

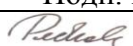
Удостоверен

ЮФКВ.431268.006-УД

Микросхема интегральная 1879ВЯ1Я

Руководство по эксплуатации

ЮФКВ.431268.006РЭ

Инв.№подл.	Подп. и дата	Взам.инв.№	Инв.№дубл.	Подп. и дата	Справ.№
22755-2	 16.01.14	22755-1			

Содержание

1	Введение.....	8
2	Общая структура СБИС СПОВС	9
3	Карта памяти СБИС СПОВС	11
4	Процессорная система на базе DSP ядра NeuroMatrix Core3 (NMU).....	12
4.1	Интервальные таймеры TIMER0 и TIMER1.....	15
4.2	Системный контроллер процессорной системы NMU (NMCSC).....	17
4.3	Контроллер прерываний (INTC)	18
4.4	PU (Permutation Unit) – блок упаковки/распаковки векторов данных..	21
4.4.1	Аппаратно поддерживаемые операции	21
4.4.2	Режимы и порядок работы PU.	25
4.4.3	Конфигурационные регистры	26
4.5	Список периферийных регистров для ядра NMC3	33
5	Процессорная система на базе ядра ARM11 (ARMU).	34
5.1	Микропроцессорное ядро ARM1176JZF.....	38
5.2	Блоки поддержки отладки — ETM11, ETB11 и JTAG интерфейс	41
5.2.1	Устройство блоков аппаратной поддержки отладки.....	41
5.2.2	Устройство порта JTAG и подключение к внешнему аппаратному отладчику	42
5.3	Системный коммутатор шин HPM_S	46
5.3.1	Параметры портов коммутатора HPM_S	47
5.3.2	Регистровая модель коммутатора HPM_S.....	50
5.4	Периферийный коммутатор шин HPM_P	55
5.4.1	Параметры портов коммутатора HPM_P	56
5.4.2	Регистровая модель коммутатора HPM_P.....	57
5.5	Блок управления сигналами сброса и синхросигналами CRG11	58
5.5.1	Устройство блока CRG11	58
5.5.2	Генератор сигналов сброса	60
5.5.3	Генератор синхросигналов.....	63

					ЮФКВ.431268.006РЭ			
Из	Лист	№ докум.	Подп.	Дата	Микросхема интегральная 1879ВЯ1Я Руководство по эксплуатации			
Разраб.	Косоруков							
Пров.								
Нач.отд.	Панфилов							
Н.контр.	Вихрова							
Утв.								
Инв.№подл.	Подп. и дата		Взам.инв.№	Инв.№подл.	Подп. и дата		Перв. применен.	
22755-2	 16.01.14		22755-1					

5.5.4	Регистровая модель блока CRG11.....	70
5.6	Системный векторный контроллер прерываний VIC	79
5.6.1	Устройство векторного контроллера прерываний	79
5.6.2	Подробности работы с контроллером VIC.....	82
5.6.3	Последовательность шагов при обработке прерываний.	84
5.6.4	Регистровая модель системного контроллера прерываний VIC	85
5.7	Контроллер внешних прерываний EXTIRC	94
5.7.1	Устройство контроллера внешних прерываний	94
5.7.2	Регистровая модель контроллера внешних прерываний EXTIRC	95
5.7.3	Прерывания	96
5.8	Блок двояных таймеров DIT.....	97
5.8.1	Устройство блока двояных таймеров	97
5.8.2	Регистровая модель блока двояных таймеров	98
5.8.3	Прерывания	104
5.9	Сторожевой таймер WDOG	105
5.9.1	Устройство блока сторожевого таймера	105
5.9.2	Регистровая модель блока сторожевого таймера	106
5.9.3	Прерывания	110
5.10	Таймер реального времени RTC.....	111
5.10.1	Устройство блока таймера реального времени.....	111
5.10.2	Спецификация регистров.	111
5.10.3	Описание регистров.....	112
5.10.4	Прерывания.....	118
5.11	Системный контроллер ARMSC.....	120
5.11.1	Устройство системного контроллера ARMSC.....	120
5.11.2	Регистровая модель контроллера ARMSC.....	122
5.11.3	Прерывания.....	128
5.12	Контроллер ПДП DMAC	129
5.12.1	Устройство контроллера ПДП.....	129
5.12.2	Регистровая модель контроллера ПДП	130
5.12.3	Описание работы каналов.....	135
5.12.4	Прерывания.....	136
5.13	Контроллер асинхронного последовательного интерфейса UART.....	137
5.13.1	Устройство UART портов СБИС СПОВС.....	137
5.13.2	Описание функционирования UART порта.....	139
5.13.3	Регистровая модель контроллера порта UART	140
5.13.4	Прерывания.....	147
5.14	Контроллер последовательного синхронного интерфейса SPI.....	148
5.14.1	Устройство SPI порта СБИС СПОВС	148
5.14.2	Протокол передачи данных в различных режимах SPI порта.....	150
5.14.3	Регистровая модель контроллера порта SPI	152
5.14.4	Прерывания.....	159

					ЮФКВ.431268.006РЭ			Лист
								3
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

5.15	Контроллер шины USB2.0	161
5.15.1	Устройство контроллера USB	162
5.15.2	Регистровая модель контроллера USB	162
5.15.3	Регистры приема и передачи данных	164
5.15.4	Регистры состояния	165
5.15.5	Регистры управления	170
5.15.6	Прерывания.....	179
5.15.7	Рекомендуемые алгоритмы работы с контроллером USB.....	180
5.16	Блок портов общего назначения GPIO.....	188
5.16.1	Устройство портов общего назначения	188
5.16.2	Регистровая модель контроллера портов GPIO	189
5.16.3	Прерывания.....	190
5.17	Загрузочное ПЗУ BROM.....	191
5.17.1	Служебное слово загрузчика.....	191
5.17.2	Алгоритм работы начального загрузчика	193
5.17.3	Состояние процессора перед началом исполнения загруженной программы	194
5.17.4	Пример загружаемой программы	195
5.17.5	Начальная загрузка через SPI.	196
5.17.6	Начальная загрузка через UART.	197
5.17.7	Начальная загрузка через USB.....	199
5.17.8	Использование GPIO для индикации состояния начального загрузчика.	201
5.18	Контроллер динамической памяти DMC.....	203
5.18.1	Описание функционирования контроллера динамической памяти DMC	203
5.18.2	Динамическое управление энергопотреблением.	205
5.18.3	Настраиваемые временные параметры.....	206
5.18.4	Регистровая модель контроллера динамической памяти DMC	208
6	Система общей памяти (SMU).	220
7	Блок интервальных таймеров (ITU).	221
7.1	Принцип работы блока ITU.....	221
7.2	Конфигурационные регистры блока ITU.	224
8	Контроллер ПДП (DMAC)	227
8.1	Принцип работы блока DMAC.....	227
8.1.1	Каналы ПДП "память – память" (MEMORY-MEMORY DMA CHANNELS)	227
8.1.2	Канал ПДП "память – DSPPU"	228
8.2	Конфигурационные регистры блока DMAC	229

					ЮФКВ.431268.006РЭ			Лист
								4
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

9	Блок аналогово-цифрового преобразования (ADCU).....	233
10	Блок предварительной обработки сигналов (DSPPU)	236
10.1	Типы данных, используемые DSPPU	240
10.2	Входное устройство (Input_unit).....	242
10.3	Цифровой гетеродин (Het).....	244
10.4	Кластер генераторов псевдослучайных последовательностей	245
10.4.1	Генератор псевдослучайных последовательностей (CODE GENERATOR).....	246
10.5	Аппаратный канал предварительной обработки сигналов (Channel) .	248
10.5.1	Округление результатов вычислений в каналах.....	250
10.6	Блоки нормализации (NORM).....	251
10.7	Кластер КИХ-фильтров (FIR FILTER)	252
10.8	Блок квадраторов (SQR).....	254
10.9	Блоки упаковки (PACK_UNIT и ADC_PACK_UNIT)	256
10.10	Блок адресных генераторов (AG)	258
10.11	Прерывания	259
11	Электрические и конструктивные характеристики СБИС СПОВС.....	260
11.1	Состав и расположение внешних выводов СБИС СПОВС.....	260
11.2	Конструктивные характеристики	270
11.3	Электрические характеристики.....	272
11.4	Порядок подачи питания при включении	274
12	Временные характеристики	275
12.1	Временные параметры SPI порта	275
13	Рекомендации по включению	277
13.1	Рекомендации по включению кварцевого резонатора для счетчика реального времени и для тактового сигнала USB.....	277
13.2	Рекомендации по подключению тактового сигнала микросхемы (до 82МГц)	277
13.3	Рекомендации по включению АЦП.....	279
13.4	Рекомендации по реализации схемы начального сброса	280
13.5	Рекомендации по подключению USB интерфейса	281
13.6	Рекомендации по включению DDR.....	282

					ЮФКВ.431268.006РЭ			Лист
								5
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редкофф</i> 16.01.14		22755-1					

13.6.1	Рекомендации по согласованию линий связи интерфейса DDR1 SDRAM	282
13.6.2	Рекомендации по проектированию печатной платы для DDR интерфейса	283

А Приложение. Система команд DSPPU 286

A.1	Основные режимы работы.....	286
A.2	Типы команд блока DSPPU	288
A.3	Команды записи	289
A.3.1	WR_w_filter_RAM (0-2) (0-3) (0-15)	289
A.3.2	WR_het_B / WR_FQ (0/1) (0-27)	291
A.3.3	WR_select (0Ah) (0-27)	292
A.3.4	WR_het_counter (2) (0-23)	294
A.3.5	WR_norm_h (3) (0-23), WR_norm_f / WR_norm_q (5/6) (0-11)	295
A.3.6	WR_filter_mask (4) (0-3/4-7/8-11)	296
A.3.7	WR_quad_counter (0Ch) (0-11)	298
A.3.8	WR_pack (7) (0-23)	298
A.3.9	WR_TCG / WR_TM / WR_TIS (2/5/6) (24-27)	299
A.3.10	WR_TCR (4) (24-27)	300
A.3.11	WR_ADC_control (0) (28)	302
A.3.12	WR_ADC_ins (1) (28)	304
A.3.13	WR_ADC_delta (2) (28)	305
A.3.14	WR_ADC_calibration (4-7) (28)	305
A.3.15	WR_Ms_counter (1-7) (29)	306
A.3.16	WR_Start / WR_Start_exp_h / WR_Start_exp_f / WR_Start_exp_q (1Ch/1Dh/1Eh/1Fh) (31)	307
A.3.17	WR_Stop / WR_Stop_ms (1Ah/1Bh) (31)	308
A.3.18	WR_Mask_trap (18) (31)	308
A.3.19	WR_Reset (0) (31)	309
A.4	Команды чтения.....	309
A.4.1	RD_het_base / RD_frequency / RD_current (0/1/8) (0-27)	309
A.4.2	RD_het_control (2) (0-23)	310
A.4.3	RD_norm_h / RD_norm_f / RD_norm_q (3/5/6) (0-23/0-11/0-11)	311
A.4.4	RD_filter_control (4) (0-3, 4-7, 8-11)	312
A.4.5	RD_quad_control (0Ch) (0-11)	312
A.4.6	RD_TCGR (2) (24-27)	313
A.4.7	RD_ADC_control (0) (28)	313
A.4.8	RD_Ms_counter (1-7) (29)	314
A.4.9	RD_Channel_check (7) (0-27)	314
A.4.10	RD_ADC_check (7) (28)	315
A.5	Команды для адресных генераторов блока DSPPU	317
A.5.1	WR_AG_adr_RAM (580h – 5CFh)	317
A.5.2	WR_AG_size / RD_AG_size (5FDh)	317
A.5.3	WR_AG_control_monitor / RD_AG_control_monitor (5Feh)	318
A.5.4	RD_monitor_count (5FCh)	320
A.5.5	RD_stack_monitor (5FFh)	320

					ЮФКВ.431268.006РЭ			Лист
								6
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

A.5.6 RD_ms_part_monitor (5FBh)321

В Приложение. Таблица преобразования синуса/косинуса в гетеродине. 323

					ЮФКВ.431268.006РЭ			Лист
								7
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата	
22755-2		<i>Редкоз</i> 16.01.14			22755-1			

1 Введение

Микросхема интегральная 1879ВЯ1Я – СБИС аналого-цифровой системы приема и обработки высокочастотных сигналов (далее по тексту СБИС СПОВС) класса система-на-кристалле, обеспечивает прием аналоговых сигналов, преобразование их в цифровой код и программную цифровую обработку.

СБИС СПОВС предназначена для создания унифицированной аппаратно-программной платформы цифровых программных приемников, включая:

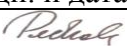
- приемники цифрового радиовещания (ЦРВ);
- радиолокационные приемники;
- приемники сигналов сотовой связи GSM, CDMA и др.
- многосистемные навигационные приемники ГЛОНАСС/ GPS/ GALILEO/ COM-PASS;

- другие задач цифровой обработки сигналов

В состав СБИС входят: тракт приема аналоговых сигналов (АЦП) и предварительной обработки сигналов (БПОС), блок первичной цифровой обработки на основе двух процессоров NeuroMatrix® NMC3 (DSP процессоры), RISC процессор вторичной обработки ARM1176JZF-S, внутреннее ОЗУ объемом 16 Мбит, блоки синхронизации, устройство контроля JTAG, а также широкий набор периферийных устройств.

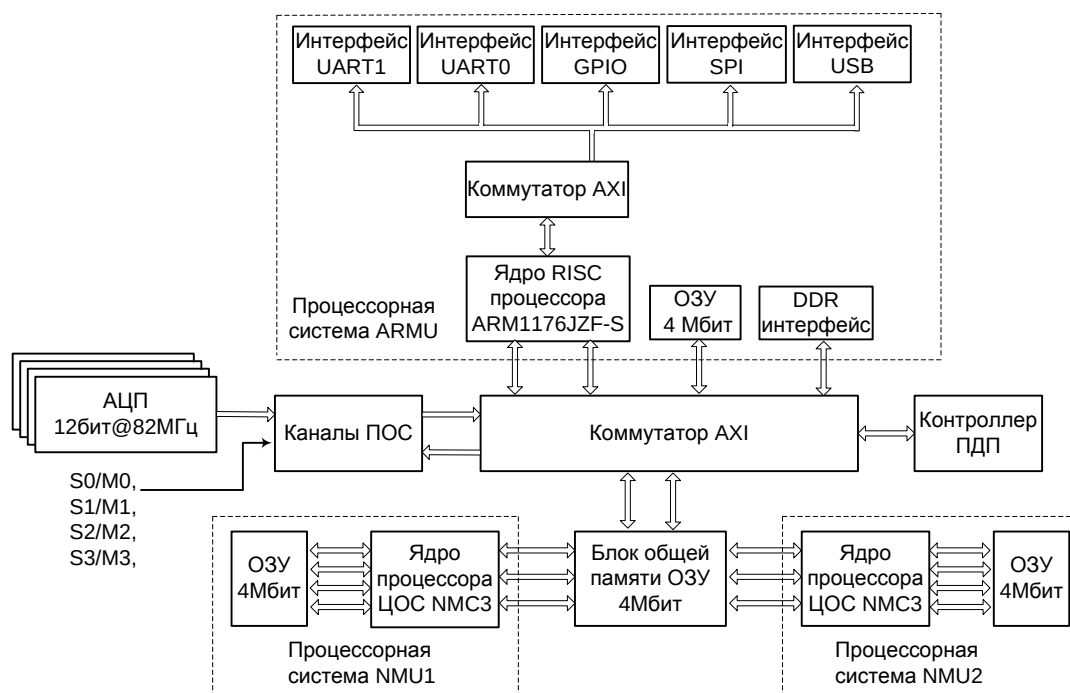
СБИС СПОВС работает на внутренней тактовой частоте 328 МГц.

Документ содержит описание общей структурной схемы СБИС СПОВС, карту памяти для всех процессорных ядер, входящих в состав СБИС, и подробное описание ее основных подсистем.

					ЮФКВ.431268.006РЭ			Лист
								8
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

2 Общая структура СБИС СПОВС

Общая структурная схема СБИС СПОВС приведена на рисунке ниже (см. Рисунок 2.1).



NMU – NeuroMatrix Unit

SMU – Shared Memory Unit

ADCU – Analog-to-Digital Converter Unit

DSPPU – Digital Signal PreProcessing Unit

DMAC – DMA Controller

ITU – Interval Timer Unit

ARMU – ARM Unit

SPI – Serial Peripheral Interface

UART – Universal Asynch Receiver Transmitter

GPIO – General Purpose Input/Output

USB – Universal Serial Bus

Рисунок 2.1 – Общая структурная схема СБИС СПОВС

СБИС содержит следующие функциональные узлы:

ARMU (ARM Unit) – процессорная система, состоящая из микропроцессорного ядра ARM11, работающего на частоте 164 МГц, памяти объемом 4 Мбит, развитой шинной структуры с набором коммутаторов, а также периферийных устройств. Более подробно процессорная система ARMU будет описана в разделе 5.

NMU1 (NeuroMatrix Unit 1) и NMU2 (NeuroMatrix Unit 2) - процессорные системы, каждая из которых состоит из DSP ядра NeuroMatrix Core 3, памяти объемом 4 Мбит, развитой шинной структуры с набором коммутаторов, а также периферийных

					ЮФКВ.431268.006РЭ			Лист
								9
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

устройств. Более подробно процессорная система NMU будет описана в разделе 4. Все узлы системы NMU работают на частоте 328 МГц.

SMU (Shared Memory Unit) – система общей памяти объемом 4 Мбит, доступная для каждой процессорной системы: **ARMU**, **NMU1** и **NMU2**. Более подробно система общей памяти SMU будет описана в разделе 6. Все узлы системы SMU работают на частоте 328 МГц.

ADCU (Analog-to-Digital Converter Unit) – блок аналогово-цифрового преобразования, состоящий из четырёх 12-разрядных АЦП, работающих на частоте 82 МГц.

DSPPU (Digital Signal PreProcessing Unit) - блок предварительной обработки сигналов, который состоит из 24 аппаратных каналов, осуществляющих следующие операции:

выбор источника сигнала (выход одного из АЦП блока ADCU, входы Sign & Magn или внутренняя память),

цифровое гетеродинирование с последующим накоплением,

фильтрацию помех и возведение комплексного сигнала в квадрат,

упаковку сигнала в 64-х разрядные слова и генерацию адреса для записи в память упакованных данных.

Управление данным блоком осуществляется процессорными системами NMU1 и NMU2.

DMAC (DMA Controller) – контроллер ПДП, обеспечивающий обмен между памятью и DSPPU, обмен память – память. Контроллер управляется процессорными системами NMU1 и NMU2.

ITU (Interval Timer Unit) – блок интервальных таймеров. Блок управляется процессорными системами NMU1 и NMU2.

DMC (Dynamic Memory Controller) – контроллер динамической памяти, реализующий обмен с внешней памятью типа DDR SDRAM по 32-разрядной шине данных с тактовой частотой 164 МГц.

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

UART (Universal Asynch. Receiver-Transmitter) - контроллер асинхронного последовательного интерфейса.

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

USB (Universal Serial Bus) – контроллер интерфейса последовательной шины USB, соответствующей спецификации 2.0 (Full-speed) и работающий как USB device.

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

Все функциональные узлы СБИС СПОВС соединяются шинами трёх типов:

64-разрядные шины, позволяющие производить обмен данными на частоте 328 МГц;

64-разрядные шины, позволяющие производить обмен данными на частоте 164 МГц;

32-разрядные шины, позволяющие производить обмен данными на частоте 41 - 82 МГц.

					ЮФКВ.431268.006РЭ			Лист
								10
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

3 Карта памяти СБИС СПОВС

Карта памяти СБИС СПОВС приведена на рисунке ниже (см. Рисунок 3.1):

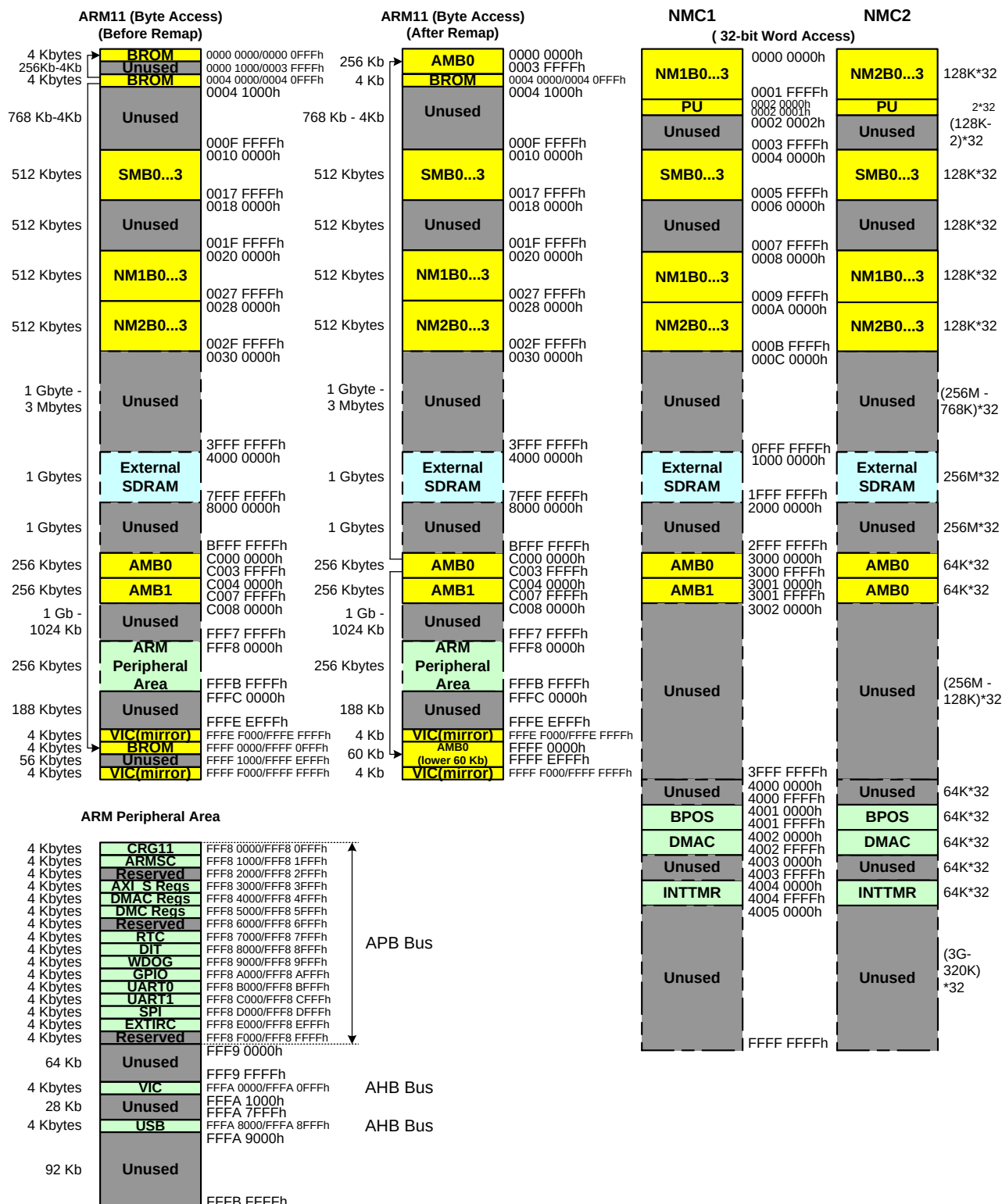
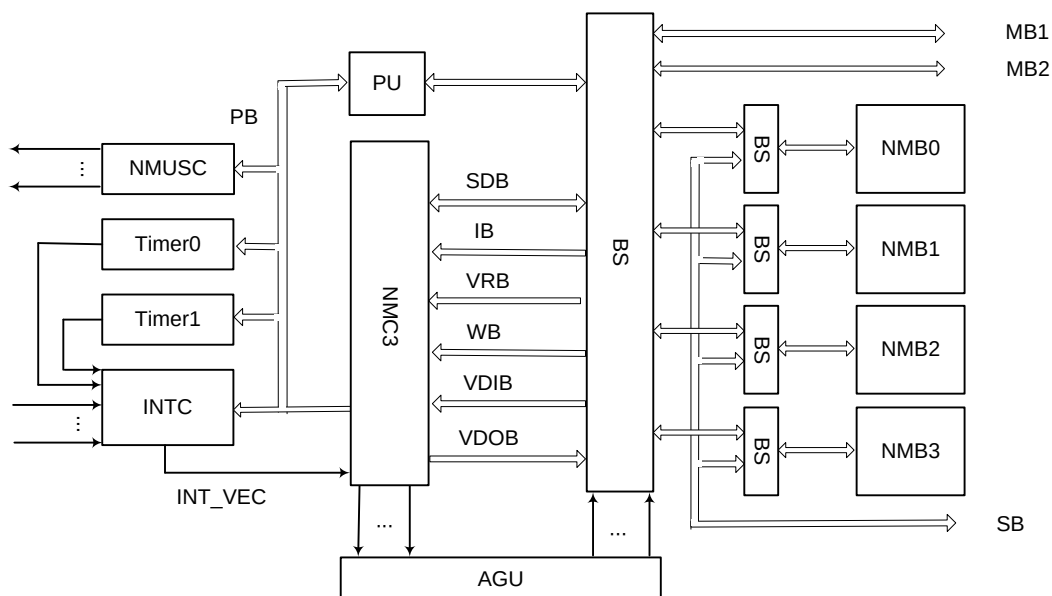


Рисунок 3.1 - Карта памяти СБИС СПОВС

					ЮФКВ.431268.006РЭ			Лист
								11
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		16.01.14		22755-1				

4 Процессорная система на базе DSP ядра NeuroMatrix Core3 (NMU)

Структурная схема процессорной системы на базе DSP ядра NeuroMatrix Core 3 (NMC3) приведена ниже (см. Рисунок 4.1).



NMC3 – Neuro Matrix Core 3
INTC – Interrupt Controller
NMUSC – NMU System Controller
PU – Permutation Unit
AGU – Address Generator Unit
BS – Bus switch
NMB – Neuro Matrix Memory Bank 16K x 64

PB – Peripheral Bus
SDB – Scalar Data Bus
VDIB – Vector Data Input Bus
VDOB – Vector Data Output Bus
WB – Weight Bus
VRB – Vector Register Bus
xxx_MB – Master Bus
xxx_SB – Slave Bus

Рисунок 4.1 – Структурная схема процессорной системы на базе DSP ядра NeuroMatrix Core 3

Процессорная система на базе DSP ядра NeuroMatrix Core 3 работает на частоте 328 МГц и содержит следующие функциональные узлы:

NMC3 (NeuroMatrix Core 3) – DSP ядро NeuroMatrix Core 3, которое представляет собой ядро высокопроизводительного векторно-матричного процессора с оригинальной динамической суперскалярной параллельной архитектурой и сверхбольшими словами команд. Обмен ядра с внешним миром осуществляется с помощью шести 64-разрядных шин: команд (IB), скалярных данных (SDB), векторных входных данных (VDIB), весов (WB), векторного регистра (VRB) и векторных выходных данных (VDOB). Описание ядра NMC3 находится в отдельном документе: “СФ - БЛОК NMC3 Техническое описание ЮФКВ.431282.007ТО”.

NMUSC (NMU System Controller) – системный контроллер процессорной системы NMU.

					ЮФКВ.431268.006РЭ			Лист
								12
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	16.01.14		22755-1					

Данный контроллер содержит программно-доступный со стороны NMC3 периферийный регистр, который выполняет функции управления прерываниями другим процессорным системам NMU и ARMU и осуществляет вывод на внешние выходы тестовых сигналов.

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

- высокоприоритетное и низкоприоритетное от процессорной системы NMU;
- высокоприоритетное и низкоприоритетное от процессорной системы ARMU;
- три прерывания от блока DSPPU;
- три прерывания от блока ITU;
- одно прерывание от блока DMAC;
- одно прерывание от блока PU;
- одно прерывание от блока TIMER0;
- одно прерывание от блока TIMER1;

а также соответствующий ему адрес-вектор прерывания (INT VEC).

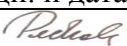
TIMER0 и **TIMER1** – два интервальных 32-разрядных таймера.

Таймеры идентичны по функциям, независимы друг от друга и предназначены для отсчета задаваемых интервалов времени для ядра NMC3. Каждый таймер содержит 32-разрядный счетчик, осуществляющий отсчет временных интервалов. Данный счетчик состоит из рабочего и теневого регистров.

Каждый из таймеров может работать как в непрерывном режиме, так и в режиме однократного запуска. Интервал счета таймера задается программно. В качестве сигнала счета выступает тактовый сигнал процессора. По достижении нулевого значения таймер формирует сигнал прерывания, который может быть обработан блоком прерываний стандартным образом.

PU (Permutation Unit) – блок упаковки/распаковки векторов данных. Данный блок выступает в роли сопроцессора для NMC3, и его задачей является преобразование потоковых данных в такой формат, который более удобен для обработки в NMC3 (распаковка) или для хранения в памяти (упаковка). Соответственно, применение блока PU позволяет резко повысить эффективность ядра NMC3 на таких задачах, как корреляция или фильтрация, при этом хранить коэффициенты для той же корреляции или фильтрации, а также результаты обработки в компактном виде. Блок PU аппаратно поддерживает следующие операции:

- Распаковка входного вектора в выходной с элементами заданной большей разрядности (до 64 разрядов), чем в исходном векторе, с возможностью распространения знака или нуля внутри элемента.
- Упаковка входного вектора в выходной с элементами заданной меньшей разрядности (до 1 разряда), чем в исходном векторе, с возможностью выбора любых необходимых разрядов (соседних) внутри элементов с отбрасыванием остальных.
- Преобразование входного вектора в диагональную матрицу (может использоваться для ускорения скалярного произведения векторов в ядре NMC3).
- Запись элементов входного вектора в столбцы выходной матрицы или обратное этому преобразование.
- Доступ к векторам, хранящимся в памяти, с адресацией (начальным смещением) до элемента этих векторов (вплоть до 1 разряда).
- Возможность работы с блоками данных в памяти как с циклическими буферами (вычисление адреса данных по модулю заданного числа).

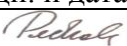
					ЮФКВ.431268.006РЭ			Лист
								13
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Более подробно PU описан в подразделе 4.4.

AGU (Address Generator Unit) – блок адресных генераторов, который получает запрос на обмен данными с памятью от ядра NMC3 (начальный и конечный адрес, смещение, число обращений) и затем сам формирует адреса, освобождая от этого процессор. Благодаря 6 шинам ядра NMC3 и наличию в блоке 6 адресных генераторов, возможно осуществлять до шести операций ввода-вывода за один такт.

BS (Bus Switch) – шинный коммутатор.

NMB (NeuroMatrix Memory Bank) - банк памяти объемом 16К×64, который может адресоваться до 32- или 64-разрядного слова.

					ЮФКВ.431268.006РЭ			Лист
								14
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

4.1 Интервальные таймеры TIMER0 и TIMER1.

Таймеры идентичны по функциям, независимы друг от друга и предназначены для отсчета интервалов времени для ядра NMC3. Таймер содержит 32-разрядный счетчик, осуществляющий отсчет временных интервалов. Данный счетчик состоит из рабочего и теневого 32-х разрядных регистров (см. Рисунок 4.2).

Таймер может работать как в непрерывном режиме, так и в режиме однократного запуска. Интервал счета таймера задается программно. В качестве сигнала счета выступает тактовый сигнал процессора. По достижении нулевого значения таймер формирует сигнал прерывания.

Таймер имеет следующие программно доступные регистры: рабочий регистр счетчика TMR_CNTw(доступен на чтение), теновой регистр счетчика TMR_CNTs(доступен на запись), регистр состояния TMR_MODE(доступен на запись и на чтение).

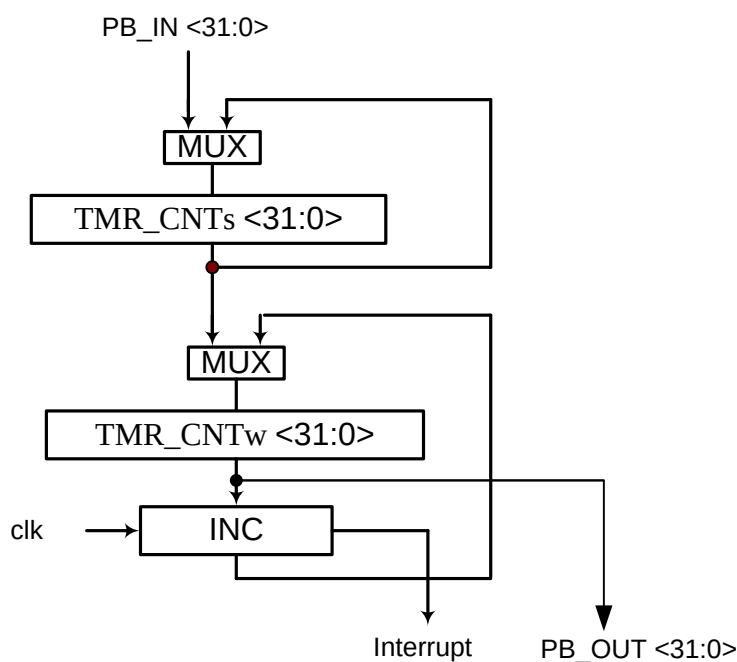


Рисунок 4.2- Структурная схема таймера.

Теновой и рабочий регистр счетчика имеют одинаковый формат (см. Рисунок 4.3).

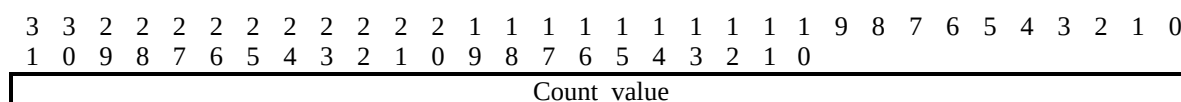


Рисунок 4.3 – Регистр счетчика.

Поле **Count_value** (разряды 31-0) задает интервал счета.

Регистр состояния управляет режимом работы и запуском таймера.

					ЮФКВ.431268.006РЭ					Лист
										15
Изм.	Лист	№ докум.	Подп.	Дата						
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата				
22755-2	Решет 16.01.14		22755-1							

3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0
1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0

зарезервировано

E M

Рисунок 4.4 – Формат регистра состояния TMR_MODE.

0-й разряд (**M**) задает режим работы таймера:

0 – режим многократного запуска;

1 – режим однократного запуска.

1-й разряд (**E**) разрешает или запрещает работу таймера:

0 – таймер не работает;

1 – таймер работает.

В теневой регистр счетчика программно записывается значение, которое одновременно переписывается в рабочий регистр. После запуска таймера рабочий регистр счетчика будет инкрементироваться каждый процессорный такт. Когда рабочий регистр счетчика будет иметь значение 0h, происходит выдача прерывания и в рабочий регистр счетчика переписывается значение теневого регистра. Если таймер работает в режиме однократного запуска, то бит **E** в регистре состояния сбросится в 0 и таймер остановится. Иначе таймер продолжит свою работу. В процессе работы программно можно прочитать рабочий регистр счетчика, а так же остановить таймер, записав 0 в поле **E** регистра состояния. Если после этого снова в поле **E** записать 1, то таймер продолжит отсчет с того значения, на котором произошел останов. Таким образом, чтобы отсчитываемый временной интервал составлял:

1 такт надо записать значение FFFF_FFFFh в теневой регистр счетчика;

2 такта надо записать значение FFFF_FFFEh в теневой регистр счетчика;

.....

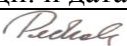
n тактов надо записать значение “-n” в дополнительном коде в теневой регистр счетчика.

Максимальный временной интервал 2^{32} процессорных такта, для этого надо записать значение 0000_0000h в теневой регистр счетчика.

Регистры таймеров являются периферийными и расположены в регистровом окне 0, адреса приведены в таблице ниже (Таблица 4.1).

Таблица 4.1 – Коды для обращения к регистрам таймеров (периферийные регистры для NMC3).

Код регистра	Регистр - источник	Регистр – приемник
	Регистровое окно 0	
110000	TMR_CNT0	TMR_CNT0
110001	TMR_CNT1	TMR_CNT1
110010	TMR_MODE0	TMR_MODE0
110011	TMR_MODE1	TMR_MODE1

					ЮФКВ.431268.006РЭ			Лист
								16
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

4.2 Системный контроллер процессорной системы NMU (NMCSC)

Контроллер управляет запросами на прерывание к процессорной системе **ARMU** и другой процессорной системе **NMU**, контролирует выходы тестового порта и содержит номер процессорной системы **NMU**. Контроллер содержит программно-доступный со стороны ядра **NMC3** периферийный регистр **NMCSCR**, который доступен на чтение и запись по адресу периферийного регистра 110111b (pr7) в регистровом окне 0.

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0										
Зарезервировано																					TD		N	A	A	N	N				
																							M	R	R	M	M				
																							B	M	M	U	U				
																							R	U	U	L	H				
																								L	H						

Рисунок 4.5 – Формат регистра **NMCSCR**.

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

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

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

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

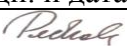
4-й разряд (**NMBR**) определяет номер процессорной системы **NMU**. Данный разряд доступен только для чтения:

0 – для процессорной системы **NMU1**;

1 – для процессорной системы **NMU2**;

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

Поле **TD** (8-5 разряды) является тестовым портом. Значение, записанное в данное поле, будет через два такта после записи выдано на внешние выходы микросхемы: на внешние выходы микросхемы **TD3 – TD0** для процессорной системы **NMU1**; на внешние выходы микросхемы **TD7 – TD4** для процессорной системы **NMU2**.

					ЮФКВ.431268.006РЭ					Лист
										17
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата				
22755-2	 16.01.14		22755-1							

4.3 Контроллер прерываний (INTC)

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

- высокоприоритетное и низкоприоритетное от процессорной системы NMU;
- высокоприоритетное и низкоприоритетное от процессорной системы ARMU;
- три прерывания от блока DSPPU;
- три прерывания от блока ITU;
- одно прерывание от блока DMAC;
- одно прерывание от блока TIMER0;
- одно прерывание от блока TIMER1;
- одно прерывание от блока PU.

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

Запросы на прерывания приходят на входы блока INTC IRQ[15:0] (см. Рисунок 4.6). По положительному фронту запроса устанавливается соответствующий бит в регистре запросов **IRR**. Далее на пути сигнала стоит регистр маски **IMR**. Значение 0 в соответствующем бите разрешает прохождение сигнала, а значение 1 блокирует. Далее схема приоритетов **Priority Unit** выбирает запрос с наибольшим приоритетом, формирует запрос на прерывания для ядра NMC3 и соответствующий ему адрес-вектор прерывания. Приоритет запросов фиксированный: IRQ0 имеет наивысший приоритет, IRQ15 – наименее приоритетный запрос.

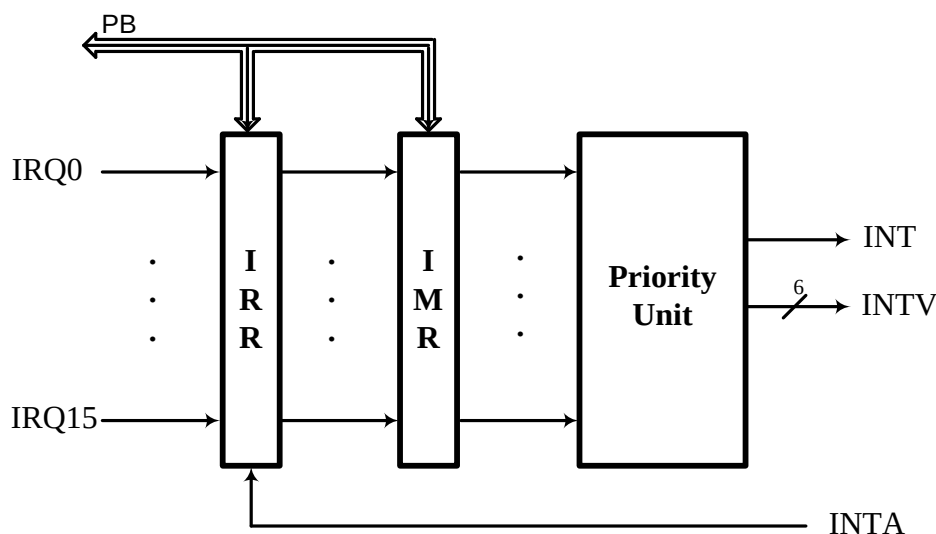


Рисунок 4.6 – Структура контроллера прерываний.

Если сигнал **INTA** имеет активный уровень, то считается, что процессорное ядро **NMC3** зафиксировало прерывание, и соответствующий бит регистра **IRR** сбрасывается. Если сигнал **INTA** имеет неактивный уровень, то запрос на прерывание **INT** будет стоять до прихода сигнала **INTA**. При этом адрес-вектор **INTV** на выходе блока **INTC** не фиксируется, т.е. если на вход **IRQx** поступит более приоритетное прерывание, то адрес-вектор изменится и будет соответствовать более приоритетному прерыванию.

					ЮФКВ.431268.006РЭ				Лист
									18
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	16.01.14		22755-1						

Регистр запросов **IRR** доступен на чтение и побитовую установку/сброс, регистр маски **IMR** доступен на чтение и запись со стороны ядра **NMC3** по периферийной шине. Коды периферийных регистров приведены ниже (Таблица 4.2).

Таблица 4.2 – Коды для обращения к регистрам блока INTC (периферийные регистры для NMC3)

Код регистра	Регистр – источник	Регистр – приемник
	Регистровое окно 0	
110100	IMR	IMR
110101	IRR	IRRreset
110110	IRR	IRRset

Формат регистра **IRR** приведен ниже (см. Рисунок 4.7). Единица в соответствующем разряде означает запрос на прерывание.

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0										
Reserved															IRQ<15:0>																

Рисунок 4.7 – Формат регистра IRR.

Формат регистра **IMR** приведен ниже (см. Рисунок 4.8). Единица в соответствующем разряде означает маскирование запроса (запрос не проходит), 0 – запрос не маскируется (запрос проходит).

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0										
Reserved															IMR<15:0>																

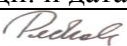
Рисунок 4.8 – Формат регистра IMR.

Примечание. Запись в регистр **IMR**, сброс регистра **IRR** и т.д. осуществляется на последней ступени конвейера, а сброса регистра **INTR** в ядре **NMC3** осуществляется на первой ступени конвейера, поэтому если необходимо сбросить какой-либо бит **INTR** строго после записи/сброса **IMR/IRR** необходимо прочистить конвейер. Это возможно несколькими способами, например подачей 9 команд **vnul** с директивой **.wait** или записью в **PC** нового значения. Пример см. Ниже:

```
.wait;
pr5 = 0ffffh; //clear all request in INT controller
vnul; vnul; vnul; vnul; vnul; vnul; vnul; vnul; vnul;
intr clear 03c0h;
```

или

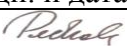
```
.wait;
pr5 = 0ffffh; //clear all request in INTC
pc = INTR_CLEAR;
<INTR_CLEAR>
intr clear 03c0h;
```

					ЮФКВ.431268.006РЭ					Лист
										19
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата				
22755-2	 16.01.14		22755-1							

Список прерываний, которые подаются на входы запросов IRQ, и соответствующий им адрес-вектор, приведен ниже (Таблица 4.3).

Таблица 4.3 – Список внешних прерываний на входах IRQ блока INTC

№ входа	источник	Адрес вектор		Описание
		bin	hex	
IRQ0	INT_ARMH_IN	1000_0000b	80h	Высокоприоритетное прерывание от ARMU
IRQ1	INT_NMUH_IN	1000_1000b	88h	Высокоприоритетное прерывание от другой процессорной системы NMU
IRQ2	INT_TIMER0	1001_0000b	90h	Прерывание от таймера 0
IRQ3	INT_MS_ITMR	1001_1000b	98h	Прерывание от интервального таймера по ms-метке
IRQ4	INT_DSPPU_CH	1010_0000b	A0h	Прерывание от каналов DSPPU
IRQ5	INT_AUX_ITMR	1010_1000b	A8h	Прерывание от интервального таймера по дополнительному временному интервалу
IRQ6	INT_DMAC	1011_0000b	B0h	Прерывания от блока DMAC
IRQ7	INT_TIMER1	1011_1000b	B8h	Прерывание от таймера 1
IRQ8	INT_DSPPU_PRG	1100_0000b	C0h	Прерывание по неправильному программированию блока DSPPU
IRQ9	INT_ARML_IN	1100_1000b	C8h	Низкоприоритетное прерывание от ARMU
IRQ10	INT_NMUL_IN	1101_0000b	D0h	Низкоприоритетное прерывание от другой процессорной системы NMU
IRQ11	INT_PU	1101_1000b	D8h	Прерывание по окончании работы упаковщика
IRQ12	INT_STRB_ITMR	1110_0000b	E0h	Прерывание от интервального таймера по началу выдачи программного строга
IRQ13	INT_DSPPU_HRD	1110_1000b	E8h	Прерывание об аппаратном сбое в блоке DSPPU
IRQ14	Резерв	1111_0000b	F0h	
IRQ15	резерв	1111_1000b	F8h	

					ЮФКВ.431268.006РЭ			Лист
								20
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

4.4 PU (Permutation Unit) – блок упаковки/распаковки векторов данных

Для преобразования потоков данных в формат, удобный для обработки в NMC3 или для хранения в памяти, потоки данных в первом случае распаковываются, а во втором – упаковываются. Этим занимается устройство упаковки/распаковки потоков данных PU (Permutation Unit). Использование PU позволяет резко повысить эффективность NMC3 на таких задачах, как корреляция и фильтрация, при этом хранить коэффициенты для корреляции и фильтрации, а также результаты обработки в компактном виде.

Устройство выполняет распаковку и упаковку потоков данных, при этом элемент входных и выходных данных может иметь размер от 1 до 64 разрядов, результаты могут выдаваться как начиная с младших элементов, так и со старших. Предусмотрено преобразование матрицы в вектор и распаковка вектора с преобразованием в матрицу.

Блок PU может получать данные от ядра NMC3 или самостоятельно формировать адрес для обращения в память за данными или для записи результата.

Настройка блока осуществляется ядром NMC3 путем записи конфигурационных регистров PU.

4.4.1 Аппаратно поддерживаемые операции

Блок PU обрабатывает длинные последовательности двоичных данных. Элементы данных в последовательности имеют одинаковый размер в диапазоне от 1 до 64 бит. Каждая последовательность данных хранится упакованной в 64-х разрядных словах, размещаемых в последовательных ячейках памяти.

Блок PU выполняет обмен с памятью 64-х разрядными машинными словами.

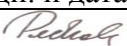
Обрабатываемые данные могут восприниматься как числа в дополнительном коде или простые наборы бит. Последовательности данных могут образовывать вектора и/или матрицы.

Вектор представляет собой несколько последовательных данных.

Матрица – массив (несколько) векторов.

Упаковка входных данных с возможностью выбора любых необходимых разрядов с отбрасыванием остальных. При упаковке настраиваются следующие параметры:

- X – величина, на которую сначала надо сдвинуть весь поток данных ($63 \geq X \geq 0$);
- N – разрядность входных неупакованных данных (Возможные значения N: 2, 4, 8, 16, 32, 64);
- XS – количество разрядов от начала данного (самого младшего разряда) до первого упаковываемого разряда;
- M – разрядность выходных упакованных данных ($N > M$. Возможные значения M: 2, 4, 8, 16, 32, 64);
- Использовать при упаковке насыщение или нет. При этом числа считаются как числа представленные в дополнительном коде. Если насыщение включено, и в старших отбрасываемых разрядах были значащие биты, то результатом операции насыщения будет максимальное положительное (011...1b) число или максимальное отрицательное (100...0b) число. Если же в старших отбрасываемых разрядах не было значащих бит, то насыщения не происходит.

					ЮФКВ.431268.006РЭ			Лист
								21
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Пример упаковки без насыщения, $N=16$, $M=8$, $XS=2$ представлен ниже (см. Рисунок 4.9)

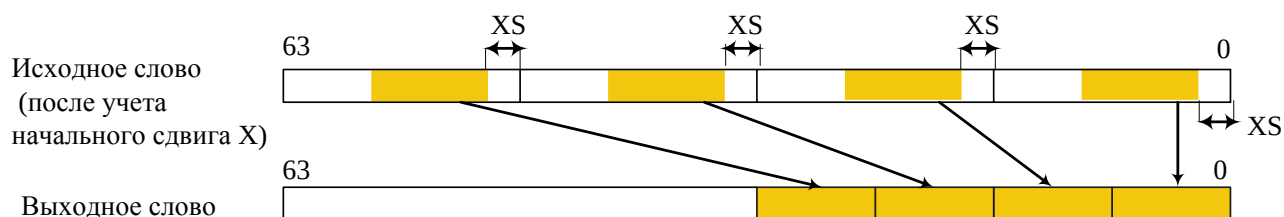


Рисунок 4.9 – Упаковка без насыщения.

Упаковка в слова с одноразрядными данными.

При данной операции входные данные должны быть либо “1”, либо “-1”. При этом единица в выходном слове кодирует значение “-1”, а ноль – “1”. При данной упаковке настраиваются следующие параметры:

- N – разрядность входных неупакованных данных (Возможные значения N : 2, 4, 8, 16, 32, 64),
- X – величина, на которую сначала надо сдвинуть весь поток данных, ($63 \geq X \geq 0$).

Пример упаковки в вектора с одноразрядными данными при $N = 16$ приведен ниже (см. Рисунок 4.10).

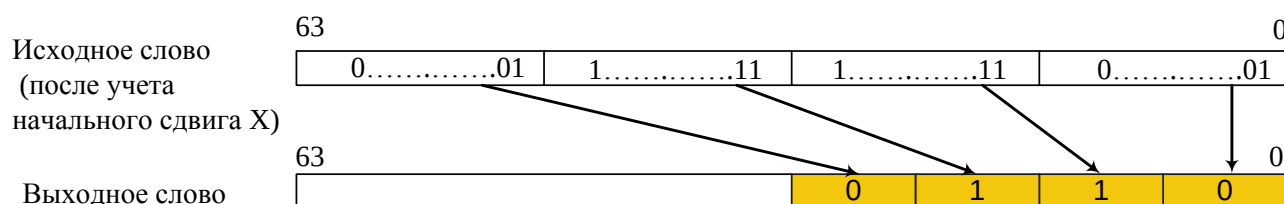


Рисунок 4.10 – Упаковка в вектора с одноразрядными данными

Распаковка. Распаковка осуществляется распространением знакового разряда или нуля в старших разрядах до необходимой разрядности. При распаковке настраиваются следующие параметры:

- X – величина, на которую сначала надо сдвинуть весь поток данных, ($63 \geq X \geq 0$),
- N – разрядность входных упакованных данных (Возможные значения N : 2, 4, 8, 16, 32, 64);
- M – разрядность выходных распакованных данных ($N \leq M$, возможные значения M : 2, 4, 8, 16, 32, 64),
- Y – шаг, на который надо сдвинуть входной поток, для получения очередного слова для дальнейшей обработки. Более подробное объяснение данного параметра показано на рисунке ниже (см. Рисунок 4.11).
- Распространять знаковый разряд или распространять ноль.

					ЮФКВ.431268.006РЭ			Лист
								22
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	Решет 16.01.14		22755-1					

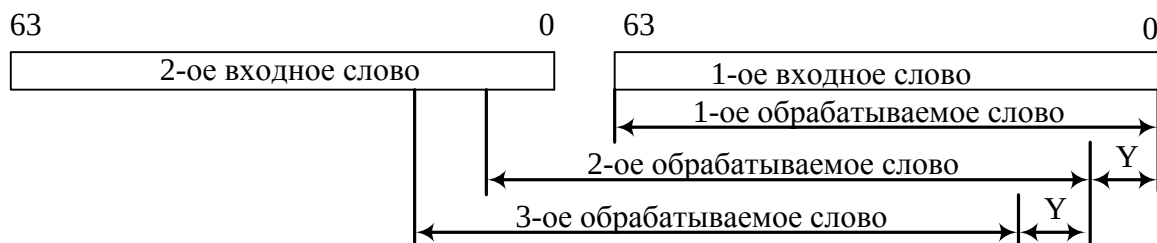


Рисунок 4.11. Получение слов для обработки из входного потока данных

Пример распаковки для $N = 8$, $M = 16$, распространение нулем представлен ниже (см. Рисунок 4.12).

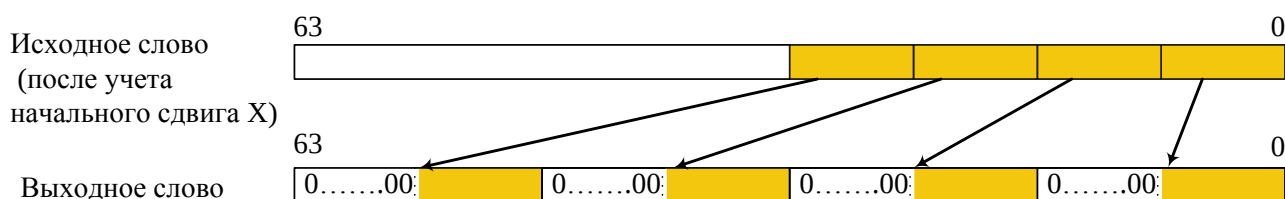
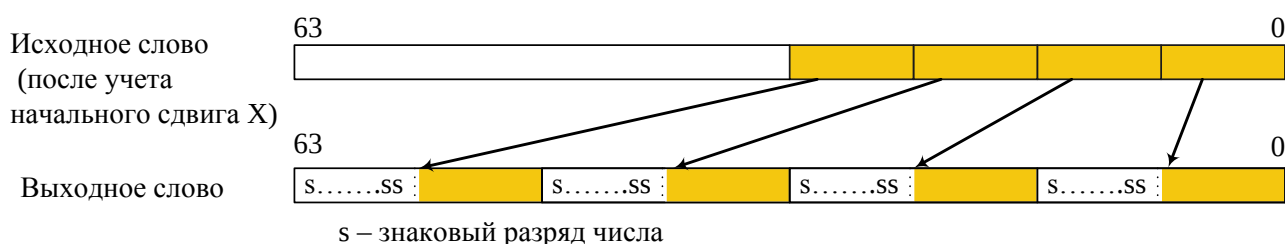


Рисунок 4.12 – Распаковка $N=8$, $M=16$, распространение нулем

Пример распаковки для $N = 8$, $M = 16$, распространение знаком представлен ниже (см. Рисунок 4.13)



s – знаковый разряд числа

Рисунок 4.13 – Распаковка $N=8$, $M=16$, распространение знаком

Распаковка одноразрядных данных. При данной операции единица во входном слове кодирует значение – “-1”, а ноль – “1” При данной распаковке настраиваются следующие параметры:

- M – разрядность выходных неупакованных данных (Возможные значения M : 2, 4, 8, 16, 32, 64),
- X – величина, на которую сначала надо сдвинуть весь поток данных, ($63 \geq X \geq 0$),
- Y – шаг, на который надо сдвинуть входной поток для получения очередного слова для дальнейшей обработки ($64 \geq Y \geq 1$).

Пример распаковки входного слова одноразрядных данных с $M = 16$ представлен ниже (см. Рисунок 4.14).

					ЮФКВ.431268.006РЭ				Лист
									23
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	Решал 16.01.14		22755-1						

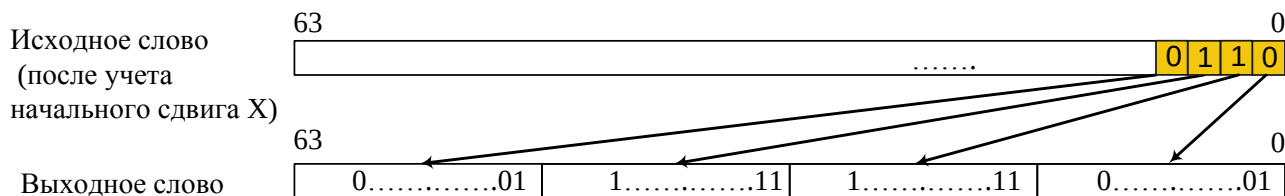


Рисунок 4.14 – Распаковка входного слова одноразрядных данных

Порядок выдачи данных в выходном слове.

Результат может быть выдан, начиная с младших элементов или начиная со старших элементов в выходном слове (см. Рисунок 4.15 и Рисунок 4.16).

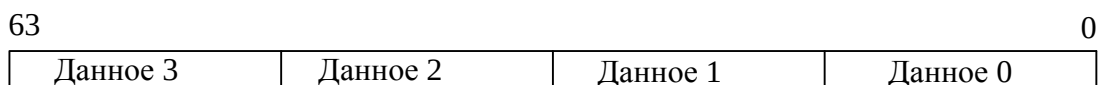


Рисунок 4.15 – Выдача слова, начиная с младших элементов.

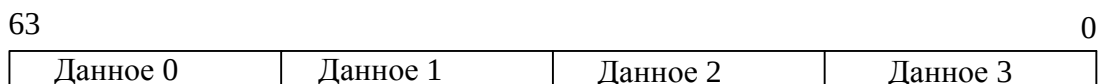


Рисунок 4.16 – Выдача вектора, начиная со старших элементов.

Операции с матрицами.

Получение диагональной матрицы. С помощью РУ можно из входного вектора получить матрицу, элементы матрицы можно расположить как в главной, так и в побочной диагонали. Ниже приведен пример для вектора из 4-х элементов (см. Рисунок 4.17). Данное действие выполняется вместе с режимом распаковки.

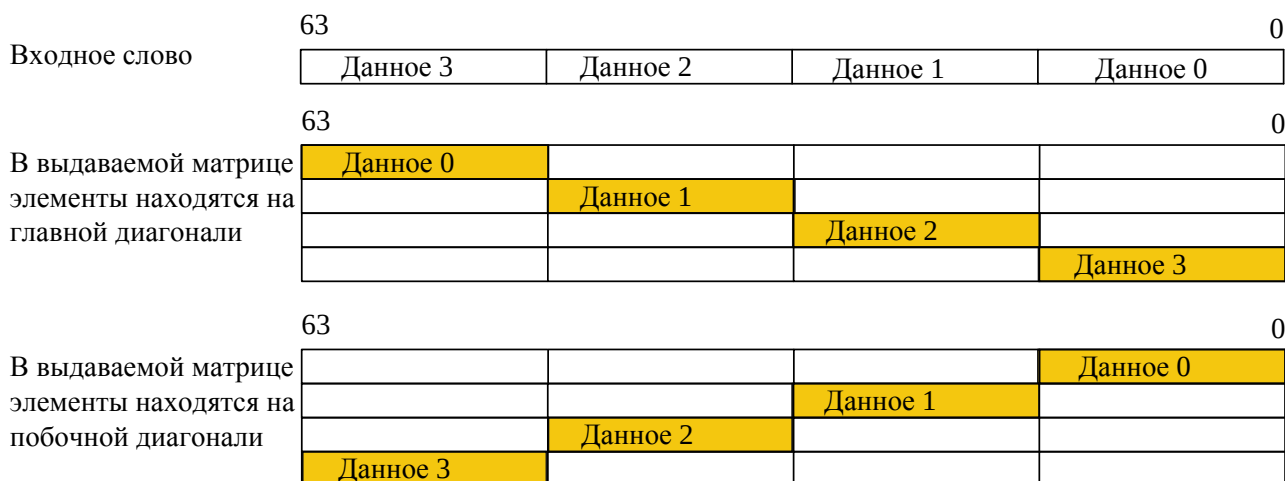


Рисунок 4.17 – Получение диагональной матрицы

Преобразование диагональной матрицы в вектор. Данное действие является обратным к описанному выше действию.

Получение матрицы, в которой элементы находятся в определенном столбце. С помощью PU из слова, состоящего из элементов размером М, можно получить матрицу, в которой элементы исходного слова стоят в определенном столбце. При этом задается номер столбца, в котором будут размещаться элементы (см. Рисунок 4.18). Данное действие выполняется вместе с режимом распаковки.

Входное слово	63					0
		Данное 3	Данное 2	Данное 1	Данное 0	
В выдаваемой матрице элементы в столбце	63					0
				Данное 0		
				Данное 1		
				Данное 2		
				Данное 3		

Рисунок 4.18– Размещение входного слова в первом столбце матрицы

Преобразование матрицы, в которой элементы находятся в определенном столбце, в вектор. Данное действие является обратным к описанному выше действию. Данное действие выполняется при помощи обычной упаковки с N=64.

4.4.2 Режимы и порядок работы PU.

Блок PU может получать данные от ядра NMC3 по шине PU_SB или самостоятельно формировать адрес для обращения в память за данными по шине PU_MB. Для передачи данных от ядра NMC3 к PU в адресном пространстве NMC3 выделен специальный адрес 20001h. При записи данных по этому адресу скалярными или векторными командами данные будут переданы в блок PU для обработки. Запись скалярными командами должна производиться 64-х разрядными словами (регистровыми парами). Результат обработки может быть считан ядром NMC3 напрямую из блока PU по шине PU_SB или PU может самостоятельно формировать адрес для записи результата в память по шине PU_MB. Для чтения данных с выхода PU ядро NMC3 использует специально выделенный адрес 20000h. При чтении данных по этому адресу скалярными или векторными командами данные будут считаны из блока PU. Чтение скалярными командами должно производиться 64-х разрядными словами (регистровыми парами). Возможны следующие режимы работы блока PU:

- Блок PU получает данные от ядра NMC3 по шине PU_SB[63:0] и записывает результат обработки в память по шине PU_MB.
- Блок PU забирает данные из памяти по шине PU_MB[63:0] и передает данные по запросу от NMC3 по шине PU_SB[63:0].
- Блок PU получает данные по шине PU_SB[63:0] и передает данные по запросу от NMC3 по шине PU_SB[63:0].
- Блок PU не работает.

Если PU настраивается таким образом, что самостоятельно читает или записывает данные (по шине PU_MB), то необходимо выделить область памяти, в которую он будет обращаться. Границы этой области сообщаются блоку PU через регистры PUR7 и PUR6; начальный (стартовый) адрес внутри области – через регистр PUR5. PU обращается последовательно по адресам в выделенной области и не может выйти за её гра-

					ЮФКВ.431268.006РЭ				Лист
									25
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	Решал 16.01.14		22755-1						

ницы – каждый раз, когда достигается старший адрес области, следующее обращение следует по младшему адресу области.

Предусмотрен следующий порядок работы с блоком PU:

- Настройка конфигурации путем записи в конфигурационные регистры PU процессорным ядром NMC3.
- Запуск блока PU.
- Передача и/или получение данных с помощью команд записи и/или чтения данных ядра NMC3. При этом количество читаемых и записываемых данных должно строго соответствовать заданным настройкам регистров CNT_IN и CNT_OUT.
- Обработка прерывания от устройства. После того, как блок PU заканчивает работу, происходит выдача прерывания. Это говорит о том что блок готов к настройке на обработку следующего пакета данных.

После системного сброса блок PU переходит в состояние “Блок PU не работает” и находится в этом состоянии до момента программного изменения поля, отвечающего за режим работы. После записи в это поле значения, отличного от 0, блок PU начинает работу в соответствии с тем режимом, который был записан, и прочими настройками, заданными в конфигурационных регистрах.

4.4.3 Конфигурационные регистры

Таблица 4.4 – Коды регистров блока PU (периферийные регистры для NMC3) .

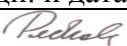
Код регистра	Регистр – приемник
	Регистровое окно 1
110000	PUR0
110001	PUR1
111000	PUR2
111001	PUR3
111010	PUR4
111011	PUR5
111100	PUR6
111101	PUR7
111110	PUR8
111111	PUR9

Регистры блока **PU** доступны *только для записи* как периферийные регистры ядра **NMC3**. Коды регистров блока **PU** приведены выше (Таблица 4.4). Конфигурационные регистры можно разделить на три группы.

Первая группа регистров содержит код операции. Группа состоит из регистров PUR8, PUR9.

Вторая группа регистров PUR7, PUR6, PUR5, PUR4, PUR3 используется для организации счетчиков. Регистры содержат параметры для обращения в память – поля TOP, BOTTOM, FIRST, количество слов, которое нужно принять из процессора или из памяти – поле CNT_IN, количество слов, которое нужно выдать в память или в процессор – поле CNT_OUT.

Третья группа регистров PUR2, PUR1, PUR0 предназначена для экономии аппаратуры и упрощения реализации функциональных блоков. Регистры содержат используемые для разных целей маски. Если неправильно задать значения этих регистров, могут быть получены ошибочные данные.

					ЮФКВ.431268.006РЭ				Лист
									26
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

Регистр PUR9 должен быть записан последним. Порядок записи остальных регистров не важен.

После сброса поле регистра PUR9, отвечающее за режим работы (MODE), принимает значение 0 (блок PU не работает), остальные поля и регистры после сброса не определены.

Структура конфигурационных регистров PU дана ниже.

Регистр PUR9. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR9 является периферийным в регистровом окне 1 и доступен по адресу 111111b (pr15).

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
зарезервировано	Xs					SHIFT					COLNUM					C	D	M	M	M	B	S	S	P	M					
																O	I	A	A	A	S	I	A	E	A	O				
																L	A	T	T	T	T	T	T	C	D					
																	G	O	I	N				K	E					

Рисунок 4.19 – Формат регистра PUR9.

Поле **MODE** (разряды с 1 по 0) определяет режим работы PU:

00 – PU не работает

01 – Входные данные поступают по шине PU_SB от процессорного ядра NMC3, выходные данные записываются в память по шине PU_MB. Адрес для записи результата вырабатывается блоком PU (PU_SB → PU_MB).

10 – Входные данные поступают по шине PU_MB. Адрес для чтения входных данных вырабатывается блоком PU, выходные данные выдаются на шину PU_SB, которые забирает процессорное ядро NMC3 (PU_MB → PU_SB).

11 – Входные данные поступают по шине PU_SB от процессорного ядра NMC3, выходные данные выдаются на шину PU_SB, которые забирает процессорное ядро NMC3 (PU_SB → PU_SB).

После того, как в поле MODE записано ненулевое значение, и до окончания работы блока (до прерывания от блока) все регистры устройства недоступны для записи.

2-й разряд (**PACK**) определяет режим работы PU распаковки или упаковки:

0 – распаковка,

1 – упаковка.

3-й разряд (**SE**) определяет режим расширения при распаковке:

0 – расширить нулем,

1 – расширить знаком.

4-й разряд (**SAT**) определяет режим насыщения при упаковке:

0 – насыщение выключено,

1 – насыщение включено.

5-й разряд (**BIT**) определяет тип распаковки/упаковки:

0 – обычная распаковка/упаковка,

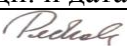
1 – распаковка/упаковка битовых данных.

6-й разряд (**MSB**) определяет положение элементов в выходном векторе:

0 – выдача вектора, начиная с младших элементов

1 – выдача вектора, начиная со старших элементов

7-й разряд (**MATIN**) определяет тип данных на входе PU:

					ЮФКВ.431268.006РЭ					Лист
										27
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата			
22755-2	 16.01.14		22755-1							

- 0 – на входе PU вектор данных,
- 1 – на входе PU матрица данных.

8-й разряд (**MATOUT**) определяет тип данных на выходе PU для распаковки. Для упаковки данный бит должен быть нулем.

- 0 – на выходе PU вектор данных,
- 1 – на выходе PU матрица данных.

9-й разряд (**DIAG**). Данное поле определяет тип матрицы на входе или выходе. Имеет значение:

- 0 – получение побочной диагонали,
- 1 – получение главной диагонали.

10-й разряд (**COL**). Выдача столбца матрицы:

- 0 – нет преобразования данных в столбец матрицы,
- 1 – есть преобразование данных в столбец матрицы.

Поле **COLNUM** (16-11 разряды) – номер столбца матрицы, в который надо записать элементы; если в столбце больше одного разряда, то это номер младшего разряда столбца.

Поле **SHIFT** (22 – 17 разряды) определяет начальный сдвиг. При распаковке это величина X, при упаковке это сумма X+Xs

Поле **Xs** (28 – 23 разряды) – величина Xs при упаковке, при распаковке следует записать 0. Значение поля Xs не может быть больше значения поля SHIFT.

Возможные режимы работы устройства и соответствующие им настройки приведены ниже (Таблица 4.5). Все другие сочетания значений являются запрещенными.

Таблица 4.5 – Список операций блока PU.

Тип операции \ управляющее поле	PACK	BIT	MATIN	MATOUT	DIAG	COL
Распаковка/переупаковка	0	0/1	0	0	x*	x
Распаковка с получением элементов в главной диагонали матрицы	0	0/1	0	1	1	0
Распаковка с получением элементов в побочной диагонали матрицы	0	0/1	0	1	0	0
Распаковка с получением элементов в столбце матрицы	0	0/1	0	1	x	1
Преобразование матрицы с элементами в главной диагонали в вектор	0	0	1	0	1	0
Преобразование матрицы с элементами в побочной диагонали в вектор	0	0	1	0	0	0
Упаковка	1	0/1	0	0	x	x

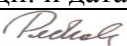
* – значение поля не оказывает влияния на результат.

Регистр PUR8. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR8 является периферийным в регистровом окне 1 и доступен по адресу 111110b (pr14)

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
NPE					NPB					Y					MCODE					NCODE										

Рисунок 4.20 – Формат регистра PUR8.

Поле **NCODE** (5 – 0 разряды). Задаёт разрядность входных данных, возможные значения поля NCODE:

					ЮФКВ.431268.006РЭ					Лист
										28
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.		Подп. и дата			Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14			22755-1					

- 0 – разрядность входных данных 1 бит;
- 1 – разрядность входных данных 2 бит;
- 2 – разрядность входных данных 4 бит;
- 4 – разрядность входных данных 8 бит;
- 8 – разрядность входных данных 16 бит;
- 16 – разрядность входных данных 32 бит;
- 32 – разрядность входных данных 64 бит;

При подаче на вход одноразрядных элементов $NCODE = 0$ и $BIT = 1$.

Поле **MCODE** (11 – 6 разряды). Задаёт разрядность выходных данных, возможные значения поля **MCODE**:

- 0 – разрядность выходных данных 1 бит;
- 1 – разрядность выходных данных 2 бит;
- 2 – разрядность выходных данных 4 бит;
- 4 – разрядность выходных данных 8 бит;
- 8 – разрядность выходных данных 16 бит;
- 16 – разрядность выходных данных 32 бит;
- 32 – разрядность выходных данных 64 бит;

При выдаче одноразрядных элементов $M = 0$ и $BIT = 1$.

Поле **Y** (18 – 12 разряды). Шаг, на который надо сдвинуть входной поток для получения очередного слова для дальнейшей обработки, при распаковке $Y = m \cdot N$ где $m = 1, 2, \dots, 64/N$, в остальных случаях $Y = 64$. Значение поля **Y** не может быть равно 0.

Поле **NPB** (24 – 19 разряды). Значение, которое вычисляется следующим образом:

- при упаковке – $64 \cdot M/N$,
- при распаковке – $64 \cdot N/M$.

Поле **NPE** (31-25 разряды) Значение, которое вычисляется:

- при упаковке – N/M
- при распаковке – 0.

Регистр PUR7. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR7 является периферийным в регистровом окне 1 и доступен по адресу 111101b (pr13)

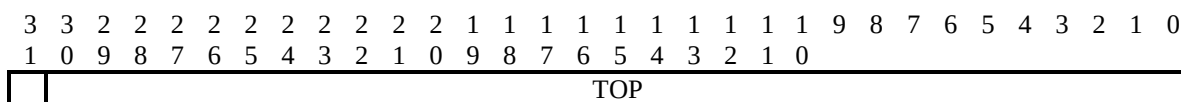
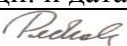


Рисунок 4.21 – Формат регистра PUR7.

Поле **TOP** (30-0 разряды) содержит сдвинутое на один бит в сторону младшего разряда значение максимального адреса при обращении к памяти по шине PU_MB (т.е. разряды адреса с 31-го по 1-й). Нулевой разряд адреса считается равным 0.

Регистр PUR6. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR6 является периферийным в регистровом окне 1 и доступен по адресу 111100b (pr12)

					ЮФКВ.431268.006РЭ					Лист
										29
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата				
22755-2	 16.01.14		22755-1							

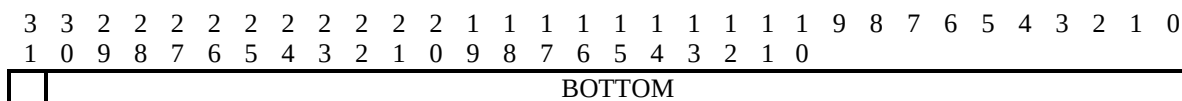


Рисунок 4.22 – Формат регистра PUR6.

Поле **BOTTOM** (30-0 разряды) содержит сдвинутое на один бит в сторону младшего разряда значение минимального адреса при обращении к памяти по шине PU_MB (т.е. разряды адреса с 31-го по 1-й). Нулевой разряд адреса считается равным 0.

Регистр PUR5. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR5 является периферийным в регистровом окне 1 и доступен по адресу 111011b (pr11)

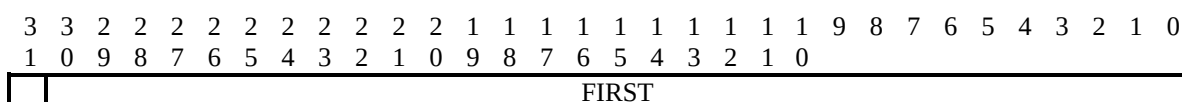


Рисунок 4.23 – Формат регистра PUR5.

Поле **FIRST** (30-0 разряды) содержит сдвинутое на один бит в сторону младшего разряда значение первого адреса при обращении к памяти по шине PU_MB (т.е. разряды адреса с 31-го по 1-й). Нулевой разряд адреса считается равным 0. Ограничение: $BOTTOM \leq FIRST \leq TOP$. Адреса генерируются, начиная с FIRST, каждый следующий адрес получается путем инкрементирования, в случае достижения TOP следующий адрес – BOTTOM.

Регистр PUR4. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR4 является периферийным в регистровом окне 1 и доступен по адресу 111010b (pr10)

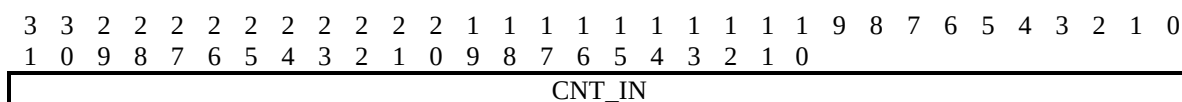


Рисунок 4.24 – Формат регистра PUR4.

Поле **CNT_IN** (31-0 разряды) содержит количество входных 64-х разрядных слов, которое должно выдать ядро **NMC3** на шину PU_SB (для режима MODE = 01, 11) или которое блок **PU** должен считать из памяти по шине PU_MB (для режима MODE = 10).

Регистр PUR3. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр PUR3 является периферийным в регистровом окне 1 и доступен по адресу 111001b (pr9)

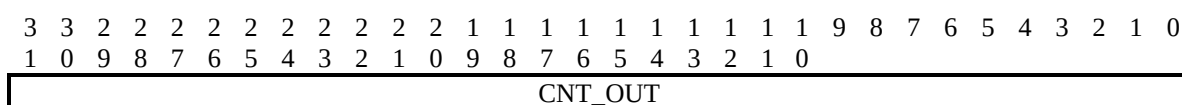


Рисунок 4.25 – Формат регистра PUR3.

					ЮФКВ.431268.006РЭ					Лист
										30
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата				
22755-2	Редько 16.01.14		22755-1							

Поле **CNT_OUT** (31-0 разряды) содержит количество выходных слов, которое должен выдать блок PU на шину PU_SB для ядра NMC3 (для режима MODE = 10, 11) или на шину PU_MB для записи в память (для режима MODE = 01). После выдачи количества слов, определяемого полем **CNT_OUT**, блок **PU** заканчивает обработку пакета и выдает прерывание ядру **NMC3**. После этого можно производить настройку и запуск для обработки следующего пакета данных.

Между полями **CNT_IN** и **CNT_OUT** есть соотношение при упаковке:

если поле **SHIFT** регистра **PUR9** = 0, то $CNT_IN = CNT_OUT * N/M$

если поле **SHIFT** регистра **PUR9** > 0, то $CNT_IN = CNT_OUT * N/M + 1$

Если соотношения не выполнены, то возможна некорректная работа блока **PU**. Например, всегда присылать лишнее слово нельзя, так как при достижении числа уже выданных устройством слов величины **CNT_OUT** вырабатывается сигнал прерывания, блок **PU** останавливается, принимает новые значения в конфигурационные регистры, а дополнительное слово данных для предыдущей команды воспримется как первое слово данных для следующей команды.

Регистр PUR2. Регистр доступен на запись со стороны ядра NMC3. Для NMC3 регистр **PUR2** является периферийным в регистровом окне 1 и доступен по адресу 111000b (pr8)

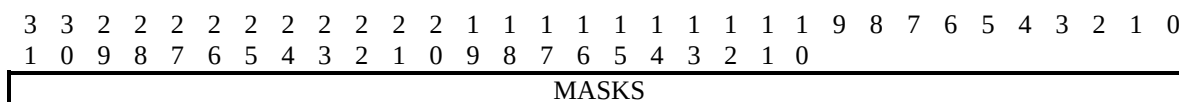


Рисунок 4.26 – Формат регистра PUR2.

Поле **MASKS** (31-0 разряды) содержит маску, которая при упаковке и распаковке имеет разное назначение. При упаковке в поле **MASKS** единица должна быть в самом старшем разряде каждого числа, в остальных разрядах должны быть нули (см. Рисунок 4.27)

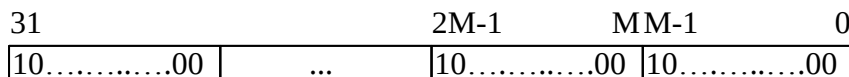


Рисунок 4.27. Формат маски MASKS при упаковке.

При распаковке в поле **MASKS** единицы должны стоять в N младших разрядах каждого распаковываемого числа, нули в остальных разрядах (см. Рисунок 4.28).

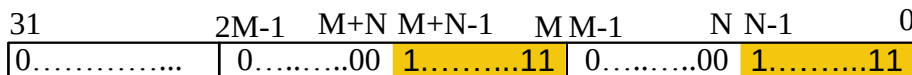
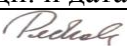


Рисунок 4.28. Формат маски MASKS при распаковке.

Регистры PUR1 и PUR0. Регистры доступны на запись для ядра NMC3. Для NMC3 регистры **PUR1** и **PUR0** являются периферийными в регистровом окне 1 и доступны по адресам 110001b (pr1), 110000b (pr0) соответственно.

					ЮФКВ.431268.006РЭ				Лист
									31
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

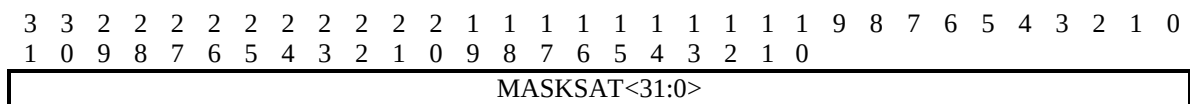


Рисунок 4.29 – Формат регистра PUR1.

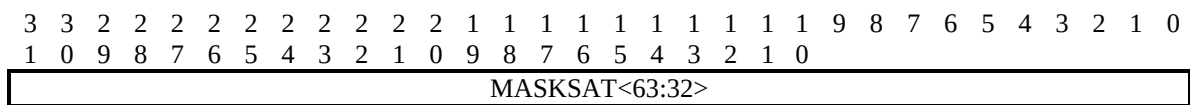


Рисунок 4.30 – Формат регистра PUR0.

Регистр **PUR1** содержит младшую часть, а регистр **PUR0** старшую часть 64-х разрядного поля **MASKSAT<63:0>**. Данное поле используется при упаковке и при получении матрицы с элементами в столбце или диагонали. При всех других операциях данное поле не используется и может быть не инициализировано.

При упаковке в поле **MASKSAT<63:0>** содержится 64/N последовательно расположенных одинаковых частей. Каждая часть содержит N бит. М-1 младших разрядов этой части заполняются нулями, следующие N-M-Xs бит заполняются единицами и оставшиеся Xs+1 разрядов заполняются нулями (см. Рисунок 4.31).

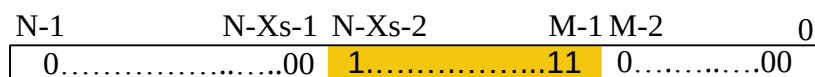


Рисунок 4.31. Формат общей части для получения маски MASKSAT при упаковке.

Оставшиеся разряды маски получаются путем дублирования этой общей части. Ниже приведен пример полученной маски с общей частью 000...111...000 (см. Рисунок 4.32).

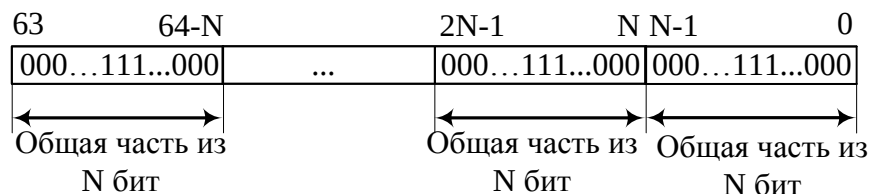


Рисунок 4.32. Пример полученной маски MASKSAT при упаковке.

При получении матрицы с элементами в столбце поле **MASKSAT<63:0>** содержит единицы в тех разрядах, где должен быть столбец матрицы с данными (см. Рисунок 4.33).

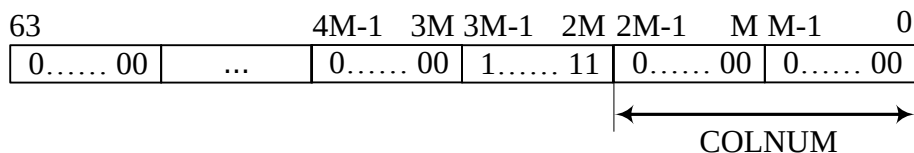


Рисунок 4.33. Пример маски MASKSAT при получении матрицы с элементами в столбце.

При получении матрицы с элементами в главной или побочной диагонали поле MASKSAT<63:0> содержит единицы в М младших разрядах (см. Рисунок 4.34).

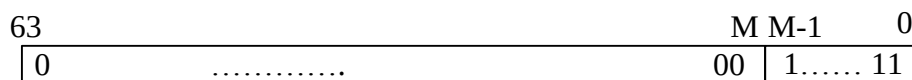


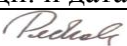
Рисунок 4.34. Пример маски MASKSAT при получении матрицы с элементами в главной или побочной диагонали.

4.5 Список периферийных регистров для ядра NMC3

Список всех периферийных регистров для ядра NMC3 приведен в таблице ниже (Таблица 4.6). Все другие сочетания значений являются запрещенными.

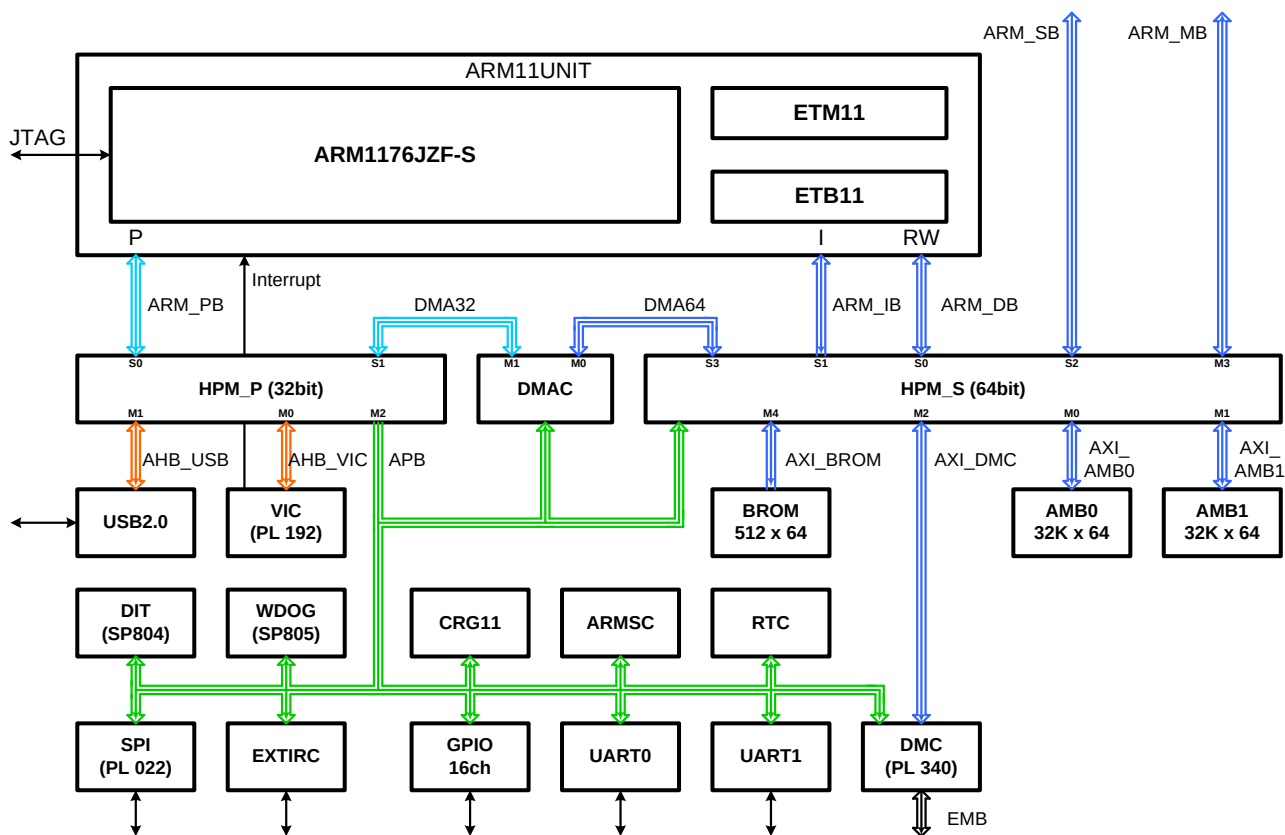
Таблица 4.6 – Список периферийных регистров для ядра NMC3

Код регистра	Регистровое окно 0		Регистровое окно 1	
	Регистр – источник	Регистр – приемник	Регистр – источник	Регистр – приемник
110000	TMR_CNT0	TMR_CNT0		PUR0
110001	TMR_CNT1	TMR_CNT1		PUR1
110010	TMR_MODE0	TMR_MODE0		
110011	TMR_MODE1	TMR_MODE1		
110100	IMR	IMR		
110101	IRR	IRRreset		
110110	IRR	IRRset		
110111	NMCSCR	NMCSCR		
111000				PUR2
111001				PUR3
111010				PUR4
111011				PUR5
111100				PUR6
111101				PUR7
111110				PUR8
111111				PUR9

					ЮФКВ.431268.006РЭ			Лист
								33
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата
22755-2		 16.01.14		22755-1				

5 Процессорная система на базе ядра ARM11 (ARMU).

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



ARM11UNIT – ARM1176 Unit
 ARM1176JZF-S – ARM Core
 ETM11 – ARM11 Embedded Trace Macrocell
 ETB11 – ARM11 Embedded Trace Buffer
 DMAC – Direct Memory Controller
 USB2.0 – Universal Serial Bus 2.0 Device
 VIC – Vectored Interrupt Controller
 BROM – Boot ROM
 AMB0,1 – AMBA Memory Bank 0,1 RAM
 DIT – Dual Timers
 WDOG – Watchdog Timer
 CRG11 – Clock and Reset Generator
 ARMSC – ARM System Controller
 RTC – Real Time Clock
 SPI – Serial Peripheral Interface
 EXTIRC – External Interrupt Controller
 GPIO – General Purpose Input/Output
 UART0,1 – Universal Asynchronous Receiver/Transmitter
 DMC – Dynamic DDR Memory Controller

HPM_S – System AXI 64b High Performance Interconnect Matrix
 HPM_P – Peripheral AXI 32b High Performance Interconnect Matrix
 ARM_IB – ARM11 AXI 64b Instruction Bus
 ARM_DB – ARM11 AXI 64b Data Bus
 ARM_PB – ARM11 AXI 32b Peripheral Bus
 ARM_MB – ARM-NMC AXI 64b Master Bus
 ARM_SB – ARM-NMC AXI 64b Slave Bus
 DMA64 – DMA Controller AXI 64b Master Bus
 DMA32 – DMA Controller AXI 32b Master Bus
 AXI_BROM – Boot ROM AXI 64b Bus
 AXI_DMC – Dynamic Memory Controller AXI 64b Bus
 AXI_AMB0 – AMBA Memory Bank 0 AXI 64b Bus
 AXI_AMB1 – AMBA Memory Bank 1 AXI 64b Bus
 EMB – External Memory DDR 32b Bus
 AHB_USB – AMBA High Speed (AHB) USB Bus
 AHB_VIC – AMBA High Speed (AHB) VIC Bus
 APB – AMBA Peripheral Bus

Рисунок 5.1 – Структурная схема ARMU части системы

Процессорная система на базе ядра ARM11 работает на частоте 164 МГц и содержит следующие функциональные узлы:

					ЮФКВ.431268.006РЭ			Лист
								34
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

ARM11UNIT – СФ блок, включающий в себя микропроцессорное ядро ARM и дополнительные блоки отвечающие за отладку программ (ETM11, ETB11 и JTAG).

ARM1176JZF-S – микропроцессорное ядро фирмы ARM Ltd. Данное ядро работает на частоте 164 МГц. В его состав входят кэш команд и данных объемом 16 Кбайт каждый, а также блок с плавающей точкой. Обмен ядра с внешним миром осуществляется с помощью двух 64-разрядных шин: команд (ARM_IB) и данных (ARM_DB), а также 32-разрядной периферийной шины (ARM_PB).

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

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

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

HPM_x (AXI High Performance Matrix) – коммутаторы шин, работающих согласно стандарту AMBA AXI фирмы ARM Ltd. ARMU содержит два таких коммутатора: системный (HPM_S) и периферийный (HPM_P). Системный коммутатор работает с 64-разрядными шинами на частоте 164 МГц, он осуществляет коммутацию 4-х мастеров и 5-ти управляемых устройств и является четырехуровневым, т.е. обеспечивает одновременную передачу данных всеми мастерами при условии работы с разными управляемыми устройствами. Также системный коммутатор позволяет программно настраивать приоритеты мастеров при обращении к одному управляемому устройству. Периферийный коммутатор работает с 32-разрядными шинами на частоте 82 МГц. Он осуществляет коммутацию 2-х мастеров, работающих по стандарту AXI, и 3-х управляемых устройств (два АНВ устройства и одно АРВ устройство) и является двухуровневым. Приоритеты мастеров являются фиксированными, используется алгоритм арбитража Round-robin.

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

VIC (Vectored Interrupt Controller) – контроллер прерываний, который поддерживает до 32-х прерываний с программно настраиваемыми адрес-векторами и приоритетами.

CRG11 (Clock and Reset Generator) – блок, управляющий генерацией синхросигналов и сигналов сброса. Позволяет программным образом менять рабочую частоту некоторых блоков, а также производить программный сброс микросхемы.

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

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

- Регистр управления отображением памяти – управляет отражением ПЗУ или ОЗУ по начальному адресу памяти
- Регистры состояния входов – показывают состояние входов MC, управляющих источником начальной загрузки, а также состоянием тестовых входов

					ЮФКВ.431268.006РЭ			Лист
								35
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редько</i> 16.01.14		22755-1				

- Регистры управления DDR – набор управляющих регистров и регистров состояния, управляющих начальной инициализацией DDR DMC интерфейса
- Регистры управления АЦП – позволяют управлять режимом работы встроенного АЦП
- Регистры генерации прерываний к NMU части – управляют генерацией прерываний к NMU части
- Регистр приоритета ядер NMU
- Регистры управления USB блока - управляют начальной инициализацией блока USB

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

WDOG (Watchdog Unit) – сторожевой таймер.

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

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

BROM (Boot ROM) – банк памяти ПЗУ объёмом 4 Кб, который содержит программу начального загрузчика СБИС СПОВС.

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

USB2.0 (Universal Serial Bus 2.0 Device) – контроллер интерфейса шины USB, соответствующей спецификации 2.0 (Full-speed) и работающий как USB device.

DMC (Dynamic Memory Controller) – контроллер динамической памяти, обеспечивающий обмен с внешней памятью типа DDR SDRAM по 32-разрядной шине данных с частотами порядка 160 МГц.

SPI (Serial Peripheral Interface) – контроллер синхронного последовательного интерфейса, обеспечивающий обмен по интерфейсу Motorola SPI. Поддерживаются частоты работы интерфейса до 40 МГц.

UARTx (Universal Asynch. Receiver-Transmitter) – 2 контроллера асинхронного последовательного интерфейса, совместимых с популярной микросхемой UART 16550. Поддерживаются скорости обмена до 460800 бод из стандартного ряда скоростей. Также поддерживаются более высокие скорости (до 5Мбод), но они не попадают в стандартный ряд скоростей, т.е. могут применяться только при соединении 2-х микросхем СПОВС между собой, либо при подключении к другой МС с UART интерфейсом, поддерживающим нестандартные скорости обмена.

GPIO (General Purpose Input/Output) – блок программируемых вводов/выводов общего назначения. Блок GPIO содержит 16 каналов, объединённых в 2 группы по 8 разрядов.

EXTIRC (External Interrupt Controller) – контроллер внешних прерываний. Имеет 4 канала и позволяет программировать тип внешнего прерывания: по фронту, по срезу, по низкому уровню или по высокому уровню.

Все функциональные узлы ARMU соединяются шинами четырех типов:

					ЮФКВ.431268.006РЭ			Лист
								36
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редько</i> 16.01.14		22755-1				

- 64-разрядные AXI (AMBA eXtensible Interconnect) шины, позволяющие производить обмен данными на частоте 164 МГц (изображены синим цветом);
- 32-разрядные AXI (AMBA eXtensible Interconnect) шины, позволяющие производить обмен данными на частоте 82 МГц (изображены бледно-голубым цветом);
- 32-разрядные АНВ (AMBA High Speed Bus) шины, позволяющие производить обмен данными на частоте 82 МГц (изображены оранжевым цветом);
- 32-разрядные APB (AMBA Peripheral Bus) шины, позволяющие производить обмен данными на частоте 82 МГц (изображены зеленым цветом);

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

ARM_IB – 64-разрядная AXI шина загрузки команд микропроцессорного ядра из памяти. Инициатор обмена – ARM1176UNIT. Частота работы шины – 164 МГц.

ARM_DB – 64-разрядная AXI шина обмена данными между микропроцессорным ядром и памятью. Инициатор обмена – ARM1176UNIT. Частота работы шины – 164 МГц.

ARM_PB – 32-разрядная AXI шина обмена данными между микропроцессорным ядром и периферийными устройствами. Инициатор обмена – ARM1176UNIT. Частота работы шины – 82 МГц.

ARM_MB – 64-разрядная AXI шина обмена данными между ARMU и SMU. Инициатор обмена – ARMU. Частота работы шины – 164 МГц.

ARM_SB – 64-разрядная AXI шина обмена данными между SMU и ARMU. Инициатор обмена – SMU. Частота работы шины – 164 МГц.

DMA64 – 64-разрядная AXI шина обмена данными между контроллером ПДП (DMAC) и памятью. Инициатор обмена – DMAC. Частота работы шины – 164 МГц.

DMA32 – 32-разрядная AXI шина обмена данными между контроллером ПДП (DMAC) и периферийными устройствами (UART0, UART1, SPI и USB). Инициатор обмена – DMAC. Частота работы шины – 82 МГц.

AXI_BROM, AXI_AMB0, AXI_AMB1 – 64-разрядные AXI шины обмена данными между соответствующими банками внутреннего ОЗУ и ПЗУ и мастерами подключенными к системному коммутатору HPM_S (ARM1176UNIT, SMU и DMAC). Частота работы шины – 164 МГц.

AXI_DMC – 64-разрядные AXI шины обмена данными между контроллером внешней памяти (DMC) и мастерами подключенными к системному коммутатору HPM_S (ARM1176UNIT, SMU и DMAC). Частота работы шины – 164 МГц.

АНВ_USB – 32-разрядная шина обмена данными между ARM1176UNIT и контроллером USB. Инициатор обмена – ARM1176UNIT. Частота работы шины – 82 МГц.

АНВ_VIC – 32-разрядная шина обмена данными между ARM1176UNIT и контроллером прерываний (VIC). Инициатор обмена – ARM1176UNIT. Частота работы шины – 82 МГц.

APB – 32-разрядная шина обмена данными между ARMU и периферийными устройствами (SPI, UART0 и UART1, GPIO, RTC, DMC). Инициатор обмена – ARMU. Частота работы шины – 82 МГц.

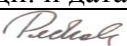
					ЮФКВ.431268.006РЭ			Лист
								37
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редкофф</i> 16.01.14		22755-1					

5.1 Микропроцессорное ядро ARM1176JZF

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

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

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

					ЮФКВ.431268.006РЭ			Лист
								38
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

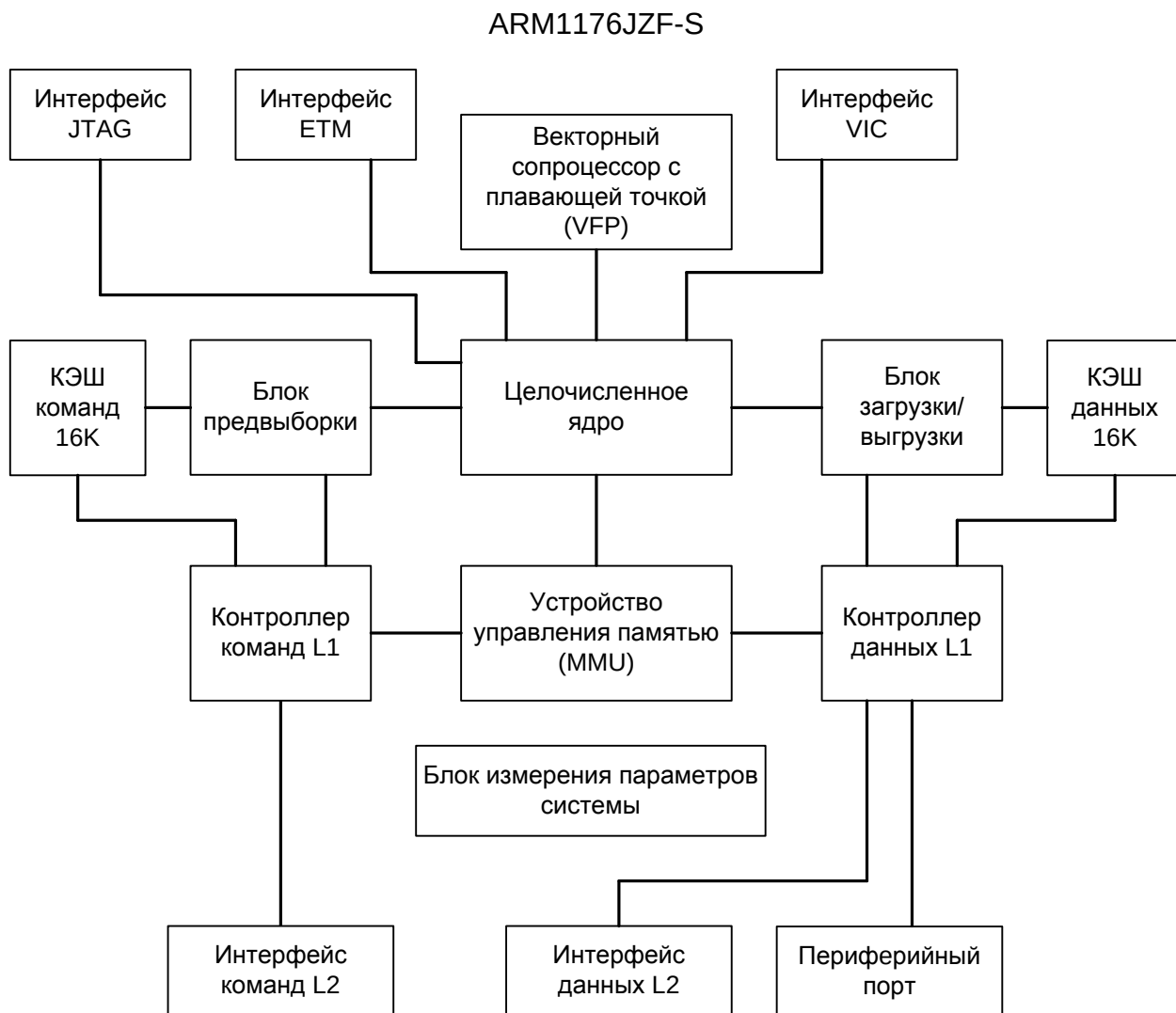


Рисунок 5.2 – Структурная схема процессорного ядра ARM1176JZF-S

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

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

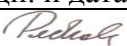
					ЮФКВ.431268.006РЭ			Лист
								39
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

ограниченного набора опций при помощи механизма переразметки TEX. Это подразумевает, что записи в таблице страниц имеют косвенный характер.

Таблица кодирования TEX CB содержит два бита, связанные с таблицей страниц, состояние которых управляется ОС. Для совместимости на уровне кода с существующими версиями ОС для ARMv6 такая структура обеспечивает работу с различными режимами функционирования MMU. Данный подход называется TEX Remap и управляется битом [28] в регистре 1 CP15.

- Изменен порядок использования бита AP. В процессоре ARM1176JZF-S код b111 в APX и AP[1:0] означает доступ «только для чтения» в привилегированном и пользовательском режимах. AP[0] указывает тип отмены доступа, ошибку бита доступа, когда CP15 c1[29] равно 1.

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

					ЮФКВ.431268.006РЭ			Лист
								40
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.2 Блоки поддержки отладки — ETM11, ETB11 и JTAG интерфейс

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

5.2.1 Устройство блоков аппаратной поддержки отладки

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

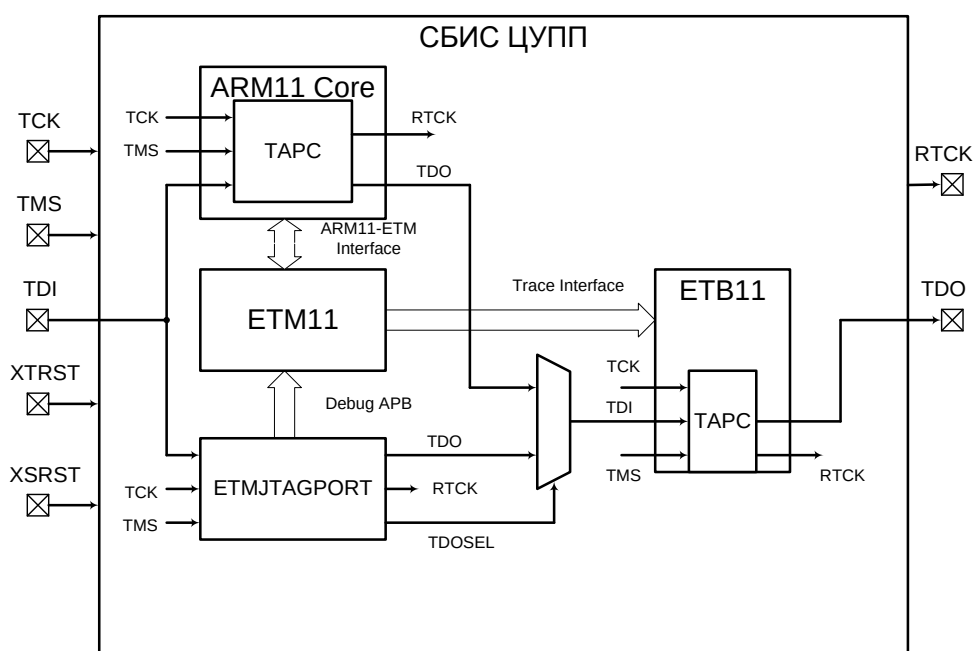


Рисунок 5.3 — Схема внутрикристалльных аппаратных средств отладки программ на ядре ARM

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

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

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

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

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

					ЮФКВ.431268.006РЭ			Лист
								41
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

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

ETB11 (Embedded Trace Buffer) — буфер отладочной информации. Поскольку интерфейс JTAG относительно медленный, то в ходе выполнения программ отладочная информация должна скапливаться в буфере. Блок ETB11 служит для временного хранения данных, читаемых через JTAG. Используемая в СБИС СПОВС конфигурация ETB11 не поддерживает функцию трассировки памяти. Спецификация блока дана в документе ARM DDI 0275D «ETB11 Technical Reference Manual».

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

Блоки аппаратной поддержки отладки имеют следующие версии реализации:

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

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

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

5.2.2 Устройство порта JTAG и подключение к внешнему аппаратному отладчику

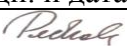
Порт JTAG в микросхеме используется для двух целей:

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

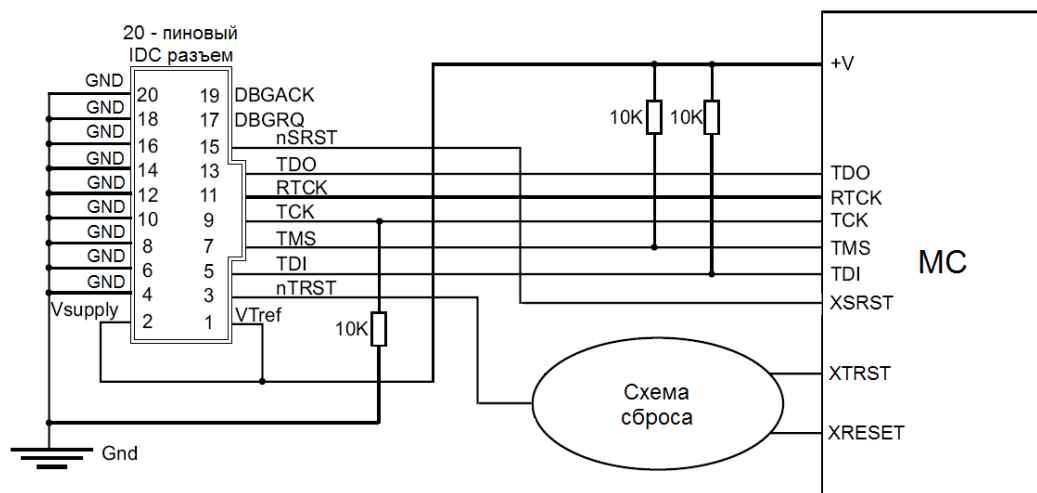
В СБИС СПОВС используются расширенный вариант порта JTAG, включающий в себя следующие выводы микросхемы:

Таблица 5.1 — Выводы микросхемы, используемые для отладки

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

					ЮФКВ.431268.006РЭ			Лист
								42
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

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

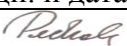


К описанному разъему подключают кабель аппаратного отладчика. Ниже в таблице приведено описание контактов JTAG разъема кабеля отладчика RVI:

					ЮФКВ.431268.006РЭ	Лист	
						43	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв. №подл.		Подп. и дата			Взам. инв. №	Инв. №дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

Таблица 5.2 — Описание контактов JTAG разъема блока RealView ICE для использования с микросхемой СПОВС

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

					ЮФКВ.431268.006РЭ			Лист
								44
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

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

Частота обмена по JTAG интерфейсу зависит от длины кабеля.

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

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

					ЮФКВ.431268.006РЭ			Лист
								45
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

5.3 Системный коммутатор шин HPM_S

Системный коммутатор шин HPM_S (High Performance Matrix) работает согласно стандарту AMBA AXI фирмы ARM. Коммутатор HPM_S работает с 64-разрядными шинами на частоте 164 МГц, он осуществляет коммутацию 4-х мастеров и 5-ти управляемых устройств и является четырехуровневым, т.е. обеспечивает одновременную передачу данных всеми мастерами при условии работы с разными управляемыми устройствами. Также системный коммутатор позволяет программно (запись внутренних конфигурационных регистров по шине APB) настраивать приоритеты мастеров при обращении к одному управляемому устройству.

ЗАМЕЧАНИЕ.

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

На рисунке ниже приведена схема включения коммутатора HPM_S (см. Рисунок 5.5).

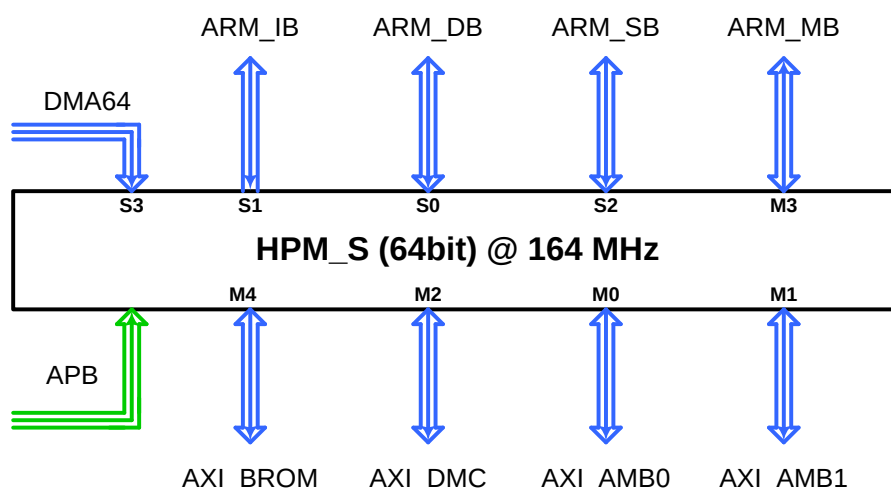


Рисунок 5.5 – Схема включения коммутатора HPM_S

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

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

					ЮФКВ.431268.006РЭ				Лист
									46
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

Таблица 5.3- Параметры slave-портов коммутатора HPM_S

	Название порта	Конвейерный регистр	Активные запросы чтения	Активные запросы записи	Схема защиты от взаимоблокировки (deadlock)	Доступные управляемые устройства
S0	ARM_DB	-	2	2	SingleSlave	все (M0, M1, M2, M3, M4)
S1	ARM_IB	-	2	-	SingleSlave	все (M0, M1, M2, M3, M4)
S2	ARM_SB	+	8	8	SingleSlave	M0, M1, M2
S3	DMA64	-	1	1	SingleSlave	M0, M1, M2, M3

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

Таблица 5.4- Параметры master-портов коммутатора HPM_S

	Название порта	Конвейерный регистр	Кол-во активных запросов записи	Кол-во перемещаемых запросов записи	Программируемое качество обслуживания. Схема QoS	Схема арбитража
M0	AXI_AMB0	+	1	1	-	ProgLRG
M1	AXI_AMB1	+	1	1	-	ProgLRG
M2	AXI_DMC	-	8	4	+	ProgLRG
M3	AXI_MB	+	1	1	-	ProgLRG
M4	AXI_BROM	-	1	1	-	ProgLRG

5.3.1 Параметры портов коммутатора HPM_S

5.3.1.1 Конвейерный регистр

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

5.3.1.2 Активные запросы чтения

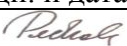
Параметр slave-портов коммутатора. Активным запросом чтения называется запрос, по которому master устройство выдало адрес чтения, но не получило всех считываемых данных. Число в таблице соответствует количеству активных запросов чтения, которое может выдать master устройство, подключенное к соответствующему slave порту.

5.3.1.3 Активные запросы записи

Параметр slave-портов коммутатора. Активным запросом записи называется запрос, по которому master устройство выдало адрес записи, но не закончило передачу записываемых данных и не получило подтверждение записи данных от slave устройства. Число в таблице соответствует количеству активных запросов записи, которое может выдать master устройство, подключенное к соответствующему slave порту.

5.3.1.4 Схема защиты от взаимоблокировки

Параметр slave-портов коммутатора. Все slave-порты коммутатора имеют схему защиты от взаимоблокировок (deadlock'ов). Идея заключается в том, что master-устройство может иметь активные запросы (записи и/или чтения) только к одному

					ЮФКВ.431268.006РЭ			Лист
								47
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

slave устройству (схема SingleSlave). Запрос, адресованный другому slave устройству, принимается только после завершения всех текущих активных запросов. На рисунке ниже приведен алгоритм работы схемы защиты от взаимоблокировок внутри slave порта коммутатора (см. Рисунок 5.6).

5.3.1.5 Доступные управляемые устройства

Параметр slave-портов коммутатора. Slave устройства, подключенные к перечисленным master-портам коммутатора, доступны для master устройства, подключенного к данному slave-порту коммутатора. Например, процессорное ядро ARM может получить доступ к любому из slave устройств, подключенных к коммутатору. Другое master устройство (контроллер ПДП) может получить доступ к обоим банкам внутренней памяти АМВ, банку внешней памяти DMC и, через порт AXI_MB, в NMU часть СБИС. Контроллер ПДП не имеет доступа к банку ПЗУ памяти.

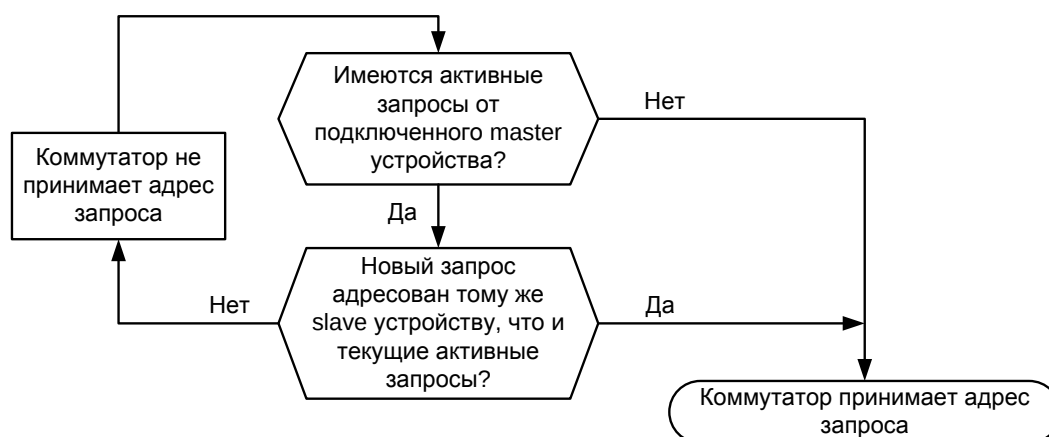


Рисунок 5.6 – Алгоритм работы схемы защиты от взаимоблокировок

5.3.1.6 Количество активных запросов записи master-порта коммутатора

Параметр master-портов коммутатора. Количество генерируемых master-портом активных запросов записи. Число в таблице соответствует числу активных запросов записи, которое может принять подключенное slave устройство. Из таблицы видно, что только контроллер внешней памяти DMC способен обслуживать несколько активных запросов записи одновременно.

5.3.1.7 Количество перемешиваемых запросов записи

Параметр master-портов коммутатора. Slave устройство, поддерживающее несколько активных запросов записи (в СБИС СПОВС это только контроллер DMC), может также поддерживать перемешивание записываемых данных от разных master устройств. Хотя контроллер DMC имеет возможность перемешивать записываемые данные всех активных запросов записи (8 штук), в таблице стоит 4, так как реально к коммутатору подключено только 4 master устройства, а запросы одного master устройства перемешиваться не могут.

Свойство перемешивать записываемые данные позволяет повысить общую скорость записи данных в случае, если подключенные к коммутатору master устройства выдают данные с различными скоростями.

					ЮФКВ.431268.006РЭ			Лист
								48
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	Редько 16.01.14		22755-1					

5.3.1.8 Программируемое качество обслуживания. Схема QoS

Параметр master-портов коммутатора. Эта функция позволяет программным образом настроить коммутатор таким образом, чтобы запросы определенного master устройства гарантированно обслуживались slave устройством. Для того чтобы схема работала, необходимо, чтобы slave устройство поддерживало несколько активных запросов. В СБИС СПОВС, единственное устройство с такими возможностями – это контроллер DMC.

Схема QoS никак не влияет на передачу данных, пока количество активных запросов к slave устройству ниже, чем заданное число в qos_tidemark регистре. Как только количество запросов становится больше или равно заданному количеству, коммутатор начинает принимать запросы только от master устройств, подключенных к заданным в регистре qos_access_control slave портам коммутатора. Ограничение действует до тех пор пока количество активных запросов не упадет ниже количества, заданного в qos_tidemark регистре. Таким образом, выбранное master устройство гарантированно получит доступ ко внешней памяти, независимо от активности других master устройств, подключенных к коммутатору.

5.3.1.9 Схема арбитража коммутатора HPM_S

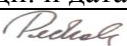
Параметр master-портов коммутатора. Т.к. одновременно несколько master устройств могут обратиться к одному slave устройству, каждый master-порт коммутатора имеет схему арбитража. Коммутатор HPM_S имеет одинаковую схему арбитража, работающую по алгоритму Least Recently Granted (LRG), на всех master портах. Этот алгоритм предполагает, что арбитраж выигрывает master устройство, которое было обслужено наиболее давно.

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

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

Если MI использует схему QoS и она активна, то выиграть арбитраж могут только разрешенные в регистре qos_access_control slave порты коммутатора.

Алгоритм арбитража LRG. Каждый master порт имеет регистры, в которых задаются приоритеты каждого slave порта, иными словами master устройства, подключенного к этому порту. По-умолчанию все master устройства имеют одинаковый приоритет для всех master портов. Приоритеты могут быть изменены путем записи соответствующих конфигурационных регистров по периферийной шине APB. На рисунке ниже приведен пример работы LRG алгоритма арбитража (см. Рисунок 5.7).

					ЮФКВ.431268.006РЭ			Лист
								49
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

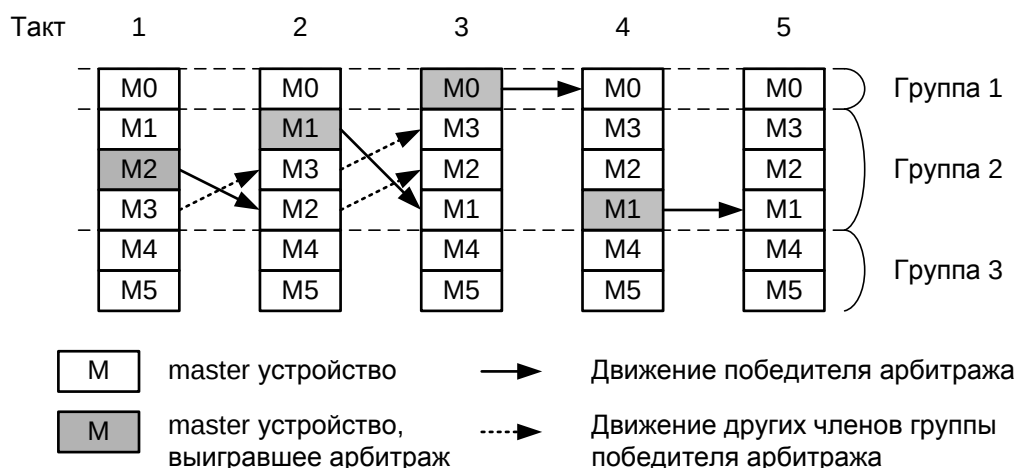


Рисунок 5.7 – Пример работы LRG алгоритма арбитража

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

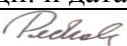
5.3.2 Регистровая модель коммутатора HPM_S

Программно доступные регистры коммутатора HPM_S расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение HPM Base = 0xFFFF83000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

					ЮФКВ.431268.006РЭ				Лист
									50
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	Решет 16.01.14		22755-1						

Таблица 5.5 – Спецификация регистров блока HPM_S

Адрес	Тип	Разряд-ность	Начальное значение	Имя	Описание
HPM Base + 0x400	ЧТ/ЗП	32	0x00000000	qos_tidemark MI0	Регистр числа активных запросов Mix
HPM Base + 0x404	ЧТ/ЗП	32	0x00000000	qos_access_control MI0	Регистр контроля доступа Mix
HPM Base + 0x408	ЧТ/ЗП	32	-	ar_arbitration MI0	Регистр конфигурации арбитража запросов чтения
HPM Base + 0x40C	ЧТ/ЗП	32	-	aw_arbitration MI0	Регистр конфигурации арбитража запросов записи
0x410 to 0x41C	-	-	-	-	Зарезервировано
HPM Base + 0x420	ЧТ/ЗП	32	0x00000000	qos_tidemark MI1	Регистр числа активных запросов Mix
HPM Base + 0x424	ЧТ/ЗП	32	0x00000000	qos_access_control MI1	Регистр контроля доступа Mix
HPM Base + 0x428	ЧТ/ЗП	32	-	ar_arbitration MI1	Регистр конфигурации арбитража запросов чтения
HPM Base + 0x42C	ЧТ/ЗП	32	-	aw_arbitration MI1	Регистр конфигурации арбитража запросов записи
0x430 to 0x43C	-	-	-	-	Зарезервировано
HPM Base + 0x440	ЧТ/ЗП	32	0x00000000	qos_tidemark MI2	Регистр числа активных запросов Mix
HPM Base + 0x444	ЧТ/ЗП	32	0x00000000	qos_access_control MI2	Регистр контроля доступа Mix
HPM Base + 0x448	ЧТ/ЗП	32	-	ar_arbitration MI2	Регистр конфигурации арбитража запросов чтения
HPM Base + 0x44C	ЧТ/ЗП	32	-	aw_arbitration MI2	Регистр конфигурации арбитража запросов записи
0x450 to 0x45C	-	-	-	-	Зарезервировано
HPM Base + 0x460	ЧТ/ЗП	32	0x00000000	qos_tidemark MI3	Регистр числа активных запросов Mix
HPM Base + 0x464	ЧТ/ЗП	32	0x00000000	qos_access_control MI3	Регистр контроля доступа Mix
HPM Base + 0x468	ЧТ/ЗП	32	-	ar_arbitration MI3	Регистр конфигурации арбитража запросов чтения
HPM Base + 0x46C	ЧТ/ЗП	32	-	aw_arbitration MI3	Регистр конфигурации арбитража запросов записи
0x470 to 0x47C	-	-	-	-	Зарезервировано
HPM Base + 0x480	ЧТ/ЗП	32	0x00000000	qos_tidemark MI4	Регистр числа активных запросов Mix
HPM Base + 0x484	ЧТ/ЗП	32	0x00000000	qos_access_control MI4	Регистр контроля доступа Mix
HPM Base + 0x488	ЧТ/ЗП	32	-	ar_arbitration MI4	Регистр конфигурации арбитража запросов чтения
HPM Base + 0x48C	ЧТ/ЗП	32	-	aw_arbitration MI4	Регистр конфигурации арбитража запросов записи
0x490 to 0x4FC	-	-	-	-	Зарезервировано
HPM Base + 0xFC0	ЧТ	32	0x00000004	HPMconfig0	Регистр конфигурации HPM 0 (количество SI)
HPM Base + 0xFC4	ЧТ	32	0x00000005	HPMconfig1	Регистр конфигурации HPM 1 (количество MI)
HPM Base	ЧТ	32	0x00000000	HPMconfig2	Регистр конфигурации HPM 2

					ЮФКВ.431268.006РЭ			Лист
								51
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

+ 0xFC8					
HPM Base + 0xFCC	ЧТ	32	0x00000000	HPMconfig3	Регистр конфигурации HPM 3
0xFD0 to 0xFDC	-	-	-	-	Зарезервировано
HPM Base + 0xFE0	ЧТ	8	0x01	HPMPeriphID0	Идентификатор периферийного устройства. HPMPeriphID0 биты [7:0]
HPM Base + 0xFE4	ЧТ	8	0x13	HPMPeriphID1	Идентификатор периферийного устройства. HPMPeriphID1 биты [15:8]
HPM Base + 0xFE8	ЧТ	8	0x3f	HPMPeriphID2	Идентификатор периферийного устройства. HPMPeriphID2 биты [23:16]
HPM Base + 0xFEC	ЧТ	8	0x00	HPMPeriphID3	Идентификатор периферийного устройства. HPMPeriphID3 биты [31:24]
HPM Base + 0xFF0	ЧТ	8	0x0D	HPMPCellID0	PrimeCell идентификатор. HPMPCellID0 биты [7:0]
HPM Base + 0xFF4	ЧТ	8	0xF0	HPMPCellID1	PrimeCell идентификатор. HPMPCellID1 биты [15:8]
HPM Base + 0xFF8	ЧТ	8	0x05	HPMPCellID2	PrimeCell идентификатор. HPMPCellID2 биты [23:16]
HPM Base + 0xFFC	ЧТ	8	0xB1	HPMPCellID3	PrimeCell идентификатор. HPMPCellID3 биты [31:24]

Далее приведено подробное описание каждого регистра. Попытка доступа в зарезервированные области может привести к непредсказуемому поведению коммутатора.

5.3.2.1 Регистр числа активных запросов Mix

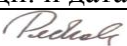
Регистр используется для задания порогового числа активных запросов, после антижения которого включается схема качества обслуживания (QoS). Если число, записанное в этот регистр, больше предельного количества активных запросов, поддерживаемых slave устройством, то схема QoS никогда не отработает для этого master порта (MI). Запись нулевого значения тоже выключает схему QoS для этого MI. Программирование регистра qos_tidemark для master интерфейсов, которые не поддерживают QoS – игнорируется.

5.3.2.2 Регистр контроля доступа Mix

Регистр используется для задания slave порта (SI), доступ от которых разрешен после активизации схемы QoS. Каждый разряд регистра соответствует slave порту коммутатора с порядковым номером этого разряда в регистре. Если бит равен 1, то доступ от этого SI разрешен при активизации схемы QoS, если бит равен 0, то запрещен. Максимальное значение этого регистра $2^4-1 = 0xF$. Запись бит не соответствующих существующим SI игнорируется. Изменения этого регистра реально происходит во время работы схемы арбитража сразу после окончания транзакции записи по шине APB.

5.3.2.3 Регистры конфигурации арбитража запросов чтения и записи

Схемы арбитража настроены по умолчанию после окончания процедуры начальной инициализации СБИС, но также существует возможность программно на-

					ЮФКВ.431268.006РЭ			Лист
								52
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

страивать параметры алгоритма арбитража. Запросы на чтение (AR) и запросы на запись (AW) используют отдельные схемы арбитража. Приоритеты записи и чтения могут быть настроены по-разному – для этого используются два регистра ar_arbitration и aw_arbitration. Карта памяти не содержит достаточно места для отображения регистров приоритетов всех SI, поэтому часть адресной информации кодируется в полях данных при записи конфигурационных регистров. В таблице ниже приведен формат регистров для доступа по записи.

Таблица 5.6 – Формат регистров ar_arbitration и aw_arbitration для записи

Биты	Название	Тип	Функции
[31:24]	interface_number	ЗП	Номер slave порта
[23:16]	-	-	зарезервировано
[15:8]	priority	ЗП	Значение приоритета slave порта 0..255. 0 – максимальный приоритет 255 – минимальный приоритет По-умолчанию после окончания инициализации все SI имеют одинаковый максимальный приоритет.
[7:0]	-	-	зарезервировано

Таким образом, чтобы установить приоритеты всех SI для коммутатора HPM_S, надо выполнить четыре (по количеству SI) записи для каждого master порта (всего $4 \times 5 = 20$ записей), меняя в записываемом значении номер SI и его приоритет.

Чтение регистров ar_arbitration и aw_arbitration выполняется в два этапа:

1. Сначала необходимо выбрать SI, конфигурацию которого необходимо считать. Для этого надо записать регистр, значение которого необходимо прочитать, причем формат записи следующий:

Таблица 5.7 – Формат данных записи регистра для первого этапа чтения

Биты	Название	Тип	Функции
[31:24]	-	ЗП	Фиксировано – 0xFF
[23:8]	-	ЗП	Фиксировано – 0x0000
[7:0]	interface_number	ЗП	Номер SI, конфигурацию арбитража которого необходимо прочитать.

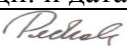
2. Вторым этапом необходимо считать регистр по тому же адресу. В считанном значении закодированы все настройки схемы арбитража заданного на первом этапе SI. Формат считанного регистра приведен в таблице ниже.

Таблица 5.8 – Формат считанного значения регистра

Биты	Название	Тип	Функции
[31:16]	-	-	Читаются нулями
[15:8]	arb_priority	ЧТ	Значение приоритета для заданного SI
[7:0]	interface_number	ЧТ	Номер SI. Совпадает с заданным номером SI на первом этапе.

5.3.2.4 Регистры конфигурации НРМ

Коммутатор содержит 4 регистра конфигурации. В регистре HPMconfig0 закодировано количество SI коммутатора (кол-во подключаемых master устройств). В регистре HPMconfig1 закодировано количество MI коммутатора (количество подключаемых slave устройств). Оставшиеся два регистра содержат нули. Значения регистров не могут быть изменены программным образом.

					ЮФКВ.431268.006РЭ			Лист
								53
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

5.3.2.5 Идентификатор периферийного устройства

Регистр HPMPeriphID0-3 представляет четыре 8-битных регистра, предназначенных только для чтения, которые охватывают адресное пространство от 0xFE0 до 0xFEC. Регистры могут быть концептуально представлены как 32-битный регистр. В таблице ниже приведено назначение полей этого регистра.

Таблица 5.9. Поля идентификатора периферийного устройства HPMPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для HPM – 0x301
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM – 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока (0x3 – для HPM r1p2).
[31:24]	Не определено

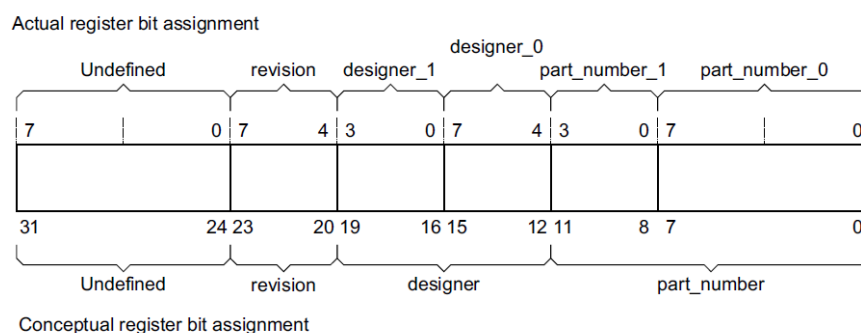


Рисунок 5.8 – Формат регистра HPMPeriphID0-3

5.3.2.6 PrimeCell идентификатор

Регистры HPMPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, предназначенные только для чтения, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. Регистр HPMPCellID установлен в 0xB105F00D. На рисунке ниже приведен формат этого регистра.

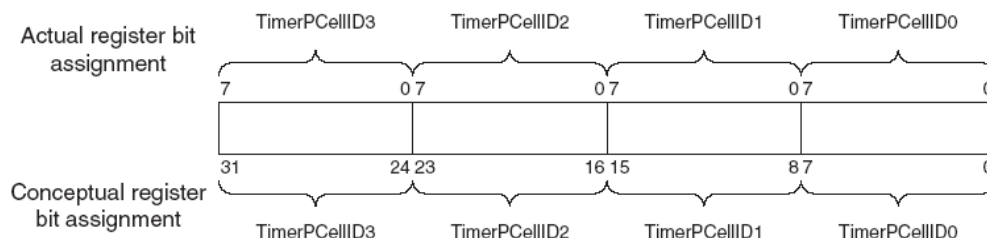


Рисунок 5.9 – Формат регистра HPMPCellID0-3

					ЮФКВ.431268.006РЭ			Лист
								54
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

5.4 Периферийный коммутатор шин HPM_P

Периферийный коммутатор шин HPM_S (High Performance Matrix) работает согласно стандарту AMBA AXI фирмы ARM. Коммутатор HPM_P работает с 32-разрядными шинами на частоте до 82 МГц. Он осуществляет коммутацию 2-х мастеров, работающих по стандарту AXI, и 3-х управляемых устройств (два АНВ устройства и одно APB устройство) и является двухуровневым. Приоритеты мастеров являются фиксированными, используется алгоритм арбитража Round-Robin (RR). На рисунке ниже приведена схема включения коммутатора HPM_P (см. Рисунок 5.21).

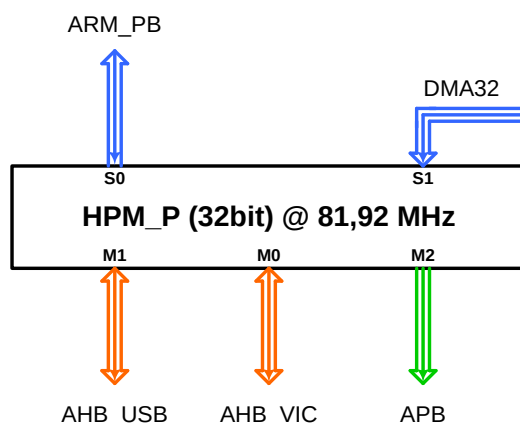


Рисунок 5.10 – Схема включения коммутатора HPM_P

					ЮФКВ.431268.006РЭ			Лист
								55
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

Шина AXI представляет собой интерфейс типа точка-точка – каждое master устройство должно быть подключено к slave-порту (коммутатора или slave устройства), а каждое slave-устройство должно быть подключено к master-порту (коммутатора или master устройства). Таким образом, master устройства подключаются к slave-портам коммутатора, а slave устройства подключаются к master-портам коммутатора.

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

Таблица 5.10- Параметры slave-портов коммутатора НРМ_Р

	Название порта	Конвейрный регистр	Активные запросы чтения	Активные запросы записи	Схема защиты от взаимоблокировки (deadlock)	Доступные управляемые устройства
S0	ARM_PB	-	1	1	SingleSlave	все (M0, M1, M2,)
S1	DMA32	-	1	1	SingleSlave	все (M1, M2)

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

Таблица 5.11- Параметры master-портов коммутатора НРМ_Р

	Название порта	Конвейрный регистр	Преобразователь протокола	Кол-во активных запросов записи	Кол-во пере-мешиваемых запросов записи	Программируемое качество обслуживания. Схема QoS	Схема арбитража
M0	АНВ_VIC	-	AXI->АНВ	1	1	-	FixedRR
M1	АНВ_USB	-	AXI->АНВ	1	1	-	FixedRR
M2	АРВ	-	AXI->АРВ	1	1	-	FixedRR

5.4.1 Параметры портов коммутатора НРМ_Р

Большинство параметров идентично параметрам коммутатора НРМ_С. Ниже приведено описание только новых параметров.

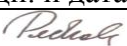
5.4.1.1 Преобразователь протокола

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

5.4.1.2 Схема арбитража коммутатора НРМ_Р

Параметр master-портов коммутатора. Т.к. одновременно несколько master устройств могут обратиться к одному slave устройству, каждый master-порт коммутатора имеет схему арбитража. Коммутатор НРМ_Р имеет одинаковую схему арбитража, работающую по фиксированному алгоритму Round-Robin (RR), на всех master портах. Этот алгоритм предполагает, что все master устройства имеют одинаковый приоритет и обслуживаются по порядку в цикле.

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

					ЮФКВ.431268.006РЭ			Лист
								56
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Алгоритм арбитража RR. В отличие от коммутатора системного коммутатора, периферийный коммутатор не имеет программно доступных регистров и не позволяет настраивать параметры алгоритма арбитража. На рисунке ниже приведен пример работы RR алгоритма арбитража (см. Рисунок 5.11).

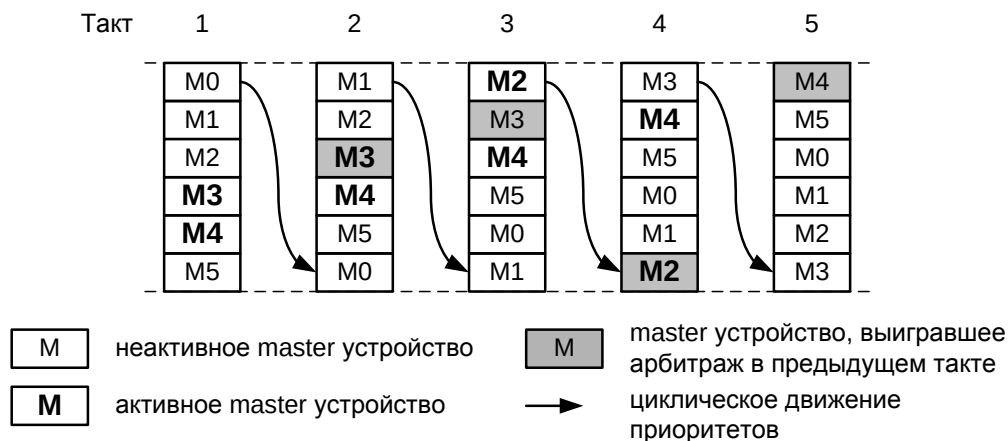


Рисунок 5.11 – Пример работы RR алгоритма арбитража

Каждый такт SI меняют свой приоритет по принципу кольцевого буфера, т.е. SI, имеющий высший приоритет в текущем такте опускается вниз, а остальные SI поднимаются в таблице приоритетов, сохраняя порядок относительно друг друга. Таким образом, активный slave интерфейс будет выигрывать арбитраж в каждом такте, пока не доберется до вершины таблицы и опустится вниз.

Как уже говорилось, решение о победителе арбитража принимается в текущем такте, а применяется в следующем такте. Это означает, что если активный SI, выигравший арбитраж в предыдущем такте, так и остался активным SI с максимальным приоритетом (см. Рисунок 5.11, такт 2), то, независимо от активности в следующем такте, он все равно будет победителем арбитража (см. Рисунок 5.11, такт 3).

5.4.2 Регистровая модель коммутатора НРМ_Р

Коммутатор не имеет подключения к периферийной шине APB и программно доступных регистров.

					ЮФКВ.431268.006РЭ			Лист
								57
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редкофф</i> 16.01.14		22755-1					

5.5 Блок управления сигналами сброса и синхросигналами CRG11

Блок управления сигналами сброса и синхросигналами CRG11 предназначен для управления генерацией синхросигналов и сигналов сброса внутри СБИС СПОВС. Блок имеет следующие характеристики:

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

5.5.1 Устройство блока CRG11

В таблице ниже приведен список внешних выводов, относящихся к блоку управления сигналами сброса и синхросигналами CRG11.

Таблица 5.12 – Выводы микросхемы, относящиеся к CRG11

Вывод	Тип буфера	Примечание
CLKINX0 CLKINX1	in	Входы внешнего синхросигнала от кварца или генератора.
XSRST	inout	Внешний программный сброс микросхемы.
XRESET	in	Вход основного сброса СБИС СПОВС
XTRST	in	Вход сброса тестовой логики
PLLBYPASS	in	Вход выбора режима работы ФАПЧ
STOP	out	Выход индикации входа СБИС в STOP режим
PLLMACRO_L	out	Тестовый выход схемы ФАПЧ (не используется)
FBCLK_PLL	out	Тестовый выход схемы ФАПЧ (не используется)

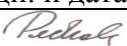
На рисунке ниже (см. Рисунок 5.12) приведена схема блока CRG11.

Входы CLKINX0 и CLKINX1 служат для подключения часового кварца или внешнего генератора на частоте до 82 МГц. Эти входы соединены с блоком АЦП (ADC), который выполняет буферизующую функцию для входного синхросигнала. Опорный для схемы ФАПЧ синхросигнал XCLK поступает с выходов АЦП на входы блока CRG11.

Блок CRG11 содержит схему управления ФАПЧ и генерацией внутренних синхросигналов (Clock Generation Block). Опорный сигнал XCLK попадает на этот блок, умножается на схеме ФАПЧ до частоты 656 МГц, после чего этот сигнал распределяется на 8 внутренних доменов синхросигнала. Каждый домен имеет собственный делитель частоты и схему отключения синхросигнала для уменьшения потребления (Clock Gating).

Большая часть внутренней логики блока CRG11, работает на частоте синхросигнала CCLK – 328 МГц. Этот сигнал получается делением выходного синхросигнала ФАПЧ на два и обладает хорошей скважностью. Именно этот сигнал делится на делителях доменов синхросигналов. СБИС СПОВС имеет вход PLLBYPASS, подача 1 на этот вход выключает схему ФАПЧ, и в этом случае синхросигнал CCLK идентичен синхросигналу XCLK.

Выход STOP предназначен для индикации входа микросхемы в режим пониженного энергопотребления (STOP-режим). Также имеются два тестовых выхода схе-

					ЮФКВ.431268.006РЭ			Лист
								58
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

мы ФАПЧ – PLLMACRO_L и FBCLK_PLL. Первый показывает, что схема ФАПЧ зафиксировалась (PLL locked) и генерируемый синхросигнал на её выходе стабилен. Второй – это вывод обратного синхросигнала схемы ФАПЧ (feedback clock). Оба вывода должны использоваться исключительно в тестовом режиме, в рабочем режиме они не должны быть подключены к печатной плате.

СБИС СПОВС имеет три внешних входа сброса – XRESET, XSRST и XTRST.

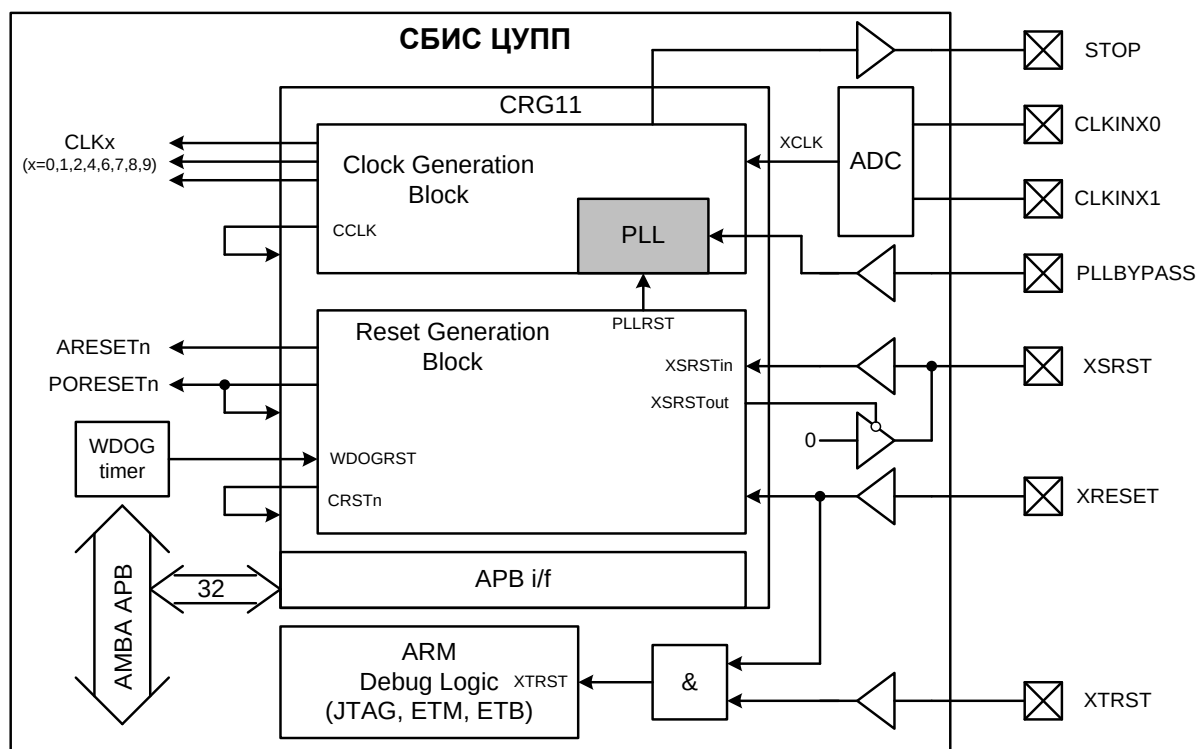


Рисунок 5.12 – Схема блока CRG11

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

- включение питания
- нажатие кнопки на плате

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

XTRST. Дополнительный сброс, отвечающий за сброс внутренней логики, обеспечивающей работу механизма отладки (ETM11, ETB11, JTAG). Сигнал с активным низким уровнем. Сигнал не подключается к блоку CRG11, изображен для полноты внутренней схемы генерации сигналов сброса. Возникает в двух случаях:

- выставляется аппаратным отладчиком через JTAG интерфейс
- также генерируется при возникновении XRESET, т.к. при полном сбросе системы тестовая логика тоже должна сбрасываться (обеспечивается со-

					ЮФКВ.431268.006РЭ			Лист
								59
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		Решет 16.01.14		22755-1				

единением сигналов XRESET и XTRST по логическому «И» внутри микросхемы)

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

При активации одного из источников сброса блок CRG11 генерирует внутренние сигналы сброса СБИС. Таких сигналов 5 штук: ARESETn, PORESETn, XSRSTout, CRSTn и PLLRST.

Блок CRG11 имеет подключение к периферийной шине APB, это позволяет изменять параметры блока путем записи программно доступных регистров. Можно настраивать следующие параметры:

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

Инициализация СБИС происходит с параметрами, заданными на этапе проектирования. После окончания инициализации блок CRG11 находится в полностью настроенном состоянии, т.е. нет необходимости в дополнительной настройке без острой необходимости.

Блок CRG11 предоставляет огромные возможности по управлению внутренними сигналами сброса и синхросигналами, ошибка в программировании конфигурационных регистров приведет к неработоспособности СБИС, вплоть до полного “зависания”! Выйти из такой ситуации возможно только сбросив СБИС внешним сигналом XRESET.

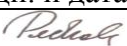
5.5.2 Генератор сигналов сброса

Как уже говорилось выше, при активизации одного из источников сброса, блок CRG11 генерирует внутренние сигналы сброса СБИС. Таких сигналов 5 штук: ARESETn, PORESETn, XSRSTout, CRSTn и PLLRST. В таблице ниже приведено соответствие источников сброса и генерируемых внутренних сигналов сброса.

Таблица 5.13 – Соответствие источников сброса и внутренних сигналов сброса

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

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

					ЮФКВ.431268.006РЭ			Лист
								60
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

5.5.2.1 Внешний системный сброс XRESET

Активация внешнего сброса XRESET приводит к полному сбросу СБИС, генерируются все внутренние сигналы сброса. Внешний сигнал сброса XRESET должен быть активным на входе СБИС в течении как минимум 3 тактов синхросигнала XCLK (40 нс). При изменении частоты синхросигнала необходимо соответствующим образом изменить длительность сигнала сброса.

5.5.2.2 Внешний программный сброс XSRST

Программный сброс XSRST генерируется внешним отладчиком СБИС, например RealView Ice. Сигнал является двунаправленным, т.к. внешний отладчик должен отслеживать состояние микросхемы. Т.е. если сброс будет инициирован изнутри (источники сброса WDOGRST или SOFTRST) или по причине системного сброса (XRESET) – отладчик отследит это событие по генерации внутреннего сигнала сброса XSRSTout. Внешний сигнал сброса XSRST должен быть активным на входе СБИС в течении как минимум 8 тактов синхросигнала CCLK (25 нс). При изменении частоты синхросигнала необходимо соответствующим образом изменить длительность сигнала сброса.

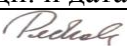
Внешний сброс XSRST игнорируется, если активен внутренний сигнал сброса XSRSTout. Также XSRST игнорируется после деактивации XSRSTout в течение времени, зависящего от длительности активного состояния самого сигнала XSRSTout, которое задается полем SRSTMODE[3:0] регистра CRRSC (см. Таблица 5.14).

Таблица 5.14 – Временя игнорирования сброса XSRST и значения SRSTMODE

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

5.5.2.3 Внутренний программный сброс SOFTRST

Внутренний программный сброс возникает сразу после установки бита SWRSTREQ регистра CRSWR. Генерация внутренних сигналов сброса зависит от значения бита SWRSTM регистра CRRSC. Если бит установлен, то генерируются все внутренние сигналы сброса, кроме PORESETn. Если не установлен, то не происходит сброса управляющих регистров блока CRG11 и схемы ФАПЧ (см. Таблица 5.13).

					ЮФКВ.431268.006РЭ			Лист
								61
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.5.2.4 Сброс от сторожевого таймера WDOGRST

Внутренний сброс, инициируемый блоком сторожевого таймера WDOGRST. Этот сброс генерируется периферийным блоком WDOG (см. п. 5.9). Генерация внутренних сигналов сброса зависит от значения бита WDRSTM регистра CRRSC. Если бит установлен, то генерируются все внутренние сигналы сброса кроме PORESETn. Если не установлен, то не происходит сброса управляющих регистров блока CRG11 и схемы ФАПЧ (см. Таблица 5.13).

5.5.2.5 Внутренний сигнала сброса ARESETn

Сигнал сброса ARESETn используется в СБИС СПОВС для сброса процессорного ядра ARM, блоков NMU, SMU, а также всех периферийных устройств NMC и ARM части. Сигнал генерируется при активации любого из четырех источников сброса. Минимальная длительность активного состояния этого сигнала сброса задается полем ARSTMODE[3:0] регистра CRRSC. Реально длительность активного состояния сигнала ARESETn больше, т.к. он деактивируется в момент времени, когда фронты всех синхросигналов, генерируемых CRG11, совпадают и переключаются из нуля в единицу.

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

5.5.2.6 Внутренний сигнал сброса XSRSTout

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

5.5.2.7 Внутренний сигнал сброса PORESETn

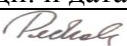
Внутренний сигнал сброса, генерируемый только при подаче внешнего сигнала сброса XRESET. Используется для сброса блоков СБИС, которые не сбрасываются при активации других источников сброса. Часть регистров блока CRG11 сбрасывается этим сигналом.

5.5.2.8 Внутренний сигнал сброса CRSTn

Этот сигнал сброса используется для сброса внутренней логики блока CRG11. Всегда генерируется при активации внешних источников сброса XRESET и XSRST. Генерирование при активации источников сброса WDOGRST и SOFTRST определяется значением конфигурационных бит SWRSTM и WDRSTM регистра CRRSC.

5.5.2.9 Внутренний сигнал сброса PLLRST

Этот сигнал сброса используется для сброса схемы ФАПЧ. Генерируется в тех же случаях, что и сигнал CRSTn, а также при смене делителей схемы ФАПЧ (т.е. при

					ЮФКВ.431268.006РЭ			Лист
								62
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

смене частоты синхросигнала CCLK) и входе СБИС в STOP-режим. Более подробное описание сигнала дано в разделе описания настроек схемы ФАПЧ (см. п. 5.5.3.1).

5.5.3 Генератор синхросигналов

На рисунке ниже приведена схема генератора синхросигналов (Clock Generation Block), входящего в состав блока CRG11 (см. Рисунок 5.13).

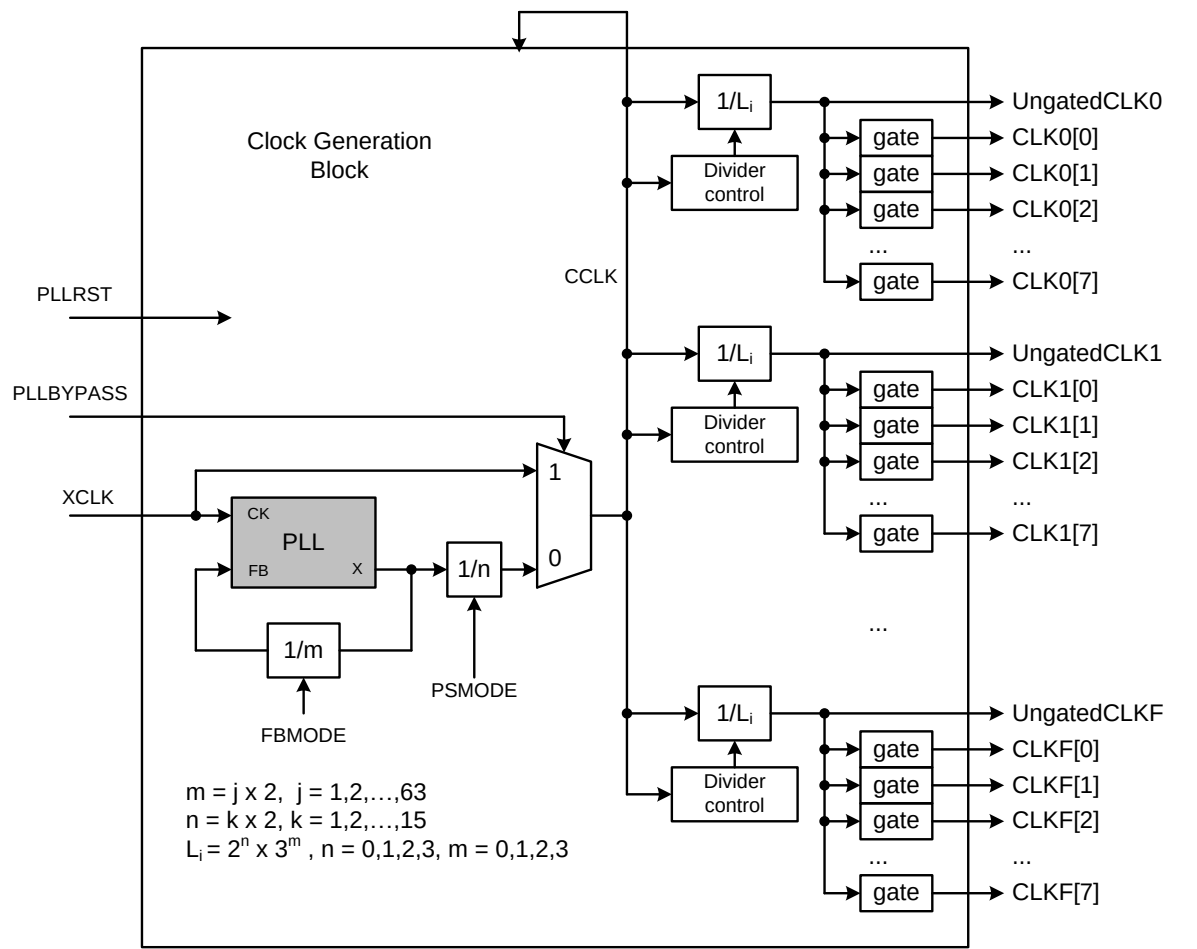


Рисунок 5.13 – Схема генератора синхросигналов

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

$$F_{cclk} = F_{xclk} * m/n$$

где значение m задается полем FBMODE регистра CRPLC, а значение n полем PSMODE того же регистра. При проектировании СБИС заданы следующие значения m=8, n=2, таким образом, на выходе схемы ФАПЧ получается сигнал CCLK частотой до 328 МГц с хорошей скважностью.

					ЮФКВ.431268.006РЭ			Лист
								63
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		16.01.14		22755-1				

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

В общем случае CRG11 поддерживает до 16 внутренних доменов синхросигналов. В СБИС СПОВС используется только 8 доменов синхросигналов – домены с номерами 0,1,2,4,6,7,8 и 9.

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

Каждый домен синхросигнала может иметь до 9 синхросигналов:

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

Реально в каждом домене используется только часть синхросигналов. В процессе проектирования СБИС были заданы частоты всех используемых доменов синхросигналов. Список используемых синхросигналов (выходов CRG11), а также их частоты по-умолчанию, приведен в таблице ниже (см. Таблица 5.15).

					ЮФКВ.431268.006РЭ			Лист
								64
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редко</i> 16.01.14		22755-1				

Таблица 5.15 – Используемые в СБИС СПОВС синхросигналы

Домен	Вывод CRG11	Подключенные блоки СБИС	Частота по-умолчанию	Комментарий
CLK0	UngatedCLK0	ARM11 CORE	164 (макс. 328)	ARM11+ETM11. По умолчанию частота ядра 164 МГц, но она может быть повышена в два раза, путем конфигурирования CRG11.
CLK1	CLK1[0]	NMC PART	328	DSPPU + NMU + SMU + NMC периферия
CLK2	CLK2[0]	HPM_S	164	Системный коммутатор HPM_S
	CLK2[1]	BROM	164	Контроллер банка памяти ПЗУ
	CLK2[2]	AMB0	164	Контроллер банка памяти AMB0
	CLK2[3]	AMB1	164	Контроллер банка памяти AMB1
	CLK2[4]	DMAC	164	Контроллер ПДП (DMAC)
CLK4	UngatedCLK4	CRG11	82	CRG11 APB интерфейс
		ETM11CS	82	Для синхронизации сигналов JTAG
		EXTIRC	82	Контроллер внешних прерываний
		ARMSC	82	Системный контроллер ARMSC
		DMC	82	DMC APB интерфейс
	CLK4[0]	HPM_P	82	Периферийный коммутатор HPM_S
	CLK4[1]	VIC	82	Системный контроллер прерываний VIC
	CLK4[2]	USB2.0	82	Контроллер шины USB2.0
	CLK4[3]	DIT	82	Блок таймеров DIT
		WDOG	82	Сторожевой таймер WDOG
		RTC	82	Часы реального времени RTC
		HPM_S APB	82	APB интерфейс коммутатора HPM_S
		GPIO	82	Порты ввода-вывода GPIO
		UART0	82	Контроллер UART0
		UART1	82	Контроллер UART1
		SPI	82	Контроллер SPI порта
CLK6	CLK6[0]	DMC	328	Синхросигнал удвоенной частоты контроллера внешней памяти (делится пополам внутри блока DMC)
CLK7	CLK7[0]	ATCLK	164	Блок ETB11
CLK8	CLK8[0]	Timer Counter	41	Частота счета счетчиков таймеров
CLK9	CLK9[0]	Watchdog Counter	41	Частота счета счетчика сторожевого таймера

5.5.3.1 Управление схемой ФАПЧ

Сигнал сброса PLLRST.

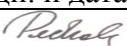
Схема ФАПЧ сбрасывается в следующих случаях:

- активен сигнал внешнего сброса XRESET
- программный сброс (SOFTIRST) и бит маски SWRSTM равен 1
- сброс сторожевого таймера (WDOGRST) и бит маски WDRSTM равен 1
- меняется частота генерации схемы ФАПЧ (значение FBMODE)
- микросхема переходит в STOP режим

Длительность активного состояния сигнала сброса PLLRST задана на этапе проектирования СБИС и не может быть изменена, т.к. жестко связана с параметрами аналоговой схемы ФАПЧ (2 мкс = 192 такта синхросигнала XCLK).

Время стабилизации схемы ФАПЧ (PLL Lockup Time).

Сразу после деактивации сигнала PLLRST, схема ФАПЧ входит в режим ожидания стабилизации выходного синхросигнала. Время стабилизации тоже является параметром схемы ФАПЧ и составляет 100 мкс, этот параметр задается полем LUW-

					ЮФКВ.431268.006РЭ			Лист
								65
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

MODE регистра CRPLC (по-умолчанию 12288 тактов синхросигнала XCLK). Установка этого параметра меньше чем 100 мкс, приведет к неправильному функционированию СБИС.

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

Установка частоты генерации схемы ФАПЧ.

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

Таблица 5.16 – Режимы работы схемы ФАПЧ

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

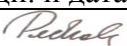
Программное изменение бита PLLBYPASS приведет к немедленному изменению режима работы схемы ФАПЧ.

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

ВНИМАНИЕ! Запрещено выставлять параметры схемы ФАПЧ таким образом, чтобы частота генерируемого синхросигнала CCLK превысила 328 МГц – это приведет к неправильному функционированию СБИС.

5.5.3.2 Изменение частоты синхросигналов доменов

Изменить частоту синхросигнала любого домена можно изменяя параметры делителя этого домена. На рисунке ниже приведена последовательность действий для смены частоты синхросигнала домена x (см. Рисунок 5.14).

					ЮФКВ.431268.006РЭ				Лист
									66
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

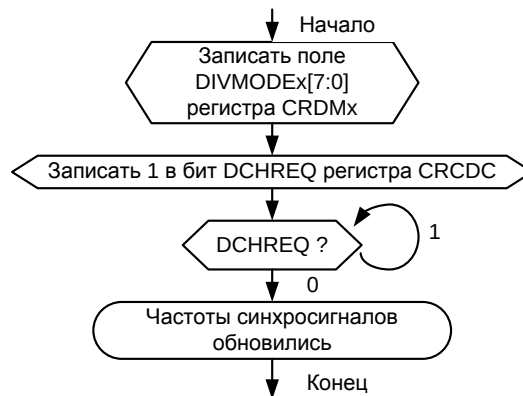


Рисунок 5.14 – Действия для изменения частоты синхросигнала домена

Частота синхросигналов домена определяется по следующей формуле:

$$F_{clkx} = F_{cclk} / (2^n \times 3^m)$$

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

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

ВНИМАНИЕ! Запрещено выставлять параметры делителей таким образом, чтобы частота синхросигнала на выходе делителя превысила максимальное значение частоты для этого домена (см. Таблица 5.15).

ВНИМАНИЕ! Частоты синхросигналов разных доменов связаны между собой! Для правильного функционирования СБИС необходимо задавать частоты доменов руководствуясь рисунком ниже (см. Рисунок 5.15).

					ЮФКВ.431268.006РЭ			Лист
								67
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

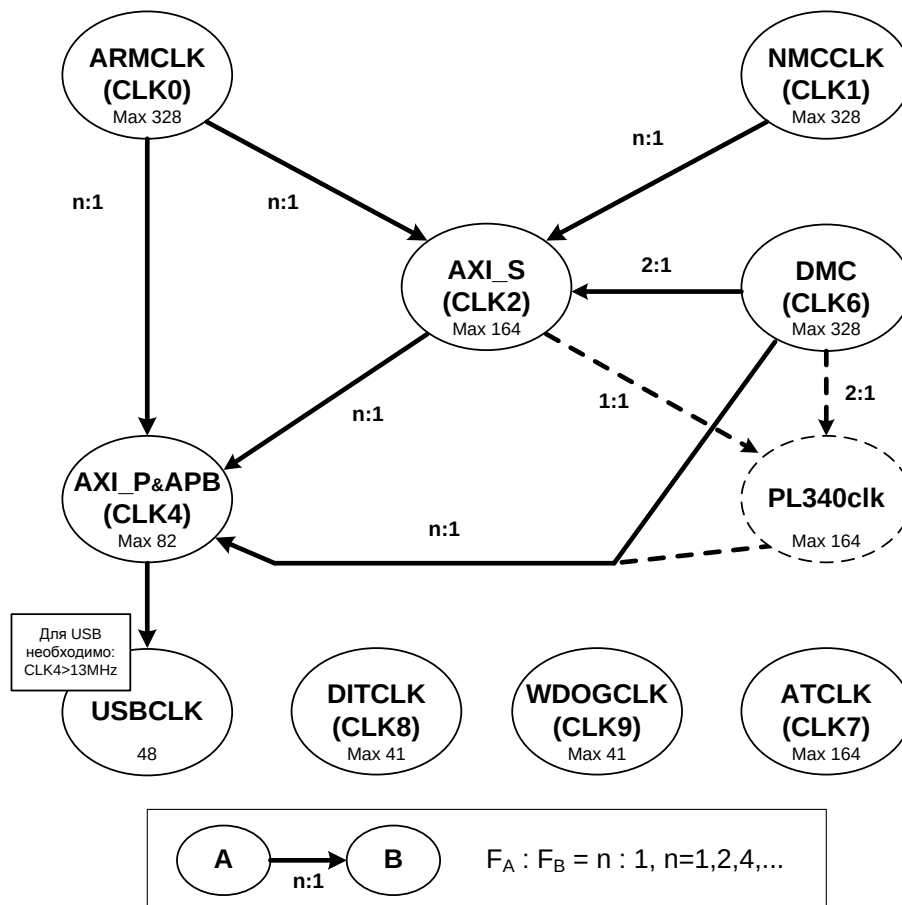


Рисунок 5.15 - Соотношение частот синхросигналов доменов

5.5.3.3 Частичное отключение синхросигналов

СБИС СПОВС позволяет отключать выдачу синхросигнала независимо для разных выходных портов CRG11 (кроме портов UngatedCLKx – синхросигнал на этих выходах можно выключить только в режиме полного выключения). CRG11 производит отключение/включение синхросигналов в соответствии со спецификацией интерфейса управления пониженным энергопотреблением (Low Power Interface) шины AMBA AXI. Основным свойством этого интерфейса является возможность предупредить периферийное устройство об отключении синхросигнала. Периферийное устройство, в свою очередь может разрешить эту операцию, а может и отклонить. Единственным периферийным устройством, поддерживающим этот интерфейс, в СБИС СПОВС является контроллер внешней памяти DMC. Для периферийных устройств, не поддерживающих интерфейс управления пониженным энергопотреблением, считается, что они всегда готовы к отключению синхросигнала.

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

					ЮФКВ.431268.006РЭ			Лист
								68
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	16.01.14		22755-1					

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

ВНИМАНИЕ! При выключении синхросигналов периферийных блоков необходимо учитывать их взаимодействие с другими блоками, которые не будут выключены. Попытка обращения к блоку с выключенным синхросигналом приведет к зависанию СБИС. Особенно аккуратно надо отключать синхросигналы промежуточных блоков. Например, если выключить синхросигнал коммутатора НРМ_Р, все периферийные устройства станут программно не доступны.

5.5.3.4 STOP режим

В этом режиме отключаются все внутренние синхросигналы микросхемы. На рисунке ниже (см. Рисунок 5.16) показаны структурная схема подключения блоков, участвующих в процессе входа/выхода из STOP режима.

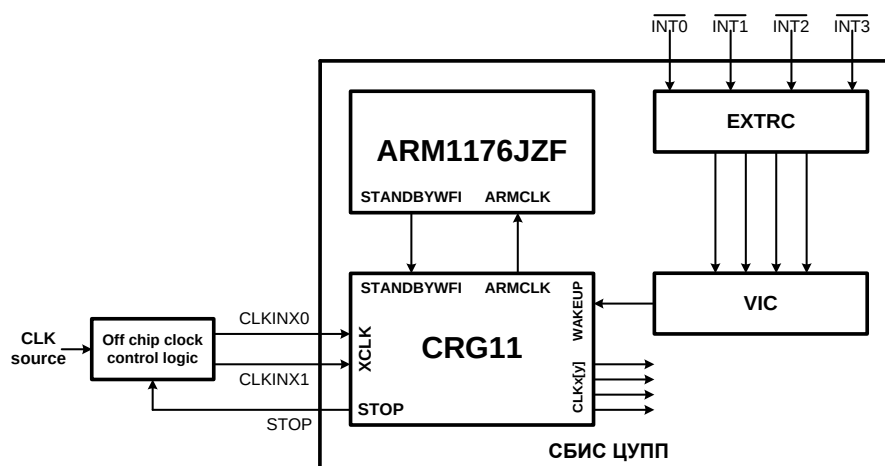


Рисунок 5.16 – Схема соединения блоков, участвующих в переходе в STOP режим.

Вход в STOP режим осуществляется следующим образом:

- Программист выполняет операцию Wait For Interrupt в сопроцессоре CP15 процессорного ядра ARM. Как только все текущие операции будут закончены, ядро переходит в Standby режим и сигнализирует об этом выставлением сигнала STANDBYWFI.
- Если разрешен вход в STOP режим (бит STOPEN регистра CRSTP равен 1), CRG11 пытается остановить все CLKx[y] сигналы (используется взаимодействие по интерфейсу управления пониженным энергопотреблением).
- Когда все CLKx[y] остановлены CRG11 останавливает все UngatedCLKx синхросигналы.
- Далее выставляется сигнал PLLRST и прекращается генерация синхросигнала на схеме ФАПЧ.
- Микросхема выдает 1 на внешний вывод STOP. По этому сигналу можно отключить подачу сигналов CLKINX0 и CLKINX1 на СБИС СПОВС, хотя это и необязательно.
- Возврата к рабочему режиму происходит после генерации внешнего сигнала прерывания - подключенному через VIC к порту WAKEUP блока CRG11. Это прерывание не должно быть маскировано регистрами VIC блока. В случае внешнего прерывания в контроллере внешних прерываний

					ЮФКВ.431268.006РЭ			Лист
								69
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	16.01.14		22755-1					

EXTIRC должен быть выставлен высокий выходной уровень сигнала прерывания. Далее микросхема снимает внешний сигнал STOP, после чего должна быть восстановлена подача сигналов CLKINX0 и CLKINX1 на СБИС, если они были выключены.

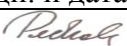
5.5.4 Регистровая модель блока CRG11

Программно доступные регистры блока управления сигналами сброса и синхросигналами CRG11 расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение CRG11 Base = 0xFFF80000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.17 – Спецификация регистров

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

Далее дано описание полей всех регистров.

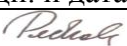
					ЮФКВ.431268.006РЭ			Лист
								70
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

					ЮФКВ.431268.006РЭ				Лист
									71
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		<i>Редкоз</i> 16.01.14			22755-1				

5.5.4.1 Регистр управления PLL. CRPLC

Таблица 5.18 – Формат регистра CRPLC

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

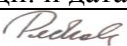
					ЮФКВ.431268.006РЭ			Лист
								72
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

				6'b 111001 m = 57 X 2 6'b 111010 m = 58 X 2 6'b 111011 m = 59 X 2 6'b 111100 m = 60 X 2 6'b 111101 m = 61 X 2 6'b 111110 m = 62 X 2 6'b 111111 m = 63 X 2 PSMODE[3:0] n 4'b0000 n = 1 4'b 0001 n = 1 / (1 X 2) 4'b 0010 n = 1 / (2 X 2) 4'b 0011 n = 1 / (3 X 2) 4'b 0100 n = 1 / (4 X 2) 4'b 0101 n = 1 / (5 X 2) 4'b 0110 n = 1 / (6 X 2) 4'b 0111 n = 1 / (7 X 2) 4'b 1000 n = 1 / (8 X 2) 4'b 1001 n = 1 / (9 X 2) 4'b 1010 n = 1 / (10 X 2) 4'b 1011 n = 1 / (11 X 2) 4'b 1100 n = 1 / (12 X 2) 4'b 1101 n = 1 / (13 X 2) 4'b 1110 n = 1 / (14 X 2) 4'b 1111 n = 1 / (15 X 2) Отметим: 1) Сквозность синхросигнала CCLK не 50/50, когда PSMODE = 4'b0000 2) Бит PLLBYPASS должен быть установлен прежде, чем биты PSMODE станут равными 4'b0000, или прежде чем сменят значение с 4'b0000 3) С изменением значений битов FBMODE, происходит сброс схемы ФАПЧ. Блок CRG11 переходит в режим ожидания стабилизации схемы ФАПЧ. 4) Бит PLLBYPASS не должен меняться пока биты FBMODE[5:0] установлены в 6'b000000
--	--	--	--	---

5.5.4.2 Регистр состояния PLL. CRRDY

Таблица 5.19 – Формат регистра CRRDY

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

					ЮФКВ.431268.006РЭ			Лист
								73
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

				0 – PLL не готова 1 – PLL готова Отметим: Этот бит не показывает, что PLL защелкнулась, а лишь только говорит об истечении времени, запрограммированного в регистре CRPLC.
--	--	--	--	---

5.5.4.3 Регистр управления STOP режимом. CRSTP

Таблица 5.20 – Формат регистра CRSTP

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 2]	Зарезервировано	ЧТ	30'h00000000	-
[1]	STOPMNT	ЧТ/ЗП	1'b0	Бит STOPMNT используется для определения режима работы CRG11. Когда CRG находится в режиме останова (STOP режиме), то бит установлен. 0 – CRG не находится в режиме останова 1 – CRG находится в режиме останова
[0]	STOPEN	ЧТ/ЗП	1'b0	Бит разрешения входа в STOP режим. 0 – вход в STOP режим запрещен 1 – разрешен вход в режим останова при установлении сигнала STANDBYWFI от ядра ARM

5.5.4.4 Регистр управления маской прерывания. CRIMA

Таблица 5.21 – Формат регистра CRIMA

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

5.5.4.5 Регистр снятия прерываний. CRPIC

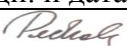
Таблица 5.22 – Формат регистра CRPIC

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

5.5.4.6 Регистр управления сигналами сброса. CRRSC

Таблица 5.23 – Формат регистра CRRSC

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
------	----------	-----	------------------------------	---------------------

					ЮФКВ.431268.006РЭ				Лист
									74
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

			специализации	
[31:20]	Зарезервировано	ЧТ	12'h000	-
[19:16]	SRSTMODE[3:0]	ЧТ/ЗП	4'b0111	Бит используется для установления длительности активного состояния XSRSTout в тактах синхросигнала CCLK. Определяется параметрами внешнего отладчика. 4'b0000 8 X TCCLK 4'b0001 12 X TCCLK 4'b0010 16 X TCCLK 4'b0011 24 X TCCLK 4'b0100 32 X TCCLK 4'b0101 48 X TCCLK 4'b0110 64 X TCCLK 4'b0111 96 X TCCLK ← начальное значение 4'b1000 128 X TCCLK 4'b1001 192 X TCCLK 4'b1010 256 X TCCLK 4'b1011 384 X TCCLK 4'b1100 512 X TCCLK 4'b1101 768 X TCCLK 4'b1110 1024 X TCCLK 4'b1111 1536 X TCCLK
[15:10]	Зарезервировано	ЧТ	6'h00	-
[9]	WDRSTM	ЧТ/ЗП	1'b0	Управляет реакцией CRG на сброс от сторожевого таймера (WDOGRST). 0 – CRSTn и PLLRST не генерируются 1 – CRSTn и PLLRST генерируются
[8]	SWRSTM	ЧТ/ЗП	1'b0	Управляет реакцией CRG на внутренний программный сброс (SOFTIRST). 0 – CRSTn и PLLRST не генерируются 1 – CRSTn и PLLRST генерируются
[7:4]	Зарезервировано	ЧТ	4'h0	-
[3:0]	ARSTMODE[3:0]	ЧТ/ЗП	4'b0111	Бит используется для установления длительностью активного состояния ARESETn в тактах синхросигнала CCLK. 4'b0000 8 X TCCLK (Min.) 4'b0001 12 X TCCLK (Min.) 4'b0010 16 X TCCLK (Min.) 4'b0011 24 X TCCLK (Min.) 4'b0100 32 X TCCLK (Min.) 4'b0101 48 X TCCLK (Min.) 4'b0110 64 X TCCLK (Min.) 4'b0111 96 X TCCLK (Min.) ← н.знач. 4'b1000 128 X TCCLK (Min.) 4'b1001 192 X TCCLK (Min.) 4'b1010 256 X TCCLK (Min.) 4'b1011 384 X TCCLK (Min.) 4'b1100 512 X TCCLK (Min.) 4'b1101 768 X TCCLK (Min.) 4'b1110 1024 X TCCLK (Min.) 4'b1111 1536 X TCCLK (Min.) В таблице выше представлена минимально возможная длительность активного состояния ARESETn.

5.5.4.7 Регистр управления программным сбросом. CRSWR

Таблица 5.24 – Формат регистра CRSWR

Биты	Название	Тип	Значение после	Выполняемая функция
Изм.	Лист	№ докум.	Подп.	Дата
Инв.№подл.	Подп. и дата	Взам.инв.№	Инв.№дубл.	Подп. и дата
22755-2	<i>Редко</i> 16.01.14	22755-1		

ЮФКВ.431268.006РЭ

Лист
75

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

5.5.4.8 Регистр мониторинга сигналов сброса. CRRSM

Таблица 5.25 – Формат регистра CRRSM

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 4]	Зарезервировано	ЧТ	28'h00000000	-
[3]	PORESET	ЧТ/ ЗП0	1'b1	Бит показывает состояние сигнала сброса PORESETn. Устанавливается при генерации сигнала сброса PORESETn. Необходимо записывать нуль для снятия этого бита. 0 – PORESETn не устанавливался 1 – PORESETn устанавливался Бит инициализируется при установлении сигнала PORESETn, запись 1 игнорируется
[2]	SRST	ЧТ/ ЗП0	1'b0	Бит показывает состояние программного сброса (XSRST) от внешнего отладчика. Устанавливается при установке сигнала XSRSTin. Необходимо записывать нуль для снятия этого бита. 0 – XSRSTin не устанавливался 1 – XSRSTin устанавливался Бит инициализируется при установлении сигнала PORESETn. Запись 1 игнорируется.
[1]	SWRST	ЧТ/ ЗП0	1'b0	Бит отслеживает программный сброс. Устанавливается только тогда, когда имел место программный сброс. Необходимо записывать нуль для снятия этого бита. 0 – программного сброса не было 1 – был программный сброс Бит инициализируется при установлении сигнала PORESETn. Запись 1 игнорируется
[0]	WDRST	ЧТ/ ЗП0	1'b0	Бит отслеживает сброс от сторожевого таймера. Устанавливается при генерации сброса от сторожевого таймера. Необходимо записывать нуль для снятия этого бита. 0 – сброса от сторожевого таймера не было 1 – был сброс от сторожевого таймера Бит инициализируется при установлении сигнала PORESETn. Запись 1 игнорируется.

5.5.4.9 Регистр управления делением синхросигнала. CRCDC

В таблице ниже приведен формат регистра CRCDC.

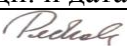
					ЮФКВ.431268.006РЭ			Лист
								76
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№	подл.	Подп. и дата		Взам.инв.№	Инв.№	дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.26 – Формат регистра CRCDC

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

5.5.4.10 Регистр режима деления синхросигнала.CRDMx

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

Таблица 5.27 – Формат регистра CRDMx

Биты	Название	Тип	Значение после инициализации	Выполняемая функция																																																																				
[31: 8]	Зарезервировано	ЧТ	24'h0000000	-																																																																				
[7: 0]	DIVMODEx [7:0]	ЧТ /ЗП	в соответствии с таблицей (Таблица 5.15)	Биты используются для установления режима деления синхросигнала для CLKx домена. Частота генерируемого синхросигнала может быть определена по следующим формулам: $F_{clkx} = F_{cclk} / (2^n \times 3^m)$ (где n=0~3 и m=0~3) f_{CLKx} – частота синхросигнала CLKx домена f_{CCLK} – частота синхросигнала CCLK Параметры M и N могут быть определены согласно битам DIVMODEx: <table><tr><td>DIVMODEx</td><td>m</td><td>n</td><td>Частота CLKx</td></tr><tr><td>8'h00</td><td>0</td><td>0</td><td>fCLKx = fCCLK x (1/1)</td></tr><tr><td>8'h01</td><td>0</td><td>1</td><td>fCLKx = fCCLK x (1/2)</td></tr><tr><td>8'h03</td><td>0</td><td>2</td><td>fCLKx = fCCLK x (1/4)</td></tr><tr><td>8'h07</td><td>0</td><td>3</td><td>fCLKx = fCCLK x (1/8)</td></tr><tr><td>8'h02</td><td>1</td><td>0</td><td>fCLKx = fCCLK x (1/3)</td></tr><tr><td>8'h05</td><td>1</td><td>1</td><td>fCLKx = fCCLK x (1/6)</td></tr><tr><td>8'h0B</td><td>1</td><td>2</td><td>fCLKx = fCCLK x (1/12)</td></tr><tr><td>8'h17</td><td>1</td><td>3</td><td>fCLKx = fCCLK x (1/24)</td></tr><tr><td>8'h08</td><td>2</td><td>0</td><td>fCLKx = fCCLK x (1/9)</td></tr><tr><td>8'h11</td><td>2</td><td>1</td><td>fCLKx = fCCLK x (1/18)</td></tr><tr><td>8'h23</td><td>2</td><td>2</td><td>fCLKx = fCCLK x (1/36)</td></tr><tr><td>8'h47</td><td>2</td><td>3</td><td>fCLKx = fCCLK x (1/72)</td></tr><tr><td>8'h1A</td><td>3</td><td>0</td><td>fCLKx = fCCLK x (1/27)</td></tr><tr><td>8'h35</td><td>3</td><td>1</td><td>fCLKx = fCCLK x (1/54)</td></tr><tr><td>8'h6B</td><td>3</td><td>2</td><td>fCLKx = fCCLK x (1/108)</td></tr><tr><td>8'hD7</td><td>3</td><td>3</td><td>fCLKx = fCCLK x (1/216)</td></tr></table>	DIVMODEx	m	n	Частота CLKx	8'h00	0	0	fCLKx = fCCLK x (1/1)	8'h01	0	1	fCLKx = fCCLK x (1/2)	8'h03	0	2	fCLKx = fCCLK x (1/4)	8'h07	0	3	fCLKx = fCCLK x (1/8)	8'h02	1	0	fCLKx = fCCLK x (1/3)	8'h05	1	1	fCLKx = fCCLK x (1/6)	8'h0B	1	2	fCLKx = fCCLK x (1/12)	8'h17	1	3	fCLKx = fCCLK x (1/24)	8'h08	2	0	fCLKx = fCCLK x (1/9)	8'h11	2	1	fCLKx = fCCLK x (1/18)	8'h23	2	2	fCLKx = fCCLK x (1/36)	8'h47	2	3	fCLKx = fCCLK x (1/72)	8'h1A	3	0	fCLKx = fCCLK x (1/27)	8'h35	3	1	fCLKx = fCCLK x (1/54)	8'h6B	3	2	fCLKx = fCCLK x (1/108)	8'hD7	3	3	fCLKx = fCCLK x (1/216)
DIVMODEx	m	n	Частота CLKx																																																																					
8'h00	0	0	fCLKx = fCCLK x (1/1)																																																																					
8'h01	0	1	fCLKx = fCCLK x (1/2)																																																																					
8'h03	0	2	fCLKx = fCCLK x (1/4)																																																																					
8'h07	0	3	fCLKx = fCCLK x (1/8)																																																																					
8'h02	1	0	fCLKx = fCCLK x (1/3)																																																																					
8'h05	1	1	fCLKx = fCCLK x (1/6)																																																																					
8'h0B	1	2	fCLKx = fCCLK x (1/12)																																																																					
8'h17	1	3	fCLKx = fCCLK x (1/24)																																																																					
8'h08	2	0	fCLKx = fCCLK x (1/9)																																																																					
8'h11	2	1	fCLKx = fCCLK x (1/18)																																																																					
8'h23	2	2	fCLKx = fCCLK x (1/36)																																																																					
8'h47	2	3	fCLKx = fCCLK x (1/72)																																																																					
8'h1A	3	0	fCLKx = fCCLK x (1/27)																																																																					
8'h35	3	1	fCLKx = fCCLK x (1/54)																																																																					
8'h6B	3	2	fCLKx = fCCLK x (1/108)																																																																					
8'hD7	3	3	fCLKx = fCCLK x (1/216)																																																																					

5.5.4.11 Регистр состояния режима LP домена CLKx. CRLPx

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

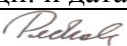
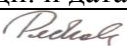
					ЮФКВ.431268.006РЭ			Лист
								77
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.28 – Формат регистра CRLPx

Биты	Название	Тип	Значение после инициализации	Выполняемая функция
[31: 24]	CENx[7:0]	ЧТ	8'hff	Разрешает генерацию CLKx[y].
				CENx[y]
				0 CLKx[y] остановлен
[23: 16]	CACTIVE_Cx[7:0]	ЧТ	8'h00	1 CLKx[y] не остановлен
				Биты отслеживают внутренние сигналы CACTIVE_Cx[7:0]. Биты устанавливаются, когда внешний порт CACTIVEEx[y] имеет высокий уровень. Этот сигнал информирует CRG, что периферийное устройство не готово к отключению синхросигнала.
				Биты отслеживают внутренние сигналы LPOWERHSx[7:0]. Биты устанавливаются, когда в интерфейсе управления пониженным энергопотреблением между CRG11 и периферийным блоком в домене CLKx[y] происходит обмен. Т.е. CRG пытается активировать или включить синхросигнал CLKx[y]. Во время такого обмена невозможна запись бита CSYSREQ_Rx[y].
[15: 8]	LPOWERHSx[7:0]	ЧТ	8'h00	LPO-WERHSx[y]
				0 Запись в CSYSREQ_Rx[y] разрешена
				1 Запись в CSYSREQ_Rx[y] игнорируется
[7: 0]	CSYSREQ_Rx[7:0]	ЧТ/ЗП	8'hff	Биты, управляющие выдачей запроса на включение/выключение синхросигнала CLKx[y] по интерфейсу управления пониженным энергопотреблением.
				CSY-SREQ_Rx[y]
				0 Запрос на остановку CLKx[y]
				1 Запрос на активацию CLKx[y]

					ЮФКВ.431268.006РЭ			Лист
								78
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.6 Системный векторный контроллер прерываний VIC

Контроллер прерываний VIC в СБИС СПОВС имеет следующие характеристики:

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

5.6.1 Устройство векторного контроллера прерываний

Блок не имеет сигналов, соединенных с внешними выводами микросхемы. На рисунке ниже приведена структурная схема блока:

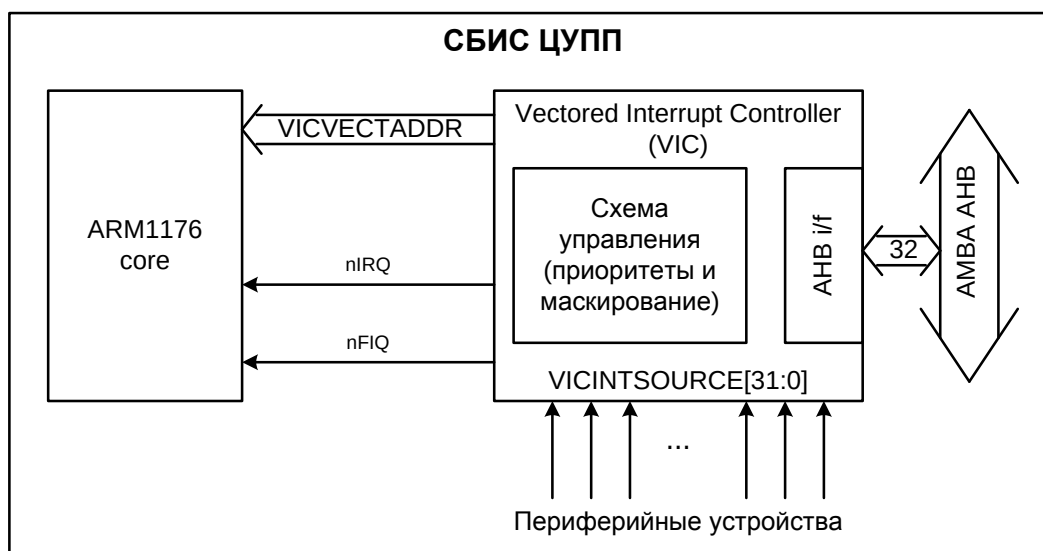


Рисунок 5.17 – Схема системного контроллера прерываний

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

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

Все эти задачи VIC решает на уровне аппаратуры. Вместе с сигналом прерывания процессору передается адрес процедуры обработки (VICVECTADDR) для прерывания с наивысшим приоритетом. Контроллер VIC может контролировать до 32-х прерываний от периферийных устройств, подключенных к входам VICINTSOURCE[31:0].

В процессорном ядре ARM различаются два типа прерываний.

					ЮФКВ.431268.006РЭ			Лист
								79
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

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

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

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

ЗАМЕЧАНИЕ.

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

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

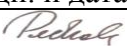
Для каждого из 32-х возможных на входе VIC прерываний в VIC есть регистр адреса процедуры обработки прерывания (VICVECTADDRx). Эти регистры заполняются программно и используются только для обычных прерываний. При возникновении одного или нескольких обычных прерываний VIC выбирает, на основании программных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

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

ЗАМЕЧАНИЕ.

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

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

					ЮФКВ.431268.006РЭ			Лист
								80
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

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

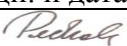
Таблица 5.29 – Соответствие прерываний VIC и прерываний от периферийных устройств

Номер порта VIC	Источник	Описание	Приоритет по умолчанию
VICINTSOURCE[0]	SWINT0	Программное прерывание 0	Высший
VICINTSOURCE[1]	SWINT1	Программное прерывание 1	
VICINTSOURCE[2]	COMMRX	Debug Communication Channel Rx от ARM1176	
VICINTSOURCE[3]	COMMTX	Debug Communication Channel Tx от ARM1176	
VICINTSOURCE[4]	TIMINT0	Прерывание таймера 0	
VICINTSOURCE[5]	TIMINT1	Прерывание таймера 1	
VICINTSOURCE[6]	WDOGINT	Прерывание сторожевого таймера	
VICINTSOURCE[7]	UART0INT	Прерывание канала UART0	
VICINTSOURCE[8]	UART1INT	Прерывание канала UART1	
VICINTSOURCE[9]	NMC0HP	NMC0 высокоприоритетное прерывание	
VICINTSOURCE[10]	NMC1HP	NMC1 высокоприоритетное прерывание	
VICINTSOURCE[11]	ITMS	Прерывание NewMS от интервального таймера	
VICINTSOURCE[12]	RTCINT	Прерывание от блока RTC	
VICINTSOURCE[13]	SSPINT	Общее прерывание от SSP блока	
VICINTSOURCE[14]	DMACSENDINT	Общее прерывание от 4 каналов FtS контроллера DMAC (каналов память->периферия)	
VICINTSOURCE[15]	DMACRECVINT	Общее прерывание от 4 каналов StF контроллера DMAC (каналов периферия->память)	
VICINTSOURCE[16]	ITAUX	Дополнительное прерывание от интервального таймера	
VICINTSOURCE[17]	-	не используются	
VICINTSOURCE[18]	-	не используются	
VICINTSOURCE[19]	-	не используются	
VICINTSOURCE[20]	USBCONINT	Прерывание от USB контроллера, генерируется при подключении и отключении USB кабеля	
VICINTSOURCE[21]	USBINT	Прерывание от USB контроллера	
VICINTSOURCE[22]	EXTINT0	Внешнее прерывание 0	
VICINTSOURCE[23]	EXTINT1	Внешнее прерывание 1	
VICINTSOURCE[24]	EXTINT2	Внешнее прерывание 2	
VICINTSOURCE[25]	EXTINT3	Внешнее прерывание 3	
VICINTSOURCE[26]	NMC0LP	NMC0 низкоприоритетное прерывание	
VICINTSOURCE[27]	NMC1LP	NMC1 низкоприоритетное прерывание	
VICINTSOURCE[28]	nPMUIRQ	От System Metric блока ARM1176	
VICINTSOURCE[29]	PLLRDYINT	Готовность PLL от CRG11	
VICINTSOURCE[30]	nVALFIQ	Тестовое FIQ прерывание от ARM1176	
VICINTSOURCE[31]	nVALIRQ	Тестовое IRQ прерывание от ARM1176	Низший

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

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

Прерывания 28-31 – это тестовые прерывания, в нормальном режиме не используются.

					ЮФКВ.431268.006РЭ			Лист
								81
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

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

Общая схема обработки любого прерывания следующая:

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

5.6.2 Подробности работы с контроллером VIC

5.6.2.1 Регистр VICRAWINTR

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

Значение в этом регистре отражает значение прерывания до маскирования.

5.6.2.2 Определение типа прерывания

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

5.6.2.3 Определение приоритета прерывания

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

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

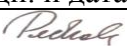
Аппаратный приоритет равен номеру прерывания на входе. 0 – высший приоритет, 31 – низший.

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

5.6.2.4 Маскирование

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

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

					ЮФКВ.431268.006РЭ			Лист
								82
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

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

5.6.2.5 Регистры состояния VICIRQSTATUS и VICFIQSTATUS.

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

5.6.2.6 Адрес процедуры обработки прерывания.

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

5.6.2.7 Регистр программных прерываний VICSOFTINT.

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

5.6.2.8 Взаимодействие процессора и VIC при обработке обычных прерываний.

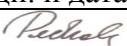
Обычное прерывание (IRQ) может обрабатываться двумя способами. Выбор способа определяется значением разряда VE (VIC Enable) контрольного регистра (Control Register) системного сопроцессора CP15 процессорного ядра ARM.

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

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

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

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

					ЮФКВ.431268.006РЭ			Лист
								83
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

5.6.3 Последовательность шагов при обработке прерываний.

5.6.3.1 *IRQ прерывание. Бит VE=1*

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

1. Сохранить контекст.
2. Разрешить IRQ прерывания процессора, чтобы прерывания более высокого приоритета могли быть обслужены.
3. Выполнить содержательную часть обработки прерывания.
4. Обнулить устройство, вызвавшее прерывание, или обнулить разряд регистра VICSOFTINT, записав единицу в соответствующий разряд регистра VICSOFTINTCLEAR, если прерывание вызвано программным способом.
5. Запретить IRQ прерывания процессора.
6. Записать любое значение в регистр VICADDRESS, чтобы сообщить VIC об окончании обработки прерывания. Эта процедура обязательна для снятия аппаратного маскирования прерываний.
7. Восстановить контекст.
8. Выполнить возврат из процедуры обработки прерывания. При этом будет восстановлено состояние процессора, таким, как оно было в момент прерывания. В частности, будут разрешены IRQ прерывания.

5.6.3.2 *IRQ прерывание. Бит VE=0. Прерывания векторные.*

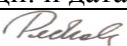
Если бит VE контрольного регистра сопроцессора CP15 равен нулю, то при возникновении прерывания адрес перехода не передается процессорному ядру автоматически. При возникновении IRQ прерывания процессор прерывается и переходит по адресу 0x18 относительно начала таблицы векторов прерываний (т.е. 0x00000018 или 0xFFFFF0018 (в High Vectors Mode)). Далее надо выполнить следующие действия.

1. Считать регистр VICADDRESS и перейти по адресу, содержащемуся в этом регистре. Это действие можно выполнить одной командой загрузки значения регистра VICADDRESS в регистр PC. При этом нет необходимости в дополнительном переходе. Команда загрузки из регистра VICADDRESS может находиться прямо в векторе прерываний.
2. Далее как в п. 5.6.3.1, начиная с пункта 1.

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

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

1. Перейти на общую процедуру обработки IRQ прерываний.
2. Сохранить контекст.
3. Разрешить IRQ прерывания процессора, чтобы прерывания более высокого приоритета могли быть обслужены.

					ЮФКВ.431268.006РЭ			Лист
								84
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

4. Выполнить холостое считывание регистра VICADDRESS для передачи VIC сигнала о начале обработки прерывания. Это чтение включает аппаратное маскирование прерываний.
5. Считать регистр VICIRQSTATUS и определить номер обрабатываемого прерывания.
6. Определить адрес процедуры обработки выбранного прерывания и перейти по этому адресу.
7. Далее как в разделе 5.6.3.1, начиная с пункта 3.

5.6.3.4 FIQ прерывание

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

1. Перейти на процедуру обработки FIQ прерываний. Команда перехода может находиться прямо в адрес-векторе прерывания FIQ.
2. Выполнить содержательную часть обработки прерывания.
3. Обнулить устройство, вызвавшее прерывание, или обнулить разряд регистра VICSOFTINT, записав единицу в соответствующий разряд регистра VICSOFTINTCLEAR, если прерывание вызвано программным способом.
4. Выполнить возврат из процедуры обработки прерывания. При этом будет восстановлено состояние процессора, таким, как оно было в момент прерывания. В частности, будут разрешены FIQ прерывания.

5.6.4 Регистровая модель системного контроллера прерываний VIC

Программно доступные регистры системного контроллера прерываний VIC расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение VIC Base = 0xFFFA0000 и общий размер 4 Кб. В отличие от других периферийных блоков VIC имеет два дополнительных отражения на карту памяти вне области периферийных устройств: по адресам 0xFFFFEF000 и 0xFFFFF000. Это сделано для ускорения входа в процедуру обработчика прерываний. Спецификация регистров представлена в таблице ниже:

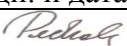
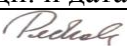
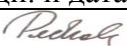
					ЮФКВ.431268.006РЭ				Лист
									85
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

Таблица 5.30. Спецификация регистров контроллера VIC

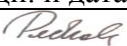
Адрес	Тип	Разр.	Начальное значение	Имя	Описание
VIC Base + 0x000	ЧТ	32	0x00000000	VICIRQSTATUS	Состояние обычных прерываний (IRQ)
VIC Base + 0x004	ЧТ	32	0x00000000	VICFIQSTATUS	Состояние быстрых прерываний (FIQ)
VIC Base + 0x008	ЧТ	32	-	VICRAWINTR	Состояние входов
VIC Base + 0x00C	ЧТ/ ЗП	32	0x00000000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

					ЮФКВ.431268.006РЭ			Лист
								86
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		 16.01.14		22755-1				

+ 0x140	3П				прерывания 16
VIC Base + 0x144	ЧТ/ 3П	32	0x00000000	VICVECTADDR17	Адрес процедуры обработки прерывания 17
VIC Base + 0x148	ЧТ/ 3П	32	0x00000000	VICVECTADDR18	Адрес процедуры обработки прерывания 18
VIC Base + 0x14C	ЧТ/ 3П	32	0x00000000	VICVECTADDR19	Адрес процедуры обработки прерывания 19
VIC Base + 0x150	ЧТ/ 3П	32	0x00000000	VICVECTADDR20	Адрес процедуры обработки прерывания 20
VIC Base + 0x154	ЧТ/ 3П	32	0x00000000	VICVECTADDR21	Адрес процедуры обработки прерывания 21
VIC Base + 0x158	ЧТ/ 3П	32	0x00000000	VICVECTADDR22	Адрес процедуры обработки прерывания 22
VIC Base + 0x15C	ЧТ/ 3П	32	0x00000000	VICVECTADDR23	Адрес процедуры обработки прерывания 23
VIC Base + 0x160	ЧТ/ 3П	32	0x00000000	VICVECTADDR24	Адрес процедуры обработки прерывания 24
VIC Base + 0x164	ЧТ/ 3П	32	0x00000000	VICVECTADDR25	Адрес процедуры обработки прерывания 25
VIC Base + 0x168	ЧТ/ 3П	32	0x00000000	VICVECTADDR26	Адрес процедуры обработки прерывания 26
VIC Base + 0x16C	ЧТ/ 3П	32	0x00000000	VICVECTADDR27	Адрес процедуры обработки прерывания 27
VIC Base + 0x170	ЧТ/ 3П	32	0x00000000	VICVECTADDR28	Адрес процедуры обработки прерывания 28
VIC Base + 0x174	ЧТ/ 3П	32	0x00000000	VICVECTADDR29	Адрес процедуры обработки прерывания 29
VIC Base + 0x178	ЧТ/ 3П	32	0x00000000	VICVECTADDR30	Адрес процедуры обработки прерывания 30
VIC Base + 0x17C	ЧТ/ 3П	32	0x00000000	VICVECTADDR31	Адрес процедуры обработки прерывания 31
VIC Base + 0x200	ЧТ/ 3П	4	0xF	VICVECTPRIORITY0	Программный приоритет 0
VIC Base + 0x204	ЧТ/ 3П	4	0xF	VICVECTPRIORITY1	Программный приоритет 1
VIC Base + 0x208	ЧТ/ 3П	4	0xF	VICVECTPRIORITY2	Программный приоритет 2
VIC Base + 0x20C	ЧТ/ 3П	4	0xF	VICVECTPRIORITY3	Программный приоритет 3
VIC Base + 0x210	ЧТ/ 3П	4	0xF	VICVECTPRIORITY4	Программный приоритет 4
VIC Base + 0x214	ЧТ/ 3П	4	0xF	VICVECTPRIORITY5	Программный приоритет 5
VIC Base + 0x218	ЧТ/ 3П	4	0xF	VICVECTPRIORITY6	Программный приоритет 6
VIC Base + 0x21C	ЧТ/ 3П	4	0xF	VICVECTPRIORITY7	Программный приоритет 7
VIC Base + 0x220	ЧТ/ 3П	4	0xF	VICVECTPRIORITY8	Программный приоритет 8
VIC Base + 0x224	ЧТ/ 3П	4	0xF	VICVECTPRIORITY9	Программный приоритет 9
VIC Base + 0x228	ЧТ/ 3П	4	0xF	VICVECTPRIORITY10	Программный приоритет 10
VIC Base + 0x22C	ЧТ/ 3П	4	0xF	VICVECTPRIORITY11	Программный приоритет 11
VIC Base + 0x230	ЧТ/ 3П	4	0xF	VICVECTPRIORITY12	Программный приоритет 12
VIC Base	ЧТ/	4	0xF	VICVECTPRIORITY13	Программный приоритет 13

					ЮФКВ.431268.006РЭ				Лист
									87
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата			Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14			22755-1				

+ 0x234	3П				
VIC Base + 0x238	ЧТ/ 3П	4	0xF	VICVECTPRIORITY14	Программный приоритет 14
VIC Base + 0x23C	ЧТ/ 3П	4	0xF	VICVECTPRIORITY15	Программный приоритет 15
VIC Base + 0x240	ЧТ/ 3П	4	0xF	VICVECTPRIORITY16	Программный приоритет 16
VIC Base + 0x244	ЧТ/ 3П	4	0xF	VICVECTPRIORITY17	Программный приоритет 17
VIC Base + 0x248	ЧТ/ 3П	4	0xF	VICVECTPRIORITY18	Программный приоритет 18
VIC Base + 0x24C	ЧТ/ 3П	4	0xF	VICVECTPRIORITY19	Программный приоритет 19
VIC Base + 0x250	ЧТ/ 3П	4	0xF	VICVECTPRIORITY20	Программный приоритет 20
VIC Base + 0x254	ЧТ/ 3П	4	0xF	VICVECTPRIORITY21	Программный приоритет 21
VIC Base + 0x258	ЧТ/ 3П	4	0xF	VICVECTPRIORITY22	Программный приоритет 22
VIC Base + 0x25C	ЧТ/ 3П	4	0xF	VICVECTPRIORITY23	Программный приоритет 23
VIC Base + 0x260	ЧТ/ 3П	4	0xF	VICVECTPRIORITY24	Программный приоритет 24
VIC Base + 0x264	ЧТ/ 3П	4	0xF	VICVECTPRIORITY25	Программный приоритет 25
VIC Base + 0x268	ЧТ/ 3П	4	0xF	VICVECTPRIORITY26	Программный приоритет 26
VIC Base + 0x26C	ЧТ/ 3П	4	0xF	VICVECTPRIORITY27	Программный приоритет 27
VIC Base + 0x270	ЧТ/ 3П	4	0xF	VICVECTPRIORITY28	Программный приоритет 28
VIC Base + 0x274	ЧТ/ 3П	4	0xF	VICVECTPRIORITY29	Программный приоритет 29
VIC Base + 0x278	ЧТ/ 3П	4	0xF	VICVECTPRIORITY30	Программный приоритет 30
VIC Base + 0x27C	ЧТ/ 3П	4	0xF	VICVECTPRIORITY31	Программный приоритет 31
VIC Base + 0xF00	ЧТ/ 3П	32	0x00000000	VICADDRESS	Адрес процедуры обработки для текущего IRQ прерывания
VIC Base + 0xFE0	ЧТ	8	0x92	VICPERIPHID0	Идентификатор периферийного устройства, разряды [7:0]
VIC Base + 0xFE4	ЧТ	8	0x11	VICPERIPHID1	Идентификатор периферийного устройства, разряды [15:8]
VIC Base + 0xFE8	ЧТ	8	0x04	VICPERIPHID2	Идентификатор периферийного устройства, разряды [23:16]
VIC Base + 0xFEC	ЧТ	8	0x00	VICPERIPHID3	Идентификатор периферийного устройства, разряды [31:24]
VIC Base + 0xFF0	ЧТ	8	0x0D	VICPCCELLID0	Идентификатор PrimeCell, разряды [7:0]
VIC Base + 0xFF4	ЧТ	8	0xF0	VICPCCELLID1	Идентификатор PrimeCell, разряды [15:8]
VIC Base + 0xFF8	ЧТ	8	0x05	VICPCCELLID2	Идентификатор PrimeCell, разряды [23:16]
VIC Base	ЧТ	8	0xB1	VICPCCELLID3	Идентификатор PrimeCell,

					ЮФКВ.431268.006РЭ			Лист
								88
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14			22755-1			

+ 0xFFC					разряды [31:24]
---------	--	--	--	--	-----------------

Далее приведено подробное описание каждого регистра.

5.6.4.1 Регистр состояния обычных прерываний (IRQ). VICIRQSTATUS

Таблица 5.31 – Формат регистра VICIRQSTATUS

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

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

5.6.4.2 Регистр состояния быстрых прерываний (FIQ). VICFIQSTATUS

Таблица 5.32 – Формат регистра VICFIQSTATUS

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

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

5.6.4.3 Регистр состояния входов. VICRAWINTR

Таблица 5.33 – Формат регистра VICRAWINTR

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

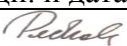
Каждый разряд регистра отражает состояние источника прерываний на входе VIC, т.е. до наложения масок. Состояние всех входов VIC отображается в регистре VICRAWINTR. Каждый разряд регистра соответствует входу с номером, равным номеру разряда. Значение разряда 0 означает отсутствие прерывания на входе,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

5.6.4.4 Регистр выбора типа прерывания. VICINTSELECT

Таблица 5.34 – Формат регистра VICINTSELECT

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

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

					ЮФКВ.431268.006РЭ			Лист
								89
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.6.4.5 Регистр маски прерываний. VICINTENABLE

Таблица 5.35 – Формат регистра VICINTENABLE

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

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

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

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

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

5.6.4.6 Регистр обнуления маски прерываний. VICINTENCLEAR

Таблица 5.36 – Формат регистра VICINTENCLEAR

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

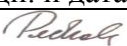
5.6.4.7 Регистр программных прерываний. VICSOFTINT

Таблица 5.37 – Формат регистра VICSOFTINT

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

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

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

					ЮФКВ.431268.006РЭ			Лист
								90
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.6.4.8 Регистр обнуления программных прерываний. VICSOFTINTCLEAR

Таблица 5.38 – Формат регистра VICSOFTINTCLEAR

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

5.6.4.9 Флаг защищенного режима. VICPROTECTION

Таблица 5.39 – Формат регистра VICPROTECTION

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

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

5.6.4.10 Регистр маски программных приоритетов. VICSWPRIORITYMASK

Таблица 5.40 – Формат регистра VICSWPRIORITYMASK

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

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

5.6.4.11 Регистры адресов процедур обработки прерывания.

VICVECTADDR[0:31]

Таблица 5.41 – Формат регистра VICVECTADDR[0:31]

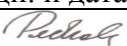
Биты	Название	Тип	Выполняемая функция
[31:0]	VectorAddr[31:0]	ЧТ/ЗП	Содержит векторные адреса ISR(процедур обработчиков прерываний). Запись в регистр должна выполняться, когда соответствующее прерывание заблокировано. В противном случае, результат не определен.

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

Таблица 5.42 – Формат регистра VICVECTPRIORITY[0:31]

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

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

					ЮФКВ.431268.006РЭ			Лист
								91
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

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

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

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

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

5.6.4.13 Регистр приоритета подчиненного VIC. $VICSWPRIORITYDAISY$

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

5.6.4.14 Регистр адреса процедуры обработки текущего IRQ прерывания. $VICADDRESS$

Таблица 5.43– Формат регистра $VICADDRESS$

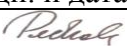
Биты	Название	Тип	Выполняемая функция
[31:0]	VectAddr	ЧТ/ЗП	Регистр содержит значение регистра $VICVECTADDR_n$, где n – номер IRQ прерывания, переданного для обработки процессору, а если такого нет, то значение регистра $VICVECTADDR_n$ для последнего обработанного IRQ прерывания.

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

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

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

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

					ЮФКВ.431268.006РЭ			Лист
								92
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.6.4.15 Идентификатор периферийного устройства. VICPERIPHID0-3

Таблица 5.44 – Поля идентификатора периферийного устройства VICPERIPHID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для VIC – 0x192
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока. Номер ревизии начинается с нуля
Configuration[31:24]	Является вариантом конфигурации периферии. Для VIC192 0x0

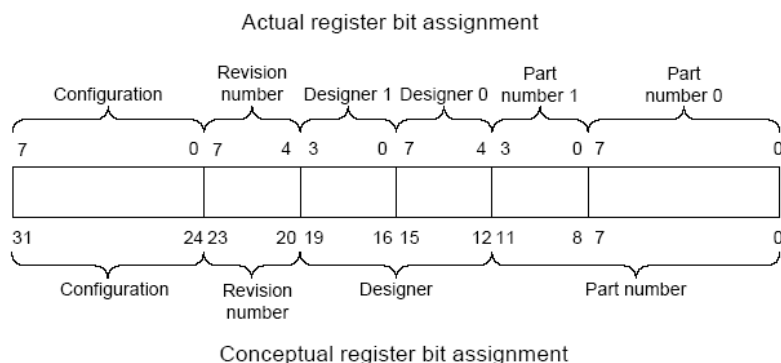


Рисунок 5.18 – Формат регистра VICPERIPHID0-3

5.6.4.16 PrimeCell идентификатор. VICPCCELLID[0:3]

Регистры VICPCCELLID[0:3] – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, предназначенные только для чтения, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. На рисунке ниже (см. Рисунок 5.19) приведено распределение полей этого регистра.

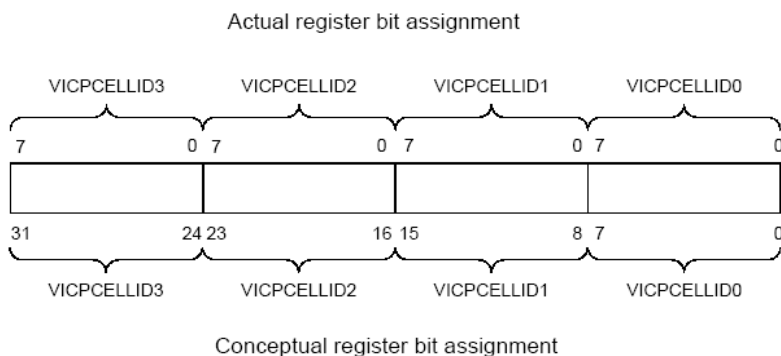


Рисунок 5.19 – Формат регистра VICPCCELLID[0:3]

					ЮФКВ.431268.006РЭ			Лист
								93
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

5.7 Контроллер внешних прерываний EXTIRC

Контроллер внешних прерываний EXTIRC в СБИС СПОВС имеет следующие характеристики:

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

5.7.1 Устройство контроллера внешних прерываний

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

Таблица 5.45. Выводы микросхемы, входящие в состав блока EXTIRC

Вывод	Тип буфера	Примечание
XINT0 ... XINT3	in	4 входа внешних прерываний

На рисунке ниже приведена схема контроллера внешних прерываний:

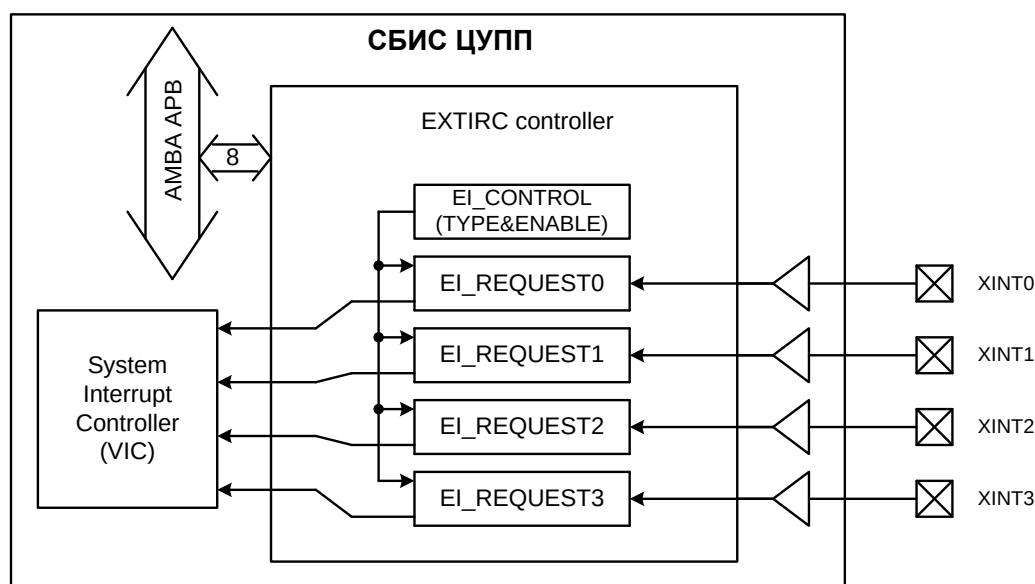


Рисунок 5.20 – Схема контроллера внешних прерываний

СБИС СПОВС имеет 4 входа внешних прерываний, каждое из которых отдельно обрабатывается системным контроллером прерываний VIC.

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

- по низкому уровню

					ЮФКВ.431268.006РЭ			Лист
								94
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	Решет 16.01.14		22755-1					

- по высокому уровню
- по фронту
- по срезу

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

5.7.2 Регистровая модель контроллера внешних прерываний EXTIRC

Программно доступные регистры контроллера портов общего назначения GPIO расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение EXTIRC Base = 0xFFF8E000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.46. Спецификация регистров контроллера EXTIRC

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

Далее приведено подробное описание каждого регистра.

5.7.2.1 Регистр разрешения внешних прерываний. EIENB

В таблице ниже приведен формат регистров EIENB.

Таблица 5.47. Формат регистров EIENB

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

5.7.2.2 Регистр запроса внешних прерываний. EIREQ

В таблице ниже приведен формат регистра EIREQ.

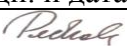
					ЮФКВ.431268.006РЭ			Лист
								95
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.48. Формат регистра EIREQ

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

5.7.2.3 Регистр уровня внешних прерываний. EILVL

В таблице ниже приведен формат регистра EILVL.

Таблица 5.49.– Формат регистра EILVL

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

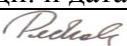
5.7.3 Прерывания

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

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

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

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

					ЮФКВ.431268.006РЭ			Лист
								96
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.8 Блок сдвоенных таймеров DIT

Блок сдвоенных таймеров в СБИС СПОВС имеет следующие характеристики:

- два 32/16 – разрядных таймера
- для каждого таймера можно настроить режим его работы (свободный счет, периодичный, одиночный)
- счетчики таймеров работают на частоте до 41 МГц
- каждый таймер имеет собственный выход прерывания
- подключение к системной шине СБИС через AMBA APB интерфейс, используются полный, 32 разрядный интерфейс шины данных

5.8.1 Устройство блока сдвоенных таймеров

Блок не имеет сигналов, соединенных с внешними выводами микросхемы.

На рисунке ниже приведена структурная схема блока:

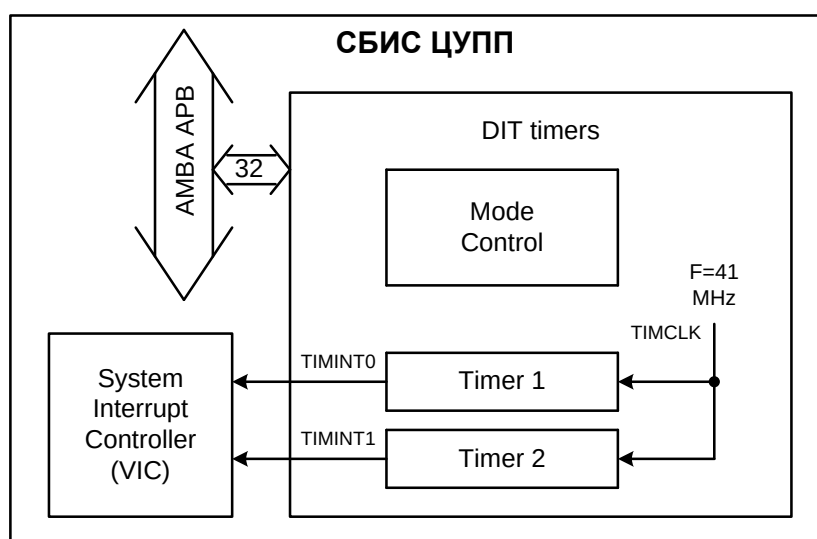


Рисунок 5.21 – Схема блока сдвоенных таймеров

Блок состоит из двух идентичных блоков Timer 1 и Timer 2, каждый из которых может функционировать в 16 или 32-битном режиме. Внутри каждого блока находится счетчик FRC (Free Running Counter). Оба счетчика переключаются на частоте до 41 МГц, причем возможно программным образом уменьшить частоту счета в 16 или 256 раз независимо для каждого счетчика.

Оба таймера могут быть программно настроены на работу в одном из следующих режимов:

- свободный счет (free-running) – счетчик таймера постоянно декрементируется, счет автоматически начинается с максимального значения после достижения нуля
- периодичный (periodic) – аналогично предыдущему, только после достижения нуля счет начинается со значения, предварительно загруженного в регистр TimerXLoad
- одиночный (one-shot) – счетчик начинает декрементироваться со значения, загруженного в регистр TimerXLoad, после достижения нуля счет останавливается.

					ЮФКВ.431268.006РЭ			Лист
								97
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

Ниже приведена последовательность работы с таймерами:

1. Необходимо провести начальные настройки используемого таймера, т.е. прописать поля регистра TimerXControl.
2. Загрузить начальное значение счетчика в регистр TimerXLoad.
3. Разрешить счет путем установки бита TimerEn регистра TimerXControl. Если после установки этого бита записать новое значение в регистр TimerXLoad, то счетчик продолжит декрементироваться с нового значения.
4. Другой способ записать новое значение счетчика заключается в записи фонового регистра TimerXBGLoad. Эта запись не будет иметь мгновенного эффекта, вместо этого записанное значение переписывается в регистр TimerXLoad автоматически по достижению счетчиком нулевого значения.
5. Текущее значение счетчика может быть считано из регистра TimerXValue в любой момент времени.
6. Каждый раз, когда счетчик достигает нулевого значения, генерируется прерывание. Для сброса сгенерированного прерывания необходимо произвести запись в регистр TimerXIntClr. Генерация прерывания может быть замаскирована путем установки соответствующих бит регистра управления TimerXControl.

5.8.2 Регистровая модель блока сдвоенных таймеров

Программно доступные регистры блока сдвоенных таймеров DIT расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение DIT Base = 0xFFF88000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

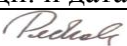
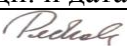
					ЮФКВ.431268.006РЭ			Лист
								98
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.50. Спецификация регистров блока DIT

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

					ЮФКВ.431268.006РЭ			Лист
								99
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

+ 0xFF4					TimerPCellID1 биты [15:8]
DIT Base + 0xFF8	ЧТ	8	0x05	TimerPCellID2	PrimeCell идентификатор. TimerPCellID2 биты [23:16]
DIT Base + 0xFFC	ЧТ	8	0xB1	TimerPCellID3	PrimeCell идентификатор. TimerPCellID3 биты [31:24]

Далее приведено подробное описание каждого регистра. Заметим, что использование X в названии регистров означает, что регистр относится к Timer 1, либо к блоку Timer 2.

5.8.2.1 Регистр загрузки. *TimerXLoad*

Регистр TimerXLoad – это 32-х битный регистр, в который программно записывается значение, начиная с которого счетчик должен декрементироваться. Если периодический режим активен и текущее значение достигает нуля, то происходит перезагрузка счетчика.

При записи, текущее значение счетчика незамедлительно меняется на значение, записанное по активному фронту синхросигнала TIMCLK.

Отметим, что минимальным значением для TimerXLoad является 1. Если значение TimerXLoad установлено в 0, то прерывания вырабатываются незамедлительно.

Значение данного регистра всегда переписывается, если прописано значение регистра TimerXBGLoad. Но текущее значение счета изменяется не сразу. Если значения записываются сразу в оба регистра: TimerXLoad и TimerXBGLoad, до прихода активного фронта сигнала TIMCLK, тогда по следующему активному фронту сигнала TIMCLK значение, записанное в регистре TimerXLoad, заменяется текущим значением счета. После чего, каждый раз, когда счетчик достигает значения нуля, текущее значение устанавливается в значение, записанное в TimerXBGLoad.

Значение, считанное из TimerXLoad, - это всегда то значение, которое будет прописано в счетчике, как только он станет равным нулю в периодическом режиме.

5.8.2.2 Регистр текущего значения. *TimerXValue*

Регистр TimerXValue – 32-х битный регистр, предназначенный только для чтения, содержит текущее значение декрементирующего счетчика.

После процесса загрузки, когда прописывается новое значение в TimerXLoad, в регистре TimerXValue незамедлительно отражается новое загруженное значение

Отметим, что старшие 16 бит 32-х битного регистра TimerXValue не должны автоматически устанавливаться в нуль, когда используется 16-битный временной режим. Если TimerXValue находится в 16-битном режиме, тогда старшие 16 бит регистра TimerXValue должны иметь ненулевые значения, если перед этим TimerXValue был в 32-битном режиме, и произошла запись в регистр TimerXLoad, прежде чем перейти в 16-битный режим.

5.8.2.3 Регистр управления. *TimerXControl*

Формат регистра управления приведен в таблице ниже.

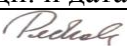
					ЮФКВ.431268.006РЭ				Лист
									100
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

Таблица 5.51. Формат регистра TimerXControl

Биты	Название	Тип	Функции
[31:8]	-	-	Резервные биты, не изменяются, игнорируются при чтении
[7]	TimerEn	ЧТ/ЗП	Бит разрешения: 0 = Timer X FRC неактивен (по умолчанию) 1 = Timer X FRC активен
[6]	TimerMode	ЧТ/ЗП	Бит режима: 0 = Timer X FRC находится в режиме свободного доступа (по умолчанию) 1 = Timer X FRC находится в периодическом режиме.
[5]	IntEnable	ЧТ/ЗП	Бит разрешения прерываний: 0 = Timer X FRC Прерывание неактивно 1 = Timer X FRC Прерывание активно (по умолчанию).
[4]	-	-	Резервные биты, не изменяются, игнорируются при чтении
[3:2]	TimerPre	ЧТ/ЗП	Биты масштабирования частоты счета (делитель частоты): 00 = синхросигнал не делится (по умолчанию) 01 = синхросигнал делится на 16 10 = синхросигнал делится на 256 11 = не используется.
[1]	TimerSize	ЧТ/ЗП	Выбирает 16/32 битный счетчик используется: 0 = 16-битный счетчик (по умолчанию) 1 = 32-битный счетчик.
[0]	OneShot	ЧТ/ЗП	Выбирает однократный или многократный режим счетчика: 0 = многократный режим (по умолчанию) 1 = однократный режим.

Заметим, что состояние счетчика, разрядность или делитель частоты – это установки, которые не должны меняться, пока Timer X FRC работает. Если необходима новая конфигурация, тогда Timer X FRC должен находиться в неактивном состоянии и значения новой конфигурации записываются в соответствующие регистры. Timer X FRC должен быть заново активирован, после чего изменения конфигурации заканчиваются. Отказ следовать данному порядку действий может послужить причиной непредсказуемого поведения блока.

5.8.2.4 Регистр снятия прерываний. TimerXIntClr

Любая запись в данный регистр снимает прерывание с выхода счетчика.

5.8.2.5 Регистр состояния прерывания до наложения маски. TimerXRIS

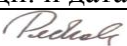
Регистр TimerXRIS указывает на состояние прерываний до наложения маски от счетчика. Формат регистра представлен в таблице ниже.

Таблица 5.52. Формат регистра TimerXRIS

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

5.8.2.6 Регистр состояния прерывания после наложения маски. TimerXMIS

Регистр TimerXMIS определяет прерывания после наложения маски от счетчика. Это значение представляется логическим элементом «И» между битом состояния

					ЮФКВ.431268.006РЭ			Лист
								101
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

необработанного прерывания (TimerXRIS) и битом разрешения прерываний (IntEnable) регистра управления, и представляет аналогичное значение выходного сигнала TIMERINTX. Формат регистра представлен в таблице ниже.

Таблица 5.53. Формат регистра TimerXMIS

Биты	Имя	Тип	Функция
[31:1]	-	-	Резервные биты, не изменяются, игнорируются при чтении
[0]	TimerXMIS	ЧТ	Состояние бита, разрешающего прерывание от счетчика

5.8.2.7 Фоновый регистр загрузки. TimerXBGLoad

Регистр TimerXBGLoad – является 32-битным регистром, который содержит значение, с которого счетчик начинает декрементироваться. Данное значение используется для перезагрузки счетчика в периодическом режиме, и когда текущее значение счета достигает нуля.

Это предоставляет альтернативный метод доступа к регистру TimerXLoad. Отличием является то, что запись в TimerXBGLoad не означает, что счетчик устанавливает новое значение сразу.

Чтение из этого регистра вернет то же самое значение, возвратное из TimerXLoad. Для получения более полной информации смотри раздел «Регистр загрузки. TimerXLoad».

5.8.2.8 Идентификатор периферийного устройства. TimerPeriphID0-3

Регистр TimerPeriphID0-3 представляет четыре 8-битных регистра, предназначенных только для чтения, которые охватывают адресное пространство от 0xFE0 до 0xFEC. Регистры могут быть концептуально представлены как 32-битный регистр. В таблице ниже приведен формат полей этого регистра.

Таблица 5.54. Поля идентификатора периферийного устройства TimerPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для DIT – 0x804
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM – 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока. Номер ревизии начинается с нуля
Configuration[31:24]	Является вариантом конфигурацией периферии. Для DIT 0x0

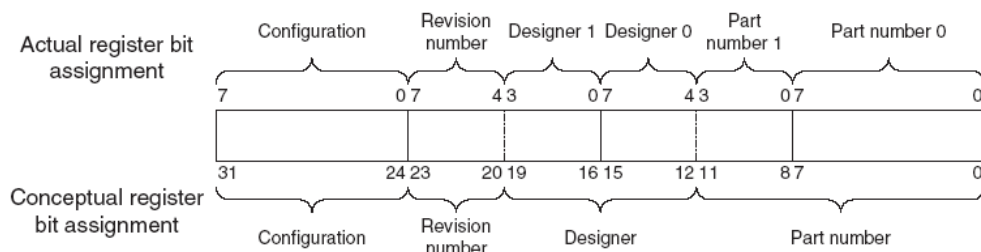


Рисунок 5.22 – Формат регистра TimerPeriphID0-3

					ЮФКВ.431268.006РЭ			Лист
								102
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

Отметим, что при проектировании важно помнить, что периферийные блоки занимают 4 Кб адресного пространства. Обращение к периферийным регистрам должно быть 32-битным, независимо от их реальной разрядности.

5.8.2.9 PrimeCell идентификатор. TimerPCellID0-3

Регистры TimerPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, предназначенные только для чтения, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. Регистр TimerPCellID установлен в 0xB105F00D. На рисунке ниже приведен формат этого регистра.

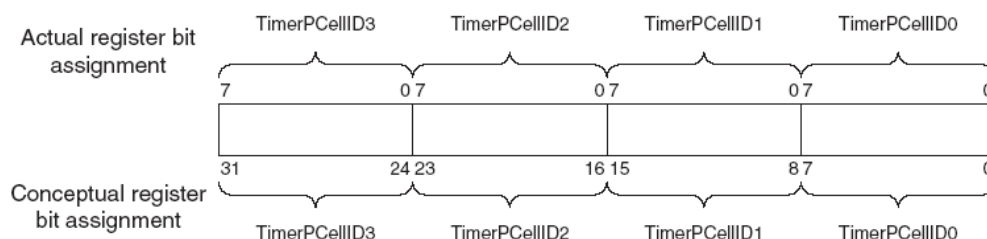


Рисунок 5.23 – Формат регистра TimerPCellID0-3

5.8.2.10 Регистр управления тестом интеграции. TimerITCR

Этот одноканальный регистр используется для активации тестового режима проверки правильности интеграции периферийного блока в систему – далее «теста интеграции». Когда используется этот режим, маскированные сигналы прерывания напрямую контролируются регистром TimerITOP. Формат регистра приведен в таблице ниже.

Таблица 5.55. Формат регистра TimerITCR

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервированы, чтение не определено, должны быть записаны нули
[0]	ITEN	ЧТ/ЗП	Разрешение режима «теста интеграции». Когда этот бит установлен в 1, DIT переходит в режим теста интеграции, иначе находится в нормальном режиме.

5.8.2.11 Регистр управления выходными сигналами. TimerITOP

Когда используется режим теста интеграции, выходные сигналы прерывания непосредственно управляются битами этого регистра. Комбинированное прерывание TIMERINTC формируется по логическому «ИЛИ» между битами, установленными в регистре TimerITOP. Формат регистра приведен в таблице ниже.

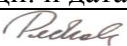
Таблица 3.15 – Формат регистра TimerITOP

					ЮФКВ.431268.006РЭ			Лист
								103
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

Биты	Название	Тип	Выполняемая функция
[31:2]	-	-	Зарезервированы, чтение не определено, должны быть записаны нули
[0]	TIMERINT2	ЗП	Значение с выхода TIMERINT2, в режиме теста интеграции
[1]	TIMERINT1	ЗП	Значение с выхода TIMERINT1, в режиме теста интеграции

5.8.3 Прерывания

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

					ЮФКВ.431268.006РЭ			Лист
								104
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.9 Сторожевой таймер WDOG

Блок сторожевого таймера в СБИС СПОВС имеет следующие характеристики:

- 32– разрядный счетчик, с программируемым временем срабатывания
- счетчик таймера работает на частоте до 41 МГц
- выход прерывания
- подключение к системной шине СБИС через AMBA APB интерфейс, используются полный, 32 разрядный интерфейс шины данных

5.9.1 Устройство блока сторожевого таймера

Блок не имеет сигналов, соединенных с внешними выводами микросхемы.

На рисунке ниже приведена структурная схема блока:

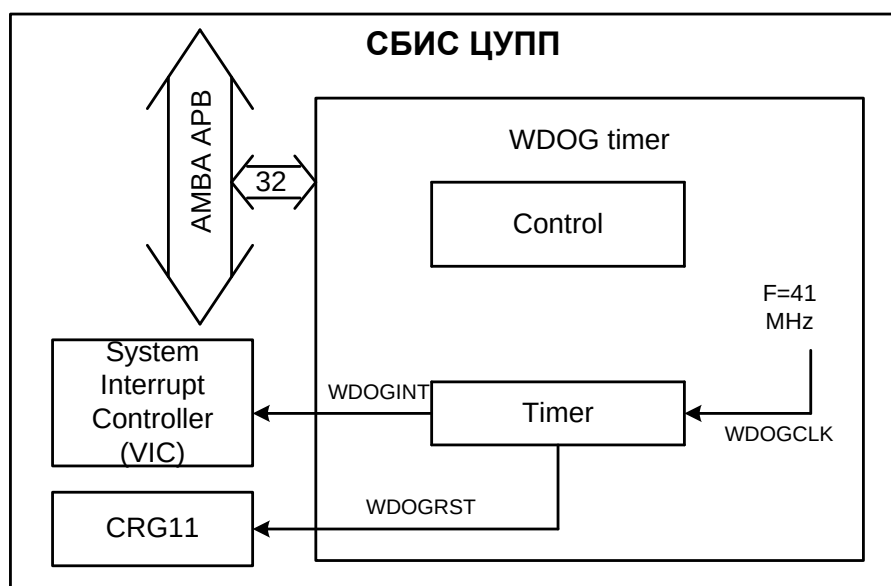


Рисунок 5.24 – Схема блока сторожевого таймера

Блок состоит из программируемого 32-разрядного таймера, работающего на частоте 40 МГц, а также схемы управления.

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

1. Программным образом задается интервал счета таймера, после чего счетчик начинает декрементироваться.
2. Когда счетчик достигает нулевого значения, генерируется сигнал прерывания WDOGINT, а счетчик перезаписывается начальным значением. Если до достижения счетчиком нулевого значения, прерывание не будет обработано (сброшено), то будет сформирован сигнал WDOGRST, который, если это разрешено в настройках CRG11, приведет к сбросу СБИС.
3. Для исключения случайного сброса СБИС вследствие неправильной записи в регистры блока сторожевого таймера, предусмотрена возможность программно заблокировать эти регистры для записи.

					ЮФКВ.431268.006РЭ				Лист
									105
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	Решет 16.01.14		22755-1						

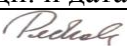
5.9.2 Регистровая модель блока сторожевого таймера

Программно доступные регистры блока сторожевого таймера WDOG расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение WDOG Base = 0xFFFF89000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.56. Спецификация регистров блока WDOG

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

Далее приведено подробное описание каждого регистра.

					ЮФКВ.431268.006РЭ			Лист
								106
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

5.9.2.1 Регистр загрузки. WdogLoad

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

5.9.2.2 Регистр значения. WdogValue

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

5.9.2.3 Регистр управления. WdogControl

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

Таблица 5.57. Формат регистра WdogControl

Биты	Название	Тип	Выполняемая функция
[31:2]	-	-	Зарезервировано
[1]	RESEN	ЧТ/ЗП	Разрешающий сигнал блока WDOG для сброса выхода, WDOGRST. Работает как маска для сброса выхода. Если установлен высокий уровень, то разрешен сброс, если низкий, то сброс неактивен.
[0]	INTEN	ЧТ/ЗП	Сигнал, разрешающий прерывание события, WDOGINTE. Если установлен высокий уровень, то счетчик находится в активном состоянии и разрешены прерывания, если установлен низкий уровень, то счетчик и прерывания неактивны. Нужно перезагрузить счетчик значением, которое хранится в WdogLoad, если прерывание ранее было неактивным, но затем стало разрешенным.

5.9.2.4 Регистр снятия прерываний. WdogIntClr

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

5.9.2.5 Регистр состояния прерывания до наложения маски. WdogRIS

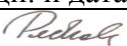
Данный регистр определяет состояние прерываний до наложения маски, формируемых счетчиком. Регистр WdogRIS возводится в единицу в момент, когда счетчик сторожевого таймера достигает нуля. В таблице ниже приведен формат регистра WdogRIS.

Таблица 5.58. Формат регистра WdogRIS

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

5.9.2.6 Регистр состояния прерывания после наложения маски. WdogMIS

Данный регистр определяет состояние прерывания счетчика после наложения аннки. Это значение представляется логическим элементом «И» между битом WDO-

					ЮФКВ.431268.006РЭ			Лист
								107
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

GRIS и битом INTEN регистра управления, и представляет аналогичное значение выходного сигнала WDOGINT. В таблице ниже приведен формат регистра WdogMIS.

Таблица 5.59. Формат регистра WdogMIS

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

5.9.2.7 Регистр блокировки. WdogLock

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

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

В таблице ниже приведен формат регистра WdogLock.

Таблица 5.60. Формат регистра WdogLock

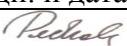
Биты	Название	Тип	Выполняемая функция
[31:0]	WDOG LOCK	ЧТ/ЗП	Запись значения 0x1ACCE551 в этот регистр разрешает запись во все регистры. Запись любого другого значения делает невозможным доступ по записи ко всем регистрам. Чтение возвращает состояние блокировки: 0x00000000 – запись во все регистры разрешена 0x00000001 – запись во все регистры запрещена

5.9.2.8 Идентификатор периферийного устройства. WdogPeriphID0-3

Регистр TimerPeriphID0-3 представляет четыре 8-битных регистра, предназначенных только для чтения, которые охватывают адресное пространство от 0xFE0 до 0xFEC. Регистры могут быть концептуально представлены как 32-битный регистр. В таблице ниже приведен формат этого регистра.

Таблица 5.61. Формат регистра WdogPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для DIT 0x805
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока. Номер ревизии начинается с нуля
Configuration[31:24]	Является вариантом конфигурацией периферии. Для DIT 0x0

					ЮФКВ.431268.006РЭ			Лист
								108
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

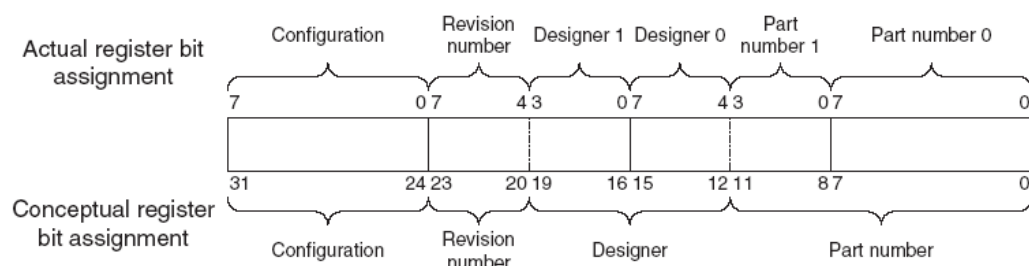


Рисунок 5.25 – Формат регистра WdogPeriphID0-3

Отметим, что при проектировании важно помнить, что периферийные блоки занимают 4 Кб адресного пространства. Обращение к периферийным регистрам должно быть 32-битным, независимо от их реальной разрядности.

5.9.2.9 PrimeCell идентификатор. WdogPCellID0-3

Регистры WdogPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, программно доступные только по чтению, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. Регистр WdogPCellID установлен в 0xB105F00D. На рисунке ниже приведен формат этого регистра.



Рисунок 5.26 – Формат регистра WdogPCellID0-3

5.9.2.10 Регистр управления тестом интеграции. WdogITCR

Этот одноканальный регистр используется для активации тестового режима проверки правильности интеграции периферийного блока в систему – далее «теста интеграции». Когда используется этот режим, прерывание WDOGINT и сигнал сброса WDOGRST контролируются регистром WdogITOP. Формат регистра приведен в таблице ниже.

Таблица 5.62. Формат регистра WdogITCR

Биты	Название	Тип	Выполняемая функция
[31:1]	-	-	Зарезервированы
[0]	ITEN	ЧТ/ЗП	Разрешение режима «теста интеграции». Когда этот бит установлен в 1, WDOG переходит в режим теста интеграции, иначе находится в нормальном режиме.

					ЮФКВ.431268.006РЭ			Лист
								109
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редкал</i> 16.01.14		22755-1					

5.9.2.11 Регистр управления выходными сигналами. WdogITOP

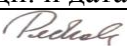
Когда используется режим теста интеграции, выходные сигналы прерывания непосредственно управляются битами этого регистра. Формат регистра приведен в таблице ниже.

Таблица 5.63. Формат регистра WdogITOP

Биты	Название	Тип	Выполняемая функция
[31:2]	-	-	Зарезервированы, чтение не определено, должны быть записаны нули
[0]	WDOGINT	ЗП	Значение выхода WDOGINT, в режиме теста интеграции
[1]	WDOGRST	ЗП	Значение выхода WDOGRST, в режиме теста интеграции

5.9.3 Прерывания

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

					ЮФКВ.431268.006РЭ			Лист
								110
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.10 Таймер реального времени RTC

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

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

5.10.1 Устройство блока таймера реального времени

Блок RTC состоит из двух основных частей: блока интерфейса и блока счётчиков (см. Рисунок 5.27).

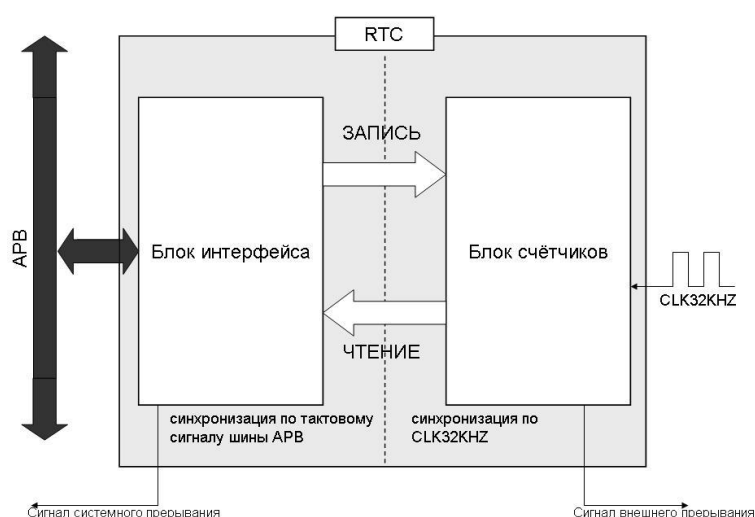


Рисунок 5.27 – Структурная схема таймера реального времени

5.10.2 Спецификация регистров.

Спецификация регистров RTC представлена в таблице ниже (Таблица 5.64).

					ЮФКВ.431268.006РЭ				Лист
									111
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	<i>Редко</i> 16.01.14		22755-1						

Таблица 5.64 – Спецификация регистров RTC

Адрес	Название регистра	Значение после сброса	Доступ	Описание
Base + 0x00h	TIMEREG	06000000h	Чтение (всегда) и запись (только когда бит TR регистра RTCFR равен 1).	Регистр реального времени
Base + 0x04h	DATEREG	14000101h	Чтение (всегда) и запись (только когда бит DR регистра RTCFR равен 1).	Регистр даты
Base + 0x08h	ALARMREG	01000000h	Чтение (всегда) и запись (только когда бит AR регистра RTCFR равен 1).	Регистр будильника
Base + 0x0Ch	RTCCR	00000000h	Чтение (всегда) и запись (только когда бит CR регистра RTCFR равен 1).	Управляющий регистр
Base + 0x10h	RTCFR	000000Fh	Чтение (всегда).	Регистр флагов

5.10.3 Описание регистров

5.10.3.1 Сброс регистров RTC.

Сброс регистров RTC может быть произведен двумя способами.

При включении питания регистры устройства сбрасываются (аппаратный сброс).

Возможен также программный сброс устройства, который происходит при записи в бит RS регистра RTCCR значения 0.

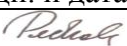
Поскольку регистры устройства тактируются тактовым сигналом с частотой 32768 Гц, сброс устройства может производиться в течение 60 мкс (2 такта тактового сигнала).

Если устройство находится в состоянии сброса, бит RA регистра RTCFR имеет значение 1, если устройство не находится в состоянии сброса, бит RA регистра RTCFR имеет значение 0.

5.10.3.2 Общие правила записи в регистры устройства.

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

После аппаратного сброса все регистры устройства, кроме RTCFR, доступны для записи. После программного сброса для записи доступны все регистры, кроме RTCFR (RTCFR всегда доступен только для чтения) и RTCCR (на время до 5 тактов тактового сигнала с частотой 32768 Гц).

					ЮФКВ.431268.006РЭ				Лист
									112
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

Для того чтобы программно сбросить устройство, нужно по адресу регистра RTCCR записать слово, в котором 6-й бит (RS) равен 0, а в остальных битах – значения, которые требуется установить после сброса.

Регистр недоступен для записи в период времени между обращением по адресу этого регистра по шине APB и установлением в регистре записываемого значения. В течение этого периода времени соответствующий этому регистру флаг готовности в регистре RTCFR имеет значение 0. Если непосредственно перед записью регистра не производился сброс устройства, то период неготовности регистра для записи может составлять от 15 до 46 мкс (0,5 – 1,5 такта тактового сигнала с частотой 32768 Гц).

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

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

Корректные значения секунд: от 0 до 59.

Корректные значения минут: от 0 до 59.

Корректные значения часов: от 0 до 23.

Корректные значения кодов дней недели: от 1 до 7.

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

Корректные значения кодов месяцев: от 1 до 12.

Корректные значения лет: от 0 до 99.

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

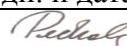
Поля в регистрах устройства имеют разные режимы доступа, обозначаемые условно, как на рисунке ниже (см. Рисунок 5.28).

-  - Бит ЗАРЕЗЕРВИРОВАН (при чтении возвращается 0, доступен только для чтения)
-  - Бит доступен только для чтения
-  - Бит доступен и для записи и для чтения

Рисунок 5.28 — Условные обозначения режимов доступа

5.10.3.3 Регистр реального времени TIMEREG.

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

					ЮФКВ.431268.006РЭ				Лист
									113
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

Регистр	TIMEREG																															
№№ бит	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Поля	0	0	0	0	0	ДЕНЬ НЕДЕЛИ			0	0	0	ЧАСЫ				0	0	МИНУТЫ				0	0	СЕКУНДЫ								
Доступ	R					RW			R			RW				R		RW				R		RW								

Рисунок 5.29 — Регистр реального времени

Таблица 5.65 — Структура регистра TIMEREG

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

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

5.10.3.4 Регистр даты DATEREG.

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

Регистр	DATEREG																																	
№№ бит	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Поля	0	0	0	СТОЛЕТИЕ					0	ГОД							0	0	0	0	МЕСЯЦ				0	0	0	ЧИСЛО						
Доступ	R			RW					R	RW							R				RW				R			RW						

Рисунок 5.30 — Регистр даты

					ЮФКВ.431268.006РЭ	Лист 114		
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14			22755-1			

Таблица 5.66 — Структура регистра DATEREG

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

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

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

Регистр	DATEREG																															
№№ бит	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Значение	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0

Рисунок 5.31 — Пример кодирования даты

5.10.3.5 Регистр будильника *ALARMREG*.

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

Регистр	ALARMREG																															
№№ бит	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Поля	0	0	0	0	0	ДЕНЬ НЕДЕЛИ			0	0	0	ЧАСЫ			0	0	МИНУТЫ			0	0	СЕКУНДЫ										
Доступ	R					RW			R			RW			R		RW			R		RW										

Рисунок 5.32 — Регистр будильника

					ЮФКВ.431268.006РЭ	Лист	
						115	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв. № подл.		Подп. и дата			Взам. инв. №	Инв. № дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

Таблица 5.67 — Структура регистра ALARMREG

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

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

5.10.3.6 Управляющий регистр.

Регистр	RTCCR																															
№№ бит	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Поля	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CEN	BPRS	0	EIE	T	0	IAW	IAE
Доступ	R																								RW	RW	R	RW	RW	R	RW	RW

Рисунок 5.33 — Управляющий регистр

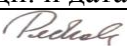
					ЮФКВ.431268.006РЭ					Лист
										116
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата				Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2	 16.01.14				22755-1					

Таблица 5.68 — Структура управляющего регистра

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

CEN сбрасывается в 0 через некоторое время после попытки записи в какой-либо из регистров устройства некорректных данных (этот интервал, как и для всех регистров устройства, составляет от 15 до 46 мкс). При этом в бит CEN не может быть записано значение 1, пока хотя бы в одном из бит TW, DW или AW сохраняется значение 1.

После аппаратного сброса в битах IAE, IAW, T, EIE, CEN устанавливаются значения 0.

При записи в регистр RTCCR слова, в котором 6-й бит (RS) имеет значение 0, происходит сброс устройства, затем бит RS принимает значение 1, а значения других бит записываемого слова устанавливаются в соответствующие значимые биты RTCCR.

5.10.3.7 Регистр флагов.

Регистр	RTCFR																							
№№ бит	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Поля	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Доступ	R																							

Рисунок 5.34 — Регистр флагов

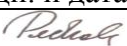
					ЮФКВ.431268.006РЭ				Лист
									117
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата				Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14				22755-1				

Таблица 5.69 — Структура регистра флагов

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

После аппаратного сброса устройства регистр RTCFR имеет значение 0000000Fh.

Программный сброс устанавливает в регистр RTCFR значение 0000000Fh.

5.10.4 Прерывания

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

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

1) Бит IAE=1.

Если бит IAW=1, то

2) Регистры TIMEREG и ALARMREG полностью равны.

Если бит IAW=0, то

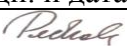
2) Поля СЕКУНДЫ, МИНУТЫ и ЧАСЫ регистров TIMEREG и ALARMREG равны.

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

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

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

1) Бит EIE=1.

					ЮФКВ.431268.006РЭ			Лист
								118
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Если бит IAW=1, то

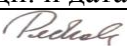
2) Регистры TIMEREG и ALARMREG полностью равны.

Если бит IAW=0, то

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

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

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

					ЮФКВ.431268.006РЭ			Лист
								119
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.11 Системный контроллер ARMSC

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

- Регистр управления отображением памяти – управляет отражением ПЗУ или ОЗУ по начальному адресу памяти
- Регистр управления местом расположения таблицы векторов прерываний ARM (управление режимом High Vectors Mode)
- Регистры состояния входов – показывают состояние входов МС, управляющих источником начальной загрузки, а также состоянием тестовых входов
- Регистры управления DDR – набор управляющих регистров и регистров состояния, управляющих начальной инициализацией DDR ДМС контроллера внешней памяти
- Регистры управления АЦП – позволяют управлять режимом работы встроенного АЦП
- Регистры управления прерываниями НМС части
- Регистр приоритета ядер NMU
- Регистры управления USB блоком управляют начальной инициализацией блока USB

5.11.1 Устройство системного контроллера ARMSC

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

Таблица 5.70. Выводы микросхемы, относящиеся контроллеру ARMSC

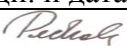
Вывод	Тип буфера	Примечание
VINITH	in	вход управления режимом High Vectors Mode
BOOTM[2:1]	in	входы управления источником начальной загрузки
TSTMODE	in	вход управления тестовым режимом
TST_I[2:0]	in	3 бита тестовых входов
USBCON	in	вход индикации состояния подключения к шине USB
TD[9] (USB-PUCTRLn)	out	выход управления ключом подтягивающего резистора на линии D+ USB

На рисунке (см. Рисунок 5.35) приведена схема включения системного ядра ARMSC.

Подробное описание работы с регистрами смотри в описании соответствующих блоков СБИС СПОВС.

5.11.1.1 Управление инициализацией контроллера внешней памяти ДМС

Для правильной работы контроллера внешней памяти ДМС необходимо провести процедуру его начальной инициализации. Инициализация ДМС состоит из двух частей: инициализации физического интерфейса ДМС и инициализации логического

					ЮФКВ.431268.006РЭ			Лист
								120
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

интерфейса. Ниже приведен код инициализации физического интерфейса на языке ассемблера ARM.

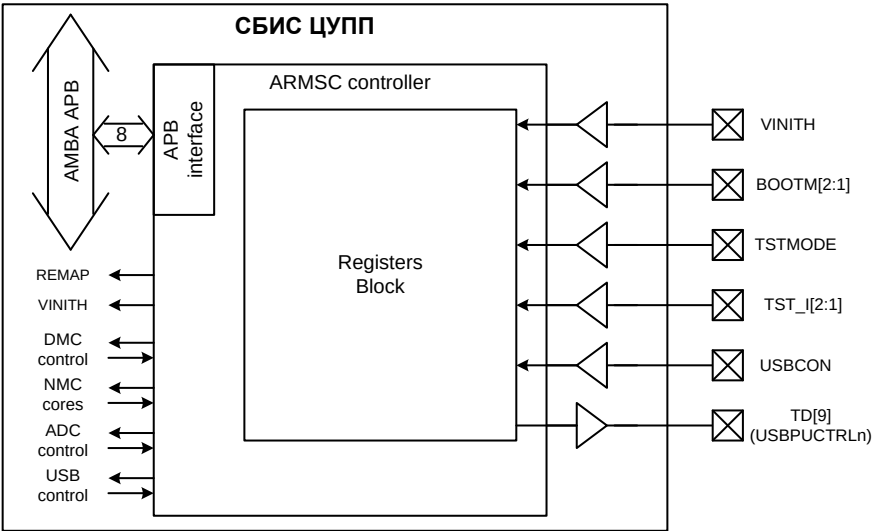


Рисунок 5.35 – Схема системного контроллера ARMSC

```
; После снятия сигнала внешнего сброса происходит
; инициализация физического интерфейса по-умолчанию
; (поля регистра DFICONFIG прописываются нулями) .
; Для правильной работы физического интерфейса DMC
; необходимо прописать следующие значения в поля регистра DFICONFIG
; cval[1:0]=01(800ps); dllmode[2:0]=001(8192cyc)
; idegsl=0 ; isft[2:0]=110(67.5) ; iudset[3:0]=0010(N32)
    LDR R2, =0x0962
    STR R2, [R3, #ARMSC_DFICONFIG]

; После изменения полей регистра DFICONFIG необходимо
; программно сбросить контроллер DMC, для этого
; необходимо установить младший бит регистра DFIRST
; (DMC контроллер сбрасывается полностью, т.е.
; сбрасываются все регистры DMC контроллера)
    LDR R2, =0x0001
    STR R2, [R3, #ARMSC_DFIRST]

; Здесь происходит сброс программный сброс контроллера DMC.
; Контроллера DMC готов к работе если бит dfi_init_ok
; регистра DFISTA равен 1.
; Пока бит dfi_init_ok равен нулю, контроллер DMC не
; инициализирован и доступ к его регистрам запрещен.
    LDR R3, =ARMSC_Base
    23 LDR R2, [R3, #ARMSC_DFISTA]
    TST R2, #1
    BEQ %B11

; Инициализация физического интерфейса DMC закончена.
; Далее необходимо провести инициализацию логического
; интерфейса контроллера DMC
```

					ЮФКВ.431268.006РЭ			Лист
								121
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редько</i> 16.01.14		22755-1				

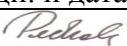
5.11.2 Регистровая модель контроллера ARMSC

Программно доступные регистры контроллера ARMSC расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение ARMSC Base = 0xFFF81000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.71. Спецификация регистров контроллера ARMSC

Адрес	Тип доступа	Разрядность	Начальное значение	Имя регистра	Описание
ARMSC Base + 0x00	ЧТ	1	b0	REMAPSTA	Текущий тип отображения памяти
ARMSC Base + 0x04	ЗП1	1	-	REMAPSET	Установить REMAPSTA
ARMSC Base + 0x08	ЗП1	1	-	REMAPCLR	Сбросить REMAPSTA
ARMSC Base + 0x0C	-	-	-	-	зарезервировано
ARMSC Base + 0x10	ЧТ	1	VINITH вход	VIHSTA	Регистр управления режимом High Vectors Mode
ARMSC Base + 0x14	ЗП1	1	-	VIHSET	Установить VIHSTA
ARMSC Base + 0x18	ЗП1	1	-	VIHCLR	Сбросить VIHSTA
ARMSC Base + 0x1C	ЧТ	6	значения на входах	BOOTM	Регистр состояния внешних входов, управляющих загрузкой
ARMSC Base + 0x20	ЧТ	1	0	DFISTA	Регистр готовности физического интерфейса DMC
ARMSC Base + 0x24	ЧТ/ЗП	13	0x00000	DFICONFIG	Регистр конфигурации физического интерфейса DMC
ARMSC Base + 0x28	ЗП1	1	-	DFIRST	Регистр программного сброса DMC
ARMSC Base + 0x2C	ЧТ/ЗП	11	0x0000	ADCCTRL	Регистр управления потреблением АЦП
ARMSC Base + 0x30	ЧТ	6	b000000	NMUINTSTA	Регистр состояния прерываний от NMU части СБИС
ARMSC Base + 0x34	ЗП1	6	-	NMUINTCLR	Регистр сброса прерываний от NMU части СБИС
ARMSC Base + 0x38	ЗП1	6	-	NMUINTREQ	Регистр генерации прерывания к NMU части
ARMSC Base + 0x3C	ЧТ/ЗП	1	b0	NMUCTRL	Регистр управления приоритетом ядер NMC доступа в SMU
ARMSC Base + 0x40	ЗП1	1	-	USBRESET	Регистр программного сброса USB контроллера
ARMSC Base + 0x44	ЧТ/ЗП	2	b00	USBCTRL	Регистр управления подключением к USB шине
ARMSC Base + 0x48	ЧТ	3	значения на входах	USBSTATE	Регистр состояния USB шины
ARMSC Base + 0x4C	ЗП1	1	-	USBINTCLR	Регистр сброса прерывания по подключению к шине USB
ARMSC Base + 0x50 to 0xFFC	-	-	-	-	зарезервировано

Далее приведено подробное описание каждого регистра.

					ЮФКВ.431268.006РЭ			Лист
								122
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

5.11.2.1 Регистры управления отображением памяти. REMAPxxx

Контроллер ARMSC содержит регистры управления отображением банка ПЗУ на карту памяти. Подсистема ARMU имеет две карты памяти: одна – до инициализации (Before Remap), вторая – после (After Remap). Это позволяет по одним и тем же адресам обращаться к ПЗУ во время инициализации и к ОЗУ во время нормальной работы, когда ПЗУ уже не требуется (подробнее см. п. 3). Текущее отображение памяти определяется значением регистра REMAPSTA. Ниже приведен формат регистра REMAPSTA.

Таблица 5.72. Формат регистра REMAPSTA

Биты	Название	Тип	Выполняемая функция
[0]	REMAPSTA	ЧТ	Состояние карты памяти: 0 – Before Remap (ПЗУ по начальным адресам) 1 – After Remap (ОЗУ по начальным адресам)

Смена отображения возможна программным образом, путем записи регистров REMAPSET и REMAPCLR. Запись 1 в регистр REMAPSET устанавливает регистр REMAPSTA, а запись 1 в регистр REMAPCLR сбрасывает его.

5.11.2.2 Регистры управления режимом High Vectors Mode. VINxxx

Таблица векторов прерываний в системе ARMU может располагаться по одному из двух адресов: 0x00000000 (Normal mode) или 0xFFFF0000 (High Vectors Mode). Управление расположением таблицы осуществляется с помощью внешнего входа СБИС – VINITH. Значение с входа VINITH используется для инициализации внутреннего регистра контроллера (VINSTA) во время подачи внешнего сигнала сброса. Ниже приведен формат регистра VINSTA.

Таблица 5.73. Формат регистра VINSTA

Биты	Название	Тип	Выполняемая функция
[0]	VINSTA	ЧТ	Регистр управления режимом High Vectors Mode 0 – Normal Mode (вектора прерываний по адресу 0x00000000) 1 – High Vectors Mode (вектора прерываний по адресу 0xFFFF0000)

После того как внешний сигнал сброса снят, значение на входе VINITH не влияет на расположение таблицы векторов прерываний. После окончания процедуры начальной инициализации возможно программным образом менять положение таблицы прерываний в системе путем записи в регистры VINCLR или VINSET. Запись 1 в регистр VINSET устанавливает регистр VINSTA, а запись 1 в регистр VINCLR сбрасывает его.

5.11.2.3 Регистр состояния внешних входов, управляющих загрузкой. BOOTM

Регистр BOOTM используется для контроля значений на следующих внешних входах СБИС: TST_I[2:0], BOOTM[2:1] и TSTMODE. Программа начального загрузчика использует часть этих регистров для определения источника начальной загрузки и режима работы (см. п. 5.17.2). После окончания работы начального загрузчика входы TST_I и BOOTM могут использоваться как входы общего назначения. Вход TSTMODE должен быть равен нулю всегда, т.к. изменение этого сигнала на единицу приведет к сбою в работе СБИС СПОВС.

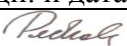
					ЮФКВ.431268.006РЭ			Лист
								123
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.74. Формат регистра BOOTM

Биты	Название	Тип	Выполняемая функция
[5:3]	TST_I[2:0]	ЧТ	Значение на входах TST_I[2:0]
[2]	TSTMODE	ЧТ	Значение на входе TSTMODE
[1:0]	BOOTM[2:1]	ЧТ	Значение на входах BOOTM[2:1]

5.11.2.4 Регистр готовности физического интерфейса DMC. DFISTA

Младший бит регистра DFISTA – dfi_init_ok, показывает готовность физического интерфейса DMC контроллера. Пока не окончена инициализация физического интерфейса DMC, запрещен доступ к регистрам DMC контроллера. В таблице ниже приведен формат регистра.

Таблица 5.75. Формат регистра DFISTA

Биты	Название	Тип	Выполняемая функция
[0]	dfi_init_ok	ЧТ	Состояние физического интерфейса DMC 0 – инициализация не закончена 1 – инициализация завершена

5.11.2.5 Регистр конфигурации физического интерфейса DMC. DFICONFIG

Регистр DFICONFIG предназначен для настройки параметров физического интерфейса DMC контроллера. Для правильной работы должен быть инициализирован значением 0x0962. После любого изменения этого регистра необходимо проводить программный сброс контроллера. В таблице ниже приведен формат регистра.

Таблица 5.76. Формат регистра DFICONFIG

Биты	Название	Тип	Выполняемая функция
[12:11]	cval[1:0]	ЧТ/ЗП	Задержка кольцевого буфера в физическом интерфейсе DMC. Необходимо записать значение b'01 во время инициализации.
[10:8]	dllmode	ЧТ/ЗП	Временной параметр DLL физического интерфейса DMC. Необходимо записать значение b'001 во время инициализации.
[7]	idegsel	ЧТ/ЗП	Временной параметр DLL физического интерфейса DMC. Необходимо записать значение b'0 во время инициализации.
[6:4]	isft	ЧТ/ЗП	Временной параметр DLL физического интерфейса DMC. Необходимо записать значение b'110 во время инициализации.
[3:0]	iudset	ЧТ/ЗП	Временной параметр DLL физического интерфейса DMC. Необходимо записать значение b'0010 во время инициализации.

5.11.2.6 Регистр программного сброса DMC. DFIRST

Регистр DFIRST предназначен для программного сброса DMC контроллера, включая физический интерфейс. Запись единицы в младший бит регистра приведет к сбросу контроллера. Сброс необходимо проводить после любого изменения полей регистра DFICONFIG.

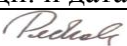
					ЮФКВ.431268.006РЭ			Лист
								124
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.77. Формат регистра DFIRST

Биты	Название	Тип	Выполняемая функция
[0]	dfi_rst	ЗП1	Программный сброс DMC контроллера. Запись 1 приведет к сбросу.

5.11.2.7 Управление режимом работы встроенных АЦП. ADCCTRL

Биты регистра ADCCTRL используются для управления потреблением и работой встроенных АЦП (см. п. «9 Блок аналогово-цифрового преобразования (ADCU)»). В таблице ниже приведен формат регистра.

Таблица 5.78. Формат регистра ADCCTRL

Биты	Название	Тип	Выполняемая функция
[10:7]	HALFPWR_CH[3:0]	ЧТ/ ЗП	Поле позволяет перевести АЦП в режим половинной мощности. В этом режиме частота работы АЦП снижается в 2 раза. HALFPWR_CH[n] = 0 – ADCn работает в нормальном режиме HALFPWR_CH[n] = 1 – ADCn работает в режиме половинной мощности где n=0...3 номер АЦП
[6:3]	PDNPIPE_CH[3:0]	ЧТ/ ЗП	Поле позволяет отключить конвейер каждого АЦП. PDNPIPE CH[n] = 1 конвейер ADCn выключен, PDNPIPE CH[n] = 0 конвейер ADCn включен, где n=0...3 номер АЦП
[2]	PDN_A	ЧТ/ ЗП	Поле позволяет отключить подачу питания на пару АЦП ADC1,ADC0. 0 – питание подается на пару АЦП ADC1,ADC0; 1 – питание не подается на пару АЦП ADC1,ADC0;
[1]	PDN_B	ЧТ/ ЗП	Поле позволяет отключить подачу питания на пару АЦП ADC2,ADC3. 0 – питание подается на пару АЦП ADC2,ADC3; 1 – питание не подается на пару АЦП ADC2,ADC3;
[0]	PDNCLK_B	ЧТ/ ЗП	Поле позволяет отключить подачу тактового сигнала на пару АЦП ADC2,ADC3. 0 – тактовый сигнал подается на пару АЦП ADC2,ADC3; 1 – тактовый сигнал не подается на пару АЦП ADC2,ADC3;

5.11.2.8 Регистр состояния прерываний от NMU части СБИС. NMUINTSTA

Регистр NMUINTSTA предназначен для определения состояния прерываний NMC0NP, NMC1NP, ITMS, ITAUX, NMC0LP и NMC1LP. В таблице ниже приведен формат регистра.

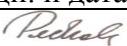
					ЮФКВ.431268.006РЭ			Лист
								125
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.79. Формат регистра NMUINTSTA

Биты	Название	Тип	Выполняемая функция
[5]	ITAUX	ЧТ	Состояние прерывания ITAUX 0 – прерывание не активно 1 – прерывание активно
[4]	ITMS	ЧТ	Состояние прерывания ITMS 0 – прерывание не активно 1 – прерывание активно
[3]	NMC1HP	ЧТ	Состояние прерывания NMC1HP 0 – прерывание не активно 1 – прерывание активно
[2]	NMC0HP	ЧТ	Состояние прерывания NMC0HP 0 – прерывание не активно 1 – прерывание активно
[1]	NMC1LP	ЧТ	Состояние прерывания NMC1LP 0 – прерывание не активно 1 – прерывание активно
[0]	NMC0LP	ЧТ	Состояние прерывания NMC0LP 0 – прерывание не активно 1 – прерывание активно

5.11.2.9 Регистр сброса прерываний от NMU части СБИС. NMUINTCLR

Регистр NMUINTCLR предназначен для сброса прерываний NMC0HP, NMC1HP, ITMS, ITAUX, NMC0LP и NMC1LP от NMU части. В таблице ниже приведен формат регистра.

Таблица 5.80. Формат регистра NMUINTCLR

Биты	Название	Тип	Выполняемая функция
[5]	ITAUXCLR	ЗП1	Сброс прерывания ITAUX. Запись 1 сбрасывает прерывание.
[4]	ITMSCLR	ЗП1	Сброс прерывания ITMS. Запись 1 сбрасывает прерывание.
[3]	NMC1HPCLR	ЗП1	Сброс прерывания NMC1HP. Запись 1 сбрасывает прерывание.
[2]	NMC0HPCLR	ЗП1	Сброс прерывания NMC0HP. Запись 1 сбрасывает прерывание.
[1]	NMC1LPCLR	ЗП1	Сброс прерывания NMC1LP. Запись 1 сбрасывает прерывание.
[0]	NMC0LPCLR	ЗП1	Сброс прерывания NMC0LP. Запись 1 сбрасывает прерывание.

5.11.2.10 Регистр генерации прерывания к NMU части. NMUINTREQ

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

Таблица 5.81. Формат регистра NMUINTREQ

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

5.11.2.11 Регистр приоритета ядер NMC. NMUCTRL

Младший бит регистра NMUCTRL задает приоритет запросов ядер NMC0 и NMC1 к системе общей памяти SMU (см. п. 6). В таблице ниже приведен формат регистра.

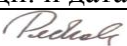
					ЮФКВ.431268.006РЭ			Лист
								126
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.82. Формат регистра NMUCTRL

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

5.11.2.12 Регистр программного сброса USB контроллера. USBRESET

Регистр предназначен для программного сброса контроллера USB. При записи 1 в бит USBRESET производится сброс контроллера USB, а также происходит сброс бит USBCONINTEN, USBPUCTRL, USBCONCHG, USBRDY. Состояние контроллера USB после программного сброса полностью идентично его начальному состоянию после аппаратного сброса всей СБИС. В таблице ниже приведен формат регистра.

Таблица 5.83. Формат регистра USBRESET

Биты	Название	Тип	Выполняемая функция
[0]	USBRESET	ЗП1	Программный сброс USB контроллера. Запись 1 приведет к сбросу.

5.11.2.13 Регистр управления подключения к USB шине. USBCTRL

Реализованная в системном контроллере ARMSC схема позволяет программно управлять подключением и отключением контроллера USB от шины USB. Подробное описание порядка работы см. в описании контроллера USB (п. 5.15). В таблице ниже приведен формат регистра.

Таблица 5.84. Формат регистра USBCTRL

Биты	Название	Тип	Выполняемая функция
[1]	USBCONINTEN	ЧТ/ЗП	Маска прерывания USBCONINT (1 – прерывание разрешено, 0 – запрещено).
[0]	USBPUCTRL	ЧТ/ЗП1	Управление ключом подтягивающего (Pull-Up) резистора на линии D+ устройства (0 – “отключено”, 1 – “подключено”). Подтягивающий резистор с ключом должен находиться на печатной плате. Если USBPUCTRL в состоянии «подключено», хаб USB понимает, что СБИС подключена к его нисходящему выводу; если USBPUCTRL в состоянии «отключено», хаб не может обнаружить СБИС на своём нисходящем выводе. Возможен только установ – этого бита в «1», установ «0» производится аппаратно при сбросе всего контроллера (USBRESET или аппаратный сброс СБИС). При сбросе СБИС бит USBPUCTRL устанавливается в состояние «отключено». При любом изменении бита USBCON (т.е. при физическом подключении или отключении кабеля USB) бит USBPUCTRL устанавливается в состояние «отключено».

5.11.2.14 Регистр состояния USB шины. USBSTATE

Регистр состояния шины USB. В таблице ниже приведен формат регистра.

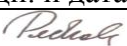
					ЮФКВ.431268.006РЭ			Лист
								127
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.85. Формат регистра USB

Биты	Название	Тип	Выполняемая функция
[2]	USBCONCHG	ЧТ	Состояние прерывания USBCONINT от блока USB, которое генерируется при каждом физическом подключении и отключении кабеля от хаба, т.е. отслеживается изменение бита USBCON. Прерывание USBCONINT может быть замаскировано с помощью бита USBCONINTEN. Сброс осуществляется путем записи бита USBCONINTCLR.
[1]	USBCON	ЧТ	Состояние физического подключения к хабу USB. (1 – есть подключение, 0 – нет подключения)
[0]	USBRDY	ЧТ	Бит готовности контроллера USB. После сброса при помощи бита USBRESET или после снятия сигнала аппаратного сброса контроллер некоторое время не готов к конфигурированию. Если бит USBRDY имеет значение «0», то нельзя производить запись или чтение регистров контроллера USB; если бит USBRDY имеет значение «1», контроллер USB готов к конфигурированию.

5.11.2.15 Регистр сброса прерывания по подключению к шине USB. USBINTCLR

Регистр сброса прерывания USBCONINT. В таблице ниже приведен формат регистра.

Таблица 5.86. Формат регистра USBINTCLR

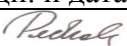
Биты	Название	Тип	Выполняемая функция
[0]	USBCONINTCLR	ЗП	Сброс прерывания USBCONINT. Запись 1 сбрасывает прерывание. Кроме того, бит USBCONCHG сбрасывается в состояние «0».

5.11.3 Прерывания

Контроллер ARMSC выполняет вспомогательную функцию для обработки прерываний от NMC части СБИС и прерывания по изменению состояния физического подключения к USB шине.

Прерывания от NMC части СБИС не могут быть подключены напрямую к системному контроллеру прерываний VIC, поэтому контроллер ARMSC выполняет функцию буферизации прерываний от NMC части, а также управляет сбросом этих прерываний. Всего к системному контроллеру прерываний VIC подключено 6 буферизованных в ARMSC прерываний от NMC части – NMC0HP, NMC1HP, ITMS, ITAUX, NMC0LP и NMC1LP.

Реализованная в системном контроллере ARMSC схема позволяет программно управлять подключением и отключением контроллера USB от шины USB. Эта схема Генерирует прерывание к системному контроллеру прерываний VIC при каждом подключении/отключении к шине USB.

					ЮФКВ.431268.006РЭ			Лист
								128
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.12 Контроллер ПДП DMAC

Контроллер прямого доступа в память (КПДП) в СБИС СПОВС имеет следующие характеристики:

- предназначен для передачи данных из периферийных устройств в память, и обратно
- КПДП имеет 8 независимых каналов передачи данных: 4 канала от периферийных устройств к памяти и 4 канала от памяти к периферийным устройствам
- КПДП со стороны памяти имеет 64-разрядный AMBA AXI интерфейс, работающий на частоте до 164 МГц, а со стороны периферийных устройств 32-разрядный AMBA AXI интерфейс, работающий на частоте до 82 МГц
- программирование КПДП производится через 32-разрядный AMBA APB интерфейс

5.12.1 Устройство контроллера ПДП

Блок не имеет сигналов, соединенных с внешними выводами микросхемы. На рисунке ниже приведена структурная схема блока:

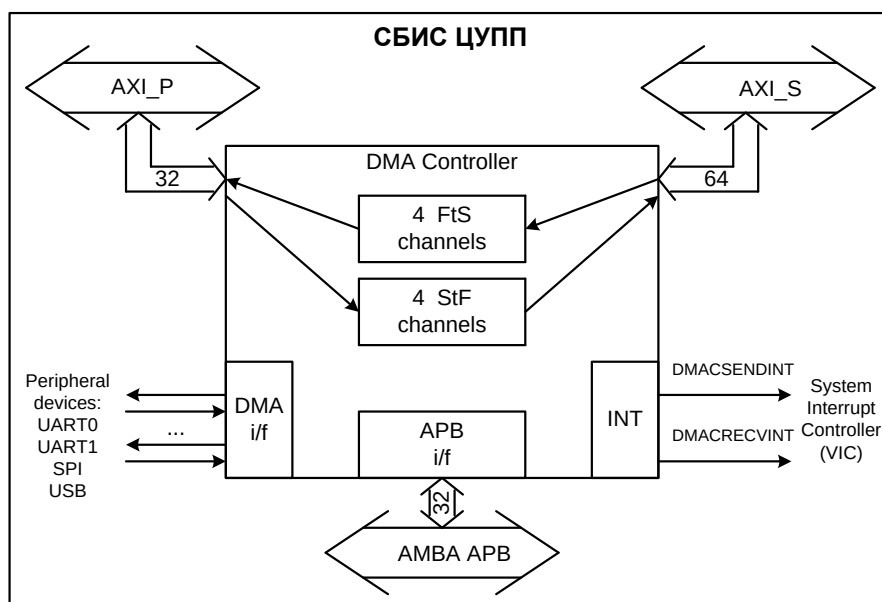


Рисунок 5.36 – Схема блока контроллера ПДП

Контроллер ПДП имеет 8 каналов. 4 из них предназначены для пересылок данных типа память-периферия (каналы FtS) и 4 – для пересылок типа периферия-память (StF). К каждому каналу относятся 8 программно доступных регистров. Программный доступ к этим регистрам осуществляется по шине APB.

К контроллеру, через интерфейс ПДП, подключены 4 периферийных устройства. Имеется следующее соответствие каналов контроллера ПДП и периферийных устройств:

- контроллер SPI подключен к каналам FtS1 и StF1
- контроллер USB подключен к каналам FtS2 и StF2
- контроллер UART0 подключен к каналам FtS3 и StF3

					ЮФКВ.431268.006РЭ			Лист
								129
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

- контроллер UART1 подключен к каналам FtS4 и StF4

Контроллер ПДП имеет два выхода прерывания, подключенных к системному контроллеру прерываний.

Для того чтобы контроллер ПДП начал пересылку данных, необходимо произвести инициализацию канала и установить младший бит регистра Enable. Инициализация канала состоит в задании адреса источника, адреса приемника, числа байтов, которые необходимо предать, типа упаковки/распаковки данных и типа сигнала ПДП интерфейса с периферийным устройством. Контроллеры UART интерфейсов имеют отличный от других периферийных блоков ПДП интерфейс, поэтому младший бит регистра PerHandsh_x каналов FtS3, StF3, FtS4 и StF4 должен быть установлен в единицу. Для остальных каналов этот бит должен быть сброшен.

Для каждого канала имеется регистр PackType_x, который фактически должен быть установлен значением, соответствующим разрядности данных, с которой может работать соответствующее периферийное устройство. Работа с памятью всегда происходит 64-разрядными словами. Периферийные устройства же могут работать 8-, 16- и 32-разрядными словами. Соответственно если периферийное устройство может работать лишь 8-разрядными словами, то одному доступу к памяти будет соответствовать 8 доступов к периферийному устройству. Поэтому в данном случае будет производиться следующая упаковка: 8 байт упаковываются в одно 64-разрядное слово, либо распаковка: 64-разрядное слово распаковывается в 8 байт (в зависимости от направления пересылки данных). Таким образом, для максимальной производительности необходимо установить следующий тип распаковки/упаковки для периферийных устройств:

- SPI – 16 бит
- USB – 16 бит
- UART0, UART1 – 8 бит

Установка неправильной разрядности приведет к пониженной производительности, либо к неправильному функционированию контроллера ПДП.

Для передачи данных необходимо задать начальный адрес передаваемого массива данных (SrcAdr_x), адрес куда данные будут передаваться (DstAdr_x), а также размер массива данных в байтах. Количество передаваемых байт данных устанавливается в регистре TrSize_x. После пересылки каналом заданного числа байтов автоматически устанавливается младший бит регистра состояния прерывания (Interrupt_x) и на системный контроллер прерываний выдается прерывание. Для того чтобы снова запустить канал, необходимо сбросить младший бит регистра прерывания Interrupt_x. Сигнал прерывания после этого сбрасывается автоматически.

Для программного останова канала необходимо сбросить младший бит регистра запуска/останова канала (Enable_x). При программном останове канала младший бит регистра прерывания (Interrupt_x) не устанавливается. Чтобы начать новую передачу после программного останова необходимо подождать пока регистр занятости канала (Busy_x) сбросится. Это произойдет сразу же, как только закончится передача текущего 64-битного слова.

Во время работы канала (Busy_x = 1) запись в регистры не будет иметь результата, за исключением записи нуля в регистр запуска/останова канала

5.12.2 Регистровая модель контроллера ПДП

Программно доступные регистры контроллера ПДП (DMAC) расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение DMAC Base = 0xFFFF84000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

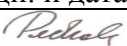
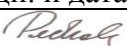
					ЮФКВ.431268.006РЭ			Лист
								130
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.87. Спецификация регистров контроллера ПДП

Адрес	Тип доступа	Разрядность	Начальное значение	Номер канала	Имя регистра	Описание
Base + 000h	R/W	32	0x00000000	FtS1	SrcAdr_FtS1	Адрес источника
Base + 004h	R/W	32	0x00000000		DstAdr_FtS1	Адрес приемника
Base + 008h	R/W	24	0x0000001		TrSize_FtS1	Число передаваемых байт
Base + 018h	R/W	2	2'b10		PackType_FtS1	Тип упаковки данных
Base + 01Ch	R/W	1	1'b0		PerHandsh_FtS_1	Тип ПДП интерфейса
Base + 00Ch	R/W	1	1'b0		Enable_FtS1	Запуск/останов канала
Base + 010h	R/W0	1	1'b0		Interrupt_FtS1	Состояние прерывания
Base + 014h	RO	1	1'b0		Busy_FtS1	Занятость канала
Base + 040h	R/W	32	0x00000000	FtS2	SrcAdr_FtS2	Адрес источника
Base + 044h	R/W	32	0x00000000		DstAdr_FtS2	Адрес приемника
Base + 048h	R/W	24	0x0000001		TrSize_FtS2	Число передаваемых байт
Base + 058h	R/W	2	2'b10		PackType_FtS2	Тип упаковки данных
Base + 05Ch	R/W	1	1'b0		PerHandsh_FtS_2	Тип ПДП интерфейса
Base + 04Ch	R/W	1	1'b0		Enable_FtS2	Запуск/останов канала
Base + 050h	R/W0	1	1'b0		Interrupt_FtS2	Состояние прерывания
Base + 054h	RO	1	1'b0		Busy_FtS2	Занятость канала
Base + 080h	R/W	32	0x00000000	FtS3	SrcAdr_FtS3	Адрес источника
Base + 084h	R/W	32	0x00000000		DstAdr_FtS3	Адрес приемника
Base + 088h	R/W	24	0x0000001		TrSize_FtS3	Число передаваемых байт
Base + 098h	R/W	2	2'b10		PackType_FtS3	Тип упаковки данных
Base + 09Ch	R/W	1	1'b0		PerHandsh_FtS_3	Тип ПДП интерфейса
Base + 08Ch	R/W	1	1'b0		Enable_FtS3	Запуск/останов канала
Base + 090h	R/W0	1	1'b0		Interrupt_FtS3	Состояние прерывания
Base + 094h	RO	1	1'b0		Busy_FtS3	Занятость канала
Base + 0C0h	R/W	32	0x00000000	FtS4	SrcAdr_FtS4	Адрес источника
Base + 0C4h	R/W	32	0x00000000		DstAdr_FtS4	Адрес приемника
Base + 0C8h	R/W	24	0x0000001		TrSize_FtS4	Число передаваемых байт
Base + 0D8h	R/W	2	2'b10		PackType_FtS4	Тип упаковки данных
Base + 0DCh	R/W	1	1'b0		PerHandsh_FtS_4	Тип ПДП интерфейса
Base + 0CCh	R/W	1	1'b0		Enable_FtS4	Запуск/останов канала
Base + 0D0h	R/W0	1	1'b0		Interrupt_FtS4	Состояние прерывания
Base + 0D4h	RO	1	1'b0		Busy_FtS4	Занятость канала
Base + 200h	R/W	32	0x00000000	StF1	SrcAdr_StF1	Адрес источника
Base + 204h	R/W	32	0x00000000		DstAdr_StF1	Адрес приемника
Base + 208h	R/W	24	0x0000001		TrSize_StF1	Число передаваемых байт
Base + 218h	R/W	2	2'b10		PackType_StF1	Тип распаковки данных
Base + 21Ch	R/W	1	1'b0		PerHandsh_StF_1	Тип ПДП интерфейса
Base + 20Ch	R/W	1	1'b0		Enable_StF1	Запуск/останов канала
Base + 210h	R/W0	1	1'b0		Interrupt_StF1	Состояние прерывания
Base + 214h	RO	1	1'b0		Busy_StF1	Занятость канала
Base + 240h	R/W	32	0x00000000	StF2	SrcAdr_StF2	Адрес источника
Base + 244h	R/W	32	0x00000000		DstAdr_StF2	Адрес приемника
Base + 248h	R/W	24	0x0000001		TrSize_StF2	Число передаваемых байт
Base + 258h	R/W	2	2'b10		PackType_StF2	Тип распаковки данных
Base + 25Ch	R/W	1	1'b0		PerHandsh_StF_2	Тип ПДП интерфейса
Base + 24Ch	R/W	1	1'b0		Enable_StF2	Запуск/останов канала
Base + 250h	R/W0	1	1'b0		Interrupt_StF2	Состояние прерывания
Base + 254h	RO	1	1'b0		Busy_StF2	Занятость канала

					ЮФКВ.431268.006РЭ			Лист
								131
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Base + 280h	R/W	32	0x00000000	StF3	SrcAdr_StF3	Адрес источника
Base + 284h	R/W	32	0x00000000		DstAdr_StF3	Адрес приемника
Base + 288h	R/W	24	0x0000001		TrSize_StF3	Число передаваемых байт
Base + 298h	R/W	2	2'b10		PackType_StF3	Тип распаковки данных
Base + 29Ch	R/W	1	1'b0		PerHandsh_StF_3	Тип ПДП интерфейса
Base + 28Ch	R/W	1	1'b0		Enable_StF3	Запуск/останов канала
Base + 290h	R/W0	1	1'b0		Interrupt_StF3	Состояние прерывания
Base + 294h	RO	1	1'b0		Busy_StF3	Занятость канала
Base + 2C0h	R/W	32	0x00000000	StF4	SrcAdr_StF4	Адрес источника
Base + 2C4h	R/W	32	0x00000000		DstAdr_StF4	Адрес приемника
Base + 2C8h	R/W	24	0x0000001		TrSize_StF4	Число передаваемых байт
Base + 2D8h	R/W	2	2'b10		PackType_StF4	Тип распаковки данных
Base + 2DCh	R/W	1	1'b0		PerHandsh_StF_4	Тип ПДП интерфейса
Base + 2CCh	R/W	1	1'b0		Enable_StF4	Запуск/останов канала
Base + 2D0h	R/W0	1	1'b0		Interrupt_StF4	Состояние прерывания
Base + 2D4h	RO	1	1'b0		Busy_StF4	Занятость канала

5.12.2.1 Регистр адреса источника каналов FtS. SrcAdr_FtSx

SrcAdr_FtSx – 32 битные регистры, задающие адрес массива данных в памяти. Этот адрес может принадлежать любому из следующих банков памяти: SMBx, NM1Bx, NM2Bx, External SDRAM или AMBx, т.е. любой области памяти кроме BROM и ARM Peripheral Area. Записываемое значение адреса должно быть выровнено до 64-разрядного слова, т.е. три младших бита должны быть нулевыми. Запись возможна только когда канал не занят (младший бит регистра Busy_FtSx равен нулю). Во время работы канала происходит инкрементирование адреса.

5.12.2.2 Регистр адреса приёмника каналов FtS. DstAdr_FtSx

DstAdr_FtSx – 32 битные регистры, задающие адрес регистра данных периферийного устройства, находящийся в области памяти ARM Peripheral Area. Для каждого периферийного устройства этот адрес уникален (см. описание соответствующего периферийного блока). Записываемое значение адреса должно быть выровнено до 32-разрядного слова, т.е. два младших бита должны быть нулевыми. Запись возможна только когда канал не занят (младший бит регистра Busy_FtSx равен нулю). Инкрементирование адреса во время работы канала не происходит.

5.12.2.3 Регистр числа передаваемых байт каналов FtS. TrSize_FtSx

TrSize_FtSx – 24 битные регистры, задающие число байт данных, передаваемых через канал. Запись возможна только когда канал не занят (младший бит регистра Busy_FtSx равен нулю). Во время чтения выдает оценочное значение, сколько байтов осталось еще передать.

5.12.2.4 Регистр типа упаковки данных каналов FtS. PackType_FtSx

PackType_FtSx – 2-битные регистры, задающие тип распаковки данных из 64-разрядных слов. Запись возможна только когда канал не занят (младший бит регистра Busy_FtSx равен нулю). Формат регистров приведен в таблице ниже.

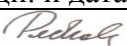
					ЮФКВ.431268.006РЭ			Лист
								132
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.88. Формат регистров PackType_FtSx

Биты	Название	Тип	Выполняемая функция
[1:0]	PACKTYPE	ЧТ/ЗП	Тип распаковки данных из 64-разрядных слов: 2'b00 – в байты; 2'b01 – в 16-разрядные слова; 2'b10 – в 32-разрядные слова;

5.12.2.5 Регистр типа ПДП интерфейса каналов FtS. PerHandsh_FtSx

PerHandsh_FtSx – 1 битный регистр, задает тип интерфейса ПДП периферийного блока. Для устройств SPI и USB (каналов FtS1 и FtS2) – должен быть сброшен, для контроллеров UART0 и UART1 (каналов FtS3 и FtS4) – должен быть установлен в единицу. Запись возможна только когда канал не занят (младший бит регистра Busy_FtSx равен нулю).

5.12.2.6 Регистр запуска/останова каналов FtS. Enable_FtSx

Enable_FtSx – 1 битные регистры управления запуском/остановом каналов. Для запуска канала необходимо записать значение 1. Для программного останова канала необходимо записать значение 0. Повторный запуск канала возможен только после завершения каналом текущей передачи 64-разрядного слова. Потери данных во внутреннем FIFO не происходит, т.к. перед остановом канала все уже считанные данные пересылаются.

5.12.2.7 Регистр состояния прерывания каналов FtS. Interrupt_FtSx

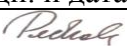
Interrupt_FtSx – 1-битные регистры индикации состояния прерывания. После завершения работы канала по причине пересылки каналом всех необходимых данных устанавливается в 1. Если причиной завершения работы канала был программный останов, то значение регистра остается 0. Чтобы снова запустить канал, необходимо записать значение 0 в младший бит регистра прерывания. Записать в данный регистр возможно только 0.

5.12.2.8 Регистр занятости каналов FtS. Busy_FtSx

Busy_FtSx – 1 битные регистры, показывающие занятость канала. Если равен 1, то канал занят, если 0 – то свободен. Основное отличие этого регистра от регистра Enable_FtS_x состоит в том, что при программном останове, когда в регистр Enable_FtS_x записывается значение 0 с шины APB, регистр Busy_FtS сбрасывается в 0 только после завершения передачи текущей 64-разрядной порции данных.

5.12.2.9 Регистр адреса источника каналов StF. SrcAdr_StFx

SrcAdr_StFx – 32 битные регистры, задающие адрес регистра данных периферийного устройства, находящийся в области памяти ARM Peripheral Area. Для каждого периферийного устройства этот адрес уникален (см. описание соответствующего периферийного блока). Записываемое значение адреса должно быть выровнено до 32-разрядного слова, т.е. два младших бита должны быть нулевыми. Запись возможна только когда канал не занят (младший бит регистра Busy_StFx равен нулю). Инкрементирование адреса во время работы канала не происходит.

					ЮФКВ.431268.006РЭ			Лист
								133
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.12.2.10 Регистр адреса приёмника каналов StF. DstAdr_StFx

DstAdr_StFx – 32 битные регистры, задающие адрес массива данных в памяти. Этот адрес может принадлежать любому из следующих банков памяти: SMBx, NM1Bx, NM2Bx, External SDRAM или AMBx, т.е. любой области памяти кроме BROM и ARM Peripheral Area. Записываемое значение адреса должно быть выровнено до 64-разрядного слова, т.е. три младших бита должны быть нулевыми. Запись возможна только когда канал не занят (младший бит регистра Busy_StFx равен нулю). Во время работы канала происходит инкрементирование адреса.

5.12.2.11 Регистр числа передаваемых байт каналов StF. TrSize_StFx

TrSize_StFx – 24 битные регистры, задающие число байт данных, передаваемых через канал. Запись возможна только когда канал не занят (младший бит регистра Busy_StFx равен нулю). Во время чтения выдает оценочное значение, сколько байтов осталось еще передать.

5.12.2.12 Регистр типа упаковки данных каналов StF. PackType_StFx

PackType_StFx – 2 битные регистры, задающие тип упаковки данных из периферийных устройств в 64-разрядные слова. Запись возможна только когда канал не занят (младший бит регистра Busy_StFx равен нулю). Формат регистров приведен в таблице ниже.

Таблица 5.89. Формат регистров PackType_StFx

Биты	Название	Тип	Выполняемая функция
[1:0]	PACKTYPE	ЧТ/ЗП	Тип упаковки данных в 64-разрядные слова: 2'b00 – из байтов 2'b01 – из 16-разрядных слов 2'b10 – из 32-разрядных слов

5.12.2.13 Регистр типа ПДП интерфейса каналов StF. PerHandsh_StFx

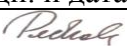
PerHandsh_StFx – 1 битный регистр, задает тип интерфейса ПДП периферийного блока. Для устройств SPI и USB (каналов StF1 и StF2) – должен быть сброшен, для контроллеров UART0 и UART1 (каналов StF3 и StF4) – должен быть установлен в единицу. Запись возможна только когда канал не занят (младший бит регистра Busy_StFx равен нулю).

5.12.2.14 Регистр запуска/останова каналов StF. Enable_StFx

Enable_StFx – 1 битные регистры управления запуском/остановом каналов. Для запуска канала необходимо записать значение 1. Для программного останова канала необходимо записать значение 0. Повторный запуск канала возможен только после завершения каналом текущей передачи 64-разрядного слова. Потери данных во внутреннем FIFO не происходит, т.к. перед остановом канала все уже считанные данные пере-сылаются.

5.12.2.15 Регистр состояния прерывания каналов StF. Interrupt_StFx

Interrupt_StFx – 1-битные регистры индикации состояния прерывания. После завершения работы канала по причине пересылки каналом всех необходимых данных

					ЮФКВ.431268.006РЭ			Лист
								134
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

устанавливается в 1. Если причиной завершения работы канала был программный останов, то значение регистра остается 0. Чтобы снова запустить канал необходимо записать значение 0 в младший бит регистра прерывания. Записать в данный регистр можно только 0.

5.12.2.16 Регистр занятости каналов StF. Busy_StFx

Busy_StFx – 1 битные регистры, показывающие занятость канала. Если равен 1, то канал занят, если 0 – то свободен. Основное отличие этого регистра от регистра Enable_StF_x состоит в том, что при программном останове, когда в регистр Enable_StF_x записывается значение 0 с шины APB, регистр Busy_StF сбрасывается в 0 только после завершения передачи текущей 64-разрядной порции данных.

5.12.3 Описание работы каналов

5.12.3.1 Каналы FtS

После инициализации и запуска канала FtS контроллер ПДП ждет прихода от периферии запроса на передачу данных по ПДП интерфейсу. Сразу же после получения сигнала начинается чтение 64-разрядного данного из памяти по быстрой шине AXI_S. Затем происходит запись данного требуемой разрядности (разрядность задается в регистре PackType_FtS_x) в регистр данных периферийного устройства по медленной шине AXI_P. После этого к периферии по интерфейсу ПДП выдается подтверждение, что данное было передано.

Далее после каждого прихода запроса от периферии контроллер ПДП записывает в соответствующий регистр периферии данное и выдает подтверждение того, что данное было передано. Так происходит до тех пор, пока не будут переданы все байты 64-разрядного регистра, в котором находятся данные, считанные из памяти.

После следующего запроса от периферии, из памяти вновь будет считано 64-разрядное слово и потом записано порциями требуемой разрядности в периферийное устройство.

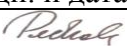
Как только будет передано необходимое число байт, канал прекращает свою работу и выдает прерывание.

Если во время работы канала произошел программный останов, то полностью передается уже считанное из памяти 64-разрядное данное, и только потом канал останавливает свою работу. Прерывание при этом не выдается.

5.12.3.2 Каналы StF

После инициализации и запуска канала контроллер ПДП ждет прихода от периферии запроса на передачу данных по ПДП интерфейсу. Сразу же после получения сигнала начинается чтение данного требуемой разрядности (разрядность задается в регистре PackType_StF_x) из соответствующего регистра периферии по медленной шине AXI_P. После этого к периферии по ПДП интерфейсу выдается подтверждение, что данное было считано.

Далее после каждого прихода запроса от периферии контроллер ПДП читает из соответствующего регистра периферии данное и выдает подтверждение того, что данное было считано. Так происходит до тех пор, пока внутри контроллера ПДП не наберется полное 64-разрядное слово, данные из которого затем будут переданы в память. Как только набирается полное слово, сразу же начинается запись в память по быстрой шине AXI_S этого 64-разрядного данного.

					ЮФКВ.431268.006РЭ			Лист
								135
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Если периферийное устройство продолжает выставлять запросы по ПДП интерфейсу, процесс наполнения полного слова и последующей его передачи будет продолжен.

Как только будет передано необходимое число байт, канал прекращает свою работу и выдает прерывание.

Если во время работы канала произошел программный останов, то происходит заполнение 64-разрядного регистра, он записывается в память, и только потом канал останавливает свою работу. Прерывание при этом не выдается.

5.12.3.3 Общие замечания

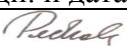
1. Контроллер ПДП всегда работает с памятью 64-разрядными словами. Если происходит передача данных из памяти в периферию и число передаваемых байтов не кратно 8, то читается из памяти все равно полное 64-разрядное слово, а в периферию происходит передача только необходимого числа байтов. Если происходит передача данных из периферии в память и число передаваемых байтов не кратно 8, то в память записывается все равно 64-разрядное слово. Но актуальны из записанных байтов лишь младшие столько байтов, сколько было необходимо передать.
2. Регистр Interrupt_x в состоянии 1 блокирует запись лишь регистра Enable_x. Во все остальные регистры соответствующего канала можно записать необходимые значения. Таким образом, происходит защита от повторного запуска канала до снятия бита прерывания. Но делать предварительные настройки канала возможно.
3. После завершения работы канала регистры не возвращаются в состояния до запуска канала. Это относится к регистрам SrcAdr, DstAdr, TrSize. Они остаются в том состоянии, до которого дошли к концу работы канала. Таким образом, нужно задать новые значения для этих регистров при запуске новой передачи.
4. При программном останове канал завершает свою работу только после сброса в 0 регистра Busy. Канал не может остановиться сразу, т.к. происходит завершение пересылки того 64-разрядного слова, которое уже начало передаваться. Это делается во избежание потери данных во внутренних регистрах контроллера и для того, чтобы пользователь имел возможность продолжить пересылку данных, которую по некоторой причине пришлось приостановить. Поэтому перед тем, как дальше работать с каналом, всегда необходимо проверить, сброшен младший бит регистра Busy_x или нет. И только после этого можно работать с каналом.

5.12.4 Прерывания

Контроллер ПДП имеет два выхода прерывания (DMACSENDINT и DMA-CRECVINT), подключенных к системному контроллеру прерываний.

Прерывание DMACSENDINT представляет собой комбинированное прерывание от 4 передающих из памяти в периферийные устройства каналов FtS. Для обработки этого прерывания необходимо, в процедуре обработчика прерываний, программно выяснить от какого именно канала(ов) FtS пришло прерывание.

Прерывание DMA-CRECVINT представляет собой комбинированное прерывание от 4 передающих из периферийных устройств в память каналов StF. Для обработки этого прерывания необходимо в процедуре обработчика прерываний программно выяснить от какого именно канала(ов) StF пришло прерывание.

					ЮФКВ.431268.006РЭ			Лист
								136
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.13 Контроллер асинхронного последовательного интерфейса UART

Контроллер асинхронного последовательного порта в микросхеме СПОВС имеет следующие характеристики:

- два идентичных порта UART
- порт совместим по формату регистров с популярной микросхемой UART 16550
- программируемая скорость обмена до 460800 бод из стандартного ряда скоростей
- также поддерживаются более высокие скорости (до 5Мбод), но они не попадают в стандартный ряд скоростей, т.е. могут применяться только при соединении 2-х микросхем СПОВС между собой, либо при подключении к другой МС с UART интерфейсом, поддерживающим нестандартные скорости обмена.
- подключение к системной шине СБИС через AMBA APB интерфейс, используется 8 младших бит шины данных
- два отдельных FIFO буфера размером по 16 байт (16 слов по байту) для приема и передачи данных
- возможность работы по прерыванию
- поддержка DMA доступа

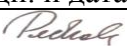
5.13.1 Устройство UART портов СБИС СПОВС

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

Таблица 5.90. Выводы микросхемы, входящие в состав UART портов

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

На рисунке (см. Рисунок 5.37) приведена блок-схема контроллера порта UART. Контроллер порта UART выполняет преобразование 8-разрядных данных, записываемых ядром ARM или контроллером ПДП через шину APB, в последовательное однобитное представление, а также обратное преобразование. Внутри контроллера выполняется буферизация передаваемых и принимаемых данных с помощью двух отдельных 16-байтных FIFO буферов. Наполненность этих буферов отслеживается по прерыванию или методом программного поллинга. Блок имеет интерфейс ПДП, подключенный к контроллеру ПДП, который позволяет передавать и принимать данные по порту без участия процессорного ядра ARM.

					ЮФКВ.431268.006РЭ				Лист
									137
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

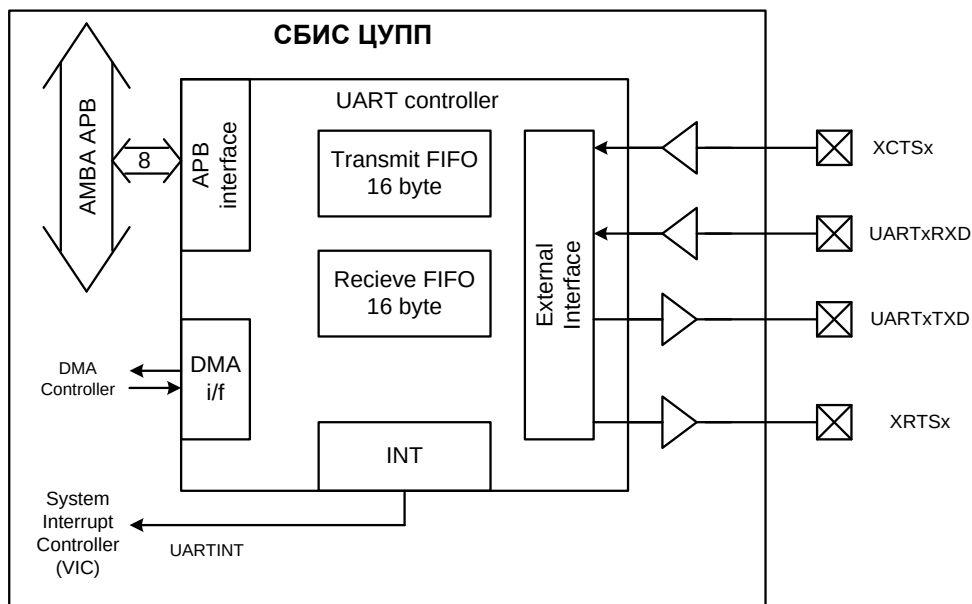


Рисунок 5.37 – Схема контроллера порта UART

Скорость передачи и приема данных (F_{uart}) программируется путем записи значений делителей системной частоты F_{apb} в регистры DLM и DLL. Эти два регистра могут быть концептуально представлены как один 16-разрядный регистр DL, где DLL – младшие 8 бит этого регистра, а DLM – старшие 8 бит этого регистра. Частота передачи вычисляется по следующей формуле:

$$F_{uart}(bps) = F_{apb}(Hz) / [16 * \{DLM, DLL\}]$$

где $\{DLM, DLL\} = 1 \dots 65536$. В таблице ниже приведены некоторые значения делителей и соответствующие им скорости передачи для входной частоты тактового сигнала 81.92 МГц:

Таблица 5.91. Возможные скорости порта UART

Делитель {DLM,DLL}	Получаемая скорость F_{uart} , bps	Стандартный ряд скоростей F_{uart} , bps	Ошибка, %
4255	1203,29	1200	-0,27
2133	2400,38	2400	-0,02
1066	4803,00	4800	-0,06
532	9624,06	9600	-0,25
355	14422,54	14400	-0,16
266	19248,12	19200	-0,25
178	28764,04	28800	0,12
134	38208,96	38400	0,50
89	57528,09	57600	0,12
66	77575,76	76800	-1,01
44	116363,64	115200	-1,01
22	232727,27	230400	-1,01
11	465454,55	460800	-1,01
6	853333,33	921600	7,41
4	1280000,00	1382400	7,41
2	2560000,00	2764800	7,41
1	5120000,00	5529600	7,41

					ЮФКВ.431268.006РЭ			Лист
								138
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		<i>Редько</i> 16.01.14		22755-1				

Передача данных через интерфейс UART выполняется асинхронно и подразумевает равенство скоростей приемника и передатчика. Если скорость приемника и передатчика отличаются более чем на $\pm 4\%$, то передача данных будет происходить с ошибками. В таблице приведены коэффициенты делителя частоты {DLM,DLL}, дающие наиболее близкие к стандартным скорости порта UART в микросхеме СБИС СПОВС. Таким образом, делитель частоты {DLM,DLL} обеспечивает требуемую точность стандартной скорости обмена UART в диапазоне от 1200 до 460800 бод включительно, а также нестандартные скорости до 5,1 Мбод, которые могут быть использованы для обмена данными между двумя микросхемами СПОВС.

Модемный интерфейс в СБИС СПОВС не используется. Входы блока, относящиеся к модемному интерфейсу (XDSR, XDCCD, XRI), фиксированы в неактивном состоянии, т.е. подключены к высокому уровню логического сигнала, а выходы (XOUT1, XOUT2, XDTR, XDDIS) – не подключены к внешним выводам. Используются только выходы XCTS и XRTS.

5.13.2 Описание функционирования UART порта

5.13.2.1 Процедура инициализации

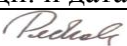
1. Установить бит DLAB регистра LCR – это позволит получить доступ к регистрам DLL и DLM
2. Установить необходимый делитель частоты в регистры DLM и DLL.
3. Сбросить бит DLAB регистра LCR.
4. Установить формат передаваемых данных в регистре LCR.
5. Разрешить необходимые прерывания путем установки бит регистра

5.13.2.2 Пример процедуры отсылки данных

1. Проверить пустоту передающего FIFO. Это может быть сделано путем анализа бита THRE регистра LSR. Также FIFO является пустым, если разрешено прерывание по пустоте передающего FIFO и это прерывание активно. Также можно проверить бит TEMT регистра LSR, если он установлен в 1'b1, то передающее FIFO пустое. Если FIFO пустое, то можно переходить к следующему шагу, иначе необходимо дождаться пока оно освободится.
2. Записать передаваемое слово данных в регистр TFR. За один раз можно записать блок данных не более 16 байт, т.к. скорость передачи через порт UART ниже, чем скорость записи в контроллер UART по шине APB, поэтому возможно переполнение передающего FIFO. В случае переполнения передающего FIFO теряются последние записываемые данные. По состоянию бита THRE (когда он установится в 1'b1) можно отследить момент, когда все данные передались. В этот же момент генерируется прерывание, сигнализирующее о пустоте передающего FIFO, если оно разрешено.

5.13.2.3 Пример процедуры приема данных

1. После генерации прерывания от контроллера UART, необходимо проверить тип прерывания, путем считывания бит NINT, ID[2:0] регистра IIR.
2. Если прерывание вызвано ошибкой в состоянии приемной линии (ID[2:0] = 011), то необходимо прочитать регистр LSR. Это делается для определения типа ошибки, а также для сброса прерывания.

					ЮФКВ.431268.006РЭ			Лист
								139
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

3. Если прерывание вызвано наличием данных в приемном FIFO буфере, то можно производить чтение принятых данных. Количество данных, вызывающих прерывание определяется значением поля RCVR[1:0] регистра FCR. Наличие данных в приемном FIFO также можно определить путем анализа бита DR регистра LSR. Прерывание снимается после того, как количество данных в приемном FIFO буфере станет меньше уровня, заданного полем RCVR[1:0] регистра FCR.

5.13.3 Регистровая модель контроллера порта UART

Программно доступные регистры контроллера порта UART расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение UART0 Base = 0xFFF8B000 и UART1 Base = 0xFFF8C000 общий размер по 4 Кб. Спецификация регистров представлена в таблице ниже:

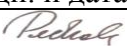
Таблица 5.92. Спецификация регистров контроллера порта UART

DLAB	Адрес	Тип	Разрядность	Значение после инициализации	Имя	Описание
0	UARTx Base+ 0x00	ЧТ	8	00h	RFR	Регистр принимающего FIFO.
		ЗП			TFR	Регистр передающего FIFO.
1		ЧТ/ЗП			DLL	Регистр значения делителя частоты (младший байт).
0	UARTx Base+ 0x04	ЧТ/ЗП	4	00h	IER	Регистр разрешения прерываний.
1			8		DLM	Регистр значения делителя частоты (старший байт).
0/1	UARTx Base+ 0x08	ЧТ	8	C1h	IIR	Регистр распознавания прерываний.
		ЗП	8	00h	FCR	Регистр управления FIFO.
0/1	UARTx Base+ 0x0C	ЧТ/ЗП	8	00h	LCR	Регистр управления линией.
0/1	UARTx Base+ 0x10	ЧТ/ЗП	8	00h	MCR	Регистр управления модемом.
0/1	UARTx Base+ 0x14	ЧТ	8	60h	LSR	Регистр состояния линии.
0/1	UARTx Base+ 0x18	ЧТ	8	x0h	MSR	Регистр состояния модема.
0	UARTx Base+ 0x1C	ЧТ/ЗП	8	00h	SCR	Рабочий регистр.
1		ЗП			TST	Регистр тестового режима.

DLAB – это 7-ой бит регистра LCR, в зависимости от значения этого бита меняется отражение регистров на память. Далее приведено подробное описание каждого регистра.

5.13.3.1 Регистр принимающего FIFO. RFR

Регистр принимаемых данных, доступен только по чтению. Принимаемые портом данные попадают в FIFO, которое может хранить до 16 байт данных. Принимаемые портом данные должны быть своевременно считаны из данного регистра, в случае

					ЮФКВ.431268.006РЭ			Лист
								140
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

переполнения теряются наиболее старые данные. Принимающее FIFO может быть сброшено путем установки бита RxF RST регистра FCR.

5.13.3.2 Регистр передающего FIFO. TFR

Регистр передаваемых данных, доступен только по записи. После записи данных в этот регистр, они попадают в передающее FIFO, которое может вмещать до 16 байт данных. Из передающего FIFO данные попадают в передающий сдвиговый регистр. При передаче данных необходимо отслеживать полноту передающего FIFO, т.к. если скорость записи в FIFO превысит скорость передачи через внешний интерфейс, произойдет переполнение с потерей наиболее старых данных. Передающее FIFO может быть сброшено путем установки бита TxF RST регистра FCR.

5.13.3.3 Регистр разрешения прерываний. IER

IER – регистр разрешения прерываний. Формат регистра приведен в таблице ниже.

Таблица 5.93. Формат регистра IER

Биты	Название	Тип	Выполняемые функции
[7:4]	-	-	Зарезервировано
[3]	EDSSI	ЧТ/ЗП	Бит, разрешающий прерывание от модема. Если EDSSI установлен в “1”, и если установлены биты [3:0] регистра MSR, то генерируется прерывание
[2]	ELSI	ЧТ/ЗП	Бит, разрешающий прерывание по состоянию приема. Если ELSI установлен в “1”, и если установлены биты [4:1] регистра LSR, то генерируется прерывание
[1]	ETBEI	ЧТ/ЗП	Бит, разрешающий прерывание по пустоте регистра TFR. Если ETBEI установлен в “1”, и если регистр TFR пуст, то генерируется прерывание
[0]	ERBFI	ЧТ/ЗП	Бит, разрешающий прерывание по достижению порога заполнения приемного FIFO. Если ERBFI установлен в “1”, и если данные в приемном FIFO антигли определенного количества, то генерируется прерывание

5.13.3.4 Регистр распознавания прерываний. IIR

IIR – регистр распознавания прерываний. Формат регистра IIR приведен в таблице ниже.

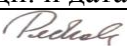
					ЮФКВ.431268.006РЭ			Лист
								141
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.94. Формат регистра IIR

Биты	Название	Тип	Выполняемые функции
[7: 6]	FIFO[1:0]	ЧТ	Состояние FIFO «11» фиксировано
[5:4]	-	ЧТ	Установлены в значение «00»
[3:1]	ID[2:0]	ЧТ	Типы прерываний (значения действительны при NINT = 0): 011(1) – состояние приемной линии. Сбрасывается чтением из регистра LSR. 010(2) – наличие данных в приемном FIFO. Сбрасывается чтением из приёмного FIFO. 110(2)- истечение периода timeout. Сбрасывается чтением из приёмного FIFO. 001(3) – передающее FIFO пусто. Сбрасывается чтением из регистра IIR, или записью в передающее FIFO. 000(4) – состояние модема. Сбрасывается чтением из регистра MSR В скобках указан приоритет прерывания.
[0]	NINT	ЧТ	Бит состояния прерывания: 0 – прерывание активно 1 – нет сигнала прерывания, ID[2:0] = 3'b000

Биты {ID[2:0],NINT} меняют состояние с 0100 на 1100, если истекает timeout период, при наличии прерывания по причине наличия данных в приемном FIFO.

5.13.3.5 Регистр управления FIFO. FCR

FCR – регистр управления FIFO. Формат регистра приведен в таблице ниже.

Таблица 5.95. Формат регистра FCR

Биты	Название	Тип	Выполняемые функции
[7: 6]	RCVR[1:0]	ЗП	Пороговый уровень принимающего FIFO: 00 – 1 байт 01 – 4 байт 10 – 8 байт 11 – 14 байт
[5:4]	не используется	-	Зарезервировано
[3]	DMA MODE	ЗП	Режим работы ПДП интерфейса: 0 – одиночный режим передачи 1 – пакетный режим передачи (этот режим не совместим с DMAC) Для правильной работы контроллером ПДП, этот бит должен быть обязательно сброшен в 0.
[2]	TxF RST	ЗП	Сброс передающего FIFO. 1 – сброс
[1]	RxF RST	ЗП	Сброс принимающего FIFO. 1 – сброс
[0]	не используется	-	Зарезервировано

5.13.3.6 Регистр управления линией. LCR

LCR – регистр управления линией. Формат регистра приведен в таблице ниже.

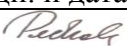
					ЮФКВ.431268.006РЭ			Лист
								142
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.96. Формат регистра LCR

Биты	Название	Тип	Выполняемые функции
[7]	DLAB	ЧТ/ЗП	Бит переключения регистровых полей 0 – доступ к регистрам RFR, TFR, IER, SCR 1 - доступ к регистрам DLL,DLM,TST
[6]	SB	ЧТ/ЗП	Set Break. Принудительная остановка передачи. 1 – сигнал UARTxTXD устанавливается в 0
[5]	SP	ЧТ/ЗП	Stick Parity. Фиксация бита четности 0 – Бит четности формируется на основе EPS и PEN. 1 – Бит четности фиксирован и зависит от состояния EPS и PEN. Значение бита выдается при передаче и проверяется при приеме. PEN=1, EPS=0, бит равен 1 PEN=1, EPS=1, бит равен 0
[4]	EPS	ЧТ/ЗП	Even Parity Select. Выбор типа контроля по четности 0 – контроль нечетности 1 – контроль четности
[3]	PEN	ЧТ/ЗП	Parity Enable. Разрешение контроля четности. 0 – бит четности не посылается, не проверяется 1 – бит четности посылается и проверяется Бит четности прибавляется после поля данных, бит остановки следует самым последним.
[2]	STB	ЧТ/ЗП	Stop Bit Length. Количество бит stop-последовательности. 0 – 1 бит 1 – 1,5 бита (последовательность данных 5 бит) 2 бита (последовательность данных 6 – 8 бит)
[1: 0]	WLS1:0	ЧТ/ЗП	Выбор длины передаваемого/принимаемого слова 00 – 5 бит 01 – 6 бит 10 – 7 бит 11 – 8 бит

5.13.3.7 Регистр управления модемом. MCR

MCR – регистр управления модемом. Модемный интерфейс используется не полностью в СБИС СПОВС. Входы блока, относящиеся к модемному интерфейсу (XDSR, XD CD, XRI), выставлены в 1, т.е. в неактивное состояние, а выходы (XOUT1, XOUT2, XDTR, XD DIS) – не подключены к внешним выводам. Используются только выходы XCTS и XRTS. Формат регистра приведен в таблице ниже.

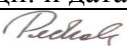
					ЮФКВ.431268.006РЭ			Лист
								143
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.97. Формат регистра MCR

Биты	Название	Тип	Выполняемые функции
[7:5]	Зарезервировано	-	Зарезервировано
[4]	LOOP	ЧТ/ЗП	Кольцевой режим (режим самопроверки). Если LOOP установлен в 1, то: 1) UARTxTXD принимает значение 1 2) UARTxRXD отделяется от входа сдвигового регистра приемника 3) Выходы сдвигового регистра передатчика соединяются с входами сдвигового регистра приемника 4) Сигналы состояния модема выключаются (NCTS,NDSR,NDCD,NRI) 5) Сигналы управления модемом соединяются с входами состояния модема. 6) CTS ← RTS DSR ← DTR RI ← OUT1 DCD ← OUT2
[3]	OUT2	ЧТ/ЗП	Управляющие сигналы. Значение, записанное в этот бит, будет выведено на соответствующий вывод контроллера.
[2]	OUT1		
[1]	RTS		
[0]	DTR		

5.13.3.8 Регистр состояния линии. LSR

LSR – регистр состояния линии. Формат регистра LSR приведен в таблице ниже.

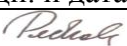
					ЮФКВ.431268.006РЭ			Лист
								144
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.98. Формат регистра LSR

Биты	Название	Тип	Выполняемые функции
[7]	ERRF	ЧТ	Ошибка в приемном FIFO. Бит устанавливается, когда в приемном FIFO присутствуют данные, имеющие ошибку контроля четности, ошибку кадрирования или ошибку определения останова. Бит сбрасывается при чтении LSR, если больше нет ошибок в приемном FIFO, за исключением той, которая вызвала установку ERRF.
[6]	TEMT	ЧТ	Сигнал пустоты передатчика. TEMT устанавливается в «1», когда сдвиговый регистр передатчика и передающее FIFO становятся пустыми
[5]	THRE	ЧТ	Сигнал пустоты передающего FIFO. Устанавливается в 1, когда передающее FIFO становится пустым.
[4]	BI	ЧТ	Прерывание по останову. Бит устанавливается, когда UARTxRXD удерживается в “0” дольше, чем время необходимое для передачи (START бит + DATA биты + PARITY бит + STOP бит). BI обнуляется, когда этот регистр считывается процессором.
[3]	FE	ЧТ	Ошибка кадрирования. Бит устанавливается, когда принятые данные не имеют правильного бита останова. FE обнуляется, когда этот регистр считывается процессорным ядром.
[2]	PE	ЧТ	Ошибка контроля четности. Бит устанавливается, когда принятые данные не имеют правильного бита четности. PE обнуляется, когда этот регистр считывается процессорным ядром.
[1]	OE	ЧТ	Ошибка переполнения принимающего FIFO. Бит устанавливается, когда принимающее FIFO полное и имеет место прием следующих данных. OE обнуляется, когда этот регистр считывается процессорным ядром.
[0]	DR	ЧТ	Готовность данных (данные находятся в приемном FIFO). Бит показывает, что в FIFO хранится более чем 1 байт данных. Бит устанавливается, когда данные присутствуют в FIFO. DR обнуляется, когда из FIFO считаны все данные

5.13.3.9 Регистр состояния модема. MSR

MSR – регистр состояния модема. Модемный интерфейс в СБИС СПОВС не используется. Входы блока, относящиеся к модемному интерфейсу (XDSR, XD CD, XRI), в фиксированы в неактивном состоянии, т.е. подключены к высокому уровню логического сигнала, а выходы (XOUT1, XOUT2, XDTR, XDDIS) – не подключены к внешним выводам. Используются только выходы XCTS и XRTS. Формат регистра приведен в таблице ниже.

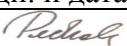
					ЮФКВ.431268.006РЭ			Лист
								145
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.99. Формат регистра MSR

Биты	Название	Тип	Выполняемые функции
[7]	DCD	ЧТ	Обнаружения несущей информации (Carrier detect) Когда Loop=0, указывает состояние входного сигнала XDCD (всегда 1'b1) Когда Loop=1, равен OUT2 регистра MCR
[6]	RI	ЧТ	Индикатор звонка (Ring Indicator) Когда Loop=0, указывает состояние входного сигнала XRI (всегда 1'b1) Когда Loop=1, равен OUT1 регистра MCR
[5]	DSR	ЧТ	Сигнал готовности к передаче данных(Data Set Ready). Когда Loop=0, указывает состояние входного сигнала XDSR (всегда 1'b1) Когда Loop=1, равен DTR регистра MCR
[4]	CTS	ЧТ	Готовность к передаче (Clear To Send). Когда Loop=0, указывает состояние входного сигнала XCTS Когда Loop=1, равен RTS регистра MCR
[3]	DDCD	ЧТ	Дельта DCD. Устанавливается, когда меняется сигнал DCD после последнего считывания CPU (реально не меняется, т.к. XDCCD – всегда 1'b1). После чтения из этого регистра бит обнуляется
[2]	TERI	ЧТ	Задний фронт индикатора звонка. Бит устанавливается, когда сигнал RI переходит из состояния «1» в «0» после последнего считывания CPU (реально не меняется, т.к. XRI – всегда 1'b1) . После чтения из этого регистра бит обнуляется
[1]	DDSR	ЧТ	Дельта DSR. Устанавливается, когда меняется сигнал DSR после последнего считывания CPU (реально не меняется, т.к. XDSR – всегда 1'b1) . После чтения из этого регистра бит обнуляется
[0]	DCTS	ЧТ	Дельта CTS. Устанавливается, когда меняется сигнал CTS после последнего считывания CPU. После чтения из этого регистра бит обнуляется

5.13.3.10 Рабочий регистр. SCR

Регистр SCR программно доступен по чтению и по записи. Значение после инициализации равняется 00h.

5.13.3.11 Регистры значения делителя частот. DLL и DLM

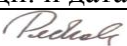
Два 8-битных регистра {DLM, DLL} могут быть представлены как один 16-разрядный регистр DL, содержащий делитель частоты. Значение этого регистра определяет скорость передачи порта. Формат регистров DLM и DLL приведен в таблице ниже.

Таблица 5.100. Формат регистров DLM и DLL

Регистр	Название	Тип	Выполняемые функции
DLM[7:0]	DL[15:8]	ЧТ/ЗП	Если бит DLAB регистра LCR установлен в «1», то регистры DLL и DLM программно доступны на чтение и на запись. После сброса DLL и DLM устанавливаются в значение 00h Скорость передачи программируется в диапазоне {DLM,DLL} = 0xFFFF ... 0x0001 (65536 ... 1) Скорость передачи вычисляется по следующей формуле: $F_{uart}(bps) = F_{apb}(Hz) / [16 * \{DLM, DLL\}]$
DLL[7:0]	DL[7:0]		

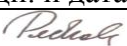
5.13.3.12 Регистр тестового режима. TST

Регистр, используемый для установки тестового режима. Пользователю не разрешен доступ. После сброса регистр принимает значение равное 00h.

					ЮФКВ.431268.006РЭ			Лист
								146
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.13.4 Прерывания

Каждый из контроллеров портов UART имеет один комбинированный выход прерывания UARTINT, подключенный к системному контроллеру прерываний VIC. Причина возникновения прерывания определяется анализом бит ID[2:0] регистра IIR.

					ЮФКВ.431268.006РЭ			Лист
								147
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.14 Контроллер последовательного синхронного интерфейса SPI

Последовательный порт в микросхеме СПОВС имеет следующие характеристики:

- формат данных – Motorola SPI в режиме master
- поддержка полнодуплексного SPI mode 0,1,2 и 3
- программируемая частота обмена, максимальная частота обмена – 41 МГц
- поддержка до 8 подключенных ведомых SPI устройств в режиме разделения времени, т.е. одновременно возможен обмен данными только с одним SPI устройством
- подключение к системной шине СБИС через AMBA APB интерфейс, используется 16 младших бит шины данных
- два отдельных 16-разрядных FIFO буфера глубиной 8 слов для приема и передачи данных
- программируемая длина SPI фрейма от 4 до 16 бит
- возможность работы по прерыванию
- поддержка DMA доступа

5.14.1 Устройство SPI порта СБИС СПОВС

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

Таблица 5.101. Выводы микросхемы, входящие в состав SPI порта

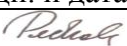
Вывод	Тип буфера	Примечание
SPICLK	out	выход тактового сигнала SPI порта
SPITXD	out	выход данных SPI порта
SPIRXD	in (pullup 33K)	вход данных SPI порта
SPI_CS0 ... SPI_CS7	out	выбор ведомого SPI устройства (сигнал с активным низким уровнем)

Выбор активного SPI slave устройства осуществляется путем предварительной записи бит [6:4] регистра PDR3 в блоке GPIO в соответствии с таблицей:

Таблица 5.102. Выбор активного ведомого SPI устройства

GPIO.PDR3[6:4]	Примечание
000	активное устройство подключено к SPI_CS0 (по умолчанию)
001	активное устройство подключено к SPI_CS1
010	активное устройство подключено к SPI_CS2
011	активное устройство подключено к SPI_CS3
100	активное устройство подключено к SPI_CS4
101	активное устройство подключено к SPI_CS5
110	активное устройство подключено к SPI_CS6
111	активное устройство подключено к SPI_CS7

На рисунке ниже приведена схема выбора активного SPI устройства:

					ЮФКВ.431268.006РЭ			Лист
								148
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

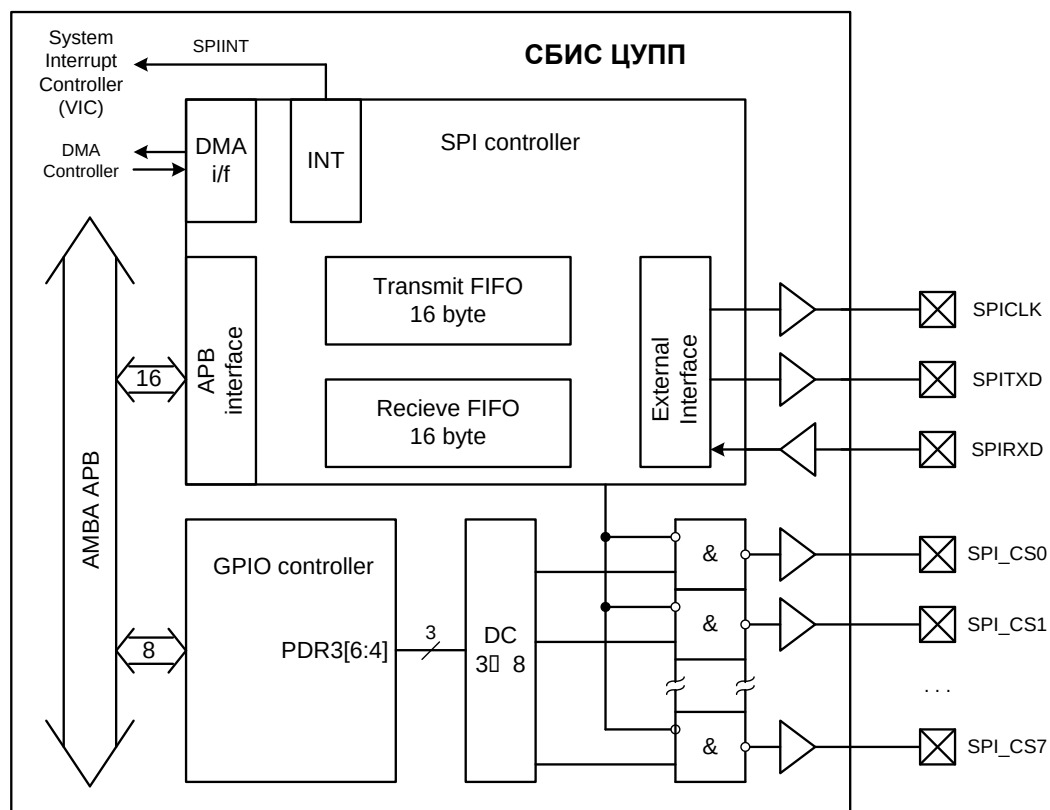


Рисунок 5.38 – Схема выбора активного SPI устройства

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

Частота передачи и приема данных (F_{spi}) программируется путем записи значений делителей системной частоты $F_{apb}=81\text{МГц}$ в поля SCR регистра SSPCR0 и поле CPSDVSR регистра SSPCPSR, по следующей формуле:

$$F_{spi} = F_{apb} / [CPSDVSR * (1 + SCR)]$$

где CPSDVSR = 2...254, а SCR = 0...255. Таким образом, возможна работа SPI интерфейса на частотах до 41 МГц.

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

- поддержка SPI mode 3
- частота работы не менее $F_{apb}/8$
- поддержка команды FAST_READ (код – 0x0B)
- загрузочная flash-память должна быть подключена к SPI_CS0 выводу СБИС

Подробное описание процесса начальной загрузки из SPI flash памяти см. п. 5.17.5.

					ЮФКВ.431268.006РЭ			Лист
								149
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	16.01.14		22755-1					

5.14.2 Протокол передачи данных в различных режимах SPI порта

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

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

Ниже приведены временные диаграммы каждого из режимов.

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

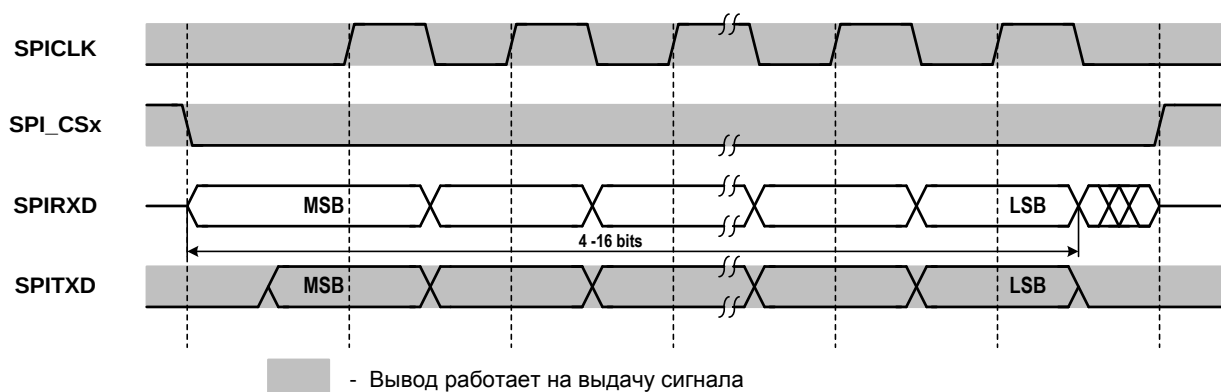


Рисунок 5.39 – SPI mode 0 (SPO=0, SPH=0) одиночная передача

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

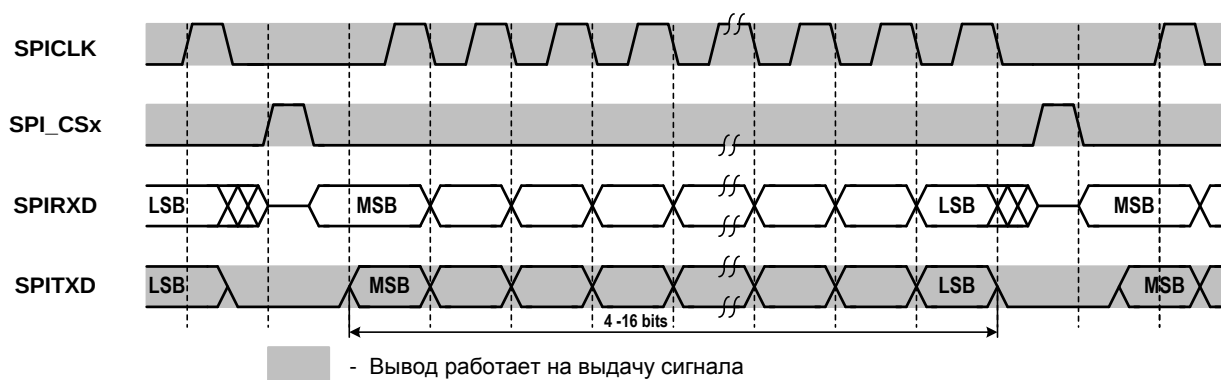


Рисунок 5.40 – SPI mode 0 (SPO=0, SPH=0) передача из нескольких пакетов

					ЮФКВ.431268.006РЭ			Лист
								150
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	Редько 16.01.14		22755-1					

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

5.14.2.2 SPI mode 1 (SPO=0, SPH=1)

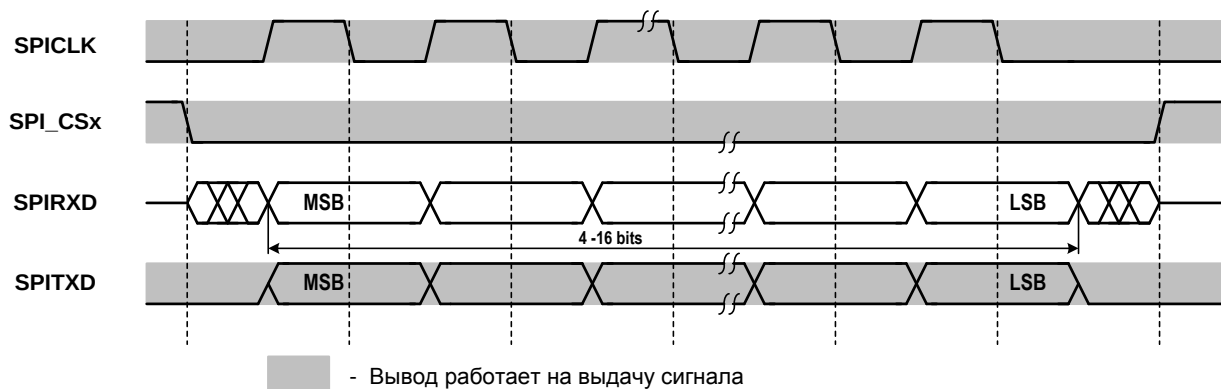


Рисунок 5.41 – SPI mode 1 (SPO=0, SPH=1) одиночная передача

Выдача данных происходит по фронту сигнала SPICLK, защелкивание входных данных – по срезу сигнала SPICLK. Сигнал SPI_CSx не меняет свое значение между словами в передаче из нескольких пакетов.

5.14.2.3 SPI mode 2 (SPO=1, SPH=0)

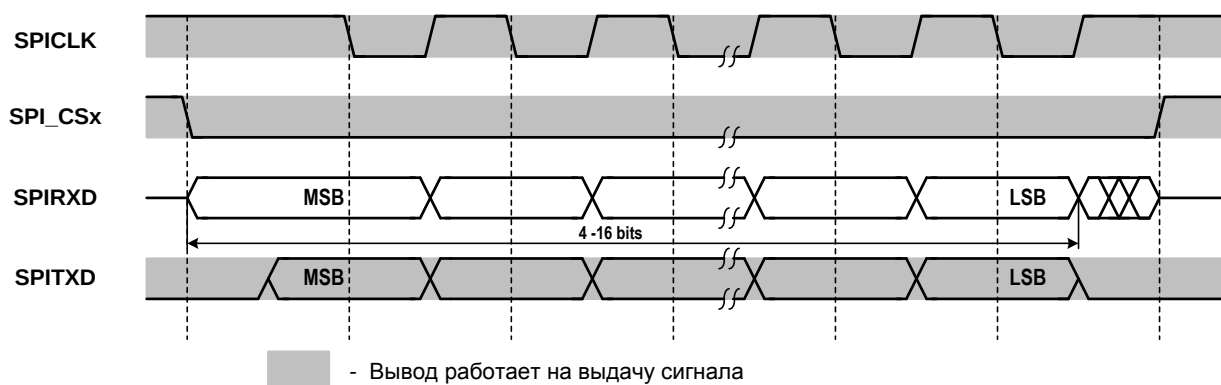


Рисунок 5.42 – SPI mode 2 (SPO=1, SPH=0) одиночная передача

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

					ЮФКВ.431268.006РЭ				Лист
									151
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	<i>Редко</i> 16.01.14		22755-1						

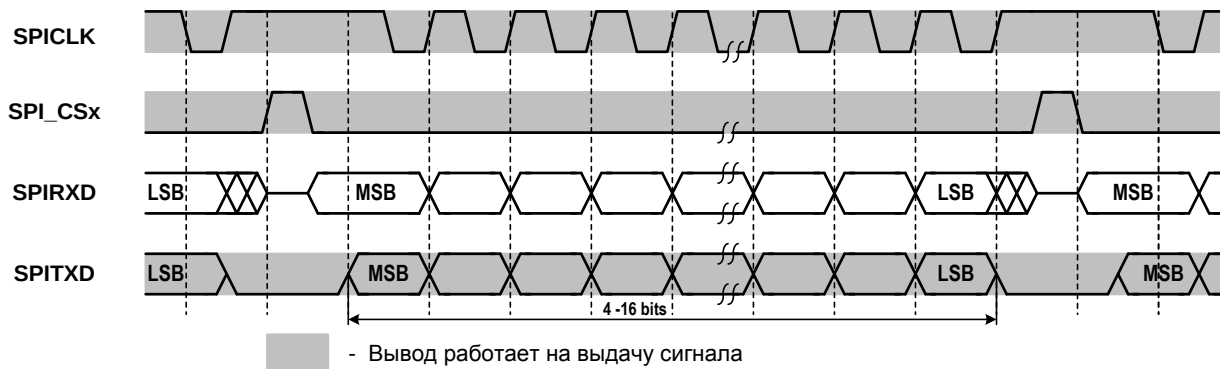


Рисунок 5.43 – SPI mode 2 (SPO=1, SPH=0) передача из нескольких пакетов

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

5.14.2.4 SPI mode 3 (SPO=1, SPH=1)

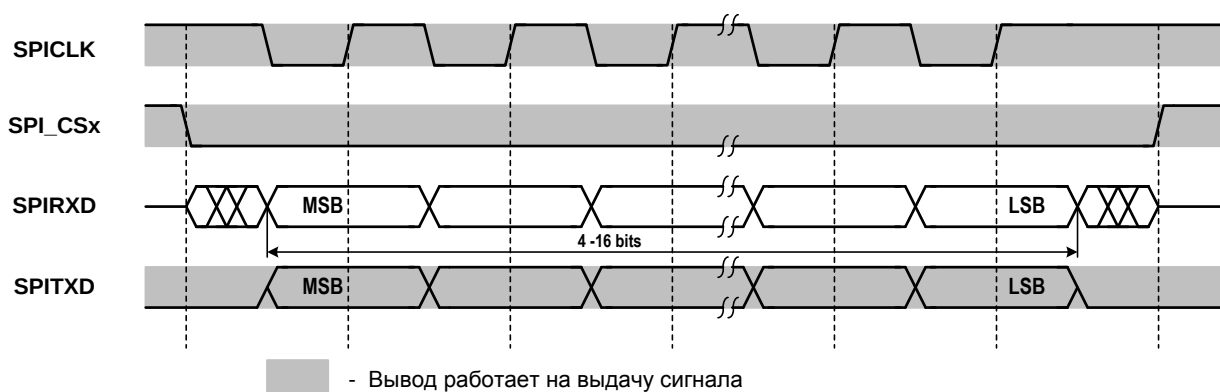


Рисунок 5.44 – SPI mode 3 (SPO=1, SPH=1) одиночная передача

Выдача данных происходит по срезу сигнала SPICLK, защелкивание входных данных – по фронту сигнала SPICLK. Сигнал SPI_CSx не меняет свое значение между словами в передаче из нескольких пакетов.

5.14.3 Регистровая модель контроллера порта SPI

Программно доступные регистры контроллера порта SPI расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение SSP Base = 0xFFF8D000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

					ЮФКВ.431268.006РЭ				Лист
									152
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№	подл.	Подп. и дата		Взам.инв.№	Инв.№	дубл.	Подп. и дата		
22755-2		16.01.14		22755-1					

Таблица 5.103. Спецификация регистров контроллера порта SPI

Адрес	Тип	Разрядность	Начальное значение	Имя	Описание
SSP Base + 0x00	ЧТ/ЗП	16	0x0000	SSPCR0	Регистр управления 0
SSP Base + 0x04	ЧТ/ЗП	4	0x0	SSPCR1	Регистр управления 1
SSP Base + 0x08	ЧТ/ЗП	16	0x----	SSPDR	Регистр данных. SSPDR
SSP Base + 0x0C	ЧТ	5	0x03	SSPSR	Регистр состояния. SSPSR
SSP Base + 0x10	ЧТ/ЗП	8	0x00	SSPCPSR	Регистр множителя синхросигнала. SSPCPSR
SSP Base + 0x14	ЧТ/ЗП	4	0x0	SSPIMSC	Регистр снятия или установки маски прерываний. SSPIMSC
SSP Base + 0x18	ЧТ	4	0x8	SSPRIS	Регистр состояния прерывания до наложения маски. SSPRIS
SSP Base + 0x1C	ЧТ	4	0x0	SSPMIS	Регистр состояния прерывания после наложения маски. SSPMIS
SSP Base + 0x20	ЗП	4	0x0	SSPICR	Регистр сброса прерывания. SSPICR
SSP Base + 0x24	ЧТ/ЗП	2	0x0	SSPDMACR	Регистр управления DMA. SSPDMACR
SSP Base + 0x28 to 0xFDC	-	-	-	-	Зарезервировано
SSP base + 0xFEO	ЧТ	8	0x22	SSPPeriphID0	Идентификатор периферийного устройства. SSPPeriphID0 биты [7:0]
SSP base + 0xFE4	ЧТ	8	0x10	SSPPeriphID1	Идентификатор периферийного устройства. SSPPeriphID1 биты [15:8]
SSP base + 0xFE8	ЧТ	8	0x04	SSPPeriphID2	Идентификатор периферийного устройства. SSPPeriphID2 биты [23:16]
SSP base + 0xFEC	ЧТ	8	0x00	SSPPeriphID3	Идентификатор периферийного устройства. SSPPeriphID3 биты [31:24]
SSP base + 0xFF0	ЧТ	8	0x0D	SSPPCellID0	PrimeCell идентификатор. SSPPCellID0 биты [7:0]
SSP base + 0xFF4	ЧТ	8	0xF0	SSPPCellID1	PrimeCell идентификатор. SSPPCellID1 биты [15:8]
SSP base + 0xFF8	ЧТ	8	0x05	SSPPCellID2	PrimeCell идентификатор. SSPPCellID2 биты [23:16]
SSP base + 0xFFC	ЧТ	8	0xB1	SSPPCellID3	PrimeCell идентификатор. SSPPCellID3 биты [31:24]

Далее приведено подробное описание каждого регистра.

5.14.3.1 Регистр управления. SSPCR0

SSPCR0 – управляющий регистр, содержащий пять битовых полей, которые управляют различными функциями порта SSP. В таблице ниже приведены формат регистра SSPCR0.

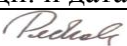
					ЮФКВ.431268.006РЭ			Лист
								153
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.104. Формат регистра SSPCR0

Биты	Название	Тип	Выполняемая функция
[15:8]	SCR	ЧТ/ЗП	Serial Clock Rate. Значение SCR используется для определения скорости приема и передачи блока SSP. Скорость вычисляется по следующей формуле: $\frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$, где CPSDVR – значение от 2 до 254, программируемая регистром SSPCSR. SCR может принимать значение от 0 до 255.
[7]	SPH	ЧТ/ЗП	SSPCLKOUT Phase. Фаза сигнала SSPCLKOUT
[6]	SPO	ЧТ/ЗП	SSPCLKOUT Polarity. Полярность SSPCLKOUT
[5:4]	FRF	ЧТ/ЗП	Frame Format. Формат данных: 00 – Motorola SPI формат данных 01 – не используется 10 – не используется 11 – не используется
[3:0]	DSS	ЧТ/ЗП	Data Size Select. Выбор размера данных: 0000 – зарезервировано 0001 – зарезервировано 0010 – зарезервировано 0011 – 4-битные данные 0100 – 5-битные данные 0101 – 6-битные данные 0110 – 7-битные данные 0111 – 8-битные данные 1000 – 9-битные данные 1001 – 10-битные данные 1010 – 11-битные данные 1011 – 12-битные данные 1100 – 13-битные данные 1101 – 14-битные данные 1110 – 15-битные данные 1111 – 16-битные данные

5.14.3.2 Регистр управления. SSPCR1

SSPCR1 – управляющий регистр, содержащий четыре битовых поля, которые управляют различными функциями порта SSP. В таблице ниже приведен формат регистра SSPCR1.

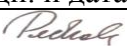
					ЮФКВ.431268.006РЭ			Лист
								154
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.105 – Формат регистра SSPCR1

Биты	Название	Тип	Выполняемая функция
[15:4]	-	-	Зарезервировано, при записи должен записываться нуль
[3]	SOD	ЧТ/ЗП	Slave-mode output disable. Бит используется только при работе SSP в режиме приема.
[2]	MS	ЧТ/ЗП	Master or Slave mode Select. Поддерживается работа только в режиме передачи (режим Master). Этот бит должен быть всегда равен 0 (значение по-умолчанию).
[1]	SSE	ЧТ/ЗП	Synchronous Serial Port Enable. Управление портом: 0 – порт неактивен 1 – порт активен
[0]	LBM	ЧТ/ЗП	Loop Back Mode. Управление тестовым кольцевым режимом: 0 – нормальный режим функционирования порта 1 – выходы передающего сдвигового регистра соединены с входами принимающего сдвигового регистра

5.14.3.3 Регистр данных. SSPDR

SSPDR – 16-битный регистр данных. Чтение из этого регистра возвращает слово данных из приемного FIFO, на которое в данный момент указывает счетчик чтения. Данные попадают в приемное FIFO автоматически из блока принимающей логики после приема полного SPI фрейма.

Запись в SSPDR приводит к записи данных в передающее FIFO, откуда автоматически перемещаются в передающий сдвиговый регистр. Далее данные последовательно сдвигаются наружу кристалла через вывод SPITXD, в соответствии с запрограммированной скоростью работы порта.

Когда разрядность передаваемых данных меньше, чем 16 бит, программист должен выравнивать данные по правому краю слова. Принимаемые данные выравниваются автоматически в принимающем буфере.

Таблица 5.106. Формат регистра SSPDR

Биты	Название	Тип	Выполняемая функция
[15:0]	DATA	ЧТ/ЗП	Принимающее/передающее FIFO: Чтение – принимающего FIFO Запись – передающего FIFO Если разрядность данных меньше 16 бит, то данные должны быть выровнены по правому краю слова. Старшие неиспользуемые биты игнорируются автоматически. В режиме приема происходит автоматическое выравнивание по правому краю слова.

5.14.3.4 Регистр состояния. SSPSR

SSPSR – регистр, доступный только по чтению, в котором хранится состояние полноты FIFO и состояние занятости порта PrimeCell SSP. В таблице ниже приведен формат регистра SSPSR.

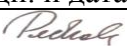
					ЮФКВ.431268.006РЭ			Лист
								155
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.107. Формат регистра SSPSR

Биты	Название	Тип	Выполняемая функция
[15:5]	-	-	Зарезервирован, должен быть прописан нуль
[4]	BSY	ЧТ	Флаг занятости порта SSP: 0 = SSP простаивает 1 = SSP передает и/или принимает пакет данных
[3]	RFF	ЧТ	Сигнал полноты принимающего FIFO: 0 = принимающее FIFO не полное 1 = принимающее FIFO полное
[2]	RNE	ЧТ	Сигнал не пустоты принимающего FIFO: 0 = принимающее FIFO пустое 1 = принимающее FIFO не пустое
[1]	TNF	ЧТ	Сигнал не полноты передающего FIFO: 0 = передающее FIFO полное 1 = передающее FIFO не полное
[0]	TFE	ЧТ	Сигнал пустоты передающего FIFO: 0 = передающее FIFO не пустое 1 = передающее FIFO пустое

5.14.3.5 Регистр множителя синхросигнала. SSPCPSR

SSPCPSR- Регистр множителя синхросигнала, определяет делитель, на который входной сигнал SSPCLK должен быть поделен для дальнейшего использования. В таблице ниже приведен формат регистра SSPCPSR.

Таблица 5.108. Формат регистра SSPCPSR

Биты	Название	Тип	Выполняемая функция
[15:8]	-	-	Зарезервирован, должен быть прописан нуль
[7:0]	CPSDVSR	ЧТ/ЗП	Clock Prescale Divisor. Значение делителя синхросигнала. Должен принимать четные значения от 2 до 254, в зависимости от частоты сигнала SSPCLK. Младшие значащие биты всегда принимают значение нуль при чтении

5.14.3.6 Регистр снятия или установки маски прерываний. SSPIMSC

SSPIMSC- регистр снятия или установки маски прерываний. Чтение регистра возвращает текущее значение маски, соответствующего прерывания. Запись 1 в соответствующий бит устанавливает маску, а запись 0 – снимает её. В таблице ниже приведен формат регистра SSPIMSC.

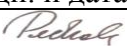
					ЮФКВ.431268.006РЭ			Лист
								156
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.109. Формат регистра SSPIMSC

Биты	Название	Тип	Выполняемая функция
[15:4]	-	-	Зарезервирован, должен быть прописан нуль
[3]	TXIM	ЧТ/ЗП	Transmit Interrupt mask. Маска прерывания передающего FIFO(SSPTXINTR): 0 = Прерывание по заполнению TxFIFO более чем на половину замаскировано 1 = Прерывание по заполнению TxFIFO более чем на половину не маскировано
[2]	RXIM	ЧТ/ЗП	Receive Interrupt Mask. Маска прерывания принимающего FIFO(SSPRXINTR): 0 = Прерывание по заполнению Rx FIFO менее чем на половину замаскировано 1 = Прерывание по заполнению Rx FIFO менее чем на половину не маскировано
[1]	RTIM	ЧТ/ЗП	Receive Timeout Interrupt Mask. Маска прерывания по превышению timeout-периода чтения из приемного FIFO (SSPRTINTR). 0 = прерывание замаскировано 1 = прерывание не маскировано
[0]	RORIM	ЧТ/ЗП	Receive Overrun Interrupt Mask. Маска прерывания по попытке записи в полное Rx FIFO: 0 = прерывание замаскировано 1 = прерывание не маскировано

5.14.3.7 Регистр состояния прерывания до наложения маски. SSPRIS

SSPRIS – регистр состояния прерывания до наложения маски, программно доступен только по чтению. При чтении регистр содержит текущее состояние прерываний до наложения маски. В таблице ниже приведен формат регистра SSPIMSC.

Таблица 5.110. Формат регистра SSPRIS

Биты	Название	Тип	Выполняемая функция
[15:4]	Зарезервировано	-	Зарезервирован, должен быть прописан нуль
[3]	TXRIS	ЧТ	Определяет состояние прерывания SSPTXINTR до наложения маски
[2]	RXRIS	ЧТ	Определяет состояние прерывания SSPRXINTR до наложения маски
[1]	RTRIS	ЧТ	Определяет состояние прерывания SSPRTINTR до наложения маски
[0]	RORRIS	ЧТ	Определяет состояние прерывания SSPRORINTR до наложения маски

5.14.3.8 Регистр состояния прерывания после наложения маски. SSPMIS

SSPMIS – Регистр состояния прерывания после наложения маски. Программно доступен только по чтению. При чтении регистр содержит текущее состояние прерываний после наложения маски. В таблице ниже приведен формат регистра SSPMIS.

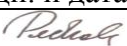
					ЮФКВ.431268.006РЭ			Лист
								157
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.111. Формат регистра SSPMIS

Биты	Название	Тип	Выполняемая функция
[15:4]	Зарезервировано	-	Зарезервирован, должен быть прописан нуль
[3]	TXMIS	ЧТ	Определяет состояние прерывания SSPTXINTR после наложения маски
[2]	RXMIS	ЧТ	Определяет состояние прерывания SSPRXINTR после наложения маски
[1]	RTMIS	ЧТ	Определяет состояние прерывания SSPRTINTR после наложения маски
[0]	RORMIS	ЧТ	Определяет состояние прерывания SSPRORINTR после наложения маски

5.14.3.9 Регистр сброса прерывания. SSPICR

SSPICR – регистр сброса прерывания, программно доступен только по записи. Запись единицы снимает соответствующее прерывание. Запись 0 не производит эффекта. В таблице ниже приведен формат регистра SSPICR.

Таблица 5.112. Формат регистра SSPMIS

Биты	Название	Тип	Выполняемая функция
[15:2]	Зарезервировано	-	Зарезервирован, должен быть прописан нуль
[1]	RTIC	ЗП	Снимает прерывание SSPRTINTR
[0]	RORIC	ЗП	Снимает прерывание SSPRORINTR

5.14.3.10 Регистр управления DMA. SSPDMACR

SSPDMACR – регистр управления DMA, программно доступен на запись и на чтение. При сбросе все биты устанавливаются в нуль. В таблице ниже приведен формат регистра SSPDMACR.

Таблица 5.113. Формат регистра SSPDMACR

Биты	Название	Тип	Выполняемая функция
[15:2]	Зарезервировано	-	Зарезервирован, должен быть прописан нуль
[1]	Transmit DMA Enable (TXDMAE)	ЧТ/ЗП	Если бит установлен в 1, то DMA-доступ к передающему FIFO разрешен.
[0]	Receive DMA Enable (RXDMAE)	ЧТ/ЗП	Если бит установлен в 1, то DMA-доступ к принимающему FIFO разрешен

5.14.3.11 Идентификатор периферийного устройства. SSPPeriphID0-3

Регистр SSPPeriphID0-3 представляет четыре 8-битных регистра, предназначенных только для чтения, которые охватывают адресное пространство от 0xFE0 до 0xFEC. Регистры могут быть концептуально представлены как 32-битный регистр. В таблице 5.10 приведен формат полей этого регистра.

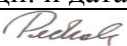
					ЮФКВ.431268.006РЭ			Лист
								158
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.114. Поля идентификатора периферийного устройства SSPPeriphID0-3

Биты	Выполняемые функции
PartNumber[11:0]	Содержит шифр компонента периферийного блока. Для SPI (PL022) 0x022
DesignerID[19:12]	Идентификационный номер проектировщика блока. Для ARM 0x41 (ASCII A)
Revision[23:20]	Является номером ревизии периферийного блока. Номер ревизии начинается с нуля
Configuration[31:24]	Является вариантом конфигурацией периферии. Для SPI (PL022) 0x0

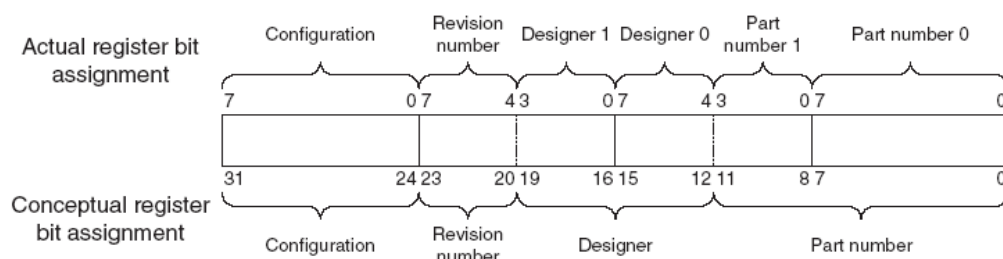


Рисунок 5.45 – Распределение полей регистра SSPPeriphID0-3

Обращение к периферийным регистрам должно быть 32-битным, независимо от их реальной разрядности.

5.14.3.12 PrimeCell идентификатор. SSPPCellID0-3

Регистры SSPPCellID0-3 – четыре 8-битных регистра, которые охватывают адресное пространство 0xFF0-0xFFC. Регистры, предназначенные только для чтения, могут быть концептуально представлены как 32-битный регистр. Регистр предназначен для идентификации стандартных периферийных блоков в составе системы. Регистр SSPPCellID0-3 установлен в 0xB105F00D. На рисунке 5.2 приведено распределение полей этого регистра.

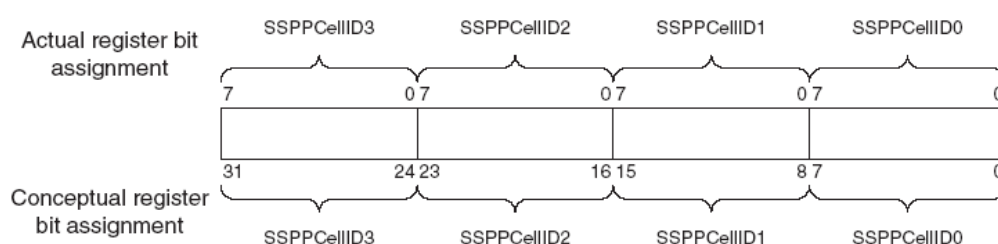


Рисунок 5.46 – Распределение полей регистра SSPPCellID0-3

5.14.4 Прерывания

Существует 5 прерываний, формируемых блоком SSP. Четыре из них являются маскируемыми с активным высоким уровнем:

					ЮФКВ.431268.006РЭ				Лист
									159
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

- **SSPRXINTR.** Прерывание по необходимости разгрузки принимающего FIFO. Это прерывание устанавливается, когда принимающее FIFO содержит четыре или более слов данных.
- **SSPTXINTR.** Прерывание по необходимости обслуживания передающего FIFO. Это прерывание устанавливается, когда передающее FIFO содержит менее четырех слов данных. Программист может работать с передающим FIFO двумя способами:
 2. Данные могут быть записаны в передающее FIFO до включения порта и до разрешения прерываний.
 3. Или, наоборот, можно сначала включить порт и разрешить прерывания – тогда данные в передающее FIFO должны записываться подпрограммой обслуживания прерываний
- **SSPRORINTR.** Прерывание по переполнению принимающего FIFO. Это прерывание устанавливается, когда приемное FIFO полное и происходит прием данных, приводя к переполнению FIFO. Причем данные перезаписываются не в FIFO, а в приемном сдвиговом регистре.
- **SSPRTINTR.** Прерывание по истечении времени ожидания приема данных (timeout). Это прерывание устанавливается, когда принятые данные находятся в приемном FIFO дольше 32 тактов. Этот механизм дополнительно оповещает программиста о том, что необходимо забрать данные из приемного FIFO. Прерывание SSPRTINTR снимается, когда принимающее FIFO становится пустым, или если новые данные получены с SPI шины. Прерывание также может быть снято путем записи бита RTIC=1 в регистре SSPICR.

Каждое из четырех прерываний может быть замаскировано путем установления соответствующих бит в регистре SSPIMSC. Состояние прерывания до и после наложения маски можно узнать, прочитав SSPRIS и SSPMIS регистры.

Пятое прерывание **SSPINTR** представляет собой общее комбинированное прерывание. Выставляется только одновременно с одним из вышеописанных прерываний. На контроллер прерываний VIC заведено только это прерывание от блока SSP.

					ЮФКВ.431268.006РЭ				Лист
									160
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редко</i> 16.01.14		22755-1						

5.15 Контроллер шины USB2.0

Контроллер USB обеспечивает аппаратную поддержку обмена данными по шине USB и соответствует спецификации устройства типа USB 2.0 Full-speed function (периферийное USB-устройство со скоростью обмена 12Мбит/с).

Контроллер имеет следующие характеристики:

- последовательный интерфейс (D+/D-),
- поддержка Remote Wakeup,
- режим с собственным питанием (Self-powered),
- программное управление подключением к хабу USB,
- 4 конечные точки (ENDPOINT) с настраиваемым максимальным размером пакета,
- из них 2 конечные точки типа BULK IN и BULK OUT, имеющие ПДП, а также состоящие каждая из двух буферов по 64 байта (теневого и программно-доступного), позволяющих максимально использовать скорость канала USB,
- аппаратная обработка стандартных запросов, исключая SET_DESCRIPTOR, GET_DESCRIPTOR, SYNCH_FRAME и специфичные запросы класса и разработчика (Class-Specific, Vendor-Specific),

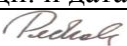
Контроллер поддерживает только конфигурации (Configuration Value) 0 (базовая, в которой доступна только конечная точка 0) и 1 (устанавливается хост-контроллером, в конфигурации 1 доступны все конечные точки). Можно программно настроить поддержку трёх интерфейсов (Interface): 0, 1 и 2 и четырёх Alternate Setting конечных точек.

Таблица 5.115 — Конфигурация конечных точек контроллера USB

Номер конечной точки	Configuration Value	Interface	Alternate Setting	Тип конечной точки	MaxPacketSize (байт)
0	--	--	--	CONTROL	8, 16, 32, 64
1	1	0, 1, 2	0, 1, 2, 3	BULK (OUT)	8, 16, 32, 64
2	1	0, 1, 2	0, 1, 2, 3	BULK (IN)	8, 16, 32, 64
3	1	0, 1, 2	0, 1, 2, 3	INTERRUPT (IN)	8, 16, 32, 64

Работу с контроллером USB можно разбить на 3 основных метода.

- Обработка прерывания от системного контроллера по подключению или отключению устройства от шины USB — при подключении необходимо произвести инициализацию контроллера и разрешить соединение с хабом USB. При инициализации настраиваются нужные в дальнейшей работе прерывания. При отключении целесообразно принудительно перевести контроллер в режим с пониженным энергопотреблением. Рекомендуемый алгоритм инициализации описан в разделах 5.15.7.1 и 5.15.7.2.
- Кольцевая проверка — эту функцию контроллера не обязательно использовать, она позволяет тестировать программный доступ к буферам данных. Алгоритм кольцевой проверки описан ниже в разделе 0.
- Обработка прерываний по событиям шины USB — здесь производится основная работа по передаче данных. Рекомендуемые алгоритмы приёма и передачи данных приведены в разделе 5.15.7.4.

					ЮФКВ.431268.006РЭ			Лист
								161
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.15.1 Устройство контроллера USB

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

Таблица 5.116 — Выводы микросхемы, входящие в состав контроллера USB

Вывод	Тип буфера	Примечание
USBCLK	in	Тактовый сигнал USB 48МГц
USBVP	inout	Сигнал D+ шины USB
USBVM	inout	Сигнал D- шины USB
USBCON	in	Сигнал наличия питания Vbus шины USB
TD9	out	Сигнал управления подтягивающим резистором линии D+

По сигналу USBCON можно программно определить, подключено ли устройство к хабу USB. Сигнал TD9 используется для управления подтягивающим резистором линии D+. Если выход TD9 имеет высокий уровень, то подтягивающий резистор отключен и линия D+ подтянута к уровню земли. В этом случае хаб USB не может обнаружить устройство на своём нисходящем порту. Чтобы хаб смог обнаружить устройство, следует привести вывод TD9 в активное состояние (низкий уровень). Внимание, вывод TD9 инвертирован по сравнению с битом USBPUCTRL системного контроллера ARMSC.

Ниже приведена блок-схема контроллера USB (см. Рисунок 5.47), на ней также показаны внешние выводы, относящиеся к контроллеру USB и части ARMSC, которая управляет подключением к USB.

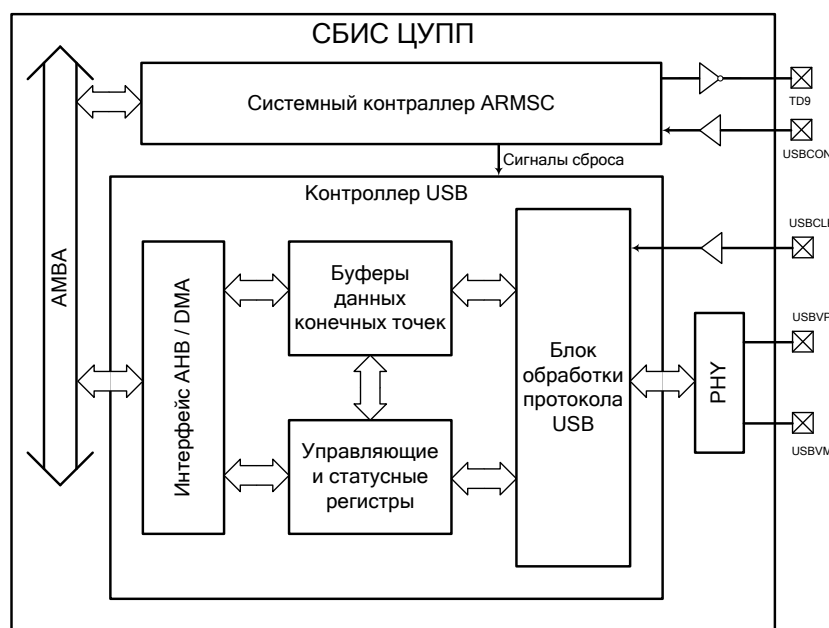


Рисунок 5.47 — Схема контроллера USB

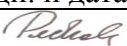
5.15.2 Регистровая модель контроллера USB

Спецификация регистров контроллера USB представлена в таблице ниже (Таблица 5.117).

					ЮФКВ.431268.006РЭ			Лист
								162
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

Таблица 5.117 — Спецификация регистров контроллера USB

Адрес [7:2]	Тип	Разрядность	Значение после инициализации	Имя	Описание
Base+ 00	ЧТ	16	----h	FIFO0o	Регистры приема/передачи данных
Base+ 04	ЗП	16	----h	FIFO0i	
Base+ 08	ЧТ	16	----h	FIFO1	
Base+ 0C	ЗП	16	----h	FIFO2	
Base+ 10	ЗП	16	----h	FIFO3	
Base+ 44	ЧТ/ЗП	[15:13]	000b	CONT1	Регистры управления
		[12:11]	-		
		[10]	0b		
		[9:5]	-		
		[4:0]	00000b		
Base+ 48	ЧТ/ЗП	16	[4:0] – 00000b	CONT2	
Base+ 4C	ЧТ/ЗП	16	[4:0] – 00000b	CONT3	
Base+ 50	ЧТ/ЗП	16	[4:0] – 00000b	CONT4	
Base+ 54	ЧТ/ЗП	16	[3:2] – 00b	CONT5	
Base+ 58	ЧТ/ЗП	16	[3:2] – 00b	CONT6	
Base+ 5C	ЧТ/ЗП	16	[4:0] – 00000b	CONT7	
Base+ 60	ЧТ/ЗП	[15:12]	-	CONT8	
		[11:8]	0000b		
		[7:5]	-		
		[4:0]	00000b		
Base+ 64	ЧТ/ЗП	[15]	0b	CONT9	
		[14:13]	-		
		[12]	0b		
		[11]	-		
		[10:7]	0000b		
		[6:4]	-		
		[3:0]	0000b		
Base+ 68	ЧТ/ЗП	[15:8]	00h	CONT10	
		[7]	-		
		[6:2]	00000b		
		[1:0]	-		
Base+ 6C	ЧТ/ЗП	16	1111h	TTSIZE	
Base+ 70	ЧТ/ЗП	16	1111h	TRSIZE	
Base+ 74	ЧТ/ЗП	16	[10:0] – 00000000000b	CONT11	
Base+ 80	ЧТ	16	[6:0] – 0000000b	RSIZE0	Регистры состояния
Base+ 88	ЧТ	16	[6:0] – 0000000b	RSIZE1	
Base+ C4	ЧТ/ЗП	16	[9:0] – 0000000000b	ST1	
Base+ D0	ЧТ	16	[6:0] – 0000000b	ST2	
Base+ D4	ЧТ/ЗП	[15:14]	00b	ST3	
		[7:0]	00b		
Base+ D8	ЧТ	16	[10:0] – 00000000000b	ST4	
Base+ DC	ЧТ/ЗП	[15]	0b	ST5	
		[14:13]	-		
		[12:11]	00b		
		[10]	-		
		[9:0]	0000000000b		
Base+ E0	ЧТ/ЗП	[15:12]	-	ST6	
		[11:6]	000000b		
		[5:4]	-		
		[3:1]	000b		
		[0]	-		

					ЮФКВ.431268.006РЭ			Лист
								163
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		 16.01.14		22755-1				

5.15.3 Регистры приема и передачи данных

Регистры приёма и передачи данных используются для доступа к буферам FIFO конечных точек контроллера USB. Запись и чтение производится 16-разрядными словами.

5.15.3.1 FIFO0o

Формат регистра FIFO0o приведён ниже.

Таблица 5.118 – Формат регистра FIFO0o

Биты	Тип	Название поля
[15:0]	ЧТ	FIFO0o

Поле **FIFO0o** (разряды 15-0) доступно только для чтения. Через регистр FIFO0o производится чтение 64-байтного приёмного буфера FIFO конечной точки 0 (CONTROL ENDPOINT). Данные с шины USB поступают в буфер пакетами, размер текущего пакета хранится в регистре RSIZE0. После приёма пакета первый байт принятых данных читается из бит 15-8 поля FIFO0o, второй байт читается из бит 7-0, третий байт читается из бит 15-8 в следующей операции чтения и так далее. При приеме пакета с нечетным количеством байтов последний байт пакета читается из бит 15-8, в этом случае биты 7-0 следует проигнорировать.

5.15.3.2 FIFO0i

Формат регистра FIFO0i приведён ниже.

Таблица 5.119 – Формат регистра FIFO0i

Биты	Тип	Название поля
[15:0]	ЗП	FIFO0i

Поле **FIFO0i** (разряды 15-0) доступно только для записи. Через регистр FIFO0i производится запись данных в 64-байтный передающий буфер FIFO конечной точки 0 (CONTROL ENDPOINT). Из записанных в буфер данных формируется один пакет, который потом забирается хост-контроллером USB. Перед записью последнего данного пакета необходимо установить признак конца пакета и чётность размера пакета. В случае, если необходимо сформировать пакет нечётного размера, устройство принимает в качестве последнего байта пакета данные из разрядов 15-8 записываемого слова, в этом случае биты 7-0 игнорируются.

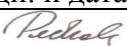
5.15.3.3 FIFO1

Формат регистра FIFO1 приведён ниже.

Таблица 5.120 – Формат регистра FIFO1

Биты	Тип	Название поля
[15:0]	ЗП	FIFO1

Поле **FIFO1** (разряды 15-0) доступно только для чтения. Через регистр FIFO1 производится чтение 64-байтного буфера FIFO конечной точки 1 (BULK IN ENDPOINT). Данные с шины USB поступают в буфер пакетами, размер текущего пакета хранится в регистре RSIZE1. После приёма пакета первый байт принятых данных читается из бит 15-8 поля FIFO1, второй байт читается из бит 7-0, третий байт читается из бит 15-8 в следующей операции чтения и так далее. При приеме пакета с нечетным количеством байтов последний байт пакета читается из бит 15-8, в этом случае биты 7-0 следует проигнорировать.

					ЮФКВ.431268.006РЭ			Лист
								164
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

В состав буфера конечной точки 1 входят два буфера по 64 байта: программно-доступный и теневой. Данные с шины USB записываются устройством сначала в теневой буфер, и при отправлении пакета ACK данные перемещаются в программно-доступный буфер.

5.15.3.4 FIFO2

Формат регистра FIFO2 приведён ниже.

Таблица 5.121 – Формат регистра FIFO2

Биты	Тип	Название поля
[15:0]	ЗП	FIFO2

Поле **FIFO2** (разряды 15-0) доступно только для записи. Через регистр FIFO2 производится запись данных в 64-байтный буфер FIFO конечной точки 2 (BULK IN ENDPOINT). Из записанных в буфер данных формируется один пакет, который потом забирается хост-контроллером USB. Перед записью последнего данного пакета необходимо установить признак конца пакета и чётность размера пакета. В случае, если необходимо сформировать пакет нечётного размера, устройство принимает в качестве последнего байта пакета данные из разрядов 15-8 записываемого слова, в этом случае биты 7-0 игнорируются.

В состав буфера конечной точки 2 входят два буфера по 64 байта: программно-доступный и теневой. Данные из программно-доступного буфера автоматически помещаются в теневой буфер после того, как в программно-доступный буфер будет записано последнее данное пакета и освободится теневой буфер.

Устройство посылает данные на шину USB только из теневого буфера. Если устройство получает запрос на передачу данных в момент, когда теневой буфер пуст, то на шину USB отправляется вместо данных пакет NAK (признак неготовности к передаче данных).

Теневой буфер освобождается в момент приёма конечной точкой 2 пакета подтверждения ACK от хост-контроллера USB.

5.15.3.5 FIFO3

Формат регистра FIFO3 приведён ниже.

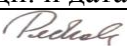
Таблица 5.122 – Формат регистра FIFO3

Биты	Тип	Название поля
[15:0]	ЗП	FIFO3

Поле **FIFO3** (разряды 15-0) доступно только для записи. Через регистр FIFO3 производится запись данных в 64-байтный буфер FIFO конечной точки 3 (INTERRUPT IN ENDPOINT). Из записанных в буфер данных формируется один пакет, который потом забирается хост-контроллером USB. Перед записью последнего данного пакета необходимо установить признак конца пакета и чётность размера пакета. В случае если необходимо сформировать пакет нечётного размера, устройство принимает в качестве последнего байта пакета данные из разрядов 15-8 записываемого слова, в этом случае биты 7-0 игнорируются.

5.15.4 Регистры состояния

Регистры состояния показывают текущее состояние контроллера USB, в том числе состояние запроса на прерывание.

					ЮФКВ.431268.006РЭ			Лист
								165
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

5.15.4.1 ST1

Формат регистра ST1 приведён ниже. При установке любого из бит регистра контроллер USB запрашивает прерывание. Все значимые биты регистра доступны для чтения и побитового сброса.

Таблица 5.123 – Формат регистра ST1

Биты	Тип	Название
[15:10]	-	-
[9]	ЧТ/ЗП	NACK3
[8]	ЧТ/ЗП	ACK3
[7]	ЧТ/ЗП	NACK2
[6]	ЧТ/ЗП	ACK2
[5]	ЧТ/ЗП	NACK1
[4]	ЧТ/ЗП	ACK1
[3]	ЧТ/ЗП	NACK0i
[2]	ЧТ/ЗП	ACK0i
[1]	ЧТ/ЗП	NACK0o
[0]	ЧТ/ЗП	ACK0o

0-й разряд (**ACK0o**) служит признаком завершения транзакции типа OUT с конечной точкой 0. Если эта причина прерывания не замаскирована, то бит ACK0o принимает значение 1 в момент, когда контроллер USB отправляет пакет подтверждения ACK.

1-й разряд (**NACK0o**) служит признаком ошибочного завершения транзакции типа OUT с конечной точкой 0. Если эта причина прерывания не замаскирована, то бит NACK0o принимает значение 1 по одной из следующих причин:

- устройство получило данные, когда оно было не готово к приему данных,
- устройство обнаружило ошибку в принятом пакете,
- истекло время ожидания данных от хост-контроллера USB (Time Out).

2-й разряд (**ACK0i**) служит признаком завершения транзакции типа IN с конечной точкой 0. Если эта причина прерывания не замаскирована, то бит ACK0i принимает значение 1 в момент получения пакета подтверждения ACK.

3-й разряд (**NACK0i**) служит признаком ошибочного завершения транзакции типа IN с конечной точкой 0. Если эта причина прерывания не замаскирована, то бит NACK0i принимает значение 1 по одной из следующих причин:

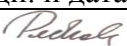
- устройство получило запрос данных, когда оно не было готово к передаче данных,
- устройство не получило подтверждения приема данных от хост-контроллера USB (Time Out).

4-й разряд (**ACK1**) служит признаком завершения транзакции с конечной точкой 1. Если эта причина прерывания не замаскирована, то бит ACK1 принимает значение 1 в момент, когда контроллер USB отправляет пакет подтверждения ACK.

5-й разряд (**NACK1**) служит признаком ошибочного завершения транзакции с конечной точкой 1. Если эта причина прерывания не замаскирована, то бит NACK1 принимает значение 1 по одной из следующих причин:

- устройство отправило пакет NAK в ответ на полученные данные,
- устройство обнаружило ошибку в принятом пакете,
- истекло время ожидания данных от хост-контроллера USB (Time Out).

6-й разряд (**ACK2**) служит признаком завершения транзакции с конечной точкой 2. Если эта причина прерывания не замаскирована, то бит ACK2 принимает значение 1 в момент получения пакета подтверждения ACK.

					ЮФКВ.431268.006РЭ			Лист
								166
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

7-й разряд (**NACK2**) служит признаком ошибочного завершения транзакции с конечной точкой 2. Если эта причина прерывания не замаскирована, то бит NACK2 принимает значение 1 по одной из следующих причин:

- устройство отправило пакет NAK в ответ на запрос данных,
- устройство не получило подтверждения приема данных от хост-контроллера USB (Time Out).

8-й разряд (**ACK3**) служит признаком завершения транзакции с конечной точкой 3. Если эта причина прерывания не замаскирована, то бит ACK3 принимает значение 1 в момент получения пакета подтверждения ACK.

9-й разряд (**NACK3**) служит признаком ошибочного завершения транзакции с конечной точкой 3. Если эта причина прерывания не замаскирована, то бит NACK3 принимает значение 1 по одной из следующих причин:

- устройство отправило пакет NAK в ответ на запрос данных,
- устройство не получило подтверждения приема данных от хост-контроллера USB (Time Out).

5.15.4.2 ST2

Формат регистра ST2 приведён ниже.

Таблица 5.124 – Формат регистра ST2

Биты	Название	Тип
[6:0]	ЧТ	DCT

Поле **DCT** (разряды 6-0) отображает размер последнего пакета данных (в байтах), прошедшего через устройство.

Биты модифицируются, когда транзакция USB завершается успешно.

5.15.4.3 ST3

Формат регистра ST3 приведён ниже.

Таблица 5.125 – Формат регистра ST3

Биты	Тип	Название
[15]	ЧТ	STDREQ2
[14]	ЧТ	STDREQ1
[13:8]	-	-
[7]	ЧТ/ЗП	PRECFEND
[6]	ЧТ/ЗП	EP0oNULL
[5]	ЧТ	CFGVAL
[4]	ЧТ/ЗП	SETUP
[3]	ЧТ	CFEND
[2]	ЧТ/ЗП	SOF
[1]	ЧТ	SUSP
[0]	-	

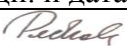
0-й разряд зарезервирован.

1-й разряд (**SUSP**) устанавливается, когда устройство приостановлено (Suspended State), и сбрасывается, когда устройство возобновляет работу (выходит из Suspended State).

2-й разряд (**SOF**) служит признаком того, что был получен пакет типа SOF.

3-й разряд (**CFEND**) служит признаком завершения инициализации устройства.

4-й разряд (**SETUP**) служит признаком того, что с конечной точкой 0 была совершена транзакция типа SETUP. Бит устанавливается при отправке подтверждения ACK. Бит не устанавливается для команд, которые автоматически обрабатываются

					ЮФКВ.431268.006РЭ			Лист
								167
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

устройством. Бит сбрасывается при завершении любой транзакции типа OUT (нет необходимости сбрасывать его программно).

5-й разряд (**CFGVAL**) показывает номер текущей конфигурации устройства (CONFIGURATION). Номер конфигурации устанавливается хост-контроллером USB при помощи запроса SET_CONFIGURATION.

6-й разряд (**EP0oNULL**) служит признаком того, что для ENDPOINT0 получен пустой пакет данных. Если бит установился, то он сохраняется в таком состоянии, пока в него не будет записано значение 0.

7-й разряд (**PRECFEND**) принимает значение 1 в момент завершения устройством обработки 64-х байт установочной информации. Установочная информация записывается в FIFO2 при программной инициализации устройства.

14-й разряд (**STDREQ1**) указывает на то, что контроллер USB запрашивает ПДП на запись.

15-й разряд (**STDREQ2**) указывает на то, что контроллер USB запрашивает ПДП на чтение.

В биты SUSP, SETUP, EP0oNULL, PRECFEND разрешена запись только значения 0.

5.15.4.4 ST4

Формат регистра ST4 приведён ниже.

Таблица 5.126 – Формат регистра ST4

Биты	Тип	Название
[10:0]	ЧТ	FMR

В поле **FMR** (разряды 10-0) хранится содержимое последнего принятого пакета SOF (номер фрейма USB).

5.15.4.5 ST5

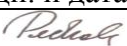
Формат регистра ST5 приведён ниже. При установке любого из бит регистра контроллер USB запрашивает прерывание. Биты регистра, вызывающие прерывание, доступны для чтения и, кроме STSTALL, для побитового сброса.

Таблица 5.127 – Формат регистра ST5

Биты	Тип	Название
[15]	ЧТ/ЗП	LPEND
[14:13]	-	-
[12]	ЧТ/ЗП	SETCFG
[11]	ЧТ/ЗП	USBRESET
[10]	-	-
[9]	ЧТ/ЗП	CLSTALL3
[8]	ЧТ/ЗП	CLSTALL2
[7]	ЧТ/ЗП	CLSTALL1
[6]	ЧТ/ЗП	CLSTALL0
[5]	ЧТ	STSTALL3
[4]	ЧТ	STSTALL2
[3]	ЧТ	STSTALL1
[2]	ЧТ	STSTALL0
[1]	ЧТ/ЗП	TRCVEND
[0]	ЧТ/ЗП	TTRSEND

0-й разряд (**TTRSEND**) служит признаком того, что была завершена запись передаваемых данных в FIFO2 при использовании счетчика TTSIZE.

1-й разряд (**TRCVEND**) служит признаком того, что было завершено чтение полученных данных из FIFO1 при использовании счетчика TRSIZE.

					ЮФКВ.431268.006РЭ			Лист
								168
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

2-й разряд (**STSTALL0**) служит признаком того, что конечная точка 0 остановлена (Stalled State).

3-й разряд (**STSTALL1**) служит признаком того, что конечная точка 1 остановлена (Stalled State).

4-й разряд (**STSTALL2**) служит признаком того, что конечная точка 2 остановлена (Stalled State).

5-й разряд (**STSTALL3**) служит признаком того, что конечная точка 3 остановлена (Stalled State).

6-й разряд (**CLSTALL0**) служит признаком того, что конечная точка 0 вышла из состояния остановки.

7-й разряд (**CLSTALL1**) служит признаком того, что конечная точка 1 вышла из состояния остановки.

8-й разряд (**CLSTALL2**) служит признаком того, что конечная точка 2 вышла из состояния остановки.

9-й разряд (**CLSTALL3**) служит признаком того, что конечная точка 3 вышла из состояния остановки.

11-й разряд (**USBRESET**) устанавливается, когда устройство обнаруживает сброс шины USB.

12-й разряд (**SETCFG**) устанавливается, когда устройство получает запрос SET_CONFIGURATION и отправляет подтверждение ACK.

15-й разряд (**LPEND**) устанавливается, когда завершается кольцевая проверка (loop-back).

При установке бита STSTALL соответствующий бит NACK регистра ST1 также устанавливается.

Для того чтобы сбросить бит TRCVEND, нужно сначала записать значение 0 в бит TRCNTEN регистра CONT10, затем записать в бит TRCVEND значение 1.

Для того чтобы сбросить бит TTRSEND, нужно сначала записать значение 0 в бит TTCNTEN регистра CONT10, затем записать в бит TTRSEND значение 1.

5.15.4.6 ST6

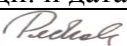
Формат регистра ST6 приведён ниже. При установке любого из бит ALTCHG контроллер USB запрашивает прерывание. Биты регистра, вызывающие прерывание, доступны для чтения и побитового сброса.

Таблица 5.128 – Формат регистра ST6

Биты	Тип	Название
[15:12]	-	-
[11:10]	ЧТ	ALT3
[9:8]	ЧТ	ALT2
[7:6]	ЧТ	ALT1
[5:4]	-	-
[3]	ЧТ/ЗП	ALTCHG3
[2]	ЧТ/ЗП	ALTCHG2
[1]	ЧТ/ЗП	ALTCHG1
[0]	-	-

1-й разряд (**ALTCHG1**) служит признаком того, что было изменено значение Alternate Setting конечной точки 1.

2-й разряд (**ALTCHG2**) служит признаком того, что было изменено значение Alternate Setting конечной точки 2.

					ЮФКВ.431268.006РЭ			Лист
								169
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

3-й разряд (**ALTCHG3**) служит признаком того, что было изменено значение Alternate Setting конечной точки 3.

Поле **ALT1** (разряды 7-6) показывает текущее значение Alternate Setting конечной точки 1.

Поле **ALT2** (разряды 9-8) показывает текущее значение Alternate Setting конечной точки 2.

Поле **ALT3** (разряды 11-10) показывает текущее значение Alternate Setting конечной точки 3.

5.15.4.7 **RSIZE0**

Формат регистра RSIZE0 приведён ниже.

Таблица 5.129 – Формат регистра RSIZE0

Биты	Тип	Название
[6:0]	ЧТ	RSIZE0

В поле **RSIZE0** (разряды 6-0) содержится размер в байтах принятого конечной точкой 0 пакета. Содержимое поля обновляется при отправке устройством подтверждения ACK.

5.15.4.8 **RSIZE1**

Формат регистра RSIZE1 приведён ниже.

Таблица 5.130 – Формат регистра RSIZE1

Биты	Тип	Название
[6:0]	ЧТ	RSIZE1

В поле **RSIZE1** (разряды 6-0) содержится размер в байтах принятого конечной точкой 1 пакета, который находится в программно-доступном буфере. Содержимое поля обновляется при поступлении нового пакета в программно-доступный буфер.

5.15.5 Регистры управления

Регистры управления используются для задания режимов работы, установки масок прерывания и управления состояниями буферов данных.

5.15.5.1 **CONT1**

Формат регистра CONT1 приведён ниже.

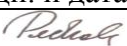
					ЮФКВ.431268.006РЭ			Лист
								170
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.131 – Формат регистра CONT1

Биты	Тип	Название
[15]	ЧТ/ЗП	LPSTART
[14]	ЧТ/ЗП	IODIS
[13]	ЧТ/ЗП	AUTOBFOK
[12:11]	-	-
[10]	ЧТ/ЗП	RESUM
[9:5]	-	-
[4]	ЧТ/ЗП	STALL3
[3]	ЧТ/ЗП	STALL2
[2]	ЧТ/ЗП	STALL1
[1]	ЧТ/ЗП	STALL0
[0]	ЧТ/ЗП	CFGEN

0-й разряд (**CFGEN**) управляет инициализацией контроллера USB. Запись значения 1 в этот бит запускает процесс инициализации устройства (в соответствии с записанной заранее в FIFO2 конфигурационной информацией). После инициализации любая запись в бит CFGEN никак не повлияет на устройство.

1-й разряд (**STALL0**) устанавливает конечную точку 0 в остановленное состояние (Stalled State).

2-й разряд (**STALL1**) устанавливает конечную точку 1 в остановленное состояние (Stalled State).

3-й разряд (**STALL2**) устанавливает конечную точку 2 в остановленное состояние (Stalled State).

4-й разряд (**STALL3**) устанавливает конечную точку 3 в остановленное состояние (Stalled State).

10-й разряд (**RESUM**) управляет функцией Remote Wakeup. Если устройство находится в приостановленном состоянии (Suspended State), то при записи в бит RESUM значения 1 запускается процесс выхода устройства из приостановленного состояния.

13-й разряд (**AUTOBFOK**) управляет автоматическим включением буфера конечной точки 0 на приём. Если AUTOBFOK=0, то даже при приёме пустого пакета буфер выключается для приёма новых данных и выставляется (если не замаскирован) бит BFOK0, который вызывает прерывание. Если AUTOBFOK=1, то при приёме устройством пустого пакета бит BFOK0 регистра CONT3 автоматически устанавливается в значение 1 и бит ACK0 не устанавливается в 1, не вызывая тем самым прерывание.

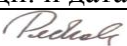
14-й разряд (**IODIS**) управляет принудительным вводом контроллера USB в приостановленное состояние (Suspended State).

15-й разряд (**LPSTART**) включает кольцевую проверку (loop-back). Запись в бит LPSTART значения 1 начинает кольцевую проверку (loop-back). Бит автоматически возвращается в значение 0 после окончания кольцевой проверки (loop-back).

Когда конечная точка, для которой установлен бит STALL_n, получает запрос передачи или приема данных, устройство отвечает на него пакетом STALL и соответствующий бит STALL сбрасывается.

Устройство реагирует нормально на получение пакета типа SETUP, даже если установлен бит STALL0. В этом случае бит STALL0 сбрасывается в 0, когда контроллер отвечает подтверждением ACK.

Имеет смысл оставлять контроллер в этом состоянии после завершения работы с шиной USB, так как в приостановленном состоянии снижается его энергопотребление.

					ЮФКВ.431268.006РЭ			Лист
								171
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

5.15.5.2 CONT2

Формат регистра CONT2 приведён ниже.

Таблица 5.132 – Формат регистра CONT2

Биты	Тип	Название
[4]	ЧТ/ЗП	INI3
[3]	ЧТ/ЗП	INI2
[2]	ЧТ/ЗП	INI1
[1]	ЧТ/ЗП	INI0i
[0]	ЧТ/ЗП	INI0o

0-й разряд (**INI0o**) управляет сбросом приёмного буфера конечной точки 0. Запись значения 1 в этот бит инициализирует приёмный буфер конечной точки 0, а также сбрасывает в 0 значения бит ACK0o и NACK0o регистра ST1, устанавливает значение 1 в бит BFOK0o регистра CONT3, сбрасывает регистр RSIZE0. Бит INI0o автоматически принимает значение 0, после того как в него была записана 1.

1-й разряд (**INI0i**) управляет сбросом приёмного буфера конечной точки 0. Запись значения 1 в этот бит инициализирует FIFO0i, а также сбрасывает в 0 значения бит ACK0i и NACK0i регистра ST1, бита BFOK0i регистра CONT3, бит LSTDi и ODD0i регистра CONT10. Бит INI0i автоматически принимает значение 0, после того как в него была записана 1.

2-й разряд (**INI1**) управляет сбросом буфера конечной точки 1. Запись значения 1 в этот бит инициализирует FIFO1, а также сбрасывает в 0 значения бит ACK1 и NACK1 регистра ST1, устанавливает значение 1 в бит BFOK1 регистра CONT3, сбрасывает регистры RSIZE1 и TRSIZE. Бит INI1 автоматически принимает значение 0, после того как в него была записана 1.

3-й разряд (**INI2**) управляет сбросом буфера конечной точки 2. Запись значения 1 в этот бит инициализирует FIFO2, а также сбрасывает в 0 значения бит ACK2 и NACK2 регистра ST1, бита BFOK2 регистра CONT3, бит LSTD2 и ODD2 регистра CONT10, сбрасывает регистр TTSIZE. Бит INI2 автоматически принимает значение 0, после того как в него была записана 1.

4-й разряд (**INI3**) управляет сбросом буфера конечной точки 3. Запись значения 1 в этот бит инициализирует FIFO3, а также сбрасывает в 0 значения бит ACK3 и NACK3 регистра ST1, бита BFOK3 регистра CONT3, бит LSTD3 и ODD3 регистра CONT10. Бит INI3 автоматически принимает значение 0, после того как в него была записана 1.

В биты INI разрешена запись только 1.

Бит INI сохраняет значение 1 на протяжении одного такта синхросигнала с частотой 12МГц. В это время операции записи/чтения не могут производиться с FIFO и регистрами, на которые бит INI влияет.

Прежде чем установить бит INI, необходимо убедиться, что его значение равно 0.

5.15.5.3 CONT3

Формат регистра CONT3 приведён ниже.

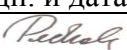
					ЮФКВ.431268.006РЭ			Лист
								172
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.133 – Формат регистра CONT3

Биты	Тип	Название
[4]	ЧТ/ЗП	BFOK3
[3]	ЧТ/ЗП	BFOK2
[2]	ЧТ/ЗП	BFOK1
[1]	ЧТ/ЗП	BFOK0i
[0]	ЧТ/ЗП	BFOK0o

0-й разряд (**BFOK0o**) включает приёмный буфер конечной точки 0. Для того чтобы устройство могло принять предназначенные для конечной точки 0 данные, следует предварительно записать в бит BFOK0o значение 1.

1-й разряд (**BFOK0i**) включает передающий буфер конечной точки 0. Для того чтобы устройство могло отправить записанные в FIFO0i данные, после записи очередного пакета данных в FIFO0i следует записывать в бит BFOK0i значение 1.

2-й разряд (**BFOK1**) включает буфер конечной точки 1. Для того чтобы устройство могло принять предназначенные для конечной точки 1 данные, следует предварительно записать в бит BFOK1 значение 1.

3-й разряд (**BFOK2**) включает буфер конечной точки 2. Для того чтобы устройство могло отправить записанные в FIFO2 данные, после записи очередного пакета данных в FIFO2 следует записывать в бит BFOK2 значение 1.

4-й разряд (**BFOK3**) включает буфер конечной точки 3. Для того чтобы устройство могло отправить записанные в FIFO3 данные, после записи очередного пакета данных в FIFO3 следует записывать в бит BFOK3 значение 1.

В любой из бит регистра CONT3 разрешена запись только значения 1.

После успешного завершения транзакции USB (при получении или отправке пакета ACK) соответствующий бит BFOK сбрасывается в 0.

Если бит BFOK имеет значение 0, то устройство всегда отменяет соответствующую транзакцию USB, посылая пакет NAK.

После программной инициализации устройства или после инициализации FIFO устанавливаются следующие значения BFOK:

- 0 для BFOK0i, BFOK2, BFOK3 (передающие FIFO),
- 1 для BFOK0o, BFOK1 (принимающие FIFO).

5.15.5.4 CONT4

Формат регистра CONT4 приведён ниже.

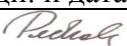
Таблица 5.134 – Формат регистра CONT4

Биты	Тип	Название
[4]	ЧТ/ЗП	FIFOBUSY3
[3]	ЧТ/ЗП	FIFOBUSY2
[2]	ЧТ/ЗП	FIFOBUSY1
[1]	ЧТ/ЗП	FIFOBUSY0i
[0]	ЧТ/ЗП	FIFOBUSY0o

0-й разряд (**FIFOBUSY0o**) устанавливает приёмный буфер конечной точки 0 в состояние «занятости»: если бит FIFOBUSY0o имеет значение 0, то транзакция типа OUT с конечной точкой 0 отменяется.

1-й разряд (**FIFOBUSY0i**) устанавливает передающий буфер конечной точки 0 в состояние «занятости»: если бит FIFOBUSY0i имеет значение 0, то транзакция типа IN с конечной точкой 0 отменяется.

2-й разряд (**FIFOBUSY1**) устанавливает буфер конечной точки 1 в состояние «занятости»: если используется программный доступ к буферу конечной точки 1 и бит

					ЮФКВ.431268.006РЭ			Лист
								173
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата
22755-2		 16.01.14		22755-1				

FIFOBUSY1 имеет значение 0, то транзакция с конечной точкой 1 отменяется. При переводе конечной точки 1 в режим работы с использованием ПДП необходимо предварительно записать в бит FIFOBUSY1 значение 0.

3-й разряд (**FIFOBUSY2**) устанавливает буфер конечной точки 2 в состояние «занятости»: если используется программный доступ к буферу конечной точки 2 и бит FIFOBUSY2 имеет значение 0, то транзакция с конечной точкой 2 отменяется. При переводе конечной точки 2 в режим работы с использованием ПДП необходимо предварительно записать в бит FIFOBUSY2 значение 0.

4-й разряд (**FIFOBUSY3**) устанавливает буфер конечной точки 3 в состояние «занятости»: если бит FIFOBUSY3 имеет значение 0, то транзакция с конечной точкой 3 отменяется.

5.15.5.5 CONT5

Формат регистра CONT5 приведён ниже.

Таблица 5.135 – Формат регистра CONT5

Биты	Тип	Название
[3]	ЧТ/ЗП	DFIFOBUSY2
[2]	ЧТ/ЗП	DFIFOBUSY1

2-й разряд (**DFIFOBUSY1**) устанавливает буфер конечной точки 1 в состояние «занятости»: если используется ПДП и бит DFIFOBUSY1 имеет значение 0, то транзакция с конечной точкой 1 отменяется. При переводе конечной точки 1 в режим работы с использованием программного доступа к буферу данных необходимо предварительно записать в бит DFIFOBUSY1 значение 0.

3-й разряд (**DFIFOBUSY2**) устанавливает буфер конечной точки 2 в состояние «занятости»: если используется ПДП и бит DFIFOBUSY2 имеет значение 0, то транзакция с конечной точкой 2 отменяется. При переводе конечной точки 2 в режим работы с использованием программного доступа к буферу данных необходимо предварительно записать в бит DFIFOBUSY2 значение 0.

5.15.5.6 CONT6

Формат регистра CONT6 приведён ниже. Биты регистра CONT6 маскируют запросы ПДП.

Таблица 5.136 – Формат регистра CONT6

Биты	Тип	Название
[3]	ЧТ/ЗП	MDREQ2
[2]	ЧТ/ЗП	MDREQ1

2-й разряд (**MDREQ1**) маскирует запрос на ПДП конечной точки 1. Если бит MDREQ1 имеет значение 0, то контроллер USB не может выдавать запрос ПДП на запись в память.

3-й разряд (**MDREQ2**) маскирует запрос на ПДП конечной точки 2. Если бит MDREQ2 имеет значение 0, то контроллер USB не может выдавать запрос ПДП на чтение из памяти.

5.15.5.7 CONT7

Формат регистра CONT7 приведён ниже. Биты регистра CONT7 маскируют запросы на прерывание регистра ST1.

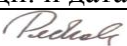
					ЮФКВ.431268.006РЭ			Лист
								174
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.137 – Формат регистра CONT7

Биты	Тип	Название
[4]	ЧТ/ЗП	МАСК3
[3]	ЧТ/ЗП	МАСК2
[2]	ЧТ/ЗП	МАСК1
[1]	ЧТ/ЗП	МАСК0i
[0]	ЧТ/ЗП	МАСК0o

0-й разряд (**МАСК0o**) маскирует прерывание по завершении транзакции типа OUT с конечной точкой 0. Если бит МАСК0o имеет значение 0, то бит АСК0o не может принять значение 1.

1-й разряд (**МАСК0i**) маскирует прерывание по завершении транзакции типа IN с конечной точкой 0. Если бит МАСК0i имеет значение 0, то бит АСК0i не может принять значение 1.

2-й разряд (**МАСК1**) маскирует прерывание по завершении транзакции с конечной точкой 1. Если бит МАСК1 имеет значение 0, то бит АСК1 не может принять значение 1.

3-й разряд (**МАСК2**) маскирует прерывание по завершении транзакции с конечной точкой 2. Если бит МАСК2 имеет значение 0, то бит АСК2 не может принять значение 1.

4-й разряд (**МАСК3**) маскирует прерывание по завершении транзакции с конечной точкой 3. Если бит МАСК3 имеет значение 0, то бит АСК3 не может принять значение 1.

5.15.5.8 CONT8

Формат регистра CONT8 приведён ниже. Биты регистра CONT8 маскируют запросы на прерывание регистров ST1 и ST5.

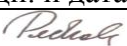
Таблица 5.138 – Формат регистра CONT8

Биты	Тип	Название
[15:12]	-	-
[11]	ЧТ/ЗП	MCLSTALL3
[10]	ЧТ/ЗП	MCLSTALL2
[9]	ЧТ/ЗП	MCLSTALL1
[8]	ЧТ/ЗП	MCLSTALL0
[7:5]	-	-
[4]	ЧТ/ЗП	MNACK3
[3]	ЧТ/ЗП	MNACK2
[2]	ЧТ/ЗП	MNACK1
[1]	ЧТ/ЗП	MNACK0i
[0]	ЧТ/ЗП	MNACK0o

0-й разряд (**MNACK0o**) маскирует прерывание по ошибочному завершению транзакции типа OUT с конечной точкой 0. Если бит MNACK0o имеет значение 0, то бит НАСК0o не может принять значение 1.

1-й разряд (**MNACK0i**) маскирует прерывание по ошибочному завершению транзакции типа IN с конечной точкой 0. Если бит MNACK0i имеет значение 0, то бит НАСК0i не может принять значение 1.

2-й разряд (**MNACK1**) маскирует прерывание по ошибочному завершению транзакции с конечной точкой 1. Если бит MNACK1 имеет значение 0, то бит НАСК1 не может принять значение 1.

					ЮФКВ.431268.006РЭ			Лист
								175
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

3-й разряд (**MNACK2**) маскирует прерывание по ошибочному завершению транзакции с конечной точкой 2. Если бит MNACK2 имеет значение 0, то бит NACK2 не может принять значение 1.

4-й разряд (**MNACK3**) маскирует прерывание по ошибочному завершению транзакции с конечной точкой 3. Если бит MNACK3 имеет значение 0, то бит NACK3 не может принять значение 1.

8-й разряд (**MCLSTALL0**) маскирует прерывание, возникающее по выходу конечной точки 0 из состояния остановки. Если бит MCLSTALL0 имеет значение 0, то бит CLSTALL0 не может принять значение 1.

9-й разряд (**MCLSTALL1**) маскирует прерывание, возникающее по выходу конечной точки 1 из состояния остановки. Если бит MCLSTALL1 имеет значение 0, то бит CLSTALL1 не может принять значение 1.

10-й разряд (**MCLSTALL2**) маскирует прерывание, возникающее по выходу конечной точки 2 из состояния остановки. Если бит MCLSTALL2 имеет значение 0, то бит CLSTALL2 не может принять значение 1.

11-й разряд (**MCLSTALL3**) маскирует прерывание, возникающее по выходу конечной точки 3 из состояния остановки. Если бит MCLSTALL3 имеет значение 0, то бит CLSTALL3 не может принять значение 1.

5.15.5.9 CONT9

Формат регистра CONT9 приведён ниже. Биты регистра CONT9 маскируют запросы на прерывание регистров ST5 и ST6.

Таблица 5.139 – Формат регистра CONT9

Биты	Тип	Название
[15]	ЧТ/ЗП	MLPEND
[14:13]	-	-
[12]	ЧТ/ЗП	MSETCFG
[11]	-	-
[10]	ЧТ/ЗП	MALTCHG3
[9]	ЧТ/ЗП	MALTCHG2
[8]	ЧТ/ЗП	MALTCHG1
[7]	ЧТ/ЗП	MUSBRESET
[6:4]	-	-
[3]	ЧТ/ЗП	MSTALL3
[2]	ЧТ/ЗП	MSTALL2
[1]	ЧТ/ЗП	MSTALL1
[0]	ЧТ/ЗП	MSTALL0

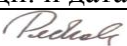
0-й разряд (**MSTALL0**) маскирует прерывание, причиной которого является установление 1 в бите STSTALL0.

1-й разряд (**MSTALL1**) маскирует прерывание, причиной которого является установление 1 в бите STSTALL1.

2-й разряд (**MSTALL2**) маскирует прерывание, причиной которого является установление 1 в бите STSTALL2.

3-й разряд (**MSTALL3**) маскирует прерывание, причиной которого является установление 1 в бите STSTALL3.

7-й разряд (**MUSBRESET**) маскирует прерывание по обнаружению сброса USB. Если бит MUSBRESET имеет значение 0, то бит USBRESET не может принять значение 1.

					ЮФКВ.431268.006РЭ			Лист
								176
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

8-й разряд (**MALTCHG1**) маскирует прерывание по изменению значения Alternate Setting конечной точки 1. Если бит MALTCHG1 имеет значение 0, то бит ALTCHG1 не может принять значение 1.

9-й разряд (**MALTCHG2**) маскирует прерывание по изменению значения Alternate Setting конечной точки 2. Если бит MALTCHG2 имеет значение 0, то бит ALTCHG2 не может принять значение 1.

10-й разряд (**MALTCHG3**) маскирует прерывание по изменению значения Alternate Setting конечной точки 3. Если бит MALTCHG3 имеет значение 0, то бит ALTCHG3 не может принять значение 1.

12-й разряд (**MSETCFG**) маскирует прерывание по получению запроса SET_CONFIGURATION. Если бит MSETCFG имеет значение 0, то бит SETCFG не может принять значение 1.

15-й разряд (**MLPEND**) маскирует прерывание по завершению кольцевой проверки. Если бит MLPEND имеет значение 0, то бит LPEND не может принять значение 1.

5.15.5.10 CONT10

Формат регистра CONT10 приведён ниже.

Таблица 5.140 – Формат регистра CONT10

Биты	Тип	Название
[15]	ЧТ/ЗП	ALWAYS1
[14]	ЧТ/ЗП	ODD3
[13]	ЧТ/ЗП	ODD2
[12]	ЧТ/ЗП	ODD0
[11]	ЧТ/ЗП	NULLSET3
[10]	ЧТ/ЗП	NULLSET2
[9]	ЧТ/ЗП	NULLSET0
[8]	ЧТ/ЗП	ALWAYS1
[7]	-	-
[6]	ЧТ/ЗП	LSTD3
[5]	ЧТ/ЗП	LSTD2
[4]	ЧТ/ЗП	LSTD0
[3]	ЧТ/ЗП	TRCNTEN
[2]	ЧТ/ЗП	TTCNTEN
[1:0]	-	-

2-й разряд (**TTCNTEN**) включает счётчик передачи данных (TTSIZE). 1 – TTSIZE включен, 0 – выключен.

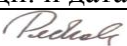
3-й разряд (**TRCNTEN**) включает счётчик приёма данных (TRSIZE). 1 – TRSIZE включен, 0 – выключен.

4-й разряд (**LSTD0**) – признак последнего данного пакета. Запись в бит LSTD0 значения 1 сообщает устройству о том, что следующая запись в FIFO0i будет содержать последние данные пакета.

5-й разряд (**LSTD2**) – признак последнего данного пакета. Запись в бит LSTD2 значения 1 сообщает устройству о том, что следующая запись в FIFO2 будет содержать последние данные пакета.

6-й разряд (**LSTD3**) – признак последнего данного пакета. Запись в бит LSTD3 значения 1 сообщает устройству о том, что следующая запись в FIFO3 будет содержать последние данные пакета.

В 8-й разряд (**ALWAYS1**) при записи всегда необходимо записывать 1.

					ЮФКВ.431268.006РЭ			Лист
								177
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

9-й разряд (**NULLSET0**) принимает значение 1, когда из передающего буфера конечной точки 0 передаётся пустой пакет данных, исключая случай стадии проверки состояния в рамках управляющей передачи (Control Transfer Status Stage).

10-й разряд (**NULLSET2**) принимает значение 1, когда из передающего буфера конечной точки 2 передаётся пустой пакет данных.

11-й разряд (**NULLSET3**) принимает значение 1, когда из передающего буфера конечной точки 3 передаётся пустой пакет данных.

12-й разряд (**ODD0**) – признак нечётности размера пакета для конечной точки 0.

13-й разряд (**ODD2**) – признак нечётности размера пакета для конечной точки 2.

14-й разряд (**ODD3**) – признак нечётности размера пакета для конечной точки 3.

15-й разряд (**ALWAYS1**) при записи всегда необходимо записывать 1.

Запись в бит ODD_n (n = 0, 2, 3) значения 1 сообщает устройству о том, что следующая запись в FIFO_n будет содержать только один старший байт данных. Если требуется передать пакет нечётного размера, то непосредственно перед записью последнего байта в FIFO_n следует записать в бит ODD_n значение 1.

После записи в FIFO_n последних данных пакета бит ODD_n автоматически сбрасывается.

Когда устанавливается бит ACK регистра ST1, соответствующий бит LSTD автоматически сбрасывается.

Запись значений 0 в биты LSTD и ODD никак не влияет на устройство.

5.15.5.11 CONT11

Формат регистра CONT11 приведён ниже.

Таблица 5.141 – Формат регистра CONT11

Биты	Название	Тип
[15:11]	-	-
[10:8]	NUMALTIF2	ЧТ/ЗП
[7:5]	NUMALTIF1	ЧТ/ЗП
[4:2]	NUMALTIF0	ЧТ/ЗП
[1:0]	NUMINTF	ЧТ/ЗП

Поле **NUMINTF** (разряды 1-0) задаёт количество интерфейсов контроллера (INTERFACE в понимании спецификации USB 2.0). Например, если требуется только нулевой интерфейс, следует записать в это поле значение 1, если требуется поддерживать нулевой и первый интерфейсы, то следует записать «2».

Поле **NUMALTIF0** (разряды 4-2) задаёт число Alternate Setting интерфейса 0.

Поле **NUMALTIF1** (разряды 7-5) задаёт число Alternate Setting интерфейса 0.

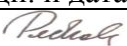
Поле **NUMALTIF2** (разряды 10-8) задаёт число Alternate Setting интерфейса 0.

Поля NUMALTIF_n устанавливают число альтернативных состояний (Alternate Setting) для каждого интерфейса (INTERFACE).

Для того чтобы поддерживать только ALTERNATE0 интерфейса INTERFACE_n (n = 0, 1, 2) следует в поле NUMALTIF_n[2:0] записать число 1.

Для того чтобы поддерживать только ALTERNATE0 и ALTERNATE1 интерфейса INTERFACE_n (n = 0, 1, 2) следует в поле NUMALTIF_n[2:0] записать число «2».

Для того чтобы поддерживать минимальный набор интерфейсов и альтернативных состояний следует при инициализации устройства записать в регистр CONT11 значение «5».

					ЮФКВ.431268.006РЭ			Лист
								178
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.15.5.12 TTSIZE

Формат регистра TTSIZE приведен ниже.

Таблица 5.142 – Формат регистра TTSIZE

Биты	Тип	Название поля
[15:0]	ЧТ/ЗП	TTSIZE

В поле **TTSIZE** (разряды 15-0) отображается счётчик передаваемых данных в байтах конечной точкой 2. Счётчик считает в обратную сторону (в сторону уменьшения своего значения). Начальное значение счётчика и значение после сброса буфера конечной точки 2 — 1111h. Если TTSIZE используется (бит TTCNTEN регистра CONT10 имеет значение 1), то при каждой записи в FIFO2 значение TTSIZE уменьшается на 2. Когда счётчик достигает значения 0, выдаётся запрос на прерывание. Нельзя записывать в поле TTSIZE значение 0.

Не следует осуществлять запись в FIFO2, если TTSIZE установлен в значение 0000h и бит TTCNTEN регистра CONT10 установлен в 1. Запись в регистр TTSIZE следует осуществлять, только когда бит MDREQ2 имеет значение 0.

5.15.5.13 TRSIZE

Формат регистра TRSIZE приведен ниже.

Таблица 5.143 – Формат регистра TRSIZE

Биты	Название поля	Тип
[15:0]	TRSIZE	ЧТ/ЗП

В поле **TRSIZE** (разряды 15-0) отображается счётчик принятых данных в байтах конечной точкой 1. Счётчик считает в обратную сторону (в сторону уменьшения своего значения). Начальное значение счётчика и значение после сброса буфера конечной точки 1 – 1111h. Если TRSIZE используется (бит TRCNTEN регистра CONT10 имеет значение 1), то при каждом чтении из FIFO1 значение TRSIZE уменьшается на 2. Когда счётчик достигает значения 0, выдаётся запрос на прерывание. Нельзя записывать в поле TRSIZE значение 0.

Не следует осуществлять чтение из FIFO1, если TRSIZE установлен в значение 0000h и бит TTCNTEN регистра CONT10 установлен в 1. Запись в регистр TRSIZE следует осуществлять, только когда бит MDREQ1 имеет значение 0.

5.15.6 Прерывания

Контроллер USB имеет один комбинированный выход прерывания, подключенный к системному контроллеру прерываний VIC. Запрос на прерывание активен, если хотя бы один из бит регистров ST1, ST5 или ST6 имеет значение 1. Для того чтобы снять запрос на прерывание, необходимо или замаскировать запрос, или сбросить бит, ставший причиной прерывания, произведя запись значения 1. Ниже приведена таблица со списком причин прерывания.

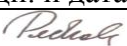
					ЮФКВ.431268.006РЭ			Лист
								179
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 5.144 — Прерывания от контроллера USB

Причина прерывания	Маска прерывания
Регистр ST1 биты ACK	Регистр CONT7 биты MACK
Регистр ST1 биты NACK	Регистр CONT8 биты MNACK
Регистр ST5 бит TTRSEND	нет
Регистр ST5 бит TRCVEND	нет
Регистр ST5 биты STSTALL	Регистр CONT9 биты MSTALL
Регистр ST5 бит USBRESET	Регистр CONT9 бит MUSBRESET
Регистр ST5 бит SETCFG	Регистр CONT9 бит MSETCFG
Регистр ST5 бит LPEND	Регистр CONT9 бит MLPEND
Регистр ST5 биты CLSTALL	Регистр CONT8 биты MCLSTALL
Регистр ST6 биты ALTCHG	Регистр CONT9 биты MALTCHG

5.15.7 Рекомендуемые алгоритмы работы с контроллером USB

Описанные в данном разделе алгоритмы рекомендуются, но не являются обязательными.

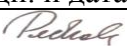
5.15.7.1 Подключение и отключение

Факт подключения устройства к хабу USB детектируется по наличию или отсутствию высокого уровня на внешнем выводе USBCON. Системный контроллер СБИС (ARMSC) позволяет отслеживать изменение состояния вывода USBCON. При подключении устройства к USB или отключении от USB генерируется прерывание USBCONINT.

При подключении необходимо с помощью системного контроллера производить программный сброс и инициализацию контроллера USB.

При отключении устройства от USB рекомендуется выключать контроллер путем снятия тактового сигнала. Это можно сделать, записав значение 4000h в регистр CONT1. После сброса контроллера подача тактового сигнала на блок возобновляется.

В показанном ниже алгоритме обработки прерывания USBCONINT используются биты регистров системного контроллера ARMSC.

					ЮФКВ.431268.006РЭ			Лист
								180
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

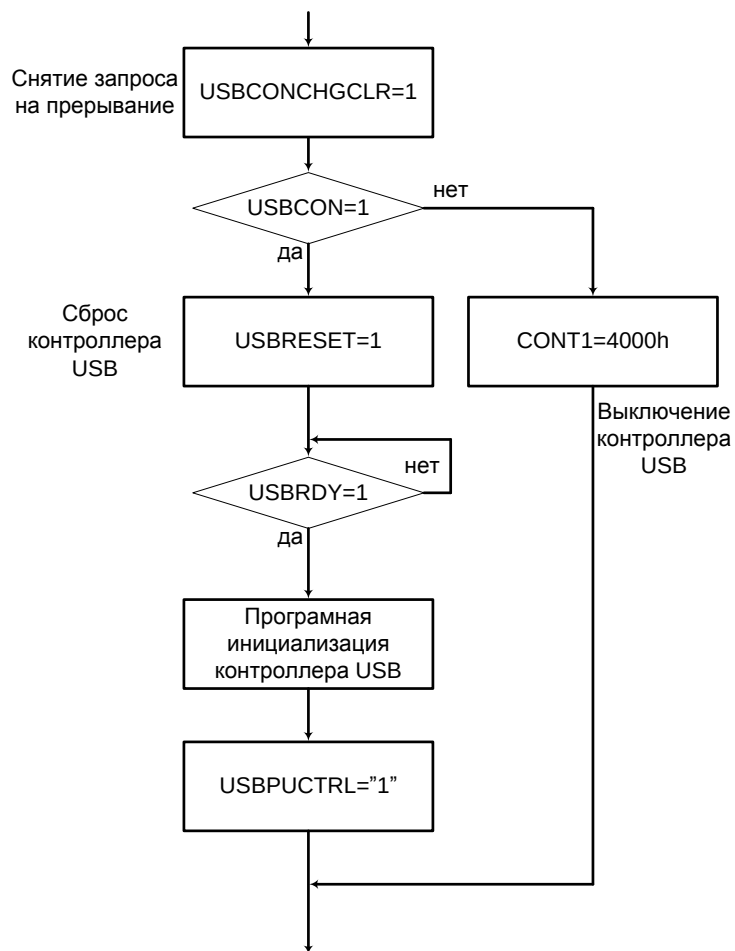


Рисунок 5.48 — Пример обработки прерывания *USBCONINT*

При работе с USB может понадобиться «переподключение» например, для того, чтобы поменять класс устройства. Эта функция реализуется с помощью управляемого Pull-Up резистора на плате. «Переподключение» следует проводить по показанному ниже алгоритму.

					ЮФКВ.431268.006РЭ			Лист
								181
Изм.	Лист	№ докум.	Подп.	Дата				
Инд.№подл.	Подп. и дата		Взам.инв.№		Инд.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

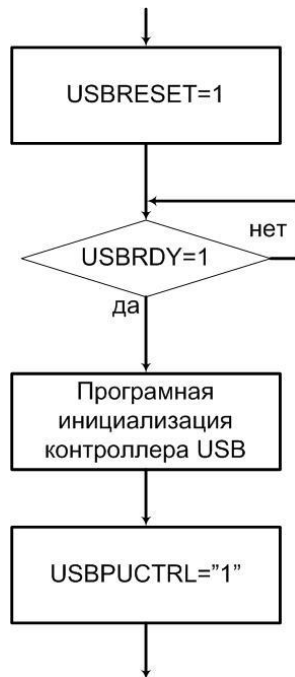


Рисунок 5.49 — Программное «переподключение» устройства к USB

Если СБИС была подключена к хабу USB, то описанные действия будут восприняты хост-контроллером USB как отключение устройства от шины и повторное подключение. Необходимый по спецификации USB 2.0 интервал регистрации отключения (TDDIS=2,5 мкс) выдерживается в данном алгоритме, так как программная инициализация контроллера USB длится дольше.

5.15.7.2 Программная инициализация

Программная инициализация проводится после сброса контроллера USB, после системного сброса или программного сброса с помощью системного контроллера ARMSC. Программная инициализация заключается в записи 65 байт конфигурационной информации в буфер конечной точки 2 и разрешения необходимых в дальнейшем прерываний.

Рекомендуется следующий алгоритм инициализации:

- 1) записать 64 байта конфигурационной информации в регистр FIFO2,
- 2) записать значение 1 в бит CFGEN регистра CONT1,
- 3) читать регистр ST3 до тех пор, пока бит PRECFEND не примет значение 1,
- 4) записать значение 0 в бит PRECFEND,
- 5) записать последнее данное в FIFO2,
- 6) читать регистр ST3 до тех пор, пока бит CFEND не примет значение 1,
- 7) записать в регистр CONT11 информацию о конфигурации интерфейсов,
- 8) разрешить используемые в дальнейшем прерывания (записать значение 1 в соответствующие биты регистров CONT7, CONT8 и CONT9).

Конфигурационная информация представляет собой массив из 65 байтов, имеющий поля, которые воспринимаются устройством как параметры конечных точек контроллера. Массив конфигурационной информации можно представить в виде таблицы.

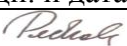
					ЮФКВ.431268.006РЭ			Лист
								182
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

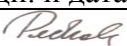
Таблица 5.145 — Конфигурационная информация контроллера USB

Конечная точка	Alternate Setting	Конфигурационный массив данных
ENDPOINT0		0000000000000000[EP0MAXPACKETSIZE]0100000000000000
ENDPOINT1	ALTERNATE0	000101[EP1INTERFACE]00100000[EP1ALT0MAXPACKETSIZE]0100000000000001
	ALTERNATE1	000101[EP1INTERFACE]01100000[EP1ALT1MAXPACKETSIZE]0100000000000001
	ALTERNATE2	000101[EP1INTERFACE]10100000[EP1ALT2MAXPACKETSIZE]0100000000000001
	ALTERNATE3	000101[EP1INTERFACE]11100000[EP1ALT3MAXPACKETSIZE]0100000000000001
ENDPOINT2	ALTERNATE0	001001[EP2INTERFACE]00101000[EP2ALT0MAXPACKETSIZE]0100000000000010
	ALTERNATE1	001001[EP2INTERFACE]01101000[EP2ALT1MAXPACKETSIZE]0100000000000010
	ALTERNATE2	001001[EP2INTERFACE]10101000[EP2ALT2MAXPACKETSIZE]0100000000000010
	ALTERNATE3	001001[EP2INTERFACE]11101000[EP2ALT3MAXPACKETSIZE]0100000000000010
ENDPOINT3	ALTERNATE0	001101[EP3INTERFACE]00111000[EP3ALT0MAXPACKETSIZE]0100000000000011
	ALTERNATE1	001101[EP3INTERFACE]01111000[EP3ALT1MAXPACKETSIZE]0100000000000011
	ALTERNATE2	001101[EP3INTERFACE]10111000[EP3ALT2MAXPACKETSIZE]0100000000000011
	ALTERNATE3	001101[EP3INTERFACE]11111000[EP3ALT3MAXPACKETSIZE]0100000000000011

Запись массива производится начиная с первой строки данной таблицы. Каждая строка состоит из байтов, младший байт строки справа; записывать следует начиная со старшего байта. Описание конфигурационных параметров дано ниже.

Таблица 5.146 — Конфигурационные параметры контроллера USB

Параметр и его размерность	Описание	Допустимые значения
EP1INTERFACE[1:0]	Номер интерфейса	0, 1, 2
EP2INTERFACE[1:0]	Номер интерфейса	0, 1, 2
EP3INTERFACE[1:0]	Номер интерфейса	0, 1, 2
EP0MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 0	8, 16, 32, 64
EP1ALT0MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 1 Alternate Setting 0	8, 16, 32, 64
EP1ALT1MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 1 Alternate Setting 1	8, 16, 32, 64
EP1ALT2MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 1 Alternate Setting 2	8, 16, 32, 64
EP1ALT3MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 1 Alternate Setting 3	8, 16, 32, 64
EP2ALT0MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 2 Alternate Setting 0	8, 16, 32, 64
EP2ALT1MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 2 Alternate Setting 1	8, 16, 32, 64
EP2ALT2MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 2 Alternate Setting 2	8, 16, 32, 64
EP2ALT3MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 2 Alternate Setting 3	8, 16, 32, 64
EP3ALT0MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 3 Alternate Setting 0	8, 16, 32, 64
EP3ALT1MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 3 Alternate Setting 1	8, 16, 32, 64
EP3ALT2MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 3 Alternate Setting 2	8, 16, 32, 64
EP3ALT3MAXPACKETSIZE[6:0]	Максимальный размер пакета для конечной точки 3 Alternate Setting 3	8, 16, 32, 64

					ЮФКВ.431268.006РЭ			Лист
								183
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

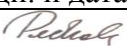
На примере (Таблица 5.147) показано, как реализовать конфигурацию с одним простым интерфейсом и максимальным размером пакета 8 байт для конечных точек 0 и 3 и 64 байта для 1 и 2.

Таблица 5.147 — Пример конфигурации контроллера USB

Конечная точка	Alternate Setting	Конфигурационный массив данных
ENDPOINT0		0000_0000_0000_0000_[1000_000]0_1000_0000_0000_0000
ENDPOINT1	ALTERNATE0	0001_01[00]_0010_0000_[1000_000]0_1000_0000_0000_0001
	ALTERNATE1	0001_01[00]_0110_0000_[1000_000]0_1000_0000_0000_0001
	ALTERNATE2	Такой же набор, как для конечной точки 0
	ALTERNATE3	Такой же набор, как для конечной точки 0
ENDPOINT2	ALTERNATE0	0010_01[00]_0010_1000_[1000_000]0_1000_0000_0000_0010
	ALTERNATE1	Такой же набор, как для конечной точки 0
	ALTERNATE2	Такой же набор, как для конечной точки 0
	ALTERNATE3	Такой же набор, как для конечной точки 0
ENDPOINT3	ALTERNATE0	0011_01[00]_0011_1000_[1000_000]0_1000_0000_0000_0011
	ALTERNATE1	Такой же набор, как для конечной точки 0
	ALTERNATE2	Такой же набор, как для конечной точки 0
	ALTERNATE3	Такой же набор, как для конечной точки 0

Показанный массив записывается в регистр FIFO2 16-разрядными словами (поле FIFO2 имеет разрядность 16 бит), ниже показано, как конфигурационный массив разбит на слова.

- 0000h //EP0
- 8080h //EP0
- 0014h //EP0/EP1A0
- 2080h //EP1A0
- 8001h //EP1A0
- 1420h //EP1A1
- 8080h //EP1A1
- 0114h //EP1A1/EP1A2
- 2080h //EP1A2
- 8001h //EP1A2
- 1420h //EP1A3
- 8080h //EP1A3
- 0124h //EP1A3/EP2A0
- 2880h //EP2A0
- 8002h //EP2A0
- 2428h //EP2A1
- 8080h //EP2A1
- 0224h //EP2A1/EP2A2
- 2880h //EP2A2
- 8002h //EP2A2
- 2428h //EP2A3
- 8080h //EP2A3
- 0234h //EP2A3/EP3A0
- 3880h //EP3A0
- 8003h //EP3A0
- 3438h //EP3A1

					ЮФКВ.431268.006РЭ			Лист
								184
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

- 8080h //EP3A1
- 0334h //EP3A1/EP3A2
- 3880h //EP3A2
- 8003h //EP3A2
- 3438h //EP3A3
- 8080h //EP3A3
- 0300h //EP3A3/x

5.15.7.3 Кольцевая проверка

Тест кольцевой проверки можно проводить для тестирования контроллера на предмет работоспособности. В этом тесте не нужно подключение к хост-контроллеру: данные сначала записываются в передающий буфер, устройство само переписывает данные в приёмный буфер, откуда они могут быть прочитаны. В тесте кольцевой проверки может быть использована одна из двух пар буферов: FIFO0i и FIFO0o или FIFO2 и FIFO1. Алгоритм кольцевой проверки:

- 1) приёмный и передающий буфер следует сбросить с помощью регистра CONT2,
- 2) в передающий буфер (FIFO0i либо FIFO2) записать данные (количество их должно быть не более указанного в конфигурационном массиве максимального размера пакета),
- 3) включить прерывание по завершению кольцевой проверки (бит MLPEND регистра CONT9),
- 4) записать значение 1 в бит LPSTART регистра CONT1,
- 5) читать регистр ST5 до тех пор, пока бит LPEND не примет значение 1,
- 6) из приёмного буфера (FIFO0o либо FIFO1 соответственно) прочитать такое же количество данных и сравнить с записанными.

Прочитанные данные должны совпасть с записанными.

5.15.7.4 Обработка прерываний

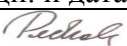
Описанные ниже алгоритмы выполняются в процедуре обработки прерывания от контроллера USB. Контроллер USB держит запрос на прерывание, если хотя бы один из бит регистров ST1, ST5 или ST6 имеет значение 1.

5.15.7.5 Обработка прерывания по сбросу шины USB

Запрос на прерывание по сбросу шины USB (бит USBRESET) возникает, когда устройство обнаруживает на шине USB сигнал сброса (состояние Single-Ended 0 в течение 3 мкс). Рекомендуется следующая процедура обработки:

- 1) сбросить бит USBRESET, записав в него значение 1,
- 2) сбросить буферы контроллера, записав в биты INI регистра CONT2 значение 1

если есть необходимость использовать ПДП, то

					ЮФКВ.431268.006РЭ			Лист
								185
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

- 3) записать FFFFh в регистр CONT5 и 0013h в регистр CONT4,
если же ПДП не нужен, то
- 3) вывести все конечные точки из состояния «занятости», записав значение FFFFh в регистр CONT4.

5.15.7.6 Обработка прерывания по завершении транзакции с конечной точкой 0

Управляющие транзакции проводятся с конечной точкой 0 хост-контроллером USB в процессе нумерации. Бит ACK0o сигнализирует о завершении транзакции типа OUT или SETUP с конечной точкой 0. Рекомендуется следующая процедура обработки:

- 1) сбросить бит ACK0o, записав в него значение 1,
- 2) прочитать регистр RSIZE0 для того, чтобы определить размер полученного пакета,
- 3) прочитать бит SETUP регистра ST3, чтобы определить фазу стандартного запроса,
- 4) прочитать полученные данные из FIFO0o, и если SETUP=1, декодировать стандартный запрос и составить ответ (если используется регистр FIFO0i, то запись в него следует производить в соответствии с правилами записи в этот регистр),
- 5) записать в бит BFOK0o значение 1 для того, чтобы устройство могло принять следующую транзакцию с конечной точкой 0.

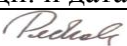
Бит ACK0i сигнализирует о завершении транзакции типа IN с конечной точкой 0. Обработка прерывания ACK0i может понадобиться, например, если размер передаваемого хост-контроллеру дескриптора больше 64-х байт. Рекомендуется следующая процедура обработки:

- 1) сбросить бит ACK0i, записав в него значение 1,
- 2) если требуется дописать очередную часть дескриптора, то записать её в FIFO0i,
- 3) записать значение 1 в бит BFOK0i.

5.15.7.7 Обработка прерывания по получению запроса SET_CONFIGURATION.

Запрос SET_CONFIGURATION автоматически обрабатывается контроллером, но по его получению контроллер USB может выдавать запрос на прерывание (бит SETCFG). Рекомендуется следующая процедура обработки:

- 1) сбросить бит SETCFG, записав в него значение 1,
- 2) прочитать бит CFGVAL регистра ST3,
если CFGVAL=1, то
- 3) включить буфер конечной точки 1, записав значение 1 в бит BFOK1,
- 4) если есть необходимость использовать ПДП, то здесь следует подготовить контроллер ПДП и демаскировать запросы на ПДП, записав значение «FFFFh» в регистр CONT6,
- 5) при необходимости записать данные в буферы конечных точек 2 и 3,
если же CFGVAL=0, то
- 3) следует сбросить буферы конечных точек 1, 2 и 3, записав значение 1 в биты INI регистра CONT2.

					ЮФКВ.431268.006РЭ			Лист
								186
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.15.7.8 Обработка прерывания по завершении транзакции с конечной точкой 1

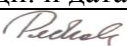
По завершении транзакции с конечной точкой 1 устройство может выдавать запрос на прерывание (АСК1). Это означает, что в буфере FIFO1 появились данные, и в случае, когда не используется ПДП, следует прочитать их. Рекомендуется следующая процедура обработки:

- 1) сбросить бит АСК1, записав в него значение 1,
- 2) прочитать размер принятого пакета из регистра RSIZE1,
- 3) прочитать данные из FIFO1,
- 4) записать в бит ВFOK1 значение 1 для того, чтобы устройство могло принять следующую транзакцию с конечной точкой 1.

5.15.7.9 Обработка прерывания по завершении транзакции с конечными точками 2 и 3

По завершении транзакции с конечными точками 2 или 3 устройство может выдавать запрос на прерывание (АСК2 или АСК3). Это означает, что записанные предварительно в передающий буфер данные были получены хост-контроллером, следовательно, буфер освободился для записи в него следующих данных. Рекомендуется следующая процедура обработки:

- 1) сбросить бит АСК2 (или АСК3), записав в него значение 1,
если используется ПДП, то
- 2) запустить канал ПДП на передачу контроллеру USB данных в количестве равном максимальному размеру пакета конечной точки 2,
если же ПДП не используется, то
- 2) записать в FIFO2 (или FIFO3) данные в соответствии с правилами записи в этот регистр,
- 3) записать значение 1 в регистр ВFOK2 (или ВFOK3).

					ЮФКВ.431268.006РЭ			Лист
								187
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.16 Блок портов общего назначения GPIO

Блок портов общего назначения GPIO в СБИС СПОВС имеет следующие характеристики:

- 16 портов общего назначения, разделенных на две группы по 8 разрядов
- подключение к системной шине СБИС через AMBA APB интерфейс, используются 8 младших бит шины данных

5.16.1 Устройство портов общего назначения

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

Таблица 5.148. Выводы микросхемы, входящие в состав SPI порта

Вывод	Тип буфера	Примечание
GPIO0 ... GPIO15	inout	16 портов общего назначения

На рисунке (см. Рисунок 5.50) приведена внутренняя структура 1 бита блока портов общего назначения.

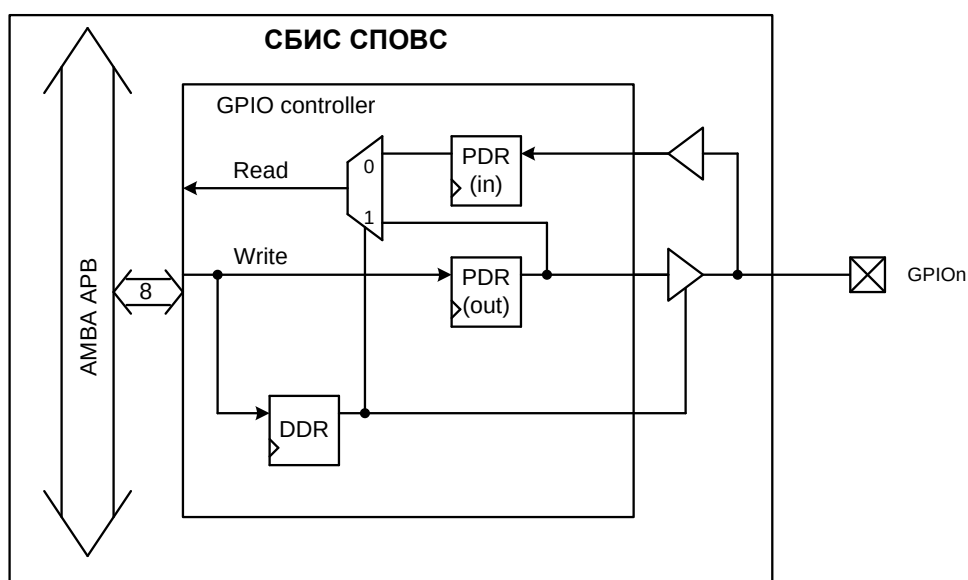


Рисунок 5.50 – Внутренняя структура бита блока портов GPIO

СБИС СПОВС имеет 16 портов GPIO, но реально биты порта разбиты на группы по 8 разрядов, т.е. за одно обращение по шине AMBA APB возможна запись или чтение значений только 8 старших (PDR1) или 8 младших портов (PDR0).

Кроме управления 16 портами общего назначения, контроллер GPIO используется для выбора активного SPI slave устройства (биты PDR3[6:4]). Подробную информацию см. в п. 5.14.1.

					ЮФКВ.431268.006РЭ			Лист
								188
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

5.16.2 Регистровая модель контроллера портов GPIO

Программно доступные регистры контроллера портов общего назначения GPIO расположены в области памяти периферийных устройств ARMU (ARM Peripheral Area), имеют базовое смещение GPIO Base = 0xFFF8A000 и общий размер 4 Кб. Спецификация регистров представлена в таблице ниже:

Таблица 5.149. Спецификация регистров контроллера GPIO

Адрес	Тип	Разрядность	Начальное значение	Имя	Описание
GPIO Base + 0x00	ЧТ/ЗП	8	-	PDR0	Регистр порта данных 0
GPIO Base + 0x04	ЧТ/ЗП	8	-	PDR1	Регистр порта данных 1
GPIO Base + 0x08	ЧТ/ЗП	8	-	PDR2	Регистр порта данных 2
GPIO Base + 0x0C	ЧТ/ЗП	8	-	PDR3	Регистр порта данных 3
GPIO Base + 0x10	ЧТ/ЗП	8	0x00	DDR0	Регистр направления передачи данных 0
GPIO Base + 0x14	ЧТ/ЗП	8	0x00	DDR1	Регистр направления передачи данных 1
GPIO Base + 0x18	ЧТ/ЗП	8	0x00	DDR2	Регистр направления передачи данных 2
GPIO Base + 0x1C	ЧТ/ЗП	8	0xFF	DDR3	Регистр направления передачи данных 3
GPIO Base + 0x20 to + 0xFFC	-	8	-	-	Зарезервирован (доступ запрещен)

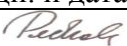
Далее приведено подробное описание каждого регистра.

5.16.2.1 Регистры портов данных. PDRx

PDRx (x=0,1,2 и 3) – четыре регистра, используемые для записи и чтения значений на внешних выводах GPIO. В таблице ниже приведен формат регистров PDRx.

Таблица 5.150. Формат регистров PDRx

Биты	Название	Тип	Выполняемая функция
[31:8]	-	-	Зарезервировано
[7:0]	PDR	ЧТ/ЗП	Входные и выходные данные передаются через данный 8-битный регистр. Биты распределены по регистрам следующим образом: PDR0[7:0]: внешние выводы GPIO[7:0] PDR1[7:0]: внешние выводы GPIO[15:8] PDR2[7:0]: не используются PDR3[7]: не используются PDR3[6:4]: задают активный сигнал SPI_CSx (см. Таблица 5.102) PDR3[3:0]: не используются Направление передачи порта определяется соответствующими битами регистра DDR.

					ЮФКВ.431268.006РЭ			Лист
								189
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.16.2.2 Регистры направления передачи данных. DDRx

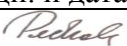
DDR_x (x=0,1,2 и 3) – четыре регистра, используемые для задания направления передачи данных портов GPIO. В таблице ниже приведен формат регистров DDR_x.

Таблица 5.151 – Формат регистров DDR_x

Биты	Название	Тип	Выполняемые функции
[31:8]	-	-	Зарезервировано
[7:0]	DDR	ЧТ/ЗП	DDR регистры – регистры, контролирующие направление передачи данных портов GPIO. Каждый бит DDR _x регистра управляет направлением, соответствующего ему GPIO порта. DDR _x = 0 : порт входных данных DDR _x = 1 : порт выходных данных Биты распределены по регистрам следующим образом: DDR0[7:0]: управление направлением портов GPIO[7:0] DDR1[7:0]: управление направлением портов GPIO[15:8] DDR2[7:0]: не используются DDR3[7:0]: не используются, запись в биты [6:4] значения отличного от 111, приведет к невозможности программным образом определить активное SPI slave устройство. Хотя в начальный момент времени порты GPIO[15:0] установлены как входы, необходимо учитывать что программа начального загрузчика, расположенного в ПЗУ имеет настройки, позволяющие ей использовать эти порты (см. п. 5.17.2).

5.16.3 Прерывания

Контроллер GPIO не имеет выводов прерываний, подключенных к системному контроллеру прерываний VIC.

					ЮФКВ.431268.006РЭ			Лист
								190
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.17 Загрузочное ПЗУ BROM

СБИС СПОВС имеет банк памяти ПЗУ объемом 4 Кб, который содержит программу начального загрузчика. Начальная загрузка микросхемы возможна через интерфейсы SPI, UART0, UART1 и USB. Выбор приоритетного режима загрузки осуществляется внешними входами BOOTM2 и BOOTM1 в соответствии с таблицей:

Таблица 5.152 – Варианты начальной загрузки СБИС СПОВС

BOOTM2	BOOTM1	Интерфейс для загрузки
0	0	SPI
0	1	UART0
1	0	UART1
1	1	USB

Начальный загрузчик – это программа, которая начинает выполняться после снятия внешнего сигнала сброса и инициализации схемы ФАПЧХ внутри СБИС СПОВС. Задача этой программы загрузить во внутреннюю память процессорного ядра ARM следующую исполняемую программу и передать ей управление. Этой следующей программой может быть, например, загрузчик операционной системы.

Внимание! Существует ограничение на размер образа, загружаемой программы – при загрузке через интерфейсы SPI или USB – 255 Кбайт, а при загрузке через интерфейс UART – 32 Кбайт.

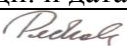
Адрес, по которому будет передано управление, является адресом вектора прерывания “Reset” и зависит от значения на входе VINITH. Этим сигналом включается т.н. “High Vectors Mode” – режим, который переносит таблицу векторов из нулевого адреса, по адресу 0xFFFF0000. Следовательно, если VINITH=0, то после выполнения программы начального загрузчика, будет выполнен переход по адресу 0x00000000, а если VINITH=1, то по адресу 0xFFFF0000.

Двоичный образ начального загрузчика (см. Рисунок 5.52(а)) находится в доступной только для чтения памяти (ROM, адреса 0x00040000-0x00040FFF) начиная с адреса 0x00040000. Первые восемь команд начального загрузчика составляют таблицу векторов прерываний. Каждый вектор таблицы – это одна команда перехода на соответствующую номеру вектора процедуру обработки прерывания.

Двоичный образ загружаемой программы (см. Рисунок 5.52(б)) считывается начальным загрузчиком с одного из четырех внешних устройств – SPI, UART0, UART1, USB. Двоичный образ загружаемой программы отличается от двоичного образа полученного компилятором двумя словами. Шестое слово двоичного образа заменено на служебное слово загрузчика, а в конец образа добавлено дополнительное слово, содержащее CRC32 свертку всех остальных слов. Шестое слово выбрано для хранения служебного слова загрузчика, поскольку предполагается, что первые восемь слов загружаемой программы это таблица векторов прерываний, а шестой вектор в таблице векторов прерываний не используется из-за отсутствия в системе соответствующего прерывания

5.17.1 Служебное слово загрузчика

Формат служебного слова загрузчика приведен на рисунке ниже (см. Рисунок 5.51).

					ЮФКВ.431268.006РЭ				Лист
									191
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

31	24 23	16 15	0
Делитель частоты SPI (четное число от 4 до 254)	Магический код (0xA5)	Размер загружаемой программы (до 0xFFFF 32-х разрядных слов)	

Рисунок 5.51 – Формат служебного слова загрузчика

Начальный загрузчик проверяет, что

- делитель частоты SPI четное число в пределах от 4 до 254;
- значение поля ‘магический код’ – 0xA5;
- размер загружаемой программы больше 8 и меньше или равно 0xFFFF 32-х разрядных слов.

Если хотя бы одно из условий не выполняется, загрузка заканчивается аварийно.

Начальная загрузка происходит следующим образом. После снятия внешнего сигнала сброса и инициализации схемы ФАПЧХ, устанавливается начальное отображение памяти (см. п. “3 Карта памяти СБИС СПОВС”, отображение “Before Remap”), при котором область памяти 0x00040000 – 0x00040FFF отображается на область памяти 0x00000000 – 0x00000FFF, и управление передается команде с адресом 0x00000000 (адресом 0xFFFF0000, если вход VINITH=1). Эта команда выполняет переход на точку входа начального загрузчика.

Начальный загрузчик выбирает устройство для считывания двоичного образа программы, считывает этот образ в область памяти начиная с адреса 0xC0000000, меняет отображение памяти (см. п. “3 Карта памяти СБИС СПОВС”, отображение “After Remap”), так что область памяти 0xC0000000 – 0xC003FFFF отображается на область памяти 0x00000000 – 0x0003FFFF, и передает управление команде с адресом 0x00000000 (адресом 0xFFFF0000, если вход VINITH=1). Работа начального загрузчика на этом закончена. Схематическое представление двоичных образов начального загрузчика и загружаемой программы приведено на рисунке ниже (см. Рисунок 5.52).

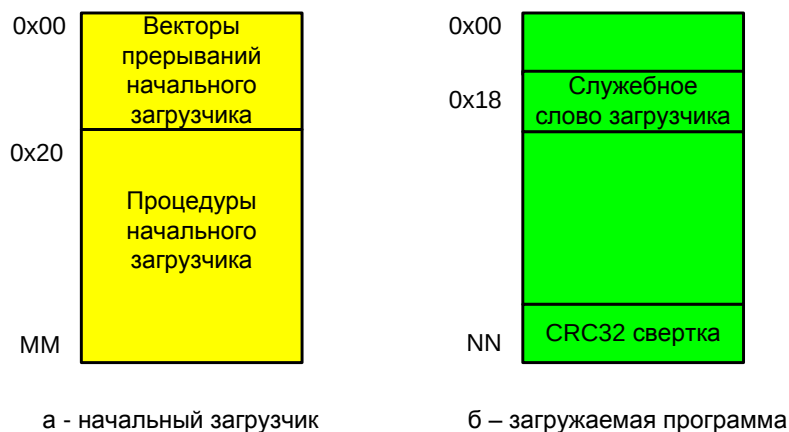
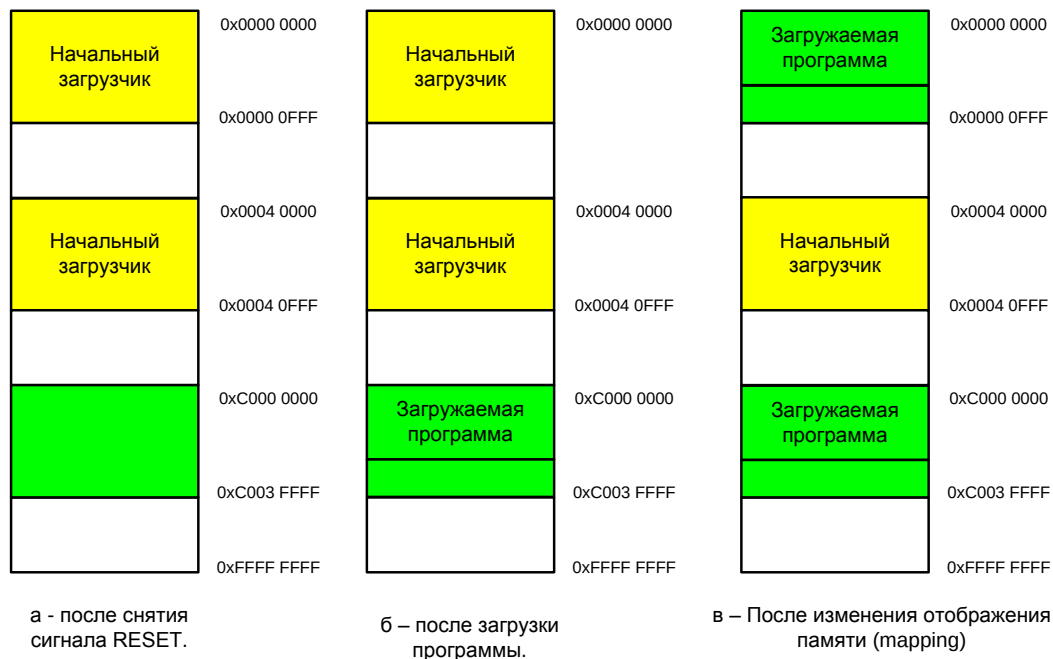


Рисунок 5.52 – Двоичные образы начального загрузчика и загружаемой программы

Изменение состояния памяти в процессе загрузки программы приведено на рисунке ниже (см. Рисунок 5.53).

					ЮФКВ.431268.006РЭ			Лист
								192
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					



Условные обозначения: - только чтение - чтение и запись - не интересует

Рисунок 5.53 – Последовательность состояний памяти в процессе загрузки

5.17.2 Алгоритм работы начального загрузчика

Работа начального загрузчика зависит от состояния четырех внешних входов микросхемы BOOTM1, BOOTM2, TSTMODE, VINITH. Сигнал TSTMODE определяет режим, в котором загружается СБИС – пользовательский или режим функционального тестирования. Для загрузки в нормальном (пользовательском) режиме, необходимо выставить TSTMODE=0. Режим функционального тестирования (TSTMODE=1) предназначен для производственного контроля качества изготовления микросхемы и не должен использоваться пользователями.

Комбинация этих четырех сигналов определяет режим работы загрузчика и устройство, с которого выполняется загрузка. Алгоритм работы начального загрузчика приведен на рисунке ниже (см. Рисунок 5.54).

					ЮФКВ.431268.006РЭ				Лист
									193
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	<i>Редко</i> 16.01.14		22755-1						

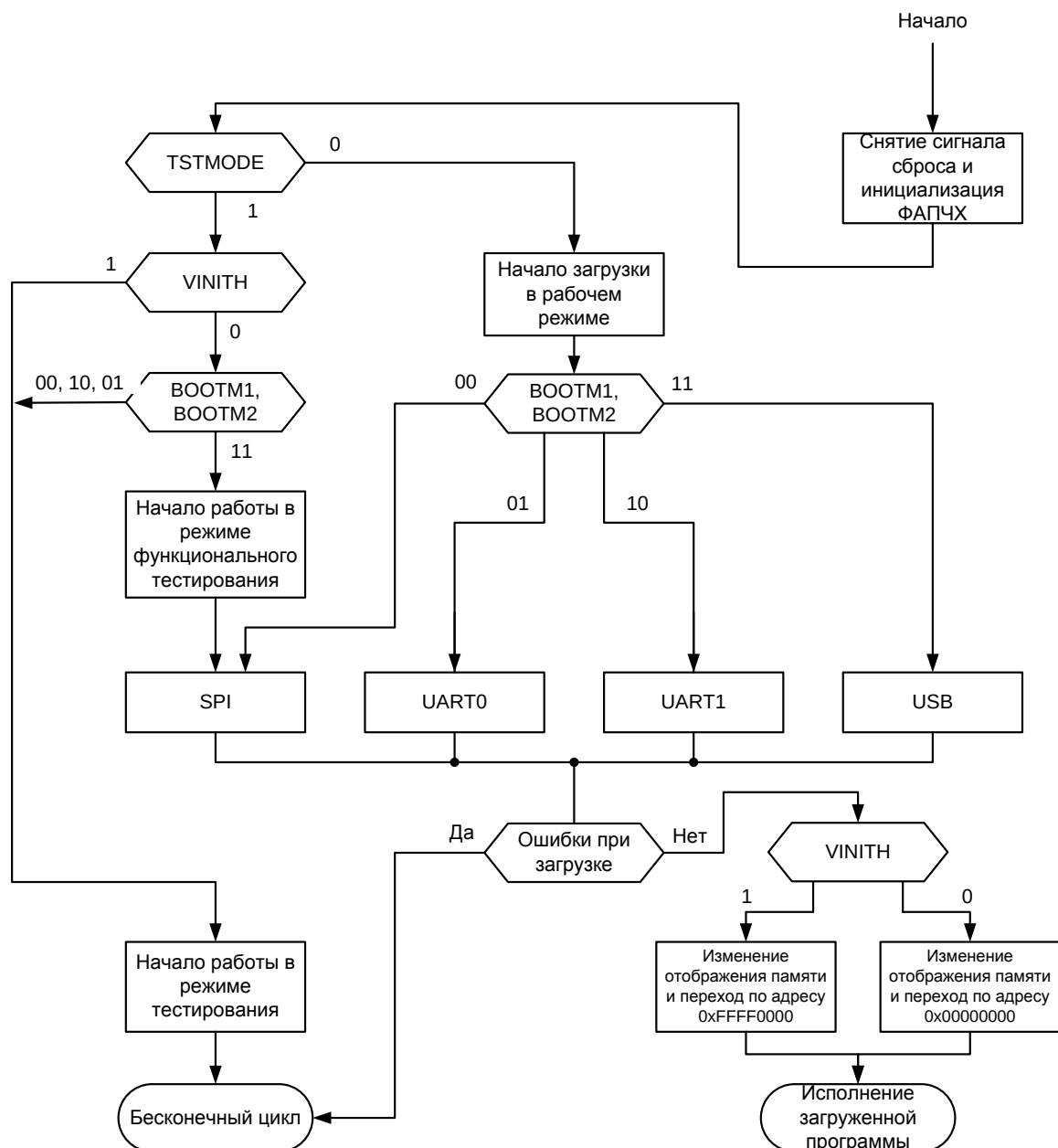


Рисунок 5.54 – Алгоритм работы начального загрузчика

5.17.3 Состояние процессора перед началом исполнения загруженной программы

Непосредственно перед началом исполнения загруженной начальным загрузчиком программы значения всех регистров процессорного ядра ARM, кроме PC и CPSR, считаются неопределенными. Регистры PC и CPSR имеют следующие значения.

Регистр PC.

PC = 0

Регистр CPSR.

Определены значения следующих разрядов регистра CPSR.

[4:0] = b10011 – Supervisor mode

[5] = 0 – Thumb state

[6] = 0 – IRQ disable

					ЮФКВ.431268.006РЭ			Лист
								194
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата			
22755-2	<i>Редко</i> 16.01.14		22755-1					

- [7] = 0 – FIQ disable
- [8] = 0 – Imprecise abort disable
- [9] = 0 – Endianess
- [24] = 0 – Java state

Значения остальных разрядов регистра CPSR не определены.

5.17.4 Пример загружаемой программы

В этом разделе приведена возможная схема программы, которая правильно работает после того, как она загружена и запущена начальным загрузчиком. При описании использован ассемблер ARM.

В результате отображения памяти, используемого при работе программы, загружаемая программа оказывается в двух адресных пространствах одновременно. Первое адресное пространство начинается с адреса 0x00000000, а второе – с адреса 0xc0000000. Программа компилируется в предположении, что она находится во втором адресном пространстве.

Первый фрагмент программы содержит вектора прерываний и располагается в памяти, начиная с адреса 0xc0000000. Этот фрагмент написан так, что он правильно работает в обоих адресных пространствах. Для этого вместо команд перехода использованы команды загрузки в регистр PC значений, адреса которых задаются смещениями относительно адресов соответствующих команд загрузки.

```
AREA Vect, CODE, READONLY
    IMPORT Reset_Handler
    IMPORT Undefined_Handler
    IMPORT SWI_Handler
    IMPORT Prefetch_Handler
    IMPORT Abort_Handler
    IMPORT IRQ_Handler
    IMPORT FIQ_Handler

ENTRY
    LDR PC, Reset_Addr
    LDR PC, Undefined_Addr
    LDR PC, SWI_Addr
    LDR PC, Prefetch_Addr
    LDR PC, Abort_Addr
    NOP ; На место этого вектора надо, после компиляции
        ; программы, записать служебное слово загрузчика.
    LDR PC, IRQ_Addr
    LDR PC, FIQ_Addr

Reset_Addr      DCD Reset_Handler
Undefined_Addr   DCD Undefined_Handler
SWI_Addr        DCD SWI_Handler
Prefetch_Addr   DCD Prefetch_Handler
Abort_Addr      DCD Abort_Handler
                DCD 0; Reserved vector
IRQ_Addr        DCD IRQ_Handler
FIQ_Addr        DCD FIQ_Handler
END
```

На написание второго фрагмента программы ограничений не накладывается и, в общем случае, он может работать только во втором адресном пространстве.

					ЮФКВ.431268.006РЭ			Лист
								195
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

```

AREA MainCode, CODE, READONLY
;
;   Далее следует остальной код программы
;
....
....
;   Сюда добавляется свертка crc32
END

```

5.17.5 Начальная загрузка через SPI.

Загрузка через SPI предполагает загрузку программы из SPI флэш-памяти. Если предполагается этот способ начальной загрузки, то подключаемая микросхема памяти должна удовлетворять следующим условиям:

- поддержка SPI mode 3
- частота работы не менее $F_{apb}/8$, т.к. начальный делитель опорного синхросигнала, выставяемый загрузчиком – 8
- поддержка команды FAST_READ (код – 0x0B)
- загрузочная flash-память должна быть подключена к SPI_CS0 выводу СБИС
- максимальный объем flash-памяти 16Мбайт.

5.17.5.1 Считывание данных из флэш-памяти.

При считывании данных из флэш-памяти загрузчик передает и принимает данные через SPI 16-ти разрядными словами. Ниже каждое такое слово представлено как пара <старший байт> <младший байт>.

Для считывания фрагмента данных из флэш-памяти загрузчик через SPI передает следующую последовательность слов.

```

Cm a2
a1 a0
xx xx
xx xx
...

```

Значения байтов передаваемой последовательности слов таковы.

- cm – команда быстрого чтения из флэш-памяти (FAST_READ = 0x0B);
- a2, a1, a0 – 24-х разрядный адрес первого считываемого из флэш-памяти байта;
- xx – байты с любым значением.

В ответ загрузчик получает через SPI следующую последовательность слов.

```

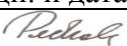
Xx xx
xx xx
xx b0
b1 b2
b3 b4
...

```

Значения байтов принимаемой последовательности слов таковы.

- b0, b1, b2... – следующие друг за другом байты из флэш-памяти;
- xx – байты с любым значением.

Число принятых слов совпадает с числом переданных.

					ЮФКВ.431268.006РЭ			Лист
								196
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

5.17.5.2 Последовательность загрузки через SPI.

Загрузка через SPI выполняется в два этапа. На первом этапе SPI инициализируется с делителем частоты равным 8, и из флэш-памяти читаются 24 байта (шесть 32-х разрядных слов), начиная с адреса 0x000000. Последнее из этих шести слов трактуется как служебное слово загрузчика. Проверяется содержимое служебного слова и если оно правильное, выполняется второе чтение. В противном случае происходит аварийное прекращение загрузки.

Перед вторым чтением SPI инициализируется с делителем частоты равным значению соответствующего поля служебного слова. После этого из флэш-памяти читаются $(N - 6)$ 32-х разрядных слов начиная с адреса 0x000018, где N – размер образа загружаемой программы, взятый из служебного слова.

5.17.6 Начальная загрузка через UART.

Загрузка через UART выполняется в два этапа. На первом этапе выбирается скорость обмена, а на втором происходит чтение данных.

5.17.6.1 Выбор скорости обмена.

Загрузчик пытается определить скорость интерфейса хост-компьютера путем перебора следующих значений для коэффициентов {DLM,DLL}: 11, 22, 44, 89, 134, 178, 266, 355, 532, 1066. Если определить скорость работы интерфейса хост-компьютера не удастся, загрузчик пытается работать со значением {DLM,DLL} 532.

Загрузчик определяет скорость передачи данных, читая пробные символы 0x0D. Выбрав скорость обмена, загрузчик инициализирует UART для работы на этой скорости и посылает символ 0x0D.

Хост-компьютер должен посылать пробные символы 0x0D с интервалом между началом передачи последовательных символов не менее 4 мс. Начать передачу пробных символов можно в любое время, как до, так и после снятия сигнала RESET. Перед передачей каждого пробного символа хост-компьютер должен проверить, не пришел ли символ 0x0D от микросхемы. Если символ 0x0D от микросхемы пришел, хост-компьютер начинает передачу данных.

5.17.6.2 Передача данных.

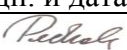
Начальный загрузчик принимает и передает данные через UART пакетами. Каждый принятый пакет должен содержать одну команду, которую начальный загрузчик выполняет, если пакет принят без ошибок. В ответ на каждый принятый пакет начальный загрузчик посылает сообщение, содержащее либо подтверждение успешного выполнения команды, либо, в случае ошибки при обработке пакета, информацию об ошибке.

5.17.6.3 Структура пакета.

Пакет – это последовательность байтов следующего вида.

<BGNPKG><BGNPKG><DATA>...<CHECK_SUM><ENDPKG>

Два первых байта это маркер начала пакета. Последний байт — маркер конца пакета. Предпоследний байт содержит контрольную сумму байтов данных пакета. Остальные байты это байты данных. Если значение байта данных или контрольной суммы совпадает со значением одного из трех служебных байтов <BGNPKG>,

					ЮФКВ.431268.006РЭ			Лист
								197
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

<ENDPKG> или <ESCAPE>, то перед ним вставляется байт <ESCAPE>. При вычислении контрольной суммы служебные байты не учитываются. Длина пакета, включая служебные байты, не должна превышать значения переменной **MaxPackageLength** (см. ниже).

Значения служебных байтов таковы.

- <BGNPKG> - 0x0f
- <ENDPKG> - 0x04
- <ESCAPE> - 0x05

Контрольная сумма <CHECK_SUM> это поразрядная сумма по модулю два числа 0xA5 и всех байтов данных.

5.17.6.4 Переменные.

Работа загрузчика зависит от значений трех переменных. Первые две переменные используются для определения ошибки при приеме пакета, а третья для определения условия аварийного прекращения загрузки.

- **MaxPackageLength** – максимально допустимая длина пакета;
- **MaxSkipLength** – максимально допустимое число символов, пропускаемых при ожидании начала пакета;
- **MaxErrorNumber** – максимальное число следующих подряд ошибок при получении пакета.
- Значения этих переменных можно изменить командой **cmdSET_OPTIONS** (см. ниже). Начальные значения переменных таковы.
- **MaxPackageLength** = 1000
- **MaxSkipLength** = 100
- **MaxErrorNumber** = 5

5.17.6.5 Команды.

Байты данных каждого принимаемого пакета рассматриваются как команда. Первый байт это код команды. Этот байт должен присутствовать обязательно. Остальные байты данных, если они есть, это параметры команды, число и смысл которых определяется кодом команды. Возможны такие команды.

- cmdBREAK
- cmdDATA
- cmdEND_OF_DATA
- cmdREPEAT_MESSAGE
- cmdRESTART
- cmdSET_OPTIONS

Ниже приводится описание каждой команды и действий загрузчика при ее получении.

cmdBREAK

Код – 0x33.

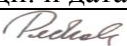
Параметров нет.

Действие загрузчика — аварийное завершение загрузки.

cmdDATA

Код - 0x11.

Параметры — от 1 до **MaxDataLength** загружаемых байтов. Значение **MaxDataLength** определяется исходя из условия, что длина пакета с учетом служебных символов не должна превышать **MaxPackageLength**.

					ЮФКВ.431268.006РЭ			Лист
								198
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Действие загрузчика — запись байтов данных в область загрузки по адресу, заданному текущим указателем и увеличение этого указателя на число записанных байтов.

cmdEND OF DATA

Код – 0x22.

Параметров нет.

Действие загрузчика — проверка загруженных данных на согласованность. Вычисление контрольной суммы и проверка ее правильности. Если проверки прошли успешно, меняется отображение памяти (mapping), так что адреса загруженной программы, начиная с адреса 0xC0000000, отображаются на адреса, начиная с адреса 0x00000000. После этого выполняется переход по адресу 0x00000000, где должен находиться нулевой вектор прерывания.

cmdREPEAT MESSAGE

Код – 0x44.

Параметров нет.

Действие загрузчика — повторная посылка последнего сообщения.

cmdRESTART

Код – 0x55.

Параметров нет.

Действие загрузчика — начать загрузку сначала.

cmdSET OPTIONS

Код – 0x66.

Параметры – <MaxPackageLength_high><MaxPackageLength_low>

<MaxSkipLength_high><MaxSkipLength_low>

<MaxErrorNumber><MaxErrorNumber_low>

Действие загрузчика — присваивание значений переменным.

5.17.6 Сообщения.

Каждое из посылаемых в ответ на принятый пакет сообщений состоит из восьми байтов и имеет следующий вид.

<BGNPKG><BGNPKG><command><error_code>
<msg_info1><msg_info2><CHECK_SUM><ENDPKG>

- <command> – код команды; обычно это код команды из последнего полученного пакета
- <error_code> – код ошибки при обработке последнего пакета
- <msg_info1><msg_info2> – старший и младший байты числа байтов, пропущенных до обнаружения маркера начала пакета при получении пакета.

Обработка хост-компьютером посылаемых начальным загрузчиком сообщений необязательна.

5.17.7 Начальная загрузка через USB.

После перехода СБИС в режим загрузки по USB программа-загрузчик проверяет наличие подключения устройства к шине USB – программно анализируется логическое значение на внешнем выводе USBCON микросхемы. Высокий уровень (логическая 1) сигнала USBCON сообщает программе о физическом подключении устройства к включенному нисходящему порту хаба USB. Если обнаружено подключение, загруз-

					ЮФКВ.431268.006РЭ			Лист
								199
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

чик инициализирует контроллер USB и сигнализирует хабу USB о готовности к нумерации, включая внешний подтягивающий резистор, управляемый через вывод TD9 микросхемы.

Когда хаб обнаруживает устройство на своём нисходящем выводе, начинается процесс нумерации. Обмен с устройством осуществляется через выводы USBVP и USBVM на скорости 12 Мбит/с.

В ходе нумерации устройством поддерживается обработка следующих стандартных запросов USB:

Таблица 5.153 – Обработываемые запросы USB

Название запроса	Функция запроса при нумерации	Реакция устройства на запрос
CLEAR_FEATURE	с помощью этого запроса хост-контроллер USB может сбрасывать некоторые настройки обмена с устройством	запрошенные настройки сбрасываются, но это никак не влияет на процесс загрузки
GET_CONFIGURATION	с помощью этого запроса хост-контроллер USB может получить номер текущей конфигурации устройства	возвращается номер текущей конфигурации: 0 или 1 в соответствии с той конфигурацией, которая установлена хост-контроллером с помощью запроса SET_CONFIGURATION
GET_INTERFACE	с помощью этого запроса хост-контроллер USB может получить значение кода текущей настройки для указанного интерфейса	загрузчиком используется только одна настройка интерфейса, поэтому всегда возвращается код 0
GET_STATUS	с помощью этого запроса хост-контроллер USB может получить информацию о том, поддерживает ли устройство механизм Remote Wakeup (инициирование выхода из режима сниженного энергопотребления), а также получает ли устройство питание от шины USB	устройство поддерживает Remote Wakeup и не получает питание от USB, в ответ на запрос выдаётся соответствующий код
SET_ADDRESS	с помощью этого запроса хост-контроллер USB устанавливает адрес устройства на USB	передаваемый в этом запросе адрес устанавливается
SET_CONFIGURATION	с помощью этого запроса хост-контроллер USB устанавливает номер конфигурации устройства	поддерживаются только конфигурации 0 (базовая, после сброса) и 1 (основная рабочая конфигурация)
SET_FEATURE	с помощью этого запроса хост-контроллер USB может устанавливать некоторые настройки обмена с устройством	запрошенные настройки устанавливаются, но это никак не влияет на процесс загрузки
SET_INTERFACE	с помощью этого запроса хост-контроллер USB может установить значение кода текущей настройки для указанного интерфейса	загрузчик поддерживает только одну базовую настройку интерфейсов, поэтому установка любого другого значения, кроме 0 является ошибочным
GET_DESCRIPTOR	с помощью этого запроса хост-контроллер USB запрашивает дескрипторы устройства	в ответ на запрос возвращаются дескрипторы устройства, значения дескрипторов соответствуют приведённому ниже описанию конфигурации устройства

Также поддерживаются запросы класса CDC:

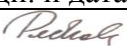
					ЮФКВ.431268.006РЭ			Лист
								200
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 5.154 – Поддерживаемые запросы класса CDC

Название запроса	Функция запроса при нумерации	Реакция устройства на запрос
SET_LINE_CODING	с помощью этого запроса хост-контроллер USB может установить настройки, специфичные для класса CDC	устройство сохраняет полученные настройки, но это никак не влияет на процесс загрузки
GET_LINE_CODING	с помощью этого запроса хост-контроллер USB может запросить настройки, специфичные для класса CDC	устройство выдаёт запрошенные настройки, но это никак не влияет на процесс загрузки
SET_CONTROL_LINE_STATE	с помощью этого запроса хост-контроллер USB может установить настройки, специфичные для класса CDC	устройство сохраняет полученные настройки, но это никак не влияет на процесс загрузки

С помощью стандартных запросов ПО хост-контроллера USB получает информацию о конфигурации устройства. Выбранная для обеспечения загрузки конфигурация не является жёстко заданной для данного устройства и после загрузки может быть частично изменена пользовательским ПО.

5.17.7.1 Тип устройства и версия протокола.

Устройство принадлежит типу USB 2.0 Full-Speed Function и может обмениваться данными с хост-контроллером USB на скорости 12 Мбит/с.

5.17.8 Использование GPIO для индикации состояния начального загрузчика.

Использование GPIO для индикации состояния начального загрузчика зависит от состояния внешнего входа микросхемы TST_I2. Если TST_I2 = 0, то загрузчик конфигурирует все разряды GPIO как входы и индикации состояния начального загрузчика не происходит.

Если TST_I2 = 1, то загрузчик конфигурирует разряды GPIO [15:8] как выходы и отображает в этих разрядах состояние загрузки. Остальные разряды GPIO конфигурируются как входы. Эта возможность может быть использована для диагностики проблем с начальной загрузкой СБИС СПОВС.

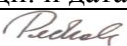
5.17.8.1 Индикация состояния начального загрузчика.

Индикация при начале работы загрузчика – **0xC0**.

Индикация при окончании работы загрузчика – **0xC0 + <код ошибки>**. Если <код ошибки> равен нулю – управление передано загруженной программе. Если <код ошибки> отличен от нуля – управление передано на команду перехода на саму себя (бесконечный цикл).

5.17.8.2 Коды ошибок начального загрузчика.

- 0x00 – загрузка прошла успешно;
- 0x01 – CRC32 свертка загруженного образа не равна ожидаемой;
- 0x02 – ошибка в служебном слове загрузчика;
- 0x03 – в процессе загрузки возникла исключительная ситуация (EXCEPTION);
- 0x04 – UART: неожиданное значение при чтении пробного символа;
- 0x05 – UART: не найден маркер начала пакета;
- 0x06 – UART: не удастся передать пакет;

					ЮФКВ.431268.006РЭ			Лист
								201
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

0x07 – UART: получен символ <ESCAPE>, за которым не следует служебный символ;

0x08 – UART: внутри пакета получен символ <BGNPKG>, перед которым нет символа <ESCAPE>;

0x09 – UART: подсчитанная контрольная сумма пакета не совпала с переданной в пакете;

0x0A – UART: неизвестная команда;

0x0B – UART: слишком длинный пакет;

0x0C – UART: слишком короткий пакет;

0x0D – USB: устройство отсутствует.

					ЮФКВ.431268.006РЭ			Лист
								202
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

5.18 Контроллер динамической памяти DMC

Контроллер динамической памяти DMC имеет следующие характеристики:

- Тип поддерживаемой памяти DDR1 SDRAM;
- Частота шины до 164 МГц;
- Разрядность внешней шины данных, бит – 32;
- Количество сигналов выбора микросхемы (chip select) – 1;
- Количество банков в микросхеме памяти – 4;
- Параметр CAS Latency – 2,5;

Программирование контроллера DMC осуществляется процессорным ядром ARM по шине APB[31:0] (см. Рисунок 5.1). Конфигурационные регистры располагаются в адресном пространстве процессорного ядра ARM по адресам FFF8_5000h – FFF8_5FFFh.

Обращения к внешней микросхеме памяти SDRAM осуществляется по шине AXI_DMC[63:0]. Адресное пространство в адресном пространстве процессорного ядра ARM для обращения к микросхеме внешней памяти 4000_0000h – 7FFF_FFFFh. Декодирование адреса AXI_DMC[63:0] контроллером DMC в сигнал выбора микросхемы (XCS), адрес столбца, адрес строки и номер банка внешней памяти происходит автоматически. Параметры декодирования настраиваются в конфигурационных регистрах chip_0_cfg, memory_cfg.

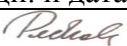
Перед тем как производить обращения к внешней памяти необходимо инициализировать контроллер DMC. Процесс инициализации состоит из двух частей: инициализация PLL, программная настройка контроллера. Инициализация PLL осуществляется при помощи системного контроллера ARMSC (см. главу 5.11 Системный контроллер ARMSC). Программная настройка контроллера включает следующие действия:

- настройку конфигурационных регистров контроллера DMC;
- запись управляющего регистра микросхемы внешней памяти (Mode Register) при помощи регистра mem_cmd;
- перевод контроллера DMC в состояние Ready.

Более подробно состояния контроллера и способы управления описаны ниже.

5.18.1 Описание функционирования контроллера динамической памяти DMC

Контроллер DMC представляет собой автомат с конечными состояниями (Finite State Machine). Его граф переходов представлен ниже на рисунке (см. Рисунок 5.55). В овалах находятся названия состояний. Информацию о текущем состоянии можно получить, прочитав memc_status регистр.

					ЮФКВ.431268.006РЭ			Лист
								203
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

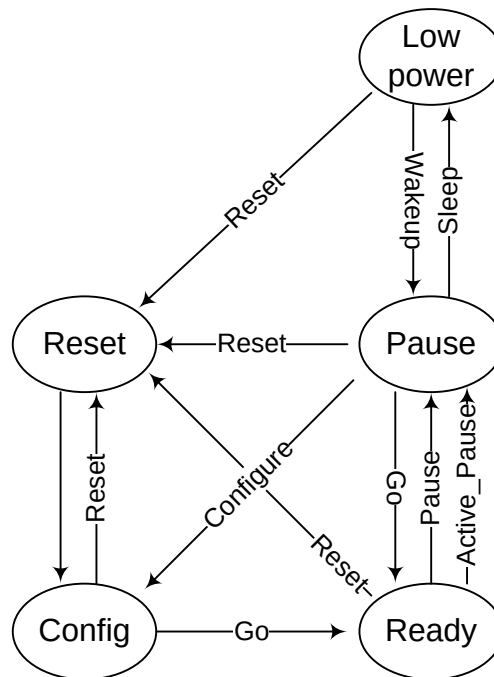


Рисунок 5.55- Состояния контроллера DMC

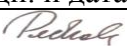
Для перехода между состояниями используются команды (на рисунке название команды расположено на стрелках), которые записываются в **memc_cmd** регистр. Переход из одного состояния в другое длится несколько тактов. В это время выдавать команды перехода в новые состояния запрещается, т.е. надо дождаться выполнения перехода по уже выданной команде. Поэтому после выдачи команды надо производить чтение **memc_status** регистра и смотреть, в каком состоянии находится контроллер DMC. Только после того, как переход выполнен, можно выдавать следующую команду перехода. Ниже приведено описание состояний контроллера DMC:

Reset – в данное состояние контроллер DMC попадает при подаче сигнала сброса.

Config – в данное состояние можно попасть автоматически после снятия сигнала сброса из состояния **RESET** или при помощи команды **Configure**. В состоянии **Config** возможна запись/чтение конфигурационных регистров, задающих режим работы контроллера DMC и параметры микросхемы внешней памяти **SDRAM**. Также в данном состоянии возможна выдача прямых команд внешней микросхеме памяти. В состоянии **Config** не формируется команд регенерации (**Auto Refresh**).

Ready – в данное состояние можно попасть из состояния **Config** или **Pause** при помощи команды **Go**. В состоянии **Ready** осуществляется запись/чтение данных во внешнюю память **SDRAM**. В данном режиме автоматически формируются команды принудительной регенерации (**Auto Refresh**) микросхем внешней памяти через время, определяемое в **refresh_prd** регистре.

Pause – в данное состояние можно попасть из состояния **Ready** или **Low power**. В состоянии **Pause** не осуществляется никаких транзакций с внешней памятью, данное состояние является промежуточным между рабочим (**Ready**) и состояниями **Config** или **Low power**. Из состояния **Ready** переход в состояние **Pause** возможен двумя способами: при помощи команды **Pause** или команды **Active Pause**. При подаче команды **Pause** переход осуществляется, когда контроллер DMC не совершает никаких транзакций с внешней памятью и не поступает никаких команд чтения или записи на вход контроллера.

					ЮФКВ.431268.006РЭ			Лист
								204
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

лера. При подаче команды Active Pause переход осуществляется после завершения текущего обмена по внешней шине с внешней памятью, при этом на вход могут поступать команды чтения и записи, которые на внешнюю шину уже выдаваться не будут, пока не придет команда Go. Если переход осуществляется при помощи команды Active Pause, то переход из состояния Pause в состояние Config запрещен.

Low Power – в данное состояние можно попасть из состояния Pause при помощи команды Sleep. При переходе в состояние Low Power микросхемы внешней памяти переводятся в режим авто-регенерации (Self-Refresh mode). В данном состоянии возможна запись/чтение конфигурационных регистров, задающих режим работы контроллера DMC и параметры микросхемы внешней памяти SDRAM. Так же в данном состоянии возможна выдача прямых команд внешней микросхеме памяти. Если в рабочем режиме появилась необходимость изменить регистр конфигурации, то состояние Low Power предпочтительнее состояния Configure. Это обусловлено тем, что в состоянии Low Power микросхемы внешней памяти SDRAM находится в режиме авто-регенерации (Self-Refresh mode) и это гарантирует сохранность данных, которые находятся в микросхеме внешней памяти.

5.18.2 Динамическое управление энергопотреблением.

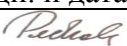
Динамическое управление энергопотреблением заключается в отключении тактового сигнала, снятии сигнала SCLKE и переводе микросхем внешней памяти SDRAM в режим авто-регенерации (self-refresh) при определенном времени бездействия. Возврат в рабочий режим производится автоматически при поступлении команд записи/чтения во внешнюю память SDRAM. Этим динамическое управление отличается от состояния Low Power, из которого переход в состояние Ready происходит только по командам перехода.

В контроллере DMC реализовано несколько функций, которые могут динамически снижать потребление, если контроллер не используется:

1) Если не производится транзакций по внешней шине, то тактовый сигнал SCLK останавливается. Данная функция управляется разрядом stop_mem_clock в mem_cfg регистре. Как только поступает новая команда чтения/записи или контроллеру нужно сделать принудительную регенерацию (Auto Refresh) тактовый сигнал SCLK начинает опять вырабатываться.

2) Перевод сигнала SCKE в неактивное состояние. Данная функция управляется разрядом auto_power_down в mem_cfg регистре. Если данная функция включена, и контроллер DMC не используется в течение времени, определенного полем power_down_prd в mem_cfg регистре, то сигнал SCKE переходит в не активное состояние. Как только поступает новая команда чтения/записи или контроллеру нужно сделать принудительную регенерацию (Auto Refresh), сигнал SCKE переводится в активное состояние. Данная функция не может работать одновременно с функцией остановки тактового сигнала.

3) Перевод внешних микросхем памяти в режим авто-регенерации (self-refresh) и перевод сигнала SCKE в неактивное состояние. Данная функция управляется разрядами auto_power_down, sr_enable и fr_enable (все разряды должны быть в установлении в состояние высокого логического уровня) в mem_cfg регистре. Если данная функция включена, и контроллер DMC не используется в течение времени, определяемого полем fr_time в mem_cfg регистре, то контроллер выдает команду закрытия всех активных строк (precharge). Затем по прошествии времени, определяемого полем power_down_prd в регистре mem_cfg и полем prescale в memory_cfg3 регистре (вычисляется как power_down_prd * prescale), микросхема переводится в режим авто-регенерации

					ЮФКВ.431268.006РЭ			Лист
								205
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

(self-refresh) и сигнал SCKE переводится в неактивное состояние. Как только поступает новая команда чтения/записи, работа контроллера возобновляется. Данная функция не может работать одновременно с функцией остановки тактового сигнала.

5.18.3 Настраиваемые временные параметры

Для конкретной микросхемы памяти SDRAM контроллер динамической памяти DMC позволяет настраивать следующие временные параметры: tMRD, tRCD, tRRD, tWR, tDQSS, tWTR, tRC, tRAS, tRP, tRFC, tESR, tXSR.

tMRD – время выполнения команды загрузки в регистр конфигурации Load Mode Register (MODREG). После команды MODREG в течении времени tMRD контроллеру DMC запрещено выдавать последующие команды (см. Рисунок 5.56).

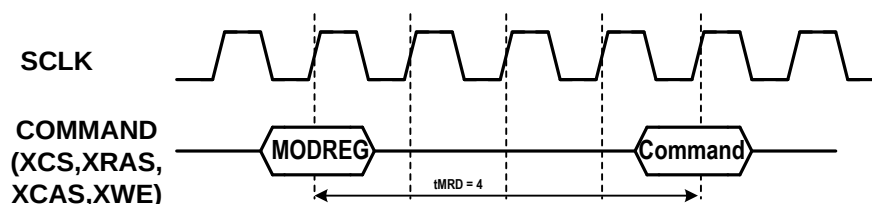


Рисунок 5.56 – Временная диаграмма, объясняющая параметр tMRD

tRCD – время открытия строки командой ACTIVE (ACT). В течение этого времени запрещено выдавать команды чтения или записи в строку, для которой была выдана команда ACT (см. Рисунок 5.57).

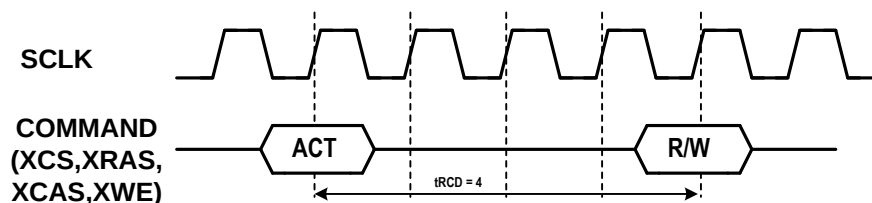


Рисунок 5.57 – Временная диаграмма, объясняющая параметр tRCD

tRRD – минимальный интервал, который должен быть выдержан между последовательно выданными командами ACTIVE (ACT) с различными банкам (см. Рисунок 5.58).

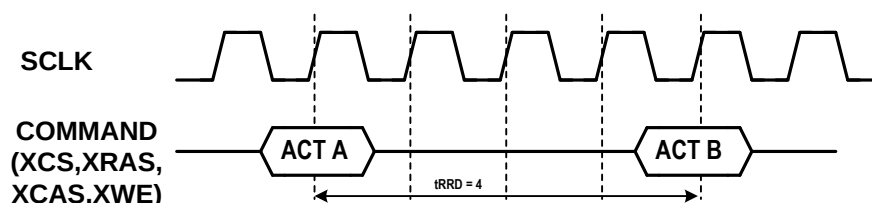


Рисунок 5.58 – Временная диаграмма, объясняющая параметр tRRD

tWR – минимальный интервал, который должен быть выдержан между последними данными для записи и командой Precharge. Интервал отсчитывается от первого положительного фронта сигнала SCLK после последних записанных данных (см. Рисунок 5.59).

					ЮФКВ.431268.006РЭ			Лист
								206
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

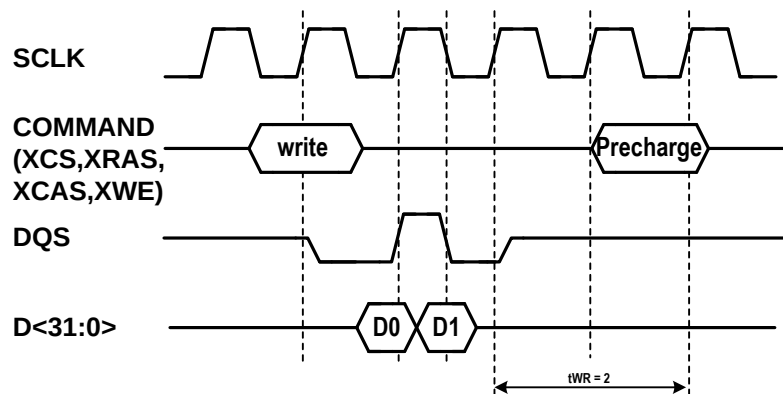


Рисунок 5.59 – Временная диаграмма, объясняющая параметр t_{WR}

t_{DQSS} – интервал между командой записи и первыми данными для этой команды записи. t_{WTR} – минимальный интервал, который должен быть выдержан между последними данными для записи и командой чтения. Интервал отсчитывается от первого положительного фронта сигнала SCLK после последних записанных данных (см. Рисунок 5.60).

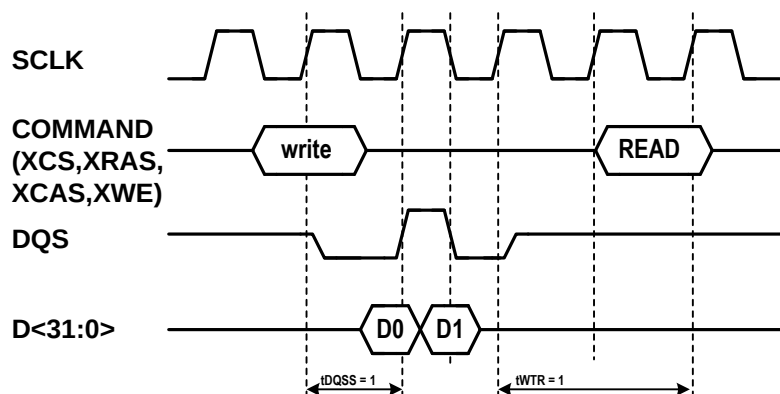


Рисунок 5.60 – Временная диаграмма, объясняющая параметры t_{DQSS} и t_{WTR}

t_{RC} – минимальный интервал между последовательными командами ACTIVE (ACT) для одного и того же банка или между ACTIVE (ACT) и AutoRefresh (см. Рисунок 5.61).

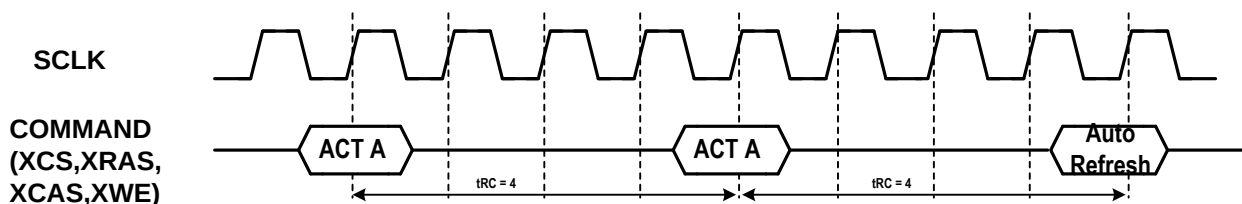


Рисунок 5.61 – Временная диаграмма, объясняющая параметр t_{RC}

t_{RAS} – минимальный интервал между командой ACTIVE (ACT) и командой Precharge. t_{RP} – время выполнения команды Precharge. В течение этого времени не выдается других команд. t_{RFC} – время выполнения команды AutoRefresh. В течение это-

					ЮФКВ.431268.006РЭ			Лист
								207
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

го времени не выдается других команд (см. Рисунок 5.62Error! Reference source not found.).

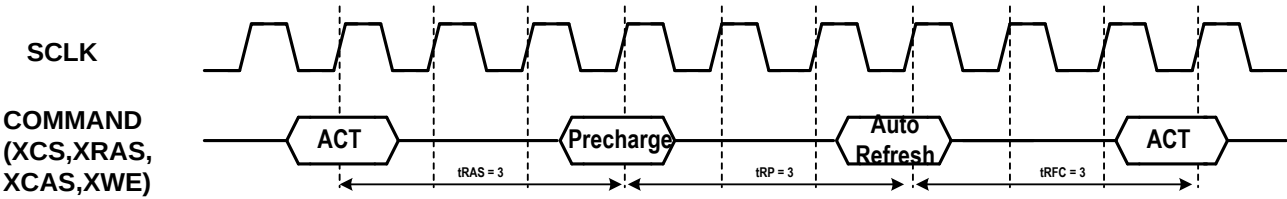


Рисунок 5.62 – Временная диаграмма, объясняющая параметр t_{RAS} , t_{RP} , t_{RFC}

t_{ESR} – минимальный интервал после команды SelfRefresh, после которого можно выдавать команду ExitSelfRefresh. t_{XSR} – минимальный интервал после команды ExitSelfRefresh, после которого можно выдавать какую-либо команду (см. Рисунок 5.63).

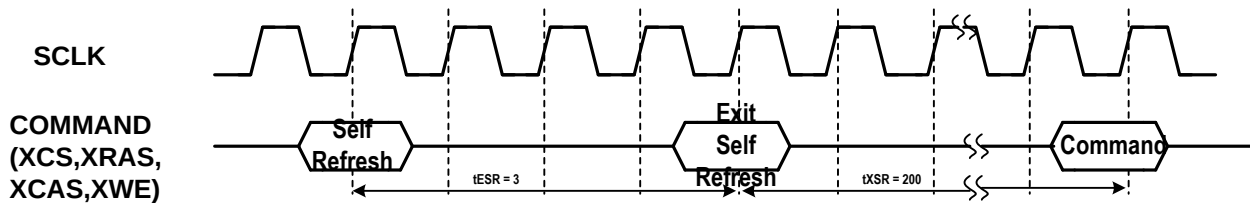


Рисунок 5.63 – Временная диаграмма, объясняющая параметр t_{ESR} , t_{XSR}

5.18.4 Регистровая модель контроллера динамической памяти DMC

В данном разделе описываются программно доступные регистры блока контроллера динамической памяти. Таблица 5.155 содержит список программно доступных регистров контроллера DMC.

					ЮФКВ.431268.006РЭ			Лист
								208
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редкофф</i> 16.01.14		22755-1				

Таблица 5.155 - Список регистров контроллера DMC

Название	Адрес регистра	Тип	Описание
memc_status	FFF8_5000h	ЧТ	Регистр состояния контроллера DMC
memc_cmd	FFF8_5004h	ЗП	Регистр команд контроллера DMC
direct_cmd	FFF8_5008h	ЗП	Регистр прямых команд
memory_cfg	FFF8_500Ch	ЧТ/ЗП	Регистр конфигурации внешней памяти контроллера DMC
refresh_prd	FFF8_5010h	ЧТ/ЗП	Регистр периода регенерации
cas_latency	FFF8_5014h	ЧТ/ЗП	Регистр задержки CAS
t_dqss	FFF8_5018h	ЧТ/ЗП	Задаёт параметр tDQSS
t_mrd	FFF8_501Ch	ЧТ/ЗП	Задаёт параметр tMRD
t_ras	FFF8_5020h	ЧТ/ЗП	Задаёт параметр tRAS
t_rc	FFF8_5024h	ЧТ/ЗП	Задаёт параметр tRC
t_rcd	FFF8_5028h	ЧТ/ЗП	Задаёт параметр tRCD
t_rfc	FFF8_502Ch	ЧТ/ЗП	Задаёт параметр tRFC
t_rp	FFF8_5030h	ЧТ/ЗП	Задаёт параметр tRP
t_rrd	FFF8_5034h	ЧТ/ЗП	Задаёт параметр tRRD
t_wr	FFF8_5038h	ЧТ/ЗП	Задаёт параметр tWR
t_wtr	FFF8_503Ch	ЧТ/ЗП	Задаёт параметр tWTR
t_xp	FFF8_5040h	ЧТ/ЗП	Задаёт параметр tXP
t_xsr	FFF8_5044h	ЧТ/ЗП	Задаёт параметр tXSR
t_esr	FFF8_5048h	ЧТ/ЗП	Задаёт параметр tESR
memory_cfg3	FFF8_5050h	ЧТ/ЗП	Регистр memory_cfg3
chip_0_cfg	FFF8_5200h	ЧТ/ЗП	Задаёт адресное пространство внешней памяти

5.18.4.1 Регистр состояния контроллера памяти memc_status

Регистр **memc_status** доступен только для чтения, содержит информацию о конфигурации контроллера памяти DMC, а так же его текущее состояние. Формат регистра приведен ниже (см. Рисунок 5.64)

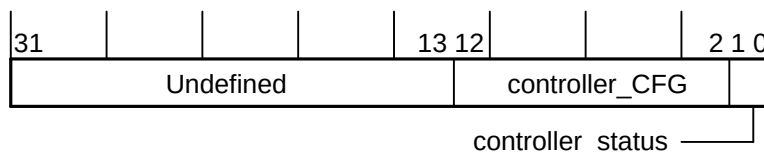


Рисунок 5.64 - Формат регистра memc_status

Поле **controller_CFG** (12-2 разряды) дает информацию о реализованной в СБИС СПОВС конфигурации контроллера. Данное поле принимает значение 205h, что соответствует:

- Тип поддерживаемой памяти DDR SDRAM;
- Разрядность внешней шины данных, бит -32;
- Количество сигналов выбора микросхемы (chip select) – 1;
- Количество банков в микросхеме памяти – 4;

Поле **controller_status** (1-0 разряды) показывает текущее состояние контроллера DMC:

- 0h = Config;
- 1h = Ready;
- 2h = Paused;
- 3h = Low_power.

Разряды 31-13 при чтении могут находиться в любом состоянии.

					ЮФКВ.431268.006РЭ			Лист
								209
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

5.18.4.2 Регистр команд контроллера памяти Memc_cmd

Регистр **memc_cmd** доступен только для записи. Регистр управляет состоянием контроллера. Изменение данного регистра переводит контроллер в соответствующее состояние. Если новая команда поступает в то время как предыдущая команда еще не отработала (контроллер еще не перешел в нужное состояние), новая команда не воспринимается (pready = 0). Формат регистра memc_cmd приведен ниже (Рисунок 5.65).

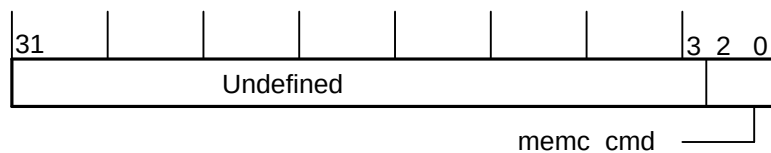


Рисунок 5.65 - Формат регистра Memc_cmd

Поле **memc_cmd** (2-0 разряды) задает команду для перехода контроллера в соответствующее состояние. Возможны следующие значения:

- 0h – Go;
- 1h – Sleep;
- 2h – Wakeup;
- 3h – Pause;
- 4h – Configure;
- 7h – Active_Pause.

Разряды 31-3 при записи должны быть нулями.

5.18.4.3 Регистр прямых команд Direct_cmd

Регистр **direct_cmd** доступен только для записи. Данный регистр позволяет записывать регистры конфигурации микросхем внешней памяти, а так же выдавать команды NOP, Precharge, Auto-refresh, таким образом, осуществляется необходимая инициализация микросхем памяти. Запись в данный регистр возможна только в режиме Config или в режиме Low-power. Формат регистра direct_cmd приведен ниже (см. Рисунок 5.66).

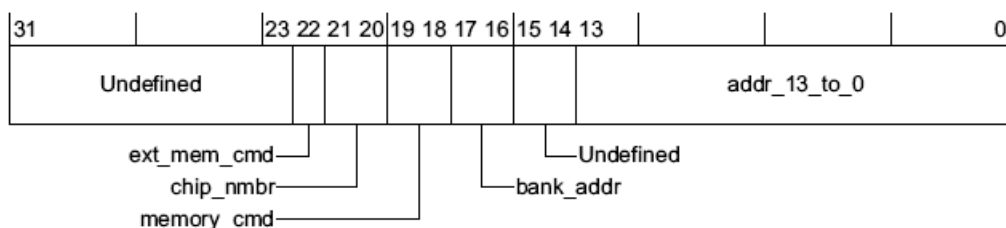


Рисунок 5.66 - Формат регистра Direct_cmd

22-й разряд **ext_mem_cmd** и поле **memory_cmd** (19-18 разряды) определяют тип команды к микросхемам внешней памяти. Допустимые значения

- 000b – Precharge
- 001b – Auto-refresh

					ЮФКВ.431268.006РЭ				Лист
									210
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

010b – Modereg or Extended modereg access

011b – NOP

100b – DPD (Deep power Down)

Разряды поля **addr_13_to_0** (13-0 разряды) выдаются на внешние выводы микросхемы A13 – A0 во время команды Modreg_access.

Поле **chip_nمبر** (21-20 разряды) – данное поле должно содержать значение 0h

Поле **bank_addr** (17-16 разряды) – данные разряды выдаются на внешние выводы микросхемы BA1 – BA0 во время команды Modreg_access.

5.18.4.4 Регистр конфигурации памяти Memory_cfg

Регистр **memory_cfg** доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или в режиме Low-power. Формат регистра Memory_cfg приведен ниже (см. Рисунок 5.67).

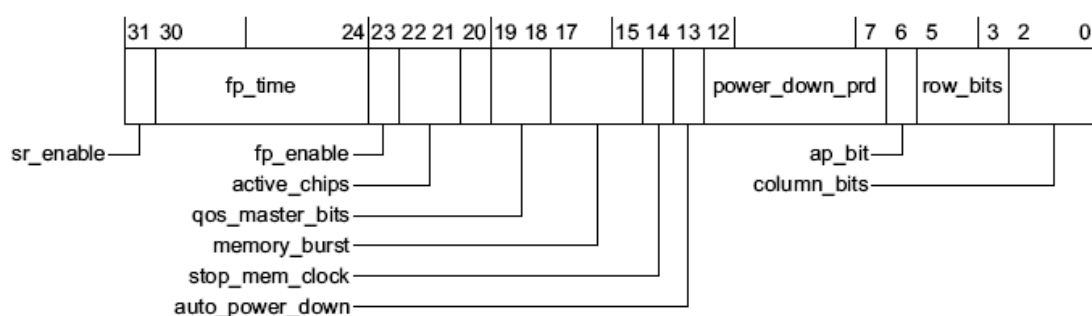


Рисунок 5.67 - Формат регистра Memory_cfg

31-й разряд **sr_enable** – это разрешение перевода микросхемы в режим self_refresh при выключении по таймауту. Используется совместно с полем **fp_enable**:

0 – запрещен автоматический перевод в режим self_refresh;

1 – разрешен автоматический перевод в режим self_refresh;

Поле **fp_time** (30-24 разряды) – время, выраженное в тактах синхросигнала SCLK, после которого будет выполнена команда precharge.

23-й разряд **fp_enable** – разрешение выполнения команды precharge, если контроллер был неактивен в течение времени, заданного полем **fp_time**.

0 – запрещено выполнение команды precharge;

1 – разрешено выполнение команды precharge;

Поле **active_chips** (22-21 разряды) – в данное поле должно быть записано значение 0h

Поле **qos_master_bits** (20-18 разряды) – в данное поле должно быть записано значение 2h

Поле **memory_burst** (17-15 разряды) – задает длину пакета данных при записи/чтении во внешнюю память SDRAM:

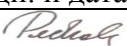
0h = 1 слово;

1h = 2 слова (рекомендуется);

2h = 4 слова;

3h = 8 слов;

4h = 16 слов;

					ЮФКВ.431268.006РЭ				Лист
									211
Изм.	Лист	№ докум.	Подп.	Дата					
Инва.№подл.	Подп. и дата		Взам.инв.№		Инва.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

Также данное значение должно быть запрограммировано в регистр конфигурации микросхемы внешней памяти SDRAM.

14-й разряд **stop_mem_clock** разрешает/запрещает выдачу сигнала SCLK, когда нет обращений к внешней памяти SDRAM:

0 – сигнал SCLK выдается, если нет обращений к внешней памяти SDRAM;

1 – сигнал SCLK не выдается, если нет обращений к внешней памяти SDRAM.

13-й разряд **auto_power_down** разрешает/запрещает перевод внешней памяти SDRAM в режим пониженного энергопотребления, переводя сигнал SCKE в неактивное состояние:

0 – перевод в режим пониженного электропотребления запрещен;

1 – перевод в режим пониженного электропотребления разрешен;

Поле **power_down_prd** (12-7 разряды) – число тактов бездействия сигнала SCLK, после которого контроллер начнет перевод внешней памяти SDRAM в режим пониженного электропотребления (если **auto_power_down**=1). Значение данного поля должно быть больше трех.

6-й разряд **ap_bit** определяет сигнал управления для команды auto-precharge:

0 – для команды auto-precharge используется вывод A10;

1 – для команды auto-precharge используется вывод A8;

Поле **row_bits** (5-3 разряды) – определяет число разрядов для формирования адреса строки:

0h = 11 разрядов;

1h = 12 разрядов;

2h = 13 разрядов;

3h = 14 разрядов;

4h = 15 разрядов;

5h = 16 разрядов;

Поле **column_bits** (2-0 разряды) – определяет число разрядов для формирования адреса столбца:

h0 = 8 разрядов;

1h = 9 разрядов;

2h = 10 разрядов;

3h = 11 разрядов;

4h = 12 разрядов;

Ограничения при программировании:

разряды **sr_enable** и **stop_mem_clock** не должны быть в единице одновременно;

разряды **auto_power_down** и **stop_mem_clock** не должны быть в единице одновременно.

5.18.4.5 Регистр периода регенерации (REFRESH) Refresh_prd

Регистр refresh_prd доступен для записи и чтения, определяет период регенерации памяти SDRAM. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра refresh_prd приведен ниже (см. Рисунок 5.68).

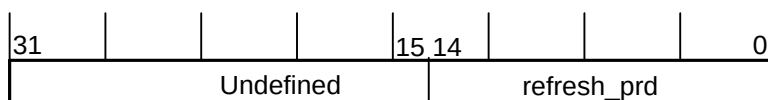


Рисунок 5.68 - Формат регистра Refresh_prd

					ЮФКВ.431268.006РЭ				Лист
									212
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

Поле **refresh_prd** (14-0 разряды) определяет период регенерации памяти SDRAM, выраженный в тактах синхросигнала SCLK.

Разряды 31-15 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.6 Регистр задержки *Cas_latency*

Регистр **Cas_latency** доступен для записи и чтения, содержит значения Cas Latency. Запись в данный регистр или чтение возможны только в режиме Config или Low-power. Формат регистра **Cas_latency** приведен ниже (см. Рисунок 5.69).

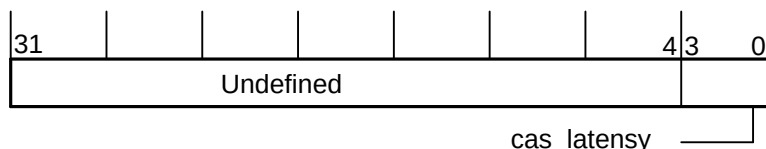


Рисунок 5.69 – Формат регистра *Cas_latency*

Поле **cas_latency** (3-0 разряды) должно содержать значение 5h, что соответствует значению cas_latency = 2,5. Разряды 31-4 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.7 Регистр *t_dqss*

Регистр **t_dqss** доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра **t_dqss** приведен ниже (см. Рисунок 5.70).

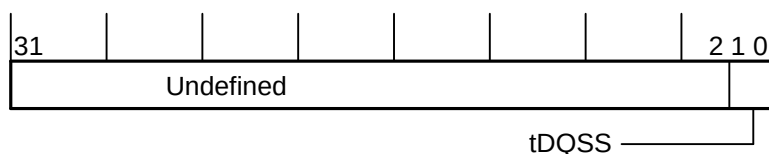


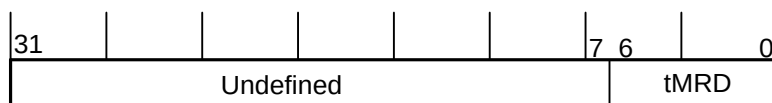
Рисунок 5.70 - Формат регистра *t_dqss*

Поле **tDQSS** (1-0 разряды) определяет параметр tDQSS, выраженный в тактах синхросигнала SCLK.

Разряды 31-2 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.8 Регистр *t_mrd*

Регистр **t_mrd** доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра **t_mrd** приведен ниже (см. Рисунок 5.71).



					ЮФКВ.431268.006РЭ					Лист
										213
Изм.	Лист	№ докум.	Подп.	Дата						
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата				
22755-2	Редько 16.01.14		22755-1							

Рисунок 5.71 - Формат регистра *t mrd*

Поле **tMRD** (6-0 разряды) определяет параметр tMRD, выраженный в тактах синхросигнала SCLK.

Разряды 31-7 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.9 *Pezucmp t ras*

Регистр **t_ras** доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра **t_ras** приведен ниже (см. Рисунок 5.72).

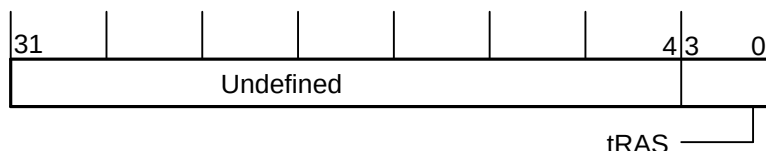


Рисунок 5.72 – Формат регистра t_{ras}

Поле **tRAS** (3-0 разряды) определяет параметр tRAS, выраженный в тактах синхросигнала SCLK.

Разряды 31-4 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.10 *Perucmp t rc*

Регистр t_rc доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра t_rc приведен ниже (см. Рисунок 5.73)

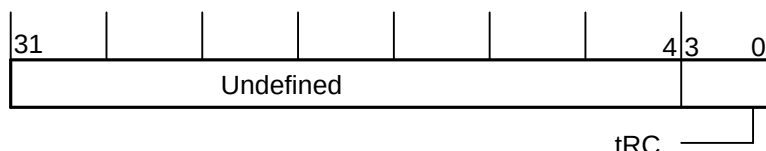


Рисунок 5.73- Формат регистра t_{rc}

Поле **tRC** (3-0 разряды) определяет параметр tRC, выраженный в тактах синхросигнала SCLK.

Разряды 31-4 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.11 *Pezucmp t rcd*

Регистр `t_rcd` доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра `t_rcd` приведен ниже (см. Рисунок 5.74).

					ЮФКВ.431268.006РЭ	Лист	
						214	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв. № подл.		Подп. и дата			Взам. инв. №	Инв. № дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

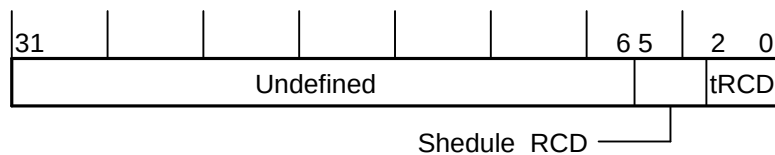


Рисунок 5.74 – Формат регистра t_rcd

Поле **tRCD** (2-0 разряды) определяет параметр tRCD, выраженный в тактах синхросигнала SCLK.

Поле **Shedule_RCD** (5-3 разряды) должно иметь значение tRCD-3.

Разряды 31-6 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.12 Регистр t_rfc

Регистр t_rfc доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра t_rfc приведен ниже (см. Рисунок 5.75).

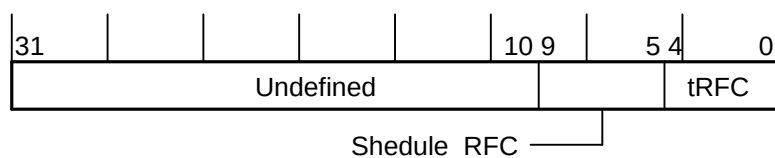


Рисунок 5.75 - Формат регистра t_rfc

Поле **tRFC** (4-0 разряды) определяет параметр tRFC, выраженный в тактах синхросигнала SCLK.

Поле **Shedule_RFC** (9-5 разряды) должно иметь значение tRFC-3.

Разряды 31-10 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.13 Регистр t_rp

Регистр t_rp доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра t_rp приведен ниже (см. Рисунок 5.76).

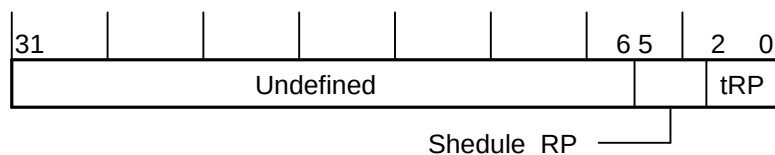


Рисунок 5.76 - Формат регистра t_rp

Поле **tRP** (2-0 разряды) определяет параметр tRP, выраженный в тактах синхросигнала SCLK.

Поле **Shedule_RP** (5-3 разряды) должно иметь значение tRP-3.

					ЮФКВ.431268.006РЭ				Лист
									215
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		Редько 16.01.14		22755-1					

Разряды 31-6 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.14 Регистр *t_rrd*

Регистр *t_rrd* доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра *t_rrd* приведен ниже (см. Рисунок 5.77).



Рисунок 5.77 - Формат регистра *t_rrd*

Поле **tRRD** (3-0 разряды) определяет параметр tRRD, выраженный в тактах синхросигнала SCLK.

Разряды 31-4 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.15 Регистр *t_wr*

Регистр *t_wr* доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра *t_wr* приведен ниже (см. Рисунок 5.78).



Рисунок 5.78 - Формат регистра *t_wr*

Поле **tWR** (2-0 разряды) определяет параметр tWR, выраженный в тактах синхросигнала SCLK.

Разряды 31-3 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.16 Регистр *t_wtr*

Регистр *t_wtr* доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра *t_wtr* приведен ниже (см. Рисунок 5.79).



Рисунок 5.79 - Формат регистра *t_wtr*

Поле **tWTR** (2-0 разряды) определяет параметр tWTR, выраженный в тактах синхросигнала SCLK.

					ЮФКВ.431268.006РЭ				Лист
									216
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

Разряды 31-3 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.17 Регистр *t_xp*

Регистр *t_xp* доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра *t_xp* приведен ниже (см. Рисунок 5.80).



Рисунок 5.80 - Формат регистра *t_xp*

Поле **tXP** (7-0 разряды) определяет параметр tXP, выраженный в тактах синхросигнала SCLK.

Разряды 31-8 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.18 Регистр *t_xsr*

Регистр *t_xsr* доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра *t_xsr* приведен ниже (см. Рисунок 5.81).



Рисунок 5.81 - Формат регистра *t_xsr*

Поле **tXSR** (7-0 разряды) определяет параметр tXSR, выраженный в тактах синхросигнала SCLK.

Разряды 31-8 при записи должны быть нулями, при чтении они могут находиться в любом состоянии

5.18.4.19 Регистр *t_esr*

Регистр *t_esr* доступен для записи и чтения. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра *t_esr* приведен ниже (см. Рисунок 5.82).



Рисунок 5.82 - Формат регистра *t_esr*

Поле **tESR** (7-0 разряды) определяет параметр tESR, выраженный в тактах синхросигнала SCLK.

					ЮФКВ.431268.006РЭ				Лист
									217
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

Разряды 31-8 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

5.18.4.20 Регистр *memory_cfg3*

Регистр **memory_cfg3** доступен для записи и чтения. Формат регистра **memory_cfg3** приведен ниже (см. Рисунок 5.83).

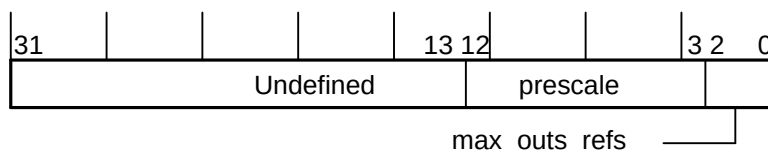


Рисунок 5.83 - Формат регистра *memory_cfg3*

Разряды 31-13 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

Поле **prescale** (12-3 разряды) определяет таймаут при выключении вместе с полем **power_down_prd** регистра **memory_cfg**.

Поле **max_outs_refs** (2-0 разряды) определяет максимальное количество ожидающих выполнения команд refresh.

5.18.4.21 Регистр *chip_0_cfg*

Регистр **chip_0_cfg** доступен для записи и чтения, он задает формат декодирования адреса. Запись в данный регистр или чтение возможно только в режиме Config или Low-power. Формат регистра **chip_0_cfg** приведен ниже (см. Рисунок 5.84).

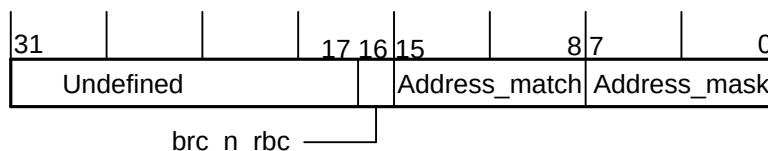


Рисунок 5.84 - Формат регистра *chip_0_cfg*

Поле **Address_match** (15-8 разряды) сравнивается со старшими разрядами адреса AXI[31:24] при обращении к памяти. Если разряды совпадают, то происходит запись/чтение во внешнюю SDRAM память.

Поле **Address_mask** (7-0 разряды) служит маской для старших разрядов адреса AXI[31:24] при сравнении поля **Address_match**:

1 – данный разряд используется при сравнении адреса и поля **Address_match**;

0 – данный разряд не используется при сравнении адреса и поля **Address_match**;

16-й разряд **brc_n_rbc** определяет правило декодирования (формирования) адреса строки, столбца и номера банка внешней памяти из адреса AXI[31:0]:

0 – адрес AXI[31:0] декодируется как строка, банк, столбец;

1 – адрес AXI[31:0] декодируется как банк, строка, столбец;

					ЮФКВ.431268.006РЭ			Лист
								218
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

Разряды 31-17 при записи должны быть нулями, при чтении они могут находиться в любом состоянии.

					ЮФКВ.431268.006РЭ			Лист
								219
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

6 Система общей памяти (SMU).

Структурная схема системы общей памяти приведена ниже (см. Рисунок 6.1).

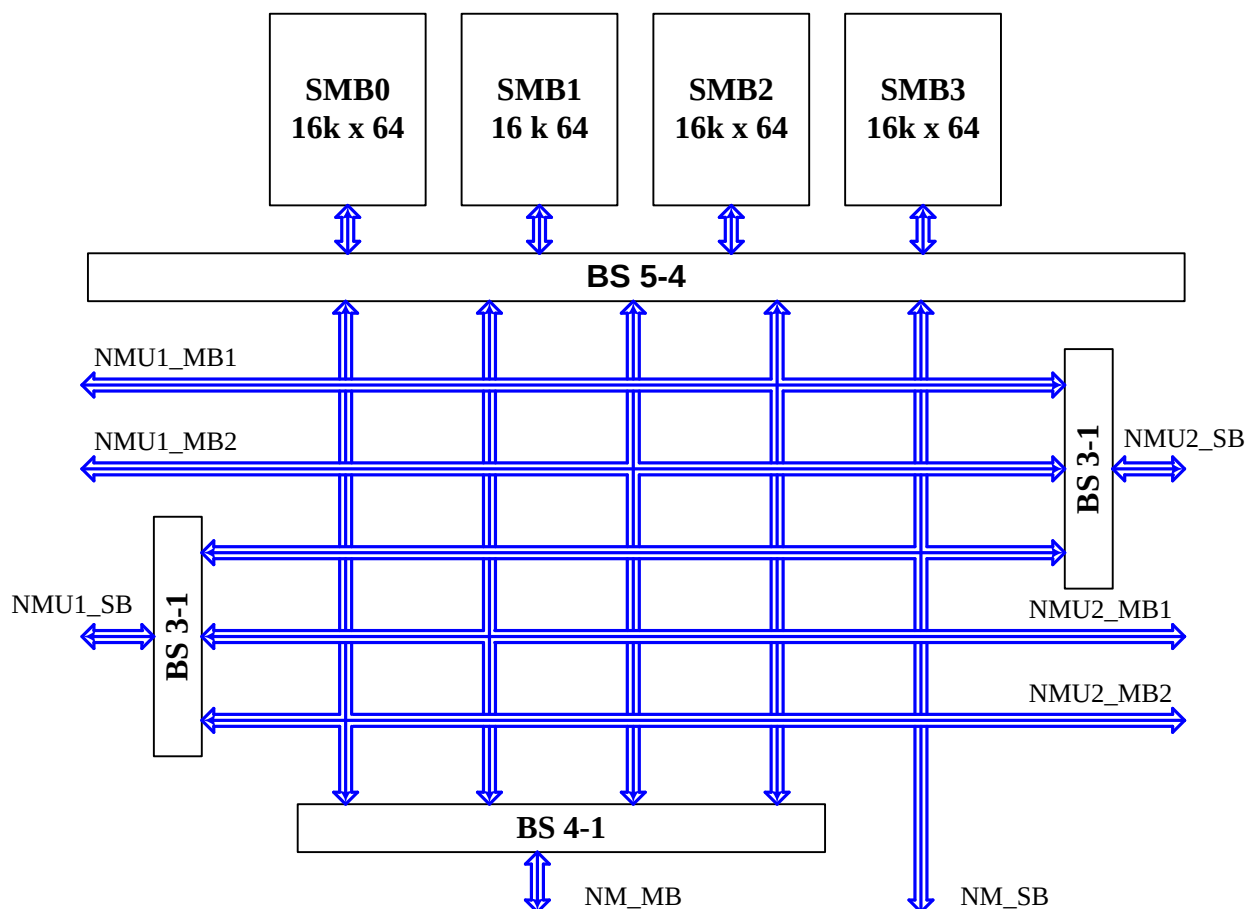


Рисунок 6.1- Структурная схема системы общей памяти (SMU).

Система общей памяти работает на частоте до 328 МГц и содержит следующие функциональные узлы:

BS (Bus Switch) – шинный коммутатор.

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

Все функциональные узлы системы общей памяти соединяются шинами, обозначенными на рисунке одним синим цветом, поскольку все они 64-разрядные и позволяют производить обмен данными на частоте до 328 МГц. Если на один приемник претендует несколько источников, то приоритет определяется следующим образом:

- **NM_SB** имеет наивысший приоритет;
- **NMU1_MB2** приоритетнее **NMU1_MB1**;
- **NMU2_MB2** приоритетнее **NMU2_MB1**;
- Приоритет между запросами **NMU1_MBx** и **NMU2_MBx** определяется регистром **NMUCTRL** в системном контроллере (**ARMSC**) системы **ARMU**. После системного сброса запросы от процессорной системы **NMU2** приоритетнее запросов от процессорной системы **NMU1**.

					ЮФКВ.431268.006РЭ			Лист
								220
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	Решет 16.01.14		22755-1					

7 Блок интервальных таймеров (ITU).

Блок интервальных таймеров (ITU) предназначен для формирования временных интервалов (основного и дополнительного) и импульсов с заданной задержкой и длительностью (программируемый генератор импульсов). Блок ITU работает на частоте до 328 МГц

7.1 Принцип работы блока ITU.

Структурная схема блока интервальных таймеров приведена ниже (см. Рисунок 7.1).

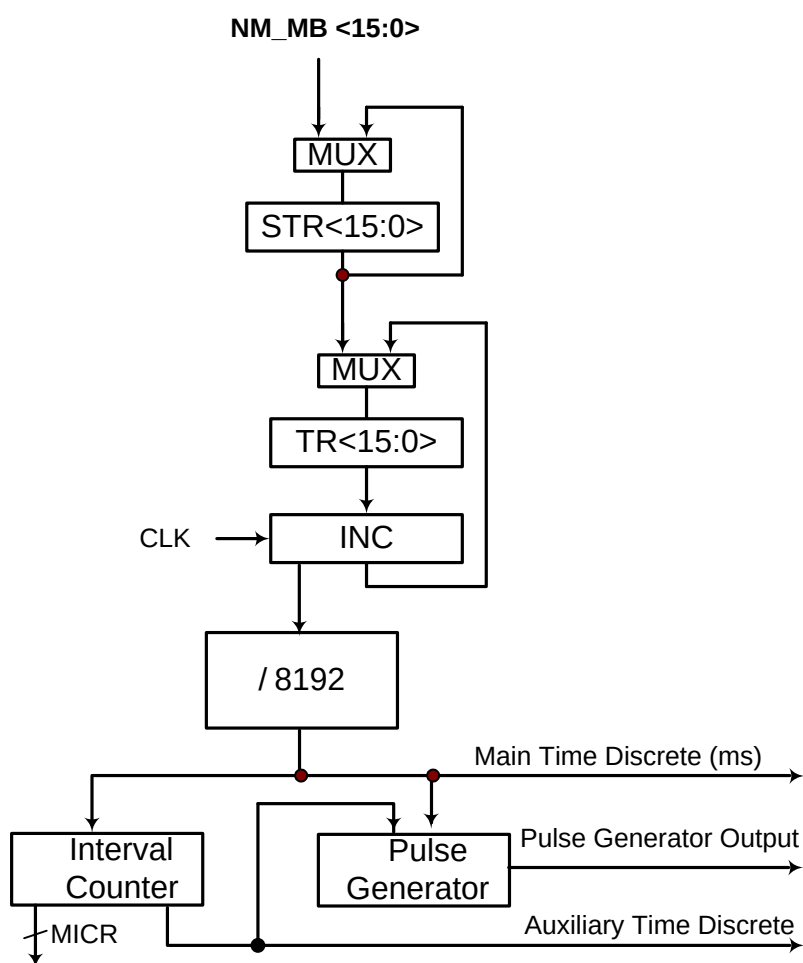


Рисунок 7.1 – Структурная схема блока интервальных таймеров.

Величина временного интервала программно задается в 16-разрядном теновом регистре **STR** в дополнительном коде. Каждый процессорный такт содержимое рабочего регистра **TR** инкрементируется, когда значение **TR** достигнет нулевого значения, выдается импульс на делитель (/8192) и происходит перезапись значения из регистра **STR** в регистр **TR**.

/8192 – делитель, предназначенный для деления частоты поступающих на него импульсов в 8192 раза. На выходе делителя формируется основной временной интер-

					ЮФКВ.431268.006РЭ			Лист
								221
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	Решет 16.01.14		22755-1					

вал (**Main Time Discrete**). Таким образом, период основного временного интервала может задаваться в пределах от 8192 до 536870912 тактов тактового сигнала с шагом в 8192 тактов тактового сигнала.

Сигнал основного интервала времени подается на интервальный счетчик **Interval Counter**. Структурная схема интервального счетчика представлена на рисунке ниже (см. Рисунок 7.2).

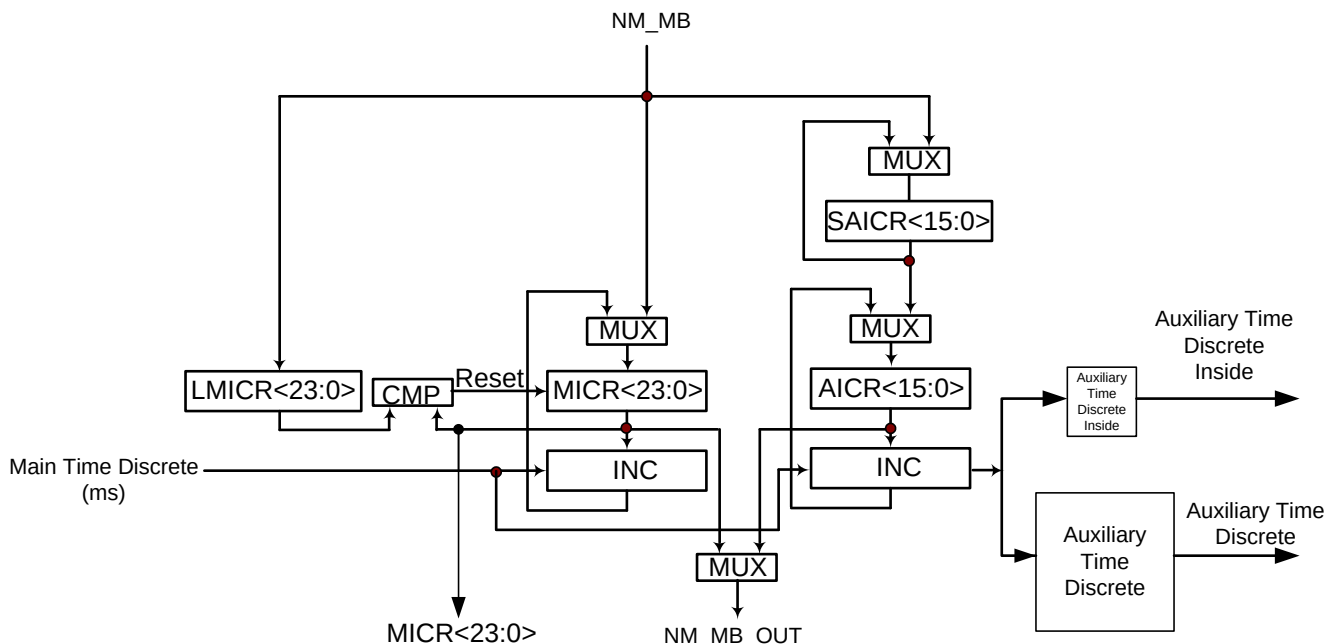


Рисунок 7.2 – Структурная схема интервального счетчика.

Интервальный счетчик состоит из основного и вспомогательного. Основной счетчик осуществляет циклический счет основных временных интервалов. Его регистр **MICR (Main Interval Count Register)** доступен программисту по чтению и по записи. Содержимое регистра **MICR** также записывается в заголовок пакета данных при изменении состояния вывода **Main Time Discrete (ms)** во время работы каналов ПДП “DSPPU – память”. По сигналу системного сброса счетчик принимает значение 000000h. Другой программно-доступный регистр – **LMICR (Last Main Interval Count Register)** – служит для обнуления **MICR** при достижении им заданного значения. После системного сброса содержимое **LMICR** равно всем единицам (FFFFFFh), и данное значение не влияет на работу **MICR**.

Дополнительный счетчик осуществляет формирование дополнительного временного интервала – **Auxiliary Time Discrete** для вторичной обработки информации. Счётчик содержит программно доступные регистры: на запись – теневой **SAICR (Shadow Auxiliary Interval Count Register)**, на чтение – рабочий регистр **AICR (Auxiliary Interval Count Register)**. Каждый основной интервал времени содержимое рабочего регистра **AICR** инкрементируется, когда значение **AICR** достигнет нулевого значения, формируется однитактовый импульс **Auxiliary Time Discrete** для подачи на программируемый генератор импульсов и 16-тактный для подачи на внешний вывод микросхемы **ITO**.

Для синхронизации нескольких микросхем по положительному фронту внешнего сигнала **ITSTROB** текущее состояние делителя (13 разрядов), регистра **TR** (16 разрядов) и регистра **MICR** (24 разряда) записываются в регистр **SPECR**, который про-

					ЮФКВ.431268.006РЭ				Лист
									222
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	16.01.14		22755-1						

граммно анупен по чтению. Кроме того, в этот момент взводится бит **F** в регистре **CR (Condition Register)**, который программно доступен на чтение и обнуляется после чтения регистра **SPECR**.

Сигналы основного и дополнительного интервалов времени поступают также на программируемый генератор импульсов. Основное назначение программируемого генератора импульсов заключается в формировании импульса заданной длительности, фронт которого с высокой точностью синхронизован с эталонной шкалой времени. Структурная схема генератора представлена ниже (см. Рисунок 7.3). Значение программно доступного регистра **SELR** определяет, какой сигнал будет использоваться для запуска генератора (основной или дополнительный временной интервал).

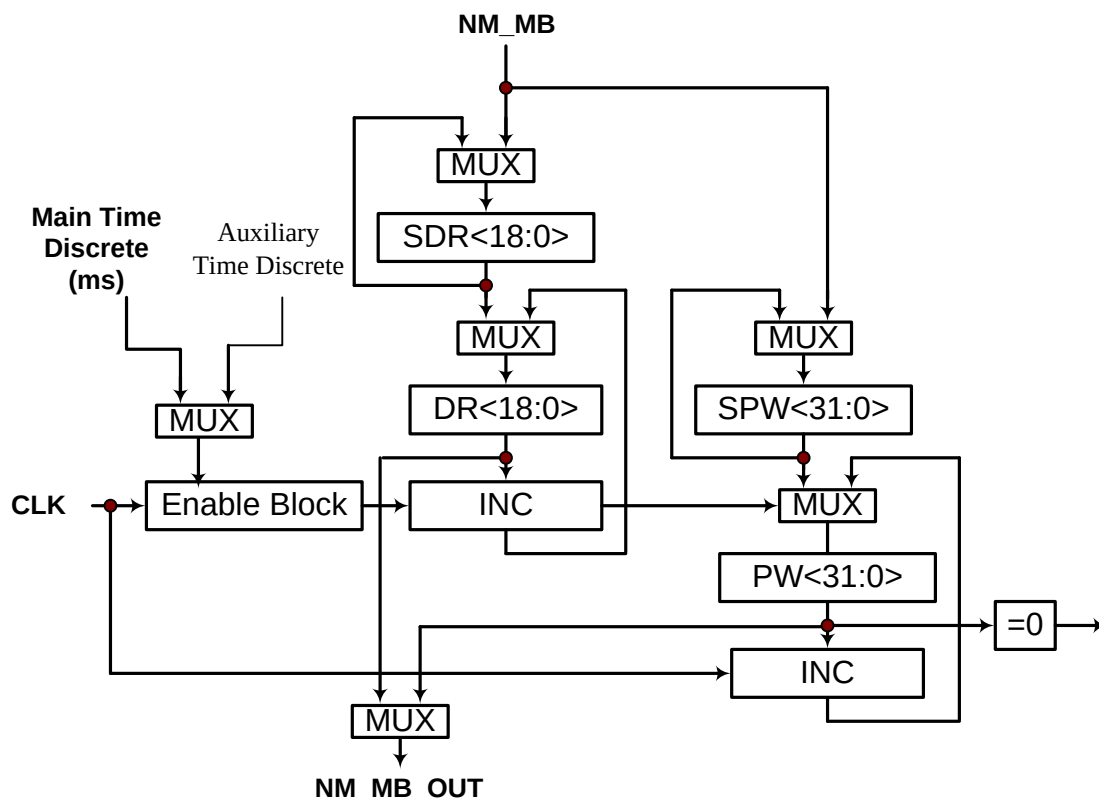


Рисунок 7.3- Структурная схема программируемого генератора импульсов.

Для выдачи указанного импульса в программно-доступный регистр **SDR (Shadow Delay Register)** записывается величина задержки в дополнительном коде. Задержка может варьироваться в пределах от одного периода тактового сигнала (код 7FFFFh) до 524288 периодов тактового сигнала (код 00000h). После запуска генератора содержимое теневого регистра **SDR** переписывается в регистр **DR**, после чего значение регистра **DR** инкрементируется с частотой тактового сигнала до тех пор, пока **DR** не обнулится. После этого запускается схема формирования импульса заданной длительности. В её состав входят программно доступные регистры: на запись – теневой **SPW (Shadow Pulse Width)**, на чтение – рабочий регистр **PW (Pulse Width)**. Для формирования импульса нужной длительности в регистр **SPW** записывается величина длительности в дополнительном коде. После запуска схемы формирования импульса заданной длительности содержимое **SPW** переписывается в **PW**, и **PW** инкрементируется с частотой тактового сигнала до тех пор, пока не сформируется перенос из старшего разряда, и после этого **PW** не обнулится. Схема формирования импульса заданной длитель-

					ЮФКВ.431268.006РЭ			Лист
								223
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	16.01.14		22755-1					

ности формирует на выходе программируемого генератора импульс до тех пор, пока содержимое **PW** не равно нулю. Выход программируемого генератора импульсов поступает на вывод микросхемы **IT_1SO**.

Чтобы повторно сформировать импульс, необходимо осуществить запись в регистр **SDR**. Запись можно производить сразу после запуска генератора.

Сигналы основного, дополнительно временных интервалов и импульса заданной длительности подаются на входы контроллеров внешних прерываний процессорных систем **NMU1**, **NMU2** и **ARMU**.

7.2 Конфигурационные регистры блока ITU.

Управление блоком **ITU** может осуществляться процессорной системой **NMU1** или **NMU2**. Чтение и запись конфигурационных регистров блока **ITU** производится скалярными командами ввода/вывода и равносильно чтению и записи в определенные ячейки памяти. Адреса регистров блока **ITU** для процессорной системы **NMU1** и **NMU2** совпадают и приведены ниже в таблице (Таблица 7.1).

Таблица 7.1 – Адреса регистров блока ITU в адресном пространстве NMU1 и NMU2.

	Источник	Адрес вектор hex	Регистр доступен на	
			Чтение	Запись
1	STR	4004_0000		+
2	SAICR	4004_0002		+
3	AICR	4004_0004	+	
4	LMICR	4004_0006		+
5	MICR	4004_0008	+	+
6	SPECR_LO	4004_000A	+	
7	SPECR_HI	4004_000C	+	
8	CR	4004_000E	+	
9	SDR	4004_0010		+
10	DR	4004_0010	+	
11	SPW	4004_0012		+
12	PW	4004_0012	+	
13	SELR	4004_0016		+

Регистр **STR** доступен на запись. Формат регистра приведен ниже (см. Рисунок 7.4).

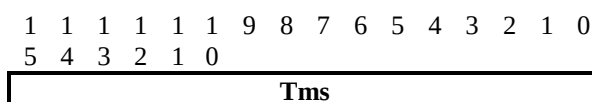


Рисунок 7.4 – Формат регистра STR.

Поле **Tms** (15-0 разряды) определяет период основного временного интервала. После системного сброса **Tms** = **FFD8h**, что соответствует периоду основного временного интервала 327680 тактам тактового сигнала.

Регистры **SAICR** и **AICR** – теневой и рабочий регистры, определяющие период дополнительного временного интервала. Регистр **SAICR** доступен на запись, регистр **AICR** доступен на чтение. Формат регистров приведен ниже (см. Рисунок 7.5).

					ЮФКВ.431268.006РЭ				Лист
									224
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0
5 4 3 2 1 0

Taux

Рисунок 7.5 – Формат регистров SAICR и AICR.

Поле **Taux** (15-0 разряды) определяет период дополнительного временного интервала. Период дополнительного временного интервала может задаваться в пределах от 1 до 65536 периодов основного временного интервала.

Taux = FFFFh – период дополнительного временного интервала 1 период основного;

Taux = FFFEh – период дополнительного временного интервала 2 периода основного;

...

Taux = 0000h – период дополнительного временного интервала 65536 периодов основного;

После системного сброса значение поля Taux принимает значение FF9Ch, что соответствует периоду дополнительного временного интервала равного 100 периодам основного временного интервала.

Регистр **MICR** – 24-х разрядный счетчик основных временных интервалов, доступен на запись и чтение. Регистр содержит номер основного временного интервала. После системного сброса значение регистра равно нулю.

Регистр **LMICR** – 24-х разрядный регистр, определяющий сброс регистра **MICR**, доступен на запись. После системного сброса значение регистра имеет значение FFFFFFFh.

Регистр **SPECR** состоит из двух частей старшей **SPECR_HI** и младшей **SPECR_LO**. Обе части доступны на чтение. Формат регистров **SPECR_HI** и **SPECR_LO** приведен ниже (см. Рисунок 7.6 Рисунок 7.7)

3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0
1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 0 0 0 MICR

Рисунок 7.6 – Формат регистра SPECR_LO.

3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0
1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
0 0 0 D<12:0> TR<15:0>

Рисунок 7.7 – Формат регистра SPECR_HI.

Поле **MICR** содержит значение регистра **MICR** в момент в момент прихода положительного фронта сигнала **ITSTROB**.

Поле **TR<15:0>** содержит значение регистра **TR** в момент в момент прихода положительного фронта сигнала **ITSTROB**.

Поле **D<12:0>** содержит значение счетчика делителя в момент прихода положительного фронта сигнала **ITSTROB**.

Регистр состояния **CR (Condition Register)** доступен на чтение. Формат регистра приведен ниже (см. Рисунок 7.8).

					ЮФКВ.431268.006РЭ				Лист
									225
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	Решал 16.01.14		22755-1						

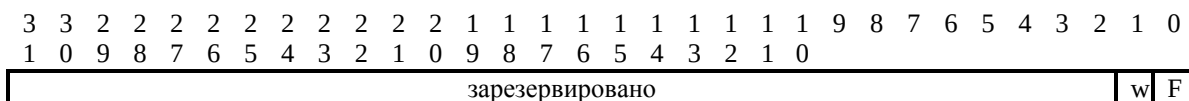


Рисунок 7.8 – Формат регистра CR.

Бит **F** (0-й разряд) показывает, что производилась синхронизация с помощью сигнала **ITSTROB**. Бит устанавливается, когда приходит положительный фронт **ITSTROB**. Сброс происходит после чтения регистра **SPECR_HI** или **SPECR_LO**.

Бит **w** (1-й разряд) информирует о работе программируемого генератора импульсов. Бит устанавливается, когда происходит запись в регистр **SDR**, показывая, что генератор будет формировать импульс после прихода сигнала запуска. Сброс бита происходит после прихода запускающего импульса, показывая, что может быть произведена запись регистра **SDR** для следующего импульса.

Регистр **SELR** определяет, по какому сигналу будет запускаться программируемый генератор импульсов. Регистр доступен на запись. Формат регистра **SELR** приведен ниже (см. Рисунок 7.9).

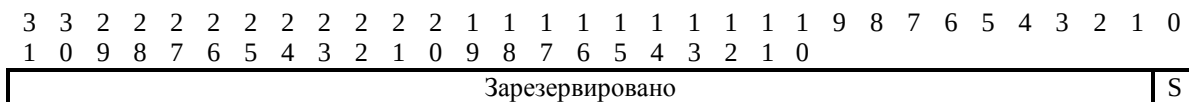


Рисунок 7.9 – Формат регистра SELR.

- Поле **S**:
- 0 – запускающий сигнал – сигнал основного временного интервала.
 - 1 – запускающий сигнал – сигнал дополнительного временного интервала.

Регистры **SDR** и **DR** – теневого и рабочий регистры, определяющие задержку перед выдачей импульса программируемым генератором импульсов. Регистр **SDR** доступен на запись, регистр **DR** доступен на чтение. Формат регистров приведен ниже (см. Рисунок 7.10).

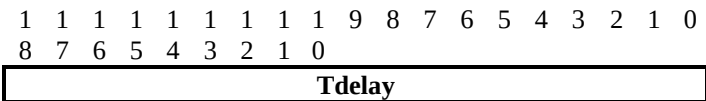


Рисунок 7.10 – Формат регистров SDR и DR.

Поле **Tdelay** (15-0 разряды) определяет время задержки.

Регистры **SPW** и **PW** – теневого и рабочий регистры, определяющие длительность импульса программируемым генератором импульсов. Регистр **SPW** доступен на запись, регистр **PW** доступен на чтение. Формат регистров приведен ниже (см. Рисунок 7.11).

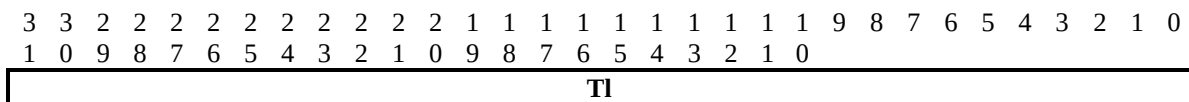


Рисунок 7.11 – Формат регистров SPW и PW.

Поле **TI** (15-0 разряды) определяет длительность импульса.

8 Контроллер ПДП (DMAC)

Контроллер ПДП (**DMAC**) обеспечивает обмен между памятью и блоком **DSPPU** и обмен память – память. Обмен осуществляется 64-х разрядными словами. Контроллер управляется процессорными системами **NMU1** и **NMU2**.

8.1 Принцип работы блока DMAC.

Структурная схема контроллера ПДП приведена ниже (см. Рисунок 8.1).

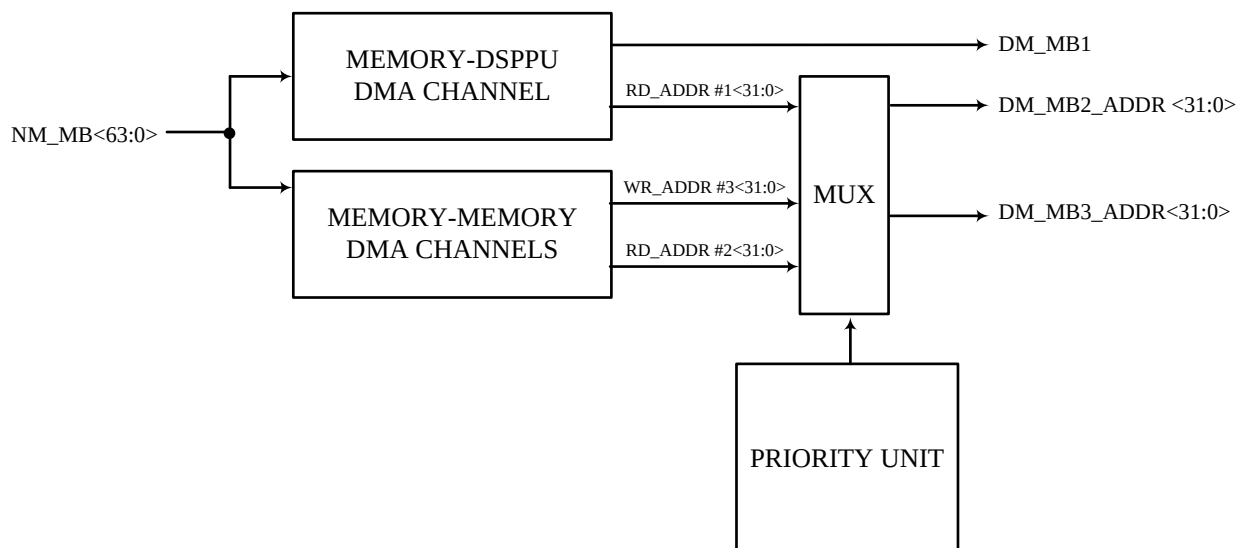


Рисунок 8.1- Структурная схема контроллера ПДП.

Контроллер ПДП работает на частоте до 328 МГц. Он содержит следующие функциональные узлы:

MEMORY – DSPPU DMA CHANNEL – канал ПДП, обеспечивающий подкачку данных из памяти в блок **DSPPU**. Более подробно работа канала будет описана ниже.

MEMORY-MEMORY DMA CHANNELS – 2 канала ПДП типа «память – память». Более подробно работа каналов будет описана ниже.

PRIORITY UNIT – блок приоритетов. Обеспечивает арбитраж при попытке обратиться на одну и ту же шину нескольких каналов ПДП. Наибольший приоритет имеет канал **MEMORY – DSPPU**, затем идут каналы **MEMORY-MEMORY**. При обращении на одну и ту же шину каналов **MEMORY-MEMORY** происходит их чередование, т.е. в одном такте выдается на шину адрес чтения, а в следующем адрес записи.

8.1.1 Каналы ПДП “память – память” (MEMORY-MEMORY DMA CHANNELS)

Структурная схема каналов ПДП “память – память” приведена ниже (см. Рисунок 8.2).

					ЮФКВ.431268.006РЭ			Лист
								227
Изм.	Лист	№ докум.	Подп.	Дата				
Инва.№подл.	Подп. и дата		Взам.инв.№		Инва.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

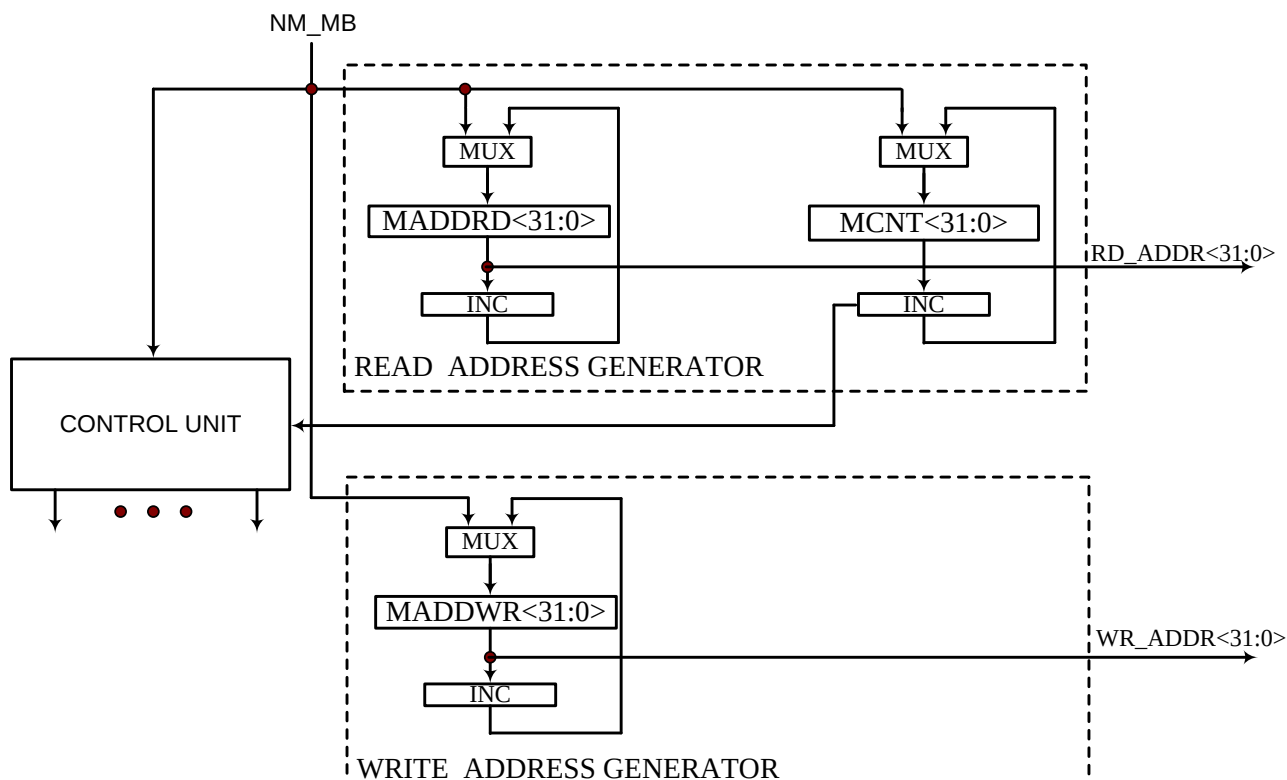


Рисунок 8.2 - Структурная схема каналов ПДП “память – память”.

Каналы ПДП “память – память” содержат следующие функциональные узлы:

READ ADDRESS GENERATOR – генератор адреса для канала ПДП чтения из памяти. Он содержит следующие программно доступные на запись регистры: **MADDRD** – регистр текущего адреса, который после запуска каналов ПДП формирует в режиме инкремента очередной адрес чтения из памяти, и **MCNT** – счётчик читаемых 64-х разрядных слов из памяти. Минимальное число слов, которое можно задать для счётчика – 1, максимальное – 2^{32} . Чтобы задать число слов для чтения из памяти, равное N, необходимо задать в регистре **MCNT** число $(-1 * N)$ в дополнительном коде. Единице соответствует код FFFF_FFFFh, числу 2^{32} соответствует код 0000_0000h. При работе канала **MCNT** инкрементируется каждый раз, когда отправлен запрос на чтение от канала ПДП, пока не появится перенос из старшего разряда. В этом случае канал ПДП чтения из памяти останавливается, а **MCNT** обнуляется.

WRITE ADDRESS GENERATOR – генератор адреса для канала ПДП записи в память. Он содержит только один программно-доступный на запись регистр: **MADDWR** – регистр текущего адреса, который после запуска каналов ПДП и чтения очередного данного в режиме инкремента каналом чтения, формирует очередной адрес записи в память. Канал ПДП записи в память продолжает свою работу до тех пор, пока не обнулится **MCNT** и не будут записаны все считанные каналом ПДП чтения из памяти данные. После окончания работы канала записи может быть выдано прерывание процессорным системам **NMU1** и **NMU2**.

8.1.2 Канал ПДП “память – DSPPU”

Структурная схема канала ПДП “память – DSPPU” приведена ниже (см Рисунок 8.3).

					ЮФКВ.431268.006РЭ			Лист
								228
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата			
22755-2	Редько 16.01.14		22755-1					

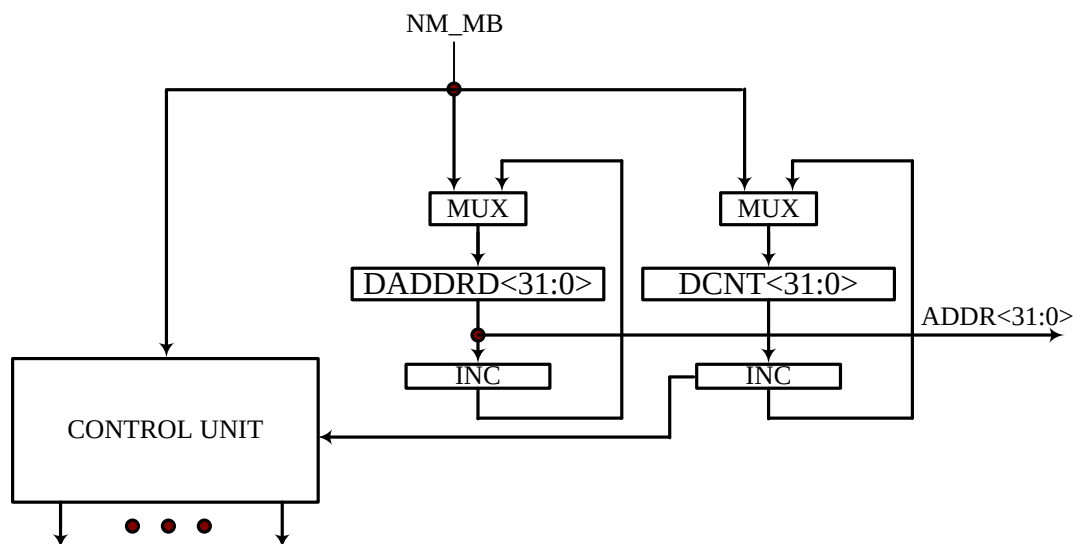


Рисунок 8.3 - Структурная схема канала ПДП “память – DSPPU”

Канал ПДП “память – DSPPU” содержит следующие функциональные узлы:

DADDRD – программно-доступный регистр текущего адреса канала ПДП, который после запуска канала формирует в режиме инкремента очередной адрес ПДП (ADDR).

DCNT – программно-доступный регистр числа читаемых из памяти 64-разрядных слов в режиме ПДП. Минимальное число слов, которое можно задать – 1, максимальное – 2^{32} . Чтобы задать число слов для чтения из памяти, равное N, необходимо задать в регистре **DCNT** число $(-1 * N)$ в дополнительном коде. Единице соответствует код FFFF_FFFFh, числу 2^{32} соответствует код 0000_0000h. При работе канала **DCNT** инкрементируется каждый раз, когда запрос от канала ПДП обслужен, пока не появится перенос из старшего разряда. В этом случае канал ПДП останавливается. После окончания работы канала может быть выдано прерывание процессорным системам NMU1 и NMU2.

8.2 Конфигурационные регистры блока DMAC

Управление блоком **DMAC** может осуществляться процессорной системой NMU1 или NMU2. Чтение и запись конфигурационных регистров блока **ITU** производится 32-х разрядными скалярными командами ввода/вывода и равносильно чтению и записи в определенные ячейки памяти. Адреса регистров блока **DMAC** для процессорной системы NMU1 и NMU2 совпадают и приведены ниже в таблице (Таблица 8.1).

					ЮФКВ.431268.006РЭ			Лист
								229
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

Таблица 8.1 – Адреса регистров блока DMAC в адресном пространстве NMU1 и NMU2.

	Источник	Адрес вектор hex	Регистр доступен на	
			Чтение	Запись
1	DMACR	4002_0102	+	+
2	DADDRD	4002_00F8		+
3	DCNT	4002_00FA		+
4	MADDRD	4002_00FC		+
5	MADDWR	4002_00FE		+
6	MCNT	4002_0100		+

Регистр управления **DMACR (DMA Control Register)** (см. Рисунок 8.4) доступен на запись и чтение со стороны процессорной системы **NMU1** и **NMU2**. Если было выдано прерывание процессорным системам, то первая команда записи в регистр **DMACR** произведет сброс сигнала прерывания. При этом в команде записи должен быть сброшен источник прерывания (бит **IRQM** или бит **IRQD**) иначе сигнал прерывания взведется в следующем такте.

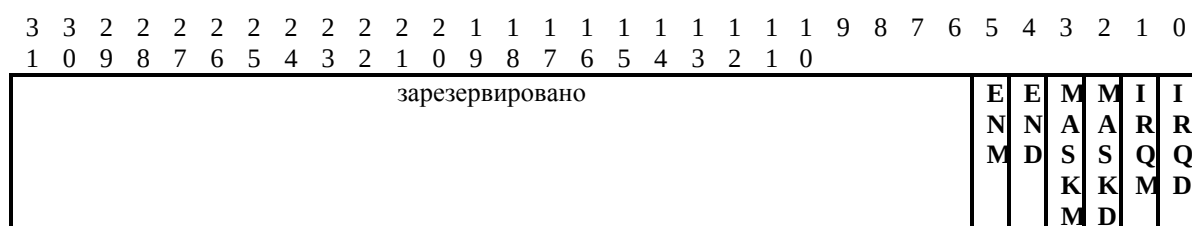


Рисунок 8.4 – Формат регистра DMACR.

Бит **IRQD** (0-й разряд) устанавливается после окончания работы канала ПДП “память – DSPPU” (после того как будет передано последнее слово в блок DSPPU). Показывает, что канал ПДП “память – DSPPU” закончил свою работу. Если при этом сброшен бит **MASKD**, то будет выдано прерывание процессорным системам **NMU1** и **NMU2**. Сброс данного бита осуществляется записью “1” в этот разряд командой записи. Программно установить данный бит в “1” нельзя.

Бит **IRQM** (1-й разряд) устанавливается после окончания работы ПДП канала записи “память – память” Показывает, что канал ПДП “память – память” закончил свою работу (отправлены последние данные на запись). Если при этом сброшен бит **MASKM**, то будет выдано прерывание процессорным системам **NMU1** и **NMU2**. Сброс данного бита осуществляется записью “1” в этот разряд командой записи. Программно установить данный бит в “1” нельзя.

Бит **MASKD** (2-й разряд) – маска прерывания канала “память – DSPPU”.

0 – прерывание не маскируется;

1 – прерывание маскируется.

Бит **MASKM** (3-й разряд) – маска прерывания канала “память – память”.

0 – прерывание не маскируется;

1 – прерывание маскируется.

Бит **END** (4-й разряд) – бит разрешения работы канала “память – DSPPU”.

0 – канал “память – DSPPU” не отправляет запрос на чтение данных;

1 – канал “память – DSPPU” отправляет запрос на чтение данных.

Данный бит устанавливается программно. После этого канал ПДП “память – DSPPU” начинает выдавать запросы на чтение данных. После выдачи необходимого числа данных (определяется регистром **DCNT**) происходит аппаратный сброс данного

					ЮФКВ.431268.006РЭ					Лист
										230
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата				
22755-2	<i>Редкал</i> 16.01.14		22755-1							

бита, и канал ПДП больше не выдает запросы на чтение. Сброс данного бита может быть осуществлен программно.

Бит **ENM** (5-й разряд) – бит разрешения работы канала “память – память”.

0 – канал “память – память” не отправляет запрос на чтение данных;

1 – канал “память – память” отправляет запрос на чтение данных.

Данный бит устанавливается программно. После этого канал ПДП “память – память” начинает выдавать запросы на чтение данных. После выдачи необходимого числа данных (определяется регистром **MCNT**) происходит аппаратный сброс данного бита, и канал ПДП больше не выдает запросы на чтение. Сброс данного бита может быть осуществлен программно.

Регистр **DCNT<31:0>** (см. Рисунок 8.5) – 32-х разрядный регистр числа читаемых из памяти 64-разрядных слов каналом ПДП “память – DSPPU”. Регистр доступен на запись со стороны процессорной системы **NMU1** и **NMU2**. Минимальное число слов, которое можно задать – 1, максимальное – 2^{32} . Чтобы задать число слов для чтения из памяти, равное N, необходимо задать в регистре **DCNT** число $(-1 * N)$ в дополнительном коде. Единице соответствует код FFFF_FFFFh, числу 2^{32} соответствует код 0000_0000h.

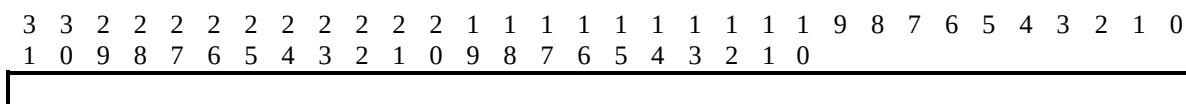


Рисунок 8.5 – Формат регистра DCNT.

Регистр **DADDRD<31:0>** (см. Рисунок 8.6) – 32-х разрядный регистр текущего адреса канала ПДП “память – DSPPU”, который после запуска канала в режиме инкремента формирует очередной адрес ПДП. Регистр доступен на запись со стороны процессорной системы **NMU1** и **NMU2**.

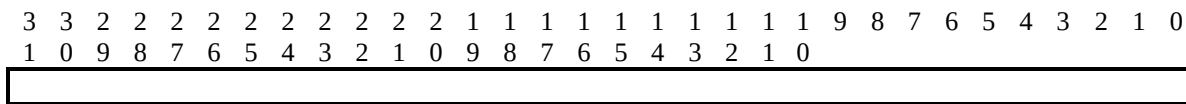


Рисунок 8.6 – Формат регистра DADDRD.

Регистр **MCNT<31:0>** (см. Рисунок 8.7) – 32-х разрядный регистр числа читаемых из памяти 64-разрядных слов каналом ПДП “память – память”. Регистр доступен на запись со стороны процессорной системы **NMU1** и **NMU2**. Минимальное число слов, которое можно задать – 1, максимальное – 2^{32} . Чтобы задать число слов для чтения из памяти, равное N, необходимо задать в регистре **MCNT** число $(-1 * N)$ в дополнительном коде. Единице соответствует код FFFF_FFFFh, числу 2^{32} соответствует код 0000_0000h.

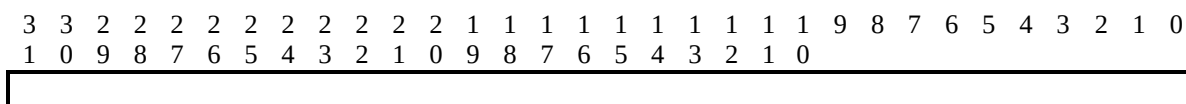


Рисунок 8.7 – Формат регистра MCNT.

					ЮФКВ.431268.006РЭ				Лист
									231
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

Регистр **MADDRD<31:0>** (см. Рисунок 8.8) – 32-х разрядный регистр текущего адреса чтения канала ПДП “память – память”, который после запуска канала в режиме инкремента формирует очередной адрес чтения ПДП. Регистр доступен на запись со стороны процессорной системы **NMU1** и **NMU2**.

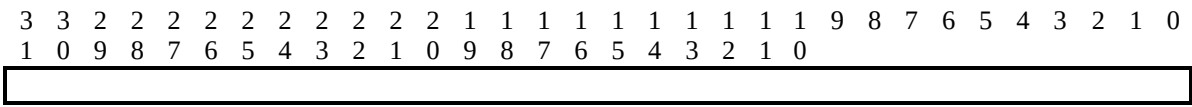


Рисунок 8.8 – Формат регистра MADDRD.

Регистр **MADDRD<31:0>** (см. Рисунок 8.9) – 32-х разрядный регистр текущего адреса, который после запуска каналов ПДП “память – память” и чтения очередного анного каналом ПДП чтения в режиме инкремента, формирует очередной адрес записи в память в режиме ПДП.

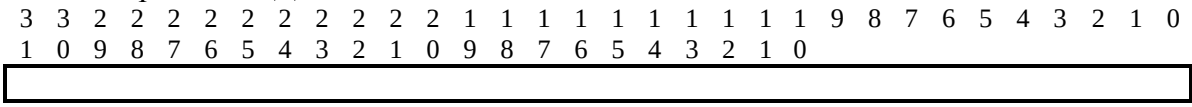
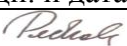


Рисунок 8.9 – Формат регистра MADDWR.

					ЮФКВ.431268.006РЭ				Лист
									232
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата			Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14			22755-1				

9 Блок аналогово-цифрового преобразования (ADCU).

Блок аналогово-цифрового преобразования (ADCU) состоит из четырёх 12-разрядных АЦП, работающих на частоте до 82 МГц. Структурная схема блока показана ниже (см. Рисунок 9.1).

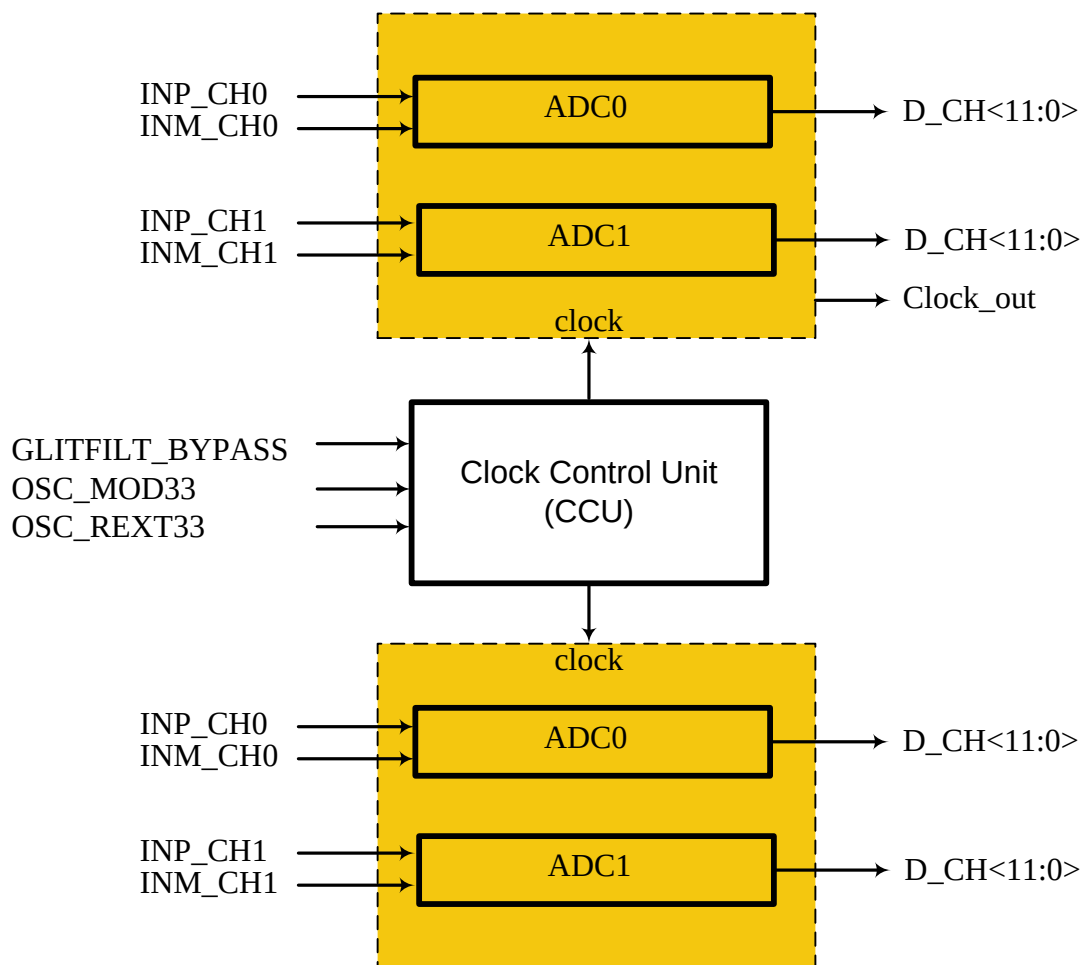


Рисунок 9.1 – Структурная схема блока ADCU

Четыре АЦП объединены в блоки попарно: ADC0, ADC1 и ADC2, ADC3. На вход каждой пары могут быть поданы независимые аналоговые сигналы или квадратурные составляющие I и Q одного сигнала. Для этого каждая пара АЦП имеет общий источник напряжения смещения и общую систему питания, что обеспечивает идентичные характеристики преобразования.

Блок управления тактовым сигналом (CCU) формирует тактовый сигнал 80МГц, который подается на АЦП. Из пары ADC0, ADC1 тактовый сигнал выдается внутрь микросхемы и является опорным для PLL цифрового генератора (входит в состав CRG11) формирующего все тактовые сигналы микросхемы. Блок CCU может работать в двух режимах: формировать тактовый сигнал самостоятельно, используя внешний кварцевый резонатор (режим генератора), или пропускать внешний тактовый сигнал (режим входного буфера). Выбор режима работы определяется сигналом OSC_MOD33. Предусмотрена возможность использовать внутренний резистор обрат-

					ЮФКВ.431268.006РЭ				Лист
									233
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

ной связи. Выбор производится выводом **OSC_REXT33**. Возможные режимы работы приведены в таблице ниже (Таблица 9.1).

Таблица 9.1 – Режимы работы блока CCU

Режим	OSC_MOD33	OSC_REXT33
Режим генератора, используется внутренний резистор обратной связи	0	0
Режим генератора, должен использоваться внешний резистор обратной связи	0	1
Режим буфера, используется внутренний резистор обратной связи	1	0
Режим буфера, внутренний резистор обратной связи не используется.	1	1

Тактовый сигнал может быть пропущен через внутренний фильтр помех или пройти мимо него. Это определяется внешним выводом **GLITFILT_BYPASS**:

- 0 – используется внутренний фильтр помех;
- 1 – не используется внутренний фильтр помех.

Есть возможность управления потреблением и работой блока **ADCU**. Управление осуществляется процессорной системой **ARMU** путем записи в управляющий регистр **ADCCTRL** контроллера **ARMSC**. Формат регистра **ADCCTRL** приведен ниже (см. Рисунок 9.2).

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
reserved												HALFPW R_CH	PDNPIPE _CH	P D N _ A	P D N _ B	P D N C L K _ B														

Рисунок 9.2 – Формат регистра ADCCTRL

Поле **PDNCLK_B** (0-й разряд) позволяет отключить подачу тактового сигнала на пару АЦП ADC2,ADC3.

- 0 – тактовый сигнал подается на пару АЦП ADC2,ADC3;
- 1 – тактовый сигнал не подается на пару АЦП ADC2,ADC3;

Поле **PDN_B** (1-й разряд) позволяет отключить подачу питания на пару АЦП ADC2,ADC3.

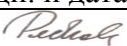
- 0 – питание подается на пару АЦП ADC2,ADC3;
- 1 – питание не подается на пару АЦП ADC2,ADC3;

Поле **PDN_A** (2-й разряд) позволяет отключить подачу питания на пару АЦП ADC1,ADC0.

- 0 – питание подается на пару АЦП ADC1,ADC0;
- 1 – питание не подается на пару АЦП ADC1,ADC0;

Поле **PDNPIPE_CH<3:0>** (6-3-й разряды) позволяет отключить конвейер каждого АЦП.

- PDNPIPE CH <3> = 1** конвейер **ADC3** выключен,
- PDNPIPE CH <3> = 0** конвейер **ADC3** включен,
- PDNPIPE CH <2> = 1** конвейер **ADC2** выключен,
- PDNPIPE CH <2> = 0** конвейер **ADC2** включен,

					ЮФКВ.431268.006РЭ					Лист
										234
Изм.	Лист	№ докум.	Подп.	Дата						
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.		Подп. и дата			
22755-2	 16.01.14		22755-1							

PDNPIPE CH <1> = 1 конвейер **ADC1** выключен,
PDNPIPE CH <1> = 0 конвейер **ADC1** включен,
PDNPIPE CH <0> = 1 конвейер **ADC0** выключен,
PDNPIPE CH <0> = 0 конвейер **ADC0** включен,

Поле **HALFPWR_CH<3:0>** (10-7-й разряды) позволяет перевести АЦП в режим половинной мощности. В этом режиме частота работы АЦП снижается в 2 раза.

HALFPWR_CH <3> = 0 – **ADC3** работает в нормальном режиме;
HALFPWR_CH <3> = 1 – **ADC3** работает в режиме половинной мощности;
HALFPWR_CH <2> = 0 – **ADC2** работает в нормальном режиме;
HALFPWR_CH <2> = 1 – **ADC2** работает в режиме половинной мощности;
HALFPWR_CH <1> = 0 – **ADC1** работает в нормальном режиме;
HALFPWR_CH <1> = 1 – **ADC1** работает в режиме половинной мощности;
HALFPWR_CH <0> = 0 – **ADC0** работает в нормальном режиме;
HALFPWR_CH <0> = 1 – **ADC0** работает в режиме половинной мощности;

					ЮФКВ.431268.006РЭ			Лист
								235
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редко</i> 16.01.14		22755-1				

10 Блок предварительной обработки сигналов (DSPPU)

Блок предварительной обработки сигналов (DSPPU) представляет собой многоканальное конвейерное настраиваемое программно устройство потоковой цифровой обработки сигналов. Блок DSPPU состоит из 24 аппаратных каналов, в которых могут осуществляться следующие операции:

- выбор источника сигнала (выход одного из ADC блока ADCU, входы SIGNx и MAGNx или внутренняя память через DMA),
- цифровое гетеродинирование с последующим накоплением,
- фильтрация помех и возведение комплексного сигнала в квадрат,
- упаковка сигналов в 64-разрядные слова и запись в память упакованных данных.

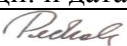
Структурная схема блока DSPPU приведена ниже (см. Рисунок 10.1).

В описании блока DSPPU и системы команд используется следующая терминология:

- *канал (аппаратный канал предварительной обработки сигналов)* – совокупность аппаратных блоков, между которыми данные передаются по цепочке от одного блока к другому от интерфейса с входным устройством (Input_unit) до блоков упаковки. «Раздвоенный» канал считается одним каналом. Каждый канал имеет свой гетеродин (с него начинается) и заканчивается одним или двумя интерфейсами с блоками упаковки.
- *канал упаковки* – входной интерфейс блока упаковки.
- *метка* – периодически приходящий к блоку DSPPU основной временной интервал от блока интервальных таймеров. Есть возможность программно включить входной делитель меток (устройство, позволяющее в n раз сократить частоту прихода меток) – в таком случае ниже в тексте под *меткой* обычно следует понимать именно результирующий сигнал.
- *временной интервал* – время между двумя метками.
- *кадр* – массив данных, сформированный за время одного временного интервала одним каналом.
- *кластер каналов* – совокупность четырех каналов, объединённых структурно.
- *кластер фильтров* – вычислительный блок, принимающий на вход до 4-х потоков данных, производящий цифровую фильтрацию (КИХ) и выдающий до 4-х потоков данных – результатов обработки. Кластер фильтров выполняет функцию одного или нескольких КИХ-фильтров конфигурируемого порядка с конфигурируемой разрядностью коэффициентов.

Непрерывные потоки оцифрованных данных проходят через предварительно настроенные и запущенные аппаратные каналы, результаты обработки упаковываются и записываются в память для дальнейшей обработки. Временные метки, источником которых является блок интервальных таймеров (см. п. 7), разделяют эти потоки на кадры.

В качестве входных сигналов можно также использовать и данные, хранящиеся во внутренней или внешней памяти микросхемы. В этом случае необходимо предварительно настроить канал DMA “память-DSPPU” в блоке DMAC (см. п. 8) на пересылку данных из памяти в блок DSPPU. Блок данных, передаваемый за один запуск канала DMA “память-DSPPU”, считается одним кадром.

					ЮФКВ.431268.006РЭ			Лист
								236
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

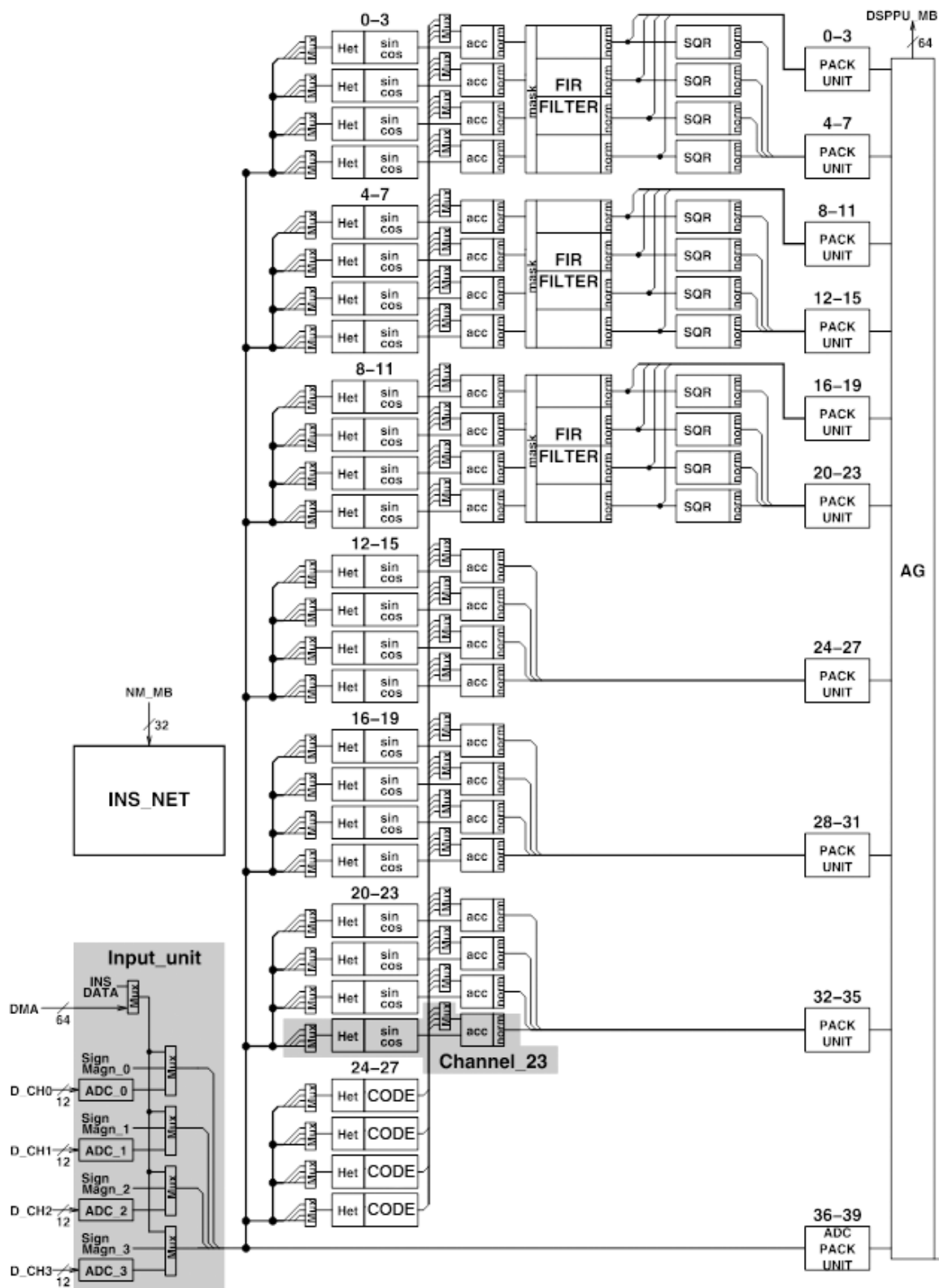


Рисунок 10.1 – Структурная схема блока предварительной обработки сигналов

Запуск и останов канала, корректировка каналов либо полная перестройка каналов осуществляется синхронно и может производиться только на границе временных интервалов.

					ЮФКВ.431268.006РЭ			Лист
								237
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		<i>Редко</i> 16.01.14		22755-1				

Управление блоком DSPPU осуществляется процессорной системой NMU1 или NMU2. Чтение и запись конфигурационных регистров блока DSPPU производятся скалярными командами и равносильны чтению и записи в определенные ячейки памяти. Адресное пространство регистров блока DSPPU для процессорной системы NMU1 и NMU2 совпадает и расположено по адресам 4001_0000h – 4001_FFFh.

Система команд блока DSPPU представлена в приложении (см. п. А). В нем дается полное описание всех команд, форматов, режимов и правил программирования блока DSPPU. Блок работает на частоте тактового сигнала Fclk до 328 МГц

DSPPU содержит следующие функциональные узлы:

Input_unit – входное устройство, осуществляющее выбор источника сигналов и передающее сигналы на один из своих 4-х выходных интерфейсов. Возможны следующие источники сигналов: блок аналогово-цифрового преобразования ADCU, внешние входы микросхемы Signx/Magnx, память микросхемы (с использованием DMA), непосредственно команды записи (на входы INS_DATA). Последний режим удобен для проверки работоспособности блока DSPPU. Также блок включает в себя схему синхронизации работы блоков ADCU и DSPPU, схему калибровки данных ADCU, делители временных меток.

Channel – аппаратные каналы предварительной обработки сигналов, осуществляющие цифровое гетеродинирование, накопление и прореживание потоков 12-разрядных данных, которые поступают с выходного интерфейса блока Input_unit с темпом Fclk/4. Выходной поток данных канала состоит из комплексных чисел (8 разрядов + 8 разрядов), выдаваемых с темпом до (Fclk/4). Блок DSPPU содержит 24 канала с номерами от 0 до 23.

CODE – блоки генерации псевдослучайных последовательностей, формирующие четыре битовые псевдослучайные последовательности, которые используются затем при обработке входного потока данных в аппаратных каналах блока DSPPU.

FIR_FILTER – кластер КИХ-фильтров, обслуживающий одновременно до 4-х аппаратных каналов. Кластер фильтров состоит из 4-х перестраиваемых секций, которые содержат буферы FIFO для данных и умножители с накопителями. Секции можно коммутировать последовательно и/или параллельно для реализации от одного до 4-х КИХ-фильтров различного типа. Входной и выходной потоки данных для кластера КИХ-фильтров состоят из комплексных данных (8 разрядов + 8 разрядов), поступающих с темпом до (Fclk/16) МГц.

DSPPU содержит три кластера КИХ-фильтров. Фильтрами снабжены только 12 аппаратных каналов с номерами от 0 до 11.

SQR – блок квадраторов, обслуживающий поток данных, поступающий с выхода КИХ-фильтров. Данный блок осуществляет возведение в квадрат входных комплексных отсчетов и накопление результата. Входной поток состоит из комплексных данных (8 разрядов + 8 разрядов), получаемых с темпом до (Fclk/4) МГц. Выходной поток – комплексных данных (8 разрядов + 8 разрядов), выдаваемых с темпом от (Fclk/8192) кГц до (Fclk/4) МГц. В DSPPU имеется двенадцать блоков квадраторов, обслуживающих аппаратные каналы с номерами от 0 до 11.

PACK_UNIT – девять блоков упаковки, на которые приходят результаты предварительной обработки сигналов, данные упаковываются в 64-разрядные слова для последующей их записи в память. Один блок обслуживает одновременно до 4-х каналов упаковки. Входной поток состоит из комплексных данных (8 разрядов + 8 разрядов), получаемых с темпом до (Fclk/4) МГц. Выходной поток – 64-разрядные упакованные данные, выдаваемые с темпом до (Fclk/16) МГц.

					ЮФКВ.431268.006РЭ				Лист
									238
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		<i>Редько</i> 16.01.14		22755-1					

ADC_PACK_UNIT – один блок упаковки 12-разрядных отсчётов ADC в 64-разрядные слова для последующей их записи в память. Блок выполняет функции, аналогичные блоку PACK_UNIT.

AG – блок адресных генераторов, осуществляющий запись упакованных данных в заранее заданные области памяти. Для каждого канала паковки выделяются две области в памяти – четная и нечетная. В каждую такую область памяти непрерывно следующие друг за другом кадры записываются попеременно, что позволяет проводить запись и обработку бесконечно длинных последовательностей сигналов, при условии что обработка данных текущего кадра успевает производиться за время, меньшее времени формирования следующего кадра.

Кадр, упакованный и сохраненный в памяти, имеет заголовок, который содержит размер кадра, номер канала, другие признаки кадра и формируется только тогда, когда все данные кадра полностью записаны в память. Заголовок помещается в младшую ячейку области памяти, выделенной под кадр (сами данные располагаются ниже заголовка). Размер выделенной области памяти должен быть достаточен для записи всех настроенных и запущенных каналов обработки данных. Оценить размер памяти, необходимый для выделения под каждый канал, можно по следующей формуле:

$$V_{\text{озу}} = [F_{\text{скл}}/1000 * (T_{\text{инт}} * L_{\text{кадр}}) / (D_{\text{акк1}} * D_{\text{акк2}} * 8/P) + 16],$$

где $V_{\text{озу}}$ – объем памяти в байтах,

$F_{\text{скл}}$ – частота тактового сигнала, выраженная в Гц,

$T_{\text{инт}}$ – значение периода в миллисекундах, с которым интервальным таймером генерируются метки, – определяется настройкой интервального.

$L_{\text{кадр}}$ – количество меток от интервального таймера во временном интервале (коэффициент, учитывающий настройку входного делителя меток) – может быть задано в диапазоне от 1 до 4096,

$D_{\text{акк1}}$ – коэффициент накопления потока данных в первом накопителе – может быть задан в диапазоне от 1 до 4096,

$D_{\text{акк2}}$ – коэффициент накопления потока данных во втором накопителе – может быть задан в диапазоне от 1 до 4096,

P – число бит упакованного результата обработки (для комплексного числа требуется хранить два результата) – может принимать значения {4, 8, 16}.

Приоритет выборки данных из блоков упаковки PACK при записи в память кадров линейно уменьшается с увеличением номера канала паковки (от 0 до 39). Блок работает на частоте до 328 МГц.

Пропускная способность интерфейса к внутренней памяти составляет около 2,6 Гбайт/сек, поэтому суммарный пиковый поток упакованных данных с учетом добавленных заголовков не должен превышать этой величины. В противном случае часть данных кадра, относящаяся к конкретному моменту времени, когда был превышен пиковый поток, не будет записана в память. В этом случае аппаратура сформирует сигнал контроля. Работа остальных каналов не нарушится. Аппаратура блока DSPPU не позволяет выполнять запись потока данных в память вне кадра.

INS_NET – схема управления. Блок DSPPU имеет глубокий конвейер обработки, в котором данные, метки времени и параметры настройки аппаратуры последовательно передаются от одного блока к другому. Поэтому команды инициализации управляющих регистров и команды пуска каналов тоже организованы конвейерным образом. INS_NET обеспечивает однозначное поведение DSPPU относительно последовательности команд и асинхронного прихода меток.

Каждый аппаратный канал имеет биты состояния: TT_stop, T_start, T_stop. Биты состояния могут быть считаны командами RD_het_base, RD_het_frequency, RD_het_current.

					ЮФКВ.431268.006РЭ			Лист
								239
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

Бит TT_Stop указывает текущее состояние канала: запущен канал или остановлен. Биты T_Start, T_Stop показывают, в какое состояние должен перейти канал по приходу метки.

T_start = 1 – выполняет два действия: запускает канал, если он был остановлен (сбрасывает бит управления TT_stop), и переписывает настройки из теневых регистров в рабочие. Оба действия выполняются на границе временных интервалов. Установка бита T_start производится командой WR_start.

T_stop = 1 – останавливает канал по приходу ближайшей метки. Установка бита T_stop осуществляется командой WR_Stop или WR_stop_ms.

Таким образом, для запуска канала после записи всех настроек следует выдать команду WR_start и в T_start установится значение «1». По приходу метки TT_stop и T_start обнуляются, настройки переписываются в рабочие регистры и данные начинают поступать в канал.

Коррекция настроек канала делается аналогично. При работающем канале запись производится в теневые регистры канала, затем следует выдать команду WR_start и после прихода ближайшей метки новые настройки вступают в силу.

Остановить канал можно командами WR_stop, WR_stop_ms. Если выдать работающему каналу одну из этих команд, устанавливается бит T_stop. По ближайшей метке канал переходит в исходное состояние TT_stop = 1, T_start = 0, T_stop = 0.

Аппаратура контролирует неправильную последовательность действий аннелеемы, приводящую к неоднозначности работы канала. Если выдать команду WR_start и начать, не дожидаясь метки, новую настройку канала, то можно повредить непредсказуемым образом настройки канала. При обнаружении блоком подобной ситуации выдается сигнал прерывания (который, впрочем, можно маскировать).

10.1 Типы данных, используемые DSPPU

DSPPU может работать с несколькими типами входных и выходных данных. Эти типы различаются разрядностью, а также делятся на комплексные и действительные.

Типы входных данных:

- 12-разрядные действительные знаковые отсчёты, получаемые от ADC или из команд записи,
- 2-разрядные данные в кодировке sgmg, получаемые с внешних входов микросхемы SIGN0 и MAGN0, SIGN1 и MAGN1, SIGN2 и MAGN3, SIGN3 и MAGN3, кодировка показана в таблице ниже (Таблица 10.1),
- упакованные sgmg или 4-, 8- и 16-разрядные действительные знаковые данные, получаемые из памяти через DMA, должны находиться в памяти выровненными 64-разрядными словами.

Типы выходных данных DSPPU:

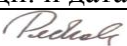
- заголовки кадров – специальные структуры (их формат показывает Таблица 10.2),
- 16-разрядные действительные знаковые данные, упакованные в 64-разрядные слова,
- комплексные данные с 2-, 4- и 8-разрядными знаковыми составляющими.

Подробное описание выходных данных дано в п. 10.9.

					ЮФКВ.431268.006РЭ			Лист
								240
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

Таблица 10.1 – Кодировка sgm

SIGNx	MAGNx	Значение
0	0	+1
0	1	+2
1	0	-1
1	1	-2

					ЮФКВ.431268.006РЭ			Лист
								241
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

10.2 Входное устройство (Input_unit)

Структурная схема входного устройства Input_unit приведена на рисунке ниже (см. Рисунок 10.2).

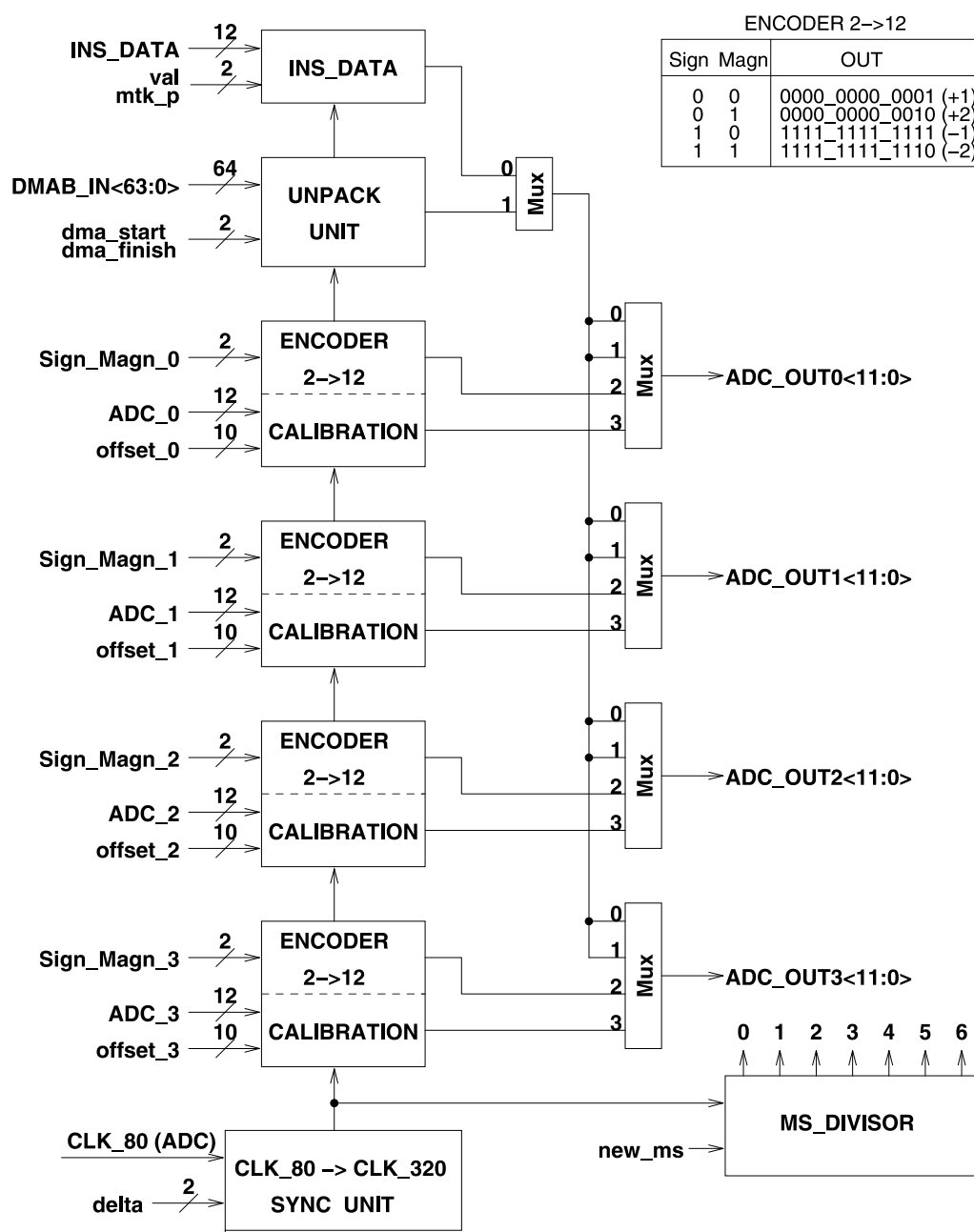


Рисунок 10.2 – Структурная схема входного устройства

Блок **SYNC UNIT** обеспечивает синхронизацию DSPPU с источниками сигналов. Входные сигналы от ADC и с входов SIGNx/MAGNx поступают с частотой (Fclk/4) МГц. Для подстройки фазы, в которой фиксируется входной сигнал, используется устройство SYNC UNIT, которое раз в 4 такта формирует опорный сигнал Strob, поступающий на все блоки DSPPU. Первоначально, во время сброса DSPPU, сигнал

					ЮФКВ.431268.006РЭ	Лист
						242
Изм.	Лист	№ докум.	Подп.	Дата		
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата
22755-2		 16.01.14		22755-1		

Strob формируется (см. Рисунок 10.3) из переднего фронта тактового сигнала с частотой ($F_{clk}/4$) МГц (clk_80). Затем Strob можно переместить относительно clk_80 в любое другое положение командой WR_ADC_delta . Команда чтения RD_ADC_check позволяет обнаруживать ошибки синхронизации.

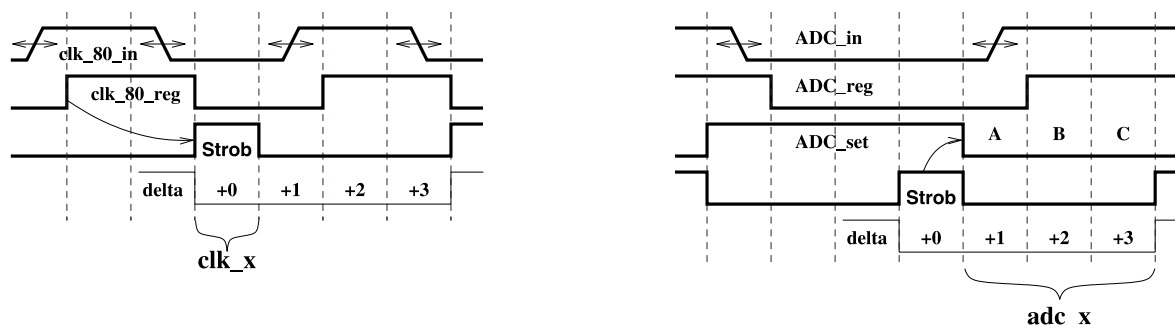


Рисунок 10.3 – Успешная синхронизация данных при переходе с clk_80 на clk_320

На рисунке выше слева (см. Рисунок 10.3) показано первоначальное положение сигнала $Strob$, которое фиксируется после сброса DSPPU. Справа показаны, в общем случае, асинхронный сигнал, приходящий от ADC (ADC_in), сигнал на выходе синхронизирующего триггера в блоке DSPPU (ADC_reg), сигнал фиксированных данных (ADC_set). ADC_set фиксирует данные в момент прихода $Strob$. Важно, чтобы сигнал $Strob$ не попадал на такты, в которых ADC_reg переключается. Синхронизацию можно считать успешной, если сигнал ADC_reg на следующий такт после фиксации (в точке А на рисунке) остаётся на таком же уровне, а через 2 такта (в точке С) меняется. Аппаратурой постоянно производится сравнение зафиксированного значения со значениями ADC_reg через 1, 2 и 3 такта. Сравнение производится по всем разрядам всех выходов ADC ($12 \times 4 = 4$ блока ADC по 12 разрядов выходной шины данных каждый). Любое несовпадение при сравнении приводит к тому, что соответствующий бит поля adc_x принимает значение 1. Командой RD_ADC_check можно прочитать поле adc_x , при этом поле adc_x обнуляется.

ENCODER 2->12 – преобразователь 2-разрядного кода $sgmg$ в 12-разрядный дополнительный код по правилу, представленному на рисунке выше (см. Рисунок 10.2).

CALIBRATION – блок калибровки оцифрованного сигнала ADC. Нулевое значение входного сигнала может отображаться не нулевым кодом (может иметь место эффект смещения нуля). Команда $WR_ADC_calibration$ записывает в регистры калибровки 10-разрядные значения в дополнительном коде, которые суммируются с оцифрованными сигналами. Предварительно, в блоке калибровки, оцифрованный сигнал переводится из прямого кода в дополнительный код. Суммирование производится с насыщением.

UNPACK UNIT – устройство распаковки 64-разрядных данных из канала DMA «память – DSPPU», которые могут использоваться для дальнейшей обработки вместо отсчётов ADC. Режим распаковки задается командой $WR_ADC_control$ и определяется типом передаваемых каналом DMA данных. Допускаются следующие 4 типа данных: упакованный 2-разрядный код $sgmg$ и упакованные целые 4-, 8- и 16-разрядные числа в дополнительном коде. Все описанные типы преобразуются к 12 разрядам. Одновременно с 64-разрядными данными на UNPACK UNIT поступают признаки первого сло-

					ЮФКВ.431268.006РЭ			Лист
								243
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1					

ва пакета (dma_start) и последнего слова пакета (dma_finish). С их помощью формируется кадр (они заменяют метки). Пакет может состоять из одного слова.

INS_DATA – устройство, формирующее данные из команд записи. Разряды команды [11:0] (Data) WR_ADC_ins рассматриваются как 12-разрядные оцифрованные значения, разряд [12] (V) определяет наличие данных в команде (1 – данные принимаются, 0 – данные не принимаются), [13] (M) – эмулирует приход метки. Возможность программно передавать блоку DSPPU данные удобно использовать для отладки.

Манипулируя битами V и M команды, можно формировать короткие кадры, «пустые» послы V = M = 0, а также метки без данных.

MS_DIVISOR – делитель меток. Устройство содержит 6 12-разрядных счетчиков, управляемых командой WR_Ms_counter. Каждый счетчик состоит из 2-х регистров – теневого и рабочего. При поступлении очередной метки от интервального таймера (см. п.7) из рабочих регистров вычитается 1. Когда счетчик достигает «0», формируется импульс на соответствующем (1-м, 2-м, 3-м, 4-м, 5-м или 6-м) выходе блока и счетчик перезагружается из теневого регистра. Нулевой выход повторяет исходную метку. В дальнейшем, каждый канал настраивается так, чтобы он принимал в качестве метки один из семи выходных сигналов делителя меток.

Mux – мультиплексоры выбора источника данных, которые управляются командой WR_ADC_control.

10.3 Цифровой гетеродин (Het)

Структурная схема цифрового гетеродина приведена на рисунке ниже (см. Рисунок 10.4).

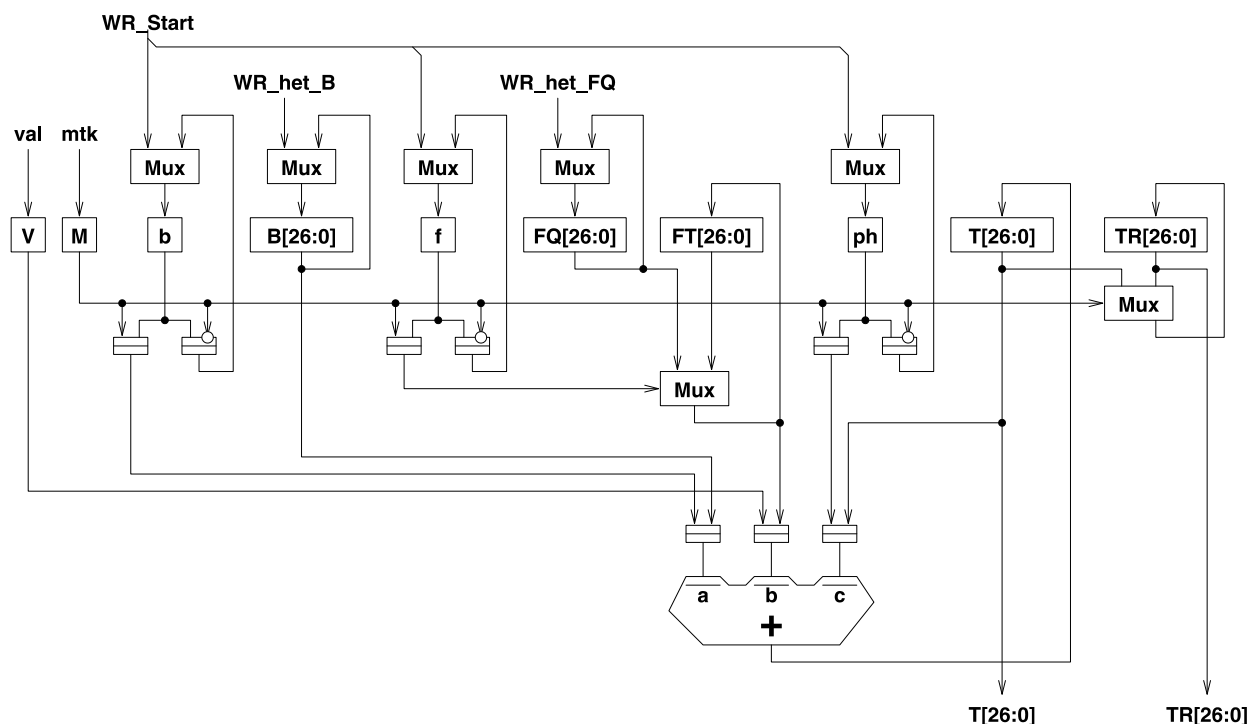


Рисунок 10.4 – Структурная схема гетеродина

Цифровые гетеродины используются в блоках генерации псевдослучайных последовательностей CODE и в аппаратных каналах предварительной обработки сигнала.

					ЮФКВ.431268.006РЭ			Лист
								244
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

лов (Channel). Гетеродин генерирует меняющееся во времени значение фазы синусоидального сигнала.

Командами WR_het_B и WR_het_FQ настраиваются начальная фаза и частота гетеродина. Команда WR_Start устанавливает синхробиты b, f и rh. Синхробит b используется для установки начальной фазы, синхробит f для установки частоты, и синхробит rh используется для обнуления текущего значения гетеродина. По приходу метки, если установлены все три синхробита, значения B[26:0] и FQ[26:0] передаются на сумматор $a + b + c$, а текущее значение гетеродина T[26:0] обнуляется и на суммирование не подается. При этом последнее текущее значение гетеродина T[26:0] (от предыдущего кадра) сохраняется в регистре TR[26:0]. Схема суммирования аппаратно контролируется по модулю 3.

Сигнал val используется в режиме, когда данные передаются блоку DSPPU командами записи или через канал DMA «память – DSPPU». В этом случае возможна задержка в подкачке данных, и val приостанавливает работу гетеродина. То есть гетеродин переключается только пока по конвейеру DSPPU движутся данные.

Команда WR_Start обеспечивает одновременный запуск всех выбранных гетеродинов.

Остановка гетеродина осуществляется командой WR_stop_ms или WR_stop.

Формулы для вычисления частоты гетеродина, выраженной в герцах, в зависимости от кода в регистре FQ[26:0] и обратно приведены ниже:

$$f = \text{code} * (\text{Fclk}/4) / 2^{27},$$

$$\text{code} = [f * 2^{27} / (\text{Fclk}/4)],$$

где Fclk – частота тактового сигнала данного блока, выраженная в Гц;

f – частота гетеродина, выраженная в герцах; code – значение регистра FT[26:0].

Формулы для вычисления смещения фазы гетеродина ϕ , выраженной в радианах, в зависимости от кода в регистре B[26:0] и обратно приведены ниже:

$$\phi = \text{code} * 2\pi / 2^{27},$$

$$\text{code} = [\phi * 2^{27} / (2\pi)],$$

где ϕ – смещение фазы гетеродина, выраженное в радианах; code – значение регистра B[26:0].

10.4 Кластер генераторов псевдослучайных последовательностей

Как показано на рисунке ниже (см. Рисунок 10.5), кластер генераторов псевдослучайных последовательностей включает в себя четыре идентичных генератора.

					ЮФКВ.431268.006РЭ			Лист
								245
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редкофф</i> 16.01.14		22755-1				

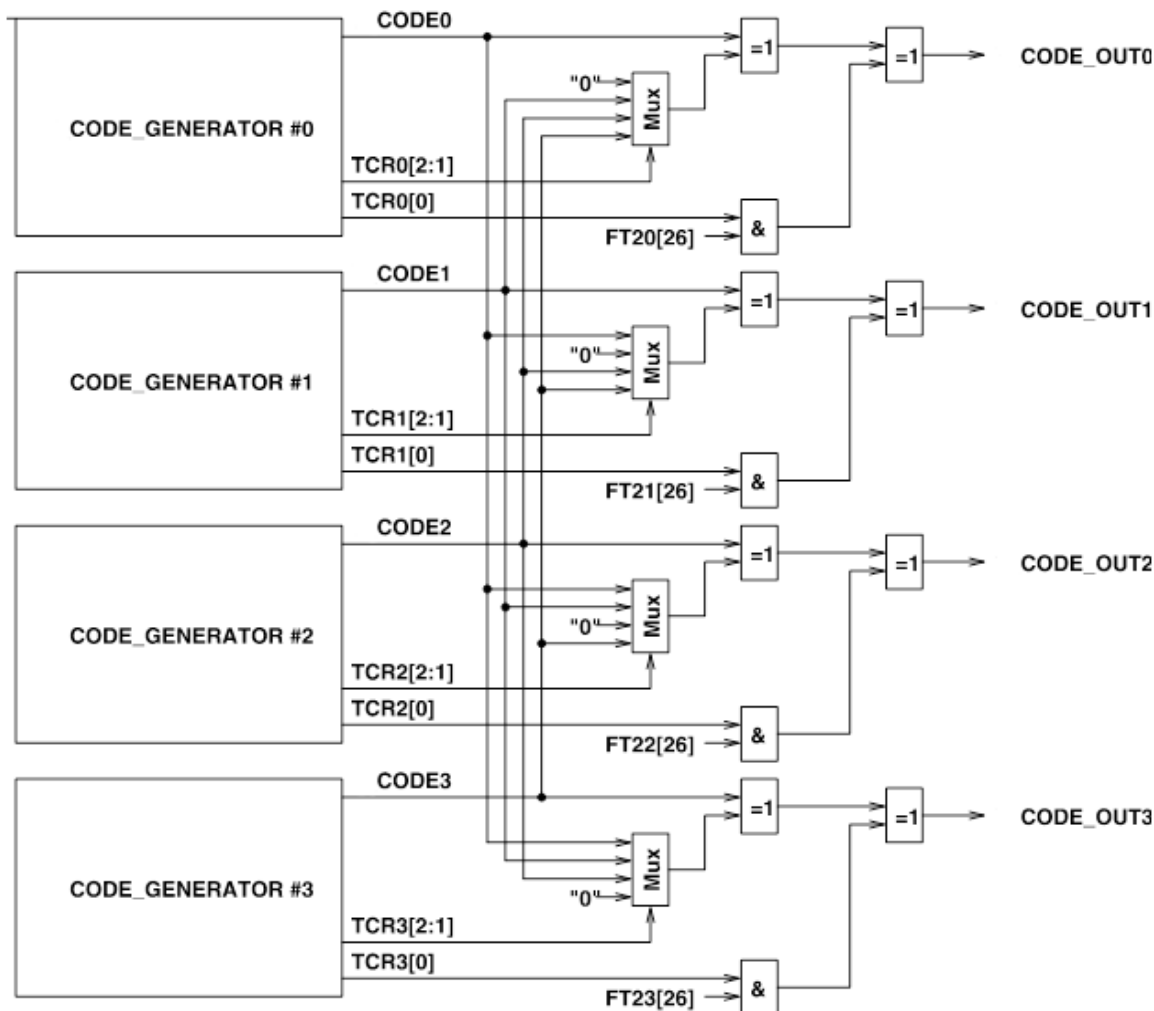


Рисунок 10.5 – Структурная схема кластера генераторов Т-кодов

Каждый из генераторов формирует на своём выходе CODE_i ($i = 0, \dots, 3$) битовую псевдослучайную последовательность заданной длины и длительности (типа М-последовательности). В соответствии с управляющим кодом для каждого генератора TCR_i[2:0] ($i = 0, \dots, 3$) его выход может суммироваться по модулю 2 с выходом другого генератора (формирование кодов Голда) или суммироваться по модулю 2 со старшим разрядом регистра текущей фазы цифрового гетеродина каналов с номерами 20, 21, 22 и 23 блока DSPPU (перенос спектра псевдослучайной последовательности на промежуточную частоту). Таким образом, на выходах кластера генераторов CODE_OUT_i ($i = 0, \dots, 3$) формируются четыре однобитовые псевдослучайные последовательности, которые используются затем при обработке входного потока данных в аппаратных каналах блока DSPPU. Кроме того, выходы CODE_OUT0 – CODE_OUT3 выдаются также на внешние выводы микросхемы.

Генераторы псевдослучайных последовательностей CODE GENERATOR #0 – CODE GENERATOR #3 более подробно описаны в следующем подразделе.

10.4.1 Генератор псевдослучайных последовательностей (CODE GENERATOR)

Структурная схема генератора Т-кодов приведена на рисунке ниже (см. Рисунок 10.6).

					ЮФКВ.431268.006РЭ			Лист
								246
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	Редкофф 16.01.14		22755-1					

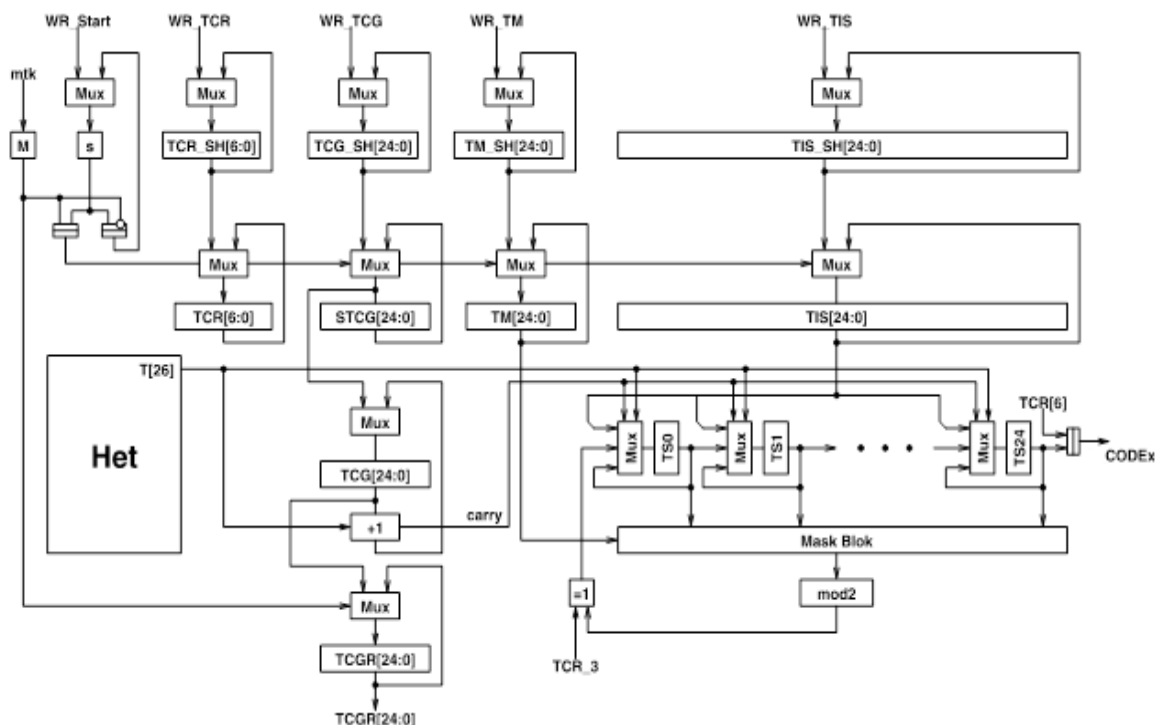


Рисунок 10.6 – Структурная схема генератора T-кодов

Запуск генератора происходит в два этапа. Первым этапом является установка синхробита с командой WR_Start. Вторым этапом является приход временной метки. По приходу метки данные из теневых регистров переписываются в рабочие регистры. После чего генератор начинает работать в заданном режиме.

Генератор содержит следующие функциональные узлы:

Het – Цифровой гетеродин. Описание данного блока представлено в главе 10.3.

TCR_SH[6:0] – 7-разрядный программно-доступный на запись теневой регистр управления генератором. Командой WR_TCR в регистр записываются значения управляющих битов. **TCR[6:0]** – рабочий регистр управления генератором. Генератор может работать в следующих режимах:

- Останов.
- Однократный режим. При однократном запуске генератор, в зависимости от того используется ли счетчик длины последовательности или нет, переходит в режим останова либо по завершении формирования псевдослучайной последовательности, либо программно (соответственно).
- Циклический режим. При циклическом режиме генератор, вне зависимости от того используется ли счетчик длины последовательности или нет, продолжает работать. В режим останова генератор переходит программно.

В однократном и циклическом режимах генератор может работать как используя счётчик длины последовательности, так и не используя. При использовании счетчика формируется псевдослучайная последовательность заданной длины. Если счетчик не используется, то работа генератора завершается программно. Генератор управляет формированием вывода CODEx (см Рисунок 10.6) и формированием выводов CODE_OUT0 – CODE_OUT3 (см Рисунок 10.5).

TCG_SH [24:0] – 25-разрядный программно-доступный на запись теневой регистр длины последовательности генератора. По команде WR_TCG в регистр записы-

					ЮФКВ.431268.006РЭ			Лист
								247
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

вается значение длины последовательности генератора. TCG [24:0] – рабочий регистр длины последовательности генератора. По приходу временной метки данные также переписываются в регистр STCG [24:0]. Формирователь длины псевдослучайной последовательности строится на основе счетчика, управляемого рабочим регистром TCG [24:0]. После этого в каждом такте работы генератора (в качестве тактового сигнала используется выход старшего разряда рабочего регистра текущей фазы T[26] гетеродина) значение счетчика инкрементируется, пока не достигнет значения 1FFFFFFh. Когда значение регистра TCG[24:0]. = 1FFFFFFh, формируется сигнал переноса carry, при условии использования счетчика длины. Если сигнал carry формируется одновременно с приходом метки, то в регистр TCG [24:0] переписывается значение из теневого регистра TCG_SH [24:0], если метка отсутствует, то в регистр TCG [24:0] переписывается значение из регистра STCG [24:0]. При однократном режиме счетчик остановится, а при циклическом продолжит работу. Минимальное число тактов для счёта, которое можно задать – 1, максимальное – 2^{25} . Чтобы задать число слов, равное N, необходимо задать число $(-1 \cdot N)$ в дополнительном коде. Единице соответствует код 1FFFFFFh, числу 2^{25} соответствует код 0000000h.

TCGR [24:0] – 25-разрядный программно-доступный на чтение регистр, хранящий текущее значение счётчика длины последовательности TCG [24:0] в момент прихода временной метки.

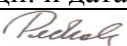
TM_SH [24:0] – 25-разрядный программно-доступный на запись теневой регистр маски. По команде WR_TM в регистр записывается значение маски. TM [24:0] – рабочий регистр маски. Данный регистр определяет, какие разряды кода псевдослучайной последовательности используются для формирования следующего бита этой последовательности.

TIS_SH [24:0] – 25-разрядный программно-доступный на запись регистр начального значения псевдослучайной последовательности генератора. По команде WR_TIS в регистр записывается начальное значение псевдослучайной последовательности генератора. TIS [24:0] – рабочий регистр начального значения псевдослучайной последовательности генератора. По приходу временной метки данные также переписываются в сдвиговый регистр TS [24:0], формирующий псевдослучайную последовательность. После этого в каждом такте работы генератора (в качестве тактового сигнала используется выход старшего разряда рабочего регистра текущей фазы T[26] гетеродина) в TS [24:0] происходит сдвиг на один разряд в сторону старших разрядов. При этом в младший разряд TS [24:0] записывается результат суммы по модулю 2 текущих значений TS [24:0], поразрядно перемноженных (логически) на регистр маски TM [24:0], либо инверсия этого результата (зависит от содержимого 3-го разряда регистра управления генератором). Если разрешена работа счётчика длины последовательности, то в случае установления сигнала carry в сдвиговый регистр TS [24:0] переписывается значение TIS [24:0], и в циклическом режиме работы сдвиговый регистр TS [24:0] продолжит свою работу.

Чтение и запись программно-доступных регистров производится соответствующими командами блока DSPPU.

10.5 Аппаратный канал предварительной обработки сигналов (Channel)

Структурная схема аппаратного канала предварительной обработки сигналов (без фильтров и квадраторов) приведена на рисунке ниже (см. Рисунок 10.7).

					ЮФКВ.431268.006РЭ			Лист
								248
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

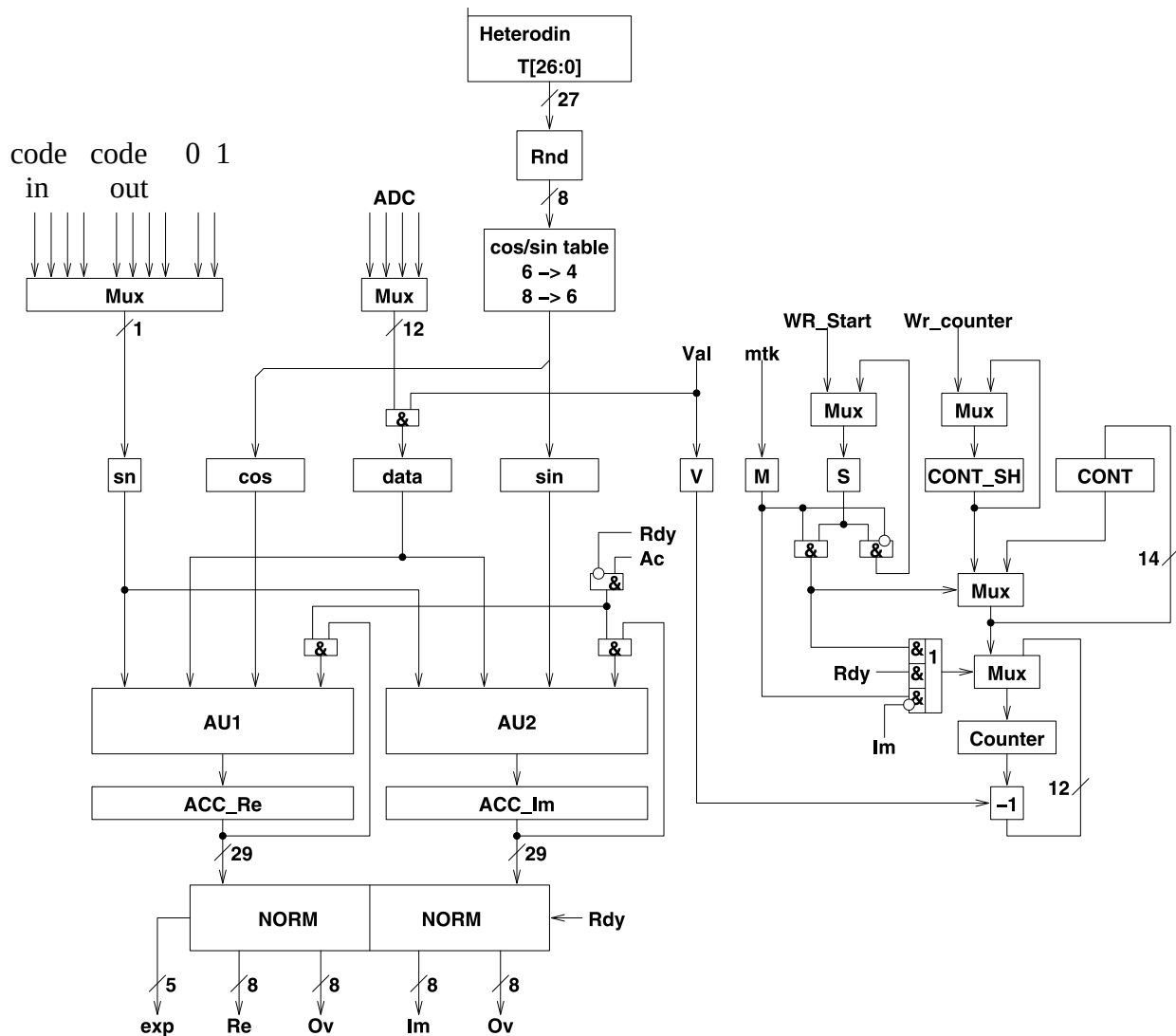


Рисунок 10.7 – Структурная схема аппаратного канала предварительной обработки сигналов

Показанный фрагмент схемы осуществляет преобразование 12-разрядных данных, поступающих с одного из интерфейсов входного устройства Input_unit, в квадратурный комплексный сигнал, а также производит их последующее накопление или прореживание.

Mux_ADC – мультиплексор входных данных. Мультиплексор Mux_ADC настраивается командой WR_select.

Mux_CODE – мультиплексор псевдослучайных последовательностей. Возможно подключение к выходам внутренних генераторов, к входам микросхемы (T13 – T10) для внешних генераторов, а также возможна подача постоянного значения “0” или “1”. Мультиплексор Mux_CODE настраивается командой WR_select.

Cos/sin_table – блок, формирующий значения синуса и косинуса текущей фазы гетеродина T[26:0]. Блок cos/sin_table содержит две таблицы для формирования синуса и косинуса:

- TBL_4 – таблица, формирующая 4-разрядные значения синуса и косинуса,
- TBL_6 – таблица, формирующая 6-разрядные значения синуса и косинуса.

					ЮФКВ.431268.006РЭ			Лист
								249
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		16.01.14		22755-1				

TBL_4 при формировании синуса и косинуса использует старшие 6 разрядов регистра текущей фазы гетеродина: T[26:21]. Таблица TBL_4 приведена в приложении (см. п. В).

При использовании TBL_6 значение T[26:0] предварительно округляется и старшие 8 разрядов результата округления используются для формирования синуса и косинуса.

Два арифметических блока AU1 и AU2 производят следующие вычисления:

$$AU1 = data * \cos * sn + ACC_Re, AU2 = data * \sin * sn + ACC_Im.$$

Каждый блок вычисляет свое выражение за один такт, производя одновременно умножение и суммирование. Сигнал знака sn кодируется следующим образом: если $sn=0$, то умножение производится на +1; если $sn=1$, то умножение производится на -1.

Counter – программно-доступный счетчик, задающий период накопления либо прореживания данных в аккумуляторе ACC_Re и ACC_Im. Командой WR_het_counter загружается теневой регистр CONT_SH. Команда WR_Start взводит синхробит S. По приходу метки и при наличии бита S происходит перезагрузка регистров CONT и Counter.

С приходом каждого данного из Counter вычитается 1. Когда достигается ноль, Counter перезагружается, результаты накопления передаются в блок нормализации, аккумуляторы обнуляются. При приходе метки, если командой WR_het_counter установлен бит Im, частично накопленные данные в аккумуляторах теряются.

NORM – блок нормализации, осуществляющий выбор 8-ми последовательных разрядов аккумулятора для формирования выходных данных. Описание блока нормализации дано ниже (см. п. 10.6).

10.5.1 Округление результатов вычислений в каналах

Многочисленные операции умножения и сложения, проводимые над данными, приводят к тому, что разрядность данных растёт. Округление результатов вычислений в аппаратных каналах предварительной обработки сигналов позволяет снизить разрядность данных за счёт точности.

В DSPPU округление производится в блоках двух типов: в блоках нормализации (см. п. 10.6) и в блоке округления фазы гетеродина (см. Рисунок 10.7, на котором данный блок помечен символами «Rnd»). В обоих случаях применяется один и тот же способ округления. На рисунке ниже (см. Рисунок 10.8) показан пример округления 13-разрядного данного до 8 разрядов. В округляемом слове выбираются биты L, R и S: L – младший разряд округлённого числа, R – старший отбрасываемый бит, S – остальные отбрасываемые биты.

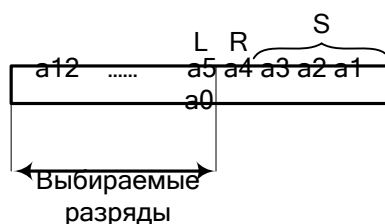


Рисунок 10.8 – Обозначение разрядов округляемого данного

В DSPPU используются следующие способы округления результата:

- to_–inf – биты R и S отбрасываются, результат не меняется,
- to_+inf – к результату прибавляется 1, если $(R + S) > 0$,

					ЮФКВ.431268.006РЭ				Лист
									250
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	Решетов 16.01.14		22755-1						

- to_zero – к отрицательному результату прибавляется 1, если $(R + S) > 0$,
- to_nearest – к результату прибавляется 1, если $R = 1$,
- to_nearest_even – к результату прибавляется 1, если $R \& (L + S) > 0$.

Выбор способа округления производится командами WR_norm и WR_select.

10.6 Блоки нормализации (NORM)

Блоки нормализации служат для уменьшения разрядности данных, накопленных в аккумуляторах каналов, и расположены в конвейере DSPPU после всех имеющихся аккумуляторов (накапливающие сумматоры после гетеродина, сумматоры секций КИХ-фильтра и накапливающие сумматоры квадраторов). Задача блока нормализации – выбор восьми смежных значимых разрядов аккумулятора и передача их дальше по конвейеру.

Какие из разрядов входных данных будут выбираться блоком, можно жёстко задать с помощью поля norm команды WR_norm. Значению 0 соответствует выбор 1-го и 0-го разрядов аккумулятора и дополнение их шестью нулями (см. Рисунок 10.9), значению 1 – выбор трёх младших разрядов и т. д.

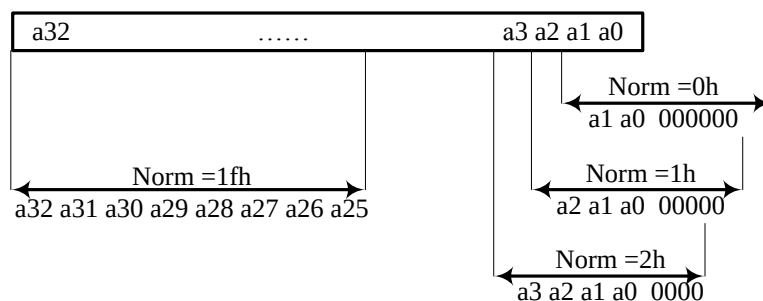


Рисунок 10.9 – Выбор разрядов при нормализации в зависимости от поля norm.

В процессе нормализации предусмотрены различные режимы округления результата (поле rnd команды WR_norm), также можно включить насыщение (полем st команды WR_norm).

Настройка насыщения играет роль в случаях переполнения – ситуации, когда хотя бы один из старших отбрасываемых разрядов не равен старшему (знаковому) из выбранных восьми. Если включен режим насыщения, то при переполнении на выходе блока нормализации будет максимальное положительное (+127) или максимальное отрицательное (-128) число.

В блоке нормализации ведется подсчет числа переполнений в текущем временном интервале для каждой составляющей комплексного сигнала (поля ov_ge и ov_im команды RD_norm).

Блок нормализации подсчитывает порядок данных, проходящих через блок. Значение порядка – позиция максимального значимого (не знакового) разряда в каждой из составляющих данного, порядок комплексного данного – максимальное значение порядка его составляющих (действительной и мнимой). Максимальное значение порядка из всех данных, прошедших за интервал, можно прочесть в поле exr команды RD_norm.

Если все разряды, с 27 по 1-й, всех данных в кадре знаковые (то есть и действительные и мнимые составляющие всех данных из набора {1, 0 -1, -2}), то exr=0; если разряды с 27 по 2-й знаковые, а 1-й разряд значимый, то exr=1, и т.д. Т.е формируемое значение exr соответствует «рекомендуемому» значению, которое сле-

					ЮФКВ.431268.006РЭ				Лист
									251
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		Решет 16.01.14		22755-1					

довало бы установить в norm для получения наибольшей точности результатов без переполнений. После окончания текущего временного интервала exp и norm доступны для чтения командой RD_norm.

Возможен режим автоматической настройки нормализатора. Вычисленное значение exp по приходу метки само помещается в поле norm. Т.е. аппаратура сама может правильно настроить нормализацию, исключив программный анализ принимаемых сигналов. В этом режиме значение exp передаётся в поле norm с дополнительным 2-разрядным смещением. Смещение задается в поле delta команды WR_norm. Кодировка смещений следующая: 00 – без смещения (norm = exp), 01 – смещение на +1 (norm = exp + 1), 10 – смещение на -2 (norm = exp – 2), 11 – смещение на -1 (norm = exp – 1). Т.е, при отрицательных delta достигается большая точность, но случаются насыщения, а при положительном – имеется запас от границы области насыщения.

Возможны два варианта включения автоматической настройки нормализатора. Командой WR_norm можно задать бит R (Repeat) – нормализатор будет подстраиваться после каждого прихода метки. Если R = 0, то автоподстройка соответствующего нормализатора будет произведена только один раз после запуска канала одной из команд WR_Start_exp_h, WR_Start_exp_f, WR_Start_exp_q.

Результату нормализации присваивается свертка по модулю 3.

10.7 Кластер КИХ-фильтров (FIR FILTER)

Кластер фильтров имеет 4 входные шины данных по 20 разрядов: 8 разрядов действительной части данного, 8 – мнимой, 2 разряда контрольной свёртки по модулю 3 на действительную часть и 2 – на мнимую.

Кластер содержит 4 одинаковые секции A,B,C,D КИХ-фильтров 64-го порядка (см. Рисунок 10.10). Каждая секция содержит банк памяти данных, банк памяти коэффициентов и вычислительные устройства. Банк данных организован в виде FIFO глубиной 16 строк и шириной 80 разрядов (4 комплексных данных).

Комплексные данные (d0, d1, ..) поступают на вход секции последовательно друг за другом, в соответствии с этим порядком секция выдаёт выходной поток (s0, s1, ..) частичных сумм:

$$\begin{aligned} s0 &= d0*w0 + d1*w1 + d2*w2 + d3*w3, \\ s1 &= d4*w4 + d5*w5 + d6*w6 + d7*w7, \\ &\dots \\ s15 &= d60*w60 + d61*w61 + d62*w62 + d63*w63. \end{aligned}$$

4 вычислительных устройства секции вычисляют каждую из таких сумм за 1 такт и передают их в аккумулятор ACC, в котором за 16 тактов производится окончательное суммирование:

$$S0 = s0 + s1 + \dots + s15 = d0*w0 + d1*w1 + \dots + d63*w63.$$

Данное d0, поступившее первым, удаляется из банка, с входа секции приходит следующее d64. Над текущими данными (d1, d2, .. d64) производится аналогичное действие:

$$S1 = d1*w0 + d2*w1 + \dots + d64*w63.$$

Поток входных комплексных данных (d0, d1, ..) преобразуется в поток выходных комплексных данных (S0, S1, ..) с помощью коэффициентов (w0, w1, .. w63). Таким образом одна секция и аккумулятор реализуют КИХ-фильтр 64-го порядка с 8-разрядными коэффициентами.

					ЮФКВ.431268.006РЭ			Лист
								252
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редкофф</i> 16.01.14		22755-1					

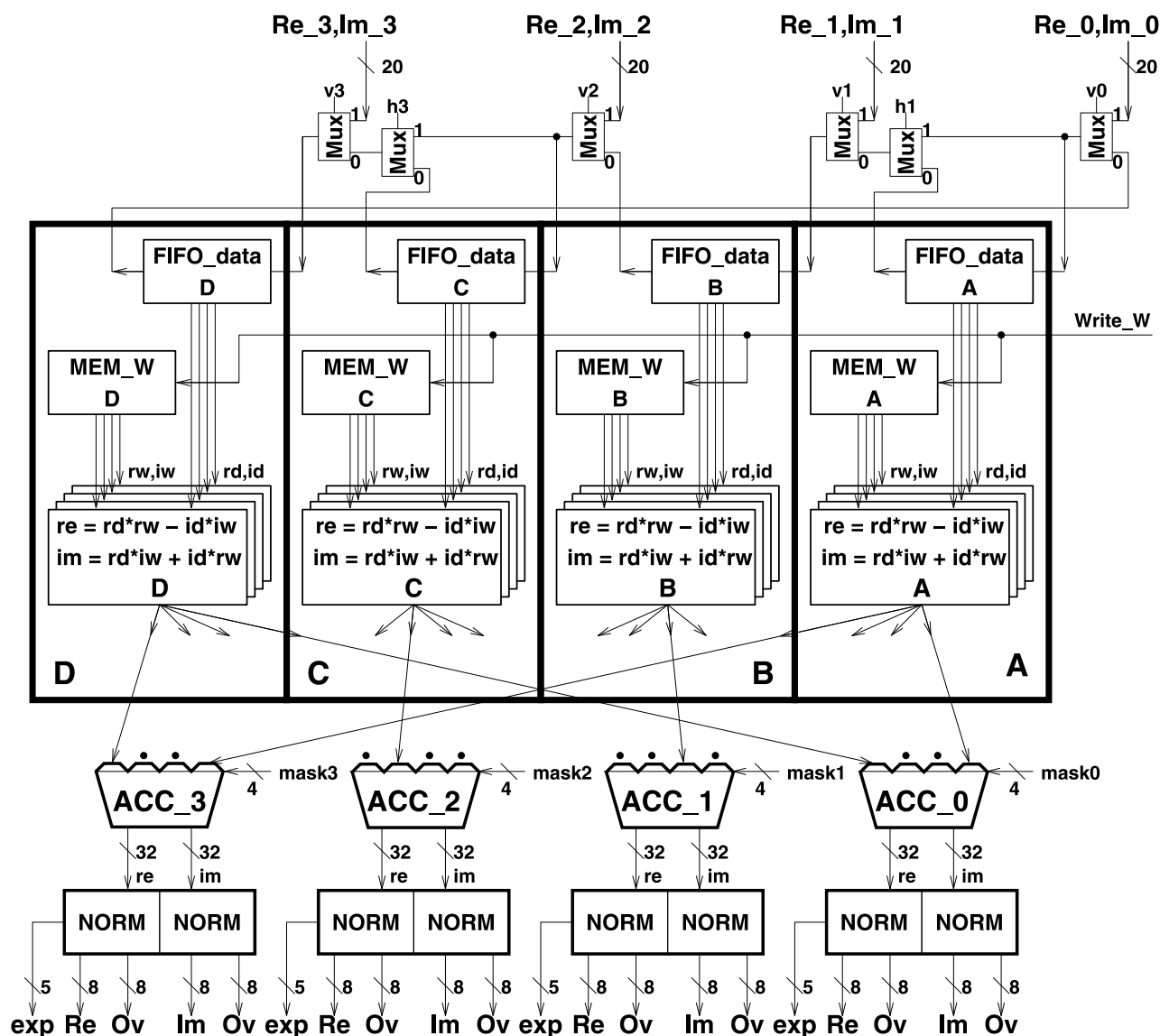


Рисунок 10.10 – Структурная схема кластера КИХ-фильтров

Максимальная скорость выходного (а значит и входного) потока данных составляет $F_{clk}/16$. **При настройке каналов, проходящих через фильтры, необходимо учитывать, что максимальная скорость потока данных на входе секции фильтра в 4 раза меньше скорости потока на входе в канал.** Т.е. необходимо чтобы было настроено прореживание в канале, который использует фильтр, в 4 или более раз.

Секции внутри кластера можно коммутировать между собой, получая фильтры большего порядка или с большей разрядностью коэффициентов. Кластер КИХ-фильтров можно программно конфигурировать:

- в один фильтр 256-го порядка с 8-разрядными коэффициентами либо в один фильтр 128-го порядка с 16-разрядными коэффициентами,
- в два фильтра 128-го порядка с 8-разрядными коэффициентами либо в два фильтра 64-го порядка с 16-разрядными коэффициентами,
- в четыре фильтра 64-го порядка с 8-разрядными коэффициентами.
- Возможны и другие комбинации конфигураций.

Мультиплексоры входных данных соединяют FIFO секций A, B, C, D в более длинные цепочки. Сигналы v0, v1, v2, v3 управляют подключением входных данных из

					ЮФКВ.431268.006РЭ			Лист
								253
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		16.01.14		22755-1				

своего канала к своей секции фильтра (см. Рисунок 10.10). Сигнал $v0 = 1$ подключает секцию А к входу Re_0 , Im_0 , $v1 = 1$ – секцию В к Re_1 , Im_1 , и т.д.

Сигналы $mask0$, $mask1$, $mask2$, $mask3$ определяют, из каких секций данный аккумулятор будет забирать частичные суммы.

Чтобы получить фильтр 128-го порядка из секций А и В, надо задать $v0 = 1$ (секция А будет получать данные из 0-го канала кластера), $v1 = 0$ (секция В будет получать данные из секции А), $h1 = 0$ (секция В будет получать данные, проходящие через FIFO секции А). Также следует задать $mask0 = 0011b$ – это позволит аккумулятору АСС_0 суммировать синхронные потоки частичных сумм, приходящие из секций А и В. Получение выходных данных производится с такой же максимальной скоростью (20,48 МГц), что и для фильтра 64-го порядка, за счёт дополнительной операции сложения на входе в аккумулятор.

Сигналы $h1$ и $h3$ переводят работу секций В и D в режим старших разрядов (эта возможность используется при работе с 16-разрядными коэффициентами) – в этом режиме на секции, например, А и В будут поступать одинаковые входные данные из 0-го канала. При переводе секций В и D в режим старших разрядов сигналы $v1$ и $v3$, соответственно, следует установить в 0. Чтобы получить из секций А и В фильтр 64-го порядка с 16-разрядными коэффициентами, надо задать $v0 = 1$ (данные будут поступать из 0-го канала кластера), $v1 = 0$, $h1 = 1$ (секция В будет также получать данные из 0-го канала кластера). И в этом случае тоже следует установить $mask0 = 0011b$. Теперь обе секции работают над идентичными потоками данных (данные дублируются в FIFO), но на выходе секции В результаты сдвинуты на 8 разрядов влево (сдвиг включается сигналом $h1$). Таким образом, коэффициенты секции В выполняют роль старших 8 разрядов 16-разрядных коэффициентов получившегося фильтра.

Если $v0=v1=v2=v3=0$ (замкнутое кольцо), то происходит останов фильтра – на фильтр не поступают входные данные.

Сигналы $h1$, $h3$, $v3$, $v2$, $v1$, $v0$, $mask0$, $mask1$, $mask2$, $mask3$ можно установить в нужное состояние командой WR_filter_mask .

Запись коэффициентов в память MEM_W производится командой $WR_w_filter_RAM$, при этом фильтр необходимо остановить.

После накопления результаты нормализуются в блоке NORM (описание блока нормализации см. п. 10.6).

10.8 Блок квадраторов (SQR)

Структурная схема блока квадраторов приведена на рисунке ниже (см. Рисунок 10.11).

					ЮФКВ.431268.006РЭ				Лист
									254
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		<i>Редель</i> 16.01.14		22755-1					

10.9 Блоки упаковки (PACK_UNIT и ADC_PACK_UNIT)

Блоки упаковки преобразуют данные, приходящие из каналов паковки, в удобный для записи в память формат – 64-разрядные слова. Как показано на структурной схеме DSPPU (см. Рисунок 10.1), имеются 9 блоков PACK_UNIT упаковки комплексных данных, которые поступают из каналов обработки, и один блок ADC_PACK_UNIT упаковки действительных данных, получаемых непосредственно с интерфейсов входного устройства DSPPU. Комплексные данные состоят из 8 разрядов действительной части и 8 разрядов мнимой, действительные данные – 12-разрядные. Непрерывные потоки данных разделяются временными метками на кадры, первое данное кадра упаковывается всегда в младшую часть 64-разрядного слова. Если количество данных в кадре таково, что их нельзя упаковать в целое количество 64-разрядных слов, то в последнем одном или двух словах имеются не заполненные данными разряды.

Действительные данные всегда преобразуются в 16-разрядные расширением знака и упаковываются по 4 в слово. Пример упаковки действительных данных показан на рисунке ниже (см. Рисунок 10.12).

Adr

6362616059585756555453525150494847464544434241403938373635343332313029282726252423222120191817161514131211109876543210

(0)

Het=00h/Size

Size=00002

Pk=1C

Channel

Int

Millisecond Number

(1)

S3

S2

S1

S0

(2)

S7

S6

S5

S4

(3)

S11

S10

S9

S8

(4)

S14

S13

S12

Рисунок 10.12 – Упаковка в память действительных данных

Действительные и мнимые составляющие комплексного данного могут упаковываться как в одно слово (см. Рисунок 10.13, Рисунок 10.15, Рисунок 10.17), так и в два соседних слова (см. Рисунок 10.14, Рисунок 10.16, Рисунок 10.18). Разрядности составляющих могут быть 2, 4 и 8 (см. Рисунок 10.17 и Рисунок 10.18, Рисунок 10.15 и Рисунок 10.16, Рисунок 10.13 и Рисунок 10.14 соответственно). На рисунках ниже показаны все возможные варианты упаковки комплексных данных в памяти. Мнимая часть первого данного кадра показана как $j0$, действительная часть первого данного – как $l0$, мнимая часть следующего данного – $j1$, действительная – $l1$ и так далее.

63 61 59 57 55 53 51 49 47 45 43 41 39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 9 7 5 3 1
62 60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2 0

Adr
(0)

Het/Size	Size=00002	Pk=1C	Channel	Int	Millisecond Number
----------	------------	-------	---------	-----	--------------------

(1)

I3	j3	I2	j2	I1	j1	I0	j0
----	----	----	----	----	----	----	----

(2)

I7	j7	I6	j6	I5	j5	I4	j4
----	----	----	----	----	----	----	----

(3)

I11	j11	I10	j10	I9	j9	I8	j8
-----	-----	-----	-----	----	----	----	----

(4)

		I14	j14	I13	j13	I12	j12
--	--	-----	-----	-----	-----	-----	-----

Рисунок 10.13 – Упаковка действительной и мнимой части в одно 64-разрядное слово, 8+8 разрядов

					ЮФКВ.431268.006РЭ	Лист 256	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

Adr (0)	63 61 59 57 55 53 51 49 47 45 43 41 39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 9 7 5 3 1 62 60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2 0																																			
	Het/Size				Size=00002								Pk=14				Channel		Int		Millisecond Number															
	(1)	j7				j6				j5				j4				j3				j2				j1				j0						
	(2)	I7				I6				I5				I4				I3				I2				I1				I0						
(3)													j12				j11				j10				j9				j8							
(4)													I12				I11				I10				I9				I8							

Рисунок 10.14 – Упаковка действительной и мнимой части в разные 64-разрядные слова, 8+8 разрядов

Adr (0)	63 61 59 57 55 53 51 49 47 45 43 41 39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 9 7 5 3 1 62 60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2 0																																
	Het/Size		Size=00002						Pk=00		Channel		Int	Millisecond Number																			
	(1)	I7	j7	I6	j6	I5	j5	I4	j4	I3	j3	I2	j2	I1	j1	I0	j0																
	(2)	I15	j15	I14	j14	I13	j13	I12	j12	I11	j11	I10	j10	I9	j9	I8	j8																
(3)	I23	j23	I22	j22	I21	j21	I20	j20	I19	j19	I18	j18	I17	j17	I16	j16																	
(4)	I31	j31	I30	j30	I29	j29	I28	j28	I27	j27	I26	j26	I25	j25	I24	j24																	

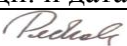
Рисунок 10.15 – Упаковка действительной и мнимой части в одно 64-разрядное слово, 4+4 разряда

Adr	63	61	59	57	55	53	51	49	47	45	43	41	39	37	35	33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
	62	60	58	56	54	52	50	48	46	44	42	40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
(0)	Het/Size		Size=00002											Pk=02		Channel		Int		Millisecond Number												
(1)	j15	j14	j13	j12	j11	j10	j9	j8	j7	j6	j5	j4	j3	j2	j1	j0																
(2)	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0																
(3)																																j16
(4)																																I16

Рисунок 10.16 – Упаковка действительной и мнимой части в разные 64-разрядные слова, 4+4 разряда

Adr	63	61	59	57	55	53	51	49	47	45	43	41	39	37	35	33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
	62	60	58	56	54	52	50	48	46	44	42	40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
(0)	Het/Size				Size=00002								Pk=1E		Channel		Int		Millisecond Number													
(1)	I15	j15	I14	j14	I13	j13	I12	j12	I11	j11	I10	j10	I9	j9	I8	j8	I7	j7	I6	j6	I5	j5	I4	j4	I3	j3	I2	j2	I1	j1	I0	j0
(2)	I31	j31	I30	j30	I29	j29	I28	j28	I27	j27	I26	j26	I25	j25	I24	j24	I23	j23	I22	j22	I21	j21	I20	j20	I19	j19	I18	j18	I17	j17	I16	j16
(3)	I47	j47	I46	j46	I45	j45	I44	j44	I43	j43	I42	j42	I41	j41	I40	j40	I39	j39	I38	j38	I37	j37	I36	j36	I35	j35	I34	j34	I33	j33	I32	j32
(4)					I61	j61	I60	j60	I59	j59	I58	j58	I57	j57	I56	j56	I55	j55	I54	j54	I53	j53	I52	j52	I51	j51	I50	j50	I49	j49	I48	j48

Рисунок 10.17 – Упаковка действительной и мнимой части в одно 64-разрядное слово, 2+2 разряда

					ЮФКВ.431268.006РЭ					Лист
										257
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.		Подп. и дата			Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14			22755-1					

Adr	63	61	59	57	55	53	51	49	47	45	43	41	39	37	35	33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
	62	60	58	56	54	52	50	48	46	44	42	40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0
(0)	Het/Size		Size=00002										Pk=1F		Channel	Int	Millisecond Number															
(1)	j31	j30	j29	j28	j27	j26	j25	j24	j23	j22	j21	j20	j19	j18	j17	j16	j15	j14	j13	j12	j11	j10	j9	j8	j7	j6	j5	j4	j3	j2	j1	j0
(2)	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
(3)		j62	j61	j60	j59	j58	j57	j56	j55	j54	j53	j52	j51	j50	j49	j48	j47	j46	j45	j44	j43	j42	j41	j40	j39	j38	j37	j36	j35	j34	j33	j32
(4)		I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Рисунок 10.18 – Упаковка действительной и мнимой части в разные 64-разрядные слова, 2+2 разряда

Тип упаковки задаётся командами WR_pack и WR_ADC_control.

Показанное на рисунках нулевое слово является заголовком кадра и всегда сопровождает данные, записываемые блоком DSPPU в память. Формат заголовка кадра описан ниже (см. п. 10.10).

10.10 Блок адресных генераторов (AG)

Блок адресных генераторов служит для записи в память данных от устройств упаковки, а также содержит монитор работы каналов паковки.

Под каждый канал паковки выделяются две области памяти: чётная и нечётная. Начальные адреса областей, в которые блок AG будет записывать данные, задаются командой WR_AG_adr_RAM. Выбор одной из двух областей (чётной или нечётной) осуществляется блоком в зависимости от текущего номера временного интервала. Узнать, в какую область производится запись в данный момент, можно с помощью команды RD_Ms_counter.

Формирование одного кадра в памяти происходит следующим образом:

- после запуска канала данные приходят к блоку AG через соответствующий канал паковки,
- первое пришедшее 64-разрядное слово записывается по адресу Memory_address+2 (Memory_address – значение поля соответствующей команды WR_AG_adr_RAM, 2 добавляется для того, чтобы оставить место для заголовка кадра),
- остальные данные записываются за первым по инкрементирующимся адресам,
- приходит временная метка,
- последние два 64-разрядных слова (обычно не полностью заполненные данными) записываются вслед за остальными,
- по адресу Memory_address записывается заголовок кадра – кадр в памяти полностью сформирован.

Заголовок кадра содержит характеристики данных в кадре. Формат заголовка кадра показан на рисунке ниже (см. Рисунок 10.19).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
PackChannel								Int		Ms																													
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32								
Het/size								size																												Pack			

Рисунок 10.19 – Формат заголовка кадра

					ЮФКВ.431268.006РЭ	Лист	
						258	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

Таблица 10.2 – Описание полей заголовка кадра

Биты	Название	Выполняемая функция
[63:56]	Het/size	Если командой WR_AG_control_monitor установлен признак H, то в данном поле отображаются старшие разряды размера кадра (продолжение поля size), если нет – то старшие восемь разрядов регистра фазы гетеродина на момент прихода метки.
[55:37]	size	Размер кадра – количество пар 64-разрядных слов (сам заголовок не учитывается).
[36:32]	Pack	Количество 4-битных фрагментов в последней паре слов (ноль означает, что все 128 бит содержат данные).
[31:26]	PackChannel	Номер канала паковки, который записал данный кадр (может принимать значения от 0 до 39).
[25:24]	Int	Показывает, возникали ли при обработке данного кадра исключительные ситуации: бит 25 показывает, было ли нарушение контроля по модулю 3 (подробности: команда RD_Channel_check, биты h,hri,fri,qri и RD_ADC_check – бит p2), бит 24 показывает, возникали ли ситуации неправильного программирования или потери данных (подробности: команда RD_Channel_check, биты sy,pk,pkq,fv и RD_ADC_check – биты sz,adc_pk_ov).
[23:0]	Ms	Показывает номер метки от интервального таймера.

Все события записи заголовков кадров фиксируются в специальном блоке монитора. Монитор организован по стековому принципу. Информация из заголовка помещается в верхнюю позицию стека, и размер стека увеличивается. При чтении командой RD_stack_monitor верхняя позиция считывается и удаляется, размер стека уменьшается. Команда RD_ms_part_monitor читает верхнюю позицию в стеке, однако при этом прочитанная строка из стека не удаляется.

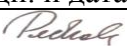
Командой WR_AG_control_monitor осуществляется управление блоком AG. С её помощью можно включить прерывания по некоторым событиям: по достижению определённого количества позиций в стеке, по переполнению стека (превышение 47 занятых позиций в стеке), по завершению работы конкретным каналом, по достижению заданной доли интервала. При переполнении самые старые позиции стека теряются.

При использовании стека для генерации прерываний рекомендуется в процедуре обработки прерывания прочитывать весь стек командой RD_stack_monitor.

10.11 Прерывания

DSPPU может выдавать 3 вида запросов на прерывание.

- INT_DSPPU_CH – прерывание от блока адресных генераторов DSPPU. Сюда входят прерывания по завершению работы каналов.
- INT_DSPPU_PRG – прерывание по обнаружению ошибок программирования или при потере данных. Данное прерывание возникает, когда хотя бы один из битов sy, pk, pkq, fv, которые могут быть прочитаны командой RD_Channel_check, или из битов sz, adc_pk_ov, которые могут быть прочитаны командой RD_ADC_check, принимает значение 1. Причиной данного прерывания может стать неправильный запуск или останов канала, а также потеря данных в случаях, когда суммарный поток данных всех работающих каналов слишком велик.
- INT_DSPPU_HRD – прерывание по обнаружению ошибки контрольной суммы по модулю 3 в вычислительных блоках DSPPU – биты h, hri, fri, qri команды RD_Channel_check и бит p2 команды RD_ADC_check.

					ЮФКВ.431268.006РЭ			Лист
								259
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

11 Электрические и конструктивные характеристики СБИС СПОВС

11.1 Состав и расположение внешних выводов СБИС СПОВС.

СБИС СПОВС изготавливается в 484-выводном пластиковом корпусе ТЕРВГА. Расположение внешних выводов микросхемы представлено ниже (см. Рисунок 11.1).

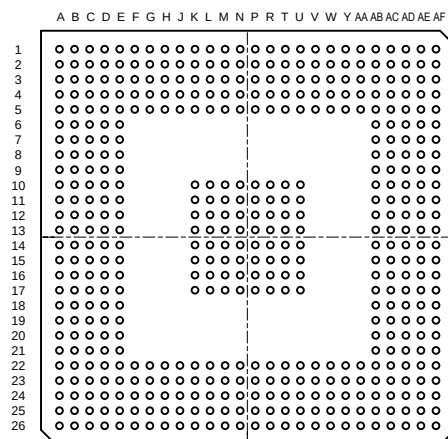


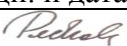
Рисунок 11.1 - Расположение внешних выводов СБИС СПОВС

В таблице ниже (Таблица 11.1) представлены выводы микросхемы, отсортированные по их функциональному назначению.

					ЮФКВ.431268.006РЭ				Лист
									260
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редко</i> 16.01.14		22755-1						

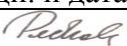
Таблица 11.1 – Выводы микросхемы, отсортированные по функциональному назначению

Обозначение ¹⁾	Кол-во	Тип ²⁾	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
Интерфейс с внешней шиной					
D31...D0	32	I/O	8,1	Шина данных с SDRAM памятью типа DDR1	D3, D2, D1, E5, F5, F4, F3, F1, G5, G3, G2, H5, H1, J5, J4, J3, V3, V4, V5, W5, Y2, Y3, Y5, AA1, AA2, AA3, AA4, AA5, AB5, AC1, AC3, AD2
A13...A0	14	O(Z)	8,1	Шина адреса	L4, L5, M4, M2, P1, R2, R4, T1, T2, T4, U1, U3, U5, V2
BA1, BA0	2	O(Z)	8,1	Выбор банка памяти внутри микросхемы SDRAM	K5, K2
SCLK1, XCLK1	2	O	8,1	Выход тактового сигнала внешней шины	N5, N3
SCLK2, XCLK2	2	O	8,1	Выход тактового сигнала внешней шины	P5, P3
SCKE	1	O(Z)	8,1	Сигнал управления для перевода микросхем памяти в режим пониженного энергопотребления	N2
XCS	1	O(Z)	8,1	Выбор банка внешней памяти	J2
XWE	1	O(Z)	8,1	Разрешение записи в синхронную память	L3
XRAS	1	O(Z)	8,1	Строб адреса строки SDRAM	K1
XCAS	1	O(Z)	8,1	Строб адреса столбца SDRAM	L2
DM3, ... DM0	4	I/O	8,1	Сигналы маскирования данных при записи	E4, H4, W4, AB2
DQS3, ... DQS0	4	I/O	8,1	Сигналы стробов данных	E2, H2, W2, AB4
VREF3, ... VREF0	4	AI	8,1	Вход напряжения смещения	E3, H3, W3, AB3
SPI интерфейс					
SPICLK	1	O	8	Выход тактового сигнала	AB11
SPITXD	1	O	8	Выход передаваемых данных	AE10
SPIRXD	1	I,U		Вход принимаемых данных	AF10
SPI_CS7, ..., SPI_CS0	8	O	4	Выбор микросхемы/устройства SPI	AE9, AD9, AC9, AB9, AF8, AE8, AD8, AB8
Порты общего назначения					
GPIO15, ..., GPIO0	16	I/O, Sh	4	Программируемы входы/выходы	AF7, AE7, AD7, AC7, AB7, AF6, AE6, AC6, AB6, AE5, AD5, AF4, AE4, AD4, AF3, AE3
UART порт 0					
UART0TXD	1	O	4	Выход передаваемых данных	AE11
UART0RXD	1	I, Sh		Вход принимаемых данных	AF12
XCTS0	1	I, Sh		Разрешение передачи данных	AD11
XRTS0	1	O	4	Запрос на передачу данных	AD12
UART порт 1					
UART1TXD	1	O	4	Выход передаваемых данных	AF14
UART1RXD	1	I, Sh		Вход принимаемых данных	AE13
XCTS1	1	I, Sh		Разрешение передачи данных	AD14
XRTS1	1	O	4	Запрос на передачу данных	AD13
USB2.0 (full speed) интерфейс					

					ЮФКВ.431268.006РЭ			Лист
								261
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

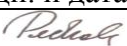
Продолжение таблицы 11.1

Обозначение ¹⁾	Кол-во	Тип ²⁾	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
USBVP	1	I/O		Линия D+ шины USB	AB13
USBVM	1	I/O		Линия D- шины USB	AB15
USBCLKA	1	I		Вход А для подключения тактового сигнала 48 МГц	AB17
USBCLKB	1	I		Вход В для подключения тактового сигнала 48 МГц	AD15
USBCON	1	I, D		Индикатор подключения устройства USB host	AE15
TD9 ³⁾	1	O	6	Сигнал управления подключением к устройству USB host	W24
Выводы счетчика реального времени					
RTCCLKA	1	I		Вход А для подключения часового кварца 32768 Гц	AE19
RTCCLKB	1	I		Вход В для подключения часового кварца 32768 Гц	AC19
RTCINT	1	O	4	Выход прерывания по будильнику	AC21
Выводы блока интервальных таймеров					
ITSTROB	1	I, Sh		Вход строга записи значения интервального таймера	M22
IT_1SO	1	O	4	Выход программируемого генератора импульсов	K22
ITO	1	O	4	Выход сигнала дополнительного временного интервала	K23
Выводы от блока предварительной обработки.					
TO3, ..., TO0	4	O	4	Выходы генераторов псевдослучайных последовательностей	E23, G23, H23, J23
TI3, ..., TI0	4	I		Входы генераторов псевдослучайных последовательностей	F22, F23, H24, J24
SIGN3, ..., SIGN0	4	I		Цифровые входы двух битовых портов (знаковый разряд)	A5, B5, C6, D7
MAGN3, ..., MAGN0	4	I		Цифровые входы двух битовых портов (значимый разряд)	A4, B4, C5, D6
JTAG-интерфейс					
TDO	1	O(Z)	4	Выход данных тестового порта	Y24
TDI	1	I		Вход данных тестового порта	AA24
TCK	1	I		Тактовый сигнал тестового порта	Y22
TMS	1	I		Сигнал выбора режима тестового порта	AA26
XTBST	1	I,U		Сброс тестового порта	Y25
Отладочный интерфейс ядра ARM					
RTCK	1	O	4	Выход синхросигнала тестового порта	AB25
XSRST	1	I/O, U	4	Программный сброс от отладчика	AA23
Тестовый порт NMC					
TD7, ..., TD4	4	O	6	Выход тестового порта ядра NMC3 процессорной системы NMU2	W26, V23, V25, V26
TD3, ..., TD0	4	O	6	Выход тестового порта ядра NMC3 процессорной системы NMU1	U23, U24, T23, T25
Общее управление					
XRESET	1	I, Sh		Системный сброс.	N24
XINT3, ..., XINT0	4	I, Sh		Входы внешних прерываний	AE17, AD17, AD16, AE16

					ЮФКВ.431268.006РЭ				Лист
									262
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата			
22755-2		 16.01.14		22755-1					

Продолжение таблицы 11.1

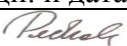
Обозначение ¹⁾	Кол-во	Тип ²⁾	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
WDT	1	O	4	Выход сторожевого таймера	AB19
BOOTM2, ..., BOOTM1 ⁴⁾	2	I		Сигналы управления выбора устройства начальной загрузкой	AD25, AE25
VINITH	1	I		Сигнал управления, определяющий начальный адрес загрузки	AC26
Тестовые выводы					
TD8	1	O	6	Тестовый вывод для тестирования на фабрике	W25
TST_I2, ..., TST_I0 ⁵⁾	3	I		Тестовые входы для тестирования. Должны быть подключены к аналоговому нулю	AD24, AE23, AC23
TSTMODE	1	I		Тестовый вход для тестирования СБИС на фабрике. Должен быть подключен к логическому нулю.	AC24
TSTCLK2, TSTCLK1	2	I		Тестовый вход для тестирования СБИС на фабрике. Должен быть подключен к логическому нулю	P22, AE21
VPD	1	I		Отключение pullup/pulldown резисторов для тестирования. Должен быть подключен к логическому нулю.	D24
Выводы управления PLL					
PLLBYPASS	1	I		Вход отключения PLL. Должен быть подключен к логическому нулю.	AD22
STOP	1	O	4	Сигнал отключения внешнего тактового сигнала	AC22
PLLMACRO_L	1	O	4	Завершение установки тактового сигнала на PLL	AF22
FBCLK_PLL	1	O	6	Тестовый выход PLL	AB22
Выводы блока АЦП					
INP_CH0, INM_CH0	2	AI		Дифференциальные входы АЦП0	C21, C22
INP_CH1, INM_CH1	2	AI		Дифференциальные входы АЦП1	C18, C17
INP_CH2, INM_CH2	2	AI		Дифференциальные входы АЦП2	C13, C14
INP_CH3, INM_CH3	2	AI		Дифференциальные входы АЦП3	C10, C9
CLKINX1/CLKINX0	2	AI		Вход тактового сигнала	A16, A15
OSC_MOD33	1	I		Управление режимом формирования тактового сигнала	B17
OSC_REXT33	1	I		Управление подключением внутреннего резистора	B15
GLIT-FILT_BYPASS	1	I		Управление фильтром пульсаций	B16
REF_CH3, ..., REF_CH0	4	АО		Среднее опорное напряжение АЦП 1,4В	A9, A11, A20, A22
REFLO_CH3, ..., REFLO_CH0	4	АО		Нижнее опорное напряжение АЦП 1,0В	A8, A12, A19, A23

					ЮФКВ.431268.006РЭ				Лист
									263
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

REFHI_CH3, ..., REFHI_CH0	4	АО		Верхнее опорное напряжение АЦП 1,8В	А7, А13, А18, А24	
					ЮФКВ.431268.006РЭ	
Изм.	Лист	№ докум.	Подп.	Дата		
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата
22755-2	<i>Редко</i> 16.01.14		22755-1			

Продолжение таблицы 11.1

Обозначение ¹⁾	Кол-во	Тип ²⁾	Вых. Ток, мА	Функциональное назначение	Выводы корпуса
BGAP_A	1	АО		Выход генератора опорного напряжения пары А 1,4В	A21
BGAP_B	1	АО		Выход генератора опорного напряжения пары В 1,4В	A10
Питание					
VDDI	48	S		Напряжение питания ядра +1,2В	AA22, AB10, AB12, AB14, AB18, AB24, AC20, AD18, AD19, AD20, AE22, AE24, AF20, E22, E7, E8, G22, J22, K10, K11, L10, L13, L14, L17, L22, N10, N11, N16, N17, N22, P10, P11, P16, P17, R22, T10, T13, T14, T17, T22, U10, U11, U13, U14, U16, U17, V22, W22
VDE1	45	S		Напряжение питания буферов ввода/вывода +3,3В	A26, A3, AB26, AC10, AC12, AC16, AC18, AC5, AC8, AD23, AD26, AF11, AF13, AF15, AF17, AF2, AF21, AF24, AF26, AF5, AF9, B2, B25, C24, C26, C7, D25, E24, E9, F26, G24, H25, K24, K26, L25, M24, M26, N23, N25, P24, R23, R26, U26, V24, Y26
VDE2	21	S		Напряжение питания буферов SSTL +2,5В	A1, AB1, AC2, AD3, AE1, C1, D4, E1, G1, G4, J1, K4, L1, M3, N1, P4, R1, T3, T5, V1, Y1
VDE3	1	S		Напряжение питания USB буфера +3,3В	AC14
AVD33	15	S		Напряжение питания аналогового блока АЦП 3,3В	D11, D12, D15, D16, D19, D20, E10, E13, E14, E17, E18, E21, K13, K15, K17
AVD12	1	S		Напряжение питания для PLL +1,2В	AB20
VSS	113	S		Цифровая земля	A2, AA25, AB16, AB23, AC11, AC13, AC15, AC17, AC25, AC4, AD1, AD10, AD21, AD6, AE12, AE14, AE18, AE2, AE20, AE26, AF1, AF16, AF18, AF19, AF23, AF25, B1, B26, B3, B6, C2, C25, C3, C4, D23, D26, D5, D8, E25, E26, E6, F2, F24, F25, G25, G26, H22, H26, J25, J26, K25, K3, L11, L12, L15, L16, L23, L24, L26, M1, M10, M11, M12, M13, M14, M15, M16, M17, M23, M25, M5, N12, N13, N14, N15, N26, N4, P12, P13, P14, P15, P2, P23, P25, P26, R10, R11, R12, R13, R14, R15, R16, R17, R24, R25, R3, R5, T11, T12, T15, T16, T24, T26, U12, U15, U2, U22, U25, U4, W1, W23, Y23, Y4
AVS	44	S		Аналоговая земля	A14, A17, A25, A6, B10, B11, B12, B13, B14, B18, B19, B20, B21, B22, B23, B24, B7, B8, B9, C11, C12, C15, C16, C19, C20, C23, C8, D10, D13, D14, D17, D18, D21, D22, D9, E11, E12, E15, E16, E19, E20, K12, K14, K16
AVS12	1	S		Земля для PLL	AB21

					ЮФКВ.431268.006РЭ				Лист
									265
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

Примечания:

- 1) Для выводов со знаком инверсии активным является низкий уровень сигнала.
- 2) Используемые обозначения типов выводов:
 - I – вход,
 - O – выход,
 - O(Z) – выход с высокоимпедансным состоянием,
 - I/O – двунаправленный вывод,
 - S –питание.
 - AI – аналоговый вход
 - AO – аналоговый выход
 - Sh – буфер с триггером Шмидта
 - U – на выводе стоит pull-up резистор
 - D – на выводе стоит pull-down резистор
- 3) TD9 =1 – подключения к USB Host-у не происходит,
TD9 =0 – есть подключение к USB Host-у.
- 4) Кодировка выводов BOOTM2, BOOTM1:
 - 00 – загрузка с SPI;
 - 01 – загрузка с UART0;
 - 10 – загрузка с UART1;
 - 11 – загрузка с USB;
5. Входы TST_I0, TST_I1 (см. 5.11.1) можно использовать как конфигурационные.
Вход TST_I2=1 показывает индикацию загрузки на выводах GPIO (см. 5.17.8). Если TST_I2=0, то загрузчик конфигурирует все разряды GPIO как входы и индикации состояния начального загрузчика не происходит.

. В таблице ниже (Таблица 11.2) представлены выводы микросхемы, отсортированные по их номерам.

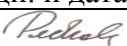
					ЮФКВ.431268.006РЭ			Лист
								266
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

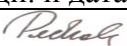
Таблица 11.2 – Выводы микросхемы, отсортированные по номеру вывода

Вывод	Сигнал	Вывод	Сигнал	Вывод	Сигнал
A1	VDE2	A2	VSS	A3	VDE1
A4	MAGN3	A5	SIGN3	A6	AVS
A7	REFHI_CH3	A8	REFLO_CH3	A9	REF_CH3
A10	BGAP_B	A11	REF_CH2	A12	REFLO_CH2
A13	REFHI_CH2	A14	AVS	A15	CLKINX0
A16	CLKINX1	A17	AVS	A18	REFHI_CH1
A19	REFLO_CH1	A20	REF_CH1	A21	BGAP_A
A22	REF_CH0	A23	REFLO_CH0	A24	REFHI_CH0
A25	AVS	A26	VDE1	B1	VSS
B2	VDE1	B3	VSS	B4	MAGN2
B5	SIGN2	B6	VSS	B7	AVS
B8	AVS	B9	AVS	B10	AVS
B11	AVS	B12	AVS	B13	AVS
B14	AVS	B15	OSC_REXT33	B16	GLIT-FILT_BYPASS
B17	OSC_MOD33	B18	AVS	B19	AVS
B20	AVS	B21	AVS	B22	AVS
B23	AVS	B24	AVS	B25	VDE1
B26	VSS	C1	VDE2	C2	VSS
C3	VSS	C4	VSS	C5	MAGN1
C6	SIGN1	C7	VDE1	C8	AVS
C9	INP_CH3	C10	INM_CH3	C11	AVS
C12	AVS	C13	INM_CH2	C14	INP_CH2
C15	AVS	C16	AVS	C17	INP_CH1
C18	INM_CH1	C19	AVS	C20	AVS
C21	INM_CH0	C22	INP_CH0	C23	AVS
C24	VDE1	C25	VSS	C26	VDE1
D1	D29	D2	D30	D3	D31
D4	VDE2	D5	VSS	D6	MAGN0
D7	SIGN0	D8	VSS	D9	AVS
D10	AVS	D11	AVD33	D12	AVD33
D13	AVS	D14	AVS	D15	AVD33
D16	AVD33	D17	AVS	D18	AVS
D19	AVD33	D20	AVD33	D21	AVS
D22	AVS	D23	VSS	D24	VPD
D25	VDE1	D26	VSS	E1	VDE2
E2	DQS3	E3	VREF3	E4	DM3
E5	D28	E6	VSS	E7	VDDI
E8	VDDI	E9	VDE1	E10	AVD33
E11	AVS	E12	AVS	E13	AVD33
E14	AVD33	E15	AVS	E16	AVS
E17	AVD33	E18	AVD33	E19	AVS
E20	AVS	E21	AVD33	E22	VDDI
E23	TO3	E24	VDE1	E25	VSS
E26	VSS	F1	D24	F2	VSS
F3	D25	F4	D26	F5	D27
F22	TI3	F23	TI2	F24	VSS
F25	VSS	F26	VDE1	G1	VDE2
G2	D21	G3	D22	G4	VDE2
G5	D23	G22	VDDI	G23	TO2
G24	VDE1	G25	VSS	G26	VSS
H1	D19	H2	DQS2	H3	VREF2
H4	DM2	H5	D20	H22	VSS
H23	TO1	H24	TI1	H25	VDE1
H26	VSS	J1	VDE2	J2	XCS

					ЮФКВ.431268.006РЭ			Лист
								267
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Решет</i> 16.01.14		22755-1				

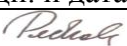
Продолжение таблицы 11.2

Вывод	Сигнал	Вывод	Сигнал	Вывод	Сигнал
J3	D16	J4	D17	J5	D18
J22	VDDI	J23	TO0	J24	TI0
J25	VSS	J26	VSS	K1	XRAS
K2	BA0	K3	VSS	K4	VDE2
K5	BA1	K10	VDDI	K11	VDDI
K12	AVS	K13	AVD33	K14	AVS
K15	AVD33	K16	AVS	K17	AVD33
K22	IT_ISO	K23	ITO	K24	VDE1
K25	VSS	K26	VDE1	L1	VDE2
L2	XCAS	L3	XWE	L4	A13
L5	A12	L10	VDDI	L11	VSS
L12	VSS	L13	VDDI	L14	VDDI
L15	VSS	L16	VSS	L17	VDDI
L22	VDDI	L23	VSS	L24	VSS
L25	VDE1	L26	VSS	M1	VSS
M2	A10	M3	VDE2	M4	A11
M5	VSS	M10	VSS	M11	VSS
M12	VSS	M13	VSS	M14	VSS
M15	VSS	M16	VSS	M17	VSS
M22	ITSTROB	M23	VSS	M24	VDE1
M25	VSS	M26	VDE1	N1	VDE2
N2	SCKE	N3	XSCLK1	N4	VSS
N5	SCLK1	N10	VDDI	N11	VDDI
N12	VSS	N13	VSS	N14	VSS
N15	VSS	N16	VDDI	N17	VDDI
N22	VDDI	N23	VDE1	N24	XRESET
N25	VDE1	N26	VSS	P1	A9
P2	VSS	P3	XSCLK2	P4	VDE2
P5	SCLK2	P10	VDDI	P11	VDDI
P12	VSS	P13	VSS	P14	VSS
P15	VSS	P16	VDDI	P17	VDDI
P22	TESTCLK2	P23	VSS	P24	VDE1
P25	VSS	P26	VSS	R1	VDE2
R2	A8	R3	VSS	R4	A7
R5	VSS	R10	VSS	R11	VSS
R12	VSS	R13	VSS	R14	VSS
R15	VSS	R16	VSS	R17	VSS
R22	VDDI	R23	VDE1	R24	VSS
R25	VSS	R26	VDE1	T1	A6
T2	A5	T3	VDE2	T4	A4
T5	VDE2	T10	VDDI	T11	VSS
T12	VSS	T13	VDDI	T14	VDDI
T15	VSS	T16	VSS	T17	VDDI
T22	VDDI	T23	TD1	T24	VSS
T25	TD0	T26	VSS	U1	A3
U2	VSS	U3	A2	U4	VSS
U5	A1	U10	VDDI	U11	VDDI
U12	VSS	U13	VDDI	U14	VDDI
U15	VSS	U16	VDDI	U17	VDDI
U22	VSS	U23	TD3	U24	TD2
U25	VSS	U26	VDE1	V1	VDE2
V2	A0	V3	D15	V4	D14
V5	D13	V22	VDDI	V23	TD6
V24	VDE1	V25	TD5	V26	TD4
W1	VSS	W2	DQS1	W3	VREF1

					ЮФКВ.431268.006РЭ				Лист
									268
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

Продолжение таблицы 11.2

Вывод	Сигнал	Вывод	Сигнал	Вывод	Сигнал
W4	DM1	W5	D12	W22	VDDI
W23	VSS	W24	TD9	W25	TD8
W26	TD7	Y1	VDE2	Y2	D11
Y3	D10	Y4	VSS	Y5	D9
Y22	TCK	Y23	VSS	Y24	TDO
Y25	XTRST	Y26	VDE1	AA1	D8
AA2	D7	AA3	D6	AA4	D5
AA5	D4	AA22	VDDI	AA23	XSRST
AA24	TDI	AA25	VSS	AA26	TMS
AB1	VDE2	AB2	DM0	AB3	VREF0
AB4	DQS0	AB5	D3	AB6	GPIO7
AB7	GPIO11	AB8	SPI_CS0	AB9	SPI_CS4
AB10	VDDI	AB11	SPICLK	AB12	VDDI
AB13	USBVP	AB14	VDDI	AB15	USBVM
AB16	VSS	AB17	USBCLK	AB18	VDDI
AB19	WDT	AB20	AVD12	AB21	AVS12
AB22	FBCLK_PLL	AB23	VSS	AB24	VDDI
AB25	RTCK	AB26	VDE1	AC1	D2
AC2	VDE2	AC3	D1	AC4	VSS
AC5	VDE1	AC6	GPIO8	AC7	GPIO12
AC8	VDE1	AC9	SPI_CS5	AC10	VDE1
AC11	VSS	AC12	VDE1	AC13	VSS
AC14	VDE3	AC15	VSS	AC16	VDE1
AC17	VSS	AC18	VDE1	AC19	RTCCLKB
AC20	VDDI	AC21	RTCINT	AC22	STOP
AC23	TST_I0	AC24	TSTMODE	AC25	VSS
AC26	VINITH	AD1	VSS	AD2	D0
AD3	VDE2	AD4	GPIO2	AD5	GPIO5
AD6	VSS	AD7	GPIO13	AD8	SPI_CS1
AD9	SPI_CS6	AD10	VSS	AD11	XCTS0
AD12	XRTS0	AD13	XRTS1	AD14	XCTS1
AD15	USBCLK	AD16	XINT1	AD17	XINT2
AD18	VDDI	AD19	VDDI	AD20	VDDI
AD21	VSS	AD22	PLLBYPASS	AD23	VDE1
AD24	TST_I2	AD25	BOOTM2	AD26	VDE1
AE1	VDE2	AE2	VSS	AE3	GPIO0
AE4	GPIO3	AE5	GPIO6	AE6	GPIO9
AE7	GPIO14	AE8	SPI_CS2	AE9	SPI_CS7
AE10	SPITXD	AE11	UART0TXD	AE12	VSS
AE13	UART1RXD	AE14	VSS	AE15	USBCON
AE16	XINT0	AE17	XINT3	AE18	VSS
AE19	RTCCLKA	AE20	VSS	AE21	TESTCLK1
AE22	VDDI	AE23	TST_I1	AE24	VDDI
AE25	BOOTM1	AE26	VSS	AF1	VSS
AF2	VDE1	AF3	GPIO1	AF4	GPIO4
AF5	VDE1	AF6	GPIO10	AF7	GPIO15
AF8	SPI_CS3	AF9	VDE1	AF10	SPIRXD
AF11	VDE1	AF12	UART0RXD	AF13	VDE1
AF14	UART1TXD	AF15	VDE1	AF16	VSS
AF17	VDE1	AF18	VSS	AF19	VSS
AF20	VDDI	AF21	VDE1	AF22	PLLMACRO_L
AF23	VSS	AF24	VDE1	AF25	VSS
AF26	VDE1				

					ЮФКВ.431268.006РЭ				Лист
									269
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

11.2 Конструктивные характеристики

СБИС СПОВС изготавливается в 484-выводном пластиковом корпусе типа Ball Grid Array. Внешний вид корпуса микросхемы представлен ниже (см. Рисунок 11.2 – Рисунок 11.4) с указанием сведений о габаритных и установочных размерах.

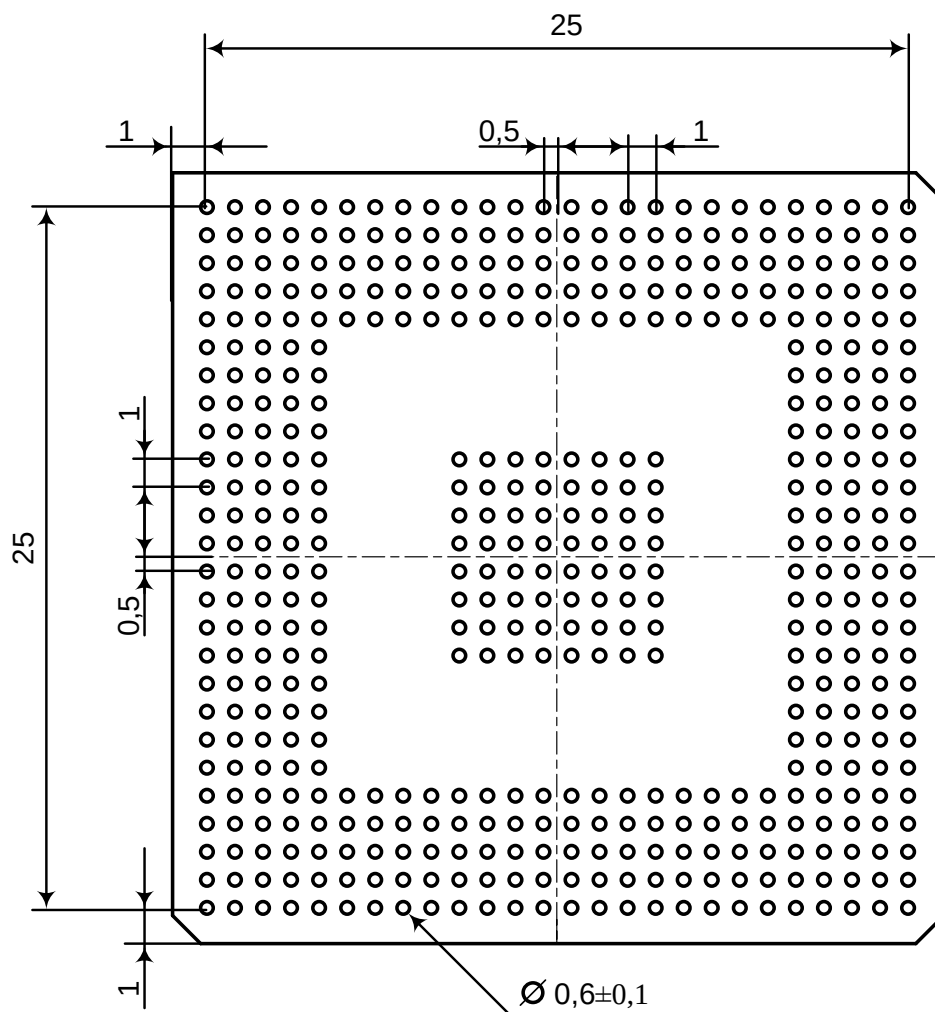


Рисунок 11.2 – Вид корпуса со стороны выводов

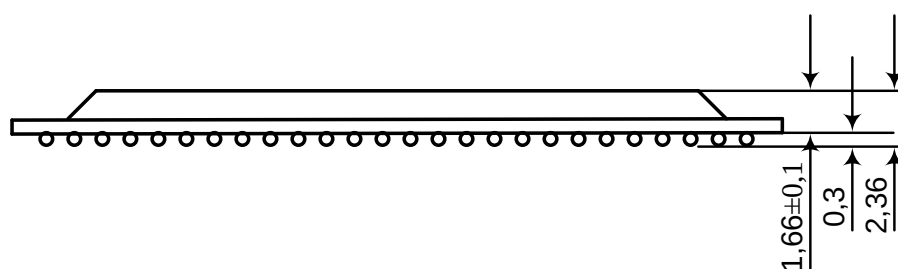


Рисунок 11.3 – Вид корпуса сбоку

					ЮФКВ.431268.006РЭ				Лист
									270
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	<i>Редко</i> 16.01.14		22755-1						

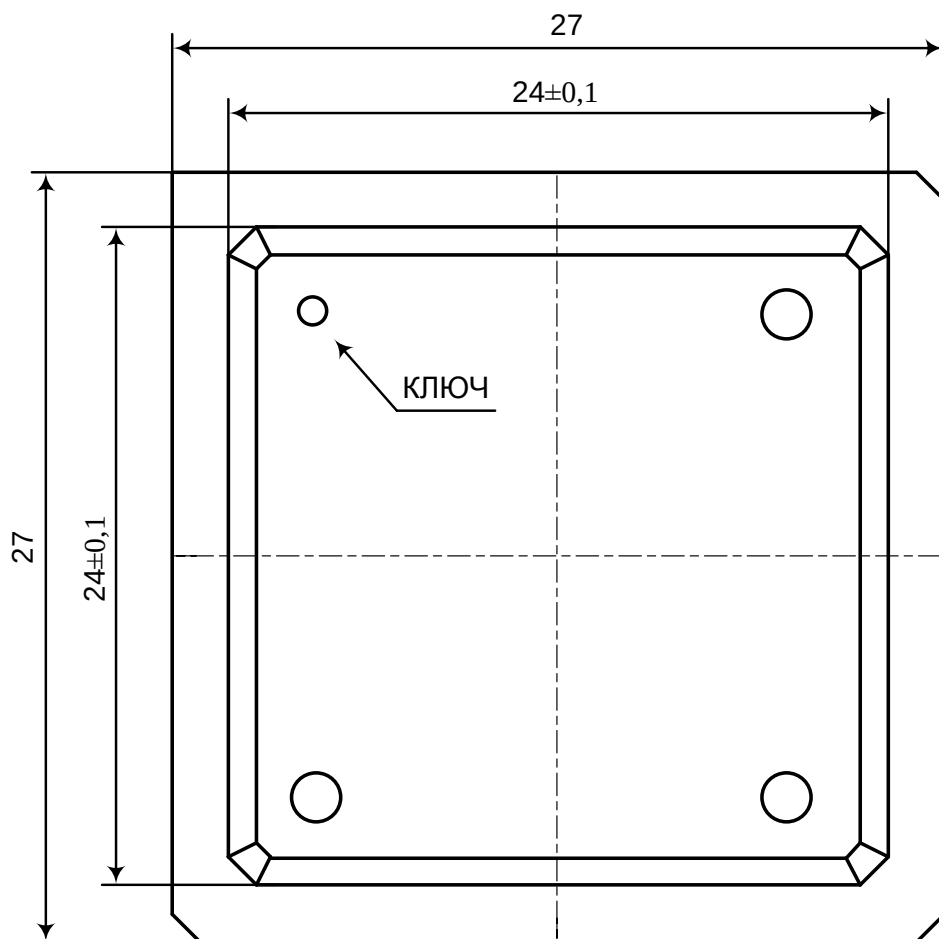


Рисунок 11.4 – Вид корпуса сверху.

Основные характеристики корпуса:

Механические характеристики:

- размер – 27×27 мм²;
- количество выводов – 484
- шаг выводов – 1 мм;
- размер вывода – 0,60 мм.
- Масса СБИС – 2,7г

Электрические характеристики:

- индуктивность вывода – не более 6,5 нГн;
- сопротивление вывода – не более 0,270 Ом;
- ёмкость вывода – не более 1,5 пФ.

Тепловые характеристики:

- Тепловое сопротивление при скорости обдува 0 м/с – 16,5 °С/Вт;
- Тепловое сопротивление при скорости обдува 1 м/с – 13,0 °С/Вт;
- Тепловое сопротивление при скорости обдува 3 м/с – 11,0 °С/Вт.

					ЮФКВ.431268.006РЭ			Лист
								271
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

11.3 Электрические характеристики

Электрические параметры микросхемы соответствуют нормам, приведенным ниже в таблицах (Таблица 11.3, Таблица 11.4).

Таблица 11.3 – Электрические параметры микросхемы при приемке и поставке

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Норма параметра			Температура среды, °С
			не менее	номинал	не более	
Выходное напряжение низкого уровня КМОП буфера, В	U_{OL1}	$I_{OL}=2\text{ мА}$ $U_{CC3}=3,6\text{ В}$ $U_{CC2}=1,3\text{ В}$	-	-	0,4	от минус 50 до +85
Выходное напряжение высокого уровня КМОП буфера, В	U_{OH1}	$I_{OH}=-2\text{ мА}$ $U_{CC3}=3,0\text{ В}$ $U_{CC2}=1,1\text{ В}$	2,0	-	-	от минус 50 до +85
Выходное напряжение низкого уровня SSTL2 буфера, В	U_{OL2}	$I_{OL}=8\text{ мА}$ $U_{CC1}=2,7\text{ В}$ $U_{CC2}=1,3\text{ В}$	-	-	$0,5U_{CC1}+0,608$	от минус 50 до +85
Выходное напряжение высокого уровня SSTL2 буфера, В	U_{OH2}	$I_{OH}=-8\text{ мА}$ $U_{CC1}=2,3\text{ В}$ $U_{CC2}=1,1\text{ В}$	$0,5U_{CC1}-0,608$	-	-	от минус 50 до +85
Ток утечки низкого уровня на входе КМОП буфера, мкА	I_{LIL}	$U_{IL}=0\text{ В}$ $U_{CC3}=3,6\text{ В}$ $U_{CC2}=1,3\text{ В}$	-	-	5,0	от минус 50 до +85
Ток утечки высокого уровня на входе КМОП буфера, мкА	I_{LIH}	$U_{IH}=3,6\text{ В}$ $U_{CC3}=3,6\text{ В}$ $U_{CC2}=1,3\text{ В}$	-	-	5,0	от минус 50 до +85
Ток потребления статический, мА	I_{CC1}	$f_{\odot}=0\text{ МГц}$ $U_{CC1}=2,7\text{ В}$ $U_{CC2}=1,3\text{ В}$ $U_{CC3}=3,6\text{ В}$	-	-	300,0	от минус 50 до +85
Ток потребления статический, мА	I_{CC2}	$f_{\odot}=0\text{ МГц}$ $U_{CC1}=2,7\text{ В}$ $U_{CC2}=1,3\text{ В}$ $U_{CC3}=3,6\text{ В}$	-	-	50,0	от минус 50 до +85
Максимальная частота следования импульсов тактовых сигналов, МГц	f_{C1}	$U_{CC1}=2,3\text{ В}$ $U_{CC2}=1,1\text{ В}$ $U_{CC3}=3,0\text{ В}$	-	-	82	от минус 50 до +85
Максимальная частота следования импульсов тактовых сигналов, Гц	f_{C2}	$U_{CC1}=2,3\text{ В}$ $U_{CC2}=1,1\text{ В}$ $U_{CC3}=3,0$	-	-	32768	от минус 50 до +85
Максимальная частота следования импульсов тактовых сигналов, МГц	f_{C3}	$U_{CC1}=2,3\text{ В}$ $U_{CC2}=1,1\text{ В}$ $U_{CC3}=3,0$	-	-	48,0	от минус 50 до +85

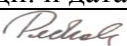
					ЮФКВ.431268.006РЭ			Лист
								272
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица 11.4 – Электрические параметры АЦП

Наименование параметра, единица измерения	Буквенное обозначение	Режим измерения	Норма параметра			Температура среды, °C
			не менее	номинал	не более	
Разрядность преобразования АЦП, бит	N		12			от минус 50 до +85
Частота преобразования, МГц	F _S		-	-	82	от минус 50 до +85
Дифференциальная нелинейность АЦП, LSB ¹⁾	DNL			-	±0,5	от минус 50 до +85
Интегральная нелинейность АЦП, LSB	INL		-	-	±4	от минус 50 до +85
Смещение нуля, мВ	U _{OFS}			-	±14,7	от минус 50 до +85
Напряжение средней точки входного аналогового сигнала, В	V _{IC}			1,4		от минус 50 до +85
Диапазон напряжений входного аналогового сигнала, В	V _{INS}		1,0		1,8	от минус 50 до +85
Амплитуда дифференциального входного сигнала ²⁾ , В	V _{IND}				1,6	от минус 50 до +85
Входное сопротивление, кОм	Z _{INDC}		100			от минус 50 до +85
Отношение сигнал/шум, dBc	SNR		59			от минус 50 до +85
Отношение мощности сигнала к мощности 5 гармоник, dBc	THD		52	-		от минус 50 до +85
Динамический диапазон, dBFc	SFDR	Fs=82MS/s Fin=20MHz	65	-		от минус 50 до +85
Развязка между АЦП, dBFc	XTALK	Fs=82MS/s Fin=20MHz	-40			от минус 50 до +85

Примечания:

1) 1 LSB = 0,39мВ;

2) АЦП имеет дифференциальный вход. Под амплитудой понимается размах сигнала, равный учетверенной амплитуде синусоидального сигнала каждой фазы сигнала.

Значения предельно-допустимых и предельных электрических режимов эксплуатации микросхемы в диапазоне рабочих температур среды приведены ниже в таблице (Таблица 11.5).

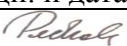
					ЮФКВ.431268.006РЭ			Лист
								273
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица 11.5 – Нормы предельно-допустимых и предельных электрических режимов эксплуатации микросхемы

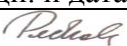
Наименование параметра режима, единица измерения	Буквен- ное обо- значение	Предельно – допустимый режим			Предельный режим	
		не менее	норм.	Не более	не менее	не более
Напряжение питания буферов ввода/вывода SSTL2 (VDE2), В	U_{CC1}	2,3	2,5	2,7	-0,5	3,6
Напряжение питания ядра (VDDI), В	U_{CC2}	1,1	1,2	1,3	-0,5	1,8
Напряжение питания буферов ввода/вывода, USB (VDE1, VDE3), В	U_{CC3}	3,0	3,3	3,6	-0,5	4,0
Напряжение питания аналогового блока АЦП (AVD33), В		3,0	3,3	3,6	-0,5	4,0
Напряжение питания для PLL (AVD12), В		1,1	1,2	1,3	-0,5	1,8
Входное напряжение высокого уровня, В	U_{IH}	2,0	-	$U_{CC3}+0,3$	-	-
Входное напряжение низкого уровня, В	U_{IL}	-0,3	-	$0,3U_{CC3}$	-	-
Напряжение на стандартных входах, В	U_{IN}	-0,3	-	$U_{CC3}+0,3$	-0,5	3,6
Выходное напряжение, В	U_{OUT}	-0,3	-	U_{CC3}	-0,5	3,6
Входной ток, мкА	I_{IN}	-	-	5	-	5
Выходной ток высокого уровня, мА 4- миллиамперный выход 6- миллиамперный выход 8- миллиамперный выход	I_{OH}	-	-	-4	-	-13
		-	-	-6	-	-13
		-	-	-8	-	-13
Выходной ток низкого уровня, мА 4- миллиамперный выход 6- миллиамперный выход 8- миллиамперный выход	I_{OL}	-	-	4	-	13
		-	-	6	-	13
		-	-	8	-	13

11.4 Порядок подачи питания при включении

Последовательность подачи напряжений питания и сигналов при включении:

- 1) Напряжение питания для внутренней логики VDDI;
- 2) Напряжение питание для аналоговых блоков AVD33, AVD12;
- 3) Напряжение питания для буферов ввода/вывода: VDE1, VDE2, VDE3;
- 4) Тактовый сигнал CLKIN0, CLKIN1;
- 5) системный сброс XRESET.

Последовательность подачи остальных сигналов не регламентируется. Выключение напряжений питания производится в порядке, обратном включению.

					ЮФКВ.431268.006РЭ			Лист
								274
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

12 Временные характеристики

12.1 Временные параметры SPI порта

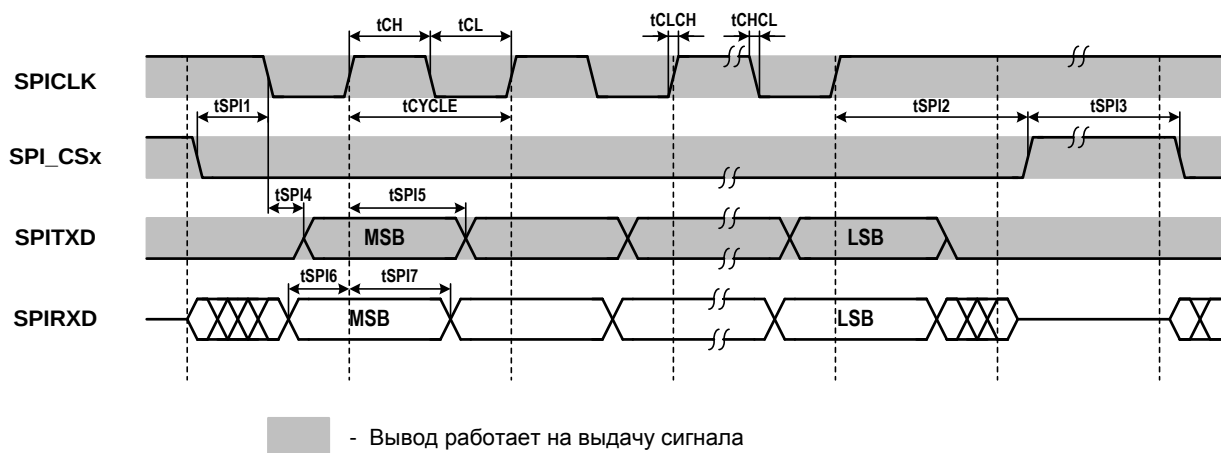


Рисунок 12.1 – Задержки SPI порта (на примере SPI mode 3)

Таблица 12.1. Задержки SPI порта

Символ	Описание	мин	макс
t_{CYCLE}	период синхросигнала (в формулах обозначается как t_C)	24.4 нс	
t_{CH}	полупериод синхросигнала	12.2 нс	
t_{CL}	полупериод синхросигнала	12.2 нс	
t_{CLCH}	фронт синхросигнала (30пф)		2.5 нс
t_{CHCL}	срез синхросигнала (30пф)		2.5 нс
t_{SPI1}	время от изменения SPI_CSx в активное состояние до первого фронта или среза синхросигнала		
	SPI mode 0 и 2	$t_C - 4$ нс	
	SPI mode 1 и 3	$t_C/2 - 4$ нс	
t_{SPI2}	время от последнего фронта или среза синхросигнала до изменения SPI_CSx в неактивное состояние		
	SPI mode 0 и 2	$t_C/2$	
	SPI mode 1 и 3	t_C	
t_{SPI3}	повторная активация сигнала SPI_CSx		
	SPI mode 0 и 2	$t_C/2$	
	SPI mode 1 и 3	нет	
t_{SPI4}	задержка выдачи данных относительно активного фронта или среза тактового сигнала		2.5 нс
t_{SPI5}	время удержания выходных данных, относительно активного фронта или среза тактового сигнала	$t_C/2$	
t_{SPI6}	время предустановки входных данных, относительно активного фронта или среза		4 нс
t_{SPI7}	время удержания входных данных, относительно активного фронта или среза	0 нс	

Работа порта была проверена с использованием моделей микросхем памяти с SPI интерфейсом фирмы Numonux MP25P10 и M25PX64.

					ЮФКВ.431268.006РЭ			Лист
								275
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

При выборе микросхемы flash памяти, необходимо в первую очередь обратить внимание на максимальную частоту работы flash памяти для всех команд, планируемых к использованию по SPI интерфейсу, т.к. часто не все команды могут обрабатываться микросхемой памяти на заявленной максимальной частоте. Например, микросхема M25PX64 правильно обрабатывает SPI команду READ(код – 0x03) только на частотах меньше 33МГц ($t_{CYCLE} < 30.3$ нс).

Также необходимо обратить внимание на задержку выдачи данных микросхемой SPI flash памяти, относительно активного фронта или среза. Эта задержка не должна превышать 8.2 нс ($t_C/2 - t_{SPI6}$).

					ЮФКВ.431268.006РЭ			Лист
								276
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редко</i> 16.01.14		22755-1				

13 Рекомендации по включению

13.1 Рекомендации по включению кварцевого резонатора для счетчика реального времени и для тактового сигнала USB

Схема подключения кварцевого резонатора приведена ниже (см. Рисунок 13.1).

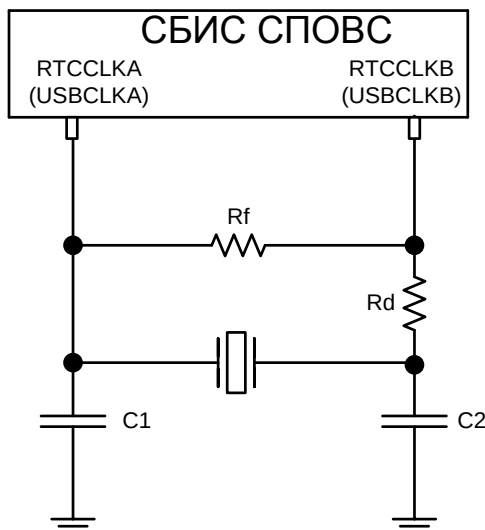


Рисунок 13.1 – Схема подключения часового кварца (кварца для USB).

Точные значения параметров схемы зависят от конкретного кварцевого резонатора и реализации печатной платы. Вариант значений параметров резисторов R_f , R_d и конденсаторов C_1 , C_2 приведен в таблице ниже (

Таблица 13.1).

Таблица 13.1 – Пример параметров резисторов R_f , R_d и конденсаторов C_1 , C_2 .

	Частота МГц	C_1, C_2 , пФ	R_f , Мом	R_d , Ом	Модель резона- тора	Порядок	Емкость на- грузки резона- тора
Значение для тактового сиг- нала RTC	0,032768	10	10	-	DT-26/DT-28 (Daishinku)	фундамен- тальная частота	12,5
Значения для тактового сиг- нала USB	48	7	1	590	FCX-04 (River eletec)	фундамен- тальная частота	8

Помимо кварцевого резонатора можно подать тактовый сигнал, сформированный внешним образом. При этом тактовый сигнал подается на вывод RTCCLKA (USBCLKA), а вывод RTCCLKB (USBCLKB) должен остаться не подключенным.

13.2 Рекомендации по подключению тактового сигнала микросхемы (до 82МГц)

Схема включения блока ADCU в режиме осциллирующего буфера с использованием внешнего кварцевого резонатора представлена ниже (см. Рисунок 13.2).

					ЮФКВ.431268.006РЭ			Лист
								277
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редько</i> 16.01.14		22755-1				

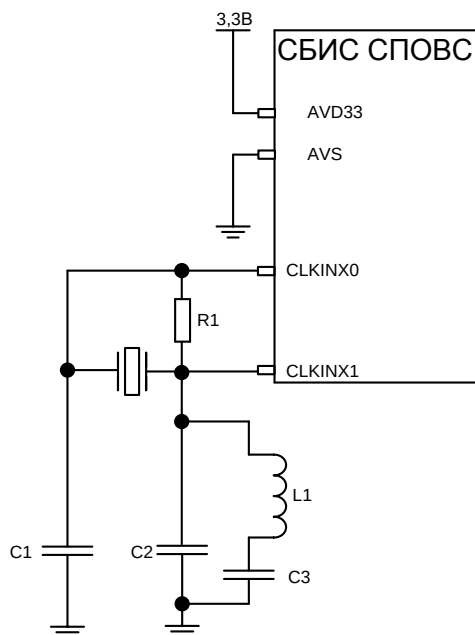


Рисунок 13.2 – Схема включения блока ADCU в режиме осциллирующего буфера

Емкость нагрузки кварцевого резонатора (C_{load}) определяется производителем и должна удовлетворять следующему условию:

$$C_{load} = C_{in} * C_{out} / (C_{in} + C_{out}) \quad (1), \text{ где}$$

$$C_{in} = C1 + C_{PCB} + C_{PKG} + C_{X0}$$

$$C_{out} = C2 + C_{PCB} + C_{PKG} + C_{X1}$$

C_{PKG} – емкость выводов корпуса ($C_{PKG} = 1,5\text{пФ}$)

C_{PCB} – емкость проводника печатной платы

C_{X0}, C_{X1} – емкость выводов (CLKINX0, CLKINX1) ($C_{X0} = 0,4\text{пФ}$, $C_{X1} = 4,4\text{пФ}$).

Ниже в таблице приведены значения элементов C1, C2, C3, L, R1.

Таблица 13.2 – значения элементов C1, C2, C3, L, R1

компонент	как определять	значение
C1, C2	Определяется из емкости нагрузки кварцевого резонатора и зависит от производств печатной платы. C1, C2 должны удовлетворять условию (1)	Типовые значения $\approx 10\text{ пФ}$
C3	Развязывающий конденсатор по постоянному току. Нужен, когда кварцевый резонатор работает не на фундаментальной частоте. Сопротивление должно быть меньше сопротивления индуктивности на резонансной частоте.	
L1	Индуктивность фильтра. Используется, когда кварцевый резонатор работает не на фундаментальной частоте. Определяется по формуле: $L1 = 1 / [(2\pi f)(2\pi f/3)C_{out}]$ f- частота гармоники	
R1	Внешний резистор обратной связи. Нужен, если не используется внутренний резистор обратной связи	100 Ком
Y1	Кварцевый резонатор может работать как на фундаментальной частоте, так и на гармониках (до третей	

					ЮФКВ.431268.006РЭ			Лист
								278
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		<i>Редько</i> 16.01.14		22755-1				

гармоники) в зависимости от частоты и последовательного сопротивления резонатора.

Схема включения блока ADCU в режиме входного буфера приведена на рисунке ниже (см. Рисунок 13.3). Возможно как дифференциальное включение внешнего тактового генератора, так и однофазное.

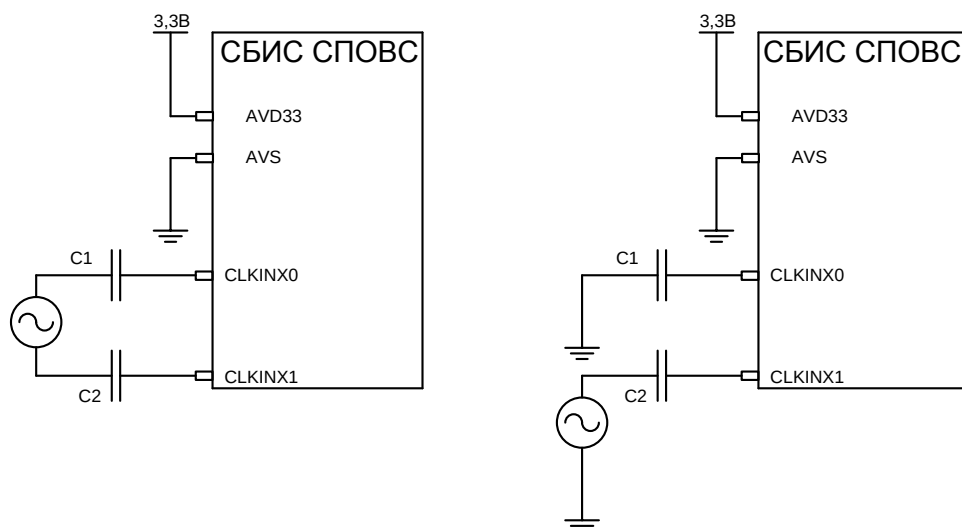


Рисунок 13.3 – Схема включения блока ADCU в режиме входного буфера.

Дифференциальное включение предпочтительнее, так как это включение менее восприимчиво к шумам на плате. C1, C2 – разделительные конденсаторы, емкостью по 10нФ.

Ниже приведены параметры сигнала тактового генератора

Таблица 13.3 – параметры сигнала тактового генератора

Тип включения	Минимальный размах входного сигнала, В	Максимальный размах входного сигнала, В
Однофазное включение	0,25	0,75
Дифференциальное включение	0,15	1,25

13.3 Рекомендации по включению АЦП

Ниже приведена рекомендованная схема включения фильтрующих конденсаторов для четырех АЦП (см. Рисунок 13.4).

					ЮФКВ.431268.006РЭ			Лист
								279
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		<i>Редко</i> 16.01.14		22755-1				

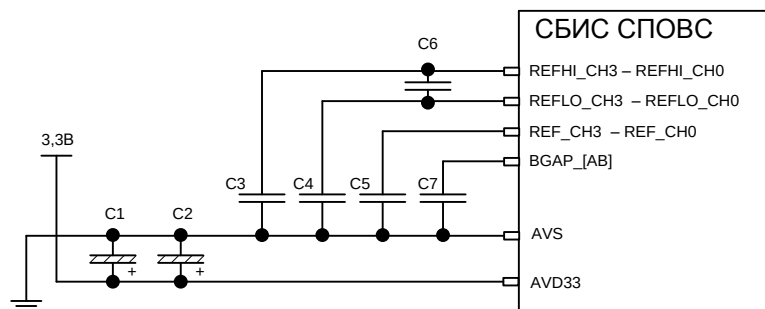


Рисунок 13.4 – подключение фильтрующих конденсаторов для АЦП

Значение емкостей конденсаторов:

$C1 = 33 \text{ мкФ}$; $C2 = 4,7 \text{ мкФ}$; $C3 = C4 = C5 = C6 = C7 = 0,1 \text{ мкФ}$.

Входной аналоговый сигнал рекомендуется подавать на входы АЦП, используя одну из приведенных ниже схем (см. Рисунок 13.5).

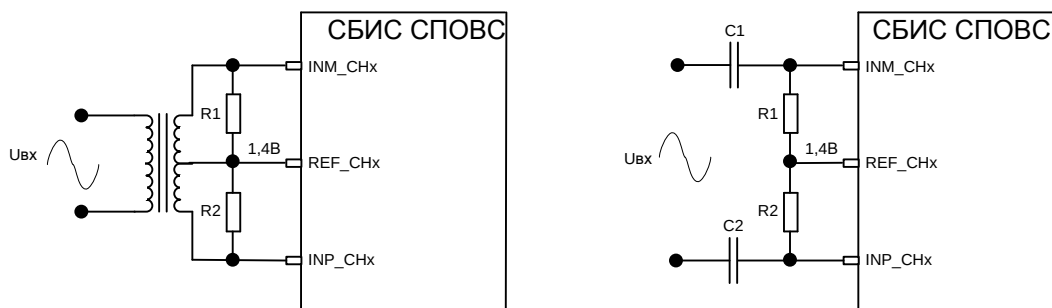


Рисунок 13.5 – Рекомендуемые варианты подключения входного аналогового напряжения к АЦП

Значение компонентов схем: $C1 = C2 = 1 \text{ мкФ}$; $R1 = R2 = 25 \text{ Ом}$.

13.4 Рекомендации по реализации схемы начального сброса

Ниже приведены примеры реализации схемы сброса. Пунктиром на рисунках обозначены дополнительные элементы, которые можно поставить для повышения помехозащищенности на входе XTRST. Это решение нельзя применить к сбросу XSRST, т.к. он двунаправленный. Также необходимо помнить, что буфера XSRST и XTRST имеют встроенные pullup резисторы.

Минимальная длительность активного состояния сигнала сброса XRESET – 3 такта синхросигнала XCLK, т.е. около 40 нс. Минимальная длительность активного состояния сигнала сброса XSRST – 7 тактов внутреннего синхросигнала CCLK, т.е. около 25 нс.

					ЮФКВ.431268.006РЭ				Лист
									280
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата			
22755-2		<i>Редкофф</i> 16.01.14		22755-1					

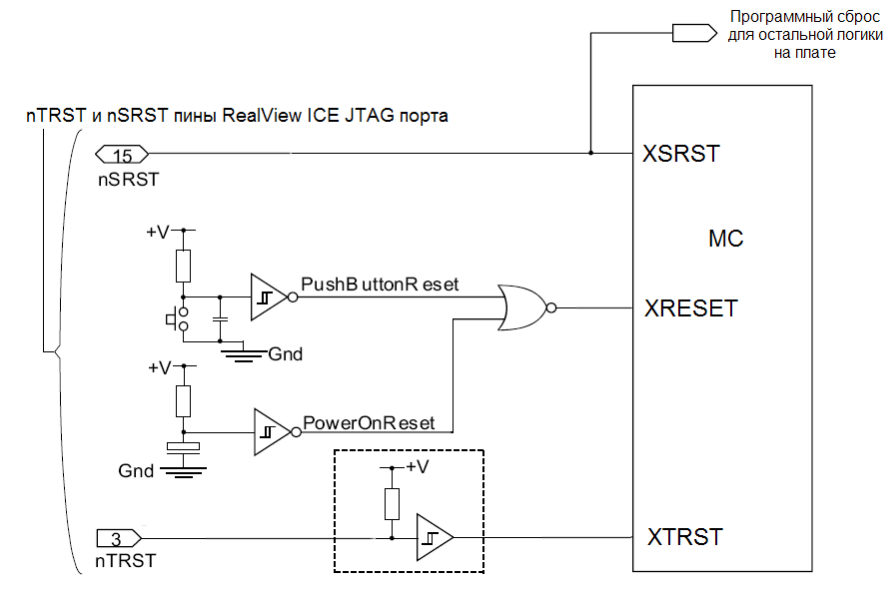


Рисунок 13.6 – Пример реализации схемы сброса

Возможно использовать специализированную микросхему монитора питания для реализации схемы сброса, например – MAX823:

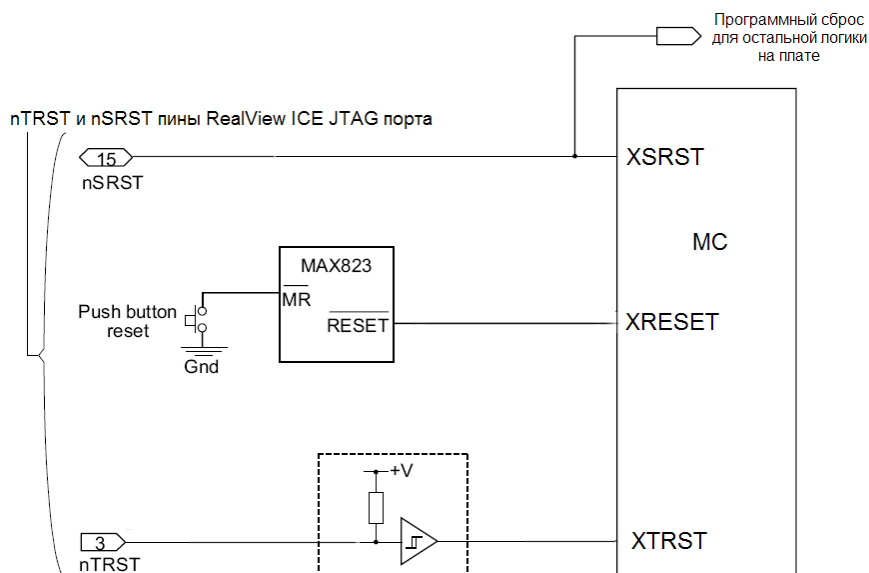


Рисунок 13.7 – Пример реализации схемы сброса с использованием микросхемы MAX823

13.5 Рекомендации по подключению USB интерфейса

СБИС СПОВС имеет интерфейс USB2.0 (full speed) device. Вариант подключения СБИС СПОВС к устройству USB HOST (USB HUB) приведен на рисунке ниже (см. Рисунок 13.8). Вывод TD9 управляет разрешением подключения. Функция управления подключением используется начальным загрузчиком микросхемы (см. п. 5.17.7), поэтому при необходимости загрузки посредством USB данная схема должна быть реализована. Рекомендуемые значения резисторов:

$R_s = 27 \text{ Ом} \pm 5\%$;

$R_{pd} = 330 \text{ Ком} \pm 5\%$;

					ЮФКВ.431268.006РЭ			Лист
								281
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№	подл.	Подп. и дата		Взам.инв.№	Инв.№	дубл.	Подп. и дата	
22755-2		<i>Редкофф</i> 16.01.14		22755-1				

$R_{pu} = 1,5 \text{ Ком} \pm 5\%$;
 $R1 = 47 \text{ Ком} \pm 5\%$;
 $R2 = 27 \text{ Ком} \pm 5\%$;

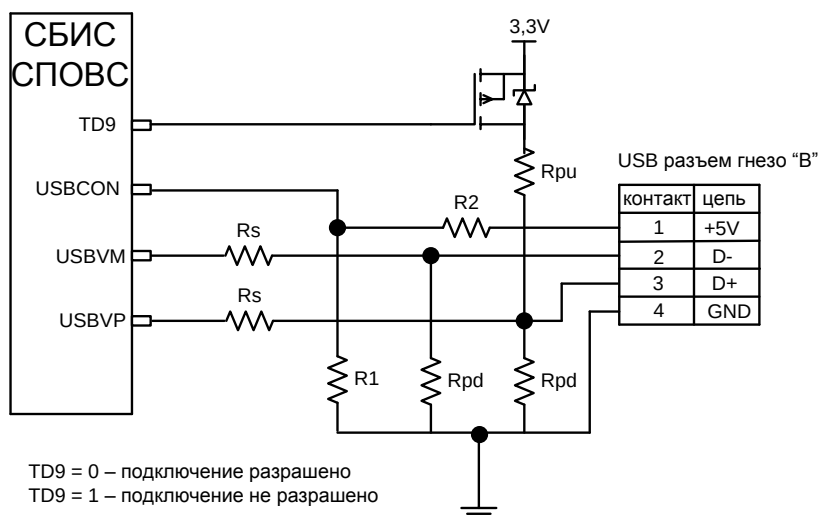


Рисунок 13.8 – Схема подключения USB интерфейса

13.6 Рекомендации по включению DDR

13.6.1 Рекомендации по согласованию линий связи интерфейса DDR1 SDRAM

Сигналы адресной шины и управления – A0...A13, XCS, XRAS, XCAS, XWE, SCKE. Для согласования однонаправленных сигналов адресной шины и сигналов управления используется параллельно-последовательная схема (см. Рисунок 13.9). Последовательные согласующие резисторы R_s размещаются непосредственно около выводов микросхемы. Со стороны приемника должны быть установлены согласующие pull-up резисторы R_t , подключенные к VTT.

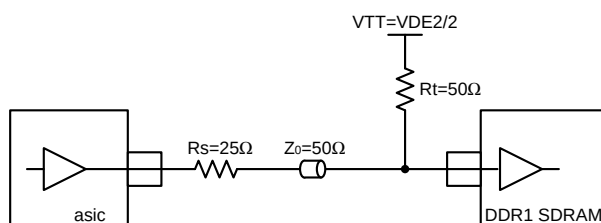


Рисунок 13.9 – Схема согласования однонаправленных сигналов

Сигналы шины данных, стробов и маски – D, DQS, DM. Согласование двунаправленных сигналов шины данных, стробов и маски осуществляется по параллельно-последовательной схеме (см. Рисунок 13.10). На обоих концах линий D, DQS и DM на минимально близком от соответствующих выводов расстоянии должны быть установлены согласующие резисторы.

					ЮФКВ.431268.006РЭ			Лист
								282
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

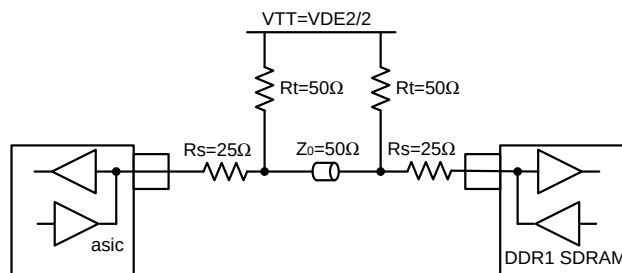


Рисунок 13.10 – Схема согласования двунаправленных сигналов

Тактовые сигналы – CLK, CLKX. Согласование дифференциальных тактовых импульсов производится в соответствии со схемой (см. Рисунок 13.11). На тактовом сигнале параллельный согласующий резистор устанавливается между прямой и обратной фазами синхросигнала на приемной стороне.

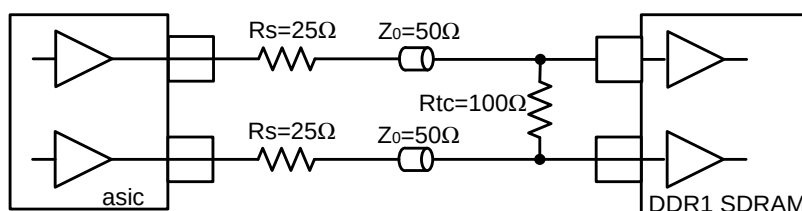


Рисунок 13.11 – Схема согласования дифференциальных сигналов

В таблице ниже (Таблица 13.4) приведены рекомендуемые значения резисторов и длин линий связи.

Таблица 13.4 – Рекомендуемые значения резисторов и длин линий.

Параметр	символ	Мин.	Тип.	Макс.	Ед. Измерения
Последовательный согласующий резистор	R_s	23.75	25	26.25	Ом
Параллельный согласующий резистор для тактового сигнала	R_{tc}	99	100	101	Ом
Параллельный согласующий резистор для всех сигналов кроме тактового сигнала.	R_t	47.5	50	52.5	Ом
Длина линии тактового сигнала, адресных сигналов и сигналов управления (XCS, XRAS, XCAS, XWE, SCKE)		45	50	55	мм
Длина линии сигнала DQS		30		50	мм
Разброс длин сигналов внутри одной DQS группы		-5		+5	мм

13.6.2 Рекомендации по проектированию печатной платы для DDR интерфейса

Чтобы избежать влияния перекрестных наводок (crosstalk), экранируйте сигнальные слои на плате, т.е., используйте структуры STRIP(проводник между экранами) или MICRO STRIP(проводник над экраном). Пример структуры печатной платы приведен на рисунке ниже (см. Рисунок 13.12)

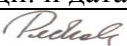
					ЮФКВ.431268.006РЭ			Лист
								283
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				



Рисунок 13.12 – Пример структуры печатной платы

Для дифференциального тактового сигнала CLK и CLKX старайтесь делать трассировку таким образом, чтобы сигналы CLK и CLKX имели одинаковую задержку. Следующие рекомендации должны помочь в достижении этой цели:

- Длина для сигналов CLK и CLKX должна быть одинаковой.
- Используйте один и тот же сигнальный слой для трассировки CLK и CLKX. Это позволит минимизировать разброс волнового сопротивления для сигналов CLK и CLKX.
- Минимизируйте разброс параметров линий, делая одинаковое число переходных отверстий по каждой линии CLK и CLKX.
- Используйте одинаковое расстояние между проводниками (wiring pitch) для сигналов CLK и CLKX. Это снизит разброс влияния перекрестных наводок.

Группировка выводов D, DQS и DM. Группируйте каждые 8 бит данных D, DQS и DM в одну группу:

DQS0 => DQ0 - DQ7, DM0 - первая группа
DQS1 => DQ8 - DQ15, DM1 - вторая группа
DQS2 => DQ16- DQ23, DM2 - третья группа
DQS3 => DQ24- DQ31, DM3 - четвертая группа

В каждой байтовой группе при трассировке печатной платы минимизируйте разницу задержек между каждым выводом группы DQ, DQS и DM, а затем добивайтесь минимальной разницы задержек между группами.

Трассировка сигналов SCLK1/XSCLK1, SCLK2/XSCLK2 и DQS особенно важна, так как эти управляющие сигналы являются опорными для формирования остальных сигналов микросхемы. Экранируйте данные цепи шинами земли. Это снизит влияние перекрестных наводок от других сигнальных выводов. Аналогичные действия рекомендуются и для сигнала SCKE. Используйте всю свободную от проводников площадь сигнальных слоев для создания земляных шин и областей, при этом с помощью множественных переходных отверстий соедините их с внутренними земляными экранами печатной платы.

Для адресных сигналов и сигналов управления (XCS, XRAS, XCAS, XWE, SCKE) старайтесь делать трассировку таким образом, чтобы длина проводников этих сигналов была одинаковой.

Экранирование сигналов VREF. Сигнал VREF определяет логический порог срабатывания для входных сигналов. Этот уровень должен иметь постоянное значение. Для обеспечения высокой стабильности VREF рекомендуется следующее:

- Экранируйте данные цепи шинами земли. Это снизит влияние перекрестных наводок от других сигнальных выводов.
- Используйте более широкие проводники для сигналов VREF и VTT. Это уменьшит сопротивление цепи и, соответственно, шум. Если возможно, используйте для этого отдельный слой.

					ЮФКВ.431268.006РЭ			Лист
								284
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

- Не объединяйте VREF и VTT. Используйте разные источники питания для VTT и VREF. Это связано с тем, что на шине VTT возможны колебания напряжения в процессе переключений шины.
- В зависимости от индуктивности или емкости корпуса, VREF может иметь LC резонансную частоту, на которой могут возникнуть колебания. Печатная плата должна быть спроектирована таким образом, чтобы можно было установить дополнительный внешний конденсатор для изменения резонансной частоты.

Используйте конденсаторы для снижения помех по питанию. Размещайте конденсаторы как можно ближе к выводам микросхемы. Используйте высоко- и низкочастотные конденсаторы на цепях питания.

Для одной DQ группы используйте один слой для трассировки.

При применении проводников змеевидной формы располагайте параллельные сегменты проводника на большем расстоянии (wiring pitch) друг от друга. Это минимизирует собственные перекрестные наводки, влияющие на задержку проводника.

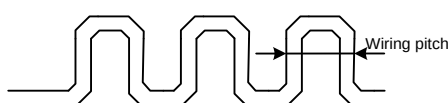


Рисунок 13.13 – Проводник змеевидной формы

В точках разветвления сигналов, подсоединенных к нескольким микросхемам (например, тактовый сигнал), может иметь место эффект отражения. Это возникает из-за различия в волновом сопротивлении. Для предотвращения отражения используйте подстроечный резистор для выравнивания волнового сопротивления.

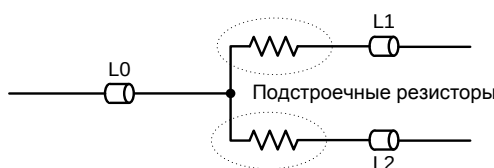


Рисунок 13.14 – Использование подстроечных резисторов

					ЮФКВ.431268.006РЭ			Лист
								285
Изм.	Лист	№ докум.	Подп.	Дата				
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата		
22755-2	<i>Редко</i> 16.01.14		22755-1					

А Приложение. Система команд DSPPU

А.1 Основные режимы работы

В блоке DSPPU потоки оцифрованных данных проходят через предварительно настроенные конвейерные арифметические каналы, пакуются и записываются в память для дальнейшей обработки. Временные метки разделяют эти потоки на кадры.

Блок DSPPU содержит 24 канала обработки данных, которые разделены на 6 кластеров, по 4 канала в каждом. Первые три кластера содержат КИХ-фильтры и квадраторы. Еще имеется дополнительно 4 канала генерации псевдослучайных последовательностей. Все эти 28 каналов содержат гетеродины.

На вход DSPPU данные могут поступать с четырёх типов источников:

- от ADC – 12-разрядные данные,
- с внешних выводов SIGNx/MAGNx – 2-разрядные данные,
- через блок DMAC – упакованные в 64-разрядные слова данные,
- из программного потока (командой WR_ADC_ins) – 12-разрядные данные.

С помощью входного устройства DSPPU можно выбрать источники данных и направить данные по четырём выходным интерфейсам входного устройства. Каждый канал берёт свои данные с какого-либо из этих четырёх интерфейсов.

Работа блока программного управления DSPPU происходит следующим образом. На фоне обработки текущего кадра управляющая программа осуществляет анализ и новую настройку каналов путем записи в теневые регистры. Затем выдается групповая команда пуска нужных каналов WR_Start. Каждый из этих каналов запоминает предназначенный ему пуск (взводится бит состояния Tstart). По приходу в канал временной метки управляющая информация настройки канала переписывается из теневых регистров в рабочие (основные) и канал начинает принимать данные. Следует отметить, что коэффициенты фильтров таким простым способом изменять нельзя, не нарушая работу текущего кадра, – в этом случае следует остановить нужный канал (каналы) и записать новые коэффициенты в уже не работающий фильтр.

Возможны 2 режима запуска каналов: однокадровый и серийный.

При однокадровом режиме запущенные каналы обрабатывают только данные, приходящие в ближайшем временном интервале. Допускается в течение этого временного интервала выдавать следующую команду запуска – в этом случае будут обработаны два смежных кадра. В однокадровом режиме можно обрабатывать данные непрерывно, нужно только своевременно (один раз за временной интервал) выдавать команду пуска. Однокадровый режим всегда используется, если входные данные канала поступают через DMAC или из программного потока. Если данные поступают в канал от ADC или с выводов SIGNx/MAGNx, то однокадровый режим может быть включен с помощью поля от команды WR_select.

В серийном режиме канал продолжает работу до прихода команды WR_Stop. Канал запоминает предназначенный ему стоп (взводится бит состояния Tstop). По приходу временной метки бит Tstop обнуляется, взводится бит Ttstop. Канал остановлен и не воспринимает больше временные метки и данные (до прихода следующей команды старта и метки). При Ttstop=1 новая команда останова игнорируется.

Таким образом, текущее состояние канала определяются 3-мя битами состояния (Tstart, Tstop, Ttstop), которые доступны по чтению командой RD_het_base, либо командой RD_frequency, либо командой RD_current. Все возможные состояния представлены в таблице ниже (см. Таблица А.1).

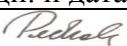
					ЮФКВ.431268.006РЭ			Лист
								286
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица А.1 – Состояния канала

Значения триггеров	Описание
Tstart=0, Tstop=0, Ttstop=1	канал остановлен – исходное состояние после команды WR_Reset и после включения питания.
Tstart=1, Tstop=0, Ttstop=1	ожидание метки – теневые регистры установлены и была выдана команда WR_Start (сам канал пока остановлен).
Tstart=0, Tstop=0, Ttstop=0	канал обрабатывает текущий кадр (и будет обрабатывать следующий кадр при той же настройке, если не поступит новых указаний).
Tstart=1, Tstop=0, Ttstop=0	была коррекция или полная перенастройка канала – канал обрабатывает текущий кадр, метка переключит канал на новую настройку.
Tstart=0, Tstop=1, Ttstop=0	канал обрабатывает текущий кадр и по приходу метки остановится (перед меткой пришла команда WR_Stop, одновременно с меткой она прийти не может).
Tstart=1, Tstop=1, Ttstop=0	состояние, возникающее в однокадровом режиме или после команды WR_Start, попавшей на состояние Tstart=0, Tstop=1, Ttstop=0.
Tstart=0, Tstop=1, Ttstop=1	не бывает
Tstart=1, Tstop=1, Ttstop=1	не бывает

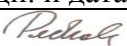
После окончания каналом записи кадра в память в специальном блоке DSPPU – стеке монитора – сохраняется запись. С помощью команды WR_AG_control_monitor можно настроить стек монитора и задать условие, при котором процессорным системам NMU1 и NMU2 будет выдаваться запрос на прерывание.

В процессе обработки потоков данных может потребоваться либо корректировка настройки каналов (не затронутые корректировкой теневые регистры сохраняют свои старые значения), либо их полная перестройка. Система команд и схемы управления блоком DSPPU допускают настройку и/или перенастройку каналов во время обработки кадра, запуск каналов в заданных режимах и их останов.

Настройка DSPPU изменяется только в моменты поступления временных меток и гарантируется целостность данных как внутри кадров, так и между кадрами (то есть при перенастройке канала данные не теряются).

Некоторые действия программы могут нарушить целостность кадра, поэтому они отслеживаются аппаратурой DSPPU. Обнаружение одной из перечисленных ситуаций может стать причиной прерывания:

- одному и тому же каналу в течение одного временного интервала выдаётся более одной команды WR_Start,
- после команды WR_Start и до прихода метки программа начинает модифицировать настройку канала,
- источник входных данных изменяется при работающем канале.

					ЮФКВ.431268.006РЭ				Лист
									287
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

A.2 Типы команд блока DSPPU

Команды в DSPPU могут поступать от любой из процессорных систем NMU1 и NMU2. Выдача команд производится программой, работающей на процессорном ядре, каждая команда представляет собой команду записи в память или чтения из памяти по зарезервированным для DSPPU адресам.

Каждая команда имеет адрес, в котором есть постоянная часть (разряды [31:11]) и значимая часть (разряды [10:0]). Данные в командах DSPPU – 32-разрядные. Формат данных зависит от типа команды и от адреса команды. Команды DSPPU разделены на 3 типа:

- команды обращения к регистрам,
- команды обращения к памяти коэффициентов фильтров,
- команды обращения к памяти адресных генераторов,
- команды обращения к регистрам адресных генераторов.

Данные типы команд разделены в адресном пространстве DSPPU, каждый тип команд имеет свой формат адреса (см. Рисунок A.1, Рисунок A.2, Рисунок A.3 и Рисунок A.4).

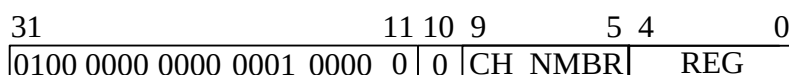


Рисунок A.1 – Формат адреса команд обращения к регистрам

Поле CH_NMBR (разряды адреса 9-5) – номер аппаратного канала обработки. Нумерация аппаратных каналов обработки следующая:

- 0-23 – каналы обработки входных данных,
 - 24-27 – каналы генераторов псевдослучайных последовательностей;
 - остальные значения поля CH_NMBR используются для кодирования специальных команд:
 - 28 – команды управления ADC (аналого-цифровые преобразователи, ADC пакковка),
 - 29 – команды обращения к регистрам счетчиков пересчета меток от интервального таймера,
 - 31 – команды синхронизации (пуски, стопы, маски прерываний, Reset DSPPU).
- Поле REG (разряды адреса 4-0) – номер регистра в канале.

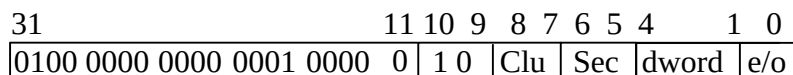


Рисунок A.2 – Формат адреса команд обращения к памяти коэффициентов фильтров

Поле Clu (разряды адреса 8-7) – номер кластера фильтров. Возможные значения 0, 1, 2.

Поле Sec (разряды адреса 6-5) – номер секции внутри кластера. Возможные значения:

- 0 – секция А,
- 1 – секция В,
- 2 – секция С,

					ЮФКВ.431268.006РЭ	Лист	
						288	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

3 – секция D.

Поля dword и e/o (разряды адреса 4-0) – формируют адрес в памяти коэффициентов внутри секции. При выдаче команды записи коэффициентов фильтров необходимо, чтобы секция фильтра, к которой идёт обращение, была остановлена. В каждой секции имеется 64 комплексных коэффициента, каждый из которых состоит из 8 разрядов действительной и 8 разрядов мнимой части. Порядок загрузки коэффициентов в секцию фильтра дан в описании команды WR_w_filter_RAM.

31	11	10	9	8	7	6	1	0
0100 0000 0000 0001 0000 0	1 0	1 1	PKCH_N		e/o			

Рисунок А.3 – Формат адреса команд обращения к памяти адресных генераторов

Поле PKCH_N (разряды адреса 6-1) – номер канала паковки. Каким образом каналы паковки подключены к каналам обработки показано на структурной схеме DSPPU (см. Рисунок 10.1).

Поле e/o (разряд адреса 0) – определяет блок RAM памяти для четного/нечетного кадра:

- 0 – для четного кадра,
- 1 – для нечетного кадра.

31	11	10	9	8	7	6	0
0100 0000 0000 0001 0000 0	1 0	1 1	OPC				

Рисунок А.4 – Формат адреса команд обращения к регистрам адресного генератора

Поле OPC (разряды адреса 6-0) – код операции команды обращения к блоку адресных генераторов.

Каждая команда обращения к регистрам имеет свой формат данных. Большинство команд записи обновляет значение только теневых регистров. Значения теневых регистров передаются в рабочие (основные) в момент запуска соответствующего канала. Команды чтения всегда позволяют прочитать только рабочие регистры.

Память адресных генераторов содержит начальные адреса областей памяти, в которые будут записаны кадры. Запись этих адресов следует производить парами команд WR_AG_adr_RAM, команды одной пары отличаются только битом e/o.

А.3 Команды записи

В описании команды записи RAM коэффициентов фильтров, в скобках, указываются возможные значения номера кластера, затем, в следующих скобках, возможные значения номера секции фильтра в кластере (номер RAM), затем, в третьих скобках, допустимые номера строк внутри RAM. Допустимы любые комбинации этих трёх параметров.

А.3.1 WR_w_filter_RAM (0-2) (0-3) (0-15)

Формат данных для команды WR_w_filter_RAM приведен ниже (см. Рисунок А.5).

31	0
Complex_filter_coefficients	

					ЮФКВ.431268.006РЭ			Лист
								289
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

Рисунок А.5 – Формат данных для команды WR_w_filter_RAM

Формат адреса для команды WR_w_filter_RAM приведен ниже (см. Рисунок А.6).

	9	8	7	6	5	4	3	2	1	0
0	0	Clu	Sec			dword				e/d

Рисунок А.6 – Формат адреса для команды WR_w_filter_RAM

В таблице ниже (Таблица А.2) приведено назначение полей данных для команды WR_w_filter_RAM

Таблица А.2 – Назначение полей данных для команды WR_w_filter_RAM

Биты	Название	Выполняемая функция
[31:0]	Complex_filter_coefficients	Задаёт значения двух комплексных коэффициентов фильтра соответствующего канала [7:0] – действительная часть одного коэффициента [15:7] – мнимая часть одного коэффициента [23:16] – действительная часть другого коэффициента [31:24] – мнимая часть другого коэффициента

Память коэффициентов одной секции фильтра состоит из 16 строк, каждая из которых содержит 4 комплексных коэффициента. Рисунок ниже (см. Рисунок А.7) показывает, как расположены комплексные коэффициенты секции фильтра внутри RAM.

imag 48	real 48	imag 32	real 32	imag 16	real 16	imag 0	real 0	dword 0
imag 49	real 49	imag 33	real 33	imag 17	real 17	imag 1	real 1	dword 1
imag 50	real 50	imag 34	real 34	imag 18	real 18	imag 2	real 2	dword 2
imag 51	real 51	imag 35	real 35	imag 19	real 19	imag 3	real 3	dword 3
imag 52	real 52	imag 36	real 36	imag 20	real 20	imag 4	real 4	dword 4
imag 53	real 53	imag 37	real 37	imag 21	real 21	imag 5	real 5	dword 5
imag 54	real 54	imag 38	real 38	imag 22	real 22	imag 6	real 6	dword 6
imag 55	real 55	imag 39	real 39	imag 23	real 23	imag 7	real 7	dword 7
imag 56	real 56	imag 40	real 40	imag 24	real 24	imag 8	real 8	dword 8
imag 57	real 57	imag 41	real 41	imag 25	real 25	imag 9	real 9	dword 9
imag 58	real 58	imag 42	real 42	imag 26	real 26	imag 10	real 10	dword 10
imag 59	real 59	imag 43	real 43	imag 27	real 27	imag 11	real 11	dword 11
imag 60	real 60	imag 44	real 44	imag 28	real 28	imag 12	real 12	dword 12
imag 61	real 61	imag 45	real 45	imag 29	real 29	imag 13	real 13	dword 13
imag 62	real 62	imag 46	real 46	imag 30	real 30	imag 14	real 14	dword 14
imag 63	real 63	imag 47	real 47	imag 31	real 31	imag 15	real 15	dword 15

Рисунок А.7 – Расположение комплексных коэффициентов секции фильтра внутри RAM

Поле dword адреса команды указывает на номер строки, поле e/o указывает, к старшей (1) или младшей (0) части строки производится данное обращение.

Необходимо соблюдать следующий порядок записи коэффициентов (каждая строка таблицы соответствует одной команде WR_w_filter_RAM).

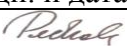
					ЮФКВ.431268.006РЭ				Лист
									290
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

Таблица А.3 – Последовательность записи коэффициентов в секцию фильтра

16-й коэфф. Мнимая часть	16-й коэфф. Действ.часть	0-й коэфф. Мнимая часть	0-й коэфф. Действ.часть	Данные первой команды первой пары
48-й коэфф. Мнимая часть	48-й коэфф. Действ.часть	32-й коэфф. Мнимая часть	32-й коэфф. Действ.часть	Данные второй команды первой пары
17-й коэфф. Мнимая часть	17-й коэфф. Действ.часть	1-й коэфф. Мнимая часть	1-й коэфф. Действ.часть	Данные первой команды второй пары
49-й коэфф. Мнимая часть	49-й коэфф. Действ.часть	33-й коэфф. Мнимая часть	33-й коэфф. Действ.часть	Данные второй команды второй пары
...	...	...	...	...
31-й коэфф. Мнимая часть	31-й коэфф. Действ.часть	15-й коэфф. Мнимая часть	15-й коэфф. Действ.часть	Данные первой команды шестнадцатой пары
63-й коэфф. Мнимая часть	63-й коэфф. Действ.часть	47-й коэфф. Мнимая часть	47-й коэфф. Действ.часть	Данные второй команды шестнадцатой пары

Пример: Надо загрузить коэффициенты фильтра секции С (номер 2) первого кластера целыми значениями от 0 до 64-х для мнимых частей, и целыми значениями от 0 до -64-х для действительных частей. Для этого необходимо послать в DSPPU 16 пар команд записи (Таблица А.4 показывает последовательность команд записи).

Таблица А.4 – Пример загрузки коэффициентов фильтра секции С первого кластера.

Адрес [10:0]	Данные [31:24]	Данные [23:16]	Данные [15:8]	Данные [7:0]
4c0h	16	-16	0	0
4c1h	48	-48	32	-32
4c2h	17	-17	1	-1
4c3h	49	-49	33	-33
...	...	...	...	...
4deh	31	-31	15	-15
4dfh	63	-63	47	-47

Данные приводятся в десятичном виде побайтово для наглядности. Младшие разряды адреса – в шестнадцатеричном виде.

А.3.2 WR_het_B / WR_FQ (0/1) (0-27)

В описании команд обращения к регистрам в скобках указываются номера регистров, а затем, в следующих скобках, допустимые значения поля CH_NMBR адреса команды. Косой чертой разделены неодинаковые значения для команд в группе, при этом, в списках, разделённых косой чертой, элементы с одинаковым порядком в списке соответствуют друг другу. Например, команде WR_het_B соответствует номер 0, команде WR_FQ соответствует номер 1, в обеих командах поле CH_NMBR может принимать значения от 0 до 27.

Формат данных для команд WR_het_B, WR_FQ приведен ниже (см. Рисунок А.8).

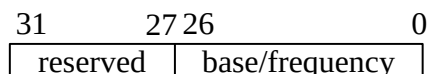


Рисунок А.8 – Формат данных для команд WR_het_B, WR_FQ

В таблице (Таблица А.5) приведено описание полей данных для команды WR_het_B, WR_FQ

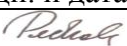
					ЮФКВ.431268.006РЭ				Лист
									291
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

Таблица А.5 – Описание полей данных для команд WR_het_B, WR_FQ

Биты	Название	Выполняемая функция
[31:27]	reserved	Зарезервированы.
[26:0]	base / frequency	Определяют значение «начальной фазы» / частоты гетеродина

Обе команды загружают 27-разрядные регистры управления гетеродином соответствующего канала – начальную фазу гетеродина и его частоту. Команда RD_het_base позволяет прочитать регистр начальной фазы. Команда RD_frequency не позволяет прочитать теневой регистр частоты гетеродина, а только основной регистр. Значение поля frequency выбирается в зависимости от требуемой частоты гетеродина f , выраженной в герцах, и частоты тактового сигнала данного блока следующим образом:

$$\text{code} = [f * 2^{27} / (F_{clk}/4)].$$

Начальное смещение фазы гетеродина φ , выраженной в радианах, в зависимости от кода в регистре B[26:0] и обратно приведены ниже:

$$\varphi = \text{code} * 2 * \pi / 2^{27},$$

$$\text{code} = [\varphi * 2^{27} / 2 * \pi],$$

A.3.3 WR_select (0Ah) (0-27)

Формат данных для команды WR_select приведен ниже (см. Рисунок А.9).

31	13	12	11	9	8	7	6	5	4	3	2	1	0
Reserved	ot	Ms	sin/cos	sel_tc	sel_cd								

Рисунок А.9 – Формат данных для команды WR_select

В таблице (Таблица А.6) приведено описание полей данных команды WR_select.

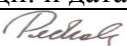
					ЮФКВ.431268.006РЭ					Лист
										292
Изм.	Лист	№ докум.	Подп.	Дата						
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата				
22755-2	 16.01.14		22755-1							

Таблица А.6 – Описание полей данных для команды WR_select

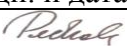
Биты	Название	Выполняемая функция
[31:13]	Reserved	Зарезервированы.
[12]	ot	1b – однокадровый режим включен. Канал обработает один кадр, затем метка переключит канал в исходное состояние, после чего канал будет остановлен. 0b – однокадровый режим выключен. Данное поле имеет значение только в случае, когда источником данных является ADC или входы SIGNx/MAGNx.
[11:9]	Ms	Выбор источника меток. В качестве источника меток для канала можно выбрать один из 7 источников (см. Рисунок 10.2): 000b – источник 0 – исходная метка интервального таймера; 001b – источник 1 – метка, прореженная в настраиваемом делителе; ... 101b – источник 5 – метка, прореженная в настраиваемом делителе; 110b, 111b – источник 6 – метка, прореженная в настраиваемом делителе. С взведением бита состояния Tstop в канале, поле Ms обнуляется.
[8:6]	sin/cos	Выбор типа преобразования фазы гетеродина в комплексные составляющие. 000b – sin/cos. Преобразование 6 старших разрядов гетеродина в 4 разряда sin/cos. 001b – (sin=0,cos=1), 010b – (sin=1,cos=0), 011b – (sin=0,cos=0). 1xxb – sin/cos. Преобразование 8 старших разрядов гетеродина (после накопления) в 6 разрядов sin/cos. 100b – с округлением to_-inf; 101b – с округлением to_+inf; 110b – с округлением to_nearest; 111b – с округлением to_nearest_even. Описание типов округления см. в п. 10.5.1.
[5:2]	sel_tc	Выбор источника генератора псевдослучайных последовательностей для канала: 0000b – выбирает значение генератора псевдослучайной последовательности 24-го канала, 0001b – выбирает значение генератора псевдослучайной последовательности 25-го канала, 0010b – выбирает значение генератора псевдослучайной последовательности 26-го канала, 0011b – выбирает значение генератора псевдослучайной последовательности 27-го канала, 0100b – выбирает значение внешнего генератора псевдослучайной последовательности 0, 0101b – выбирает значение внешнего генератора псевдослучайной последовательности 1, 0110b – выбирает значение внешнего генератора псевдослучайной последовательности 2, 0111b – выбирает значение внешнего генератора псевдослучайной последовательности 3, 1000b – 1011b – выбирает значение «0», 1100b – 1111b – выбирает значение «1».
[1:0]	sel_cd	Выбор интерфейса входного устройства в качестве источника данных для данного канала. 00b – выбирает интерфейс 0; 01b – выбирает интерфейс 1; 10b – выбирает интерфейс 2; 11b – выбирает интерфейс 3. Источник данных для интерфейса выбирается командой WR_ADC_control.

Примечание: Для внутренних генераторов псевдослучайных последовательностей (24-27) имеют значение только поля **ot**, **Ms**, **sel_cd** (определяется факт прихода данных, но сами данные не используются). Для каналов 0-23 действительны все поля.

Команда настраивает входные мультиплексоры канала, признак однокадрового режима и выбор используемой таблицы sin/cos с соответствующим режимом округления. Значение поля Ms при работе с программным потоком не существенно.

Запись настроек данной командой происходит *одноуровневым* образом, то есть теневые регистры отсутствуют. Если данная команда производит запись своих регистров во время обработки кадра, то произойдет выдача сигнала прерывания (INT_DSPPU_PRG). Прерывание можно замаскировать с помощью команды WR_Mask_trap.

Команда RD_het_control позволяет прочитать значения полей ot, Ms, sin/cos, sel_tc, sel_cd.

					ЮФКВ.431268.006РЭ			Лист
								293
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата		
22755-2		 16.01.14		22755-1				

A.3.4 WR_het_counter (2) (0-23)

Формат данных для команды WR_het_counter приведен ниже (см. Рисунок А.10).

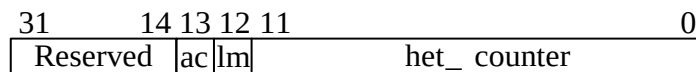


Рисунок А.10 – Формат данных для команды WR_het_counter

В таблице (Таблица А.7) приведено описание полей данных для команды WR_het_counter.

Таблица А.7 – Описание полей данных для команды WR_het_counter

Биты	Название	Выполняемая функция
[31:14]	Reserved	Зарезервированы.
[13]	ac	Выбирает режим накопления или прореживания в аккумуляторе канала, расположенном после гетеродина: 0h – режим прореживания (децимации); 1h – режим накопления
[12]	lm	Определяет влияние границы кадра на накопление/прореживание: 0h – по границе кадра накопление/прореживание начинается заново; 1h – граница кадра не прерывает процесс накопления/прореживания
[11:0]	het_counter	Период накопления/прореживания входного комплексного сигнала. Задается в прямом коде. Если 0, то накопления/прореживания не происходит.

Команда управляет блоком накопления, расположенным в конвейере DSPPU после гетеродина. Запись данной командой производится в теневые регистры.

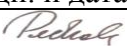
Данные, пришедшие в канал и преобразованные с учетом выбранного источника генератора псевдослучайных последовательностей, преобразуются в комплексные данные – действительная часть данных формируется через умножение на значение $\cos-\alpha$, мнимая часть на значение $\sin-\alpha$.

Если поле команды $\text{het_counter}=0$, то все входные комплексные данные проходят без изменений. Если $\text{het_counter}=1$, то поток данных становится реже в два раза, причем пропускаются данные идущие не вместе с меткой, а следующие (в случае накопления выдается сумма двух смежных комплексных отсчетов). Для последовательно возрастающих значений поля het_counter поток данных становится все реже, и в каждом накоплении будут участвовать все больше комплексных отсчетов. В случае, когда кадр кончился, а данные накопились не полностью – эти накопленные данные пропадают, если $\text{lm}=0$.

Накопление комплексных данных производится без потери точности, независимо от значения поля counter . Комплексные данные после накопления/прореживания передаются в нормализатор, где из них выделяется 8 смежных разрядов с учетом переполнения (нормализация и округления данных настраиваются командой WR_norm_h).

При настройке аккумулятора необходимо учитывать скорость выдачи данных в выходном потоке аккумулятора. Например, на кластер КИХ-фильтров данные через каждый канал могут подаваться со скоростью не более $F_{\text{clk}}/16$ МГц.

Команда RD_het_control позволяет прочитать текущие значения полей – ac , lm , het_count .

					ЮФКВ.431268.006РЭ			Лист
								294
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

A.3.5 WR_norm_h (3) (0-23), WR_norm_f / WR_norm_q (5/6) (0-11)

Формат данных для команды WR_norm приведен ниже (см. Рисунок A.11).

31	11	10	9	8	7	6	5	4	0
Reserved	R	delta	rnd	st	norm				

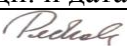
Рисунок A.11 – Формат данных для команды WR_norm

В таблице (Таблица A.8) приведено описание полей данных для команд WR_norm_h, WR_norm_f, WR_norm_q.

Таблица A.8 – Описание полей данных для команд WR_norm_h, WR_norm_f, WR_norm_q

Биты	Название	Выполняемая функция
[31:11]	Reserved	Зарезервированы.
[10]	R	Включает режим автоматической настройки поля norm. 0b – автоматическая подстройка нормализатора запрещена, 1b – автоматическая подстройка нормализатора разрешена.
[9:8]	delta	Управляет автоматической настройкой поля norm. Можно выбрать из следующих вариантов автоподстройки: 00b – norm = exp + 0; 01b – norm = exp + 1; 10b – norm = exp – 2; 11b – norm = exp – 1; Значение exp вычисляется устройством, как показано ниже и в п. 10.6. Если значение exp = 0, то вычитания не производится; если значение exp = 1, то вычитается не больше единицы.
[7:6]	rnd	Управляет режимом округления результата: 00b – to_–inf; 01b – to_+inf; 10b – to_zero; 11b – to_nearest_even. Описание типов округления см. в п. 10.5.1.
[5]	st	Управляет включением/выключением режима насыщения: 0h – насыщение выключено, 1h – насыщение включено.
[4:0]	norm	Выбирает, какие смежные 8 разрядов аккумулятора выбираются в качестве результата нормализации: 00h – выбираются разряды 1-0 из аккумулятора, хвост – 6 нулей в младших разрядах, 01h – выбираются разряды 2-0 из аккумулятора, хвост – 5 нулей в младших разрядах, ... 06h – выбираются разряды 7-0 из аккумулятора, ... 1fh – выбираются разряды 32-25 из аккумулятора. В режиме насыщения выходящие за указанные (в поле norm) пределы значения на входе нормализатора превращаются в предельные значения (127, -128) на выходе нормализатора. Подробно процесс нормализации описан в п. 10.6.

Команда WR_norm_h записывает теневые регистры управления нормализатором, расположенным в конвейере указанного канала после аккумулятора гетеродина. Данной команде записи соответствует команда чтения RD_norm_h, которая позволяет прочитать основные регистры.

					ЮФКВ.431268.006РЭ			Лист
								295
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Команда WR_norm_f записывает теневые регистры управления нормализатором, расположенным в конвейере указанного канала после кластера фильтров. Данной команде записи соответствует команда чтения RD_norm_f, которая позволяет прочитать основные регистры.

Команда WR_norm_q записывает теневые регистры управления нормализатором, расположенным в конвейере указанного канала после аккумулятора квадратора. Данной команде записи соответствует команда чтения RD_norm_q, которая позволяет прочитать основные регистры.

Все три команды записи работают аналогично. В текущем кадре доступна информация о результатах работы нормализатора в предыдущем кадре, поэтому нормализатором можно управлять только через кадр, если не использовать режим автоматической подстройки.

Аппаратно в нормализаторе вычисляется для каждого комплексного отсчета (и для мнимой, и для действительной составляющей) номер разряда (нулевой разряд расположен справа), в котором находится первая слева единица для положительных данных и первый слева ноль для отрицательных данных. Затем определяется большее из всех значений и в результате к концу кадра формируется наибольшее значение экспоненты (exp) для всех комплексных отсчетов каждого кадра, которое доступно по чтению, то есть определяется порядок максимального значения по абсолютной величине.

В режиме автоматической настройки полученное таким образом значение exp со смещением на delta помещается в теневой регистр norm, замещая предыдущее значение. Результат работы нормализатора в предыдущем кадре сразу оказывает влияние на обработку следующего кадра. Режим автоматической настройки можно установить для каждого канала соответствующей командой WR_norm_h, WR_norm_f, WR_norm_q (бит R).

Соответствующие команды чтения (RD_norm_h, RD_norm_f, RD_norm_q) позволяют определить текущее значение порядка (поле norm), с которым работает соответствующий (по конвейеру) нормализатор.

A.3.6 WR_filter_mask (4) (0-3/4-7/8-11)

Формат данных для команды WR_filter_mask приведен ниже (см. Рисунок A.12).

31	22	21	20	19	16	15	12	11	8	7	4	3	0
Reserved	h3	h1	v3	v2	v1	v0	filter_mask3	filter_mask2	filter_mask1	filter_mask0			

Рисунок A.12 – Формат данных для команды WR_filter_mask

В таблице (Таблица A.9) приведено описание полей данных для команды WR_filter_mask.

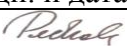
					ЮФКВ.431268.006РЭ					Лист
										296
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата			
22755-2	 16.01.14		22755-1							

Таблица А.9 – Описание полей данных для команды WR_filter_mask

Биты	Название	Выполняемая функция
[31:22]	Reserved	Зарезервированы.
[21:16]	filter_config (h3, h1, v3, v2, v1, v0)	Задаёт, как секции кластера фильтров (D,C,B,A) соединяются между собой и как поступают на эти секции каналные комплексные данные. Каким образом биты h3, h1, v3, v2, v1, v0 управляют соединением секций кластера, показано на схеме кластера фильтров (см. Рисунок 10.10). При записи h1=1 обязательно следует записать v1=0, при записи h3=1 обязательно следует записать v3=0. 001111b – значение после сброса, xx0000b – фильтр выключен.
[15:12]	filter_mask3	Производит выбор секций, данные с которых будут накапливаться в аккумуляторе 3-го канала данного кластера. Значение 1 в бите поля включает накопление из соответствующей секции: [15] – D, [14] – C, [13] – B, [12] – A.
[11:8]	filter_mask2	Производит выбор секций, данные с которых будут накапливаться в аккумуляторе 2-го канала данного кластера. Значение 1 в бите поля включает накопление из соответствующей секции: [11] – D, [10] – C, [9] – B, [8] – A.
[7:4]	filter_mask1	Производит выбор секций, данные с которых будут накапливаться в аккумуляторе 1-го канала данного кластера. Значение 1 в бите поля включает накопление из соответствующей секции: [7] – D, [6] – C, [5] – B, [4] – A.
[3:0]	filter_mask0	Производит выбор секций, данные с которых будут накапливаться в аккумуляторе 0-го канала данного кластера. Значение 1 в бите поля включает накопление из соответствующей секции: [3] – D, [2] – C, [1] – B, [0] – A.

Конфигурирование одного кластера фильтров производится *одной* командой WR_filter_mask – значение поля CH_NMBR в адресе команды может быть любым из четырёх номеров каналов, подключенных к данному кластеру (см. Рисунок 10.1).

Команда WR_filter_mask производит запись в теньевые регистры. Передача настроек в основные регистры произойдет, если придет команда пуска хотя бы в *один* из каналов кластера (WR_Start с соответствующим битом в маске каналов), а затем метка.

Поскольку секции фильтров содержат большое количество данных и от каждого из них зависит результат, то при смене настроек во время работы фильтра возможно искажение результата.

Все вычисления в кластере фильтров производятся без потери точности, для уменьшения разрядности данных используются блоки нормализации, расположенные в конвейере DSPPU после кластера фильтров.

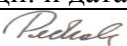
Команда RD_filter_control позволяет прочитать текущие значения полей h3, h1, v3, v2, v1, v0, filter_mask3, filter_mask2, filter_mask1, filter_mask0.

Пример: два фильтра порядка 64 с 16-разрядными коэффициентами можно построить для 0-го канала и 1-го канала одновременно, задавая filter_mask0=0011b и filter_mask1=1100b, а filter_config данного кластера равным 110101b.

Пример: фильтр порядка 128 с 16-разрядными коэффициентами можно построить для 0-го канала, задавая filter_mask0=1111b и filter_config данного кластера равным 110001b.

Пример: фильтр порядка 128 с 16-разрядными коэффициентами можно построить для 1-го канала, задавая filter_mask1=1111b и filter_config данного кластера равным 110100b.

Пример: четыре фильтра порядка 64 с 8-разрядными коэффициентами можно построить для 0-го, 1-го, 2-го и 3-го каналов одновременно, задавая filter_mask0=0001b, filter_mask1=0010b, filter_mask2=0100b, filter_mask3=1000b, а filter_config данного кластера равным 001111b.

					ЮФКВ.431268.006РЭ			Лист
								297
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

A.3.7 WR_quad_counter (0Ch) (0-11)

Формат данных для команды WR_quad_counter приведен ниже (см. Рисунок А.13).

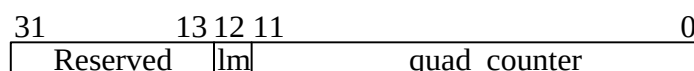


Рисунок А.13 – Формат данных для команды WR_quad_counter

В таблице (Таблица А.10) приведено описание полей данных для команды WR_quad_counter.

Таблица А.10 – Описание полей данных для команды WR_quad_counter

Биты	Название	Выполняемая функция
[31:13]	Reserved	Зарезервированы.
[12]	lm	Определяет влияние границы кадра на накопление: 0h – по границе кадра накопление начинается заново, 1h – граница кадра не оказывает влияние на накопление.
[11:0]	quad_counter	Период накопления входного комплексного сигнала. Задается в прямом коде. Если 0, то накопления не происходит.

Команда WR_quad_counter управляет блоком накопления, расположенным в конвейере DSPPU после блока квадраторов. Запись данной командой производится в теньевые регистры.

Если поле команды quad_counter=0, то все входные комплексные данные будут проходить без изменений. Если quad_counter=1, то поток данных становится реже в два раза. Для других возрастающих значений поля quad_counter в каждом накоплении будут участвовать все больше комплексных отсчетов.

Накопление комплексных данных осуществляется без потери точности, независимо от значения поля counter.

Команда RD_quad_control позволяет прочитать текущие значения полей: lm, quad_counter.

A.3.8 WR_pack (7) (0-23)

Формат данных для команды WR_pack приведен ниже (см. Рисунок А.14).

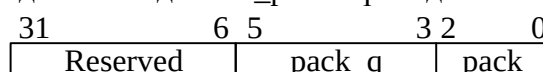


Рисунок А.14 – Формат данных для команды WR_pack

В таблице (Таблица А.11) приведено описание полей данных для команды WR_pack.

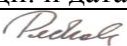
					ЮФКВ.431268.006РЭ				Лист
									298
Изм.	Лист	№ докум.	Подп.	Дата					
Инва.№подл.	Подп. и дата		Взам.инв.№		Инва.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

Таблица А.11 – Описание полей данных для команды WR_pack

Биты	Название	Выполняемая функция
[31:6]	Reserved	Зарезервированы.
[5:3]	pack_q	Выбирает режим упаковки данных с выхода блока квадраторов. Данное поле имеет значение только для каналов с 0 по 11 (поле адреса CH_NMBR 0-11), так как только эти каналы снабжены квадраторами. Описание различных способов паковки дано в 10.9. [5] – выбирает, как будет паковаться комплексный сигнал: 1b – действительная и мнимая части пакуются в разных 64-разрядных словах, 0b – действительная и мнимая части пакуются в одном 64-разрядном слове. [4:3] – выбирает разрядность упакованных данных: 00b – упаковка выключена – не происходит записи в память, 01b – 2 бита, 10b – 4 бита, 11b – 8 бит.
[2:0]	pack	Выбирает режим упаковки данных канала, не прошедших через квадраторы. Описание различных способов паковки дано в 10.9. [2] – выбирает, как будет паковаться комплексный сигнал: 1b – действительная и мнимая части пакуются в разных 64-разрядных словах, 0b – действительная и мнимая части пакуются в одном 64-разрядном слове. [1:0] – выбирает разрядность упакованных данных: 00b – упаковка выключена – не происходит записи в память, 01b – 2 бита, 10b – 4 бита, 11b – 8 бит.

Внимание! В поле адреса CH_NMBR команды WR_pack используются номера каналов обработки, а не каналов паковки.

Команда WR_pack управляет блоками упаковки данных, поступающих из каналов (только PACK_UNIT из описания в п. 10.9). Запись данной командой производится в теневые регистры.

Входные данные для блоков PACK_UNIT – всегда комплексные данные с 8-разрядными действительной и мнимой частью. Если настроенная разрядность упаковки составляет 8 бит, то данные записываются в память без потери точности. Когда командой задана разрядность 2 или 4, то из входного байта берутся, соответственно, старшие два или четыре бита.

Текущие значения полей pack_q и pack можно прочитать командой RD_quad_control.

A.3.9 WR_TCG / WR_TM / WR_TIS (2/5/6) (24-27)

Формат данных для команд WR_TCG, WR_TM, WR_TIS приведен ниже (см. Рисунок А.15).

31	25	24	0
Reserved	length/mask/significance		

Рисунок А.15 – Формат данных для команд WR_TCG, WR_TM, WR_TIS

В таблице (Таблица А.12) приведено описание полей данных для команд WR_TCG, WR_TM, WR_TIS.

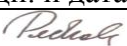
					ЮФКВ.431268.006РЭ			Лист
								299
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Таблица А.12 – Описание полей данных для команд WR_TCG, WR_TM, WR_TIS

Биты	Название	Выполняемая функция
[31:25]	Reserved	Зарезервированы.
[24:0]	length(TCG)/maskI/significance(TIS)	length – определяет длину последовательности генератора псевдослучайной последовательности; чтобы задать число слов, равное N, необходимо задать число (-N) в дополнительном коде. Mask – определяет, какие разряды сдвигового регистра используются для формирования сигнала обратной связи сдвигового регистра. Significance – определяет начальное значение сдвигового регистра генератора T-кодов, устанавливается в начале кадра и по истечении счетчика длины (если разрешено).

Все три команды загружают свои теневые регистры. Передача настроек в основные регистры произойдет, если придет команда пуска в канал (WR_Start с соответствующим битом в маске каналов), а затем метка.

Регистр длины используется для загрузки счетчика длины последовательности, перенос из которого определяет момент новой загрузки сдвигового регистра, формирующего внутренний сигнал code, значением из регистра significance. Увеличение значения счетчика длины на 1 (как и подвижка сдвигового регистра) происходит по переднему фронту старшего разряда гетеродина в своем канале.

Регистр маски используется в формировании значения в обратной связи сдвигового регистра, выбирая какие разряды сдвигового регистра использовать в свертке по модулю два, полученная свертка инвертируется 3-м разрядом регистра TCR и полученный сигнал подается в обратную связь сдвигового регистра.

Прочитать можно только значение регистра длины последовательности length на момент прихода последней метки – командой RD_TCGR.

А.3.10 WR_TCR (4) (24-27)

Формат данных для команды WR_TCR приведен ниже (см. Рисунок А.16).

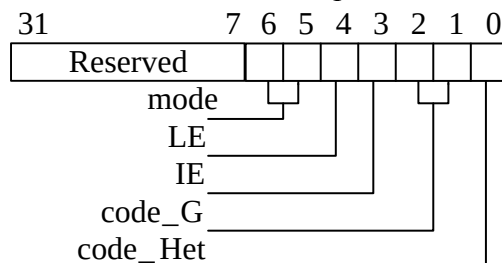


Рисунок А.16 – Формат данных для команды WR_TCR

В таблице (Таблица А.13) приведено описание полей данных для команды WR_TCR.

					ЮФКВ.431268.006РЭ				Лист
									300
Изм.	Лист	№ докум.	Подп.	Дата					
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата			
22755-2	<i>Редко</i> 16.01.14		22755-1						

Таблица А.13 – Описание полей данных для команды WR_TCR

Биты	Название	Код	Выполняемая функция
[6:5]	mode	00b, 01b	Режим останова (вырабатывается code сигнал, равный 0).
		10b	Множественный запуск.
		11b	Однократный запуск (до первого истечения счетчика длины последовательности).
[4]	LE	0b	Не используется счетчик длины последовательности (длина последовательности бесконечно велика).
		1b	Используется сигнал переноса счетчика длины последовательности.
[3]	IE	0/1b	Не инвертируется/инвертируется значение в обратной связи сдвигового регистра.
[2:1]	code_G	00b	сигнал code всех каналов проходит на следующую ступень формирования напрямую.
		01b	сигнал code 0-го канала проходит на следующую ступень формирования после свертки по модулю два с сигналом code 1-го канала; сигнал code 1,2,3-го каналов проходит на следующую ступень формирования после свертки по модулю два с сигналом code 0-го канала.
		10b	сигнал code 0,1-го каналов проходит на следующую ступень формирования после свертки по модулю два с сигналом code 2-го канала; сигнал code 2,3-го каналов проходит на следующую ступень формирования после свертки по модулю два с сигналом code 1-го канала.
		11b	сигнал code 3-го канала проходит на следующую ступень формирования после свертки по модулю два с сигналом code 2-го канала; сигнал code 0,1,2-го каналов проходит на следующую ступень формирования после свертки по модулю два с сигналом code 3-го канала.
[0]	code_Het	1/0b	Разрешение/запрет старшему разряду регистра текущей фазы гетеродина (26-й разряд) смежного канала суммироваться по модулю два с сигналом code предшествующей фазы формирования сигнала code. 24 каналу соответствует 20 канал; 25 каналу соответствует 21 канал; 26 каналу соответствует 22 канал; 27 каналу соответствует 23 канал.

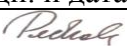
Схема, которой управляет данная команда, приведена в описании кластера генераторов псевдослучайных последовательностей (см. п. 10.4, Рисунок 10.5).

Разряды [2:0] используются для управления формированием сигнала code с использованием генераторов других каналов и гетеродинов выбранных каналов.

Данная команда записывает теневой регистр. Команды считывания этих полей не существует.

По метке после команды WR_Start (с соответствующими битами в маске каналов) запускается гетеродин канала генератора и теневые регистры переписываются в основные. Старший разряд гетеродина каждый раз по своему переднему фронту продвигает сдвиговый регистр (на входе сдвигового регистра формируется значение, исходя из его содержимого, регистра маски и 3-го разряда регистра TCR) и инкрементирует счетчик длины последовательности (TCR регистр должен разрешать такие действия).

По сигналу переноса счетчика длины перезаписывается содержимое сдвигового регистра (из регистра significance) и цикл повторяется, до тех пор пока не придет команда WR_Stop (или изменится содержимое регистра TCR соответствующим образом – например, установится в ноль 6-й разряд TCR) и затем придет метка. Перед передачей сигналов code в другие каналы или на внешние выходы микросхемы сигналы генератора псевдослучайных последовательностей могут быть смешаны по модулю два с

					ЮФКВ.431268.006РЭ			Лист
								301
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

сигнал code 24 канала свертывается по модулю два со старшим разрядом гетеродина 20 канала, сигнал code 25 канала свертывается со старшим разрядом гетеродина 21 канала, сигнал code 26 канала свертывается со старшим разрядом гетеродина 22 канала, сигнал code 27 канала свертывается со старшим разрядом гетеродина 23 канала, разрешением свертки управляет 0-й разряд TCR.

Формат данных для команды WR_ADC_control приведен ниже (см. Рисунок A.17).

31	19	18	17	16	15	14	12	11	8	7	6	5	4	3	2	1	0
Reserved				dk	d/i	un	pk	Ms	pack		adc	pack		sel3	sel2	sel1	sel0

В таблице (Таблица A.15) приведено описание полей данных для команды WR ADC control.

Команда осуществляет настройку входного устройства DSPPU, включает блоки упаковки действительных данных, а также может включать принудительную генерацию ошибок контрольной суммы.

					ЮФКВ.431268.006РЭ	Лист	
						302	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

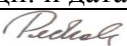
Таблица А.14 – Описание полей данных для команды WR_ADC_control

Биты	Название	Выполняемая функция
[31:19]	Reserved	Зарезервированы.
[18]	dk	Включает внесение ошибок в контрольную сумму данных по модулю 3. 1b – вносить ошибки в контрольную сумму, 0b – не вносить ошибок контрольной суммы.
[17]	d/i	Производит выбор между двумя внутренними источниками данных. 1b – DMA, 0b – командный поток (команда WR_ADC_ins). Данный бит имеет значение только если в одном из полей sel3, sel2, sel1, sel0 выбран внутренний источник данных.
[16:15]	un_pk	Выбирает тип распаковки упакованных данных из DMA. 11b – 16 бит преобразуются в 12 отбрасыванием старших 4 разрядов, 10b – 8 бит преобразуются в 12 расширением знака, 01b – 4 бита преобразуются в 12 расширением знака, 00b – 2 бита в кодировке sgmg преобразуются в 12 в дополнительном коде по правилу, показанному на схеме входного устройства (см. Рисунок 10.2).
[14:12]	Ms_pack	Выбор источника меток для блока ADC_PACK_UNIT (один на все 4 канала паковки). В качестве источника меток для канала можно выбрать один из 7 источников (см. Рисунок 10.2): 000b – источник 0 – исходная метка интервального таймера, 001b – источник 1 – метка, прореженная в настраиваемом делителе, ... 101b – источник 5 – метка, прореженная в настраиваемом делителе, 110b, 111b – источник 6.
[11:8]	adc_pack	Включает каналы паковки ADC_PACK_UNIT: [11] – включает канал паковки 39, подключенный к интерфейсу 3 входного устройства, [10] – включает канал паковки 38, подключенный к интерфейсу 2 входного устройства, [9] – включает канал паковки 37, подключенный к интерфейсу 1 входного устройства, [8] – включает канал паковки 36, подключенный к интерфейсу 0 входного устройства.
[7:0]	sel3, sel2, sel1, sel0	Производит выбор источника входных данных для каждого из выходных интерфейсов входного устройства. Sel3 выбирает источник данных для интерфейса 3 входного устройства, sel2 – для интерфейса 2, sel1 – для интерфейса 1, sel0 – для интерфейса 0. Значения selx: 00b или 01b – выбирает внутренний источник, 10b – выбирает внешние входы SIGNx/MAGNx, 11b – выбирает ADC.

Если в каком-либо из полей sel3, sel2, sel1, sel0 выбран внутренний источник данных, то тип этого источника (DMA или WR_ADC_ins) определяется битом d/i. То есть нельзя одновременно для одного из интерфейсов задать в качестве источника DMA, а для другого – WR_ADC_ins.

Если в качестве источника данных для выходного интерфейса входного устройства выбраны входы SIGNx/MAGNx, то на каждый интерфейс можно подать сигналы только с входов SIGNx/MAGNx, имеющих такой же номер. То есть для интерфейса 0 источником сигналов sgmg могут быть только входы SIGN0/MAGN0, для интерфейса 1 – SING1/MAGN1, для интерфейса 2 – SING2/MAGN2, для интерфейса 3 – SING3/MAGN3.

Если в качестве источника данных для выходного интерфейса входного устройства выбраны входы ADC, то на каждый интерфейс можно подать сигналы только с блока ADC с соответствующим номером. То есть для интерфейса 0 источником оцифрованных сигналов может быть только ADC0, для интерфейса 1 – ADC1, для интерфейса 2 – ADC2, для интерфейса 3 – ADC3.

					ЮФКВ.431268.006РЭ			Лист
								303
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

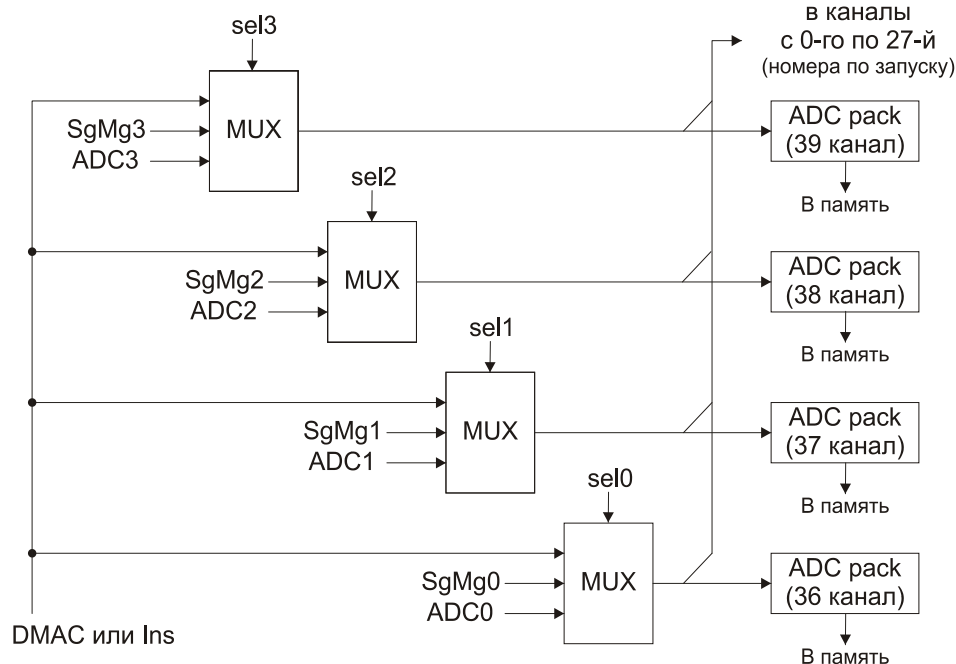


Рисунок А.18 – Схема коммутации входных потоков данных

На рисунке выше (см. Рисунок А.18) показано соответствие источников данных, интерфейсов выходного устройства, полей selx и каналов паковки 36-39.

Запись sel3, sel2, sel1, sel0 полей команды производится в теньевые регистры, которые переписываются в основные после прихода команды WR_Start с битом а (28 разряд), а затем метки (здесь имеется в виду только метка от интервального таймера – по программной метке перезаписи основных регистров не произойдет). Запись d/i, un_pk полей команды производится сразу, без ожидания прихода метки. Запись Ms_pack, adc_pack полей команды производится сразу для тех каналов паковки, которые настроены на внутренний источник данных, иначе запись откладывается до момента прихода команды WR_Start с битом а (28 разряд), а затем метки от интервального таймера.

Текущие значения всех полей, записываемых данной командой, можно прочитать командой RD_ADC_control.

Запрещается изменять настройки источников данных для интерфейсов, используемых работающими каналами (также и каналами паковки). Необходимо остановить канал, прежде чем изменять источник данных для него.

А.3.12 WR_ADC_ins (1) (28)

Формат данных для команды WR_ADC_ins приведен ниже (см. Рисунок А.19).

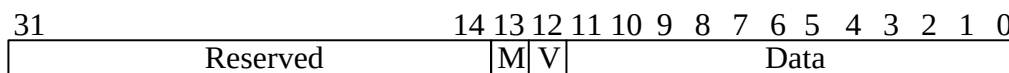


Рисунок А.19 – Формат данных для команды WR_ADC_ins

В таблице (Таблица А.15) приведено описание полей данных для команды WR_ADC_ins.

					ЮФКВ.431268.006РЭ	Лист
						304
Изм.	Лист	№ докум.	Подп.	Дата		
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата
22755-2		 16.01.14		22755-1		

Таблица A.15 – Описание полей для команды WR_ADC_ins

Биты	Название	Выполняемая функция
[31:14]	Reserved	Зарезервированы.
[13]	M	Наличие программной метки в данной команде.
[12]	V	Наличие данных в данной команде.
[11:0]	Data	Данные (в дополнительном коде, имитируя данные от ADC).

Команда введена, чтобы простыми программными средствами проверить работу DSPPU. После сброса (аппаратного или программного) входное устройство настраивается на режим приёма данных из командного потока.

Выдача команды WR_ADC_ins с битом M=1 эмулирует приход метки, выдача команды WR_ADC_ins с битом V=1 передаёт DSPPU одно данное Data такого же типа, что и от ADC.

Чтобы данные, передаваемые этой командой, поступали в канал обработки, необходимо предварительно настроить его и запустить командой WR_Start.

При передаче данных блоку DSPPU из программного потока всегда используется однокадровый режим.

A.3.13 WR_ADC_delta (2) (28)

Формат данных для команды WR_ADC_delta приведен ниже (см. Рисунок A.20).



Рисунок A.20 – Формат данных для команды WR_ADC_delta

В таблице (Таблица A.16) приведено описание полей данных для команды WR_ADC_delta.

Таблица A.16 – Описание полей данных для команды WR_ADC_delta

Биты	Название	Выполняемая функция
[31:3]	Reserved	Зарезервированы.
[1:0]	delta	Задаёт величину подстройки синхронизации во входном устройстве.

Процесс синхронизации между источником сигнала с частотой Fclk/4 и блоком DSPPU, который работает на частоте Fclk МГц, описан в п. 10.2. Поле delta позволяет сдвигать момент фиксации входных данных от 0 до 3 тактов DSPPU. Успешность подстройки проверяется командой RD_ADC_check.

A.3.14 WR_ADC_calibration (4-7) (28)

Формат данных для команды WR_ADC_calibration приведен ниже (см. Рисунок A.21).

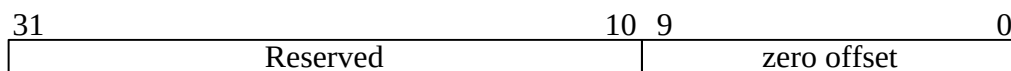


Рисунок A.21 – Формат данных для команды WR_ADC_calibration

В таблице (Таблица A.17) приведено описание полей данных для команды WR_ADC_calibration.

					ЮФКВ.431268.006РЭ				Лист
									305
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

Таблица A.17 – Описание полей данных для команды WR_ADC_calibration

Биты	Название	Выполняемая функция
[31:10]	Reserved	Зарезервированы.
[9:0]	zero_offset	Задаёт постоянное смещение для 12-разрядных данных с выходов каждого из ADC_0, ADC_1, ADC_2, ADC_3. Поле zero_offset задается в дополнительном коде.

Данные от аналого-цифрового преобразователя можно программно смещать на некоторую постоянную величину (zero_offset), которая задается в дополнительном коде. Коррекция данных производится с насыщением в отрицательную и положительную стороны.

A.3.15 WR_Ms_counter (1-7) (29)

Формат данных для команды WR_Ms_counter приведен ниже (см. Рисунок A.22).

31	13 12 11	0
Reserved	e/o	Ms_count (1-6)

Рисунок A.22 – Формат данных для команды WR_Ms_counter

В таблице (Таблица A.18) приведено описание полей данных для команды WR_Ms_counter. Данная команда устанавливает значения в регистры делителей меток входного устройства DSPPU (см. Рисунок 10.2). Команда с полем адреса REG=1 записывает поля e/o и Ms_count в делитель 1, REG=2 – в делитель 2 и так далее. Команда с полем адреса REG=7 записывает только бит e/o – этот бит используется каналами, для которых источником данных является программный поток.

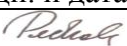
Таблица A.18 – Описание полей данных для команды WR_Ms_counter

Биты	Название	Выполняемая функция
[31:13]	Reserved	Зарезервированы.
[12]	e/o	Задаёт чётность номера текущего временного интервала. От этого бита зависит, в какую из двух (чётную или нечётную) выделенных областей памяти записываются текущие данные кадра. Биты e/o делителей меток (с 1-го по 6-й) инвертируются с приходом каждой метки. Бит e/o, доступный для записи командой с полем адреса REG=7, не меняется.
[11:0]	Ms_count (1-6)	Задаёт значение счетчика для прореживания меток интервального таймера: 0 – прореживания нет, 1 – результирующая метка в 2 раза реже метки от интервального таймера, 2 – результирующая метка в 3 раза реже метки от интервального таймера, ... 4095 – результирующая метка в 4096 раз реже метки от интервального таймера.

Существует одна особенность при перезагрузке счетчиков меток. Новое значение в рабочий регистр счетчика не записывается до тех пор, пока он не досчитал до нуля.

Нельзя производить запись в регистры делителя, если его метки используются каким-либо запущенным каналом.

Если в бит e/o делителя меток записать значение 0 и выдать команду WR_Start запуска канала, использующего данный делитель меток, то данные кадра будут запи-

					ЮФКВ.431268.006РЭ			Лист
								306
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

саны в нечётную выделенную область памяти, так как команда задаёт чётность *текущего* интервала, а канал начнёт работу в *следующем*.

Прочитать значения полей e/o и Ms_count можно командой RD_Ms_counter.

A.3.16 WR_Start / WR_Start_exp_h / WR_Start_exp_f / WR_Start_exp_q (1Ch/1Dh/1Eh/1Fh) (31)

Формат данных для команд WR_Start, WR_Start_exp_h, WR_Start_exp_f, WR_Start_exp_q приведен ниже (см. Рисунок A.23).

31	30	29	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0
b	f	ph	a		cl_6		cl_5		cl_4		cl_3		cl_2		cl_1		cl_0

Рисунок A.23 – Формат данных для команд WR_Start, WR_Start_exp_h, WR_Start_exp_f, WR_Start_exp_q

В таблице (Таблица A.19) приведено описание полей данных для команд WR_Start, WR_Start_exp_h, WR_Start_exp_f, WR_Start_exp_q.

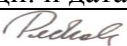
Таблица A.19 – Описание полей данных для команд WR_Start, WR_Start_exp_h, WR_Start_exp_f, WR_Start_exp_q

Биты	Название	Выполняемая функция
[31]	b	1b – начальная фаза гетеродина, записанная командой WR_het_B, прибавится к текущей фазе (после прихода метки), 0b – сложения не произойдет.
[30]	f	1b – значение частоты гетеродина, записанное командой WR_FQ, перепишется из теневого регистра в рабочий регистр (после прихода метки), 0b – перезаписи не будет.
[29]	ph	1b – обнулится текущая фаза гетеродина перед занесением нового значения (после прихода метки), 0b – обнуления не произойдет, коррекция будет произведена со старым значением текущей фазы гетеродина.
[28]	a	1b – значения некоторых полей, записанные командой WR_ADC_control, перепишутся из теневых в основные регистры (после прихода метки), 0b – сохранятся предыдущие значения.
[27:0]	cl_6, cl_5,cl_4,cl_3, cl_2,cl_1,cl_0	Маска пуска каналов 27-0. 1b – пуск соответствующего канала (после прихода метки), 0b – сигнал пуска в канал не передается.

Все четыре команды позволяют запускать каналы по маске. Команды WR_Start_exp_h, WR_Start_exp_f, WR_Start_exp_q, кроме пуска канала, устанавливают на один временной интервал режим автоматической настройки в соответствующих (после гетеродина, после фильтра, после квадратора) нормализаторах. Если сразу требуется установить режим автоматической настройки в двух и более нормализаторах для следующего интервала, то этого сделать с помощью команд пуска нельзя, так как нельзя выдавать более одной команды пуска в один канал за один временной интервал.

Существует еще одно ограничение – нельзя после команды пуска производить настройку канала – следует дождаться метки.

Биты b, f, ph определяют фазу гетеродина на момент начала временного интервала. Допустимы все 8 комбинаций этих битов, однако, только некоторые из них подходят для реализации определённой функциональности:

					ЮФКВ.431268.006РЭ				Лист
									307
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14		22755-1					

b=0, f=0, ph=1 – в следующем временном интервале гетеродин начнёт работу с фазы 0,
b=1, f=0, ph=1 – в следующем временном интервале гетеродин начнёт работу с фазы, записанной предварительно командой WR_het_B,
b=0, f=1, ph=0 – функция фазы от времени *не потерпит разрыв* при переходе в другой временной интервал,
b=1, f=1, ph=0 – на границе интервалов произойдёт моментальный сдвиг фазы на значение, предварительно записанное командой WR_het_B (например, если было записано 4000000h, то есть ровно половина максимального значения, то фаза гетеродина сдвинется на 180°).

A.3.17 WR_Stop / WR_Stop_ms (1Ah/1Bh) (31)

Формат данных для команд WR_Stop, WR_Stop_ms приведен ниже (см. Рисунок A.24).

31	29	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0
Reserved	pk		cl_6		cl_5		cl_4		cl_3		cl_2		cl_1		cl_0	

Рисунок A.24 – Формат данных для команд WR_Stop, WR_Stop_ms

В таблице (Таблица A.20) приведено описание полей данных для команд WR_Stop, WR_Stop_ms.

Таблица A.20 – Описание полей данных для команд WR_Stop, WR_Stop_ms

Биты	Название	Выполняемая функция
[31:29]	Reserved	Зарезервированы.
[28]	pk	1b – останов каналов паковки 36-39 (после прихода метки), 0b – сигнал останова в каналы паковки 36-39 не передается.
[27:0]	cl_6, cl_5,cl_4,cl_3, cl_2,cl_1,cl_0	Маска останова для каналов 27-0. 1b – останов соответствующего канала (после прихода метки), 0b – сигнал останова в канал не передается.

Обе команды осуществляют останов каналов и каналов паковки по маске. Команда WR_Stop_ms останавливает каналы по ближайшей метке интервального таймера, в отличие от WR_Stop, которая осуществляет останов по метке в канале.

A.3.18 WR_Mask_trap (18) (31)

Формат данных для команды WR_Mask_trap приведен ниже (см. Рисунок A.25).

31	29	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0
Reserved	a		cl_6		cl_5		cl_4		cl_3		cl_2		cl_1		cl_0	

Рисунок A.25 – Формат данных для команды WR_Mask_trap

В таблице (Таблица A.21) приведено описание полей данных для команды WR_Mask_trap.

					ЮФКВ.431268.006РЭ	Лист 308		
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата			Взам.инв.№	Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14			22755-1			

Таблица A.21 – Описание полей данных для команды WR_Mask_trap

Биты	Название	Выполняемая функция
[31:29]	Reserved	Зарезервированы.
[28]	a	1b – разрешается выдача запроса на прерывание из каналов паковки 36-39, 0b – прерывание от каналов паковки 36-39 не выдается (состояние после начальной установки).
[27:0]	cl_6, cl_5,cl_4,cl_3, cl_2,cl_1,cl_0	Задают значение маски для каналов 27-0. 1b – разрешается выдача запроса на прерывание из соответствующих каналов, 0b – прерывание от канала не выдается (состояние после начальной установки).

Команда маскирует прерывания по причине обнаружения ошибок (INT_DSPPU_PRG и INT_DSPPU_HRD) от каналов по маске одноуровневым образом без ожидания прихода метки. Данная команда не маскирует выдачу признаков ошибок в заголовок кадра. Записанные разряды маски можно прочитать командами RD_Channel_check (бит M) и RD_ADC_check (бит M).

A.3.19 WR_Reset (0) (31)

Формат данных для команды WR_Reset приведен ниже (см. Рисунок A.26).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															

Рисунок A.26 – Формат данных для команды WR_Reset

Команда осуществляет программный сброс аппаратуры DSPPU, который длится 160 тактов (на частоте работы DSPPU). DSPPU устанавливается в начальное состояние. Каналы остановлены, все памяти и регистры сброшены. Состояние идентично состоянию после включения питания.

A.4 Команды чтения

A.4.1 RD_het_base / RD_frequency / RD_current (0/1/8) (0-27)

Формат данных для команд RD_het_base, RD_frequency, RD_current приведен ниже (см. Рисунок A.27).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
het_base / frequency / current																															

Рисунок A.27 – Формат данных для команд RD_het_base, RD_frequency, RD_current

В таблице (Таблица A.22) приведено описание полей данных для команд RD_het_base, RD_frequency, RD_current.

					ЮФКВ.431268.006РЭ						Лист
											309
Изм.	Лист	№ докум.	Подп.	Дата							
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.		Подп. и дата				
22755-2	<i>Решет</i> 16.01.14		22755-1								

Таблица A.22 – Описание полей данных для команд RD_het_base, RD_frequency, RD_current

Биты	Название	Выполняемая функция
[31]	tt	Бит состояния Ttstop.
[30]	t	Бит состояния Tstop.
[29]	r	Бит состояния Tstart.
[28:27]	k	Разряды свертки по модулю 3 значения поля het_base / frequency / current.
[26:0]	het_base	Значение начальной фазы гетеродина.
	Frequency	Значение частоты гетеродина (основной регистр).
	Current	Значение фазы гетеродина на момент прихода последней по времени метки.

Команды показывают текущее состояние каналов и текущие настройки гетеродинов. Описание состояний канал (биты Ttstop, Tstop, Tstart) содержит Таблица A.1.

Регистры начальной фазы и частоты гетеродина записываются командами WR_het_B и WR_FQ. Команда RD_het_base позволяет прочитать регистр начальной фазы. Команда RD_frequency не позволяет прочитать теневой регистр частоты гетеродина, а только основной – регистр FT на схеме гетеродина (см. Рисунок 10.4).

Команда RD_current позволяет считать фазу гетеродина на момент прихода метки – TR регистр.

A.4.2 RD_het_control (2) (0-23)

Формат данных для команды RD_het_control приведен ниже (см. Рисунок A.28).

31	30	29	28	26	25	23	22	20	19	16	15	14	13	12	11	0
Undefined	ot	Ms	pack	sin/cos	sel_tc	sel_cd	ac	lm	het_counter							

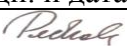
Рисунок A.28 – Формат данных для команды RD_het_control

В таблице (Таблица A.23) приведено описание полей данных для команды RD_het_control.

Таблица A.23 – Описание полей данных для команды RD_het_control

Биты	Название	Выполняемая функция
[31:30]	undefined	Не определены.
[29]	ot	Показывает значение бита ot, записанного командой WR_select.
[28:26]	Ms	Показывает значение поля Ms, записанного командой WR_select.
[25:23]	pack	Показывает значение поля pack, записанного командой WR_pack.
[22:20]	sin/cos	Показывает значение поля sin/cos, записанного командой WR_select.
[19:16]	sel_tc	Показывает значение поля sel_tc, записанного командой WR_select.
[15:14]	sel_cd	Показывает значение поля sel_cd, записанного командой WR_select.
[13]	ac	Показывает значение поля ac, записанного командой WR_het_counter.
[12]	lm	Показывает значение поля lm, записанного командой WR_het_counter.
[11:0]	het_counter	Показывает значение поля het_counter, записанного командой WR_het_counter.

Новые значения полей, записанные командой WR_select (sel_cd, sel_tc, sin/cos, Ms, ot), можно прочитать данной командой сразу после записи, новые значения остальных полей, которые записываются командами WR_het_counter (het_counter, lm, ac) и WR_pack (поле pack), – только после того, как эти значения перепишутся из теневых регистров.

					ЮФКВ.431268.006РЭ				Лист
									310
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

A.4.3 RD_norm_h / RD_norm_f / RD_norm_q (3/5/6) (0-23/0-11/0-11)

Формат данных для команд RD_norm_h, RD_norm_f, RD_norm_q приведен ниже (см. Рисунок А.29).

31	30	29	28	24	23	16	15	8	7	6	5	4	0
R	delta	exp	ov_im	ov_re	rnd	st	norm						

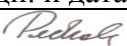
Рисунок А.29 – Формат данных для команд RD_norm_h, RD_norm_f, RD_norm_q

В таблице (Таблица А.24) приведено описание полей данных для команд RD_norm_h, RD_norm_f, RD_norm_q.

Таблица А.24 – Описание полей данных для команд RD_norm_h, RD_norm_f, RD_norm_q

Биты	Название	Выполняемая функция
[31]	R	Показывает значение поля R, записанного командами WR_norm_h / WR_norm_f / WR_norm_q.
[30:29]	delta	Показывает значение поля delta, записанного командами WR_norm_h / WR_norm_f / WR_norm_q.
[28:24]	exp	Показывает максимальное значение порядка из всех данных, прошедших через нормализатор за предшествующий интервал. Например, если через нормализатор проходили только данные из набора {0, 1, -1, -2}, то exp=0. Например, если через нормализатор проходили только данные из набора {2, 3, -3, -4}, то exp=1. Например, если через нормализатор проходили только данные из набора {1xxb(4,5,6,7), -(1xxb + 1)}, то exp=2. Например, если через нормализатор проходили только данные из набора {1xxx(8,9,...,15), -(1xxx + 1)}, то exp=3.
[23:16]	ov_im	Количество переполнений в нормализаторе по мнимой составляющей за предшествующий интервал.
[15:8]	ov_re	Количество переполнений в нормализаторе по действительной составляющей за предшествующий интервал.
[7:6]	rnd	Показывает значение поля rnd, записанного командами WR_norm_h / WR_norm_f / WR_norm_q.
[5]	st	Показывает значение поля st, записанного командами WR_norm_h / WR_norm_f / WR_norm_q.
[4:0]	norm	Показывает текущее значение поля norm. Данное поле могло быть как записано командами WR_norm_h / WR_norm_f / WR_norm_q, так и установлено автоматически в режиме автоподстройки нормализатора (см. п. 10.6).

Команды позволяют прочитать регистры нормализаторов гетеродина (RD_norm_h), фильтра (RD_norm_f), квадратора (RD_norm_q). Значения norm, st, rnd, delta, R записываются командами WR_norm_h, WR_norm_f, WR_norm_q, для чтения доступны только основные регистры. Значения ov_re, ov_im, exp вырабатываются внутри каждого нормализатора, обновляются после прихода метки. Максимальное значение ov_re, ov_im – 255, если переполнений больше, то значение остается 255. В режиме автоматической настройки нормализатора в теневой регистр norm (вместо старого) записывается значение exp (которое можно прочитать одной из данных команд) с поправкой delta.

					ЮФКВ.431268.006РЭ				Лист
									311
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

A.4.4 RD_filter_control (4) (0-3, 4-7, 8-11)

Формат данных для команды RD_filter_control приведен ниже (см. Рисунок A.30).

31	22	21	16	15	12	11	8	7	4	3	0
Undefined			filter_config		filter_mask3	filter_mask2	filter_mask1	filter_mask0			

Рисунок A.30 – Формат данных для команды RD_filter_control

В таблице (Таблица A.25) приведено описание полей данных для команды RD_filter_control.

Таблица A.25 – Описание полей данных для команды RD_filter_control

Биты	Название	Выполняемая функция
[31:22]	undefined	Не определены.
[21:16]	filter_config	Показывает значение поля filter_config, записанного командой WR_filter_mask.
[15:12]	filter_mask3	Показывает значение поля filter_mask3, записанного командой WR_filter_mask.
[11:8]	filter_mask2	Показывает значение поля filter_mask2, записанного командой WR_filter_mask.
[7:4]	filter_mask1	Показывает значение поля filter_mask1, записанного командой WR_filter_mask.
[3:0]	filter_mask0	Показывает значение поля filter_mask0, записанного командой WR_filter_mask.

Команда позволяет прочитать настройки кластера фильтров, записанные командой WR_filter_mask.

Так же как и для команды WR_filter_mask, чтение текущих настроек одного кластера производится одной командой.

A.4.5 RD_quad_control (0Ch) (0-11)

Формат данных для команд RD_quad_control приведен ниже (см. Рисунок A.31).

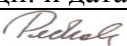
31	19	18	16	15	13	12	11	0
Undefined			pack_q	pack	lm	quad _ counter		

Рисунок A.31 – Формат данных для команды RD_quad_control

В таблице (Таблица A.26) приведено описание полей данных для команды RD_quad_control.

Таблица A.26 – Описание полей данных для команды RD_quad_control

Биты	Название	Выполняемая функция
[31:19]	undefined	Не определены.
[18:16]	pack_q	Показывает значение поля pack_q, записанного командой WR_pack.
[15:13]	pack	Показывает значение поля pack, записанного командой WR_pack.
[12]	lm	Показывает значение поля lm, записанного командой WR_quad_counter.
[11:0]	quad_counter	Показывает значение поля quad_counter, записанного командой WR_quad_counter.

					ЮФКВ.431268.006РЭ				Лист
									312
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	 16.01.14		22755-1						

Команда позволяет прочитать основные регистры, записанные командой WR_pack (поля pack_q, pack), и основные регистры, записанные командой WR_quad_counter (quad_counter, lm).

A.4.6 RD_TCGR (2) (24-27)

Формат данных для команды RD_TCGR приведен ниже (см. Рисунок A.32).

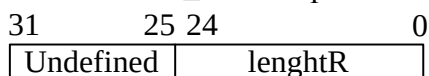


Рисунок A.32 – Формат данных для команды RD_TCGR

В таблице (Таблица A.27) приведено описание полей данных для команды RD_TCGR.

Таблица A.27 – Описание полей данных для команды RD_TCGR

Биты	Название	Выполняемая функция
[31:25]	undefined	Не определены
[24:0]	lengthR	Показывает значение счетчика длины последовательности TCG [24:0] на момент прихода последней по времени метки.

Команда читает текущее значение регистра длины последовательности, определенной командой WR_TCG, на момент прихода метки.

A.4.7 RD_ADC_control (0) (28)

Формат данных для команды RD_ADC_control приведен ниже (см. Рисунок A.33).

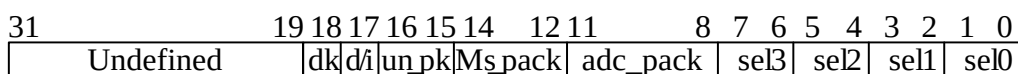


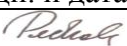
Рисунок A.33 – Формат данных для команды RD_ADC_control

В таблице (Таблица A.28) приведено описание полей данных для команды RD_ADC_control.

Таблица A.28 – Описание полей данных для команды RD_ADC_control

Биты	Название	Выполняемая функция
[31:19]	undefined	Не определены
[18]	dk	Показывает значение поля dk, записанного командой WR_ADC_control.
[17]	d/i	Показывает значение поля d/i, записанного командой WR_ADC_control.
[16:15]	un_pk	Показывает значение поля un_pk, записанного командой WR_ADC_control.
[14:12]	Ms_pack	Показывает значение поля Ms_pack, записанного командой WR_ADC_control.
[11:8]	adc_pack	Показывает значение поля adc_pack, записанного командой WR_ADC_control.
[7:0]	sel3, sel2, sel1, sel0	Показывает значение поля sel3, sel2, sel1, sel0, записанного командой WR_ADC_control.

Команда позволяет прочитать основные регистры управления входным устройством, блоками упаковки действительных данных, а также принудительной генерации ошибок контрольной суммы. Формат поля данных полностью соответствует команде WR_ADC_control.

					ЮФКВ.431268.006РЭ				Лист
									313
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата		
22755-2	 16.01.14		22755-1						

A.4.8 RD_Ms_counter (1-7) (29)

Формат данных для команды RD_Ms_counter приведен ниже (см. Рисунок A.34).

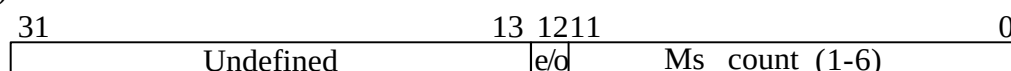


Рисунок A.34 – Формат данных для команды RD_Ms_counter

В таблице (Таблица A.29) приведено описание полей данных для команды RD_Ms_counter.

Таблица A.29 – Описание полей данных для команды RD_Ms_counter

Биты	Название	Выполняемая функция
[31:13]	undefined	Не определены.
[12]	e/o	Показывает текущее значение поля e/o. Поле e/o записывается командой WR_Ms_counter, но может меняться с приходом метки.
[11:0]	Ms_count (1-6)	Задают значения счетчиков для прореживания меток интервального таймера. В DSPPU имеется 6 счетчиков делителей меток с нумерацией 1-6. Номер регистра 7 определяет счетчик с номером 6.

Команда читает текущее значение счетчиков в делителях интервальных меток и разряд четности, используемый адресными генераторами. Данная команда может использоваться для определения, в какую выделенную область памяти (чётную или нечётную) производится запись в данный момент. Если e/o=1, то кадр пишется в чётную область памяти, если e/o=0, то кадр пишется в нечётную область.

A.4.9 RD_Channel_check (7) (0-27)

Формат данных для команды RD_Channel_check приведен ниже (см. Рисунок A.35).

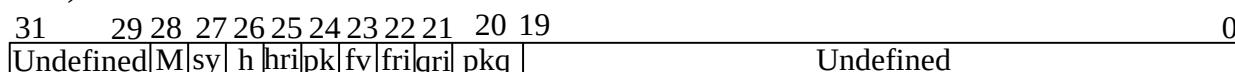


Рисунок A.35 – Формат данных для команды RD_Channel_check

В таблице (Таблица A.30) приведено описание полей данных для команды RD_Channel_check.

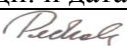
					ЮФКВ.431268.006РЭ					Лист
										314
Изм.	Лист	№ докум.	Подп.	Дата						
Инов.№подл.	Подп. и дата		Взам.инв.№		Инов.№дубл.	Подп. и дата				
22755-2	 16.01.14		22755-1							

Таблица А.30 – Описание полей данных для команды RD_Channel_check

Биты	Название	Выполняемая функция
[31:29]	undefined	Не определены.
[28]	M	Показывает бит маски для каналов 0 – 27, установленный командой WR_Mask_trap; 28 битов маски записываются одной командой, а читаются – каждый своей командой с соответствующим полем адреса CH_NMBR.
[27]	sy	Показывает, была ли ошибка синхронизации управляющей программы с метками. Этим битом отслеживаются следующие ошибки синхронизации: - 2 команды пуска данного канала без метки между ними, - любая команда записи после команды пуска канала, но до прихода метки, - смена источника данных канала во время его работы.
[26]	h	Показывает, было ли аппаратное нарушение контроля по модулю 3 в гетеродине.
[25]	hri	Показывает, было ли аппаратное нарушение контроля по модулю 3 в аккумуляторе блока гетеродина.
[24]	pk	Показывает, было ли переполнение буферов в блоке упаковки (нумерация поясняется ниже).
[23]	fv	Показывает, что данные на фильтр подаются чаще, чем раз в 4 такта работы DSPPU.
[22]	fri	Показывает, было ли аппаратное нарушение контроля по модулю 3 в аккумуляторе фильтра.
[21]	qri	Показывает, было ли аппаратное нарушение контроля по модулю 3 в аккумуляторе квадратора.
[20]	pkq	Показывает, было ли переполнение буферов в блоке упаковки данных, прошедших через квадраторы (нумерация поясняется ниже).
[19:0]	undefined	Не определены.

Биты, которые можно прочитать данной командой, отслеживают ошибки работы программы с DSPPU, а также аппаратные ошибки.

Обнаружение устройством ошибки приводит к тому, что соответствующий бит устанавливается в значение 1, это может стать причиной прерывания (см. п. 10.11). Биты ошибок сбрасываются только после выполнения данной команды или сброса блока DSPPU.

Переполнение буферов упаковки, которое можно обнаружить битами pk и pkq данной команды, может происходить при недостаточной пропускной способности шины записи в память. Данные теряются именно в канале, поэтому при чтении поле адреса CH_NMBR данной команды соответствует номеру канала обработки, а не канала упаковки. Для «раздваивающихся» каналов дополнительно введён бит rkq, который показывает, было ли переполнение в ветке канала, проходящей через квадратор.

A.4.10 RD_ADC_check (7) (28)

Формат данных для команды RD_ADC_check приведен ниже (см. Рисунок А.36).

31	29	28	27	24	23	22	21	10	9	8	7	5	4	2	1	0
Undefined	M	adc_pk_ov	sz	p2				Undefined	clk_x	sgmg_x	adc_x	delta				

Рисунок А.36 – Формат данных для команды RD_ADC_check

В таблице (Таблица А.31) приведено описание полей данных для команды RD_ADC_check.

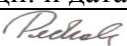
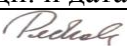
					ЮФКВ.431268.006РЭ				Лист
									315
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата		
22755-2	 16.01.14		22755-1						

Таблица А.31 – Описание полей данных для команды RD_ADC_check

Биты	Название	Выполняемая функция
[31:29]	undefined	Не определены.
[28]	M	Показывает бит маски a, установленный командой WR_Mask_trap.
[27:24]	adc_pk_ov	Показывает, было ли переполнение буферов в блоке упаковки ADC_PACK_UNIT, отдельно для каждого из каналов паковки 36-39: [27] – в канале паковки 39, [26] – в канале паковки 38, [25] – в канале паковки 37, [24] – в канале паковки 36.
[23]	sz	Показывает, что количество данных в кадре превысило установленное командой WR_AG_size значение size.
[22]	p2	Показывает, было ли аппаратное нарушение четности данных в адресных генераторах.
[21:10]	undefined	Не определены.
[9:8]	clk_x	Показывает, как соотносятся момент чтения данных с выхода ADC и фаза тактового сигнала ADC: 10h – момент чтения находится в отрицательном полутакте тактового сигнала ADC, 11h – момент чтения находится в положительном полутакте тактового сигнала ADC, 00h, 01h – однозначно нельзя определить полутакт, в котором находится момент чтения.
[7:5]	sgmg_x	Показывает, насколько успешно произведена синхронизация с внешним источником сигналов на выходах SIGNx/MAGNx, которые работают на меньшей в 4 раза частоте, чем DSPPU. Данное поле при сбросе и при чтении принимает значение 000b, затем производится сравнение каждого зафиксированного отсчёта с тремя следующими по времени значениями на входах – через 1, 2 и 3 такта. Несовпадение зафиксированного значения со значением сигнала на следующий такт устанавливает бит [7] в состояние 1. Несовпадение зафиксированного значения со значением сигнала на 2-й такт устанавливает бит [6] в состояние 1. Несовпадение зафиксированного значения со значением сигнала на 3-й такт устанавливает бит [5] в состояние 1. Синхронизацию можно считать успешной, если в данном поле всегда читаются значения 001b или 011b.
[4:2]	adc_x	Показывает, насколько успешно произведена синхронизация с блоками ADC, которые работают на меньшей в 4 раза частоте, чем DSPPU. Данное поле при сбросе и при чтении принимает значение 000b, затем производится сравнение каждого зафиксированного отсчёта с тремя следующими по времени значениями на входах – через 1, 2 и 3 такта. Несовпадение зафиксированного значения со значением сигнала на следующий такт устанавливает бит [4] в состояние 1. Несовпадение зафиксированного значения со значением сигнала на 2-й такт устанавливает бит [3] в состояние 1. Несовпадение зафиксированного значения со значением сигнала на 3-й такт устанавливает бит [2] в состояние 1. Синхронизацию можно считать успешной, если в данном поле всегда читаются значения 001b или 011b.
[1:0]	delta	Показывает значение поля delta, установленного командой WR_ADC_delta.

Переполнение буферов паковки, которое можно обнаружить битами adc_pk_ov данной команды, может происходить при недостаточной пропускной способности шины записи в память.

Биты adc_pk_ov, sz, p2 могут стать причиной прерывания (см. п. 10.11). Эти биты сбрасываются при чтении.

					ЮФКВ.431268.006РЭ			Лист
								316
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	 16.01.14		22755-1					

Процесс синхронизации между источником сигнала с частотой $F_{clk}/4$ и блоком DSPPU, который работает на частоте F_{clk} (до 328 МГц), описан в п. 10.2.

При настройке синхронизации, подбирая командой `WR_ADC_delta` поле `delta`, следует добиться того, чтобы в поле `adc_x` (или `sgmg_x`, в зависимости от того, какой источник сигналов используется) всегда читалось значение 001b или 011b. При этом необходимо, чтобы входной сигнал менялся во времени достаточно часто, то есть несколько изменений происходило между выполнениями команды `RD_ADC_check`.

A.5 Команды для адресных генераторов блока DSPPU

В описании команды обращения к памяти адресных генераторов в скобках указан интервал, в котором может лежать совокупность полей адреса `PKCH_N` и `e/o`.

A.5.1 `WR_AG_adr_RAM` (580h – 5CFh)

Формат данных для команды `WR_AG_adr_RAM` приведен ниже (см. Рисунок A.37).

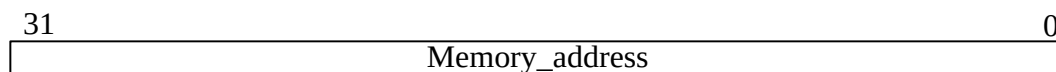


Рисунок A.37 – Формат данных для команды `WR_AG_adr_RAM`

В таблице (Таблица A.32) приведено описание поля данных для команды `WR_AG_adr_RAM`.

Таблица A.32 – Описание полей данных для команды `WR_AG_adr_RAM`

Биты	Название	Выполняемая функция
[31:0]	Memory_address	Задаёт начальный адрес в памяти для кадра соответствующего канала.

Команда устанавливает начальный адрес в памяти как для чётного, так и для нечётного кадра канала. Механизм чётных и нечётных кадров описан в п. 10.10. Подавать необходимо последовательно по две команды: для четного кадра, затем для соответствующего нечетного кадра (загрузить начальный адрес только четного кадра и сохранить старое значение начального адреса нечетного кадра нельзя). Команды с полем адреса `e/o=1` устанавливают адрес нечётных кадров, команды с `e/o=0` – чётных.

A.5.2 `WR_AG_size / RD_AG_size` (5FDh)

В описании команд обращения к регистрам адресных генераторов в скобках указано значение поля адреса ОРС, соответствующее описываемой команде.

Формат данных для команд `WR_AG_size`, `RD_AG_size` приведен ниже (см. Рисунок A.38).

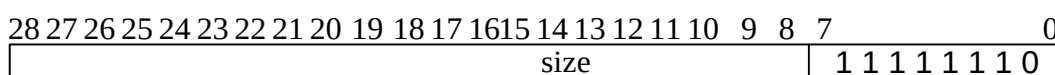


Рисунок A.38 – Формат данных для команд `WR_AG_size`, `RD_AG_size`

					ЮФКВ.431268.006РЭ				Лист
									317
Изм.	Лист	№ докум.	Подп.	Дата					
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата			
22755-2	<i>Редько</i> 16.01.14		22755-1						

В таблице (Таблица A.33) приведено описание полей данных для команд WR_AG_size, RD_AG_size.

Таблица A.33 – Описание полей для команды WR_AG_size, RD_AG_size

Биты	Название	Выполняемая функция
[28:0]	size	Задаёт ограничение размера кадра в 32-х разрядных словах. Служит для защиты памяти от программных ошибок. Во время начальной установки происходит обнуление разрядов с 8-го по 28-й. Разряды поля size [7:0] = 1111110b = 254 не могут быть изменены.

Команда записи WR_AG_size устанавливает максимально возможный размер кадра в памяти в 32-х разрядных словах. *Например*, запись командой WR_AG_size числа 00000100h установит максимальный размер кадра $256 + 254 = 510$ 32-разрядных слов. Постоянно установленные разряды [7:0] не позволяют задать максимальный размер кадра меньше, чем 254 32-х разрядных слова.

Если при записи в память количество данных одного кадра начинает превышать максимально возможный размер, то последующие записи в память для этого кадра будут производиться по одному и тому же последнему разрешенному адресу. В заголовке кадра в поле size (см. Рисунок 10.19) при этом запишется реальное количество состоявшихся записей (для нашего примера – 255 64-х разрядных слов).

Команда чтения RD_AG_size позволяет прочитать значение битов [28:8] поля size, нули всегда считываются в разрядах с 31-го по 29-й и с 7-го по 0-й.

A.5.3 WR_AG_control_monitor / RD_AG_control_monitor (5Feh)

Формат данных для команд WR_AG_control_monitor, RD_AG_control_monitor size приведен ниже (см. Рисунок A.39).

26	25	24	23	20	19	14	13	8	7	0
sel	H	Ms_	part	channel	number	Reserved				

Рисунок A.39 – Формат данных для команд WR_AG_control_monitor, RD_AG_control_monitor

В таблице (Таблица A.34) приведено описание полей данных для команд WR_AG_control_monitor, RD_AG_control_monitor.

					ЮФКВ.431268.006РЭ	Лист		
						318		
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.		Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата	
22755-2		 16.01.14		22755-1				

Таблица А.34 – Описание полей данных для команд WR_AG_control_monitor, RD_AG_control_monitor

Биты	Название	Выполняемая функция
[26:25]	sel	Указывает, какие события на шине записи в память следует подсчитывать: 00b – шина не активна (нет запроса в память), 01b – шина активна и нет сигнала готовности RDY (блокировка), 10b – шина активна и есть сигнал RDY (запрос в память проходит без блокировки), 11b – шина активна (выставлен запрос в память). Результат подсчёта можно прочитать командой RD_monitor_count.
[24]	H	Управляет записью в разряды [63:56] заголовка кадра: 0b – в [63:56] разряды заголовка записываются 8 старших разрядов гетеродина, 1b – в [63:56] разряды заголовка записываются старшие разряды размера кадра.
[23:20]	Ms_part	Задаёт номер шестнадцатой доли периода прихода метки от интервального таймера, при достижении которой будет выдаваться прерывание, при этом не должны были сработать условия возникновения прерывания по полям channel и number. Если Ms_part равно 0, то прерывание по этому условию вырабатываться не будет (маскируется).
[19:14]	channel	Задаёт значение номера канала, по завершении обработки кадра которым, будет выдаваться прерывание. Если channel принимает значения от 40 до 63, то прерывание по этому условию вырабатываться не будет (маскируется).
[13:8]	number	Задаёт значение количества каналов, которые должны завершить обработку своих кадров. Достижение этого количества приведёт к выдаче прерывания. Если number равно 0 или от 48 до 63, то прерывание по этому условию вырабатываться не будет (маскируется).
[7:0]	Reserved	Не определены.

Команда WR_AG_control_monitor позволяет задавать ситуации, по которым выдается прерывание типа INT_DSPPU_CH. Поле команды number определяет количество каналов, которые должны завершить обработку своих кадров, и только тогда выдается запрос на прерывание. Поле команды channel определяет номер канала, который должен завершить обработку своего кадра, и только тогда выдается запрос на прерывание. Поле команды Ms_part определяет номер шестнадцатой доли периода прихода метки от интервального таймера, при достижении которой выдается прерывание, при этом не должны были сработать два первых условия из команды. Все три составляющие этого прерывания можно замаскировать независимо друг от друга этой же командой записи. Команды RD_ms_part_monitor и RD_stack_monitor позволяют прочитать статус всех трёх составляющих прерывания одновременно, после выполнения команды RD_stack_monitor запросы на прерывание сбрасываются.

Между двумя соседними метками от интервального таймера прерывание выдается один раз.

Разряд H рекомендуется установить в 1, если размер кадра будет более 7FFFFh 128-разрядных слов: это позволит точно определять размер записанного кадра, однако, при этом теряется информация о старших разрядах гетеродина (с 19-го по 26-й). Таблица 10.2 содержит полное описание формата заголовка кадра.

Поле sel определяет, какие события на шине записи в память следует подсчитывать специальным счетчиком. Счётчик событий шины устанавливается в ноль каждой меткой от интервального таймера, значение счётчика на момент прихода метки от интервального таймера можно прочитать командой RD_monitor_count.

Команда чтения RD_AG_control_monitor позволяет прочитать содержимое полей, записанных командой WR_AG_control_monitor; нули всегда считываются в разрядах с 31-го по 29-й и с 7-го по 0-й.

					ЮФКВ.431268.006РЭ			Лист
								319
Изм.	Лист	№ докум.	Подп.	Дата				
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата		
22755-2	<i>Редько</i> 16.01.14		22755-1					

A.5.4 RD_monitor_count (5FCh)

Формат данных для команды RD_monitor_count приведен ниже (см. Рисунок A.40).

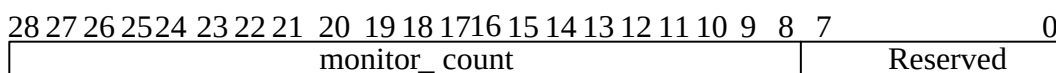


Рисунок А.40 – Формат данных для команды RD_monitor_count

В таблице (Таблица А.35) приведено описание полей данных для команды RD monitor count.

Таблица А.35 – Описание полей данных для команды RD monitor count

Биты	Название	Выполняемая функция
[28:8]	monitor_count	Показывает количество событий, выбранных для подсчёта полем sel команды WR_AG_control_monitor, которые были зафиксированы за последний временной интервал.

Команда позволяет прочитать значение счетчика, накопленное за период между последними двумя метками интервального таймера, нули всегда считываются в разрядах с 31-го по 29-й и с 7-го по 0-й. Счетчик подсчитывает события, происходящие на шине записи в память, согласно значению поля `sel`, записанного командой `WR_AG_control_monitor`. Можно пересчитать количество свободных тактов (`sel=0`), количество занятых тактов (`sel=3`), количество выданных запросов в память (`sel=2`) или количество тактов, в течение которых шина была заблокирована (`sel=1`). При одном запросе в память записывается 64-разрядное слово.

Такой мониторинг шины введен для того, чтобы облегчить принятие решения о дополнительной нагрузке шины (например при добавлении канала) или для распределения каналов по кластерам (работа одного и того же количества каналов в одном кластере или нескольких может по-разному нагружать шину).

A.5.5 RD_stack_monitor (5FFh)

Формат данных для команды RD_stack_monitor приведен ниже (см. Рисунок A.41).

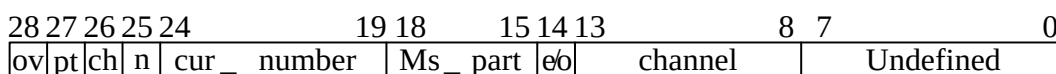


Рисунок А.41 – Формат данных для команды RD stack monitor

В таблице (Таблица A.36) приведено описание полей данных для команды RD stack monitor.

					ЮФКВ.431268.006РЭ	Лист	
						320	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв. №подл.		Подп. и дата			Взам. инв. №	Инв. №дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

Таблица A.36 – Описание полей данных для команды RD_stack_monitor

Биты	Название	Выполняемая функция
[28]	ov	Прерывание, вызванное переполнением стека монитора (более 47 позиций).
[27]	pt	Прерывание по значению шестнадцатой доли периода интервальной метки.
[26]	ch	Прерывание по номеру канала.
[25]	n	Прерывание по количеству позиций в стеке монитора.
[24:19]	cur_number	Показывает число позиций, оставшихся в стеке монитора (без учета прочитанной).
[18:15]	Ms_part	Берется из вершины стека мониторинга. Показывает значение шестнадцатой доли периода метки интервального таймера в момент завершения данного кадра.
[14]	e/o	Берется из вершины стека мониторинга. Показывает, в четную или нечетную область памяти был записан кадр.
[13:8]	channel	Берется из вершины стека мониторинга. Показывает номер канала паковки (от 0 до 39), завершение кадра которым вызвало данную запись в стек монитора.
[7:0]	undefined	Не определены.

Записи в стек попадают одновременно с записью в память заголовка кадра и могут быть прочитаны данной командой.

Команда, кроме чтения значений полей в верхней позиции стека, сбрасывает биты прерывания и продвигает указатель стека мониторинга на следующую позицию. Если стек пуст, то в разрядах cur_number читается 0, в разрядах Ms_part читается значение текущей шестнадцатой доли интервала, в разряде e/o читается 1, в разрядах channel читается 11111b. Поле channel=11111b однозначно показывает, что текущее чтение было произведено из пустого стека.

Установка любого из битов ov, pt, ch, n вызывает запрос на прерывание типа INT_DSPPU_CH. Прерывания по каждому из битов можно маскировать командой WR_AG_control_monitor.

A.5.6 RD_ms_part_monitor (5FBh)

Формат данных для команды RD_ms_part_monitor приведен ниже (см. Рисунок A.42).

28	27	26	25	24	19	18	15	14	8	7	0
ov	pt	ch	n	cur_number	cur_Ms_part	1	1	1	1	1	1
										Undefined	

Рисунок A.42 – Формат данных для команды RD_ms_part_monitor

В таблице (Таблица A.37) приведено описание полей данных для команды RD_ms_part_monitor.

Таблица A.37 – Описание полей данных для команды RD_ms_part_monitor

Биты	Название	Выполняемая функция
[28]	ov	Прерывание, вызванное переполнением стека монитора (более 47 позиций).
[27]	pt	Прерывание по значению шестнадцатой доли периода интервальной метки.
[26]	ch	Прерывание по номеру канала.
[25]	n	Прерывание по количеству позиций в стеке монитора.
[24:19]	cur_number	Показывает число позиций, занятых в стеке монитора.
[18:15]	cur_Ms_part	Показывает значение текущей (на момент прихода команды) шестнадцатой доли периода метки от интервального таймера.

Команда читает статусы прерываний, текущий размер стека монитора и долю периода метки. В отличие от команды RD_stack_monitor, данная команда не сбрасывает биты прерывания и не продвигает указатель стека монитора на следующую позицию.

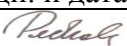
					ЮФКВ.431268.006РЭ					Лист
										321
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.		Подп. и дата			
22755-2	 16.01.14		22755-1							

Таблица системы команд DSPPU

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
1	base/frequency																									0/1												b, f	(0-27)							
2													ot		Ms		sin/cos			sel_tc			sel_cd			A	(0-27) (0-23)																			
3	(stop → reset_Ms)												ac		lm		het_counter												2	s	(0-23)															
4	R - repeat, delta (00→0, 01→+1, 10→-2, 11→-1)												R		delta		rnd		st		norm				3 5,6	s	(0-23) (0-11)																			
5	filter-config								filter_mask3				filter_mask2				filter_mask1				filter_mask0				4	s	(0-11)																			
6													lm		quad_counter												C	s	(0-11)																	
7													pack_q				pack				7	s	(0-23)																							
8	T-code_length/mask/significance																									2/5/6												s	(24-27)							
9	(stop → reset → Ms_pack, pack_adc)												T-code_control												4	s	(24-27)																			
10	ADC_control												dk		d/i		un_pk		Ms_pack		pack_adc/pr		sel3		sel2		sel1		sel0		0	s	(28)													
11	ADC_ins				(M=V=0 → NOP)												M		V		11		10		9		8		7		6		5		4		3		2		1		0		1	(28)
12	ADC_delta				(DMA)																								delta				2	(28)												
13	ADC_calibration												zero offset												4-7				(28)																	
14													e/o_pr(7)				e/o		Ms_count(1-6)												1-7				(29)											
15	b		f		ph		a		T-code				cl_5				cl_4				cl_3				cl_2				cl_1				cl_0				1C-1F				Start +exp(h,f,q)					
16					pk		T-code				cl_5				cl_4				cl_3				cl_2				cl_1				cl_0				1A-1B				Stop Stop_ms							
17					a		T-code				cl_5				cl_4				cl_3				cl_2				cl_1				cl_0				18				Mask_trap							
18																										00				Reset																

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
1	tt				t		r		k		het_base/frequency/current																			0/1/8		(0-27)																	
2					ot		Ms		pack		sin/cos		sel_tc		sel_cd		ac		lm		het_counter										2		(0-23)																
3	R		delta				exp		ov_im					ov_re					rnd		st		norm					3 5,6		(0-23) (0-11)																			
4											filter-config					filter_mask3					filter_mask2					filter_mask1					filter_mask0					4		(0-11)											
5											pack_q					pack					lm					quad_counter										C		(0-11)											
6											length_R																				2		(24-27)																
7	ADC										dk					d/i					un_pk					Ms					pack					sel3		sel2		sel1		sel0		0		(28)			
8											e/o_Ms(0)										e/o_pr(7)					e/o					Ms_count_curr(1-6)										0-7		(29)						
9					M		sy		g		gri		pk		fv		fri		qri		pkq		Channel_check										7		(0-27)														
10					M		pk		pk		pk		pk		sz		p2		ADC_check										clk_x					sgmg_x					adc_x					delta		7		(28)	

```

sin/cos - 000 (6 -> 4), 001 - (sin=0,cos=1), 010 - (sin=1,cos=0), 011 - (sin=0,cos=0)
sin/cos - 1XX (8 -> 6) 100 (rnd to -inf), 101 (to +inf), 110 (to nearest), 111 (to nearest_even)
st - saturation (on overflow) rnd - 00 (to -inf), 01 (to +inf), 10 (to zero), 11 (to nearest_even)
ac - accumulation/decimation lm - lock_metka dk - damage_control
pack - {d/s_w, (11 - 8b, 10 - 4b, 01 - 2b, 00 - stop)}
un_pk - 111 - 16b, 110 - 8b, 101 - 4b, 100 - sgmg; 0XX - adc_ins
sel_adc - 11 - ADC, 10 - sqmg, 0X - DMA / ADC ins

```

channel 0-23
channel 24-27 – T-code
28 – ADC
29 – Ms_count
30 – reserved
31 – SYN

Wr	N_ch										N_reg											
	10	9	8	7	6	5	4	3	2	1	0											
	"0"																					
	"1" "0"		W		0,1,2				RAM				word64				0/1					
	AG		3		AG_ch								0/1									
Rd	N_ch										N_reg											
	9	8	7	6	5	4	3	2	1	0												

					ЮФКВ.431268.006РЭ	Лист	
						322	
Изм.	Лист	№ докум.	Подп.	Дата			
Инв. №подл.		Подп. и дата			Взам. инв. №	Инв. №дубл.	Подп. и дата
22755-2		 16.01.14			22755-1		

В Приложение. Таблица преобразования синуса/косинуса в гетеродине.

Таблица В.1 – Таблица TBL 4 для формирования синуса и косинуса

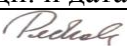
Значение T<26:21>	sin	cos	Значение T<26:21>	sin	cos	Значение T<26:21>	sin	cos	Значение T<26:21>	sin	cos
0	0	7	16	7	0	32	0	-7	48	-7	0
1	1	7	17	7	-1	33	-1	-7	49	-7	1
2	1	7	18	7	-1	34	-1	-7	50	-7	1
3	2	7	19	7	-2	35	-2	-7	51	-7	2
4	3	6	20	6	-3	36	-3	-6	52	-6	3
5	3	6	21	6	-3	37	-3	-6	53	-6	3
6	4	6	22	6	-4	38	-4	-6	54	-6	4
7	4	5	23	5	-4	39	-4	-5	55	-5	4
8	5	5	24	5	-5	40	-5	-5	56	-5	5
9	5	4	25	4	-5	41	-5	-4	57	-4	5
10	6	4	26	4	-6	42	-6	-4	58	-4	6
11	6	3	27	3	-6	43	-6	-3	59	-3	6
12	6	3	28	3	-6	44	-6	-3	60	-3	6
13	7	2	29	2	-7	45	-7	-2	61	-2	7
14	7	1	30	1	-7	46	-7	-1	62	-1	7
15	7	1	31	1	-7	47	-7	-1	63	-1	7

Таблица В.2 – Таблица TBL 6 для формирования синуса и косинуса

T<26:19>	sin	cos	T<26:19>	sin	cos	T<26:19>	sin	cos	T<26:19>	sin	cos
0	0	31	64	31	0	128	0	-31	192	-31	0
1	1	31	65	31	-1	129	-1	-31	193	-31	1
2	2	31	66	31	-2	130	-2	-31	194	-31	2
3	2	31	67	31	-2	131	-2	-31	195	-31	2
4	3	31	68	31	-3	132	-3	-31	196	-31	3
5	4	31	69	31	-4	133	-4	-31	197	-31	4
6	5	31	70	31	-5	134	-5	-31	198	-31	5
7	5	31	71	31	-5	135	-5	-31	199	-31	5
8	6	30	72	30	-6	136	-6	-30	200	-30	6
9	7	30	73	30	-7	137	-7	-30	201	-30	7
10	8	30	74	30	-8	138	-8	-30	202	-30	8
11	8	30	75	30	-8	139	-8	-30	203	-30	8
12	9	30	76	30	-9	140	-9	-30	204	-30	9
13	10	29	77	29	-10	141	-10	-29	205	-29	10
14	10	29	78	29	-10	142	-10	-29	206	-29	10
15	11	29	79	29	-11	143	-11	-29	207	-29	11
16	12	29	80	29	-12	144	-12	-29	208	-29	12
17	13	28	81	28	-13	145	-13	-28	209	-28	13
18	13	28	82	28	-13	146	-13	-28	210	-28	13
19	14	28	83	28	-14	147	-14	-28	211	-28	14
20	15	27	84	27	-15	148	-15	-27	212	-27	15
21	15	27	85	27	-15	149	-15	-27	213	-27	15
22	16	27	86	27	-16	150	-16	-27	214	-27	16
23	17	26	87	26	-17	151	-17	-26	215	-26	17
24	17	26	88	26	-17	152	-17	-26	216	-26	17
25	18	25	89	25	-18	153	-18	-25	217	-25	18
26	18	25	90	25	-18	154	-18	-25	218	-25	18
27	19	24	91	24	-19	155	-19	-24	219	-24	19
28	20	24	92	24	-20	156	-20	-24	220	-24	20
29	20	23	93	23	-20	157	-20	-23	221	-23	20

					ЮФКВ.431268.006РЭ	Лист 323
Изм.	Лист	№ докум.	Подп.	Дата		
Инв.№подл.	Подп. и дата		Взам.инв.№		Инв.№дубл.	Подп. и дата
22755-2	 16.01.14		22755-1			

T<26:19>	sin	cos	T<26:19>	sin	cos	T<26:19>	sin	cos	T<26:19>	sin	cos
30	21	23	94	23	-21	158	-21	-23	222	-23	21
31	21	22	95	22	-21	159	-21	-22	223	-22	21
32	22	22	96	22	-22	160	-22	-22	224	-22	22
33	22	21	97	21	-22	161	-22	-21	225	-21	22
34	23	21	98	21	-23	162	-23	-21	226	-21	23
35	23	20	99	20	-23	163	-23	-20	227	-20	23
36	24	20	100	20	-24	164	-24	-20	228	-20	24
37	24	19	101	19	-24	165	-24	-19	229	-19	24
38	25	18	102	18	-25	166	-25	-18	230	-18	25
39	25	18	103	18	-25	167	-25	-18	231	-18	25
40	26	17	104	17	-26	168	-26	-17	232	-17	26
41	26	17	105	17	-26	169	-26	-17	233	-17	26
42	27	16	106	16	-27	170	-27	-16	234	-16	27
43	27	15	107	15	-27	171	-27	-15	235	-15	27
44	27	15	108	15	-27	172	-27	-15	236	-15	27
45	28	14	109	14	-28	173	-28	-14	237	-14	28
46	28	13	110	13	-28	174	-28	-13	238	-13	28
47	28	13	111	13	-28	175	-28	-13	239	-13	28
48	29	12	112	12	-29	176	-29	-12	240	-12	29
49	29	11	113	11	-29	177	-29	-11	241	-11	29
50	29	10	114	10	-29	178	-29	-10	242	-10	29
51	29	10	115	10	-29	179	-29	-10	243	-10	29
52	30	9	116	9	-30	180	-30	-9	244	-9	30
53	30	8	117	8	-30	181	-30	-8	245	-8	30
54	30	8	118	8	-30	182	-30	-8	246	-8	30
55	30	7	119	7	-30	183	-30	-7	247	-7	30
56	30	6	120	6	-30	184	-30	-6	248	-6	30
57	31	5	121	5	-31	185	-31	-5	249	-5	31
58	31	5	122	5	-31	186	-31	-5	250	-5	31
59	31	4	123	4	-31	187	-31	-4	251	-4	31
60	31	3	124	3	-31	188	-31	-3	252	-3	31
61	31	2	125	2	-31	189	-31	-2	253	-2	31
62	31	2	126	2	-31	190	-31	-2	254	-2	31
63	31	1	127	1	-31	191	-31	-1	255	-1	31

					ЮФКВ.431268.006РЭ					Лист
										324
Изм.	Лист	№ докум.	Подп.	Дата						
Инв.№подл.		Подп. и дата			Взам.инв.№		Инв.№дубл.		Подп. и дата	
22755-2		 16.01.14			22755-1					

[illegible]

					ЮФКВ.431268.006РЭ	Лист
						325
Изм.	Лист	№ докум.	Подп.	Дата		
Инв.№подл.		Подп. и дата		Взам.инв.№	Инв.№дубл.	Подп. и дата
22755-2		 16.01.14		22755-1		