр адреса 1
MEMADDR2	0x8	RW	Регистр адреса 2
COMMAND	0xC	RW	Регистр команды
PROGRAM	0x10	RW	Регистр программы
INTERRUPT_STATUS_EN	0x14	RW	Регистр разрешения статуса прерываний
INTERRUPT_SIGNAL_EN	0x18	RW	Регистр разрешения сигнала прерываний
INTERRUPT_STATUS	0x1C	RW	Регистр статуса прерываний
ID1	0x20	RO	Регистр ID1
ID2	0x24	RO	Регистр ID2
FLASH_STATUS	0x28	RO	Регистр статуса флэш-памяти
TIMING	0x2C	RW	Регистр временных параметров
BUFFER_DATA	0x30	RW	Регистр данных буфера
ECC	0x34	RW	Регистр ЕСС
ECC_ERROR_CNT	0x38	RO	Счётчик ошибок ЕСС
ECC_SPARE_CMD	0x3c	RW	Регистр команд ЕСС
ERROR_CNT_1BIT	0x40	RW	Счётчик 1-битных ошибок
ERROR_CNT_2BIT	0x44	RW	Счётчик 2-битных ошибок
ERROR_CNT_3BIT	0x48	RW	Счётчик 3-битных ошибок
ERROR_CNT_4BIT	0x4C	RW	Счётчик 4-битных ошибок
DMA_ADDR	0x50	RW	Регистр адреса DMA
DMA_BUFFER_BOUNDA RY	0x54	RW	Регистр границы буфера DMA
RESERVED	0x58	RO	Зарезервировано
ERROR_COUNT_5BIT	0x5c	RW	Счётчик 5-битных ошибок
ERROR_COUNT_6BIT	0x60	RW	Счётчик 6-битных ошибок
ERROR_COUNT_7BIT	0x64	RW	Счётчик 7-битных ошибок
ERROR_COUNT_8BIT	0x68	RW	Счётчик 8-битных ошибок
WRITE_PROTECT	0x6C	RW	Регистр управления защитой памяти
RESERVED	0x70-0xFF	RO	Не используется



При чтении и записи регистров используются 32-разрядные данные.

13.2.2 Регистр РАСКЕТ

Формат регистра РАСКЕТ приведен в Таблица 13.2.

Таблица 13.2. Формат регистра РАСКЕТ

Поле	Биты	Значение по сбросу	Описание
packet_size	10:0	0x200	Размер пакета. Типичный размер пакета в режиме коррекции ошибок ВСН 4, 8, 12 и 16 бит – 512 байт.
			Типичный размер пакета в режиме коррекции ошибок ВСН 24 бит - 1024 байт
reserved	11	0	Зарезервировано.
packet_count	23:12	0	Количество пакетов. 0x1 – 1. 0x800 - 2048
page_count	31:24	0	Количество страниц: 0x00 – 1: 0xFF - 256

В Таблица 13.3 показаны особенности настройки регистра РАСКЕТ в зависимости от способа подключения памяти, выполняемой команды и выбранного временного режима (см. «Переключение временных режимов»).

Таблица 13.3. Настройка полей регистра РАСКЕТ в зависимости от способа подключения памяти, выполняемой команды и выбранного временного режима

Режим	Команды	Packet_size (послед.)	Packet_size (паралл.)	Packet_count
Асинхронный	Set Features, Get Features, Read ID	4	8	1
Асинхронный	Read Status, Read Status Enhanced	1	2	1
Синхронный	Set Features, Get Features, Read ID	8	16	1
Синхронный	Read Status, Read Status Enhanced	2	4	1

Для команд Set Features, Get Features, Read Status, Read Status Enhanced, Read ID в синхронном режиме байты данных повторяются дважды.



13.2.3 Регистр MEMADDR1

Формат регистра MEMADDR1 приведен в Таблица 13.4.

Таблица 13.4. Формат регистра MEMADDR1

Поле	Биты	Значение по сбросу	Описание
column_addr	15:0	0	Адрес столбца (разряды 15:13 должны быть нулями)
page_addr	22:16	0	Адрес страницы.
lsb_block_addr	31:23	0	Адрес блока (младшие разряды). Разряды [24:23] — имеют особое значение для interleaved/mulit-plane операций. 2'b00 — адрес (plane 0). 2'b01 — адрес (plane1). 2'b10 — адрес (plane 2). 2'b11 — адрес (plane 3).

Примечание. Для команд Block Erase и Read Status Enhanced регистр MEMADDR1 должен быть запрограммирован следующим образом:

```
Разряды [6:0] – адрес страницы.
```

Разряды [8:7] – выбор одного из чередующихся адресов:

2'b00 – адрес (plane 0).

2'b01 – адрес (plane1).

2'b10 – адрес (plane 2).

2'b11 – адрес (plane 3).

Разряды [17:9] – адрес блока.

Разряды [18] – выбор логического устройства (LUN).

1'b0 - выбран LUN0.

1'b1 - выбран LUN1.

Разряды [31:19] – должны быть равны нулю.

13.2.4 Peгистр MEMADDR2

Формат регистра MEMADDR2 приведен в Таблица 13.5.

Таблица 13.5. Формат регистра MEMADDR2

Поле	Биты	Значение по сбросу	Описание
msb_block_addr	1:0	0	Адрес блока (старшие разряды).
lun_addr	7:2	0	Адрес LUN. 0x0 – выбор LUN0. 0x1 – выбор LUN1



Поле	Биты	Значение по сбросу	Описание
reserved	15:8	0	Зарезервировано
ncsf	23:16	0	Управление выходами микросхемы nCSF[7:0]: код на этих выходах соответствует инверсному коду, содержащемуся в поле ncsf
reserved	24	0	Зарезервировано
nfc_bch_mode	27:25	0	Режим работы ВСН: 3'b000 – 16-битный код ЕСС 3'b001 – 12-битный код ЕСС 3'b010 – 8-битный код ЕСС 3'b011 – 4-битный код ЕСС 3'b100 – 24-битный код ЕСС
mode	29:28	0	Режим работы с блоками флэш-памяти: 2'b00 – Последовательный; 2'b01 – Параллельный; 2'b10 – Зарезервировано; 2'b11 - Зарезервировано
chip_select	30	0	Выбор формирования сигналов управления флэш-памяти: 0 –формируются сигналы ALE0, CLE0, nWE0, nRE0; 1 – формируются сигналы ALE1, CLE1, nWE1, nRE1. При параллельном способе работы с блоками флэш-памяти формируются сигналы ALE0, CLE0, nWE0, nRE0 и ALE1, CLE1, nWE1, nRE1
reserved	3	0	Зарезервировано

Примечание. Для команд Block Erase и Read Status Enhanced биты [7:0] регистра MEMADDR2 должны быть равны нулю.

NFC при работе в режиме последовательного подключения NAND (mode=0 регистра MEMADDR2) позволяет подключить 16 микросхем NAND. При этом микросхемы разбиты на 8 пар с номерами с 7 по 0. К каждой из 8 пар подсоединяются сигналы nCSF, с номером соответствующей номеру пары. К первой микросхеме каждой пары подключаются сигналы управления ALE0, CLE0, nWE0, nRE0, а ко второй - сигналы управления ALE1, CLE1, nWE1, nRE1. ALE0, CLE0, nWE0, nRE0 или ALE1, CLE1, nWE1, nRE1 выбирается разрядом ctr_select регистра MEMADDR2. Таким образом, выбирая соответствующие сигналы nCSF и ALE, CLE, nWE, nRE1 можно обменяться данными с любой из 16 микросхем NAND.

Сигналы готовность/занятость RB каждой пары NAND объединяются и подключаются к NFC – RB[7:0]. При переходе состояния пары RB из 0 в 1 (признак окончания операции



обмена данными) NFC формирует соответствующее прерывание RB_NFC. Наличие этих прерываний позволяет запись данных в NAND осуществлять в два этапа. 1 этап - программирование NFC с его DMA и запуск последнего, после чего выполняется запись данных в NAND и программирование NAND. 2 этап (по соответствующему прерыванию RB_NFC) — проверка правильности программирования NAND посредством считывания регистра статуса NAND.

13.2.5 Регистр COMMAND

Формат регистра COMMAND приведен в Таблица 13.6.

Таблица 13.6. Формат регистра COMMAND

Поле	Биты	Значение по сбросу	Описание
command1	7:0	8'h0	Код операции первой команды для флэш-памяти.
command2	15:8	8'h0	Код операции второй команды для флэш-памяти.
asyn_modes	18:16	3'd0	Асинхронный режим. 3'b000- режим 0. 3'b001- режим 1. 3'b010- режим 2. 3'b011- режим 3. 3'b100- режим 4. 3'b101- режим 5. 3'b110- зарезервировано.
asyn_syn	19	1'd0	Выбор режима синхронизации 1'd0 - асинхронный режим. 1'd1 – синхронный режим.
syn_modes	22:20	3'd0	Синхронный режим. 3'b000- режим 0. 3'b001- режим 1. 3'b010- режим 2. 3'b011- режим 3. 3'b100- режим 4. 3'b101- режим 5. 3'b110- зарезервировано.
page_size	25:23	3'd2	Размер страницы. 3'd0 – 512 байт. 3'd1 – 2К. 3'd2 – 4К. 3'd3 - 8К. 4-7 - Зарезервировано.



Поле	Биты	Значение по сбросу	Описание
dma_enable	27:26	2'd0	Режим обмена данными. 0 - режим РІО; 1 - зарезервировано; 2 - режим МDMA; 3 - режим MDMA с проверкой статуса при выполнении программирования.
num_addr_cycles	30:28	3'd0	Число циклов адреса в команде для флэш-памяти. 3'd0 — Не используется. 3'd1 — Один цикл адреса. 3'd2- Два цикла адреса 3'd7 — Семь циклов адреса.
ecc_on_off	31	0	Включение/отключение коррекции ошибок. 1 — ECC включён. 0 - ECC выключен.

Примечание. Значение поля **num_addr_cycles** должно быть равно:

- 0 для команд reset, synchronous reset, read_status (см. регистр PROGRAM);
- 1 для команд read_parameter, get_feature, set_feature, read_id (см. регистр PROGRAM);
- 3 для команд erase, read status enhanced, reset lun (см. регистр PROGRAM);
- определяется значением байта «101» страницы параметров для команд read, write. Например, 101-й байт содержит значения column address cycle [7:4] = 2 и row address cycle [3:0] = 3, в этом случае процессор должен записать в поле num addr cycles значение 5 (column address cycle + row address cycle).

Примечание. Значение полей **asyn_modes**, **asyn_syn**, **syn_modes**, **page_size** настраивается в соответствии со значением **Страницы Параметров**. Изначально флэш-память работает в асинхронном режиме 0. После смены временного режима (после выполнения команды Set Features) необходимо настроить эти поля в соответствии с выбранным временным режимом.

Примечание. Для команд, требующих пересылок данных (Read, Program Page и др.), существует несколько способов обмена данными:

- в режиме PIO (Processor IO) пересылки данных выполняются процессором;
- в режиме MDMA (Master DMA) пересылки данных выполняются контроллером, при минимальном участии процессора.



13.2.6 Регистр PROGRAM

Через регистр PROGRAM производится инициирование выдачи команд для флэш-памяти. Запись единицы в необходимый разряд вызывает выполнение соответствующей команды. После завершения команды регистр PROGRAM сбрасывается и устанавливается прерывание «Transfer Complete».

Формат регистра PROGRAM приведен в Таблица 13.7.

Таблица 13.7. Формат регистра PROGRAM

Поле	Биты	Значение по сбросу	Описание
read	0	1'b0	Операция чтения из флэш-памяти.
multi_die	1	1'b0	Признак Interleaved/Multiplane операции (Page Program, Read, Block Erase).
block_erase	2	1'b0	Операция Block Erase.
read_status	3	1'b0	Операция Read Status.
page_program	4	1'b0	Операция Page Program.
multi_die_rd	5	1'b0	Операция чтения из разных LUN.
read_id	6	1'b0	Операция Read ID.
read_parameter_page	7	1'b0	Операция Read Parameter Page.
reset	8	1'b0	Операция Reset.
get_features	9	1'b0	Операция Get Features.
set_features	10	1'b0	Операция Set Features.
read_unique_id	11	1'b0	Операция Read Unique ID.
read_status_enhanced	12	1'b0	Операция Read Status Enhanced.
read_interleaved	13	1'b0	Операция Interleaved/Multiplane чтения.
change_read_column_enhance	14	1'b0	Операция Change Read Column Enhanced.
copy_back_interleaved	15	1'b0	Операция Copy Back Interleaved.
read_cache_start	16	1'b0	Контроллер выполняет операцию Read.
read_cache_sequential	17	1'b0	Операция Read Cache Sequential.
read_cache_random	18	1'b0	Операция Read Cache Random.



Поле	Биты	Значение по сбросу	Описание
read_cache_end	19	1'b0	Операция Read Cache End.
small_data_move	20	1'b0	Операция Small Data Move.
change_row_addr	21	1'b0	Операция Change Row Address.
change_row_addr_end	22	1'b0	1'b1 - контроллер завершает операцию Change Row Address.
reset_lun	23	1'b0	1'b1 - контроллер выполняет операцию сброса заданного логического устройства (LUN)
pgm_pg_reg_clr	24	1'b0	1'b1 - контроллер выполняет операцию с включённым свойством Page Register Clear Enhancement.
disable_busy	25	0	признак выполнения операции без ожидания готовности памяти к выполнению следующей операции
read_data	26	0	1'b1 - контроллер выполняет чтение регистра страницы флэш-памяти
reserved	31:27	0	Зарезервировано

Примечания:

- 1. При запуске команды разрешается устанавливать только один разряд в регистре PROGRAM (см. исключения ниже). После установки прерывания «Transfer Complete» регистр PROGRAM обнуляется. В случае одновременной установки нескольких разрядов поведение NFC предсказать невозможно.
- 2. Операции, для которых допускается установка двух разрядов в регистре PROGRAM:
 - 2.1. Multi die Page Program одновременно устанавливаются биты multi_die и page_program.
 - 2.2. Multi die Read одновременно устанавливаются биты multi_die и read.
 - 2.3. Multi die Block Erase одновременно устанавливаются биты multi_die и block_erase.
 - 2.4. Change Row Address. Для выполнения этой операции необходимо установить разряды change_row_addr и page_program.
 - 2.5. Для памяти flash с включённым свойством «Program Page Register Clear Enhancement» используется увеличенное время tADL. Для использования увеличенного времени tADL необходимо установить



pgm_pg_reg_clr. Это увеличение времени применяется только к последовательностям команд Program (80h).

2.6. Бит disable_busy может устанавливаться только с битами reset, reset_lun, block_erase, page_program,, read, read_parameter_page.

Для команд reset, reset_lun, block_erase, page_program с включенным признаком disable_busy прерывание «Transfer Complete» устанавливается при установке в 0 соответствующего сигнала RB[i] (завершение операции в контроллере). В этом случае NFC может выполнять операции с другими микросхемами NAND. Признаком завершения операции в NAND памяти является установка прерывания RB_NFC[i].

Для команд reset, reset_lun, block_erase, page_program с выключенным признаком disable_busy прерывание «Transfer Complete» устанавливается при переходе соответствующего сигнала RB[i] из 0 в 1 (завершение операции в NAND памяти).

Для операций **read, read_parameter_page** с включенным признаком disable_busy прерывание «Transfer Complete» устанавливается при установке низкого уровня соответствующего сигнала RB[i] (завершение операции в контроллере). В этом случае NFC может выполнять операции с другими микросхемами NAND. Признаком завершения операции в NAND памяти является установка прерывания RB NFC[i].

После установки прерывания RB_NFC[i] необходимо выполнить команду **read_data**. Прерывание «Transfer Complete» устанавливается после считывания всех байтов из FIFO.

- 3. Для команд запущенных битами read и read_cache_start выполняется обычная операция чтения. Для команды чтения, запущенной разрядом read, прерывание «Transfer Complete» установится после считывания всех байтов данных из флэш-памяти. Для команды чтения, запущенной разрядом read_cache_start, прерывание «Transfer Complete» установится после получения готовности от флэш-памяти к считыванию данных из неё.
- 4. Разряд page program используется для следующих команд:
 - 4.1. Page Program (80h-10h).
 - 4.2. Page Program Interleaving (80h-11h).
 - 4.3. Page Cache Program (80h-15h).



13.2.7 Peructp INTERRUPT_STATUS_EN

Формат регистра INTERRUPT_STATUS_EN приведен в Таблица 13.8.

Таблица 13.8. Формат регистра INTERRUPT_STATUS_EN

Поле	Биты	Значение по сбросу	Описание
buff_wr_rdy_sts_en	0	1'b0	Разрешение признака прерывания «Buffer Write Ready» в регистре INTERRUPT_STATUS (разряд 0). 1'b1 — разрешено. 1'b0 — запрещено.
buff_rd_rdy_sts_en	1	1'b0	Разрешение признака прерывания «Buffer Read Ready» в регистре INTERRUPT_STATUS (разряд 1). 1'b1 — разрешено. 1'b0 — запрещено.
trans_comp_sts_en	2	1'b0	Разрешение признака прерывания «Transfer Complete» в регистре INTERRUPT_STATUS (разряд 2). 1'b1 — разрешено. 1'b0 — запрещено.
mul_bit_err_sts_en	3	1'b0	Разрешение признака прерывания «Multi Bit Error» в регистре INTERRUPT_STATUS (разряд 3). 1'b1 — разрешено. 1'b0 — запрещено. Значение данного поля используется при коррекции ошибок по алгоритму Хемминга (SLC), иначе заменяется нулем.
err_intrpt_sts_en	4	1'b0	Разрешение признака прерывания «Single Bit Error» в режиме SLC или «BCH Detect Error» в режиме MLC в регистре INTERRUPT_STATUS (разряд 4). 1'b1 — разрешено. 1'b0 — запрещено.
err_program_sts_en	5	0	Разрешение признака прерывания «err_program» в регистре INTERRUPT_STATUS (разряд 5): 1 — разрешено; 0 — запрещено



Поле	Биты	Значение по сбросу	Описание
dma_int_sts_en	6	1'b0	Разрешение признака прерывания «dma_int» в регистре INTERRUPT_STATUS (разряд 6). 1'b1 — разрешено. 1'b0 — запрещено. Используется только при пересылке в режиме MDMA, иначе заменяется нулем.
reserved	7	1'b0	Зарезервировано.
reserved	31:8	24'h0	Зарезервировано.

13.2.8 Регистр INTERRUPT_SIGNAL_EN

Формат регистра INTERRUPT_SIGNAL_EN приведен в Таблица 13.9.

Таблица 13.9. Формат регистра INTERRUPT_SIGNAL_EN

Поле	Биты	Значение по сбросу	Описание
buff_wr_rdy_sig_en	0	1'b0	Разрешение установки сигнала прерывания по событию «Buffer Write Ready». 1'b1 — разрешено. 1'b0 — запрещено.
buff_rd_rdy_sig_en	1	1'b0	Разрешение установки сигнала прерывания по событию «Buffer Read Ready». 1'b1 — разрешено. 1'b0 — запрещено.
trans_comp_sig_en	2	1'b0	Разрешение установки сигнала прерывания по событию «Transfer Complete». 1'b1 — разрешено. 1'b0 — запрещено.
mul_bit_err_sig_en	3	1'b0	Разрешение установки сигнала прерывания по событию «Multi Bit Error». 1'b1 — разрешено. 1'b0 — запрещено. Значение данного поля используется при коррекции ошибок по алгоритму Хемминга (SLC), иначе заменяется нулем.
err_intrpt_sig_en	4	1'b0	Разрешение установки сигнала прерывания по событию «Single Bit Error» в режиме SLC или «BCH Detect Error» в режиме MLC. 1'b1 — разрешено. 1'b0 — запрещено.



Поле	Биты	Значение по сбросу	Описание
err_program_sig_en	5	0	Разрешение установки сигнала прерывания «err_program» в режиме MDMA: 1 – разрешено; 0 – запрещено
dma_int_sig_en	6	1'b0	Разрешение установки сигнала прерывания по событию «dma_int». 1'b1 — разрешено. 1'b0 — запрещено. Используется только при пересылке в режиме MDMA, иначе заменяется нулём.
reserved	7	1'b0	Зарезервировано.
reserved	31:8	24'h0	Зарезервировано.

13.2.9 Регистр INTERRUPT_STATUS

Формат регистра INTERRUPT_STATUS приведен в Таблица 13.10.

Таблица 13.10. Формат регистра INTERRUPT_STATUS

Поле	Биты	Значение по сбросу	Описание
buff_wr_rdy_reg	0	1'b0	«Buffer Write Ready». Готовность FIFO к приёму данных от процессора. Устанавливается при условии, что в буфере достаточно места для получения блока данных от процессора.
buff_rd_rdy_reg	1	1'b0	«Buffer Read Ready». Готовность FIFO к передаче данных процессору. Устанавливается при условии, что считано достаточно данных из флэш-памяти.
trans_comp_reg	2	1'b0	«Transfer Complete». Устанавливается после выполнения заданной команды.
mul_bit_err_reg	3	1'b0	«Multi Bit Error». Устанавливается при условии, что подтверждается ошибка в нескольких битах. Используется только в режиме SLC, иначе считается нулём.
err_intrpt_reg	4	1'b0	«Single Bit Error»/«ВСН Detect Error». Устанавливается при условии, что обнаруживается ошибка ВСН в МLС или одиночная ошибка в SLC режиме.



Поле	Биты	Значение по сбросу	Описание
err_program_reg	5	0	Устанавливается при обнаружении ошибки программирования после выполнения команды page_program в режиме MDMA
dma_int_reg	6	1'b0	Установка бита происходит при достижении границы буфера в режиме MDMA. Используется только при пересылке в режиме MDMA, иначе заменяется нулём.
status_RB	15:8	-	Порт входных сигналов RB
reserved	31:16	0	Зарезервировано

Примечание. Для сброса статуса прерывания нужно записать единицу в соответствующий разряд регистра INTERRUPT STATUS.

13.2.10 Регистр ID1

Хранит результат выполнения команды Read ID.

Формат регистра ID1 приведен в Таблица 13.11.

Таблица 13.11. Формат регистра ID1

Поле	Биты	Значение по сбросу	Описание
device_id1	31:0	32'h0	Регистр идентификатора (ID) флэш- памяти. (младшие разряды)

13.2.11 Регистр ID2

Хранит результат выполнения команды Read ID.

Формат регистра ID2 приведен в Таблица 13.12.

Таблица 13.12. Формат регистра ID2

Поле	Биты	Значение по сбросу	Описание
device_id2	7:0	8'h0	Регистр идентификатора (ID) флэш-памяти. (старшие 8 разрядов).
reserved	31:8	24'h0	Зарезервировано

Примечание. При выполнении команды Read ID с адресом 0x20 (ONFI ID) считанные из флэш-памяти четыре байта сохраняются в ID1[31:0].



При выполнении команды Read ID с адресом 0x00 (JEDEC ID) считанные из флэш-памяти пять байтов сохраняются в $\{ID2[7:0], ID1[31:0]\}$.

13.2.12 Регистр FLASH_STATUS

Хранит результат выполнения команды Read Status.

Формат регистра FLASH STATUS приведен в Таблица 13.13.

Таблица 13.13. Формат регистра FLASH STATUS

Поле	Биты	Значение по сбросу	Описание
flash_status	15:0	16'h0	Значение статуса флэш-памяти.
reserved	31:16	16'h0	Зарезервировано

Примечание. При выполнении команды Read Status результат команды записывается в регистр FLASH_STATUS, устанавливается прерывание «Transfer Complete». Только после этого процессору следует прочитать регистр FLASH_STATUS.

Примечания

При последовательном способе работы с флэш-памятью:

- разряды FLASH_STATUS [7:0] содержат значение статуса подключённого флэш-устройства;
- разряды FLASH STATUS[15:8] установлены в ноль.

При параллельном способе работы с флэш-памятью:

- разряды FLASH_STATUS [7:0] содержат значение статуса значение статуса первого флэш-устройства;
- разряды FLASH_STATUS [15:8] содержат значение статуса значение статуса второго флэш-устройства.

13.2.13 Регистр ТІМІNG

Perucтр TIMING содержит временные параметры памяти NAND Flash.

Формат регистра TIMING приведен в Таблица 13.14.



Таблица 13.14. Формат регистра TIMING

Поле	Биты	Значение по сбросу	Описание
tccs_time	1:0	2'd0	tccs (время установки после смены столбца, см. ONFI). 2'b00 - 500 ns 2'b01 - 100 ns 2'b10 - 200 ns 2'b11 - 300 ns Поле tccs time настраивается на основе значений байтов [139:140] страницы параметров.
slow_fast_tc ad	2	1'b0	tcad (Command, Address, Data delay, см. ONFI). 0 — медленное устройство (tCADs). 1 — быстрое устройство (tCADf). Поле slow_fast_tcad настраивается на основе значения байта [143] страницы параметров.
dqs_buff_sel	6:3	4'h6	Задержка двунаправленного сигнала dqs. Входной и выходной буфера двунаправленного сигнала dqs имеют настраиваемую задержку. Поле dqs_buff_sel позволяет изменять время задержки с шагом в 370-480 пикосекунд. Для вычисления задержки сигнала dqs необходимо умножить значение поля dqs_buff_sel на шаг задержки. Для операций чтения необходимо настроить значение данного поля, основываясь на значении tDQSQ (см. ONFI). Для операций записи необходимо настроить значение данного поля, основываясь на значении tDS (см. ONFI). Рекомендуемые значения dqs_buff_sel для операции чтения данных: 4'h6 — синхронный режим 0 4'h4 — синхронный режим 2 и 3 4'h2 — синхронный режим 4 и 5 Рекомендуемые значения dqs_buff_sel для операции записи данных: 4'h6 — синхронный режим 0 и 1 4'h2 — синхронный режим 0 и 1 4'h2 — синхронный режим 0, 3, 4 4'h1 — синхронный режим 5
tadl_time	14:7	7'd0	tadl (см. ONFI). Время между циклом адреса и загрузкой данных. Данное поля должно быть запрограммировано при операции программирования флэш-памяти, если установлен бит pgm_pg_reg_clr. Значения поля tadl_time определяется значениями байтов [154:155] страницы параметров



Поле	Биты	Значение по сбросу	Описание
reserved	31:15	25'h0	Зарезервировано

13.2.14 Peructp BUFFER_DATA

Формат регистра BUFFER_DATA приведен в Таблица 13.15.

Таблица 13.15. Формат регистра BUFFER DATA

Поле	Биты	Значение по сбросу	Описание
data_port	31:0	-	Порт доступа к данным внутреннего FIFO (буфера). Через данный регистр производится чтение/запись FIFO контроллера.

13.2.15 Регистр ЕСС

Регистр используется при включённом режиме обнаружения и коррекции ошибок. Регистр настраивает способ коррекции, адрес и размер области флэш-памяти, где находятся данные ECC.

Формат регистра ЕСС приведен в Таблица 13.16.

Таблица 13.16. Формат регистра ЕСС

Поле	Биты	Значение по сбросу	Описание
ecc_addr	15:0	16'h104c	Используется для формирования адреса ECC с командами Change Read Column, Change Write Column.
ecc_size	24:16	9'h34	Задаёт количество транзакций записи или чтения для области памяти с данными ЕСС.
slc_mlc	25	1'b1	Выбор способа коррекции. 1'b0 - SLC 1'b1 - MLC
reserved	31:26	6'h0	Зарезервировано.



Таблица 13.17. Регистр содержимого поля есс_addr в зависимости от размера страницы и режима ECC

Размер Страницы\Ре жим ЕСС	SLC	MLC-4	MLC-8	MLC-12	MLC-16
512	16'h 020d	16'h 0209	16'h 0203	-	-
2k	16'h 0834	16'h 0826	16'h 080C	16'h 0822	16'h 0808
4k	16'h 1068	16'h 104C	16'h 1018	16'h 1044	16'h 1010
8k	16'h 20d0	16'h 2098	16'h 2030	16'h 2088	16'h 2020

Таблица 13.18. Регистра содержимого поля есс_size в зависимости от размера страницы и режима ECC (последовательное соединение флэш-памяти)

Размер Страницы\Ре жим ЕСС	SLC	MLC-4	MLC-8	MLC-12	MLC-16
512	9'h 003	9'h 007	9'h 00D	-	-
2k	9'h 00C	9'h 01A	9'h 034	9'h 04E	9'h 068
4k	9'h 018	9'h 034	9'h 068	9'h 09C	9'h 0D0
8k	9'h 030	9'h 068	9'h 0D0	9'h 138	9'h 1A0

Таблица 13.19. Регистр содержимого поля есс_size в зависимости от размера страницы и режима ЕСС (параллельное соединение флэш-памяти)

Размер Страницы\Ре жим ЕСС	SLC	MLC-4	MLC-8	MLC-12	MLC-16
512	9'h 006	9'h 00E	9'h 00A	-	-
2k	9'h 018	9'h 034	9'h 068	-	9'h 0D0
4k	9'h 030	9'h 068	9'h 0D0	-	9'h 1A0
8k	9'h 060	9'h 0D0	9'h 1A0	-	-



Таблица 13.20

Поле	Биты	Значение по сбросу	Описание
packet_bound_err_count	7:0	8'b0	Счётчик ошибок при чтении пакета данных. Процессор должен считывать значение данного регистра каждый раз после чтения данных размером packet_size.
page_bound_err_count	16:8	0	Общий счётчик ошибок при чтении страницы. Процессор должен считывать значение данного регистра после прочтения каждой страницы.
reserved	23:17	0	Зарезервировано
packet_bound_err_max	31:24	0	Максимальное количество ошибок при чтении пакета данных

13.2.16 Peructp ECC_ERROR_CNT

Формат регистра ECC ERROR CNT приведен в Таблица 13.21.

Таблица 13.21. Формат регистра ECC ERROR CNT

Содержимое данного регистра может быть считано при обработке прерывания «Transfer Complete» для следующих операций: read (c dis_busy = 0), read_data, multi_die_rd, change read column enhanced, read cache sequential, read cache random, read cache end.

Регистр сбрасывается при записи в регистр PROGRAM.

13.2.17 Peructp ECC_SPARE_CMD

Формат регистра ECC_SPARE_CMD приведен в Таблица 13.22.

Таблица 13.22. Формат регистра ECC_SPARE_CMD

Поле	Биты	Значение по сбросу	Описание
ecc_spare_cmd	15:0	16'h 00	Используется для программирования кода операций ЕСС, которые выполняются совместно с командами Change Write Column, Change Read Column.



Поле	Биты	Значение по сбросу	Описание
reserved	27:16	12'h 00	Зарезервировано
ecc_spare_num_addr_c ycles	30:28	3'b000	Число циклов адреса в команде ЕСС для флэш-памяти. 3'b000 – Не используется 3'b001 – Один цикл 3'b111 — Семь циклов
reserved	31	1'b0	Зарезервировано

13.2.18 Регистры ERROR_CNT_nBIT

Регистры ERROR_CNT_1BIT - ERROR_CNT_8BIT содержат количество 1-битных – 8-битных ошибок соответственно.

Формат регистра ERROR_CNT_nBIT приведен в Таблица 13.23.

Таблица 13.23. Формат регистров ERROR_CNT_nBIT, где n от 1 до 8

Поле	Биты	Значение по сбросу	Описание
error_count_n_bit	31:0	32'h 00	Количество обнаруженных n-битных ошибок. При необходимости сбрасывается процессором.



13.2.19 Регистр DMA_ADDR

Формат регистра DMA_ADDR приведен в Таблица 13.24.

Таблица 13.24. Формат регистра DMA_ADDR

Поле	Биты	Значение по сбросу	Описание
dma_addr	31:0	32'h 00	Используется только в режиме MDMA. Содержит адрес системной памяти для DMA-передачи. После прекращения DMA-передачи указывает на область данных, следующую за обработанной. Доступ к регистру возможен только когда не выполняется передача (т.е. после её завершения). Чтение регистра в момент передачи возвращает недостоверное значение. Процессор должен инициализировать данный регистр перед началом DMA-передачи. При достижении границы, заданной в регистре DMA_BUFFER_BOUNDARY, контроллер генерирует прерывание. Процессор должен обработать этот запрос и настроить адрес следующей транзакции.



13.2.20 Регистр DMA BUFFER BOUNDARY

Формат регистра DMA_BUFFER_BOUNDARY приведен в Таблица 13.25.

Таблица 13.25. Формат регистра DMA BUFFER BOUNDARY

Поле	Биты	Значение по сбросу	Описание
dma_buffer_size	2:0	3'h0	Данное поле определяет размер непрерывного буфера данных в системной памяти: 3'b000 - 4КВ 3'b001 - 8КВ 3'b010 - 16КВ 3'b011 - 32КВ 3'b100 - 64КВ 3'b101 - 128КВ 3'b110 - 256КВ 3'b111 - 512КВ Достижение границы для каждого из размеров буфера определяется по изменению соответствующего разряда адреса (12, 13, 14, 15, 16, 17, 18, 19). При достижении границы буфера контроллер устанавливает прерывание DMA, по которому процессор должен обновить регистр DMA_SYS_ADDR

13.2.21 Peructp WRITE_PROTECT

Perucтp WRITE_PROTECT предназначен для управления внешним выводом nWP.

Формат регистра WRITE PROTECT приведен в Таблица 13.26.

Таблица 13.26. Формат регистра WRITE PROTECT

Поле	Биты	Значение по сбросу	Описание
nwp	0	1'b0	Порт выходного сигнала nWP: 0— на выводе nWP устанавливается низкий уровень сигнала (защита памяти включена); 1— на выводе nWP устанавливается высокий уровень сигнала (защита памяти отключена)
reserved	31:1	31'b0	Зарезервировано.



При изменении режима защиты памяти следует руководствоваться рекомендациями производителя. Так, при использовании микросхем MT29F8G08ABABA, необходимо дождаться завершения очередной командной последовательности, проверить готовность памяти и записать в регистр WRITE PROTECT требуемый режим защиты памяти.

Необходимую паузу tWW NFC не контролирует.

13.3 Описание выполнения операций с флэш-памятью

13.3.1 Схема выполнения команды «Без доступа к FIFO»

1. NFC выдаёт команду флэш-устройству.

Для команды, не требующей пересылок данных через FIFO, обычно выполняется следующая последовательность действий:

Пункты 1-7 — инициирование команды

- 2. CPU разрешает формирование признака прерывания «Transfer Complete» (см. регистр INTERRUPT STATUS EN).
- 3. CPU разрешает формирование сигнала прерывания (см. регистр INTERRUPT SIGNAL EN).
- 4. CPU настраивает регистр COMMAND.
- 5. CPU настраивает регистр MEMADDR1.
- 6. CPU настраивает регистр MEMADDR2.
- 7. CPU инициирует выполнение команды установкой необходимого бита в регистре PROGRAM.

Пункты 8-14 — завершение отработки команды

- 8. CPU ожидает прерывание «Transfer Complete».
- 9. По готовности флэш-памяти и завершении выполнения команды NFC устанавливает признак прерывания «Transfer Complete» в регистре INTERRUPT_STATUS.
- 10. Формируется физический сигнал прерывания.
- 11. CPU считывает регистр INTERRUPT STATUS.
- 12. CPU сбрасывает бит «Transfer Complete» регистра INTERRUPT SIGNAL EN.
- 13. CPU сбрасывает бит «Transfer Complete» регистра INTERRUPT STATUS EN.
- 14. CPU сбрасывает бит «Transfer Complete» в регистре INTERRUPT STATUS.

Можно использовать альтернативный метод без использования сигнал прерывания. В этом случае разрешать формирование физического сигнала прерывания в регистре INTERRUPT_SIGNAL_EN не следует. Вместо этого CPU должен опрашивать регистр INTERRUPT STATUS и проверять состояние разряда «Transfer complete».



13.3.2 Схема выполнения команды «С чтением из FIFO»

Для команды, требующей от CPU чтения из FIFO, обычно выполняется следующая последовательность действий:

Пункты 1-8 – инициирование команды

- 1. CPU разрешает формирование признака прерывания «Buffer Read Ready» (см. регистр INTERRUPT STATUS EN).
- 2. CPU разрешает формирование сигнала прерывания «Buffer Read Ready» (см. регистр INTERRUPT_SIGNAL_EN).
- 3. СРИ настраивает регистр СОММАND.
- 4. CPU настраивает регистр MEMADDR1.
- 5. CPU настраивает регистр MEMADDR2.
- 6. CPU настраивает регистр РАСКЕТ.
- 7. При включённых возможностях ECC (автоматическом обнаружении и коррекции ошибок) CPU должен настроить содержимое регистров ECC и ECC_SPARE_CMD.
- 8. CPU инициирует выполнение команды установкой необходимого бита в регистре PROGRAM.
- 9. NFC выдаёт команду флэш-устройству и начинает считывать из него данные.

Пункты 10-21 — считывание данных из FIFO

- 10. СРU ожидает прерывание «Buffer Read Ready».
- 11. После того как в FIFO появляется хотя бы один доступный пакет данных в регистре INTERRUPT_STATUS устанавливается признак прерывания «Buffer Read Ready».
- 12. Формируется физический сигнал прерывания.
- 13. CPU считывает регистр INTERRUPT STATUS.
- 14. CPU запрещает формирование сигнала прерывания «Buffer Read Ready» (см. perистр INTERRUPT_SIGNAL_EN).
- 15. Если число прерываний равно числу переданных пакетов, выполняется шаг 16, иначе шаг 17.
- 16. CPU разрешает формирование признака прерывания «Transfer Complete» (см. регистр INTERRUPT STATUS EN). Далее шаг 18.
- 17. СРU сбрасывает бит «Buffer Read Ready» регистра INTERRUPT_STATUS_EN.
- 18. СРU сбрасывает бит «Buffer Read Ready» регистра INTERRUPT STATUS.
- 19. CPU считывает пакет данных через регистр BUFFER_DATA.
- 20. Если число прерываний равно числу переданных пакетов, выполняется шаг 23, иначе шаг 21.
- 21. CPU разрешает формирование признака прерывания «Buffer Read Ready» (см. регистр INTERRUPT STATUS EN).
- 22. CPU разрешает формирование сигнала прерывания «Buffer Read Ready» (см. регистр INTERRUPT SIGNAL EN). Далее шаг 10.

Пункты 23-30 — завершение отработки команды



- 23. CPU разрешает формирование сигнала прерывания «Transfer Complete» (см. регистр INTERRUPT SIGNAL EN).
- 24. CPU ожидает прерывание «Transfer Complete».
- 25. После завершения пересылки всех байтов данных для CPU NFC устанавливает признак прерывания «Transfer Complete» в регистре INTERRUPT_STATUS.
- 26. Формируется физический сигнал прерывания.
- 27. CPU считывает регистр INTERRUPT_STATUS.
- 28. CPU сбрасывает бит «Transfer Complete» регистра INTERRUPT SIGNAL EN.
- 29. CPU сбрасывает бит «Transfer Complete» регистра INTERRUPT STATUS EN.
- 30. CPU сбрасывает бит «Transfer Complete» в регистре INTERRUPT_STATUS.

Можно использовать альтернативный метод без использования сигнал прерывания. В этом случае разрешать формирование физического сигнала прерывания в регистре INTERRUPT_SIGNAL_EN не следует. Вместо этого CPU должен опрашивать регистр INTERRUPT_STATUS и проверять состояние разрядов «Transfer Complete», «Buffer Read Ready».

13.3.3 Схема выполнения команды «С записью в FIFO»

Для команды, требующей от CPU записи в FIFO обычно выполняется следующая последовательность действий:

Пункты 1-8 – инициирование команды

- 1. CPU разрешает формирование признака прерывания «Buffer Write Ready» (см. регистр INTERRUPT_STATUS_EN).
- 2. CPU разрешает формирование сигнала прерывания «Buffer Write Ready» (см. perucтp INTERRUPT_SIGNAL_EN).
- 3. СРИ настраивает регистр СОММАND.
- 4. СРИ настраивает регистр РАСКЕТ.
- 5. CPU настраивает регистр MEMADDR1.
- 6. CPU настраивает регистр MEMADDR2.
- 7. При включённых возможностях ECC (автоматическом обнаружении и коррекции ошибок) CPU должен настроить содержимое регистров ECC и ECC SPARE CMD.
- 8. CPU инициирует выполнение команды установкой необходимого бита в регистре PROGRAM.
- 9. NFC выдаёт команду флэш-устройству и начинает выполнять её.

Пункты 10-22 — запись данных в FIFO

- 10. СРU ожидает прерывание «Buffer Write Ready».
- 11. Если в FIFO достаточно места для приёма данных от CPU, в регистре INTERRUPT STATUS устанавливается признак прерывания «Buffer Write Ready».
- 12. Формируется физический сигнал прерывания.



- 13. CPU считывает регистр INTERRUPT STATUS.
- 14. CPU запрещает формирование сигнала прерывания «Buffer Write Ready» (см. регистр INTERRUPT_SIGNAL_EN).
- 15. Если число прерываний равно числу переданных пакетов, выполняется шаг 16, иначе шаг 17.
- 16. CPU разрешает формирование признака прерывания «Transfer Complete» (см. регистр INTERRUPT STATUS EN). Далее шаг 18.
- 17. СРU сбрасывает бит «Buffer Write Ready» регистра INTERRUPT STATUS EN.
- 18. СРU сбрасывает бит «Buffer Write Ready» регистра INTERRUPT STATUS.
- 19. CPU записывает пакет данных в FIFO через регистр BUFFER DATA.
- 20. Если число прерываний равно числу переданных пакетов, выполняется шаг 23, иначе шаг 21.
- 21. CPU разрешает формирование признака прерывания «Buffer Write Ready» (см. регистр INTERRUPT_STATUS_EN).
- 22. CPU разрешает формирование сигнала прерывания «Buffer Write Ready» (см. регистр INTERRUPT SIGNAL EN). Далее шаг 10.

Пункты 23-31 — завершение отработки команды

- 23. После того как все байты данных будут записаны во флэш-память NFC завершает выполнение команды.
- 24. CPU разрешает формирование сигнала прерывания «Transfer Complete» (см. регистр INTERRUPT_SIGNAL_EN).
- 25. CPU ожидает прерывание «Transfer Complete».
- 26. После завершения пересылки всех байтов данных NFC устанавливает признак прерывания «Transfer Complete» в регистре INTERRUPT_STATUS.
- 27. Формируется физический сигнал прерывания.
- 28. CPU считывает регистр INTERRUPT STATUS.
- 29. CPU сбрасывает бит «Transfer Complete» регистра INTERRUPT SIGNAL EN.
- 30. CPU сбрасывает бит «Transfer Complete» регистра INTERRUPT STATUS EN.
- 31. CPU сбрасывает бит «Transfer Complete» в регистре INTERRUPT STATUS.

Можно использовать альтернативный метод без использования сигнал прерывания. В этом случае разрешать формирование физического сигнала прерывания в регистре INTERRUPT_SIGNAL_EN не следует. Вместо этого CPU должен опрашивать регистр INTERRUPT_STATUS и проверять состояние разрядов «Transfer Complete», «Buffer Write Ready».



13.3.4 Команда Reset

Команда Reset (FFh) должна быть первой командой после включения питания флэшпамяти.

Команда Reset выполняется по схеме «Без доступа к FIFO», со следующими особенностями:

- настройку регистра MEMADDR1 можно не производить;
- в регистр COMMAND записывается значение 0xFF;
- в регистре PROGRAM необходимо установить бит reset.

13.3.5 Команда Read Status

Команда Read Status используется для получения статуса последней выполненной операции.

Данная команда выполняется по схеме «Без доступа к FIFO», со следующими особенностями:

- настройку регистра MEMADDR1 можно не производить;
- перед запуском команды через регистр PROGRAM необходимо дополнительно настроить регистр PACKET;
- в регистре PROGRAM необходимо установить бит read status.

После выполнения команды процессор может произвести чтение регистра FLASH_STATUS (данные о состоянии последней операции были загружены из флэшпамяти в этот регистр в результате выполнения команды).

13.3.6 Команда Read ID

С помощью команды Read ID определяется соответствие целевого устройства стандарту ONFI.

Команда Read ID с адресом 20h – соответствие стандарту ONFI.

Команда Read ID с адресом 00h – JEDEC ID.

Данная команда выполняется по схеме «Без доступа к FIFO», со следующими особенностями:

- перед запуском команды через регистр PROGRAM необходимо дополнительно настроить регистр PACKET;
- в регистре PROGRAM необходимо установить бит read id.



После выполнения команды процессор может произвести чтение регистров ID1 и ID2 (данные из флэш-памяти были загружены в эти регистры в результате выполнения команды). При выполнении команды Read ID с адресом 20h (ONFI ID) считанные из флэш-памяти четыре байта сохраняются в {ID2[7:0], ID1[31:8]}. При выполнении команды Read ID с адресом 00h (JEDEC ID) считанные из флэш-памяти пять байтов сохраняются в {ID2[7:0], ID1[31:0]}.

13.3.7 Команда Read Parameter Page

Команда Read Parameter Page получает информацию об организации, характеристиках, временных параметрах и другие особенностях целевого устройства.

Данная команда выполняется по схеме «С чтением из FIFO», со следующими особенностями:

• в регистре PROGRAM необходимо установить бит read_parameter_page.

13.3.8 Команда Page Program без ECC

Данная команда осуществляет передачу страницы данных или части страницы данных, определяемой адресом столбца, в регистр страницы. Содержимое регистра страницы затем пересылается в массив флэш-памяти по заданному адресу строки.

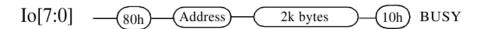


Рисунок 13.1. Команда Page Program без ECC

Данная команда выполняется по схеме «Команды с записью в FIFO», со следующими особенностями:

• в регистре PROGRAM установлен бит раде program.

При последовательном способе работы с флэш-памятью общее число пересылок (packet_size x packet_count) не должно превышать размер страницы флэш-памяти. Для параллельной работы (одновременно с двумя флэш-памятями) максимальный размер пересылок может быть в два раза больше размера страницы, так как поле packet_count для параллельного режима должно быть в два раза больше чем для последовательного.

Пример.

При последовательном подключении флэш-памяти с размером страницы 8k можно выбрать следующие значения полей регистра PACKET:

packet size = 512, packet count = 16.



При параллельном подключении флэш-памяти с размером страницы 8k можно выбрать следующие значения полей регистра PACKET:

packet_size = 512, packet_count = 32.

13.3.9 Команда Page Program с ECC

Данная команда осуществляет передачу страницы или части страницы данных, определяемой адресом столбца, в регистр страницы. Содержимое регистра страницы затем пересылается в массив флэш-памяти по заданному адресу строки.



Рисунок 13.2. Команда Page Program с ECC

Данная команда выполняется по схеме «С записью в FIFO», со следующими особенностями:

- в регистре PROGRAM установлен бит раде program;
- в регистре COMMAND установлен бит ECC_ON_OFF;
- в регистре ЕСС:
 - для SLC поле $slc_mlc = 0$, $ecc_addr = 0x834$, $ecc_size = 0x0c$;
 - для MLC поле $slc_mlc = 1$, $ecc_addr = 0x826$, $ecc_size = 0x1a$;
- в регистре ECC SPARE CMD:

Поле ecc spare cmd = 0x0085 (Change Write Column).

Поле $ecc_spare_num_addr_cycles = 0x02$.

Байты данных, записанные в FIFO для пересылки во флэш-память, подаются на вход модуля ECC для вычисления контрольных разрядов ECC. Контрольные разряды вычисляются для каждой последовательности из 512 байт.

После того как все пакеты данных переданы во флэш-память, NFC выполняет команду 0x85 (Change Write Column) с двумя адресными циклами. После чего NFC записывает вычисленные контрольные разряды в резервную область флэш-памяти.

Примечание. Значения поля есс_addr 0x834 или 0x826 используется для страницы размером 2k. Данные значения выбраны таким образом, чтобы записать контрольные разряды ECC в последние свободные резервные (spare) байты данной страницы.

Адрес контрольных разрядов для страницы размером 2k в режиме SLC = общее количество байтов в странице – длина кода <math>ECC = 0x840 - 0x0C = 0x834.

Адрес контрольных разрядов для страницы размером 2k в режиме MLC = общее количество байтов в странице – длина кода <math>ECC = 0x840 - 0x0A = 0x826.



Примечание. Подробнее о значениях полей регистра ЕСС см. в пункте «Регистр ЕСС».

Примечания

Для режима SLC, страницы размером 512 при синхронном подключении, разряд есс_on_off в регистре COMMAND должно быть нулем.

Для режима MLC-4, страницы размером 512 при синхронном подключении, разряд есс on off в регистре COMMAND должно быть нулем.

13.3.10 Команда Page Program для spare области

Программирование резервной области выполняется аналогично обычной команде программирования. Адрес области для программирования должен указывать на резервную область.

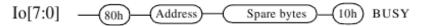


Рисунок 13.3. Команда Page Program (spare)

Таблица 13.27. Рекомендуемые параметры программирования резервной области

Параметр\Размер страницы	2k	4k	8k
Количество байт в spare пересылке	<= 64	<= 128	<= 256
MEMADDR1	> 0x7FF	> 0xFFF	> 0x1FFF

13.3.11 Команда Read без ECC

Команда считывает страницу данных, заданную адресом строки для заданного LUN. Страница данных становится доступной для чтения из регистра страницы, начиная с заданного адреса столбца.

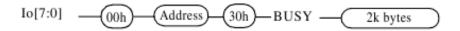


Рисунок 13.4. Команда Чтение без ЕСС

Данная команда выполняется по схеме «С чтением из FIFO», со следующими особенностями:

• в регистре PROGRAM необходимо установить бит read.

При последовательном способе работы с флэш-памятью общее число пересылок (packet_size x packet_count) не должно превышать размер страницы флэш-памяти. Для параллельной работы (одновременно с двумя флэш-памятями) максимальный размер



пересылок может быть в два раза больше размера страницы, так как поле packet_count для параллельного режима должно быть в два раза больше чем для последовательного.

Таблица 13.28. Пример настроек регистра РАСКЕТ для команды Read

Параметр/Размер Страницы	512	2k	4k	8k
packet_size	512	512	512	512
packet_count (последов.)	1	4	8	16
packet_count (парал.)	2	8	16	32

13.3.12 Команда Read с ECC

Команда считывает страницу данных, заданную адресом строки для заданного LUN. Страница данных становится доступной для чтения из регистра страницы, начиная с заданного адреса столбца.



Рисунок 13.5. Команда Чтение с ЕСС

Данная команда выполняется по схеме «С чтением из FIFO», со следующими особенностями:

- в пунктах 1 и 2 дополнительно разрешить прерывания «Single Bir Error», «Multi Bir Error»;
- в регистре COMMAND (пункт 3) необходимо разрешить возможности ЕСС (разряд есс on off);
- необходимо настроить ЕСС регистры (пункт 7):

Регистр ЕСС (0х34):

Для SLC поле $slc_mlc = 0$, $ecc_addr = 0x834$, $ecc_size = 0x0c$.

Для MLC поле $slc_mlc = 1$, $ecc_addr = 0x826$, $ecc_size = 0x1a$.

Регистр ECC SPARE CMD (0x3c):

Поле $ecc_spare_cmd = 0xe005$ (Change Read Column).

Поле num of addr cycles = 0x02.

• в регистре PROGRAM (пункт 8) установить разряд read.

Сначала NFC выполнит команду Read с адресом заданным в регистре ECC и произведёт считывание контрольных байтов. После этого NFC выполнит команду Change Read Column с двумя циклами адреса и приступит к считыванию данных из основной области.



Данные, считанные из основной области флэш-памяти в FIFO подаются на вход блока ECC для вычисления контрольных разрядов ECC. Контрольные разряды ECC вычисляются для каждого блока из 512 байт. Блок ECC осуществляет проверку вычисленного кода ECC со считанным из флэш-памяти.

Haстройка полей ecc_size, ecc_addr регистра ECC выполняется аналогично команде Page Program (c ECC).

Возможны два сценария ошибок SLC:

- корректируемые ошибки;
- некорректируемые ошибки.

Корректируемые ошибки.

При выполнении операции чтения (с использованием ECC) NFC сравнивает контрольные байты ECC из флэш-памяти с вычисленными. При обнаружении ошибки в одном бите NFC автоматически исправляет ошибку, и устанавливает прерывание «Single Bit Error». После приёма пакета устанавливается прерывание «Buffer Read Ready». По окончанию всей пересылки устанавливается прерывание «Transfer Complete».

Рассмотрим, следующий пример (размер страницы – 2k, размер пакета – 512, количество пакетов – 4). При считывании первого пакета NFC обнаруживает однобитную ошибку. Затем NFC исправляет её и устанавливает прерывание «Single Bit Error». После этого контроллер считывает второй, третий и четвертый пакеты из флэш-памяти. Прерывание «Вuffer Read Ready» устанавливается после приёма каждого пакета. После чтения всех данных из флэш-памяти NFC устанавливает прерывание «Transfer Complete».

Некорректируемые ошибки.

При выполнении операции чтения (с использованием ECC) NFC сравнивает контрольные байты ECC из флэш-памяти с вычисленными. Если контроллер обнаруживает ошибки в двух битах, устанавливается прерывание «Multi Bit Error». В этом случае прерывания «Buffer Read Ready» и «Transfer Complete» не устанавливаются.

Например, размер страницы – 2k, размер пакета – 512, количество пакетов – 4. При чтении первого пакета данных, если контроллер обнаруживает ошибку в двух битах, устанавливается прерывание «Multi Bit Error». NFC переключается в состояние IDLE.

13.3.13 Команда Read для spare области

Чтение из резервной области выполняется аналогично обычной команде чтения. Адрес считываемой области должен указывать на резервную область.



Рисунок 13.6. Чтение из резервной области

Ограничения на MEMADDR1 и размер пересылок аналогично команде Page Program для spare области.

13.3.14 Команда Block Erase

Команда Block Erase стирает блок данных, определяемый адресом блока в заданном логическом устройстве (LUN).



Рисунок 13.7. Команда Block Erase

Данная команда выполняется по схеме «Без доступа к FIFO», со следующими особенностями:

- COMMAND[15:0] = 0xD060;
- в регистре PROGRAM необходимо установить бит block_erase.

13.3.15 Команда Multi-plane Page Program

Команда Multi-plane Page Program (80h-11h) позволяет производить одновременную пересылку нескольких регистров страниц в различные блоки массива флэш-памяти, что позволяет увеличить производительность системы. Для того чтобы выполнить операцию программирования одновременно над несколькими блоками, необходимо перед командой Page Program (80h-10h) добавить одну или несколько команд Multi-plane Page Program.

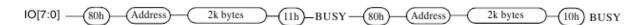


Рисунок 13.8. Multi-plane Page Program

Пример использования Multi-plane Page Program:

• команда Multi-plane Page Program выполняется по схеме «С записью в FIFO» со следующими особенностями:

COMMAND[15:0] = 0x1180.

В регистре PROGRAM необходимо установить бит page program.

• команда Page Program выполняется по схеме «С записью в FIFO», со следующими особенностями:

COMMAND[15:0] = 0x1080.



В регистре PROGRAM необходимо установить бит page program.

13.3.16 Команды Multi-plane Copyback

Обычная операция копирования памяти состоит из двух шагов:

• выполняется команда Copyback Read.

Команда загружает заданную страницу в регистр страницы.

• выполняется команда Copyback Program

Команда программирует заданную страницу содержимым регистра страницы.

Для увеличения производительности можно использовать команды Multi-plane Read и Multi-plane Copyback Program.

Чтобы выполнить одновременное копирование нескольких страниц в различные регистры страниц, необходимо перед командой Copyback Read (00h-35h) добавить одну или несколько команд Multi-plane Read (00h-32h).

Чтобы выполнить одновременное программирование нескольких страниц из различных регистров страниц необходимо перед командой Copyback Program (85h-10h) добавить одну или несколько команд Multi-plane Copyback Program (85h-11h).

Следующая последовательность команд позволяет произвести одновременное копирование двух блоков памяти:



Рисунок 13.9. Одновременное копирование двух блоков в одном LUN

Пример одновременного копирование двух блоков в одном LUN:

1. Команда Read Multi-plane выполняется по схеме «Без доступа к FIFO», со следующими особенностями:

В регистре PROGRAM необходимо установить бит copy_back_interleaved.

COMMAND[15:0] = 0x3200.

Команда начинает загрузку заданной страницы в регистр страницы и разрешает выполнение следующей команды, не дожидаясь завершения текущей.

2. Выполнятся команда Copyback Read по схеме «Без доступа к FIFO», со следующими особенностями:

В регистре PROGRAM необходимо установить бит сору back interleaved.



COMMAND[15:0] = 0x3500.

Команда загружает следующую необходимую страницу в другой регистр страницы (определяется адресом блока). Таким образом выполняется параллельное копирование двух разных областей памяти в соответствующие регистры страниц.

3. Выполнятся команда Copyback Program Multi-plane по схеме «Без доступа к FIFO», со следующими особенностями:

В регистре PROGRAM необходимо установить бит сору back interleaved.

COMMAND[15:0] = 0x1185.

Команда начинает сохранение регистра страницы в необходимую область памяти и разрешает выполнение следующей команды, не дожидаясь завершения текущей.

4. Выполнятся команда Copyback Program по схеме «Команды без доступа к FIFO», со следующими особенностями:

В регистре PROGRAM установлен бит сору back interleaved.

COMMAND[15:0] = 0x1085.

Команда выполняет сохранение другого регистра страницы в необходимую область памяти. Таким образом, выполняется одновременное программирование двух различных областей памяти

Замечание.

При выполнение multi-plane операций необходимо учитывать ограничения на адреса областей памяти (см. стандарт ONFI).

13.3.17 Команда Multi-plane Block Erase

Команда Multi-plane Block Erase (60h-D1h) позволяет производить одновременное стирание более чем одного блока в массиве флэш-памяти, что позволяет увеличить производительность системы. Для того чтобы выполнить команду Erase одновременно над несколькими блоками, необходимо перед командой Erase Block (60h-D0h) добавить одну или несколько команд Multi-plane Block Erase.



Рисунок 13.10. Одновременное стирание двух блоков памяти в одном LUN

Пример одновременного стирания двух блоков памяти в одном LUN:



1. Выполняется команда Block Erase Multi-plane по схеме «Команды без доступа к FIFO», со следующими особенностями:

В регистре PROGRAM установлен разряд block erase.

COMMAND[15:0] = 0xD160.

2. Выполнятся команда Block Erase (см. Описание команды Block Erase)

13.3.18 Команда Multi-plane Read

Команда Read Multi-plane (00h-32h) позволяет производить одновременную загрузку нескольких страниц в различные регистры страниц, что позволяет увеличить производительность системы. Для того чтобы выполнить команду чтения над несколькими блоками одновременно, необходимо перед командой Read (00h-30h) добавить одну или несколько команд Read Multi-plane. Выбор блока для считывания данных из флэш-памяти после её готовности осуществляется командой Change Read Column Enchanced (06h-E0h).

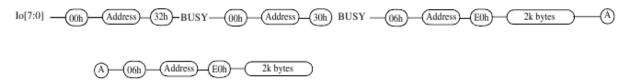


Рисунок 13.11. Ускоренное чтение двух блоков памяти из одного LUN

Ускоренное чтение двух блоков памяти из одного LUN:

1. Выполнятся команда Read Multi-plane по схеме «Без доступа к FIFO» со следующими особенностями:

В регистре PROGRAM установлен бит read_interleaved.

COMMAND[15:0] = 0x3200.

2. Выполнятся команда Read по схеме «Без доступа к FIFO» со следующими особенностями:

В регистре PROGRAM установлен бит read interleaved.

COMMAND[15:0] = 0x3000.

3. Выполнятся команда Change Read Column Enchanced по схеме «С чтением из FIFO» со следующими особенностями:

В PROGRAM установлен бит read column enhanced.

COMMAND[15:0] = 0xE006.



4. Выполнятся команда Change Read Column Enchanced по схеме «С чтением из FIFO» со следующими особенностями:

В PROGRAM установлен бит read column enhanced.

COMMAND[15:0] = 0xE006.

13.3.19 Команда Multi-plane Read с ECC

Команда Read Multi-plane (00h-32h) позволяет производить одновременную загрузку нескольких страниц в различные регистры страниц, что позволяет увеличить производительность системы. Для того чтобы выполнить команду чтения над несколькими блоками одновременно, необходимо перед командой Read (00h-30h) добавить одну или несколько команд Read Multi-plane. Выбор блока для считывания данных из флэш-памяти после её готовности осуществляется командой Change Read Column Enchanced (06h-E0h).

Если требуется обновить адрес стобца (адрес байта внутри страницы) без смены LUN, блока или Plane, вместо команды Change Read Column Enhanced может быть использована команда Changre Read Column (05h-E0h).

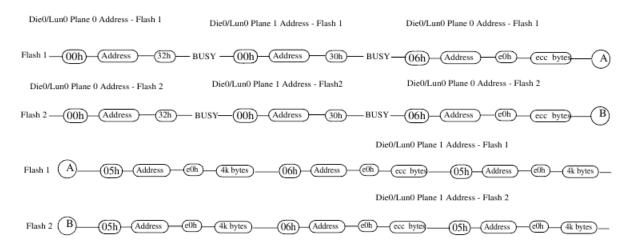


Рисунок 13.12. Одновременное чтение двух блоков памяти из разных флэш-памятей, с ECC для страницы 2k

Параллельное считывание областей из двух разных флэш-памятей (с параллельным подключением) с кодом коррекции ECC:

• выполнятся команда Multi-plane Read по схеме «Без доступа к FIFO» со следующими особенностями:

В регистре PROGRAM установлен бит read interleaved.

COMMAND[15:0] = 0x3200.



• выполнятся команда Read по схеме «Без доступа к FIFO» со следующими особенностями:

В регистре PROGRAM установлен бит read_interleaved.

COMMAND[15:0] = 0x3000.

• выполнятся команда Change Read Column Enchanced по схеме «С чтением из FIFO» со следующими особенностями:

COMMAND[15:0] = 0xE006, COMMAND[ecc on off] = 1

В регистре PROGRAM необходимо установить бит change read column enhanced.

Регистр ECC = 0x0268080C (ecc addr=16'h080C, ecc size=9'h68, slc mlc = 1).

Значение perистра ECC_SPARE_CMD равно 0x2000e005 (ecc_spare_cmd=16'he005, number_of_addr cycles=3'h2).

В результате запуска команды NFC выполнит команду Change Read Column Enhanced и считает байты ECC, а затем выполнит команду Change Read Column и считает байты основных данных.

• выполнятся команда Change Read Column Enchanced по схеме «С чтением из FIFO» со следующими особенностями:

 $COMMAND[15:0] = 0xE006, COMMAND[ECC_ON_OFF] = 1.$

В регистре PROGRAM необходимо установить бит change read column enhanced.

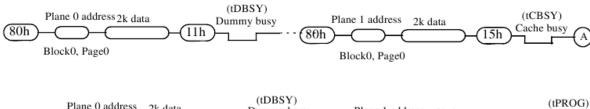
Регистр ECC = 0x0268080C (ecc addr=16'h080C, ecc size=9'h68, slc mlc = 1).

Значение регистра ECC_SPARE_CMD равно 0x2000e005 (ecc_spare_cmd=16'he005, number of addr cycles=3'h2).

В результате запуска команды NFC выполнит команду Change Read Column Enhanced и считает байты ECC, а затем выполнит команду Change Read Column и считает байты основных данных.



13.3.20 Команда Page Cache Program (2 interleaved адреса)



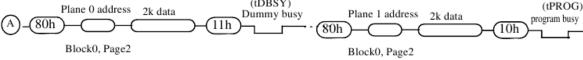


Рисунок 13.13. Page Cache Program (2 interleaved адреса)

Выполнение последовательности:

- 1. Выполнятся команда Multi-plane Page Program.
- 2. Выполнятся команда Page Cache Program по схеме «С записью в FIFO» со следующими особенностями:

В регистре PROGRAM установлен бит page_program.

- 1. Выполнятся команда Multi-plane Page Program.
- 2. Выполнятся команда Page Program.

13.3.21 Команда Page Cache Program

Команда Page Cache Program (80h-15h) используется для увеличения производительности операций программирования.



Рисунок 13.14. Page Cache Program

Пример использования:

1. Команда Page Cache Program выполняется по схеме «С записью в FIFO» со следующими особенностями:

COMMAND[15:0] = 0x1580.

В регистре PROGRAM необходимо установить бит page program.

2. Команда Page Program выполняется по схеме «С записью в FIFO», со следующими особенностями:

COMMAND[15:0] = 0x1080.

В регистре PROGRAM необходимо установить бит page program.



13.3.22 Команда Read Cashe Sequential

Использование серии команд Read Cache Sequential (31h) позволяет увеличить скорость передачи данных. Этого достигается за счёт того, что в момент выдачи данных из регистра кэша, новая (следующая, последовательная) страница данных загружается из массива флэш-памяти в регистр данных.



Рисунок 13.15. Read Cache Sequential

Пример использования:

1. Выполняется команда Read по схеме «Без доступа к FIFO» со следующими особенностями:

COMMAND[15:0] = 0x3000.

В регистре PROGRAM необходимо установить бит read cache start.

2. Выполняется команда Read Cashe Sequential по схеме «С чтением из FIFO», со следующими особенностями:

COMMAND[7:0] = 0x31.

В PROGRAM необходимо установить бит read cache sequential.

3. Выполняется команда Read Cashe End по схеме «С чтением из FIFO», со следующими особенностями:

COMMAND[7:0] = 0x3F.

В PROGRAM необходимо установить бит read cache end.

13.3.23 Команда Read Cashe Random

Команда Read Cache Random (00h-31h) начинает копирование заданной страницы из массива флэш-памяти в соответствующий регистр данных.

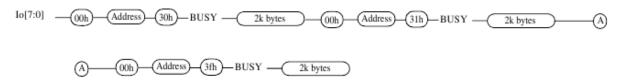


Рисунок 13.16. Read Cache Random

Пример использования:



1. Выполняется команда Read по схеме «Без доступа к FIFO» со следующими особенностями:

COMMAND[15:0] = 0x3000.

В регистре PROGRAM необходимо установить бит read cache start.

2. Выполняется команда Read Cashe Random по схеме «С чтением из FIFO», со следующими особенностями:

COMMAND[15:0] = 0x3100.

В PROGRAM необходимо установить бит read cache random.

3. Выполняется команда Read Cashe End по схеме «С чтением из FIFO», со следующими особенностями:

COMMAND[7:0] = 0x3F.

В PROGRAM необходимо установить бит read cache end.

13.3.24 Выбор LUN/die

LUN (или die) - наименьший элемент, который может независимо выполнять команды и сообщать о своём состоянии. Для увеличения производительности при работе с устройствами, которые состоят из нескольких LUN, можно параллельно выполнять операции с разными LUN. NFC поддерживает следующие операции над несколькими LUN:

- Page Program LUN0, Page Program LUN1;
- Read LUN0, Read LUN1;
- Erase LUN0, Erase LUN1;
- Page Program LUN0, Read LUN1.

Для выбора необходимого из нескольких LUN используется команда Read Status Enhanced. После выполнения этой команды только LUN, который был выбран будет откликаться на циклы чтения данных.

Последовательность работы с несколькими LUN:

- выбор LUN0;
- запуска команды для LUN0 (Program, Erase, Read);
- установка занятости LUN0;
- выбор LUN1 (в этот момент LUN0 занят пересылкой);
- запуска команды для LUN1 (Program, Erase, Read);
- ожидание завершения операций.



13.3.25 Команда Read Status Enhanced

Данная команда выполняется по схеме «Без доступа к FIFO», со следующими особенностями:

- перед выполнением запуска команды через регистр PROGRAM необходимо дополнительно настроить регистр PACKET (packet_size = 1, packet_count = 1);
- в регистре PROGRAM необходимо установить бит read status enhanced.

После выполнения команды процессор может произвести чтение регистр FLASH_STATUS (данные о состоянии последней операции были загружены из флэшпамяти в этот регистр в результате выполнения команды).

13.3.26 Page Program LUN0, Page Program LUN1

Последовательность программирования LUN0, LUN1:

- командой Read Status Enhanced выбирается LUN0;
- выполняется команда Page Program по схеме «С записью в FIFO» со следующими особенностями:

В регистре PROGRAM необходимо установить биты page program и multi die.

- командой Read Status Enhanced выбирается LUN1;
- выполняется команда Page Program по схеме «С записью в FIFO» со следующими особенностями:

В регистре PROGRAM необходимо установить бит page program.

Если необходимо произвести программирование нескольких LUN, бит multi_die совместно с page_program в регистре PROGRAM должен быть установлен для всех команд чтения, кроме последней.

Пример программирования LUN0, LUN1, LUN2:

Для LUN0 и LUN1 бит multi_die необходимо устанавливать совместно с page_program, а для LUN2 только page_program.

13.3.27 Read LUN0, Read LUN1

Последовательность чтения из LUN0 и LUN1:

- командой Read Status Enhanced выбирается LUN0;
- выполняется команда Read по схеме «Без доступа к FIFO» со следующими особенностями:

COMMAND[15:0] = 0x3000.



В регистре PROGRAM необходимо установить биты read и multi die.

- командой Read Status Enhanced выбирается LUN1;
- выполняется команда Read по схеме «Без доступа к FIFO» со следующими особенностями:

COMMAND[15:0] = 0x3000.

В регистре PROGRAM необходимо установить биты read и multi_die.

• командой Read Status Enhanced выбирается LUN0.

Данный шаг повторяется до тех пор пока в 6-ом (RDY) разряде регистра FLASH_STATUS не будет считана единица (признак того, что LUN готов принимать следующие команды и остальные разряды FLASH_STATUS достоверны).

• выполняется команда по схеме «С чтением из FIFO» со следующими особенностями:

COMMAND[7:0] = 0x00.

В PROGRAM необходимо установить бит 5 (multi_die_rd).

• командой Read Status Enhanced выбирается LUN1.

Данный шаг повторяется до тех пор пока в 6-ом (RDY) разряде регистра FLASH_STATUS не будет считана единица (признак того, что LUN готов принимать следующие команды и остальные разряды FLASH STATUS достоверны).

• выполняется команда по схеме «С чтением из FIFO» со следующими особенностями:

COMMAND[7:0] = 0x00.

В PROGRAM необходимо установить бит 5 (multi die rd).

13.3.28 Block Erase LUN0, Block Erase LUN1

Последовательность стирания в LUN0 и LUN1:

- командой Read Status Enhanced выбирается LUN0;
- выполняется команда по схеме «Без доступа к FIFO» со следующими особенностями:

В регистре PROGRAM необходимо установить бит 2 (block erase) и бит 1 (multi die).

- командой Read Status Enhanced выбирается LUN1;
- выполняется команда по схеме «Без доступа к FIFO» со следующими особенностями:

В регистре PROGRAM необходимо установить бит 2 (block_erase).



Если требуется выполнить команду Erase для нескольких блоков, бит multi_die совместно с block_erase в регистре PROGRAM должен быть установлен для всех команд, кроме последней.

13.3.29 Page Program LUN0, Read LUN1

Последовательность Page Program в LUN0 и Read в LUN1:

- командой Read Status Enhanced выбирается LUN0;
- выполняется команда по схеме «С записью в FIFO» со следующими особенностями:

В регистре PROGRAM необходимо установить бит 4 (page_program) и бит 1 (multi_die).

- командой Read Status Enhanced выбирается LUN1;
- выполняется команда по схеме «С чтением из FIFO» со следующими особенностями:

В регистре PROGRAM необходимо установить бит 0 (read).

13.3.30 Small Data Move

Если флэш-память поддерживает команду Small Data Move, процессор может записывать данные во флэш-память с приращением меньшим, чем размер страницы (применимо для команд Program и Copyback). Наименьший размер данных, которые можно записать во флэш-память, используя Small Data Move — двойное слово. При выполнении команд Small Data Move поддержка ЕСС должна быть отключена. Процессор может определить наличие поддержки Small Data Move во флэш-памяти, произведя чтение страницы параметров.

13.3.31 Small Data Move, Page Program

На следующем рисунке показан пример программирования 4-х областей флэш-памяти с размером страницы 2k, использующий свойство Small Data Move:



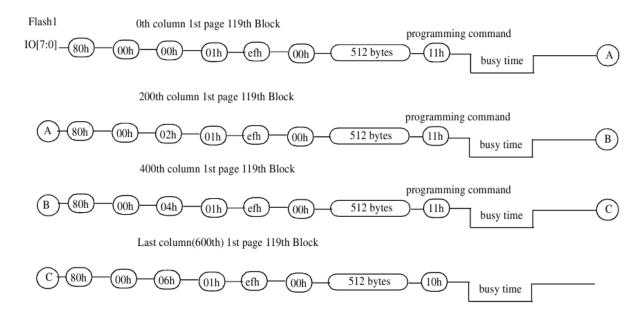


Рисунок 13.17. Пример записи во флэш-память – Small Data Move

При обычном программировании страницы размером 2k достаточно было выполнить одну команду программирования, например, настроив 4 пакета по 512 байт. Small Data Move (с размером пакета 512 байт и количеством пакетов 1) требует выполнения нескольких команд:

- Multi-plane Page Program по схеме «С записью в FIFO»: COMMAND[15:0] = 0x1180, MEMADDR1 = 000, в PROGRAM установлен small_data_move;
- Multi-plane Page Program по схеме «С записью в FIFO»: COMMAND[15:0] = 0x1180, MEMADDR1 = 200, в PROGRAM установлен small_data_move.
- Multi-plane Page Program по схеме «С записью в FIFO»: COMMAND[15:0] = 0x1180, MEMADDR1 = 400, в PROGRAM установлен small data move.
- Multi-plane Page Program по схеме «С записью в FIFO»: COMMAND[15:0] = 0x1080, MEMADDR1 = 600, в PROGRAM установлен small_data_move.



13.3.32 Small Data Move, Copyback Program

Выполняется аналогично Page Program, Small Data Move. Единственное отличие в настройке кода операции в регистре PROGRAM: 85h вместо 80h.

13.3.33 Small Data Move, Change Read Column Enhanced

После запуска команд Small Data Move процессор может использовать команду Change Read Column Enhanced для чтения данных, записанных до этого:

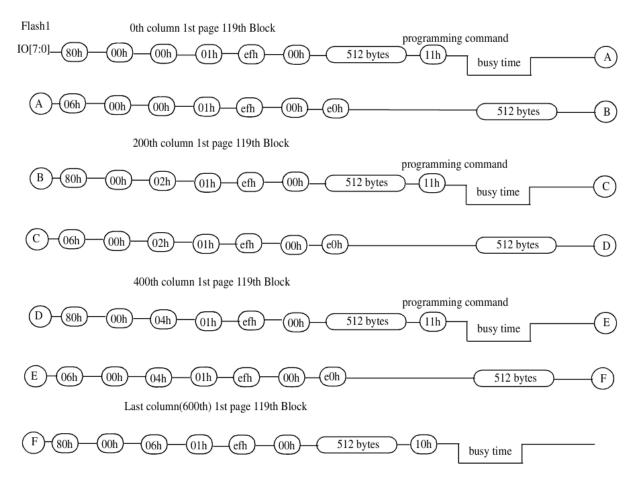


Рисунок 13.18. Small Data Move, Change Read Column Enhanced

Команда Change Read Column Enhanced выполняется по схеме «С чтением из FIFO»: COMMAND[15:0] = 0xE006, в PROGRAM установлен change_read_column_enhanced.

13.3.34 Команда Change Row Address

Команда Change Row Address изменяет адрес строки и столбца LUN, куда производится запись. Можно использовать совместно с командой программирования флэш-памяти (адрес LUN и interleaved адреса должны быть те же, что и в команде программирования):



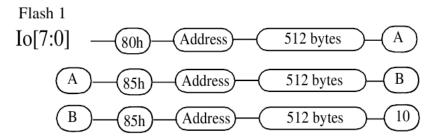


Рисунок 13.19. Change Row Address

В данном примере процессору необходимо запрограммировать следующие области флэшпамяти:

- LUN0, Plane0, Блок 1, Страница 0 (512 байт);
- LUN0, Plane0, Блок 2, Страница 0 (512 байт);
- LUN0, Plane0, Блок 3, Страница 0 (512 байт).

Для этого выполняются команды:

1. Page Program по схеме «С записью в FIFO»: COMMAND[7:0] = 0x80.

B perистре PACKET поле packet_count = 1 и packet_size = 512.

В PROGRAM установлены change row addr и page program.

2. Change Row Address по схеме «С записью в FIFO»: COMMAND[7:0] = 0x85.

В регистре PACKET поле packet count = 1 и packet size = 512.

B PROGRAM установлен change row addr.

3. Change Row Address по схеме «С записью в FIFO»: COMMAND[15:0] = 0x1085.

B регистре PACKET поле packet count = 1 и packet size = 512.

B PROGRAM установлен change_row_addr_end.

Шаг 2 может повторяться несколько раз, если необходимо запрограммировать несколько страниц.



13.3.35 Команда Reset LUN

Команда Reset LUN используется для сброса логического устройства (LUN).

Команда выполняется по схеме «Команды без доступа к FIFO», со следующими особенностями:

• в регистре PROGRAM необходимо установить разряд reset lun.

13.3.36 Свойство Page Register Clear Enhancement

В стандарте ONFI 2.1 (пункт 3.1.3. Multiple LUN Operation Restrictions) указано, что не следует начинать программирование LUN, если выполняется операция чтения другого LUN. Связано с тем, что в ONFI 2.1 при поступлении команды программирования LUN допускается сброс регистров страниц других LUN. И если начать операцию программирования, не завершив операцию чтения, данные чтения, загруженные в регистр страницы, могут быть утеряны.

В стандарте ONFI 2.2 (пункт 3.1.3. Multiple LUN Operation Restrictions) ограничение на запись во время чтения ослаблено. В случае, если флэш-память поддерживает свойство «Page Register Clear Enhancement» (определяется в Странице Параметров) и командой Set Features установлен параметр PC в свойстве «Timing Mode» (см. ONFI 2.2), запись во время чтения допускается.

Для программирования флэш-памяти (команда 80h) во время чтения (для флэш-памяти с установленным признаком РС в свойстве «Timing Mode») необходимо использовать особое время tADL (время установки данных для программирования после выдачи команды программирования). Время tADL настраивается в регистре TIMING на основании значений байтов 154-155 Страницы Параметров. Для того, чтобы контроллер использовал это время, необходимо установить разряд pgm_pg_reg_clr в регистре PROGRAM.

13.3.37 Команда Set Features

Команда Set Features позволяет изменить значение необходимого параметра флэш-памяти. Команда используется для включения настроек, которые отключены по сбросу питания (смена временного режима, тип интерфейса данных).

Команда выполняется по схеме «С записью в FIFO» со следующими особенностями:

• в регистре PROGRAM необходимо установить бит set feature.



13.3.38 Команда Get Features

Команда Get Features позволяет определить текущее значение настроек флэш-памяти.

Команда выполняется по схеме «С чтением из FIFO» со следующими особенностями:

• в регистре PROGRAM необходимо установить бит get feature.

13.3.39 Переключение временных режимов

Работа с флэш-памятью может осуществляться асинхронным или синхронным способом (см. ONFI).

Для асинхронного способа определено 6 временных режимов. Считается, что по включению питания флэш-память работает в асинхронном режиме 0. Асинхронный режим 0 должен поддерживаться во всех реализациях флэш-памяти. Поддержка других асинхронных временных режимов определяется процессором на оснований значений байтов 129-130 Страницы Параметров.

Для синхронного способа определено 6 временных режимов. Поддержка синхронного способа работы с флэш-памятью определяется процессором на оснований значений байта 6 Страницы Параметров. Байты 141-142 Страницы Параметров определяют какие из синхронных режимов поддерживаются. Синхронный интерфейс использует DDR протокол, всегда передаётся чётное число байт.

Переключение временных режим осуществляется с помощью команд Set Feature и Get Feature. Необходимо корректно настроить регистр PACKET (см. пункт «Регистр PACKET»).

При смене временного режима значение регистра MEMADDR1 должно быть равно 1.

При смене временных режимов рекомендуется использовать следующую последовательность:

- 1. Команда Get Features (не обязательно).
- 2. Команда Set Features.
- 3. Команда Get Features.



13.3.39.1 Переход из асинхронного режима в асинхронный

В таблице указаны значения параметров P1-P4 (см. ONFI), которые необходимо задать команде Set Features (через FIFO при возникновении прерывания «Buffer Write Ready»).

Таблица 13.29. Значения параметров P1-P4 для команды Set Features (переход из асинхронного в асинхронный режим)

Режим	Параметры Р1-Р4 для последовательного соединения	Параметры Р1-Р4 для параллельного соединения
mode0	32'h0000_0000	64'h0000_0000_0000_0000
mode1	32'h0000_0001	64'h0000_0000_0000_0101
mode2	32'h0000_0002	64'h0000_0000_0000_0202
mode3	32'h0000_0003	64'h0000_0000_0000_0303
mode4	32'h0000_0004	64'h0000_0000_0000_0404
mode5	32'h0000_0005	64'h0000_0000_0000_0505

13.3.39.2 Переход из синхронного режима в синхронный

Таблица 13.30. Значения параметров P1-P4 для команды Set Features (переход из синхронного в синхронный режим)

Режим	Параметры Р1-Р4 для последовательного соединения	Параметры Р1-Р4 для параллельного соединения
mode0	64'h0000_0000_0000_1010	128'h0000_0000_1010_1010
mode1	64'h0000_0000_0000_1111	128'h0000_0000_1111_1111
mode2	64'h0000_0000_0000_1212	128'h0000_0000_1212_1212
mode3	64'h0000_0000_0000_1313	128'h0000_0000_1313_1313
mode4	64'h0000_0000_0000_1414	128'h0000_0000_1414_1414
mode5	64'h0000_0000_0000_1515	128'h0000_0000_1515_1515

После успешного завершения команды Set Features процессору следует настроить новую частоту NFC SCLK.

13.3.39.3 Переход из асинхронного режима в синхронный

Последовательность действий:

- команда Get Features (не обязательно);
- команда Set Features (после успешного завершения процессор должен настроить новую частоту NFC SCLK).

Отключить NFC SCLK.

Настроить внешний блок PLL (в зависимости от режима).



Настроить необходимую частоту.

Дождаться установки стабильной частоты, записать выбранный режим в регистр PROGRAM

команла Get Features.

Таблица 13.31. Значения параметров P1-P4 для команды Set Features (переход из асинхронного в синхронный режим)

Режим	Параметры Р1-Р4
mode0	32'h0000_0010
mode1	32'h0000_0011
mode2	32'h0000_0012
mode3	32'h0000_0013
mode4	32'h0000_0014
mode5	32'h0000_0015

Таблица 13.32. Частоты синхронного режима

Синхронный режим	Частота (МГц)	Период (нс)
mode0	20	50
mode1	33	30
mode2	50	20
mode3	66	15
mode4	83	12
mode5	100	10

13.3.39.4 Переход из синхронного режима в асинхронный

Процессору необходимо выполнить следующие действия:

- 1. Выполнить команду Get Features (не обязательно).
- 2. Установить значение частоты NFC SCLK 100 Мгц.

Убелиться в стабильности частоты.

3. Выполнить команду Reset в асинхронном режиме (mode0). Флэш-память в любом режиме должна распознать команду сброса, выполненную в асинхронном режиме 0.

- 4. Выполнить команду Get Fetures.
- 5. Выполнить команду Set Features и выбрать необходимый асинхронный режим.



13.3.40 Page Program (режим MDMA)

Команда Page Program пересылает страницу данных (или часть страницы) в регистр страницы. Затем содержимое регистра страницы записывается в массив флэш-памяти. В режиме MDMA данные страницы напрямую загружаются из памяти, используя DMA контроллер.

Схема выполнения команды Page Program в режиме MDMA:

- 1. CPU разрешает формирование признака прерывания «Transfer Complete» (см. регистр INTERRUPT STATUS EN).
- 2. CPU разрешает формирование сигнала прерывания «Transfer Complete» (см. регистр INTERRUPT_SIGNAL_EN).
- 3. CPU включает режим MDMA в регистре COMMAND.
- 4. СРИ настраивает регистр РАСКЕТ.
- 5. CPU настраивает регистр DMA SYS ADDR.
- 6. CPU настраивает регистр DMA_BUFFER_BOUNDARY.
- 7. CPU настраивает регистр MEMADDR1.
- 8. CPU настраивает регистр MEMADDR2.
- 9. CPU инициирует выполнение команды установкой разряда page_program в регистре PROGRAM.
- 10. DMA контроллер считывает данные из системной памяти, сохраняет в FIFO и отправляет во флэш-память.
- 11. По приходу прерывания от DMA контроллера CPU обновляет значение регистра DMA_SYS_ADDR.
- 12. После завершения выполнения команды NFC устанавливает признак прерывания «Transfer Complete» в регистре INTERRUPT STATUS.
- 13. Формируется физический сигнал прерывания.
- 14. CPU считывает регистр INTERRUPT STATUS.
- 15. CPU сбрасывает бит «Transfer Complete» регистра INTERRUPT SIGNAL EN.
- 16. CPU сбрасывает бит «Transfer Complete» регистра INTERRUPT STATUS EN.
- 17. CPU сбрасывает бит «Transfer Complete» в регистре INTERRUPT_STATUS.



13.3.41 Read (режим MDMA)

Команда Read считывает страницу данных, определяемую адресом строки LUN. Страница данных становится доступной для чтения с заданного адреса столбца. В режиме MDMA данные страницы загружаются в системную память, используя DMA контроллер.

Команда Read выполняется аналогично Page Program в режиме MDMA:

- 1. Пункты 1-8 совпадают с командой Page Program (MDMA).
- 2. Пункт 9: CPU инициирует выполнение команды установкой разряда read в регистре PROGRAM.
- 3. Пункт 10: DMA контроллер считывает данные из FIFO и записывает их в системную память.
- 4. Пункты 11-17 совпадают с командой Page Program (MDMA).

13.4 Обнаружение и инициализация флэш-памяти

В одном корпусе флэш-памяти может содержаться от одного до восьми независимых целевых устройств (target). Выбор необходимого компонента осуществляется отдельным сигналом СЕ#.

Обнаружение и инициализация устройств подключённых к СЕ#

После сброса по включению питания необходимо:

- 1. Отключить защиту памяти.
- 2. Выполнить команду Reset для целевого устройства.
- 3. Определить состояние командой Read Status.
- 4. Выполнить команду Read ID с адресом 20h.
- 5. Убедиться, что данные, считанные командой Read ID, соответствуют стандарту ONFI. В случае несоответствия стандарту, наличия ошибки или истечения времени ожидания, подключать и в дальнейшем использовать проверяемый СЕ# не следует.
- 6. Выполнить команду Read Parameter Page. Данная команда считывает информацию о возможностях и параметрах флэш-памяти. После считывания данных необходимо проверить CRC для подтверждения того, что данные были получены корректно и без ошибок. Если проверка CRC для первой операции Read Parameter Page не прошла, то необходимо считать резервные копии страницы параметров. После успешного извлечения данных из страницы параметров процессор обладает всей необходимой информацией для взаимодействия с целевым устройством.



13.5 Настройка тактирования NFC

Для работы NFC'а необходимо настроить следующие частоты:

- NFC SCLK рабочая частота Nand Flash Интерфейса;
- NFC HCLK системная частота.

Асинхронный режим

В асинхронном режиме частота NFC SCLK должна быть равна 96 МГц.

Синхронный режим

В стандарте ONFI определено 6 синхронных временных режимов и соответствующих им частот. При этом стандарт не требует точного совпадения рабочей частоты и частоты стандартного временного режима. Оговаривается, что если командой Set Features был выбран режим N, то рабочая частота должна быть больше частоты режима N-1 и не должна превышать частоту режима N. Например, для временного режима 2 (частота - 50 МГц) должно выполняться следующее соотношение:

33 М
$$\Gamma$$
ц < NFC SCLK <= 50 М Γ ц

Для режима 0 рабочая частота должны быть выше 10 МГц (подробности и исключения см. в ONFI).

13.6 Прерывания

Работа с прерываниями осуществляется через регистры INTERRUPT_STATUS_EN, INTERRUPT SIGNAL EN, INTERRUPT STATUS.

В регистре INTERRUPT_STATUS_EN задаётся разрешение установки необходимого признака. В регистре INTERRUPT_STATUS отображается текущее состояние признаков прерываний. Регистр INTERRUPT_SIGNAL_EN разрешает или запрещает формирование физического сигнала прерывания.

Возможно формирование прерывания по следующим событиям (см. описание регистра INTERRUPT_STATUS):

- «Buffer Write Ready»;
- «Buffer Read Ready»;
- «Transfer Complete»;
- «Multi Bit Error»;
- «Single Bit Error»;
- «dma int».



13.7 Подключение NAND флэш-памяти

Способ обмена с флэш-устройствами (последовательный или параллельный) определяется значением поля mode регистра MEMADDR2.

13.7.1 Последовательный способ работы

При последовательном способе значение поля chip_select регистра MEMADDR2 определяет флэш-устройство, с которым будет производится обмен данными.

Обмен производится только с одним выбранным устройством.

13.7.2 Параллельный способ работы

При параллельном способе производится одновременный обмен с двумя флэш-устройствами. Параллельное подключение флэш-памяти в основном используется для операций с данными (Read, Page Program, Erase).



14. ПРИНЦИПЫ КОРРЕКЦИИ ОШИБОК

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

Все защищаемые кодом Хэмминга модули памяти (ICACHE, ITAG, DCACHE, DTAG, CRAM и внешняя память DDR) организуются в виде двух блоков: основной блок для хранения данных и блок для хранения контрольных разрядов. Для памятей, имеющих байтовую организацию (CRAM и внешняя память), контрольные разряды формируются операцией "чтение-модификация-запись". Количество контрольных разрядов для 32-разрядных данных – 7 (см. Рисунок 14.1).

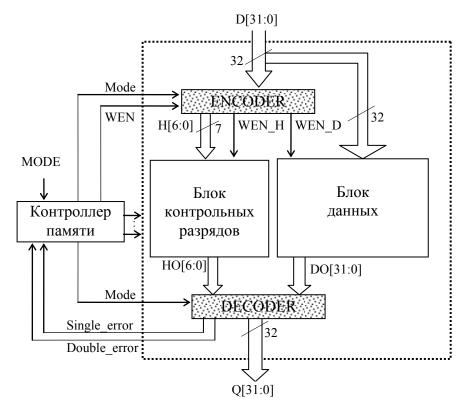


Рисунок 14.1. Структурная схема 32-разрядного модуля памяти с коррекцией ошибок

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) DECODER формирует сигналы Single_Error. При обнаружении двойной ошибки, данные, не корректируются, и формируется сигнал Double_Error.



Каждый модуль памяти имеет регистр управления и состояния CSR: CSR_ICACHE, CSR_DCACHE, CSR_CRAM0A, CRAM0B (для контроля кодом Хэмминга память CRAM разбита на 4 блока объемом по 32 Кбайт; память CRAM имеет 2 порта: A – со стороны CPU и B – со стороны DMA), CSR_CRAM1A, CSR_CRAM1B, CSR_CRAM2A, CSR_CRAM3A, CSR_CRAM3B, CSR_EXT. Формат регистра CSR приведен в Таблица 14.1.

Таблица 14.1. 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
23:16	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 ≤ Num_CERR.	W/R	0

Для CSR_CRAM0, CSR_CRAM1, CSR_CRAM2, CSR_CRAM3 поле MODE едино и может быть записано (и считано) по любому адресу CSR_CRAM[i]. Например, при записи поля MODE в регистр CSR_CRAM2, это же значение принимают все остальные поля MODE регистров CSR_CRAM0, CSR_CRAM1 и CSR_CRAM3.

Основные режимы работы (МОDE) контроллера памяти приведены в Таблица 14.2. 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 памяти.



Таблица 14.2. Режимы работы контроллера памяти

MODE	Разряд- ность	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]}

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 ошибок не формируются.

Ошибки Single_Error накапливаются в счетчике Cnt_SERR. Ошибки Double_Error накапливаются в счетчике Cnt_DERR. Контроллер памяти формирует прерывание при Cnt_CERR > Num_CERR или при обнаружении двойной ошибки. 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 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.

Каждый модуль памяти содержит блок FIFO ошибочных адресов AERROR (AERROR_ICACHE, AERROR_DCACHE, AERROR_CRAM0A, AERROR_CRAM0B, AERROR_CRAM1A, AERROR_CRAM1B, AERROR_CRAM2A, AERROR_CRAM2B, AERROR_CRAM3A, AERROR_CRAM3B, AERROR_EXT), 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 14.3 - Таблица 14.5.

Таблица 14.3. Формат слова FIFO ошибочных адресов AERROR_CRAM0, AERROR_CRAM1, AERROR_CRAM2, AERROR_CRAM3

Номер разряда	Условное обозначение	Назначение
1:0	Code_ERR	Код ошибки.
		0 – нет ошибки
		1 – одиночная ошибка
		2 - двойная ошибка
		3 – ошибка в контрольном разряде общей четности
14:2	ADDR[14:2]	Адрес слова памяти, в котором произошла ошибка.
31:15	-	0



Таблица 14.4. Формат слова FIFO ошибочных адресов ICACHE

Номер разряда	Условное обозначение	Назначение	
1:0	Code_ERR_ICACH	Код ошибки памяти ІСАСНЕ:	
	E	0 – нет ошибки	
		1 – одиночная ошибка	
		2 - двойная ошибка	
		3 – ошибка в контрольном разряде общей четности	
3:2	Code_ERR_ITAG	Код ошибки памяти ITA:	
		0 – нет ошибки	
		1 – одиночная ошибка	
		2 - двойная ошибка	
		3 – ошибка в контрольном разряде общей четности	
15:4	PC[13:2]	Адрес слова, в котором произошла ошибка.	
31:16	-	0	

При возникновении двойной ошибки в ICACHE, ITAG происходит перезапись данной строки в ICACHE из внешней памяти (процедура Refill).

Таблица 14.5. Формат слова FIFO ошибочных адресов DCACHE

Номер разряда	Условное обозначение	Назначение	
1:0	Code_ERR_DCAC	Код ошибки памяти DCACHE.	
	HE	0 – нет ошибки	
		1 – одиночная ошибка	
		2 - двойная ошибка	
		3 – ошибка в контрольном разряде общей четности	
3:2	Code_ERR_DTAG	Код ошибки памяти DTAG.	
		0 – нет ошибки	
		1 – одиночная ошибка	
		2 - двойная ошибка	
		3 – ошибка в контрольном разряде общей четности	
15:4	ADDR[13:2]	Адрес слова, в котором произошла ошибка.	
31:16	-	0	

При возникновении двойной ошибки в DCACHE, DTAG необходимо записать 1 в бит FLUSH D регистра CSR.

Формат слов FIFO ошибочных адресов внешней памяти AERROR_EXT приведен в разделе 6.



15. ПОРТ ЈТА**G** И ВСТРОЕННЫЕ СРЕДСТВА ОТЛАДКИ ПРОГРАММ

В данную микросхему встроен порт JTAG, реализованный в соответствии со стандартом IEEE 1149.1. Этот порт предназначен только для доступа к встроенным средствам отладки программ (OnCD) и не реализует Boundary Scan.

Модуль OnCD обеспечивает:

- выполнение остановки программы CPU по контрольным точкам (Breakpoint);
- выполнение заданного числа команд CPU (трассы) в реальном масштабе времени или пошаговое выполнение команд;
- доступ к адресуемым регистрам и памяти микросхемы.

Для подключения микросхемы к персональному компьютеру через порт JTAG необходимо использовать эмулятор JTAG, предназначенный для работы с данным микропроцессором.



16. ЭЛЕКТРИЧЕСКИЕ И ВРЕМЕННЫЕ ПАРАМЕТРЫ

16.1 Электропитание

Номинальные значения напряжений электропитания микросхемы:

- напряжение питания ядра U_{CCC} (обозначение выводов: CVDD) должно быть 1,8 В;
- напряжение питания входных и выходных драйверов U_{CCP} (обозначение выводов: PVDD) должно быть 3,3 B;
- напряжение питания цифровой части приёмопередатчиков портов
- SpaceFibre/GigaSpaceWire (SpFM) U_{CCD} (обозначение выводов: SpF_VDD) должно быть 1,8 В;
- напряжение питания аналоговой части приёмников портов SpaceFibre/ GigaSpaceWire (SpFM) U_{CCA} (обозначение выводов: SpF_RXVDD) должно быть 3,3 B;
- напряжение питания аналоговой части передатчиков портов SpaceFibre/
- GigaSpaceWire (SpFM) U_{CCA1} (обозначение выводов: SpF_TXVDD) должно быть 1,8 В;
- напряжение питания приёмопередатчиков SSTL портов DDR_PORT U_{CCD1} (обозначение выводов: DDR_PVDD) должно быть 2,5 B;
- относительное напряжение для приёмников SSTL портов DDR_PORT (обозначение выводов: VREF) должно быть 1,25 В.

Допустимые отклонения значения напряжения питания от номинального значения с учётом нестабильности и пульсаций должны быть не более $\pm 5 \%$.

Порядок подачи и снятия напряжений питания и входных сигналов на микросхему должен быть следующим:

- при включении на микросхему сначала подают напряжения питания U_{CCC} , U_{CCD} , а затем напряжения питания U_{CCP} , U_{CCA} . Задержка между подачей напряжений питания U_{CCC} , U_{CCD} и напряжений питания U_{CCP} , U_{CCA} должна быть не более 10 мс. Входные сигналы подают после подачи напряжений питания или одновременно с напряжениями питания U_{CCP} , U_{CCA} ;
- при выключении микросхемы сначала снимают входные сигналы, затем напряжения питания U_{CCP} , U_{CCA} , затем с задержкой не более 10 мс напряжения питания U_{CCC} , U_{CCD} ;
- длительность фронта нарастания напряжения питания должна быть не более 5 мс.

Для фильтрации напряжений электропитания микросхемы, необходимо подключить к каждому источнику (U_{CCP} и U_{CCC}) не менее десяти высокочастотных конденсаторов номиналом 0,1 мкФ типа CC 0603 Y5V 0,1 uF Z 25V. Конденсаторы необходимо разместить по возможности равномерно по периметру 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.

16.2 Электрические параметры

Электрические параметры микросхемы при эксплуатации приведены в Таблица 16.1.

Таблица 16.1. Электрические параметры микросхемы

Наименование параметров,	Буквенное	Норма		Tarranazuna	
единица измерения,	обозначение	не менее	не более	⊤Температура ⁰С	
режим измерения 1 Ток потребления периферии в статическом					
режиме, мА при:	_ 1)			от -60	
$U_{CCC} = 1.9 \text{ B}, \ U_{CCP} = 3.47 \text{ B}, \ U_{CCD} = 1.9 \text{ B},$	$I_{CCP}^{1)}$	_	10	до +85	
$U_{CCA} = 3,47 \text{ B}, U_{CCA1} = 1,9 \text{ B}, U_{CCD1} = 2,62 \text{ B}$					
2 Ток потребления ядра в статическом режиме, мА					
при:	т 1)		200	от -60	
$U_{CCC} = 1.9 \text{ B}, \ U_{CCP} = 3.47 \text{ B}, \ U_{CCD} = 1.9 \text{ B},$	$I_{CCC}^{1)}$	_	200	до +85	
$U_{CCA} = 3,47 \text{ B}, U_{CCA1} = 1,9 \text{ B}, U_{CCD1} = 2,62 \text{ B}$					
3 Ток потребления ядра в динамическом режиме,					
мА при:				от -60	
$U_{CCC} = 1.9 \text{ B}, \ U_{CCP} = 3.47 \text{ B}, \ U_{CCD} = 1.9 \text{ B},$	$I_{OCCC}^{2)}$	_	2000	от -60 до +85	
$U_{CCA} = 3,47 \text{ B}, U_{CCA1} = 1,9 \text{ B}, U_{CCD1} = 2,62 \text{ B},$				до +83	
$f_C = 100 \text{ M}\Gamma_{\text{LL}}$					
4 Ток утечки высокого уровня на входе					
(за исключением выводов таблицы 17.12), мкА				от -60	
при $U_{CCC} = 1.9 \text{ B}, U_{CCP} = 3.47 \text{ B}, U_{CCD} = 1.9 \text{ B},$	I_{ILH}	_	10	до +85	
$U_{CCA} = 3,47 \text{ B}, U_{CCA1} = 1,9 \text{ B}, U_{CCD1} = 2,62 \text{ B},$				до 103	
$2,0 \text{ B} \le U_{IH} \le (U_{CCP} + 0,2) \text{ B}$					
5 Ток утечки низкого уровня на входе					
(за исключением выводов таблицы 17.12), мкА				от -60	
при $U_{CCC} = 1.9 B$, $U_{CCP} = 3.47 B$, $U_{CCD} = 1.9 B$,	I_{ILL}	_	10	до +85	
$U_{CCA} = 3,47 \text{ B}, U_{CCA1} = 1,9 \text{ B}, U_{CCD1} = 2,62 \text{ B},$				A0 . 02	
$0 B \leq U_{IL} \leq 0.8 B$					
5 Выходное напряжение низкого уровня, В					
$при: U_{CCC} = 1,9 \text{ B}, U_{CCP} = 3,47 \text{ B}, U_{CCD} = 1,9 \text{ B},$	U_{OL}	_	0,4	от -60	
$U_{CCA} = 3,47 \text{ B}, U_{CCA1} = 1,9 \text{ B}, U_{CCD1} = 2,62 \text{B},$	COL		, .	до +85	
$I_{OL} = 4.0 \text{ MA}$					
6 Выходное напряжение высокого уровня, В					
при: $U_{CCC} = 1.7 B, U_{CCP} = 3.13 B, U_{CCD} = 1.7 B,$	U _{OH}	2,4	_	от -60	
$U_{CCA} = 3{,}13 \text{ B}, U_{CCA1} = 1{,}7 \text{ B}, U_{CCD1} = 2{,}37 \text{ B},$,		до +85	
I _{OH} = минус 2,8 мА					
7 Входная емкость, пФ	$C_{\rm I}$	_	30	25 ± 10	
8 Емкость входа/выхода, пФ	C _{I/O}	-	30	$ 25 \pm 10 $	
¹⁾ Ток измеряется при уровне $U_{II} = 0$ В на выволе AH26 (XTI)					

 $^{^{(1)}}$ Ток измеряется при уровне U_{IL} = 0 В на выводе АН26 (XTI).

²⁾ Измеряется в режиме функционального контроля.

Значения предельно-допустимых и предельных электрических режимов эксплуатации микросхемы приведены в Таблица 16.2.



Таблица 16.2. Значения предельно-допустимых и предельных электрических режимов эксплуатации

		Норма			
Наименование параметра, единица измерения	Буквенное обозначение	Предельно допустимый режим		Предельный Режим	
единица измерения		не менее	не более	не менее	не более
1. Напряжение питания периферии, В	U_{CCP}	3,13	3,47	_	3,9
2. Напряжение питания ядра, В	Uccc	1,7	1,9	_	2,3
3. Входное напряжение высокого уровня, В	U _{IH}	2,0	U _{CCP} +0,2	_	U _{CCP} +0,3
4. Входное напряжение низкого уровня, В	U _{IL}	0	0,8	минус 0,3	_
5. Емкость нагрузки каждого выхода, пФ	C_{L}	-	30	-	50

16.3 Динамическая потребляемая мощность

Динамическая потребляемая мощность микросхемы имеет две составляющие: потребление ядра (по цепи CVDD) и потребление выходных драйверов (по цепи PVDD).

Мощность, потребляемая ядром микросхемы по цепи CVDD, зависит от последовательности выполняемых процессорными ядрами команд, от операндов, а также от активности DMA и периферийных устройств. Максимальный ток, потребляемый ядром микросхемы, не превышает 2000 мА при внутренней частоте синхронизации 100 МГп.

Мощность, потребляемая выходными драйверами по цепи PVDD, зависит от следующих параметров:

- число выходных драйверов (О);
- максимальная частота, на которой выходные драйверы переключаются (F);
- емкости нагрузки выходных драйверов (С);
- величина напряжения электропитания выходных драйверов (Ucc).

Мощность, потребляемая выходными драйверами по цепи PVDD, определяется следующим уравнением:

$$Pext = O*C* U_{CC}^{2*}F.$$



17. ОПИСАНИЕ ВНЕШНИХ ВЫВОДОВ

Перечень сигналов микросхемы 1892ВК016 по группам, приведен в Таблица 17.1.

Таблица 17.1

Назначение	Число выводов
Порт DDR_PORT, 32 разряда	73
8 портов NFC	41*8=328
Управление	9
2 портов SpaceWire	16
4 порта SpaceFibre/GigaSpaceWire	16
UART	2
Шина SPI	4
Порт ввода-вывода общего назначения	3
Итого сигналов	451
Электропитание	268
Итого	719

Описание выводов микросхемы MCT-04R приведено в Таблица 17.2 - Таблица 17.10.

Таблица 17.2. Порт DDR_PORT

Название вывода	Количество	Тип	Назначение
A[12:0]	13	O	Шина адреса
DQ[31:0]	32	IO	Шина данных
DQH[6:0]	7	IO	Шина данных контроля по коду Хэмминга
Ncs	1	O	Разрешение выборки блоков внешней памяти
RAS	1	O	Строб адреса строки
CAS	1	O	Строб адреса колонки
WE	1	O	Разрешение записи
DQS[3:0]	4	O	Строб данных
DQSH	1	O	Строб данных кода Хэмминга
DM[3:0]	4	O	Маска выборки байта
DMH	1	O	Маска выбора байта кода Хэмминга
CK[1:0],	4	O	Тактовая частота
ckn[1:0]			
CKE	1	O	Разрешение частоты
BA[1:0]	2	О	Номер банка
Всего73 вывода			

Таблица 17.3. NFC (8 штук)

Название вывода	Количество	Тип	Назначение
DFi[15:0]	16	IO	Шина данных
ALEi[1:0]	2	О	Разрешение защелкивания адреса памяти типа NAND Flash
CLEi[1:0]	2	О	Разрешение защелкивания команды памяти типа NAND Flash
nREi[1:0]	2	O	Разрешение чтения памяти типа NAND Flash
nWEi[1:0]	2	O	Разрешение записи памяти типа NAND Flash



Название вывода	Количество	Тип	Назначение	
RBi[7:0]	8	I	Готовность/занятость памяти типа NAND Flash	
nCSFi[7:0]	8	O	Выборка микросхемы	
DQSi	1	I/O	Строб данных	
Всего 41 вывода				

Таблица 17.4. Управление

Название вывода	Коли- чество	Тип	Назначение
nIRQ	1	I	Запросы прерывания. Потенциальные сигналы, активный низкий уровень. Эти сигналы устанавливаются асинхронно источником запроса прерывания. После обработки соответствующего запроса прерывания источник прерывания должен быть сброшен программно
XTI	1	I	Вход системной частоты. Если используется встроенный умножитель частоты (PLL_CORE_EN = 1), то допускается на вход XTI подавать частоту 10 МГц. Если не используется встроенный умножитель частоты (PLL_CORE_EN = 0), то допускается на вход XTI подавать частоту от 1 до 110 МГц. Стабильность входной системной частоты — не хуже +-50 ppm, коэффициент заполнения — от 40 до 60% (скважность — от 1,7 до 2,5), джиттер — не более 1 %
XTI125	1	I	Сигнал тактовой частоты 125 МГц для портов SpaceFibre. Стабильность частоты – не хуже +-50 ppm, коэффициент заполнения – от 40 до 60% (скважность – от 1,7 до 2,5), джиттер – не более 1 %
nRST	1	I	Вход сигнала установки исходного состояния
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	О	Выход данных теста (ЈТАС)
Всего 9 выводов			

Таблица 17.5. Порты SpaceWire (2 штуки)

Название вывода	Коли- чество	Тип	Назначение
DINp0,	2	I	Вход данных положительный
DINp1			
DINn0,	2	I	Вход данных отрицательный
DINn1			
SINp0,	2	I	Вход строба положительный
SINp1			_
SINn0,	2	I	Вход строба отрицательный
SINn1			
DOUTp0,	2	О	Выход данных положительный
DOUTp1			
DOUTn0,	2	О	Выход данных отрицательный
DOUTn1			_



Название вывода	Коли- чество	Тип	Назначение
SOUTp0,	2	O	Выход строба положительный
SOUTp1			
SOUTn0,	2	O	Выход строба отрицательный
SOUTn1			
Всего 16 выводов			

Таблица 17.6. Порты SpaceFibre/GigaSpaceWire (4 штуки)

Название вывода	Тип	Назначение			
SpaceFibre/GigaSpaceWi	SpaceFibre/GigaSpaceWire 0				
SpF_TXP0/TXN0	O	Дифференциальный выход передачи данных.			
SpF_RXP0/RXN0	I	Дифференциальный вход приема данных.			
SpaceFibre/GigaSpaceWi	re 1				
SpF_TXP1/TXN1	О	Дифференциальный выход передачи данных.			
SpF_RXP1/RXN1	I	Дифференциальный вход приема данных.			
SpaceFibre/GigaSpaceWi	ree 2				
SpF_TXP2/TXN2	O	Дифференциальный выход передачи данных.			
SpF_RXP2/RXN2	I	Дифференциальный вход приема данных.			
SpaceFibre/GigaSpaceWire 3					
SpF_TXP3/TXN3	О	Дифференциальный выход передачи данных.			
SpF_RXP3/RXN3	I	Дифференциальный вход приема данных.			
Всего 16 выводов					

Таблица 17.7. UART

Наименование сигнала	Коли- чество	Тип	Назначение
SIN	1	I	Вход последовательных данных
SOUT	1	O	Выход последовательных данных
Всего 2 вывода			

Таблица 17.8. Шина SPI

Наименование сигнала	Коли- чество	Тип	Назначение	
SCK	1	О	Сигнал тактовой частоты	
SO	1	О	Выход данных	
SI	1	I	Вход данных	
CS	1	О	Сигнал выбора внешнего устройства	
Всего 4 вывода				

Таблица 17.9. Порт ввода-вывода общего назначения

Наименование сигнала	Коли- чество	Тип	Назначение
GPIO	3	IO	Сигналы ввода-вывода
Всего 3 вывода			

Таблица 17.10. Электропитание

Название вывода	Количество	Назначение
CVDD	66	Напряжение электропитания ядра (U_{CC2}), (1,8 B)
PVDD	30	Напряжение электропитания входных и выходных драйверов (U_{CC1}), (3,3 B)



Название вывода	Количество	Назначение		
SpF_VDD_0 - SpF_VDD_3	4	Напряжение электропитания цифровой части приемопередатчиков портов SpaceFibre/GigaSpaceWire (2,0 B)		
GND	138	Земля ядра, входных и выходных цифровых драйверов		
SpF_TXVDD_0 - SpF_TXVDD_3	4	Напряжение электропитания аналоговой части передатчиков портов SpaceFibre/GigaSpaceWire (1,8 B)		
SpF _TXGND_0 - SpF _TXGND_3	4	Земля аналоговой части передатчиков портов SpaceFibre/GigaSpaceWire		
SpF_RXVDD_0 - SpF_RXVDD_3	4	Напряжение электропитания элементов защиты и PLL приемников портов SpaceFibre/GigaSpaceWire (3,3 B)		
SpF _RXGND_0 - SpF _RXGND_3	4	Земля электропитания элементов защиты и PLL приемников портов SpaceFibre/GigaSpaceWire		
DDR_PVDD	6	Напряжение электропитания приемопередатчиков SSTL порта DDR_PORT (2,5 B)		
VREF	1	Относительное напряжение для приемников типа SSTL порта DDR_PORT (1,25 B)		
Всего 261 вывод				

Нумерация выводов микросхемы 1892ВК016 в корпусе CPGA-720 приведена в Таблица 17.11.

Таблица 17.11. Нумерация выводов микросхемы 1892BK016 в корпусе CPGA-720

№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
A3	GND	L2	DQS7	V3	DF6[3]
F14	PVDD	L3	RB6[7]	V4	DF6[2]
A29	GND	L4	RB6[6]	W1	DF6[1]
A1	CVDD	L5	RB6[5]	W2	DF6[0]
D1	RB7[3]	D27	GND	W3	DQS6
D2	RB7[2]	B3	CVDD	W4	RB5[7]
E1	RB7[1]	E26	GND	Y1	RB5[6]
E2	RB7[0]	M1	RB6[4]	Y11	PVDD
F1	nCSF7[7]	M2	RB6[3]	L14	GND
F2	nCSF7[6]	M3	RB6[2]	D4	CVDD
F3	nCSF7[5]	M4	RB6[1]	L15	GND
E3	nCSF7[4]	M5	RB6[0]	Y2	RB5[5]
A30	GND	N1	nCSF6[7]	Y3	RB5[4]
A2	CVDD	N2	nCSF6[6]	Y4	RB5[3]
B28	GND	N3	nCSF6[5]	AA1	RB5[2]
F4	nCSF7[3]	R6	PVDD	AA2	RB5[1]
G5	nCSF7[2]	F25	GND	AA3	RB5[0]
Н6	nCSF7[1]	C2	CVDD	AA4	nCSF5[7]
J6	nCSF7[0]	K14	GND	AB3	nCSF5[6]
K6	nWE7[1]	N4	nCSF6[4]	AA10	PVDD
L6	nWE7[0]	N5	nCSF6[3]	L16	GND
M6	nRE7[1]	P5	nCSF6[2]	E4	CVDD



№ вывода	Условное	№ вывода	Условное	№ вывода	Условное
корпуса	обозначение	корпуса	обозначение	корпуса	обозначение
N6	nRE7[0]	R5	nCSF6[1]	AB1	nCSF5[5]
F15	PVDD	T5	nCSF6[0]	AB2	nCSF5[4]
B29	GND	U5	nWE6[1]	AC1	nCSF5[3]
A28	CVDD	V5	nWE6[0]	AC2	nCSF5[2]
B30	GND	P1	nRE6[1]	AC3	nCSF5[1]
G1	ALE7[1]	R25	PVDD	AD1	nCSF5[0]
G2	ALE7[0]	K15	GND	AD2	nWE5[1]
G3	CLE7[1]	C3	CVDD	AD3	nWE5[0]
G4	CLE7[0]	P2	nRE6[0]	L17	GND
H1	DF7[15]	P3	ALE6[1]	AA11	PVDD
H2	DF7[14]	P4	ALE6[0]	L20	GND
Н3	DF7[13]	R1	CLE6[1]	E5	CVDD
H4	DF7[12]	R2	CLE6[0]	AE2	nRE5[1]
P6	PVDD	R3	DF6[15]	AE1	nRE5[0]
C1	GND	R4	DF6[14]	AF1	ALE5[1]
B1	CVDD	T1	DF6[13]	AE3	ALE5[0]
H5	DF7[11]	K16	GND	AF3	CLE5[1]
J1	DF7[10]	Y10	PVDD	AF2	CLE5[0]
J2	DF7[9]	K17	GND	V6	DF5[15]
J3	DF7[8]	C30	CVDD	AG1	DF5[14]
J4	DF7[7]	T2	DF6[12]	L21	GND
J5	DF7[6]	Т3	DF6[11]	F5	CVDD
K1	DF7[5]	T4	DF6[10]	M12	GND
K2	DF7[4]	U1	DF6[9]	W5	DF5[13]
C28	GND	U2	DF6[8]	W6	DF5[12]
P25	PVDD	U3	DF6[7]	Y5	DF5[11]
C29	GND	U4	DF6[6]	Y6	DF5[10]
B2	CVDD	V1	DF6[5]	AB4	DF5[9]
K3	DF7[3]	K20	GND	AB5	DF5[8]
K4	DF7[2]	D3	CVDD	AA5	DF5[7]
K5	DF7[1]	K21	GND	AA6	DF5[6]
L1	DF7[0]	V2	DF6[4]	AD6	PVDD
M13	GND	AF11	DF4[12]	P13	GND
F6	CVDD	AE9	DF4[11]	L10	CVDD
M14	GND	AE10	DF4[10]	AJ17	SpF_RXGND_
AC4	DF5[5]	AE11	DF4[9]	AH17	SpF RXN2
AC6	DF5[4]	AE12	DF4[8]	AH18	SpF_RXP2
AC5	DF5[3]	AF12	DF4[7]	AK17	SpF_RXVDD_
AD5	DF5[2]	N15	GND	P14	GND
AB6	DF5[1]	AF13	PVDD	AE17	SpF VDD 2
AE4	DF5[0]	N16	GND	AJ18	SpF_TXGND_
AD4	DQS5	K12	CVDD	AG17	SpF TXN2
AG2	RB4[7]	AK9	DOUTn1	AG17	SpF_TXP2
					SpF_TXVDD_
AE5	PVDD	AK10	DOUTp1	AK18	2
M15	GND	AJ9	SOUTn1	L11	CVDD
F16	CVDD	AJ10	SOUTp1	P15	GND



№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
M16	GND	AH9	SINn1	AJ19	SpF_RXGND_
AE6	PVDD	AH10	SINp1	AH19	SpF RXN1
M17	GND	AG9	DINn1	AH20	SpF RXP1
F17	CVDD	AG10	DINp1	AK19	SpF_RXVDD_
AK4	RB4[6]	N17	GND	P16	GND
AJ4	RB4[5]	K13	CVDD	AE19	SpF_VDD_1
AH4	RB4[4]	N18	GND	AJ20	SpF_TXGND_
AK5	RB4[3]	AK12	DOUTn0	AG19	SpF_TXN1
AJ5	RB4[2]	AK11	DOUTp0	AG20	SpF_TXP1
AH5	RB4[1]	AJ12	SOUTn0	AK20	SpF_TXVDD_
AG5	RB4[0]	AJ11	SOUTp0	P17	GND
AK6	nCSF4[7]	AH12	SINn0	L12	CVDD
M18	GND	AH11	SINp0	AJ21	SpF_RXGND_ 0
G6	CVDD	AG12	DINn0	AH21	SpF RXN0
M19	GND	AG11	DINp0	AH22	SpF RXP0
AE7	nCSF4[6]	AF14	PVDD	AK21	SpF_RXVDD_ 0
AE8	nCSF4[5]	N19	GND	P18	GND
AJ6	nCSF4[4]	K18	CVDD	AE21	SpF VDD 0
AH6	nCSF4[3]	AJ13	DF4[6]	AJ22	SpF_TXGND_ 0
AG6	nCSF4[2]	AH13	DF4[5]	AG21	SpF_TXN0
AF6	nCSF4[1]	AH14	DF4[4]	AG22	SpF TXP0
AK7	nCSF4[0]	AK13	DF4[3]	AK22	SpF_TXVDD_
AJ7	nWE4[1]	P10	GND	L13	CVDD
AF4	PVDD	AE13	DF4[2]	P19	GND
N12	GND	AJ14	DF4[1]	AJ23	CS
K10	CVDD	AG13	DF4[0]	AH23	SCK
N13	GND	AG14	DQS4	AH24	SO
AH7	nWE4[0]	AK14	nRST	AK23	SI
AG7	nRE4[1]	K19	CVDD	P20	GND
AF7	nRE4[0]	P11	GND	AE23	TMS
AK8	ALE4[1]	AJ15	SpF_RXGND _3	AJ24	TDO
AJ8	ALE4[0]	AH15	SpF_RXN3	AG23	TRST
AH8	CLE4[1]	AH16	SpF_RXP3	AG24	TCK
AG8	CLE4[0]	AK15	SpF_RXVDD _3	AK24	TDI
AF8	DF4[15]	P12	GND	P21	GND
AF5	PVDD	AE15	SpF_VDD_3	AF17	PVDD
N14	GND	AJ16	SpF_TXGND _3	R10	GND
K11	CVDD	AG15	SpF_TXN3	L18	CVDD
AF9	DF4[14]	AG16	SpF_TXP3	AK25	XTI125



№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение	№ вывода корпуса	Условное обозначение
AF10	DF4[13]	AK16	SpF_TXVDD	AJ25	GPIO[2]
AH25	GPIO[1]	AA30		Y28	DF2[11]
AK26	GPIO[0]	AA29	DF3[6]	Y27	DF2[10]
AJ26	nIRQ	R20	GND	W28	DF2[9]
AH26	XTI	N10	CVDD	W27	DF2[8]
AK27	RB3[7]	R21	GND	V28	DF2[7]
AJ27	RB3[6]	AA28	DF3[5]	L25	PVDD
R11	GND	AA27	DF3[4]	T17	GND
AF18	PVDD	AA26	DF3[3]	T25	CVDD
R12	GND	AA25	DF3[2]	T18	GND
L19	CVDD	Y26	DF3[1]	V27	DF2[6]
R13	GND	Y25	DF3[0]	U28	DF2[5]
M10	CVDD	W26	DQS3	U27	DF2[4]
R14	GND	W25	RB2[7]	T28	DF2[3]
AG30	RB3[5]	AG4	PVDD	T27	DF2[2]
AG29	RB3[4]	T10	GND	R28	DF2[1]
AG28	RB3[3]	N11	CVDD	R27	DF2[0]
AF30	RB3[2]	T11	GND	P28	DQS2
AF29	RB3[1]	V26	RB2[6]	U6	CVDD
AF28	RB3[0]	V25	RB2[5]	T19	GND
AF27	nCSF3[7]	T26	RB2[4]	D28	RB1[7]
AE30	nCSF3[6]	U26	RB2[3]	P27	RB1[6]
AF21	PVDD	Y30	RB2[2]	N28	RB1[5]
R15	GND	R26	RB2[1]	N27	RB1[4]
M11	CVDD	W30	RB2[0]	M28	RB1[3]
R16	GND	Y29	nCSF2[7]	M27	RB1[2]
AE29	nCSF3[5]	N25	PVDD	L28	RB1[1]
AE28	nCSF3[4]	T12	GND	L27	RB1[0]
AE27	nCSF3[3]	N20	CVDD	K28	nCSF1[7]
AE26	nCSF3[2]	W29	nCSF2[6]	T20	GND
AD30	nCSF3[1]	V30	nCSF2[5]	K25	PVDD
AD29	nCSF3[0]	V29	nCSF2[4]	T21	GND
AD28	nWE3[1]	U30	nCSF2[3]	U25	CVDD
AD27	nWE3[0]	U29	nCSF2[2]	K27	nCSF1[6]
AF22	PVDD	T30	nCSF2[1]	J30	nCSF1[5]
R17	GND	T29	nCSF2[0]	J29	nCSF1[4]
M20	CVDD	R30	nWE2[1]	H30	nCSF1[3]
AD26	nRE3[1]	T13	GND	H29	nCSF1[2]
AD25	nRE3[0]	N21	CVDD	G30	nCSF1[1]
AC30	ALE3[1]	T14	GND	G29	nCSF1[0]
AC29	ALE3[0]	M25	PVDD	F30	nWE1[1]
AC28	CLE3[1]	R29	nWE2[0]	U10	GND
AC27	CLE3[0]	P30	nRE2[1]	V10	CVDD
AC26	DF3[15]	P29	nRE2[0]	U11	GND
AC25	DF3[14]	N30	ALE2[1]	F29	nWE1[0]
R18	GND	N29	ALE2[0]	E30	nRE1[1]
AG3	PVDD	M30	CLE2[1]	E29	nRE1[0]
R19	GND	M29	CLE2[0]	D30	ALE1[1]
M21	CVDD	L30	DF2[15]	P26	ALE1[0]



№ вывода	Условное	№ вывода	Условное	№ вывода	Условное
корпуса АВ30	обозначение	корпуса Т15	обозначение GND	корпуса N26	обозначение CLE1[1]
AB30 AB29	DF3[13]	T6	CVDD	M26	
AB29 AB28	DF3[12]	T16			CLE1[0]
AB28 AB27	DF3[11]		GND	L26	DF1[15]
	DF3[10]	L29	DF2[14]	J25	PVDD
AB26	DF3[9]	K30	DF2[13]	U12	GND
AB25	DF3[8]	K29	DF2[12]	V11	CVDD
U13	GND	A17	DQS[2]	E23	WE
K26	DF1[14]	B17	DM[2]	E24	RAS
J27	DF1[13]	C26	DQ[15]	E21	CAS
J26	DF1[12]	D26	DQ[14]	E22	nCS
J28	DF1[11]	V12	GND	E19	BA[1]
H28	DF1[10]	F22	DDR_PVD D	E20	BA[0]
H27	DF1[9]	V13	GND	E17	A[12]
H26	DF1[8]	W21	CVDD	E18	A[11]
G28	DF1[7]	C25	DQ[13]	W13	GND
H25	PVDD	D25	DQ[12]	AA12	CVDD
U14	GND	C24	DQ[11]	F19	DDR_PVDD
V20	CVDD	D24	DQ[10]	E16	A[10]
G27	DF1[6]	C23	DQ[9]	C15	A[9]
G26	DF1[5]	D23	DQ[8]	D15	A[8]
F28	DF1[4]	C22	DQS[1]	E15	A[7]
F27	DF1[3]	D22	DM[1]	C14	A[6]
F26	DF1[2]	V14	GND	D14	A[5]
E28	DF1[1]	Y12	CVDD	E14	A[4]
E27	DF1[0]	C27	VREF	C13	A[3]
D29	DQS1	D21	CK[1]	F18	DDR PVDD
U15	GND	C21	CKn[1]	W14	GND
G25	PVDD	D20	DQ[7]	AA13	CVDD
U16	GND	C20	DQ[6]	W15	GND
V21	CVDD	D19	DQ[5]	C12	A[2]
U17	GND	C19	DQ[4]	B11	A[1]
W10	CVDD	D18	DQ[3]	C11	A[0]
F24	NC	C18	DQ[2]	F13	RB0[7]
A27	SOUT	F21	DDR_PVD	E12	RB0[6]
B27	SIN	V15	GND	F11	RB0[5]
A26	DQ[31]	Y13	CVDD	F12	RB0[4]
B26	DQ[30]	V16	GND	E11	RB0[3]
A25	DQ[29]	C17	DQ[1]	W16	GND
B25	DQ[28]	D17	DQ[1]	AA18	CVDD
A24	DQ[28]	C16	DQS[0]	W17	GND
B24	DQ[27]	D16	DM[0]	E10	RB0[2]
W11	CVDD	A16	CK[0]	F10	RB0[1]
W11 U18	GND	B16	CK[0]	E9	RB0[0]
F23	DDR_PVD	A15		F9	
1110		D15	DQH[6]	ЕО	nCSF0[7]
U19	GND	B15	DQH[5]	E8	nCSF0[6]
B23	DQ[25]	V17	GND	F8	nCSF0[5]
A23	DQ[24]	Y18	CVDD	E7	nCSF0[4]



№ вывода	Условное	№ вывода	Условное	№ вывода	Условное
корпуса	обозначение	корпуса	обозначение	корпуса	обозначение
B22	DQS[3]	V18	GND	F7	nCSF0[3]
A22	DM[3]	A14	DQH[4]	W18	GND
B21	DQ[23]	B14	DQH[3]	AH2	PVDD
A21	DQ[22]	A13	DQH[2]	W19	GND
A20	DQ[21]	B13	DQH[1]	AA19	CVDD
B20	DQ[20]	A12	DQH[0]	E6	nCSF0[2]
U20	GND	B12	DQSH	E13	nCSF0[1]
W20	CVDD	A11	DMH	D13	nCSF0[0]
U21	GND	E25	CKE	D12	nWE0[1]
A19	DQ[19]	V19	GND	D11	nWE0[0]
B19	DQ[18]	F20	DDR_PVD D	A10	nRE0[1]
A18	DQ[17]	W12	GND	B10	nRE0[0]
B18	DQ[16]	Y19	CVDD	C10	ALE0[1]
Y14	GND	Y20	GND	AE14	GND
AH3	PVDD	D6	DF0[2]	AE16	GND
AF15	CVDD	A5	DF0[1]	AE18	GND
D10	ALE0[0]	B5	DF0[0]	AE20	GND
A9	CLE0[1]	C5	DQS0	AE22	GND
B9	CLE0[0]	D5	RB7[7]	AE24	GND
C9	DF0[15]	B4	RB7[6]	AE25	GND
D9	DF0[14]	C4	RB7[5]	AF25	GND
A8	DF0[13]	A4	RB7[4]	AF26	GND
B8	DF0[12]	Y21	GND	AG25	GND
C8	DF0[11]	AJ2	PVDD	AG26	GND
AJ1	PVDD	AA14	GND	AG27	GND
Y15	GND	AF20	CVDD	AH1	GND
AF16	CVDD	AF23	CVDD	AH27	GND
Y16	GND	AF24	CVDD	AH28	GND
A7	DF0[10]	AH30	CVDD	AH29	GND
D8	DF0[9]	AK28	CVDD	AJ28	GND
C7	DF0[8]	AJ3	PVDD	AJ29	GND
B7	DF0[7]	AK1	PVDD	AJ30	GND
A6	DF0[6]	AK2	PVDD	AK3	GND
D7	DF0[5]	AA15	GND	AK29	GND
C6	DF0[4]	AA16	GND	AK30	GND
B6	DF0[3]	AA17	GND		
Y17	GND	AA20	GND		
AF19	CVDD	AA21	GND		

В таблица 17.12 представлен перечень выводов с внутренними резисторами в цепи между выводом от источника напряжения U_{CCP} и выводом.



Таблица 17.12. Перечень выводов с внутренними резисторами в цепи между выводом от источника напряжения $U_{\rm CCP}$ и выводом

Номер	Тип	Обозначение
вывода АG23	вывода IR	вывода TRST
AE23	IR	TMS
AK24	IR	TDI
E2	IR	RB7[0]
E1	IR	RB7[0]
D2	IR	RB7[1]
D1	IR	RB7[3]
A4	IR	RB7[4]
C4	IR	RB7[5]
B4	IR	RB7[6]
D5	IR	RB7[7]
M5	IR	RB6[0]
M4	IR	RB6[1]
M3	IR	RB6[2]
M2	IR	RB6[3]
M1	IR	RB6[4]
L5	IR	
L3		RB6[5]
L3	IR	RB6[6]
AA3	IR IR	RB6[7]
AA3		RB5[0]
	IR	RB5[1]
AA1	IR	RB5[2]
Y4	IR	RB5[3]
Y3	IR	RB5[4]
Y2	IR	RB5[5]
Y1	IR	RB5[6]
W4	IR	RB5[7]
AG5	IR	RB4[0]
AH5	IR	RB4[1]
AJ5	IR	RB4[2]
AK5	IR	RB4[3]
AH4	IR	RB4[4]
AJ4	IR	RB4[5]
AK4	IR	RB4[6]
AG2	IR	RB4[7]
AF28	IR	RB3[0]
AF29	IR	RB3[1]
AF30	IR	RB3[2]
AG28	IR	RB3[3]
AG29	IR	RB3[4]
AG30	IR	RB3[5]
AJ27	IR	RB3[6]
AK27	IR	RB3[7]
W30	IR	RB2[0]
R26	IR	RB2[1]
Y30	IR	RB2[2]
U26	IR	RB2[3]



Номер вывода	Тип вывода	Обозначение вывода
T26	IR	RB2[4]
V25	IR	RB2[5]
V26	IR	RB2[6]
W25	IR	RB2[7]
L27	IR	RB1[0]
L28	IR	RB1[1]
M27	IR	RB1[2]
M28	IR	RB1[3]
N27	IR	RB1[4]
N28	IR	RB1[5]
P27	IR	RB1[6]
D28	IR	RB1[7]
E9	IR	RB0[0]
F10	IR	RB0[1]
E10	IR	RB0[2]
E11	IR	RB0[3]
F12	IR	RB0[4]
F11	IR	RB0[5]
E12	IR	RB0[6]
F13	IR	RB0[7]



18. ИСТОРИЯ ИЗМЕНЕНИЙ

18.1 25 мая 2017 г

- уточнена табл. 2.1. Введен следующий текст пол этой таблицей: Следует отметить, что виртуальный базовый адрес исключений 0x8000_0000 будет размещен во внешней памяти по физическому адресу 0x0000_0000. Виртуальный базовый адрес исключений 0xBFC0_0000 будет размещен во внешней памяти по физическому адресу, который зависит от объема внешней памяти следующим образом: 32 Мбайт 0x01C0_0000, 64 Мбайт 0x03C0_0000, 128 Мбайт 0x07C0_0000;
- уточнен п. 2.2.3;
- уточнены табл. 2.4 в части разрядов (31:24) и табл. 2.5 в части разрядов (6, 0);
- уточнена табл. 2.5 в части 6 разряда;
- уточнена табл. 2.2 в части диапазона адресов ПЗУ начальной загрузки;
- уточнена табл. 2.10;
- уточнена табл. 3.32;
- уточнена табл. 2.3 в части системных регистров;
- уточнены табл. 2.12, 2.13;
- в п. 2.4 введено описание формата регистра ВООТ;
- уточнен и дополнен п. 2.5;
- уточнена табл. 3.5. Введен текст под этой таблицей;
- уточнена табл. 6.7 в части разряда 8;
- комментарии под табл. 6.7 дополнены текстом: Для инициализации работы DDR_PORT и внешней памяти DDR необходимо сначала выполнить последовательно команды INIT, EYEW, EYEW. Затем в регистре DDR_MOD установить разряды TMODE=1, SEL=0x1F и выполнить последовательно команды APPLY, INIT, EYEW, EYEW;
- уточнена табл. 16.10 в части электропитания SpF VDD 0 SpF VDD 3;
- уточнены табл. 13.17, 13.18, 13.19;
- уточнен п. 13.1.1.



18.2 20 ноября 2017 г

• уточнена табл. 17.11 в части выводов DF6[13], DF6[14].

18.3 29 января 2018 г

• в п. 12.1.3 добавлено: «В данной микросхеме внешним выводам SCK, SO, SI, CS (шина SPI) соответствуют выводы порта TSCK, MOSI, MISO, SS[0] соответственно».

18.4 15 марта 2018 г

• в таблице 17.4. «Управление» поля «Назначение» для входов XTI и XTI125 уточнены.

18.5 18 декабря 2018 г

- уточнено описание регистра QSTR3 в таблице 2.9;
- уточнено описание подключения SPI-флэш в разделе 2.5;
- уточнено описание регистра QSTR2 в таблице 2.2.

18.6 16 сентября 2019 г

- скорректирована таблица 11.7, 11.8 в части номеров битов (rf#5325);
- раздел 10.6.6 скорретирован (rf#11522);
- таблица 2.9: скорректировано содержимое (rf#9976);
- таблицы 11.7 и 11.8 скорректированы значения полей регистов (rf#5325);
- раздел 10.5.1 исправлен номер аппаратной версии SWIC (rf#9283);
- названия IT0/IT1 по тексту заменены на IRT0/IRT1 (rf#5245);
- таблица 2.8: скорретированы номера разрядов (rf#9831).