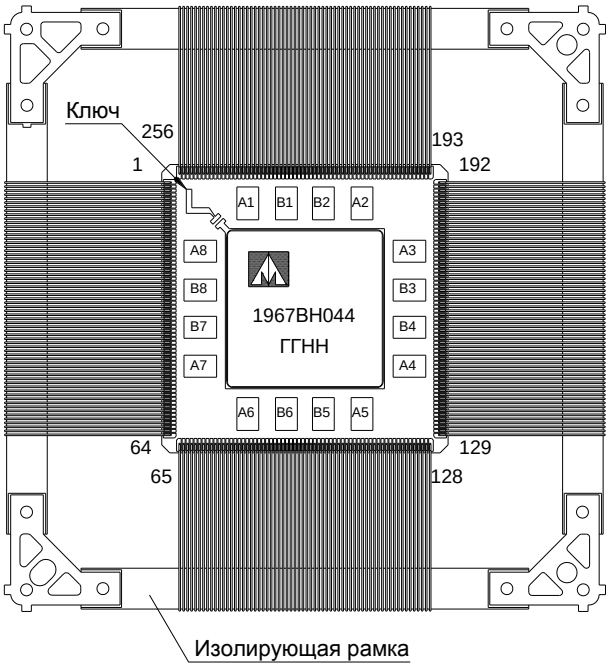
Микросхема высокопроизводительного процессора цифровой обработки сигнала с суперскалярной архитектурой 1967ВН044, К1967ВН044, К1967ВН044К, К1967ВН04BG, 1967ВН04Н4, К1967ВН04Н4



Основные параметры микросхемы

- Два последовательных порта;
- Разрядность внешней шины данных 32 бит;
- Разрядность внешней шины адреса 22 бит;
- Напряжение питания ввода/вывода U_{CCIO} от 3,0 до 3,6 В;
- Динамический ток потребления схем ввода/вывода I_{OCC} не более 800 мА;
- Тактовая частота процессора 230 МГц;

ГГ/YY – год выпуска
НН/WW – неделя выпуска

- Рабочий диапазон температур:

Обозначение	Диапазон
1967ВН044	минус 60 –105 °С
К1967ВН044	минус 60 – 105 °С
К1967ВН044К	0 – 70 °С
К1967ВН04BG	минус 40 – 85 °С

Тип корпуса:

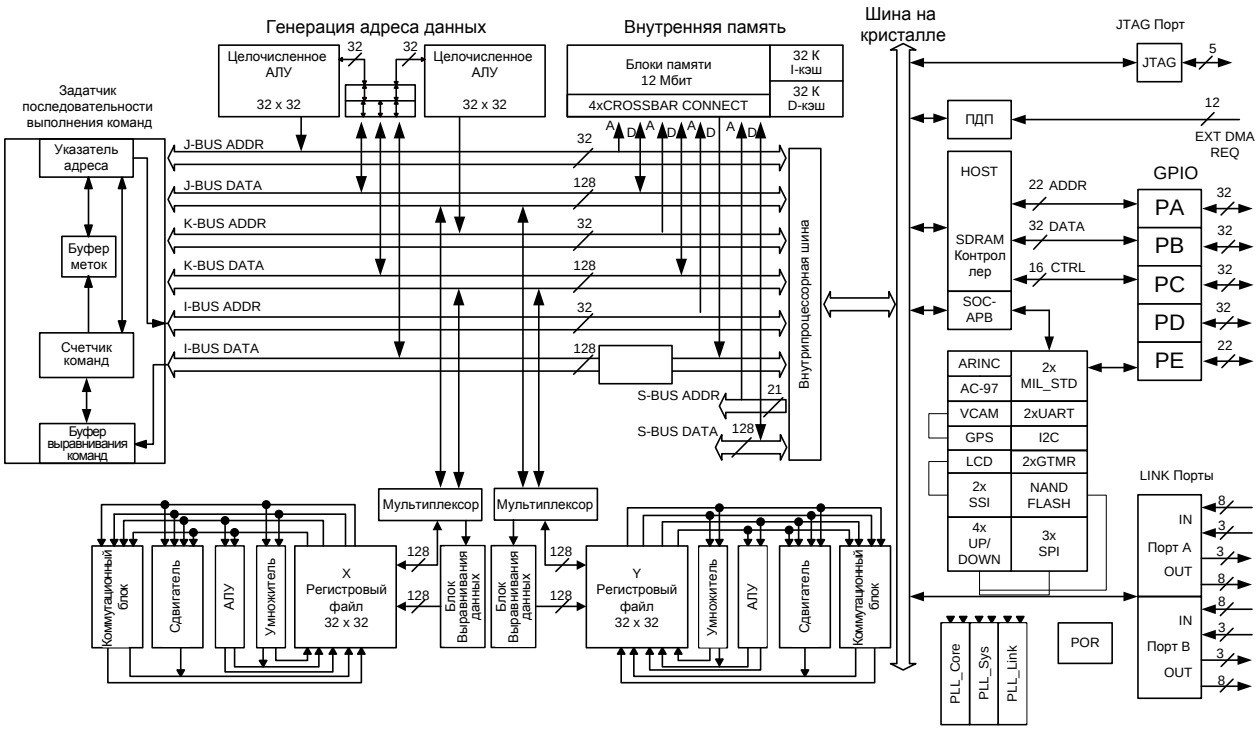
- для микросхем 1967ВН044, К1967ВН044, К1967ВН044К – 256-ти выводной металлокерамический корпус 4244.256-3;
- для микросхем К1967ВН04BG – 288-ми выводной пластиковый корпус BGA288;
- микросхемы 1967ВН04Н4, К1967ВН04Н4 поставляются в бескорпусном исполнении.

Общее описание и области применения микросхемы

Микросхемы интегральные 1967ВН044 (далее – микросхемы) предназначены для использования в аппаратуре специального назначения в качестве процессора цифровой обработки сигналов с ОЗУ 12 Мбит и тактовой частотой 230 МГц.

Содержание

1 Структурная блок-схема микросхемы



Используемые сокращения:

HOST – интерфейс доступа внешнего ведущего устройства;
POR – сброс по включению питания;
АЛУ – арифметико-логическое устройство;
ПДП – контроллер прямого доступа в память.

Рисунок 1 – Структурная блок-схема микросхемы

2 Условное графическое обозначение

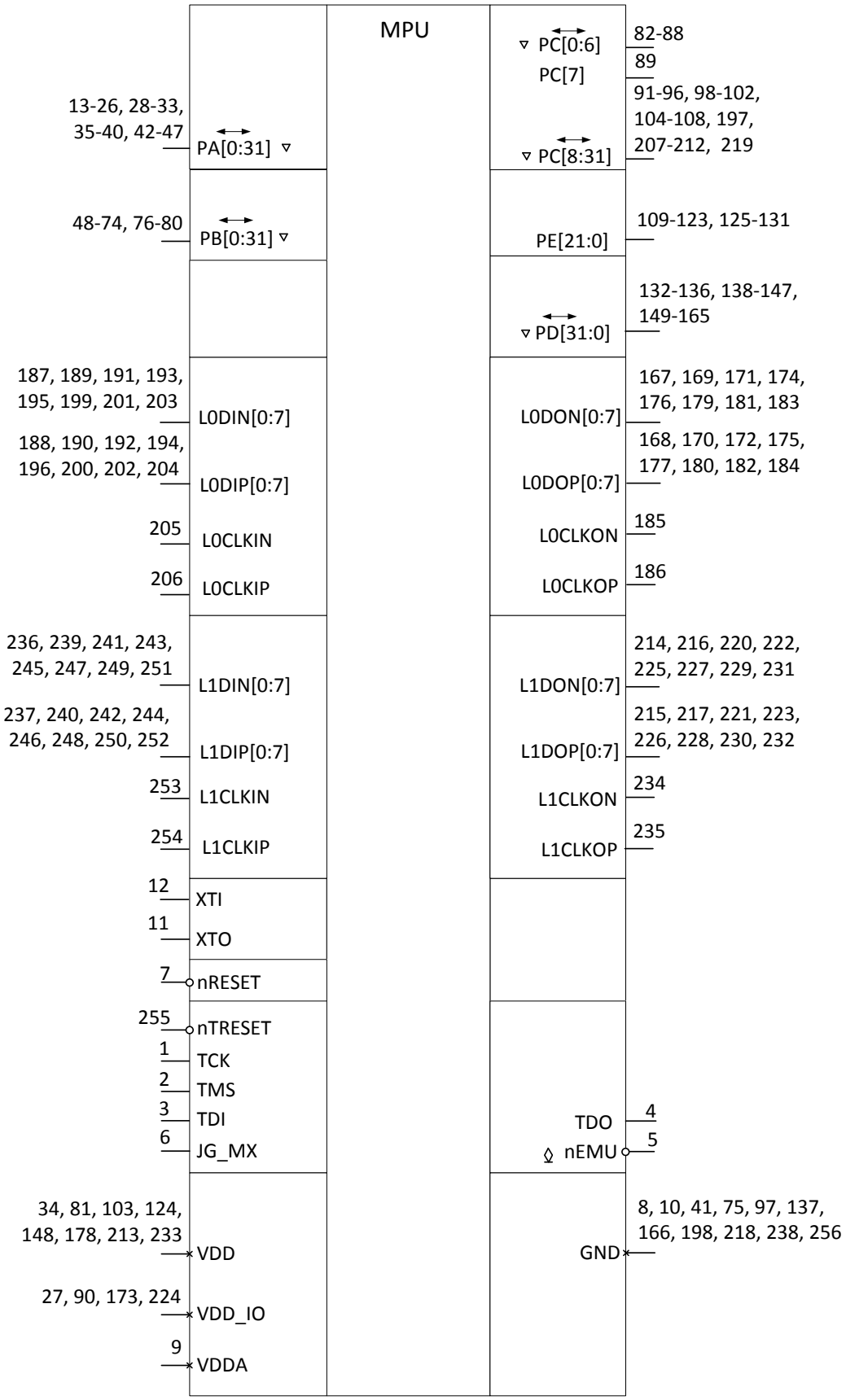


Рисунок 2 – Условное графическое обозначение микросхем в корпусе 4244.256-3

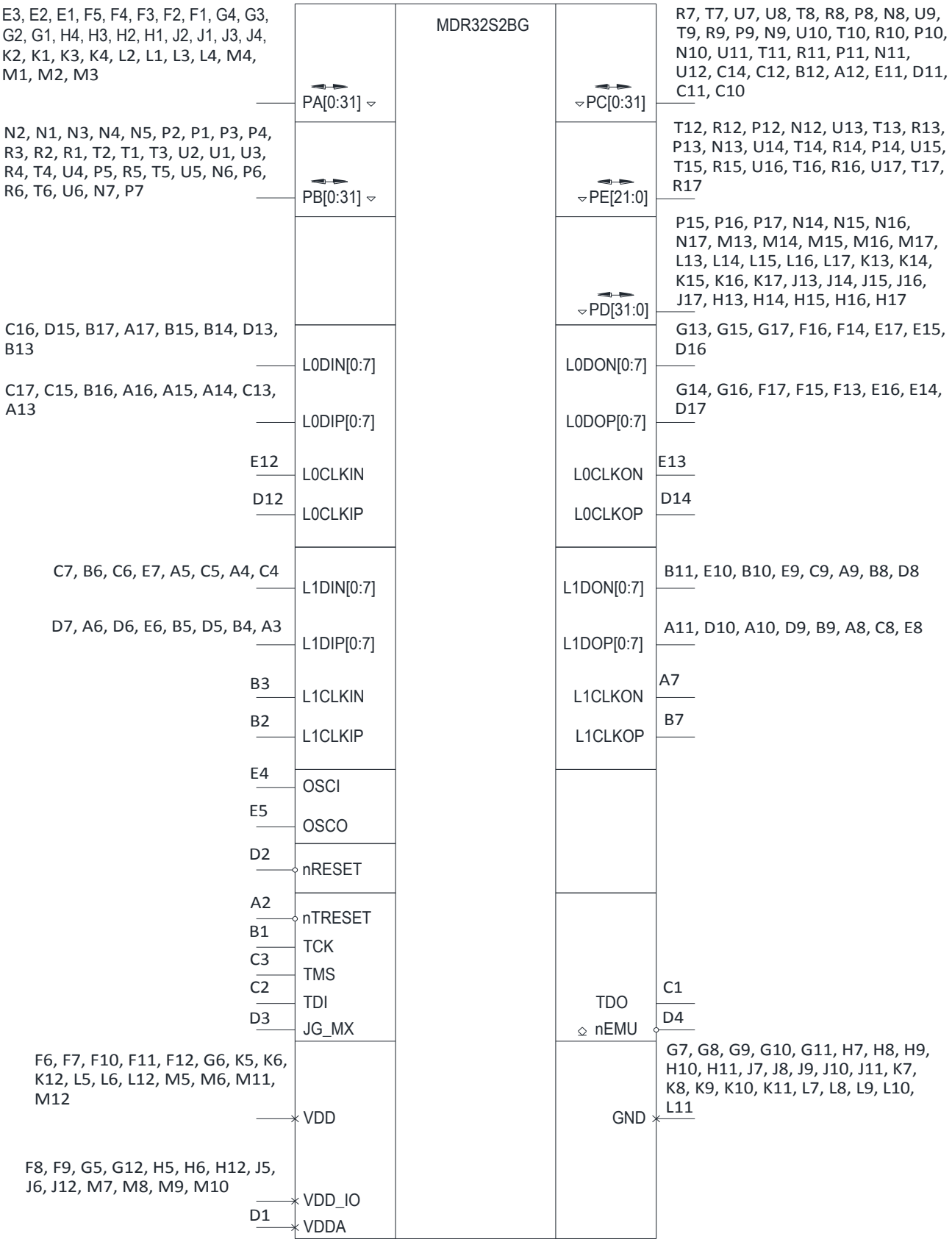


Рисунок 3 – Условное графическое обозначение микросхем в корпусе BGA288

3 Описание выводов

Таблица 1 – Описание выводов микросхем в корпусе 4244.256-3

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
1	1	TCK	–	I (pu)	Интерфейс JTAG. Синхросигнал
2	2	TMS	–	I (pu)	Интерфейс JTAG. Выбор режима
3	3	TDI	–	I (pu)	Интерфейс JTAG. Вход данных
4	4	TDO	–	O	Интерфейс JTAG. Выход данных
5	5	nEMU	–	O	Индикатор режима отладки
6	6	JG_MX	–	I (pd)	Выбор внутреннего JTAG-контроллера (0 – функциональный JTAG-контроллер, 1 – тестовый JTAG-контроллер). Имеет внутреннюю подтяжку к земле
7	7	nRESET	–	I	Общий сброс (активный низкий уровень)
8	8	GND	–	GND	Общий
9	9	VDDA	–	PWR	Питание аналоговых блоков
10	10, 11	GND	–	GND	Общий
11	12	XTO	ADDR	O	Выход тактовой частоты, вывод подключения резонатора. Шина адреса внешнего интерфейса
12	13	XTI	–	AI	Вход тактовой частоты, вывод подключения резонатора
13	14	PA[0]	U0_TXD	IO (ppu)	Порт А общего назначения. Интерфейс UART0. Выход передатчика
14	15	PA[1]	U0_RXD	IO (ppu)	Порт А общего назначения. Интерфейс UART0. Вход приемника
15	16	PA[2]	U1_TXD	IO (ppu)	Порт А общего назначения. Интерфейс UART1. Выход передатчика
			NF_RE		Интерфейс Nand Flash. Строб чтения
16	17	PA[3]	U1_RXD	IO (ppu)	Порт А общего назначения. Интерфейс UART1. Вход приемника
			NF_WE		Интерфейс Nand Flash. Строб записи
17	18	PA[4]	SPI0_CLK	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Синхросигнал
			NF_ALE		Интерфейс Nand Flash. Строб адреса
18	19	PA[5]	SPI0_DO	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Выход данных
			NF_D[0]		Интерфейс Nand Flash Бит данных 0
19	20	PA[6]	SPI0_DI	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Вход данных
			NF_D[1]		Интерфейс Nand Flash Бит данных 1
20	21	PA[7]	SPI0_CS[0]	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Выбор SPI устройства 0
21	22	PA[8]	SPI0_CS[1]	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Выбор SPI устройства 1
			NF_CS[1]		Интерфейс Nand Flash. Выборка модуля 1
22	23	PA[9]	SPI0_CS[2]	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Выбор SPI устройства 2
			NF_D[2]		Интерфейс NAND Flash. Бит данных 2

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
23	24	PA[10]	SPI0_CS[3]	IO (ppu)	Порт A общего назначения. Интерфейс SPI. Выбор SPI устройства 3
			NF_D[3]		Интерфейс NAND Flash. Бит данных 3
24	25	PA[11]	SPI0_CS[4]	IO (ppu)	Порт A общего назначения. Интерфейс SPI. Выбор SPI устройства 4
			NF_D[4]		Интерфейс NAND Flash. Бит данных 4
25	26	PA[12]	SPI0_CS[5]	IO (ppu)	Порт A общего назначения. Интерфейс SPI. Выбор SPI устройства 5
			NF_D[5]		Интерфейс NAND Flash. Бит данных 5
26	27	PA[13]	SSI0_TCLK	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Синхронигнал передатчика
			AC97_0_CLK		Интерфейс AC97_0. Синхросигнал
			SPI1_CLK		Интерфейс SPI1. Синхросигнал
			NF_CLE		Интерфейс NAND Flash. Строб команды
27	28, 29, 30	VDD_IO	–	PWR	Питание площадок ввода/вывода
28	31	PA[14]	SSI0_TFS	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Начало кадра или выбор канала левый-правый
			AC97_0_RST		Интерфейс AC97_0. Сброс
			SPI1_DO		Интерфейс SPI1. Выход данных
			NF_D[6]		Интерфейс NAND Flash. Бит данных 6
29	32	PA[15]	SSI0_TXD	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Данные передатчика
			AC97_0_SDO		Интерфейс AC97_0. Выходные данные
			SPI1_DI		Интерфейс SPI1. Вход данных
			NF_D[7]		Интерфейс NAND Flash. Бит данных 7
30	33	PA[16]	SSI0_RCLK	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Синхросигнал приемника
			SPI1_CS		Интерфейс SPI1. Выбор SPI устройства
31	34	PA[17]	SSI0_RFS	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Начало кадра или выбор канала левый-правый
			AC97_0_SYNC		Интерфейс AC97_0. Синхронизация фрейма
			NF_CS[0]		Интерфейс NAND Flash. Выборка модуля 0
32	35	PA[18]	SSI0_RXD	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Данные приемника
			AC97_0_SDI		Интерфейс AC97_0. Входные данные
			NF_RDY		Интерфейс NAND Flash. Вход готовности
33	36	PA[19]	SSI1_TCLK	IO (ppu)	Порт A общего назначения. Интерфейс SSI1. Синхросигнал передатчика
			AC97_1_CLK		Интерфейс AC97_1. Синхросигнал
34	37, 38	VDD	–	PWR	Питание ядра
35	39	PA[20]	SSI1_TFS	IO (ppu)	Порт A общего назначения. Интерфейс SSI1. Начало кадра или выбор канала левый-правый
			AC97_1_RST		Интерфейс AC97_1. Сброс

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
36	40	PA[21]	SSI1_TXD	IO (ppu)	Порт А общего назначения. Интерфейс SSI1. Данные передатчика
			AC97_1_SDO		Интерфейс AC97_1. Выходные данные
37	41	PA[22]	SSI1_RCLK	IO (ppu)	Порт А общего назначения. Интерфейс SSI1. Синхросигнал приемника
			GTMR0 CH0o		Таймер 0. Выход 0 ШИМ (+)
38	42	PA[23]	SSI1_RFS	IO (ppu)	Порт А общего назначения. Интерфейс SSI1. Начало кадра или выбор канала левый-правый
			AC97_1_SYNC		Интерфейс AC97_1. Синхронизация фрейма
			GTMR0 nCH0o		Таймер 0. Выход 0 ШИМ (-)
39	43	PA[24]	SSI1_RXD	IO (ppu)	Порт А общего назначения. Интерфейс SSI1. Данные приемника
			AC97_1_SDI		Интерфейс AC97_1. Входные данные
			GTMR0 CH1o		Таймер 0. Выход 1 ШИМ (+)
40	44	PA[25]	VC_CLK	IO (ppu)	Порт А общего назначения. Интерфейс видеокамеры. Синхросигнал
			GTMR0 nCH1o		Таймер 0. Выход 1 ШИМ (-)
41	45, 46	GND	–	GND	Общий
42	47	PA[26]	VC_VSYNC	IO (ppu)	Порт А общего назначения. Интерфейс видеокамеры. Вход вертикальной развертки
			MIL0_OU1P		Интерфейс МКПД0. Выход основного канала (прямой)
			GTMR0 CH2o		Таймер 0. Выход 2 ШИМ (+)
43	48	PA[27]	VC_HSYNC	IO (ppu)	Порт А общего назначения. Интерфейс видеокамеры. Вход горизонтальной развертки видеоинтерфейса
			MIL0_OU1N		Интерфейс МКПД0. Выход основного канала (инверсный)
			GTMR0 nCH2o		Таймер 0. Выход 2 ШИМ (-)
44	49	PA[28]	VC_DATA[0]	IO (ppu)	Порт А общего назначения. Интерфейс видеокамеры. Бит 0 данных
			MIL0_OU1X		Интерфейс МКПД0. Разрешение передачи основного канала
			GTMR0 CH3o		Таймер 0. Выход 3 ШИМ (+)
45	50	PA[29]	VC_DATA[1]	IO (ppu)	Порт А общего назначения. Интерфейс видеокамеры. Бит 1 данных
			MIL0_OU2P		Интерфейс МКПД0. Выход резервного канала (прямой)
			GTMR0 nCH3o		Таймер 0. Выход 3 ШИМ (-)
46	51	PA[30]	VC_DATA[2]	IO (ppu)	Порт А общего назначения. Интерфейс видеокамеры. Бит 2 данных
			MIL0_OU2N		Интерфейс МКПД0. Выход резервного канала (инверсный)
			GTMR0 BRK		Таймер 0. Блокировка выходов

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
47	52	PA[31]	VC_DATA[3]	IO (ppu)	Порт A общего назначения. Интерфейс видеокамеры. Бит 3 данных
			MIL0_OU2X		Интерфейс МКПД0. Разрешение передачи резервного канала
			GTMR0 ETR		Таймер 0. Универсальный регистратор событий
48	53	PB[0]	VC_DATA[4]	IO (ppu)	Порт B общего назначения. Интерфейс видеокамеры. Бит 4 данных
			MIL0_IN1P		Интерфейс МКПД0. Вход основного канала (прямой)
			SPI2_CLK		Интерфейс SPI2. Синхросигнал
			GTMR0 CH0i		Таймер 0. Вход 0 захвата
49	54	PB[1]	VC_DATA[5]	IO (ppu)	Порт B общего назначения. Интерфейс видеокамеры. Бит 5 данных
			MIL0_IN1N		Интерфейс МКПД0. Вход основного канала (инверсный)
			SPI2_DO		Интерфейс SPI2. Выход данных
			GTMR0 CH1i		Таймер 0. Вход 1 захвата
50	55	PB[2]	VC_DATA[6]	IO (ppu)	Порт B общего назначения. Интерфейс видеокамеры. Бит 6 данных
			MIL0_IN2P		Интерфейс МКПД0. Вход резервного канала (прямой)
			SPI2_DI		Интерфейс SPI2. Вход данных
			GTMR0 CH2i		Таймер 0. Вход 2 захвата
51	56	PB[3]	VC_DATA[7]	IO (ppu)	Порт B общего назначения. Интерфейс видеокамеры. Бит 7 данных
			MIL0_IN2N		Интерфейс МКПД0. Вход резервного канала (инверсный)
			SPI2_CS		Интерфейс SPI2. Выбор SPI устройства
			GTMR0 CH3i		Таймер 0. Вход 3 захвата
52	57	PB[4]	LC_B[0]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 0
			AR_IN1P		Интерфейс ARINC. Вход приемника 1 (прямой)
			nDMAR[0]		Запрос канала 0 контроллера ПДП
53	58	PB[5]	LC_B[1]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 1
			AR_IN1N		Интерфейс ARINC. Вход приемника 1 (инверсный)
			nDMAR[1]		Запрос канала 1 контроллера ПДП
54	59	PB[6]	LC_B[2]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 2
			AR_IN2P		Интерфейс ARINC. Вход приемника 2 (прямой)
			nDMAR[2]		Запрос канала 2 контроллера ПДП
55	60	PB[7]	LC_B[3]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 3
			AR_IN2N		Интерфейс ARINC. Вход приемника 2 (инверсный)
			nDMAR[3]		Запрос канала 3 контроллера ПДП
56	61	PB[8]	LC_B[4]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 4
			AR_IN3P		Интерфейс ARINC. Вход приемника 3 (прямой)
			GTMR1 CH0o		Таймер 1. Выход 0 ШИМ (+)

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
57	62	PB[9]	LC_B[5]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 5
			AR_IN3N		Интерфейс ARINC. Вход приемника 3 (инверсный)
			GTMR1 nCH0o		Таймер 1. Выход 0 ШИМ (-)
58	63	PB[10]	LC_G[0]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 0
			AR_IN4P		Интерфейс ARINC. Вход приемника 4 (прямой)
			GTMR1 CH1o		Таймер 1. Выход 1 ШИМ (+)
59	64	PB[11]	LC_G[1]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 1
			AR_IN4N		Интерфейс ARINC. Вход приемника 4 (инверсный)
			GTMR1 nCH1o		Таймер 1. Выход 1 ШИМ (-)
60	65	PB[12]	LC_G[2]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 2
			AR_IN5P		Интерфейс ARINC. Вход приемника 5 (прямой)
			GTMR1 CH2o		Таймер 1. Выход 2 ШИМ (+)
61	66	PB[13]	LC_G[3]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 3
			AR_IN5N		Интерфейс ARINC. Вход приемника 5 (инверсный)
			GTMR1 nCH2o		Таймер 1. Выход 2 ШИМ (-)
62	67	PB[14]	LC_G[4]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 4
			AR_IN6P		Интерфейс ARINC. Вход приемника 6 (прямой)
			GTMR1 CH3o		Таймер 1. Выход 3 ШИМ (+)
63	68	PB[15]	LC_G[5]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 5
			AR_IN6N		Интерфейс ARINC. Вход приемника 6 (инверсный)
			GTMR1 nCH3o		Таймер 1. Выход 3 ШИМ (-)
64	69	PB[16]	LC_R[0]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 0
			AR_IN7P		Интерфейс ARINC. Вход приемника 7 (прямой)
			nDMAR[1]		Запрос канала 1 контроллера ПДП
			GTMR1 BRK		Таймер 1. Блокировка выходов
65	70	PB[17]	LC_R[1]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 1
			AR_IN7N		Интерфейс ARINC. Вход приемника 7 (инверсный)
			GTMR1 ETR		Таймер 1. Универсальный регистратор событий
66	71	PB[18]	LC_R[2]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 2
			AR_IN8P		Интерфейс ARINC. Вход приемника 8 (прямой)
			GTMR1 CH0i		Таймер 1. Вход 0 захвата
67	72	PB[19]	LC_R[3]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 3
			AR_IN8N		Интерфейс ARINC. Вход приемника 8 (инверсный)
			GTMR1 CH1i		Таймер 1. Вход 1 захвата

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
68	73	PB[20]	LC_R[4]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 4
			AR_OU1P		Интерфейс ARINC. Выход передатчика 1 (прямой)
			GTMR1 CH2i		Таймер 1. Вход 2 захвата
69	74	PB[21]	LC_R[5]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 5
			AR_OU1N		Интерфейс ARINC. Выход передатчика 1 (инверсный)
			GTMR1 CH3i		Таймер 1. Вход 3 захвата
70	75	PB[22]	LC_T[0]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Дополнительный выход 0
			nIRQ[0]		Вход прерывания 0
			nDMAR[4]		Запрос канала 4 контроллера ПДП
71	76	PB[23]	LC_T[1]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Дополнительный выход 1
			nIRQ[1]		Вход прерывания 1
			nDMAR[5]		Запрос канала 5 контроллера ПДП
72	77	PB[24]	LC_T[2]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Дополнительный выход 2
			nIRQ[2]		Вход прерывания 2
			nDMAR[6]		Запрос канала 6 контроллера ПДП
73	78	PB[25]	LC_T[3]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Дополнительный выход 3
			nIRQ[3]		Вход прерывания 3
			nDMAR[7]		Запрос канала 7 контроллера ПДП
74	79	PB[26]	LC_PWM	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Выход ШИМ для управления яркостью ЖКИ
			AR_OU2P		Интерфейс ARINC. Выход передатчика 2 (прямой)
			nDMAR[2]		Запрос канала 2 контроллера ПДП
75	80, 81	GND	–	GND	Общий
76	82	PB[27]	LC_DRDY	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Сигнал готовности данных для ЖКИ
			AR_OU2N		Интерфейс ARINC. Выход передатчика 2 (инверсный)
			nDMAR[8]		Запрос канала 8 контроллера ПДП
77	83	PB[28]	LC_VSYNC	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Сигнал вертикальной синхронизации ЖКИ
			AR_OU3P		Интерфейс ARINC. Выход передатчика 3 (прямой)
			nDMAR[9]		Запрос канала 9 контроллера ПДП
78	84	PB[29]	LC_HSYNC	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Сигнал горизонтальной синхронизации ЖКИ
			AR_OU3N		Интерфейс ARINC. Выход передатчика 3 (инверсный)
			nDMAR[10]		Запрос канала 10 контроллера ПДП

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
79	85	PB[30]	LC_CLK	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Синхросигнал
			AR_OU4P		Интерфейс ARINC. Выход передатчика 4 (прямой)
			nDMAR[11]		Запрос канала 11 контроллера ПДП
80	86	PB[31]	nDMAR3	IO (ppu)	Порт B общего назначения. Запрос канала 3 контроллера ПДП
			AR_OU4N		Интерфейс ARINC. Выход передатчика 4 (инверсный)
81	87	VDD	–	PWR	Питание ядра
82	88	PC[0]	FLAG[0]	IO (ppu)	Порт C общего назначения. Управление флагом 0
83	89	PC[1]	FLAG[1]	IO (ppu)	Порт C общего назначения. Управление флагом 1
84	90	PC[2]	FLAG[2]	IO (ppu)	Порт C общего назначения. Управление флагом 2
85	91	PC[3]	FLAG[3]	IO (ppu)	Порт C общего назначения. Управление флагом 3
86	92	PC[4]	BOOT[0]	IO (ppu)	Порт C общего назначения. Выбор режима загрузки
			HCLK		Хост интерфейс. Синхросигнал
87	93	PC[5]	BOOT[1]	IO (ppu)	Порт C общего назначения. Выбор режима загрузки
			HDI		Хост интерфейс. Вход данных
88	94	PC[6]	BOOT[2]	IO (ppu)	Порт C общего назначения. Выбор режима загрузки
			HDO		Хост интерфейс. Выход данных
89	95	PC[7]	SCLK	O	Синхросигнал внешнего интерфейса
90	96, 97, 98	VDD_IO	–	PWR	Питание площадок ввода/вывода
91	99	PC[8]	SD_CKE	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Разрешение синхронизации
92	100	PC[9]	MSSD[0]	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Активация модуля 0
93	101	PC[10]	MSSD[1]	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Активация модуля 1
94	102	PC[11]	MSSD[2]	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Активация модуля 2
			I2C_SDA		Интерфейс I2C. SDA
95	103	PC[12]	MSSD[3]	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Активация модуля 3
			I2C_SCL		Интерфейс I2C. SCL
96	104	PC[13]	SD_nCAS	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Выбор колонки
97	105, 106	GND	–	GND	Общий
98	107	PC[14]	SD_nRAS	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Выбор строки
99	108	PC[15]	SD_nWE	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Признак записи
100	109	PC[16]	SD_DQM	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Маска

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
101	110	PC[17]	SD_A10	IO (ppu)	Порт С общего назначения. Интерфейс SDRAM. Бит 10 адреса
102	111	PC[18]	nMS[1]	IO (ppu)	Порт С общего назначения. Интерфейс статической памяти. Выбор типа 1
103	112	VDD	–	PWR	Питание ядра
104	113	PC[19]	nMS[0]	IO (ppu)	Порт С общего назначения. Интерфейс статической памяти. Выбор типа 0
105	114	PC[20]	nBMS	IO (ppu)	Порт С общего назначения. Интерфейс статической памяти. Выбор внешней памяти начальной загрузки
106	115	PC[21]	nRD	IO (ppu)	Порт С общего назначения. Интерфейс статической памяти. Строб чтения
107	116	PC[22]	nWR	IO (ppu)	Порт С общего назначения. Интерфейс статической памяти. Строб записи
108	117	PC[23]	ACK	IO (ppu)	Порт С общего назначения. Интерфейс статической памяти. Сигнал готовности
109	118	PE[21]	ADDR[21]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 21
110	119	PE[20]	ADDR[20]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 20
111	120	PE[19]	ADDR[19]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 19
112	121	PE[18]	ADDR[18]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 18
113	122	PE[17]	ADDR[17]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 17
114	123	PE[16]	ADDR[16]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 16
115	124	PE[15]	ADDR[15]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 15
116	125	PE[14]	ADDR[14]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 14
117	126	PE[13]	ADDR[13]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 13
118	127	PE[12]	ADDR[12]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 12
119	128	PE[11]	ADDR[11]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 11
120	129	PE[10]	ADDR[10]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 10
121	130	PE[9]	ADDR[9]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 9
122	131	PE[8]	ADDR[8]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 8
123	132	PE[7]	ADDR[7]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 7
124	133	VDD	–	PWR	Питание ядра
125	134	PE[6]	ADDR[6]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 6
126	135	PE[5]	ADDR[5]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 5
127	136	PE[4]	ADDR[4]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 4
128	137	PE[3]	ADDR[3]	O	Порт Е общего назначения. Шина адреса внешнего интерфейса. Бит 3

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
129	138	PE[2]	ADDR[2]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 2
130	139	PE[1]	ADDR[1]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 1
131	140	PE[0]	ADDR[0]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 0
132	141	PD[31]	DATA[31]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 31
			NF_RDY		Интерфейс NAND Flash. Вход готовности
133	142	PD[30]	DATA[30]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 30
			NF_CS[2]		Интерфейс NAND Flash. Выборка модуля 2
134	143	PD[29]	DATA[29]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 29
			NF_CS[1]		Интерфейс NAND Flash. Выборка модуля 1
135	144	PD[28]	DATA[28]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 28
			NF_CS[0]		Интерфейс NAND Flash. Выборка модуля 0
136	145	PD[27]	DATA[27]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 27
			NF_WE		Интерфейс NAND Flash. Подсветка команды
137	146, 147, 148	GND	–	GND	Общий
138	149	PD[26]	DATA[26]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 26
			NF_RE		Интерфейс NAND Flash. Подсветка адреса
139	150	PD[25]	DATA[25]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 25
			NF_ALE		Интерфейс NAND Flash. Строб чтения
			MIL1_OU2X		Интерфейс МКПД1. Разрешение передачи резервного канала
140	151	PD[24]	DATA[24]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 24
			NF_CLE		Интерфейс NAND Flash. Строб записи
			MIL1_OU2N		Интерфейс МКПД1. Выход резервного канала (инверсный)
141	152	PD[23]	DATA[23]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 23
			NF_D[7]		Интерфейс NAND Flash. Бит данных 7
			MIL1_OU2P		Интерфейс МКПД1. Выход резервного канала (прямой)
142	153	PD[22]	DATA[22]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 22
			NF_D[6]		Интерфейс NAND Flash. Бит данных 6
			MIL1_IN2N		Интерфейс МКПД1. Вход резервного канала (–)
143	154	PD[21]	DATA[21]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 21
			NF_D[5]		Интерфейс NAND Flash. Бит данных 5
			MIL1_IN2P		Интерфейс МКПД1. Вход резервного канала (прямой)

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
144	155	PD[20]	DATA[20]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 20
			NF_D[4]		Интерфейс NAND Flash. Бит данных 4
			MIL1_OU1X		Интерфейс МКПД1. Разрешение передачи основного канала
145	156	PD[19]	DATA[19]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 19
			NF_D[3]		Интерфейс NAND Flash. Бит данных 3
			MIL1_OU1N		Интерфейс МКПД1. Выход основного канала (–)
146	157	PD[18]	DATA[18]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 18
			NF_D[2]		Интерфейс NAND Flash. Бит данных 2
			MIL1_OU1P		Интерфейс МКПД1. Выход основного канала (прямой)
147	158	PD[17]	DATA[17]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 17
			NF_D[1]		Интерфейс NAND Flash. Бит данных 1
			MIL1_IN1N		Интерфейс МКПД1. Вход основного канала (–)
148	159	VDD	–	PWR	Питание ядра
149	160	PD[16]	DATA[16]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 16
			NF_D[0]		Интерфейс NAND Flash. Бит данных 0
			MIL1_IN1P		Интерфейс МКПД1. Вход основного канала (прямой)
150	161	PD[15]	DATA[15]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 15
151	162	PD[14]	DATA[14]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 14
152	163	PD[13]	DATA[13]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 13
153	164	PD[12]	DATA[12]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 12
154	165	PD[11]	DATA[11]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 11
155	166	PD[10]	DATA[10]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 10
156	167	PD[9]	DATA[9]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 9
157	168	PD[8]	DATA[8]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 8
158	169	PD[7]	DATA[7]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 7
159	170	PD[6]	DATA[6]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 6
160	171	PD[5]	DATA[5]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 5
161	172	PD[4]	DATA[4]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 4
162	173	PD[3]	DATA[3]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 3
163	174	PD[2]	DATA[2]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 2
164	175	PD[1]	DATA[1]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 1

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
165	176	PD[0]	DATA[0]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 0
166	177, 178	GND	–	GND	Общий
167	179	L0DON[0]	–	O	LINK порт 0. Выход данных (инверсный). Бит 0
168	180	L0DOP[0]	–	O	LINK порт 0. Выход данных (прямой). Бит 0
169	181	L0DON[1]	–	O	LINK порт 0. Выход данных (инверсный). Бит 1
170	182	L0DOP[1]	–	O	LINK порт 0. Выход данных (прямой). Бит 1
171	183	L0DON[2]	–	O	LINK порт 0. Выход данных (инверсный). Бит 2
172	184	L0DOP[2]	–	O	LINK порт 0. Выход данных (прямой). Бит 2
173	185, 186, 187	VDD_IO	–	PWR	Питание контактных площадок
174	188	L0DON[3]	–	O	LINK порт 0. Выход данных (инверсный). Бит 3
175	189	L0DOP[3]	–	O	LINK порт 0. Выход данных (прямой). Бит 3
176	190	L0DON[4]	–	O	LINK порт 0. Выход данных (инверсный). Бит 4
177	191	L0DOP[4]	–	O	LINK порт 0. Выход данных (прямой). Бит 4
178	192	VDD	–	PWR	Питание ядра
179	193	L0DON[5]	–	O	LINK порт 0. Выход данных (инверсный). Бит 5
180	194	L0DOP[5]	–	O	LINK порт 0. Выход данных (прямой). Бит 5
181	195	L0DON[6]	–	O	LINK порт 0. Выход данных (инверсный). Бит 6
182	196	L0DOP[6]	–	O	LINK порт 0. Выход данных (прямой). Бит 6
183	197	L0DON[7]	–	O	LINK порт 0. Выход данных (инверсный). Бит 7
184	198	L0DOP[7]	–	O	LINK порт 0. Выход данных (прямой). Бит 7
185	199	L0CLKON	–	O	LINK порт 0. Выход синхросигнала (инверсный)
186	200	L0CLKOP	–	O	LINK порт 0. Выход синхросигнала (прямой)
187	201	L0DIN[0]	–	I	LINK порт 0. Вход данных (инверсный). Бит 0
188	202	L0DIP[0]	–	I	LINK порт 0. Вход данных (прямой). Бит 0
189	203	L0DIN[1]	–	I	LINK порт 0. Вход данных (инверсный). Бит 1
190	204	L0DIP[1]	–	I	LINK порт 0. Вход данных (прямой). Бит 1
191	205	L0DIN[2]	–	I	LINK порт 0. Вход данных (инверсный). Бит 2
192	206	L0DIP[2]	–	I	LINK порт 0. Вход данных (прямой). Бит 2
193	207	L0DIN[3]	–	I	LINK порт 0. Вход данных (инверсный). Бит 3
194	208	L0DIP[3]	–	I	LINK порт 0. Вход данных (прямой). Бит 3
195	209	L0DIN[4]	–	I	LINK порт 0. Вход данных (инверсный). Бит 4
196	210	L0DIP[4]	–	I	LINK порт 0. Вход данных (прямой). Бит 4
197	211	PC[24]	L0ACKO GPS0_CLKO	IO (ppu)	Порт C общего назначения. LINK порт 0. Выход разрешения передачи Интерфейс GPS0. Выходной синхросигнал
198	212, 213	GND	–	GND	Общий
199	214	L0DIN[5]	–	I	LINK порт 0. Вход данных (инверсный). Бит 5
200	215	L0DIP[5]	–	I	LINK порт 0. Вход данных (прямой). Бит 5
201	216	L0DIN[6]	–	I	LINK порт 0. Вход данных (инверсный). Бит 6
202	217	L0DIP[6]	–	I	LINK порт 0. Вход данных (прямой). Бит 6
203	218	L0DIN[7]	–	I	LINK порт 0. Вход данных (инверсный). Бит 7

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
204	219	L0DIP[7]	–	I	LINK порт 0. Вход данных (прямой). Бит 7
205	220	L0CLKIN	–	I	LINK порт 0. Вход синхросигнала (инверсный)
206	221	L0CLKIP	–	I	LINK порт 0. Вход синхросигнала (прямой)
207	222	PC[25]	L0ACKI	IO (ppd)	Порт С общего назначения. LINK порт 0. Вход разрешения передачи
			GPS0_MAG		Интерфейс GPS0. Амплитуда
208	223	PC[26]	L0BCMPO	IO (ppu)	Порт С общего назначения. LINK порт 0. Выход окончания блока
209	224	PC[27]	L0BCMPI	IO (ppd)	Порт С общего назначения. LINK порт 0. Вход разрешения передачи
			GPS0_SIGN		Интерфейс GPS0. Знак
210	225	PC[28]	L1ACKO	IO (ppu)	Порт С общего назначения. LINK порт 1. Выход разрешения передачи
			GPS1_CLKO		Интерфейс GPS1. Выходной синхросигнал
211	226	PC[29]	L1ACKI	IO (ppd)	Порт С общего назначения. LINK порт 1. Вход разрешения передачи
			GPS1_MAG		Интерфейс GPS1. Амплитуда
212	227	PC[30]	L1BCMPO	IO (ppu)	Порт С общего назначения. LINK порт1. Выход окончания блока
213	228	VDD	–	PWR	Питание ядра
214	229	L1DON[0]	–	O	LINK порт 1. Выход данных (инверсный). Бит 0
215	230	L1DOP[0]	–	O	LINK порт 1. Выход данных (прямой). Бит 0
216	231	L1DON[1]	–	O	LINK порт 1. Выход данных (инверсный). Бит 1
217	232	L1DOP[1]	–	O	LINK порт 1. Выход данных (прямой). Бит 1
218	233	GND	–	GND	Общий
219	234	PC[31]	L1BCMPI	IO (ppd)	Порт С общего назначения LINK порт 1. Вход разрешения передачи
			GPS1_SIGN		Интерфейс GPS1. Знак
220	235	L1DON[2]	–	O	LINK порт 1. Выход данных (инверсный). Бит 2
221	236	L1DOP[2]	–	O	LINK порт 1. Выход данных (прямой). Бит 2
222	237	L1DON[3]	–	O	LINK порт 1. Выход данных (инверсный). Бит 3
223	238	L1DOP[3]	–	O	LINK порт 1. Выход данных (прямой). Бит 3
224	239, 240, 241	VDD_IO	–	PWR	Питание контактных площадок
225	242	L1DON[4]	–	O	LINK порт 1. Выход данных (инверсный). Бит 4
226	243	L1DOP[4]	–	O	LINK порт 1. Выход данных (прямой). Бит 4
227	244	L1DON[5]	–	O	LINK порт 1. Выход данных (инверсный). Бит 5
228	245	L1DOP[5]	–	O	LINK порт 1. Выход данных (прямой). Бит 5
229	246	L1DON[6]	–	O	LINK порт 1. Выход данных (инверсный). Бит 6
230	247	L1DOP[6]	–	O	LINK порт 1. Выход данных (прямой). Бит 6
231	248	L1DON[7]	–	O	LINK порт 1. Выход данных (инверсный). Бит 7
232	249	L1DOP[7]	–	O	LINK порт 1. Выход данных (прямой). Бит 7
233	250, 251	VDD	–	PWR	Питание ядра
234	252	L1CLKON	–	O	LINK порт 1. Выход синхросигнала (инверсный)
235	253	L1CLKOP	–	O	LINK порт 1. Выход синхросигнала (прямой)
236	254	L1DIN[0]	–	I	LINK порт 1. Вход данных (инверсный). Бит 0

№ вывода	№ КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
237	255	L1DIP[0]	–	I	LINK порт 1. Вход данных (прямой). Бит 0
238	256	GND	–	GND	Общий
239	257	L1DIN[1]	–	I	LINK порт 1. Вход данных (инверсный). Бит 1
240	258	L1DIP[1]	–	I	LINK порт 1. Вход данных (прямой). Бит 1
241	259	L1DIN[2]	–	I	LINK порт 1. Вход данных (инверсный). Бит 2
242	260	L1DIP[2]	–	I	LINK порт 1. Вход данных (прямой). Бит 2
243	261	L1DIN[3]	–	I	LINK порт 1. Вход данных (инверсный). Бит 3
244	262	L1DIP[3]	–	I	LINK порт 1. Вход данных (прямой). Бит 3
245	263	L1DIN[4]	–	I	LINK порт 1. Вход данных (инверсный). Бит 4
246	264	L1DIP[4]	–	I	LINK порт 1. Вход данных (прямой). Бит 4
247	265	L1DIN[5]	–	I	LINK порт 1. Вход данных (инверсный). Бит 5
248	266	L1DIP[5]	–	I	LINK порт 1. Вход данных (прямой). Бит 5
249	267	L1DIN[6]	–	I	LINK порт 1. Вход данных (инверсный). Бит 6
250	268	L1DIP[6]	–	I	LINK порт 1. Вход данных (прямой). Бит 6
251	269	L1DIN[7]	–	I	LINK порт 1. Вход данных (инверсный). Бит 7
252	270	L1DIP[7]	–	I	LINK порт 1. Вход данных (прямой). Бит 7
253	271	L1CLKIN	–	I	LINK порт 1. Вход синхросигнала (инверсный)
254	272	L1CLKIP	–	I	LINK порт 1. Вход синхросигнала (прямой)
255	273	nTRESET	–	I (pu)	Интерфейс JTAG. Сброс
256	274	GND	–	GND	Общий

* При использовании альтернативной функции вывод порта GPIO переходит под управление одного из интерфейсов процессора, работа которого в данный момент разрешена. Дополнительные назначения выводов указаны в таблице в порядке возрастания приоритета: сверху указано назначение с наименьшим приоритетом. В мультиплексировании участвуют только дополнительные назначения, используемые интерфейсом как выход.

Примечание – Обозначение типов выводов:

I – вход;
O – выход;
IO – вход/выход;
PWR – питание;
GND – общий;
pu – резистор доопределения до питания;
ppu – резистор доопределения до питания, включаемый программно. По сбросу включен;
pd – резистор доопределения до нуля;
ppd – резистор доопределения до нуля, включаемый программно. По сбросу включен.

Таблица 2 – Описание выводов микросхем в корпусе BGA288

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
A2	273	nTRESET	-	I(pu)	Интерфейс JTAG. Сброс
A3	270	L1DIP[7]	-	I	LINK порт 1. Вход данных (прямой). Бит 7
A4	267	L1DIN[6]	-	I	LINK порт 1. Вход данных (инверсный). Бит 6
A5	263	L1DIN[4]	-	I	LINK порт 1. Вход данных (инверсный). Бит 4
A6	258	L1DIP[1]	-	I	LINK порт 1. Вход данных (прямой). Бит 1
A7	252	L1CLKON	-	O	LINK порт 1. Выход синхросигнала (инверсный)
A8	245	L1DOP[5]	-	O	LINK порт 1. Выход данных (прямой). Бит 5
A9	244	L1DON[5]	-	O	LINK порт 1. Выход данных (инверсный). Бит 5
A10	236	L1DOP[2]	-	O	LINK порт 1. Выход данных (прямой). Бит 2
A11	230	L1DOP[0]	-	O	LINK порт 1. Выход данных (прямой). Бит 0
A12	224	PC[27]	L0BCMPI	IO (ppd)	Порт С общего назначения. LINK порт 0. Вход разрешения передачи
			GPS0_SIGN		Интерфейс GPS0. Знак
A13	219	L0DIP[7]	-	I	LINK порт 0. Вход данных (прямой). Бит 7
A14	215	L0DIP[5]	-	I	LINK порт 0. Вход данных (прямой). Бит 5
A15	210	L0DIP[4]	-	I	LINK порт 0. Вход данных (прямой). Бит 4
A16	208	L0DIP[3]	-	I	LINK порт 0. Вход данных (прямой). Бит 3
A17	207	L0DIN[3]	-	I	LINK порт 0. Вход данных (инверсный). Бит 3
B1	1	TCK	-	I (pu)	Интерфейс JTAG. Синхросигнал
B2	272	L1CLKIP	-	I	LINK порт 1. Вход синхросигнала (прямой)
B3	271	L1CLKIN	-	I	LINK порт 1. Вход синхросигнала (инверсный)
B4	268	L1DIP[6]	-	I	LINK порт 1. Вход данных (прямой). Бит 6
B5	264	L1DIP[4]	-	I	LINK порт 1. Вход данных (прямой). Бит 4
B6	257	L1DIN[1]	-	I	LINK порт 1. Вход данных (инверсный). Бит 1
B7	253	L1CLKOP	-	O	LINK порт 1. Выход синхросигнала (прямой)
B8	246	L1DON[6]	-	O	LINK порт 1. Выход данных (инверсный). Бит 6
B9	243	L1DOP[4]	-	O	LINK порт 1. Выход данных (прямой). Бит 4
B10	235	L1DON[2]	-	O	LINK порт 1. Выход данных (инверсный). Бит 2
B11	229	L1DON[0]	-	O	LINK порт 1. Выход данных (инверсный). Бит 0
B12	223	PC[26]	L0BCMPO	IO (ppu)	Порт С общего назначения. LINK порт 0. Выход окончания блока
B13	218	L0DIN[7]	-	I	LINK порт 0. Вход данных (инверсный). Бит 7
B14	214	L0DIN[5]	-	I	LINK порт 0. Вход данных (инверсный). Бит 5
B15	209	L0DIN[4]	-	I	LINK порт 0. Вход данных (инверсный). Бит 4
B16	206	L0DIP[2]	-	I	LINK порт 0. Вход данных (прямой). Бит 2
B17	205	L0DIN[2]	-	I	LINK порт 0. Вход данных (инверсный). Бит 2
C1	4	TDO	-	O	Интерфейс JTAG. Выход данных
C2	3	TDI	-	I (pu)	Интерфейс JTAG. Вход данных
C3	2	TMS	-	I (pu)	Интерфейс JTAG. Выбор режима
C4	269	L1DIN[7]	-	I	LINK порт 1. Вход данных (инверсный). Бит 7
C5	265	L1DIN[5]	-	I	LINK порт 1. Вход данных (инверсный). Бит 5
C6	259	L1DIN[2]	-	I	LINK порт 1. Вход данных (инверсный). Бит 2
C7	254	L1DIN[0]	-	I	LINK порт 1. Вход данных (инверсный). Бит 0
C8	247	L1DOP[6]	-	O	LINK порт 1. Выход данных (прямой). Бит 6
C9	242	L1DON[4]	-	O	LINK порт 1. Выход данных (инверсный). Бит 4
C10	234	PC[31]	L1BCMPI	IO (ppd)	Порт С общего назначения LINK порт 1. Вход разрешения передачи
			GPS1_SIGN		Интерфейс GPS1. Знак
C11	227	PC[30]	L1BCMPO	IO (ppu)	Порт С общего назначения. LINK порт1. Выход окончания блока

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
C12	222	PC[25]	L0ACKI	IO (ppd)	Порт С общего назначения. LINK порт 0. Вход разрешения передачи
			GPS0_MAG		Интерфейс GPS0. Амплитуда
C13	217	L0DIP[6]	-	I	LINK порт 0. Вход данных (прямой). Бит 6
C14	211	PC[24]	L0ACKO	IO (ppu)	Порт С общего назначения. LINK порт 0. Выход разрешения передачи
			GPS0_CLKO		Интерфейс GPS0. Выходной синхросигнал
C15	204	L0DIP[1]	-	I	LINK порт 0. Вход данных (прямой). Бит 1
C16	201	L0DIN[0]	-	I	LINK порт 0. Вход данных (инверсный). Бит 0
C17	202	L0DIP[0]	-	I	LINK порт 0. Вход данных (прямой). Бит 0
D1	9	VDDA	-	PWR	Питание аналоговых блоков
D2	7	nRESET	-	I	Общий сброс (активный низкий уровень)
D3	6	JG_MX	-	I (pd)	Выбор блока JTAG
D4	5	nEMU	-	O	Индикатор режима отладки
D5	266	L1DIP[5]	-	I	LINK порт 1. Вход данных (прямой). Бит 5
D6	260	L1DIP[2]	-	I	LINK порт 1. Вход данных (прямой). Бит 2
D7	255	L1DIP[0]	-	I	LINK порт 1. Вход данных (прямой). Бит 0
D8	248	L1DON[7]	-	O	LINK порт 1. Выход данных (инверсный). Бит 7
D9	238	L1DOP[3]	-	O	LINK порт 1. Выход данных (прямой). Бит 3
D10	232	L1DOP[1]	-	O	LINK порт 1. Выход данных (прямой). Бит 1
D11	226	PC[29]	L1ACKI	IO (ppd)	Порт С общего назначения. LINK порт 1. Вход разрешения передачи
			GPS1_MAG		Интерфейс GPS1. Амплитуда
D12	221	L0CLKIP	-	I	LINK порт 0. Вход синхросигнала (прямой)
D13	216	L0DIN[6]	-	I	LINK порт 0. Вход данных (инверсный). Бит 6
D14	200	L0CLKOP	-	O	LINK порт 0. Выход синхросигнала (прямой)
D15	203	L0DIN[1]	-	I	LINK порт 0. Вход данных (инверсный). Бит 1
D16	197	L0DON[7]	-	O	LINK порт 0. Выход данных (инверсный). Бит 7
D17	198	L0DOP[7]	-	O	LINK порт 0. Выход данных (прямой). Бит 7
E1	16	PA[2]	U1_TXD	IO (ppu)	Порт А общего назначения. Интерфейс UART1. Выход передатчика
			NF_RE		Интерфейс Nand Flash. Строб чтения
E2	15	PA[1]	U0_RXD	IO (ppu)	Порт А общего назначения. Интерфейс UART0. Вход приемника
E3	14	PA[0]	U0_TXD	IO (ppu)	Порт А общего назначения. Интерфейс UART0. Выход передатчика
E4	13	OSCI	-	I	Вход тактовой частоты, вывод подключения резонатора
E5	12	OSCO	ADDR	O	Выход тактовой частоты, вывод подключения резонатора. Шина адреса внешнего интерфейса
E6	262	L1DIP[3]	-	I	LINK порт 1. Вход данных (прямой). Бит 3
E7	261	L1DIN[3]	-	I	LINK порт 1. Вход данных (инверсный). Бит 3
E8	249	L1DOP[7]	-	O	LINK порт 1. Выход данных (прямой). Бит 7
E9	237	L1DON[3]	-	O	LINK порт 1. Выход данных (инверсный). Бит 3
E10	231	L1DON[1]	-	O	LINK порт 1. Выход данных (инверсный). Бит 1
E11	225	PC[28]	L1ACKO	IO (ppu)	Порт С общего назначения. LINK порт 1. Выход разрешения передачи
			GPS1_CLKO		Интерфейс GPS1. Выходной синхросигнал
E12	220	L0CLKIN	-	I	LINK порт 0. Вход синхросигнала (инверсный)
E13	199	L0CLKON	-	O	LINK порт 0. Выход синхросигнала (инверсный)
E14	196	L0DOP[6]	-	O	LINK порт 0. Выход данных (прямой). Бит 6
E15	195	L0DON[6]	-	O	LINK порт 0. Выход данных (инверсный). Бит 6

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
E16	194	L0DOP[5]	-	O	LINK порт 0. Выход данных (прямой). Бит 5
E17	193	L0DON[5]	-	O	LINK порт 0. Выход данных (инверсный). Бит 5
F1	21	PA[7]	SPI0_CS[0]	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Выбор SPI устройства 0
F2	20	PA[6]	SPI0_DI	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Вход данных
			NF_D[1]		Интерфейс Nand Flash Бит данных 1
F3	19	PA[5]	SPI0_DO	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Выход данных
			NF_D[0]		Интерфейс Nand Flash Бит данных 0
F4	18	PA[4]	SPI0_CLK	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Синхросигнал
			NF_ALE		Интерфейс Nand Flash. Строб адреса
F5	17	PA[3]	U1_RXD	IO (ppu)	Порт А общего назначения. Интерфейс UART1. Вход приемника
			NF_WE		Интерфейс Nand Flash. Строб записи
F6	112	VDD	-	PWR	Питание ядра
F7	250	VDD	-	PWR	Питание ядра
F8	185	VDD_IO	-	PWR	Питание контактных площадок
F9	187	VDD_IO	-	PWR	Питание контактных площадок
F10 - F12	251	VDD	-	PWR	Питание ядра
F13	191	L0DOP[4]	-	O	LINK порт 0. Выход данных (прямой). Бит 4
F14	190	L0DON[4]	-	O	LINK порт 0. Выход данных (инверсный). Бит 4
F15	189	L0DOP[3]	-	O	LINK порт 0. Выход данных (прямой). Бит 3
F16	188	L0DON[3]	-	O	LINK порт 0. Выход данных (инверсный). Бит 3
F17	184	L0DOP[2]	-	O	LINK порт 0. Выход данных (прямой). Бит 2
G1	25	PA[11]	SPI0_CS[4]	IO (ppu)	Порт А общего назначения. Интерфейс SPI. Выбор SPI устройства 4
			NF_D[4]		Интерфейс NAND Flash. Бит данных 4
G2	24	PA[10]	SPI0_CS[3]	IO (ppu)	Порт А общего назначения. Интерфейс SPI. Выбор SPI устройства 3
			NF_D[3]		Интерфейс NAND Flash. Бит данных 3
G3	23	PA[9]	SPI0_CS[2]	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Выбор SPI устройства 2
			NF_D[2]		Интерфейс NAND Flash. Бит данных 2
G4	22	PA[8]	SPI0_CS[1]	IO (ppu)	Порт А общего назначения. Интерфейс SPI0. Выбор SPI устройства 1
			NF_CS[1]		Интерфейс Nand Flash. Выборка модуля 1
G5	28	VDD_IO	-	PWR	Питание площадок ввода/вывода
G6	133	VDD	-	PWR	Питание ядра
G7	147	GND	-	GND	Общий
G8	8	GND	-	GND	Общий
G9	80	GND	-	GND	Общий
G10	213	GND	-	GND	Общий
G11	274	GND	-	GND	Общий
G12	240	VDD_IO	-	PWR	Питание контактных площадок
G13	179	L0DON[0]	-	O	LINK порт 0. Выход данных (инверсный). Бит 0
G14	180	L0DOP[0]	-	O	LINK порт 0. Выход данных (прямой). Бит 0
G15	181	L0DON[1]	-	O	LINK порт 0. Выход данных (инверсный). Бит 1
G16	182	L0DOP[1]	-	O	LINK порт 0. Выход данных (прямой). Бит 1
G17	183	L0DON[2]	-	O	LINK порт 0. Выход данных (инверсный). Бит 2

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
H1	32	PA[15]	SSI0_TXD	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Данные передатчика
			AC97_0_SDO		Интерфейс AC97_0. Выходные данные
			SPI1_DI		Интерфейс SPI1. Вход данных
			NF_D[7]		Интерфейс NAND Flash. Бит данных 7
H2	31	PA[14]	SSI0_TFS	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Начало кадра или выбор канала левый-правый
			AC97_0_RST		Интерфейс AC97_0. Сброс
			SPI1_DO		Интерфейс SPI1. Выход данных
			NF_D[6]		Интерфейс NAND Flash. Бит данных 6
H3	27	PA[13]	SSI0_TCLK	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Синхросигнал передатчика
			AC97_0_CLK		Интерфейс AC97_0. Синхросигнал
			SPI1_CLK		Интерфейс SPI1. Синхросигнал
			NF_CLE		Интерфейс NAND Flash. Строб команды
H4	26	PA[12]	SPI0_CS[5]	IO (ppu)	Порт A общего назначения. Интерфейс SPI. Выбор SPI устройства 5
			NF_D[5]		Интерфейс NAND Flash. Бит данных 5
H5	29	VDD_IO	-	PWR	Питание площадок ввода/вывода
H6	96	VDD_IO	-	PWR	Питание площадок ввода/вывода
H7	148	GND	-	GND	Общий
H8	10	GND	-	GND	Общий
H9	81	GND	-	GND	Общий
H10	233	GND	-	GND	Общий
H11	274	GND	-	GND	Общий
H12	241	VDD_IO	-	PWR	Питание контактных площадок
H13	172	PD[4]	DATA[4]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 4
H14	173	PD[3]	DATA[3]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 3
H15	174	PD[2]	DATA[2]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 2
H16	175	PD[1]	DATA[1]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 1
H17	176	PD[0]	DATA[0]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 0
J1	34	PA[17]	SSI0_RFS	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Начало кадра или выбор канала левый-правый
			AC97_0_SYNC		Интерфейс AC97_0. Синхронизация фрейма
			NF_CS[0]		Интерфейс NAND Flash. Выборка модуля 0
J2	33	PA[16]	SSI0_RCLK	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Синхросигнал приемника
			SPI1_CS		Интерфейс SPI1. Выбор SPI устройства
J3	35	PA[18]	SSI0_RXD	IO (ppu)	Порт A общего назначения. Интерфейс SSI0. Данные приемника
			AC97_0_SDI		Интерфейс AC97_0. Входные данные
			NF_RDY		Интерфейс NAND Flash. Вход готовности
J4	36	PA[19]	SSI1_TCLK	IO (ppu)	Порт A общего назначения. Интерфейс SSI1. Синхросигнал передатчика
			AC97_1_CLK		Интерфейс AC97_1. Синхросигнал
J5	30	VDD_IO	-	PWR	Питание площадок ввода/вывода
J6	97	VDD_IO	-	PWR	Питание площадок ввода/вывода
J7	177	GND	-	GND	Общий

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
J8	11	GND	-	GND	Общий
J9	105	GND	-	GND	Общий
J10	256	GND	-	GND	Общий
J11	274	GND	-	GND	Общий
J12	241	VDD_IO	-	PWR	Питание площадок ввода/вывода
J13	167	PD[9]	DATA[9]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 9
J14	168	PD[8]	DATA[8]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 8
J15	169	PD[7]	DATA[7]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 7
J16	170	PD[6]	DATA[6]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 6
J17	171	PD[5]	DATA[5]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 5
K1	40	PA[21]	SSI1_TXD	IO (ppu)	Порт A общего назначения. Интерфейс SSI1. Данные передатчика
			AC97_1_SDO		Интерфейс AC97_1. Выходные данные
K2	39	PA[20]	SSI1_TFS	IO (ppu)	Порт A общего назначения. Интерфейс SSI1. Начало кадра или выбор канала левый-правый
			AC97_1_RST		Интерфейс AC97_1. Сброс
K3	41	PA[22]	SSI1_RCLK	IO (ppu)	Порт A общего назначения. Интерфейс SSI1. Синхросигнал приемника
			GTMR0 CH0o		Таймер 0. Выход 0 ШИМ (+)
K4	42	PA[23]	SSI1_RFS	IO (ppu)	Порт A общего назначения. Интерфейс SSI1. Начало кадра или выбор канала левый-правый
			AC97_1_SYNC		Интерфейс AC97_1. Синхронизация фрейма
K5	37	VDD	-	PWR	Питание ядра
K6	159	VDD	-	PWR	Питание ядра
K7	178	GND	-	GND	Общий
K8	45	GND	-	GND	Общий
K9	106	GND	-	GND	Общий
K10, K11	274	GND	-	GND	Общий
K12	251	VDD	-	PWR	Питание ядра
K13	162	PD[14]	DATA[14]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 14
K14	163	PD[13]	DATA[13]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 13
K15	164	PD[12]	DATA[12]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 12
K16	165	PD[11]	DATA[11]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 11
K17	166	PD[10]	DATA[10]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 10
L1	44	PA[25]	VC_CLK	IO (ppu)	Порт A общего назначения. Интерфейс видеокамеры. Синхросигнал
			GTMR0 nCH1o		Таймер 0. Выход 1 ШИМ (-)
L2	43	PA[24]	SSI1_RXD	IO (ppu)	Порт A общего назначения. Интерфейс SSI1. Данные приемника
			AC97_1_SDI		Интерфейс AC97_1. Входные данные
			GTMR0 nCH0o		Таймер 0. Выход 0 ШИМ (-)

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
L3	47	PA[26]	VC_VSYNC	IO (ppu)	Порт A общего назначения. Интерфейс видеокамеры. Вход вертикальной развертки
			MIL1_OU1P		Интерфейс МКПД0. Выход основного канала (прямой)
			GTMR0 CH2o		Таймер 0. Выход 2 ШИМ (+)
L4	48	PA[27]	VC_HSYNC	IO (ppu)	Порт A общего назначения. Интерфейс видеокамеры. Вход горизонтальной развертки видеоинтерфейса
			MIL0_OU1N		Интерфейс МКПД0. Выход основного канала (инверсный)
			GTMR0 nCH2o		Таймер 0. Выход 2 ШИМ (-)
L5	38	VDD	-	PWR	Питание ядра
L6	192	VDD	-	PWR	Питание ядра
L7	212	GND	-	GND	Общий
L8	46	GND	-	GND	Общий
L9	146	GND	-	GND	Общий
L10, L11	274	GND	-	GND	Общий
L12	251	VDD	-	PWR	Питание ядра
L13	156	PD[19]	DATA[19]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 19
			NF_D[3]		Интерфейс NAND Flash. Бит данных 3
			MIL1_OU1N		Интерфейс МКПД1. Выход основного канала (–)
L14	157	PD[18]	DATA[18]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 18
			NF_D[2]		Интерфейс NAND Flash. Бит данных 2
			MIL1_OU1P		Интерфейс МКПД1. Выход основного канала (прямой)
L15	158	PD[17]	DATA[17]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 17
			NF_D[1]		Интерфейс NAND Flash. Бит данных 1
			MIL1_IN1N		Интерфейс МКПД1. Вход основного канала (–)
L16	160	PD[16]	DATA[16]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 16
			NF_D[0]		Интерфейс NAND Flash. Бит данных 0
			MIL1_IN1P		Интерфейс МКПД1. Вход основного канала (прямой)
L17	161	PD[15]	DATA[15]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 15
M1	50	PA[29]	VC_DATA[1]	IO (ppu)	Порт A общего назначения. Интерфейс видеокамеры. Бит 1 данных
			MIL0_OU2P		Интерфейс МКПД0. Выход резервного канала (прямой)
			GTMR0 nCH3o		Таймер 0. Выход 3 ШИМ (-)
M2	51	PA[30]	VC_DATA[2]	IO (ppu)	Порт A общего назначения. Интерфейс видеокамеры. Бит 2 данных
			MIL0_OU2N		Интерфейс МКПД0. Выход резервного канала (инверсный)
			GTMR0 BRK		Таймер 0. Блокировка выходов

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
M3	52	PA[31]	VC_DATA[3]	IO (ppu)	Порт A общего назначения. Интерфейс видеокамеры. Бит 3 данных
			MIL0_OU2X		Интерфейс МКПД0. Разрешение передачи резервного канала
			GTMR0 ETR		Таймер 0. Универсальный регистратор событий
M4	49	PA[28]	VC_DATA[0]	IO (ppu)	Порт A общего назначения. Интерфейс видеокамеры. Бит 0 данных
			MIL0_OU1X		Интерфейс МКПД0. Разрешение передачи основного канала
			GTMR0 CH3o		Таймер 0. Выход 3 ШИМ (+)
M5	87	VDD	-	PWR	Питание ядра
M6	228	VDD	-	PWR	Питание ядра
M7	98	VDD_IO	-	PWR	Питание площадок ввода/вывода
M8	186	VDD_IO	-	PWR	Питание контактных площадок
M9, M10	239	VDD_IO	-	PWR	Питание контактных площадок
M11, M12	251	VDD	-	PWR	Питание ядра
M13	151	PD[24]	DATA[24]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 24
			NF_CLE		Интерфейс NAND Flash. Строб записи
			MIL1_OU2N		Интерфейс МКПД1. Выход резервного канала (инверсный)
M14	152	PD[23]	DATA[23]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 23
			NF_D[7]		Интерфейс NAND Flash. Бит данных 7
			MIL1_OU2P		Интерфейс МКПД1. Выход резервного канала (прямой)
M15	153	PD[22]	DATA[22]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 22
			NF_D[6]		Интерфейс NAND Flash. Бит данных 6
			MIL1_IN2N		Интерфейс МКПД1. Вход резервного канала (–)
M16	154	PD[21]	DATA[21]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 21
			NF_D[5]		Интерфейс NAND Flash. Бит данных 5
			MIL1_IN2P		Интерфейс МКПД1. Вход резервного канала (прямой)
M17	155	PD[20]	DATA[20]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 20
			NF_D[4]		Интерфейс NAND Flash. Бит данных 4
			MIL1_OU1X		Интерфейс МКПД1. Разрешение передачи основного канала
N1	54	PB[1]	VC_DATA[5]	IO (ppu)	Порт B общего назначения. Интерфейс видеокамеры. Бит 5 данных
			MIL0_IN1N		Интерфейс МКПД0. Вход основного канала (инверсный)
			SPI2_DO		Интерфейс SPI2. Выход данных
			GTMR0 CH1i		Таймер 0. Вход 1 захвата

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
N2	53	PB[0]	VC_DATA[4]	IO (ppu)	Порт B общего назначения. Интерфейс видеокамеры. Бит 4 данных
			MIL0_IN1P		Интерфейс МКПД0. Вход основного канала (прямой)
			SPI2_CLK		Интерфейс SPI2. Синхросигнал
			GTMR0 CH0i		Таймер 0. Вход 0 захвата
N3	55	PB[2]	VC_DATA[6]	IO (ppu)	Порт B общего назначения. Интерфейс видеокамеры. Бит 6 данных
			MIL0_IN2P		Интерфейс МКПД0. Вход резервного канала (прямой)
			SPI2_DI		Интерфейс SPI2. Вход данных
			GTMR0 CH2i		Таймер 0. Вход 2 захвата
N4	56	PB[3]	VC_DATA[7]	IO (ppu)	Порт B общего назначения. Интерфейс видеокамеры. Бит 7 данных
			MIL0_IN2N		Интерфейс МКПД0. Вход резервного канала (инверсный)
			SPI2_CS		Интерфейс SPI2. Выбор SPI устройства
			GTMR0 CH3i		Таймер 0. Вход 3 захвата
N5	57	PB[4]	LC_B[0]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 0
			AR_IN1P		Интерфейс ARINC. Вход приемника 1 (прямой)
			nDMAR[0]		Запрос канала 0 контроллера ПДП
N6	78	PB[25]	LC_T[3]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Дополнительный выход 3
			nIRQ[3]		Вход прерывания 3
			nDMAR[7]		Запрос канала 7 контроллера ПДП
N7	85	PB[30]	LC_CLK	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Синхросигнал
			AR_OU4P		Интерфейс ARINC. Выход передатчика 4 (прямой)
			nDMAR[11]		Запрос канала 11 контроллера ПДП
N8	95	PC[7]	SCLK	O	Порт C общего назначения. Синхросигнал внешнего интерфейса
N9	103	PC[12]	MSSD[3]	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Активация модуля 3
			I2C_SCL		Интерфейс I2C. SCL
N10	110	PC[17]	SD_A10	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Бит 10 адреса
N11	116	PC[22]	nWR	IO (ppu)	Порт C общего назначения. Интерфейс статической памяти. Строб записи
N12	121	PE[18]	ADDR[18]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 18
N13	126	PE[13]	ADDR[13]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 13
N14	144	PD[28]	DATA[28]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 28
			NF_CS[0]		Интерфейс NAND Flash. Выборка модуля 0
N15	145	PD[27]	DATA[27]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 27
			NF_WE		Интерфейс NAND Flash. Подсветка команды
N16	149	PD[26]	DATA[26]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 26
			NF_RE		Интерфейс NAND Flash. Подсветка адреса

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
N17	150	PD[25]	DATA[25]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 25
			NF_ALE		Интерфейс NAND Flash. Строб чтения
			MIL1_OU2X		Интерфейс МКПД1. Разрешение передачи резервного канала
P1	59	PB[6]	LC_B[2]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 2
			AR_IN2P		Интерфейс ARINC. Вход приемника 2 (прямой)
			nDMAR[2]		Запрос канала 2 контроллера ПДП
P2	58	PB[5]	LC_B[1]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 1
			AR_IN1N		Интерфейс ARINC. Вход приемника 1 (инверсный)
			nDMAR[1]		Запрос канала 1 контроллера ПДП
P3	60	PB[7]	LC_B[3]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 3
			AR_IN2N		Интерфейс ARINC. Вход приемника 2 (инверсный)
			nDMAR[3]		Запрос канала 3 контроллера ПДП
P4	61	PB[8]	LC_B[4]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 4
			AR_IN3P		Интерфейс ARINC. Вход приемника 3 (прямой)
			GTMR1 CH0o		Таймер 1. Выход 0 ШИМ (+)
P5	74	PB[21]	LC_R[5]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 5
			AR_OU1P		Интерфейс ARINC. Выход передатчика 1 (инверсный)
			GTMR1 CH3i		Таймер 1. Вход 3 захвата
P6	79	PB[26]	LC_PWM	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Выход ШИМ для управления яркостью ЖКИ
			AR_OU2P		Интерфейс ARINC. Выход передатчика 2 (прямой)
			nDMAR[2]		Запрос канала 2 контроллера ПДП
P7	86	PB[31]	nDMAR3	IO (ppu)	Порт B общего назначения. Запрос канала 3 контроллера ПДП
			AR_OU4N		Интерфейс ARINC. Выход передатчика 4 (инверсный)
P8	94	PC[6]	BOOT[2]	IO (ppu)	Порт C общего назначения. Выбор режима загрузки
			HDO		Хост интерфейс. Выход данных
P9	102	PC[11]	MSSD[2]	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Активация модуля 2
			I2C_SDA		Интерфейс I2C. SDA
P10	109	PC[16]	SD_DQM	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Маска
P11	115	PC[21]	nRD	IO (ppu)	Порт C общего назначения. Интерфейс статической памяти. Строб чтения
P12	120	PE[19]	ADDR[19]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 19
P13	125	PE[14]	ADDR[14]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 14

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
P14	130	PE[9]	ADDR[9]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 9
P15	141	PD[31]	DATA[31]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 31
			NF_RDY		Интерфейс NAND Flash. Вход готовности
P16	142	PD[30]	DATA[30]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 30
			NF_CS[2]		Интерфейс NAND Flash. Выборка модуля 2
P17	143	PD[29]	DATA[29]	IO (ppu)	Порт D общего назначения. Шина данных внешнего интерфейса. Бит 29
			NF_CS[1]		Интерфейс NAND Flash. Выборка модуля 1
R1	64	PB[11]	LC_G[1]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 1
			AR_IN4N		Интерфейс ARINC. Вход приемника 4 (инверсный)
			GTMR1 nCH1o		Таймер 1. Выход 1 ШИМ (-)
R2	63	PB[10]	LC_G[0]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 0
			AR_IN4P		Интерфейс ARINC. Вход приемника 4 (прямой)
			GTMR1 CH1o		Таймер 1. Выход 1 ШИМ (+)
R3	62	PB[9]	LC_B[5]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит синего цвета 5
			AR_IN3N		Интерфейс ARINC. Вход приемника 3 (инверсный)
			GTMR1 nCH0o		Таймер 1. Выход 0 ШИМ (-)
R4	71	PB[18]	LC_R[2]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 2
			AR_IN8P		Интерфейс ARINC. Вход приемника 8 (прямой)
			GTMR1 CH0i		Таймер 1. Вход 0 захвата
R5	75	PB[22]	LC_T[0]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Дополнительный выход 0
			nIRQ[0]		Вход прерывания 0
			nDMAR[4]		Запрос канала 4 контроллера ПДП
R6	82	PB[27]	LC_DRDY	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Сигнал готовности данных для ЖКИ
			AR_OU2N		Интерфейс ARINC. Выход передатчика 2 (инверсный)
			nDMAR[8]		Запрос канала 8 контроллера ПДП
R7	88	PC[0]	FLAG[0]	IO (ppu)	Порт C общего назначения. Управление флагом 0
R8	93	PC[5]	BOOT[1]	IO (ppu)	Порт C общего назначения. Выбор режима загрузки
			HDI		Хост интерфейс. Вход данных
R9	101	PC[10]	MSSD[1]	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Активация модуля 1
R10	108	PC[15]	SD_nWE	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Признак записи
R11	114	PC[20]	nBMS	O	Порт C общего назначения. Интерфейс статической памяти. Выбор внешней памяти начальной загрузки
R12	119	PE[20]	ADDR[20]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 20

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
R13	124	PE[15]	ADDR[15]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 15
R14	129	PE[10]	ADDR[10]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 10
R15	134	PE[6]	ADDR[6]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 6
R16	137	PE[3]	ADDR[3]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 3
R17	140	PE[0]	ADDR[0]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 0
T1	66	PB[13]	LC_G[3]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 3
			AR_IN5N		Интерфейс ARINC. Вход приемника 5 (инверсный)
			GTMR1 nCH2o		Таймер 1. Выход 2 ШИМ (-)
T2	65	PB[12]	LC_G[2]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 2
			AR_IN5P		Интерфейс ARINC. Вход приемника 5 (прямой)
			GTMR1 CH2o		Таймер 1. Выход 2 ШИМ (+)
T3	67	PB[14]	LC_G[4]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 4
			AR_IN6P		Интерфейс ARINC. Вход приемника 6 (прямой)
			GTMR1 CH3o		Таймер 1. Выход 3 ШИМ (+)
T4	72	PB[19]	LC_R[3]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 3
			AR_IN8N		Интерфейс ARINC. Вход приемника 8 (инверсный)
			GTMR1 CH1i		Таймер 1. Вход 1 захвата
T5	76	PB[23]	LC_T[1]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Дополнительный выход 1
			nIRQ[1]		Вход прерывания 1
			nDMAR[5]		Запрос канала 5 контроллера ПДП
T6	83	PB[28]	LC_VSYNC	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Сигнал вертикальной синхронизации ЖКИ
			AR_OU3P		Интерфейс ARINC. Выход передатчика 3 (прямой)
			nDMAR[9]		Запрос канала 9 контроллера ПДП
T7	89	PC[1]	FLAG[1]	IO (ppu)	Порт C общего назначения. Управление флагом 1
T8	92	PC[4]	BOOT[0]	IO (ppu)	Порт C общего назначения. Выбор режима загрузки
			HCLK		Хост интерфейс. Синхросигнал
T9	100	PC[9]	MSSD[0]	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Активация модуля 0
T10	107	PC[14]	SD_nRAS	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Выбор строки
T11	113	PC[19]	nMS[0]	O	Порт C общего назначения. Интерфейс статической памяти. Выбор типа 0
T12	118	PE[21]	ADDR[21]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 21
T13	123	PE[16]	ADDR[16]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 16

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
T14	128	PE[11]	ADDR[11]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 11
T15	132	PE[7]	ADDR[7]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 7
T16	136	PE[4]	ADDR[4]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 4
T17	139	PE[1]	ADDR[1]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 1
U1	69	PB[16]	LC_R[0]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 0
			AR_IN7P		Интерфейс ARINC. Вход приемника 7 (прямой)
			nDMAR[1]		Запрос канала 1 контроллера ПДП
			GTMR1 BRK		Таймер 1. Блокировка выходов
U2	68	PB[15]	LC_G[5]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит зеленого цвета 5
			AR_IN6N		Интерфейс ARINC. Вход приемника 6 (инверсный)
			GTMR1 nCH3o		Таймер 1. Выход 3 ШИМ (-)
U3	70	PB[17]	LC_R[1]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 1
			AR_IN7N		Интерфейс ARINC. Вход приемника 7 (инверсный)
			GTMR1 ETR		Таймер 1. Универсальный регистратор событий
U4	73	PB[20]	LC_R[4]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Бит красного цвета 4
			AR_OU1P		Интерфейс ARINC. Выход передатчика 1 (прямой)
			GTMR1 CH2i		Таймер 1. Вход 2 захвата
U5	77	PB[24]	LC_T[2]	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Дополнительный выход 2
			nIRQ[2]		Вход прерывания 2
			nDMAR[6]		Запрос канала 6 контроллера ПДП
U6	84	PB[29]	LC_HSYNC	IO (ppu)	Порт B общего назначения. Интерфейс ЖКИ. Сигнал горизонтальной синхронизации ЖКИ
			AR_OU3N		Интерфейс ARINC. Выход передатчика 3 (инверсный)
			nDMAR[10]		Запрос канала 10 контроллера ПДП
U7	90	PC[2]	FLAG[2]	IO (ppu)	Порт C общего назначения. Управление флагом 2
U8	91	PC[3]	FLAG[3]	IO (ppu)	Порт C общего назначения. Управление флагом 3
U9	99	PC[8]	SD_CKE	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Разрешение синхронизации
U10	104	PC[13]	SD_nCAS	IO (ppu)	Порт C общего назначения. Интерфейс SDRAM. Выбор колонки
U11	111	PC[18]	nMS[1]	O	Порт C общего назначения. Интерфейс статической памяти. Выбор типа 1
U12	117	PC[23]	ACK	IO (ppu)	Порт C общего назначения. Интерфейс статической памяти. Сигнал готовности
U13	122	PE[17]	ADDR[17]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 17

Номер вывода	Номер КП кристалла	Обозначение вывода	Дополнительное назначение вывода*	Тип вывода	Функциональное назначение
U14	127	PE[12]	ADDR[12]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 12
U15	131	PE[8]	ADDR[8]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 8
U16	135	PE[5]	ADDR[5]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 5
U17	138	PE[2]	ADDR[2]	O	Порт E общего назначения. Шина адреса внешнего интерфейса. Бит 2

* При использовании альтернативной функции вывод порта GPIO переходит под управление одного из интерфейсов процессора, работа которого в данный момент разрешена. Дополнительные назначения выводов указаны в таблице в порядке возрастания приоритета: сверху указано назначение с наименьшим приоритетом. В мультиплексировании участвуют только дополнительные назначения, используемые интерфейсом как выход.

Примечание – Обозначение типов выводов:
I – вход;
O – выход;
IO – вход/выход;
PWR – «Питание»;
GND – «Общий»;
pu – резистор доопределения до питания;
ppu – резистор доопределения до питания, включаемый программно. По сбросу включен;
pd – резистор доопределения до нуля;
ppd – резистор доопределения до нуля, включаемый программно. По сбросу включен

4 Указания по применению и эксплуатации

Металлизированные площадки основания корпуса A1÷A8, предназначенные для монтажа навесных элементов, электрически соединены с выводами питания ядра 81, 103, 124, 213, 233.

Металлизированные площадки основания корпуса B1÷B8, предназначенные для монтажа навесных элементов, электрически соединены с выводами «Общий» 75, 97, 198, 218, 238, 256.

Крышка корпуса электрически изолирована от выводов микросхемы.

Типовая схема включения микросхем приведена на рисунке 120.

Рекомендуется конденсаторы C9÷C16 емкостью 10 нФ распаивать непосредственно на металлизированные площадки основания корпуса A1÷A8, B1÷B8.

Выполнение класса задач типа чтения данных из одного порта и их трансляцию в другие порты обеспечивается при тактовой частоте процессора $f_C \leq 200$ МГц. Устойчивая работа микросхем при реализации более сложных алгоритмов обеспечивается при $f_C \leq 230$ МГц.

При ремонте аппаратуры и измерении электрических параметров микросхем замену микросхем необходимо проводить только при отключенных источниках питания.

Запрещается подведение каких-либо электрических сигналов (в том числе шин «Питание», «Общий») к выходам микросхем, не используемым согласно схеме электрической.

5 Архитектура процессора ЦОС

Процессор состоит их трех архитектурных частей:

- ядро процессора (Рисунок 4), где исполняются команды;
- внутренняя память (Рисунок 5), где хранятся данные;
- периферийные устройства (Рисунок 6), которые осуществляют операции обмена с внешними устройствами.

В процессоре можно выделить следующие элементы:

- два вычислительных модуля: X и Y, каждый из которых содержит умножитель, ALU, CLU, сдвиговое устройство и регистровый файл объемом в 32 слова;
- два блока целочисленных ALU: J и K, каждый из которых содержит 32-битное целочисленное ALU, а также регистровый файл объемом в 32 слова;
- устройство управления (Sequencer), управляющее ходом исполнения программы и содержащее буфер выравнивания команд (instruction alignment buffer – IAB) и буфер целевых адресов перехода (branch target buffer – BTB);
- три 128-битные шины, обеспечивающие возможность высокоскоростного обмена между внутренней памятью и другими компонентами ядра процессора (вычислительными блоками, блоками целочисленных ALU, устройством управления и SOC-интерфейсом);
- 128-битная шина, обеспечивающая возможность высокоскоростного обмена между внутренней памятью и периферийными устройствами внешнего ввода/вывода (DMA, внешним портом и LINK-портами);
- SOC-интерфейс, обеспечивающий связь между внутренними шинами ядра и шиной периферийных устройств;
- периферийные устройства: интерфейс внешнего порта, контроллер SDRAM, конвейерный интерфейс со статической организацией конвейера, двенадцать каналов DMA, два порта LVDS-линков (с двумя каналами DMA каждый) и др.;
- 12 Мбит внутренней памяти, организованной как шесть блоков по 2 Мбит, каждый из которых содержит 64 К 32-битных слов;
- средства поддержки отладки;
- тестовый порт JTAG.

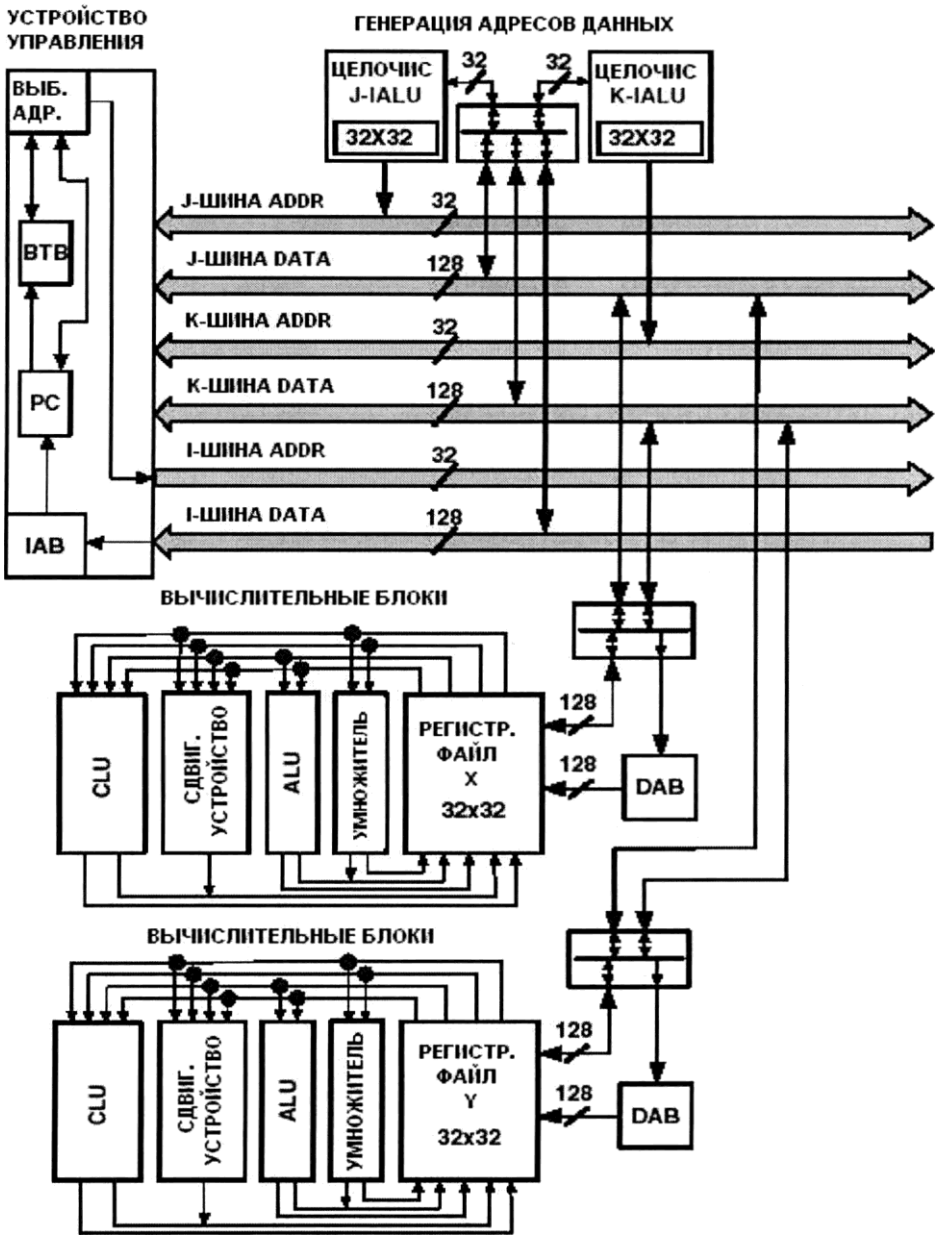


Рисунок 4 – Схема ядра процессора

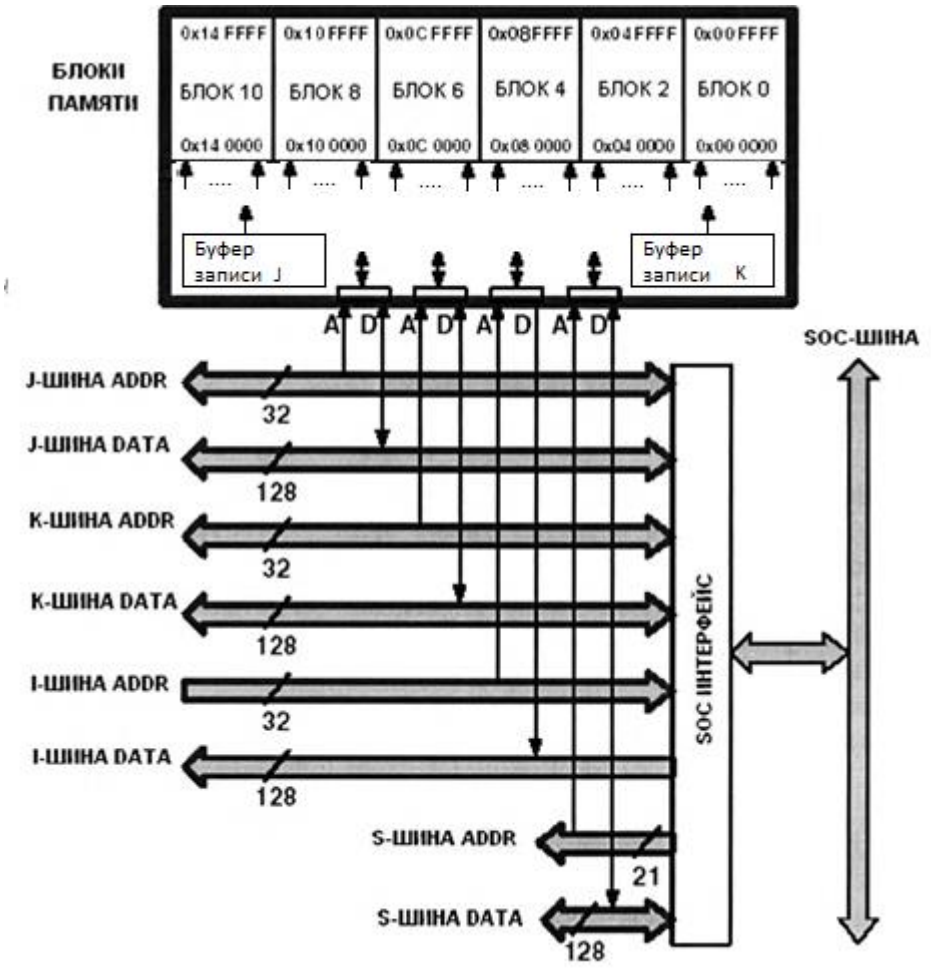


Рисунок 5 – Подключение внутренней памяти к ядру процессора

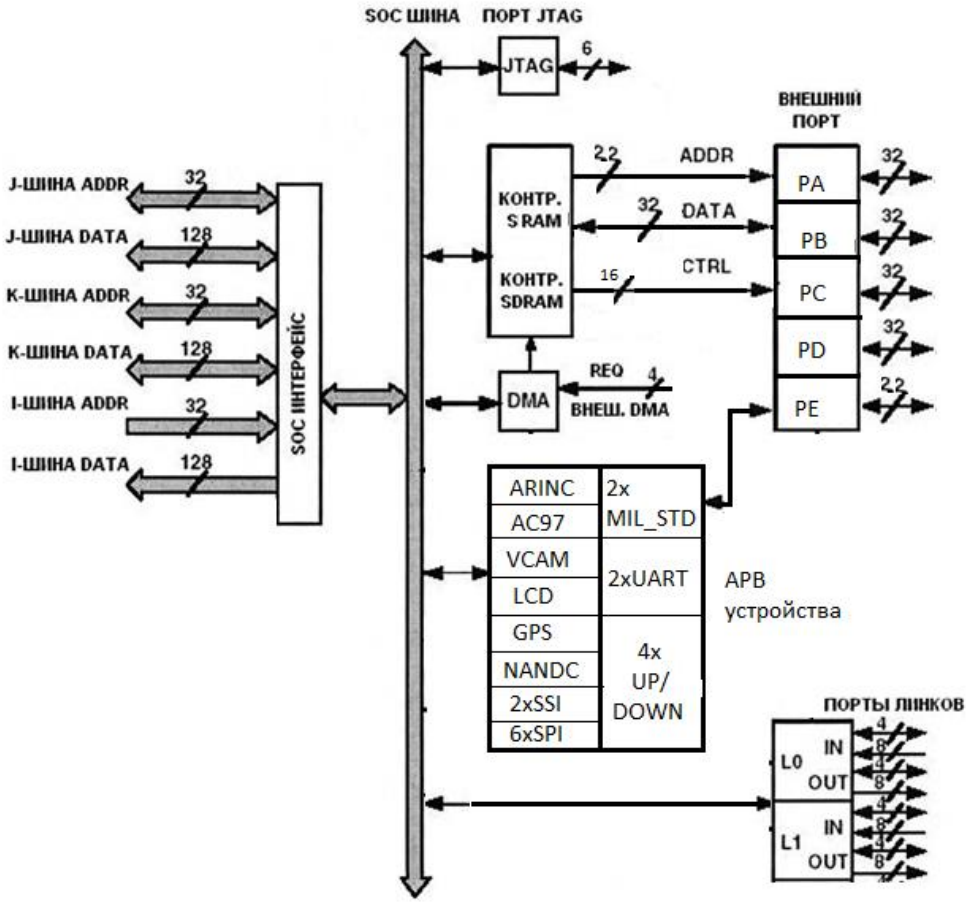


Рисунок 6 – Подключение периферийных устройств к ядру процессора

Внутренняя память объемом 12 Мбит разбита на шесть блоков памяти по 64 К слов. Каждая из четырех пар внутренних шин «адрес/данные» подсоединена ко всем шести блокам памяти через коммутационную матрицу. Шесть блоков памяти поддерживают до четырех обращений в каждом такте, причем каждый блок памяти может выполнить 128-битное обращение за такт.

Внешний порт поддерживает ширину шины данных 16 или 32 бита. Высокая пропускная способность ввода/вывода сочетается с высокой скоростью работы ядра. Для достижения высокой тактовой частоты процессор использует конвейерную внешнюю шину для синхронной статической памяти (SSRAM) и для синхронной динамической памяти (SDRAM).

Высокую пропускную способность передачи данных из точки в точку поддерживают два порта LVDS-линков. Каждый LINK-порт обеспечивает полнодуплексную связь.

Наличие разнообразных контроллеров периферийных устройств позволяет подключать:

- внешнюю NAND флэш-память;
- последовательную флэш-память с интерфейсом SPI;
- внешние аудио кодеки с интерфейсом SSI или AC97;
- внешние устройства, поддерживающие интерфейсы ARINC и MIL_STD;
- внешнюю LCD-панель произвольного разрешения;
- внешнюю видеокамеру;
- внешние ЦАП\АЦП.

5.1 Ядро процессора

В ядре процессора можно выделить пять самостоятельных функциональных модуля (Рисунок 4):

- Модули X и Y – вычислительные модули (устройства), в которых происходит основная обработка данных, и они в свою очередь включают в себя различные устройства обработки данных.
- Модули J и K – целочисленные АЛУ, которые также являются адресными генераторами (или адресными АЛУ), т.к. только они имеют возможность формирования адреса для доступа к памяти. Однако кроме адресных функций эти модули способны выполнять функции и по обработке данных. При этом данные (как и адреса) могут иметь только тип Integer, что соответствует 32-разрядному целому числу.
- Модуль S – устройство управления, которое формирует адреса команд, управляет потоком команд, а также управляет работой всего конвейера ядра.

Модули X и Y являются симметричными: все, что делает модуль X, может делать и модуль Y. Каждый модуль обрабатывает свой поток команд, при этом модули X и Y можно рассматривать, как объединенный модуль, управляемый одной командой.

Модули J и K также симметричны, и каждый из них выполняет свой поток команд.

Устройство управления (модуль S) также имеет способность выполнять несколько потоков команд определенных типов. Среди них основными можно назвать команды переходов и вызовов.

Модульная архитектура процессора с независимыми потоками команд позволяет организовать высокопараллельные вычисления. Так в процессе интенсивных вычислений одно или оба целочисленных ALU вычисляют или генерируют адреса для выборки до двух операндов размером в квадрослово из двух блоков памяти, в то время как устройство управления одновременно извлекает следующую четверку команд из третьего блока памяти. Параллельно вычислительные устройства могут обрабатывать ранее считанные операнды, а устройство управления подготавливать переход.

Пока ядро процессора занято вышеописанными действиями, каналы DMA могут в фоновом режиме обновлять содержимое внутренней памяти квадрословами данных из внешнего порта или из портов линков.

Вычислительное ядро процессора достигает исключительно высокой производительности при цифровой обработке сигналов благодаря использованию следующих особенностей:

- вычислительного конвейера;
- пары вычислительных устройств;
- исполнения до четырех команд за такт;
- выборки/записи до восьми слов памяти за такт.

Два идентичных вычислительных устройства (X и Y) выполняют арифметические операции как с плавающей, так и с фиксированной точкой. Эти устройства выполняют до 6 операций с плавающей точкой или до 24 операций с фиксированной точкой за такт. Модули J и K выполняют SISD-обработку, модули X и Y могут выполнять SISD- и SIMD-обработку. При этом модули X и Y могут

управляться одной общей командой. Модули X и У имеют сложную структуру и включают в себя разнообразные вычислительные блоки.

5.1.1 Вычислительные модули

Ядро процессора содержит два вычислительных устройства, называемых *вычислительными модулями.* Каждый вычислительный модуль содержит регистровый файл и четыре независимых вычислительных блока: ALU, CLU, умножитель и сдвиговое устройство.

Вычислительные блоки способны обрабатывать данные в нескольких форматах представления с фиксированной и плавающей точкой (Рисунок 7):

- Форматы данных с фиксированной точкой:
- 64-битное длинное слово (Long Long);
- 32-битное обычное слово (Integer);
- 32-битное комплексное слово (16-битная действительная и 16-битная мнимая часть);
- 16-битное короткое слово (Short);
- 8-битное однобайтное слово (Char).

  Для коротких слов арифметики с фиксированной точкой учетверенные операции над данными, выровненными на границу квадрослова, обеспечивают быструю обработку вектора данных. Байтовые операции поддерживаются также для данных, выровненных на границу октослова (т.е. 256-битного слова).
- Форматы данных с плавающей точкой:
- 32-битное обычное слово (float);
- 64-битное двойное слово (double);
- 40-битное расширенное слово.

  Операции с плавающей точкой выполняются с одинарной, двойной и расширенной точностью. Формат обычного слова с плавающей точкой соответствует стандартному IEEE-формату, а 40-битное число формата с увеличенной точностью размещается в двойном слове (64 бита), занимая дополнительно к 32 битам восемь наименее значащих битов (Least Significant Bits – LSBs) мантиссы для достижения большей точности.

Каждый вычислительный модуль имеет многопортовый регистровый файл, содержащий регистры общего назначения для обмена данными между вычислительными блоками и шинами данных, а также для хранения промежуточных результатов. Ко всем этим регистрам можно обращаться как к обычному, двойному или квадрорегистру. Все операции вычислительных модулей исполняются на двух стадиях конвейера. Результат операции доступен только на второй стадии, поэтому если следующая команда вычислительного модуля использует результат текущей команды как операнд-источник, всегда возникает такт простоя процессора (пузырь).

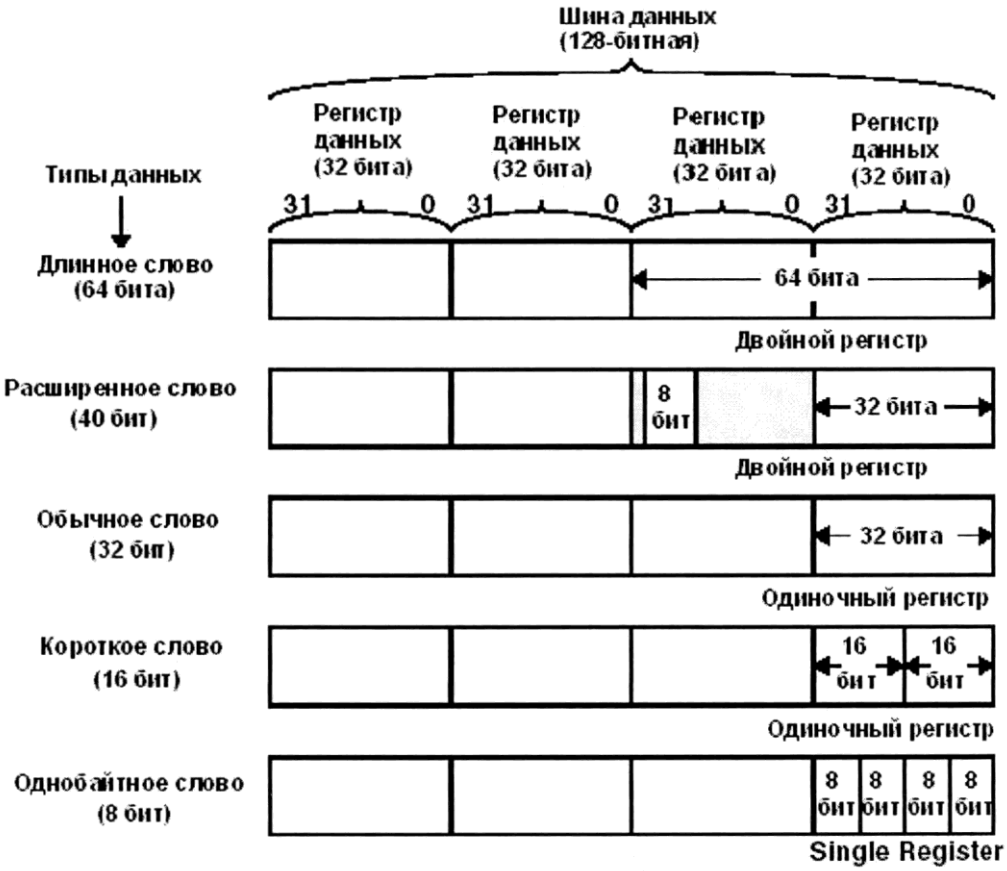


Рисунок 7 – Определения форматов слов

5.1.1.1 Арифметико-логическое устройство (ALU)

ALU выполняет арифметические операции над данными с фиксированной и с плавающей точкой, а также логические операции над данными с фиксированной точкой. Источником и приемником большинства операций ALU является регистровый файл вычислительного блока. Операнды ALU могут иметь ширину 32 или 64 бита, что соответствует одиночным операциям над типами Int, Float, Double, Long Long. Для типов меньшей разрядности (Short, Char) поддерживается векторная обработка. Очевидно, что количество элементов в векторах для типа Int равно двум, для типа Short – двум или четырем, а для типа Char – четырем или восьми. Учитывая, что вычислительные модули X и Y могут работать как один сдвоенный модуль, длина обрабатываемых векторов может быть увеличена в два раза, т.е. за такт процессор способен обрабатывать четыре типа Int, восемь типов Short или 16 типов Char. Все это соответствует максимальной ширине шины памяти в 128 бит.

5.1.1.2 Коммуникационно-логическое устройство (CLU)

Каждый вычислительный модуль содержит вычислительное устройство специального назначения, называемое *коммуникационно-логическим устройством (CLU).* Команды CLU предназначены для поддержки различных алгоритмов, используемых в коммуникационных приложениях, а именно алгоритмов:

- Декодирования по Витерби;
- Декодирования турбо кода;
- Сжатия [частотной полосы сигнала] (despreading) для систем множественного доступа с кодовым разделением (CDMA);
- Кросс-корреляций, используемых для отыскания пути.

Особенностью данного блока является собственный 32-словный набор регистров общего назначения. Таким образом, команды CLU могут использовать операнды из РОН вычислительного модуля и из РОН собственного блока.

5.1.1.3 Умножитель с накоплением (Умножитель)

Блок умножителя выполняет умножение чисел с фиксированной или плавающей точкой, а также операции умножения с накоплением для чисел с фиксированной точкой. Умножитель поддерживает несколько типов данных для форматов с фиксированной и плавающей точкой. Для плавающей точки – это стандартные форматы одинарной и двойной точности, а также формат с расширенной точностью. Источником и приемником для большинства операций является регистровый файл вычислительного блока. Умножитель также поддерживает операции умножения над комплексными числами, представленными в виде пары 16-разрядных коротких слов упакованными в одно 32-разрядное слово. Младшие значащие биты этого слова представляют собой действительную часть комплексного числа, а старшие значащие биты – мнимую часть. Команды умножения с накоплением используют в качестве приемника специальные регистры-аккумуляторы.

5.1.1.4 Побитное сдвиговое устройство (сдвиговое устройство)

Сдвиговое устройство выполняет логические и арифметические сдвиги, манипуляции с битами, заполнение полей и извлечение полей. Сдвиговое устройство выполняет операции с одним 64-битным, одним или двумя 32-битным, двумя или четырьмя 16-битными, а также с четырьмя или восемью 8- битными операндами с фиксированной точкой. Операции сдвигового устройства включают в себя:

- сдвиги и циклические сдвиги влево и вправо;
- операции манипуляции с битами, включая установку бита, очистку бита, переключение бита и проверку,
- операции манипуляции с битовыми полями, включая извлечение полей, заполнение полей с использованием регистра BFOTMP (внутреннего регистра сдвигового устройства);
- операции с однобитным FIFO для поддержки потоков битов с полями переменной длины;
- поддержка операций преобразования фиксированной и плавающей точки (такими, как извлечение порядка, определение числа ведущих единиц и нулей).

5.1.2 Целочисленное арифметико-логическое устройство (IALU)

Целочисленные ALU могут исполнять стандартные автономные операции ALU с регистровыми файлами IALU, операции загрузки/сохранения регистров и пересылки из регистра в регистр, а также формировать адреса при обмене данными между памятью и регистрами. Процессор имеет пару IALU (J- IALU и K-IALU), что позволяет формировать адреса одновременно для двух параллельно исполняемых транзакций по 128 бит. Наличие целочисленных ALU позволяет вычислительным операциям исполняться с максимальной эффективностью, поскольку вычислительные устройства могут быть заняты исключительно обработкой данных.

Каждое IALU имеет многопортовый 32-словный регистровый файл. Любая вычислительная операция IALU исполняется за один такт. При генерации адресов целочисленные ALU поддерживают предмодификацию регистров без их обновления

и постмодификацию с обновлением. Кольцевые буферы реализованы аппаратно. Целочисленные ALU поддерживают следующие типы команд:

- обычные команды IALU;
- команды пересылки данных [из регистра в регистр];
- команды загрузки константы в регистр;
- команды загрузки/сохранения данных с модификацией значением регистра;
- команды загрузки/сохранения данных с модификацией непосредственным значением.

При косвенной адресации (команды с модификацией) значение одного из регистров регистрового файла может быть модифицировано значением другого регистра файла или непосредственным 8- или 32-битным значением, как до (предмодификация), так и после (постмодификация) выполнения обращения к памяти. При адресации кольцевых буферов каждому из четырех первых регистров IALU может быть сопоставлено значение длины для выполнения автоматической адресации по ее модулю внутри такого буфера. Границами кольцевых буферов могут служить любые адреса памяти. Кольцевые буферы делают возможной эффективную реализацию линий задержки и других структур данных, используемых обычно в цифровых фильтрах и преобразованиях Фурье. Переход адресного указателя через границу кольцевого буфера обрабатывается процессором автоматически, что позволяет сократить накладные расходы и упростить программную реализацию алгоритмов.

Целочисленные ALU поддерживают также бит-реверсивную адресацию, ориентированную на алгоритм БПФ. Бит-реверсивная адресация реализуется с использованием сложения с обратным переносом, подобного обычному сложению, но у которого бит переноса берется из старших битов и направляется в младшие.

IALU обеспечивают гибкость в пересылках данных обычными, двойными или квадрословами. Каждая команда может исполняться с производительностью одна на такт. Обычно задержки, обусловленные зависимостью между командами IALU, отсутствуют, но если таковые имеются, их длительность может достигать трех тактов. Задержки могут появляться, если текущая команда производит загрузку данных из памяти в регистр IALU или пересылку регистра в регистр IALU, а следующая команда использует загружаемый регистр в качестве операнда-источника. В таком случае возникает три такта задержки до готовности операнда. Необходимо отметить, что IALU обладает практически такими же вычислительными возможностями по обработке данных типа Integer, как и вычислительный модуль. Наиболее проблемным местом является только отсутствие в IALU операции умножения.

5.1.3 Устройство управления потоком команд

Устройство управления (УУ) предоставляет адреса для выборки команд из памяти. УУ совместно с IALU позволяет вычислительным операциям выполняться с максимальной эффективностью. Эффективность ветвления обеспечивается устройством управления за счет использования буфера целевых адресов перехода (BTB), позволяющего сократить задержки на исполнение условных и безусловных переходов. УУ выполняет две задачи:

- декодирование извлеченных из памяти команд (выделение команд из полей командной строки) и отсылка их на соответствующие исполнительные устройства (вычислительные блоки, целочисленные ALU или УУ);
- управление ходом исполнения программы.

Команды устройства управления подразделяются на два типа:

- *команды управления ходом исполнения программы* – используются для смены текущего адреса исполнения (перехода) и для условного исполнения индивидуальных команд;
- *команды непосредственного расширения* – используются для расширения числовых полей, указываемых в непосредственных операндах для УУ и IALU.

Команды управления ходом исполнения программы подразделяются, в свою очередь, на две категории:

- Явные переходы и вызовы, основанные на непосредственно указываемом в коде команды адресном операнде. Например, по команде 'if <cond> jump 100;' всегда осуществляется переход по адресу 100, если только вычисленное значение выражения <cond> есть «истина».
- Косвенные переходы, базирующиеся на адресе, предоставляемом регистром. (Команды, используемые для указания на условное исполнение командной строки, представляют собой подкатегорию косвенных переходов). Например, 'if <cond> cjmp;' является переходом по адресу, содержащемуся в регистре CJMP.

Примечание – Команды управления ходом исполнения должны располагаться в первом поле командной строки. Непосредственными расширениями могут обладать команды IALU и УУ (в последнем случае только команды управления ходом исполнения).

Команды управления ходом исполнения программы не задаются программистом явно, а порождаются автоматически в случае, если размер непосредственных данных, использованных в командах, достаточно велик. Программист должен поместить команду, которая требует непосредственного расширения, в первое поле командной строки и оставить свободное поле в этой строке (задействовав только три поля) с тем, чтобы ассемблер мог поместить непосредственное расширение во второе поле командной строки. В одной командной строке может присутствовать только одно непосредственное расширение.

Процессор достигает своей высокой скорости исполнения за счет использования 9-ти стадий конвейера. Каждая стадия соответствует одному такту частоты процессора. Упрощенная структура конвейера приведена ниже (Рисунок 8).

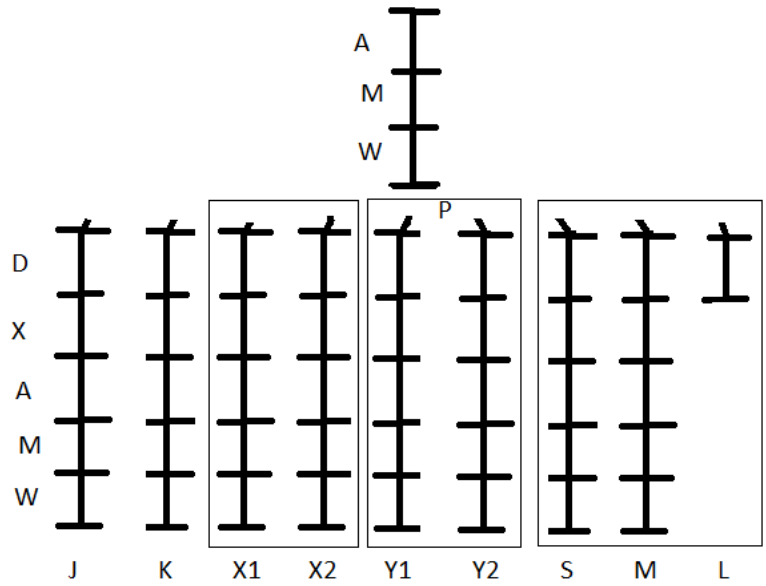


Рисунок 8 – Структура конвейера процессора

На рисунке горизонтальным линиям конвейера соответствуют регистры для хранения информации между стадиями конвейера, а вертикальные линии – задержке обрабатывающей логики. Самая верхняя горизонтальная линия соответствует регистру IP – указателю на текущую команду устройства управления. IP содержит адрес, который отправляется в подсистему памяти для выборки команды.

Стадия адреса (стадия A) соответствует логике декодирования адреса, а также логике арбитража доступа к памяти.

Следующая стадия M соответствует чтению данных из накопителя в буфер памяти.

На стадии W происходит передача команды из буфера памяти в буфер выравнивания команд IAB. Шина команд имеет ширину 128 бит, что соответствует выборке 4-х команд за такт.

На стадии P одна линия конвейера *чтения* команд разветвляется на 9 линий (ветвей) конвейера *исполнения* команд.

Из рисунка видно, что модули J и K имеют по одной линии конвейера, модули X и Y – по две линии конвейера каждый, а модуль управления включает три ветви S, M, L. Ветвь L короткая, т.к. она обрабатывает команды-расширители константного поля, и вся обработка этой команды сводится к определению, какой линии конвейера соответствует расширение и передачи данных на эту линию. Для модулей обработки J, K, X, Y последняя горизонтальная линия конвейера соответствует РОН, в который записывается результат операции. Название стадий конвейера обработки данных D, X, A, M, W выбрано исходя из логики выполнения команды загрузки из памяти модулей J и K.

- D – декодирование команды;
- X – исполнение команды (вычисление адреса доступа);
- A – передача адреса в подсистему памяти;
- M – чтение данных из накопителя в буфер;
- W – передача данных из буфера памяти в регистр ядра.

В зависимости от типа команды операции, выполняемые на данных стадиях конвейера, будут различны. Так для модулей X и Y наиболее важными являются

стадии A, M, W. На стадии A происходит декодирование команды, на стадии M – первый такт исполнения и на стадии W – второй (заключительный такт исполнения).

Одной из задач УУ является отслеживание зависимостей по операндам во всех линиях конвейера и приостановка конвейера в случае, если операнды не готовы. Все линии конвейера продвигаются синхронно. При этом максимум 6 линий из 9-ти могут исполнять одновременно команды.

Для УУ основной линией конвейера является линия S, на которой происходит обработка команд переходов. Для такого глубоко конвейеризированного процессора скорость выполнения переходов является одним из ключевых факторов эффективности исполнения программы. Переходы могут быть безусловными и условными. Условные переходы могут быть двух типов: переходы, анализируемые на стадии X конвейера и на стадии W. Последние (на W) соответствуют анализу флагов операций в модулях X и У. Можно видеть, что в случае выполнения перехода на стадии X процессор потеряет пять тактов, прежде чем следующая команда достигнет стадии X. Для ветвлений на стадии W таких тактов будет уже восемь. Для уменьшения потерь, связанных с переходами, УУ содержит буфер целевых адресов перехода (BTB) и имеет механизм статического прогнозирования переходов (выполняется пользователем). Если переход помечен как прогнозируемый, он выполняется на стадии D линии S конвейера. Информация о переходе заносится в строку BTB. При первом выполнении данного перехода теряются четыре такта, однако при последующих его выполнениях потери будут равны 0.

Еще одной задачей УУ является обработка запросов прерываний.

Процессор имеет внешние линии запроса прерываний общего назначения, а также может обрабатывать прерывания от внутренних периферийных устройств. Есть возможность программным способом обеспечить обработку вложенных прерываний. Прерывания имеют низкую латентность и не могут сбросить с конвейера текущие команды, уже начавшие свое исполнение. Прерывание векторизуется непосредственно по заданному пользователем адресу в регистровом файле таблицы векторов прерываний.

Запрос прерывания поступает в УУ и если прерывания разрешены, стадия P конвейера прекращает поставлять команды на линии обработки команд, а УУ помещает в регистр IP адрес вектора прерывания (сбрасывая содержимое конвейера чтения команд) и запускает чтение нового потока.

5.1.4 Управление ядром процессора

Существует несколько режимов работы процессора, а также средств управления его работой. В данном разделе описываются режимы работы, которые не управляются посредством выполнения команд. Эти средства управления включают управление:

- доменами синхронизации;
- режимом работы;
- режимом загрузки.

5.1.4.1 Режимы работы

Процессор может работать в одном из трех режимов:

- пользовательский;
- супервизор;
- эмулятор.

В режиме пользователя и супервизора все команды выполняются стандартным образом (т.е. путем чтения из памяти). Отличие режима пользователя от режима супервизора заключается только в ограничении доступа к отдельным компонентам системы для пользователя. Режим также влияет на прием и обработку исключений. Приоритеты режимов от наименьшего приоритета к наивысшему: пользовательский, супервизор, эмулятор.

Пользовательский режим

Работа в пользовательском режиме используется для алгоритмов и управляющего кода, который не требует манипуляций с системными ресурсами. Многие системные ресурсы недоступны в пользовательском режиме. Если процессор пытается получить доступы к таким ресурсам, возникает исключение. Ядро ОС работает в режиме супервизора, но пользовательский код ограничен пользовательским режимом.

Следующие регистры доступны ядру программы в пользовательском режиме:

- группы регистров от 0x00 до 0x09 – регистры вычислительных модулей;
- группы регистров от 0x0C до 0x0F – регистры целочисленных АЛУ;
- регистры УУ – CJMP, регистры счетчика циклов LC0, LC1 и регистр статичного флага (SFREG).

Все другие регистры не могут быть записаны ядром программы в пользовательском режиме. Попытка записи в защищенный регистр вызовет исключение.

Режим супервизора

Режим супервизора позволяет программе получить доступ ко всем ресурсам процессора. Процессор работает в режиме супервизора при выполнении одного из условий:

- бит NMOD в SQCTL установлен;
- процедура прерывания запущена.

При аппаратном или программном сбросе процессор переходит в состояние ожидания прерывания. При получении прерывания он переходит в состояние исполнения программы в режиме супервизора.

Если бит NMOD в SQCTL сброшен, процессор входит в пользовательский режим после выхода из прерывания, если только прерывание не является вложенным.

Режим эмулятора

Режим эмулятора используется при управлении процессором отладочным инструментом через порт JTAG. Процессор входит в режим эмулятора при генерации специального исключения. Исключение эмулятора генерируется при одном из событий:

- команда EMUTRAP;
- срабатывание точки наблюдения для входа в эмулятор;
- передний фронт TMS при установленном бите TEME в EMUCTL.

Исключения эмулятора имеют высочайший приоритет среди исключений и прерываний.

Пока процессор находится в режиме эмулятора, единственным источником команд для него является регистр EMUIR. Регистр EMUIR загружается через порт тестового доступа JTAG. При вхождении в режим эмулятора, внешний контроллер JTAG должен быть включен.

После работы в режиме эмулятора, единственным способом выхода из данного режима является выполнение команды возврата из прерывания (RTI).

В режиме эмулятора, доступ к регистрам отладки (группа регистров 0x1B) может быть получен путем выполнения команд пересылки «регистр-регистр» или командой непосредственной загрузки данных. Эти регистры не могут быть загружены из памяти.

BTB, счетчик циклов, мониторы производительности и буфер трассировки неактивны в режим эмулятора. Кроме команды RTI (np), все управляющие команды (переход, вызов, RDS, условные переходы, и т.д.) и команда IDLE не должны быть использованы в режиме эмулятора. Команда RTI может быть использована только безусловно.

5.1.4.2 Режимы старта

После сброса процессор начинает работу с выполнения специальной загрузочной программы, записанной во внутреннее ПЗУ. Данная программа осуществляет анализ внешних конфигурационных выводов и выполняет заданную последовательность действий. Подробное описание программы начального старта будет приведено в разделе «Начальный старт процессора».

6 Память, регистры и шины

Глава содержит описание карты памяти и регистров. Для информации по использованию регистров для конфигурирования процессорной периферии необходимо смотреть соответствующие разделы данной спецификации, описывающие периферийные модули. Для получения информации об использовании регистров для вычислений и памяти для загрузки и хранения необходимо обращаться к разделу «Инструкция по программированию».

Процессор имеет шесть внутренних блоков памяти, как показано на карте памяти (Рисунок 9). Каждый блок памяти состоит из 2 МБит пространства памяти и сконфигурирован как 64 Кслова шириной по 32 бита. Существует четыре отдельные 128-битные внутренние шины, каждая из которых может получать доступ к блокам памяти. Процессор ввода-вывода (DMA) имеет собственную внутреннюю шину и не «соревнуется» с ядром за использование внутренней шины. Таким образом, за один цикл может осуществляться до четырех 128-битных передач внутри ядра (две передачи данных, одна команда и одна передача интерфейса ввода-вывода).

Процессор имеет 32-битные адресные шины, обеспечивающие адресное пространство до четырех гигаслов. Процессор поддерживает словную адресацию памяти, а также возможен режим байтовой адресации (см. ниже). Данная глава определяет карту памяти для процессора и показывает, где в пространстве памяти определяется каждый элемент. Зоны пространства памяти составлены из следующих областей:

- пространство внешней памяти – область для стандартной адресации внешней памяти, включая SDRAM, банк 0 (MS0), банк 1 (MS1)
- пространство внутренней памяти – область для стандартной внутренней адресации.

Для быстродействия процессора большое значение имеет конфигурация банков внутренней памяти. Шесть банков внутренней памяти обеспечивают возможность одновременного параллельного доступа к ним в течение одного такта ядра. Такой уровень быстродействия можно достичь только, если процессор будет, например, производить чтение команд из банка 0, чтение данных по шине J из банка 1, чтение данных по шине K из банка 2, чтение-запись по шине S внешним мастером банка 3, отложенная запись по шине J в банк 4 и отложенная запись по шине K в банк 5.

Общая карта памяти показана на рисунке 9.

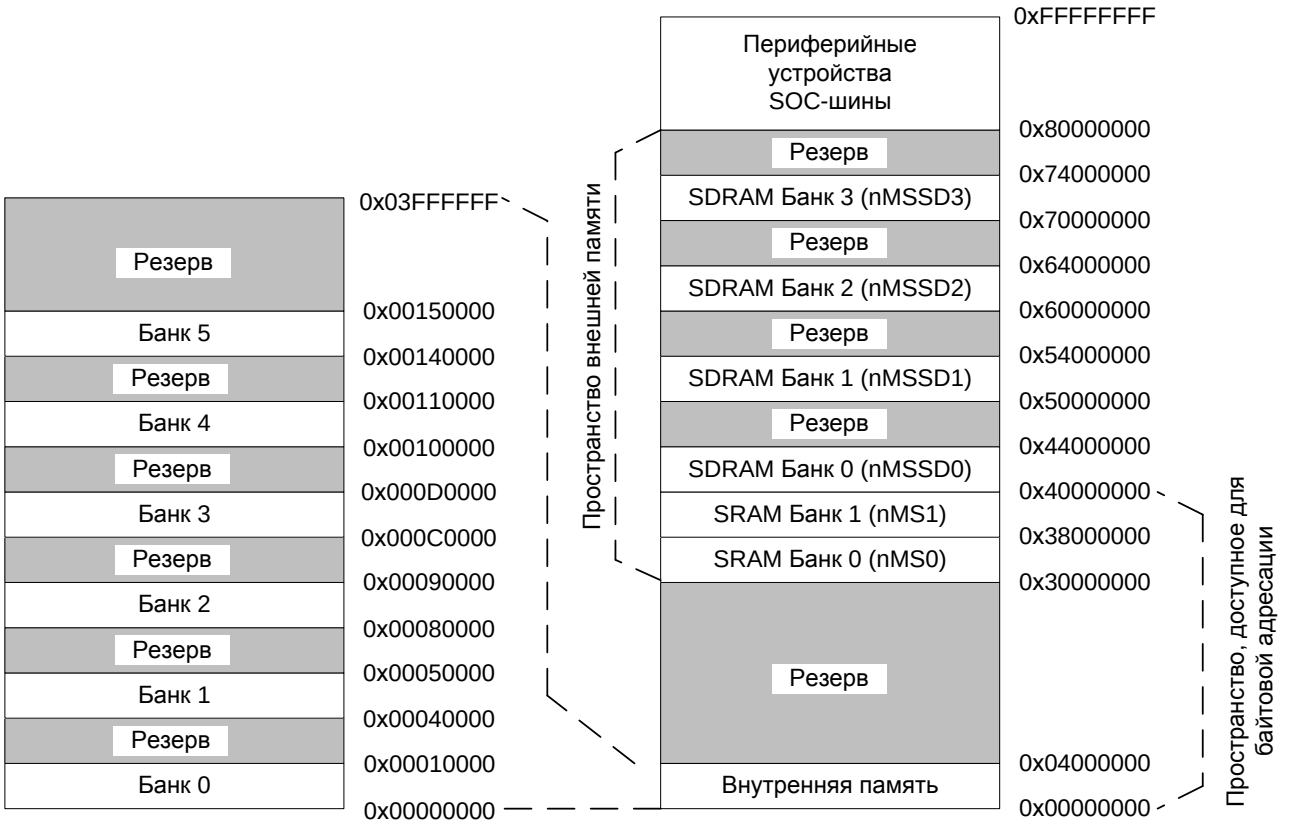


Рисунок 9 – Базовая карта памяти

Процессор поддерживает возможность байтовой адресации не только внутренней, но и внешней памяти. В связи с тем, что перед доступом к памяти адрес байта предварительно сдвигается вправо на два разряда, максимально адресуемое адресное пространство 32-разрядных слов ограничено диапазоном адресов от 0 до 0x3FFF_FFFF. Невозможен доступ к данным с помощью указателя на байт для области динамической памяти, так как для адреса слова (начиная с 0x4000_0000 и выше) для хранения адреса байта недостаточно 32-х разрядов.

В связи с этим регистр управления внешним интерфейсом SYSCON дополнен новой функцией бита 27. Если данный бит установить в 1, то доступ к динамической памяти становится возможным с помощью двух адресных областей

Карта памяти процессора с включенной дополнительной областью для доступа к динамической памяти приведена на Рисунке 10.

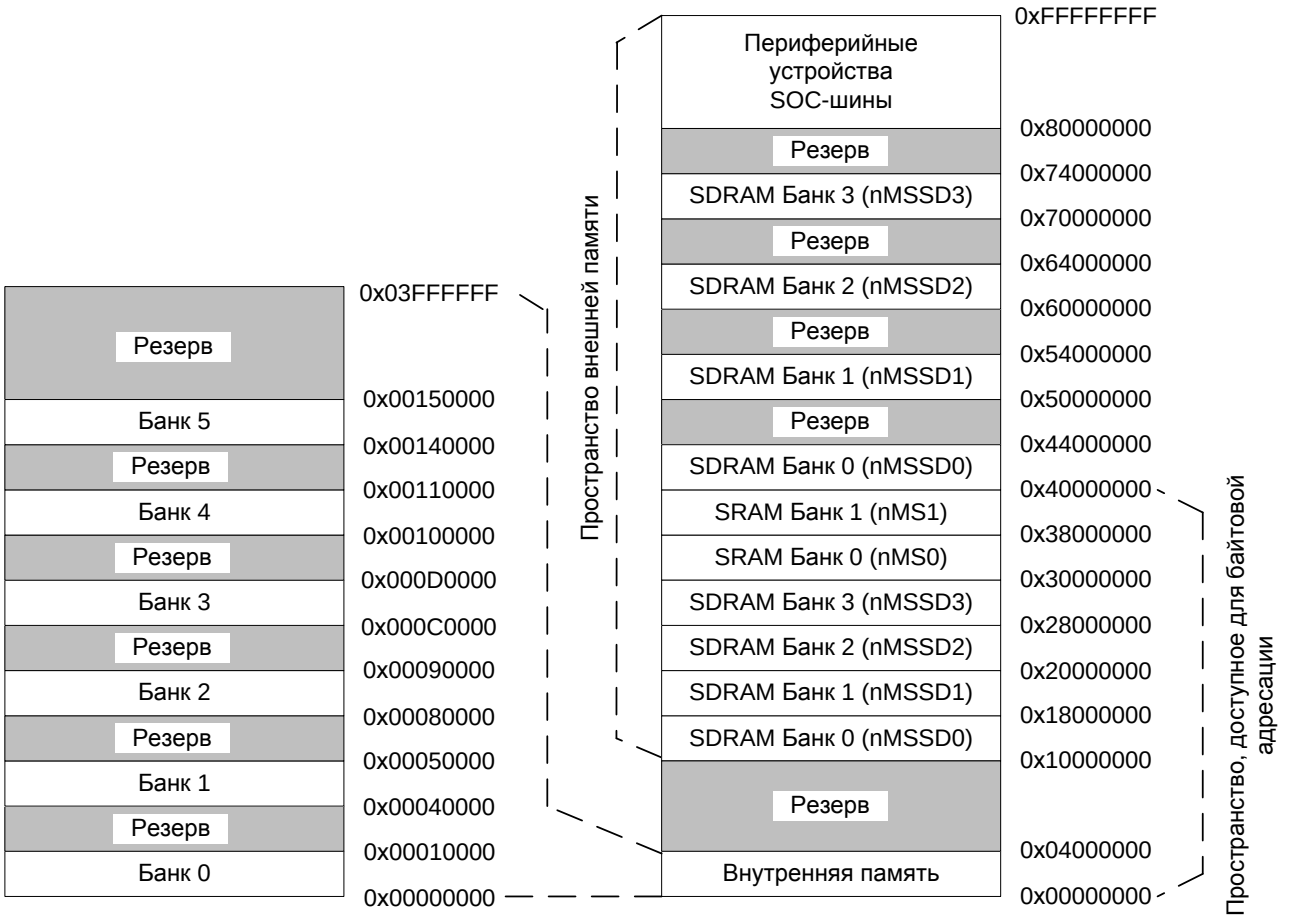


Рисунок 10 – Карта памяти с включенной дополнительной областью

Если для работы прикладной программы необходимо использование внешней динамической памяти, и выполняется обработка данных, требующая их байтовой адресации, в качестве адресного пространства динамической памяти должен быть выбран диапазон адресов (адрес слова) от 0x1000_0000 до 0x2FFF_FFFF. Это соответствует диапазону байтовой адресации от 0x4000_0000 до 0xBFFF_FFFF. Использование дополнительного адресного пространства упрощает преобразование указателей. Установка бита SYSCON[27] не исключает использование базового адресного пространства динамической памяти, т.е. доступ к одному и тому же слову динамической памяти можно сделать как по адресу 0x4000_0000, так и по адресу 0x1000_0000.

Область периферийных устройств имеет только словную адресацию и не поддерживает доступ к отдельным байтам или коротким.

6.1 Пространство периферийных устройств SOC-шины

Пространство периферийных устройств SOC-шины используется процессором и контроллером DMA для адресации внутренних периферийных устройств, подключенных к SOC-шине. Если 31-й бит адреса равен 1, возможен доступ к данному адресному пространству. При этом доступ к регистрам некоторых периферийных устройств может поддерживаться непосредственно по имени (номеру) из команды процессора. Однако это возможно только к регистрам устройств, совместимых с процессором 1967ВН028.

Карта памяти периферийных устройств приведена Таблице 3.

Таблица 3 – Карта памяти периферийных устройств с режимами доступа

Периферийное устройство	Базовый адрес	CPU	DMA	HOST
Контроллер DMA	0x8000_0000 0x8000_0020 0x8000_0040 0x8000_0060	RW	W	W
Хост-интерфейс	0x8000_005c	R	-	-
Интерфейс внешней шины EBIU	0x8000_0080	RW	W	W
Порты связи	0x8000_00a0 0x8000_00e0	RW	W	W
Интерфейс UART0	0x8000_0100	RW	R*W	W
Интерфейс UART1	0x8000_0120	RW	R*W	W
Интерфейс SPI0	0x8000_0140	RW	R*W	W
Интерфейс ЖКИ	0x8000_0180	RW	R*W	W
Интерфейс видеокамеры	0x8000_01a0	RW	R*W	W
Блок управления синхронизацией и энергопотреблением	0x8000_01c0	RW	R*W	W
Часы реального времени RTC	0x8000_01e0	RW	R*W	W
Интерфейс I2S0	0x8000_0200	RW	R*W	W
Интерфейс I2S1	0x8000_0220	RW	R*W	W
Интерфейс NAND Flash	0x8000_0240	RW	R*W	W
Модуль цифровой обработки 0	0x8000_0280	RW	R*W	W
Модуль цифровой обработки 1	0x8000_0260	RW	R*W	W
Модуль цифровой обработки 2	0x8000_02a0	RW	R*W	W
Модуль цифровой обработки 3	0x8000_02c0	RW	R*W	W
Интерфейс I2C	0x8000_02e0	RW	R*W	W
Контроллер прерываний и таймеры общего назначения	0x8000_0300 0x8000_0320 0x8000_0340	RW	W	W
Интерфейс SPI1	0x8000_0360	RW	R*W	W
Интерфейс SPI2	0x8000_0380	RW	R*W	W
Порт JTAG	0x8000_03a0	RW	W	W
Регистры AutoDMA	0x8000_03e0	W	W	-
Таймер 0 с функцией Захвата/ШИМ	0x8000_0400	RW	R*W	W
Таймер 1 с функцией Захвата/ШИМ	0x8000_0440	RW	R*W	W
Интерфейс ARINC (приемники)	0x8000_2000	RW	R*W	W
Интерфейс ARINC (передатчики)	0x8000_3000	RW	R*W	W
Цифровой коррелятор	0x8000_5000	RW	R*W	W
Интерфейс МКПД0	0x8000_6000	RW	R*W	W
Интерфейс МКПД1	0x8000_7000	RW	R*W	W

Обозначения в таблице:
RW – чтение и запись;
R – чтение;
W – запись;
R*W – чтение и запись, но операция чтения регистров периферийных устройств доступна только для DMA каналов 0, 1, 2, 3 и 8, 9, 10, 11

6.2 Пространство банка внешней памяти

Пространство банка внешней памяти относится к внешней памяти и устройствам ввода-вывода (SDRAM, SRAM, периферия ввода-вывода и другие стандартные устройства памяти). Внешний интерфейс поддерживает разбиение общего банка внешней памяти на отдельные банки и ставит им в соответствие

специальные сигналы выборки и другие управляющие сигналы. Имеется набор банков для SDRAM, доступ к которым производится через контакты выбора внешней памяти MSSD0, MSSD1, MSSD2 и MSSD3. Доступ к этим банкам осуществляется по протоколу SDRAM. Другой набор банков определен контактами выбора внешней памяти MS0 и MS1, где протокол доступа конфигурируется пользователем как конвейерный или протокол медленного устройства, и параметры этих банков определяются регистром SYSCON. Адрес внешней памяти разделяется на поля, как показано ниже (Таблица 4).

Таблица 4 – Пространство банка внешней памяти

Биты	Имя	Определение
ADDR25–0	Address	Биты, которые осуществляют выбор данных внутри адресного пространства определенного банка
ADDR30–26	MS/SD	Выбор типа банка внешней памяти: 01100 – банк 0 (MS0) 01110 – банк 1 (MS1) 10000 – SDRAM банк 0 (MSSD0) 10001 – зарезервировано 10100 – SDRAM банк 1 (MSSD1) 10101 – зарезервировано 11000 – SDRAM банк 2 (MSSD2) 11001 – зарезервировано 11100 – SDRAM банк 3 (MSSD3) 11101 – зарезервировано
ADDR31		Значение бита всегда будет равно нулю

6.3 Внутреннее адресное пространство

Внутреннее адресное пространство (см. Рисунок 9) соответствует собственному адресному пространству процессора. Оно используется для передач внутри процессора, т.е. для доступа к блокам внутренней памяти.

6.4 Универсальные регистры

К универсальным регистрам относятся все регистры вычислительного ядра. Универсальные регистры не отображаются в карте памяти, т.е. доступ к ним по адресу памяти невозможен. Доступ к регистрам возможен только при выполнении команд. Все регистры разбиты по группам. В одной группе находится 32 регистра. При явном доступе к регистру в коде команды указывается номер группы и номер регистра в группе. Регистры распределены по группам в соответствии с их отношением к различным модулям процессора. Например, регистры общего назначения (РОН) данных (XR31–0, YR31–0, или XYR31–0) вычислительных блоков находятся в одной группе регистров, тогда как РОН данных для целочисленного АЛУ (J30–0 или K30–0) – в других. Термин «универсальный регистр» указывает на следующие особенности.

Во-первых, содержимое регистра может быть загружено из памяти, сохранится в памяти или передаваться в\из других универсальных регистров процессора.

Во-вторых, регистр может быть доступен в одинарный, двойной и счетверенный (квадрорегистр). Ряд регистров может быть доступен только как одинарные.

Загрузка, сохранение и пересылка содержимого универсальных регистров доступна внутри процессора с использованием имени регистра в командах процессора, а не по адресу в памяти. Имена регистров содержатся в таблице групп регистров. Команда загрузки регистра читает данные из памяти процессора и помещает их в регистр. Команда сохранения регистра берет данные из регистра и помещает их в память процессора. Команда пересылки копирует данные из одного регистра в другой. Имеются также команды загрузки константы в регистр, когда значение операнда берется из кода команды и помещается в регистр. Следующий пример показывает доступы загрузки, сохранения и передачи одного регистра.

```
XR0 = 0x76543210 ;; /* загружает в XR0  конкретные 32-бит  данные */
XR4 = [J31 + 0x43] ;; /* загружает в XR4 данные из ячейки памяти с адресом 0x43 */
[J0 + J4] = YR0 ;; /* сохраняет YR0 в память */
XR7 = SQSTAT ;; /* передает содержимое SQSTAT в XR7 */
```

При возможности выполнения процессором до четырех команд, одновременно могут быть выполнены:

- две команды загрузки регистров из памяти
- две команды сохранения в память
- две команды пересылки.
- две загрузки регистров константами.

Все эти типы команд выполняются в целочисленных J и K модулях.

Для одновременного доступа к двум или четырем регистрам, образующим длинные слова и квадрослова, предусмотрена специальная техника образования нового имени регистра. При этом изменение имени регистра указывает на диапазон регистров, а не на новый регистр. Например, имя сдвоенного регистра XR1:0 указывает регистр R1 и R0 в вычислительном блоке X, а счетверенное имя регистра J31:28 указывает регистры J31, J30, J29, и J28 в J-IALU. Обязательно выравнивание границ адресов сдвоенных и счетверенных регистров. Для сдвоенного регистра его младший номер должен быть кратен двум (например, R1:0, R3:2, R27:26), а для счетверенного кратен 4 (например, R3:0, R7:4, R31:28). Следующий пример кода показывает загрузку, сохранение и передачу сдвоенного и счетверенного регистра.

```
YR5:4 = L [J31 +  0xf0] ;; /* загрузка YR5:4 из памяти */
XR3:0 = Q [K4 += K5] ;; /* загрузка XR3:0 из памяти */
L [J0 + J2] = YR31:30 ;; /* сохранение YR31:30 в память */
Q [K31 + K28] = YR7:4 ;; /* сохранение YR7:4 в память */
DCS0 = YR11:8 ;; /* передача YR11:8 в DCS0 (DP, DY, DX, DI) */
```

Обычно цикл загрузки или сохранения регистра в памяти происходит между регистром и словом памяти одинакового размера – 32, 64 и 128 бит. В этом случае операция называется «нормальный доступ чтения\записи».

Для использования преимуществ обработки «одна команда – много данных» (SIMD) архитектура шины процессора поддерживает типы доступа, в которых содержимое одного слова памяти может быть загружено в два регистра (широковещательное чтение), и тип доступа, в котором несколько регистров могут быть загружены\сохранены с различными данными, используя одно слово в памяти (доступ объединенного чтения или записи).

Пространство регистров состоит их 64 регистровых групп с количеством регистров в группе равным 32. Группы регистров определены в диапазоне 0x3F–0 (63–0). Группы 0x1F–0 (31-0) доступны для любых команд передачи данных (загрузка константой, пересылка регистра, загрузка и сохранение в памяти). Эти группы относятся к вычислительному ядру процессора.

Группы 0x3F–0x20 доступны только командам пересылки регистров и соответствуют регистрам некоторых периферийных устройств SOC-шины. Доступ к этим регистрам возможен как с помощью команд пересылки, так и с помощью адреса в пространстве адресов периферийных устройств.

Группы регистров вычислительного ядра:
- 0x00–0x09 вычислительные модули X и У;
- 0x0C–0x0F целочисленные J и K АЛУ;
- 0x0A,0x1B модуль отладки;
- 0x1A устройство управления;
- 0x1E– 0x1F устройство защиты памяти.

Группы регистров периферийных устройств.
- 0x20 – 0x23 контроллер DMA;
- 0x24 контроллер внешнего порта;
- 0x25 – 0x27 порты связи;
- 0x38, 0x39, и 0x3A контроллер прерываний;
- 0x3D модуль JTAG.

Прочие группы зарезервированы и не могут быть доступны для приложений, т.к. они могут вызвать непредсказуемую реакцию процессора.

Существует прямая связь между адресом универсального регистра в карте памяти и группой, в которой он находится. Важно знать номер группы, которой принадлежит регистр, т.к. они могут иметь различные ограничения доступа. Для определения номера группы регистров (см. Рисунок 11).

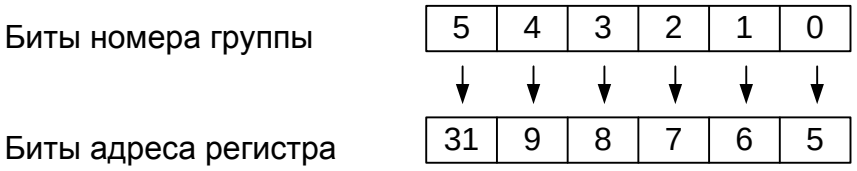


Рисунок 11 – Номер группы регистров в зависимости от адреса регистра

Важно обратить внимание на номер регистра в плане его (регистра) расположения в процессоре: в ядре процессора (внутренние универсальные регистры) или в периферийном модуле SOC-шины, т.к. регистры имеют различные ограничения доступа. Некоторые из регистров не используют все 32 бита. Неиспользованные биты зарезервированы. При записи в регистр с резервными битами резервные биты должны быть записаны нулевым значением. При чтении регистра с резервными битами резервные биты могут иметь любое значение.

Ниже приведен перечень групп регистров вычислительного ядра. Описание групп регистров периферийных модулей приведено в главах, посвященных конкретным периферийным устройствам.

6.5 Группы регистров вычислительных модулей

Файл регистров вычислительного модуля содержит 32 регистра общего назначения (РОН). Существует два подобных файла, по одному в каждом

вычислительном модуле (X и Y). Однако, фактически имея две группы по 32 регистра, регистры вычислительных модулей используют 10 номеров групп (с 0 по 9) для своей адресации. Можно сказать, что каждый регистр имеет собственное имя и несколько псевдонимов. Использование подобных псевдонимов удобно в командах пересылки регистров, а также в командах чтения/записи при работе с памятью. С помощью псевдонима происходит дополнительное кодирование выполняемой с регистром операции.

Ниже указаны виды доступов к регистрам вычислительного блока в соответствии с их распределением в память (Таблица 5 – Таблица 9). Номер регистра образуется конкатенацией номера группы (старшие 6 бит) и номера регистра в группе (младшие 5 бит).

Таблица 5 – Группа регистров вычислительного блока X

Имя	Номер	Значение по умолчанию	Имя	Номер	Значение по умолчанию
XR0	0x 0000	Не определено	XR16	0x0010	Не определено
XR1	0x 0001	Не определено	XR17	0x 0011	Не определено
XR2	0x 0002	Не определено	XR18	0x 0012	Не определено
XR3	0x 0003	Не определено	XR19	0x 0013	Не определено
XR4	0x 0004	Не определено	XR20	0x 0014	Не определено
XR5	0x 0005	Не определено	XR21	0x 0015	Не определено
XR6	0x 0006	Не определено	XR22	0x 0016	Не определено
XR7	0x 0007	Не определено	XR23	0x 0017	Не определено
XR8	0x 0008	Не определено	XR24	0x 0018	Не определено
XR9	0x 0009	Не определено	XR25	0x 0019	Не определено
XR10	0x 000A	Не определено	XR26	0x 001A	Не определено
XR11	0x 000B	Не определено	XR27	0x 001B	Не определено
XR12	0x 000C	Не определено	XR28	0x 001C	Не определено
XR13	0x 000D	Не определено	XR29	0x 001D	Не определено
XR14	0x 000E	Не определено	XR30	0x 001E	Не определено
XR15	0x 000F	Не определено	XR31	0x 001F	Не определено

Таблица 6 – Группа регистров вычислительного блока Y

Имя	Номер	Значение по умолчанию	Имя	Номер	Значение по умолчанию
YR0	0x 0040	Не определено	YR16	0x 0050	Не определено
YR1	0x 0041	Не определено	YR17	0x 0051	Не определено
YR2	0x 0042	Не определено	YR18	0x 0052	Не определено
YR3	0x 0043	Не определено	YR19	0x 0053	Не определено
YR4	0x 0044	Не определено	YR20	0x 0054	Не определено
YR5	0x 0045	Не определено	YR21	0x 0055	Не определено
YR6	0x 0046	Не определено	YR22	0x 0056	Не определено
YR7	0x 0047	Не определено	YR23	0x 0057	Не определено
YR8	0x 0048	Не определено	YR24	0x 0058	Не определено
YR9	0x 0049	Не определено	YR25	0x 0059	Не определено

YR10	0x 004A	Не определено	YR26	0x 005A	Не определено
YR11	0x 004B	Не определено	YR27	0x 005B	Не определено
YR12	0x 004C	Не определено	YR28	0x 005C	Не определено
YR13	0x 004D	Не определено	YR29	0x 005D	Не определено
YR14	0x 004E	Не определено	YR30	0x 005E	Не определено
YR15	0x 004F	Не определено	YR31	0x 005F	Не определено

Таблица 7 – Объединенная группа регистров вычислительного модуля XY

Имя	Адрес	Значение по умолчанию	Имя	Адрес	Значение по умолчанию
XYR0	0x 0080	Не определено	XYR16	0x 0090	Не определено
XYR1	0x 0081	Не определено	XYR17	0x 0091	Не определено
XYR2	0x 0082	Не определено	XYR18	0x 0092	Не определено
XYR3	0x 0083	Не определено	XYR19	0x 0093	Не определено
XYR4	0x 0084	Не определено	XYR20	0x 0094	Не определено
XYR5	0x 0085	Не определено	XYR21	0x 0095	Не определено
XYR6	0x 0086	Не определено	XYR22	0x 0096	Не определено
XYR7	0x 0087	Не определено	XYR23	0x 0097	Не определено
XYR8	0x 0088	Не определено	XYR24	0x 0098	Не определено
XYR9	0x 0089	Не определено	XYR25	0x 0099	Не определено
XYR10	0x 008A	Не определено	XYR26	0x 009A	Не определено
XYR11	0x 008B	Не определено	XYR27	0x 009B	Не определено
XYR12	0x 008C	Не определено	XYR28	0x 009C	Не определено
XYR13	0x 008D	Не определено	XYR29	0x 009D	Не определено
XYR14	0x 008E	Не определено	XYR30	0x 009E	Не определено
XYR15	0x 008F	Не определено	XYR31	0x 009F	Не определено

Таблица 8 – Объединенная группа регистров вычислительного модуля YX

Имя	Адрес	Значение по умолчанию	Имя	Адрес	Значение по умолчанию
YXR0	0x 00C0	Не определено	YXR16	0x 00D0	Не определено
YXR1	0x 00C1	Не определено	YXR17	0x 00D1	Не определено
YXR2	0x 00C2	Не определено	YXR18	0x 00D2	Не определено
YXR3	0x 00C3	Не определено	YXR19	0x 00D3	Не определено
YXR4	0x 00C4	Не определено	YXR20	0x 00D4	Не определено
YXR5	0x 00C5	Не определено	YXR21	0x 00D5	Не определено
YXR6	0x 00C6	Не определено	YXR22	0x 00D6	Не определено
YXR7	0x 00C7	Не определено	YXR23	0x 00D7	Не определено
YXR8	0x 00C8	Не определено	YXR24	0x 00D8	Не определено
YXR9	0x 00C9	Не определено	YXR25	0x 00D9	Не определено
YXR10	0x 00CA	Не определено	YXR26	0x 00DA	Не определено
YXR11	0x 00CB	Не определено	YXR27	0x 00DB	Не определено

YXR12	0x 00CC	Не определено	YXR28	0x 00DC	Не определено
YXR13	0x 00CD	Не определено	YXR29	0x 00DD	Не определено
YXR14	0x 00CE	Не определено	YXR30	0x 00DE	Не определено
YXR15	0x 00CF	Не определено	YXR31	0x 00DF	Не определено

Таблица 9 – Группа регистров широковещательного доступа вычислительного модуля XY

Имя	Адрес	Значение по умолчанию	Имя	Адрес	Значение по умолчанию
XYR0	0x 0100	Не определено	XYR16	0x 0110	Не определено
XYR1	0x 0101	Не определено	XYR17	0x 0111	Не определено
XYR2	0x 0102	Не определено	XYR18	0x 0112	Не определено
XYR3	0x 0103	Не определено	XYR19	0x 0113	Не определено
XYR4	0x 0104	Не определено	XYR20	0x 0114	Не определено
XYR5	0x 0105	Не определено	XYR21	0x 0115	Не определено
XYR6	0x 0106	Не определено	XYR22	0x 0116	Не определено
XYR7	0x 0107	Не определено	XYR23	0x 0117	Не определено
XYR8	0x 0108	Не определено	XYR24	0x 0118	Не определено
XYR9	0x 0109	Не определено	XYR25	0x 0119	Не определено
XYR10	0x 010A	Не определено	XYR26	0x 011A	Не определено
XYR11	0x 010B	Не определено	XYR27	0x 011B	Не определено
XYR12	0x 010C	Не определено	XYR28	0x 011C	Не определено
XYR13	0x 010D	Не определено	XYR29	0x 011D	Не определено
XYR14	0x 010E	Не определено	XYR30	0x 011E	Не определено
XYR15	0x 010F	Не определено	XYR31	0x 011F	Не определено

Отметим, что дополнительно имеются номера групп, которые используются командами альтернативного доступа с использованием буфера DAB и кольцевых буферов.

6.6 Регистры вычислительного модуля без номера

Есть несколько регистров в вычислительных модулях, которые не являются универсальными, поэтому не доступны посредством их адресации с помощью номера группы и номера в группе. Эти регистры доступны только при выполнении специальных команд, которые передают данные между ними и РОН вычислительного модуля.

6.6.1 Регистры статуса вычислительных модулей (XSTAT/YSTAT)

Регистры XSTAT и YSTAT – 32-битные регистры вычислительных модулей, которые хранят состояние флагов модулей. Каждый флаг регистра обновляется при выполнении соответствующей команды. Вычислительный модуль состоит из вычислительных блоков: ALU, умножитель, сдвигатель, CLU. Каждый из этих блоков

формирует специальные флаги-признаки результата операции. Регистр состояний хранит данные флаги. Флаги имеют разный тип. Большинство флагов регистра изменяет свое значение при выполнении соответствующих команд, однако есть специальные флаги (sticky), которые после их установки в 1-е значение уже невозможно сбросить. Сброс возможен только командой загрузки регистра. Stiky биты очень удобны, когда нужно в конце алгоритма проверить были ли аномальные ситуации при выполнении вычислений на протяжении всего алгоритма. В отличие от них другие флаги нужно проверять (если необходимо) сразу после выполнения операции. Подробное описание бит регистров X/YSTAT (значение после сброса = 0x0000 0000) приведено в Таблице 10.

Таблица 10 – Регистр состояния вычислительного модуля

Бит	Имя	Назначение
0	AZ	ALU. Признак нулевого результата операции
1	AN	ALU. Признак отрицательного результата операции
2	AV	ALU. Признак переполнения результата операции
3	AC	ALU. Признак переноса
4	MZ	Умножитель. Признак нулевого результата операции
5	MN	Умножитель. Признак отрицательного результата операции
6	MV	Умножитель. Признак переполнения результата операции
7	MU	Умножитель. Признак исчезновения (underflow) результата операции
8	SZ	Сдвигатель. Признак нулевого результата операции
9	SN	Сдвигатель. Признак отрицательного результата операции
10	BF0	Сдвигатель. Флаг блока операций с плавающей запятой
11	BF1	Сдвигатель. Флаг блока операций с плавающей запятой
12	AI	Ошибочная (Invalid) операция АЛУ с плавающей запятой
13	MI	Ошибочная (Invalid) операция умножителя с плавающей запятой
14	TROV	CLU. Переполнение операции trellis
15	TRSOV	CLU. Переполнение операции trellis. Stiky бит
16	-	
17	-	
18	-	
19	-	
20	UEN	Разрешение (если 1) генерации исключительной ситуации при возникновении флага исчезновения (underflow) результата
21	OEN	Разрешение (если 1) генерации исключительной ситуации при возникновении флага переполнения(overflow) результата
22	IVEN	Разрешение (если 1) генерации исключительной ситуации при возникновении флага ошибочnouj (invalid) результата при операциях с плавающей запятой
23	-	
Stiky биты		
24	AUS	Underflow в АЛУ с плавающей запятой
25	AVS	Переполнение в АЛУ с плавающей запятой
26	AOS	Переполнение в АЛУ с фиксированной точкой
27	AIS	Ошибочный результат в АЛУ с плавающей запятой
28	MUS	Underflow в умножителе с плавающей запятой
29	MVS	Переполнение в умножителе с плавающей запятой
30	MOS	Переполнение в умножителе с фиксированной точкой
31	MIS	Ошибочный результат в умножителе с плавающей запятой

6.6.2 Регистры АЛУ

Регистры параллельных результатов (PR0 и PR1) – 32-битные регистры, используемые для специальных команд суммирования.

6.6.3 Регистры умножителя

Регистры результатов умножителя (MR3–0 и MR4) используются для накопления при выполнении операций умножения с накоплением, т.е. выполняют функции аккумуляторов.

6.6.4 Регистры сдвигателя

Регистр BFOTMP – 64-битный регистр для команды PUTBITS, используемой для организации потока бит.

6.6.5 Регистры блока CLU

Модуль CLU использует дополнительный набор регистров для выполнения своих команд. Это регистры данных (TR31–0), регистры истории (THR3–0) и регистр управления (CMCTL). Каждый регистр имеют ширину 32 бита.

6.7 Группы регистров целочисленного АЛУ

Каждое целочисленное АЛУ имеет две группы универсальных регистров:
- 1-ая группа – файл регистров общего назначения, включающий 32 слова;
- 2-ая группа – регистры для указания параметров кольцевой буферизации.

Группы регистров целочисленного АЛУ показаны ниже (Таблица 11 – Таблица 14).

Файл регистров кольцевого буфера содержит следующие регистры:

3 ÷ 0 – кольцевого буфера JB3–0 и KB3–0;
7 ÷ 4 – длина кольцевого буфера JL3–0 и KL3–0;
26 ÷ 8 – резерв;
27 – регистр указателя стека режима пользователя (если включено разрешение на его использование);
31 ÷ 28 – резерв.

Таблица 11 – Группа регистров целочисленного АЛУ типа J (J-IALU)

Имя	Номер	Значение по умолчанию	Имя	Номер	Значение по умолчанию
J0	0x 0180	Не определено	J16	0x 0190	Не определено
J1	0x 0181	Не определено	J17	0x 0191	Не определено
J2	0x 0182	Не определено	J18	0x 0192	Не определено
J3	0x 0183	Не определено	J19	0x 0193	Не определено
J4	0x 0184	Не определено	J20	0x 0194	Не определено
J5	0x 0185	Не определено	J21	0x 0195	Не определено

J6	0x 0186	Не определено	J22	0x 0196	Не определено
J7	0x 0187	Не определено	J23	0x 0197	Не определено
J8	0x 0188	Не определено	J24	0x 0198	Не определено
J9	0x 0189	Не определено	J25	0x 0199	Не определено
J10	0x 018A	Не определено	J26	0x 019A	Не определено
J11	0x 018B	Не определено	J27	0x 019B	Не определено
J12	0x 018C	Не определено	J28	0x 019C	Не определено
J13	0x 018D	Не определено	J29	0x 019D	Не определено
J14	0x 018E	Не определено	J30	0x 019E	Не определено
J15	0x 018F	Не определено	J31	0x 019F	0x0000 0000

Таблица 12 – Группа регистров целочисленного АЛУ типа K (K-IALU)

Имя	Номер	Значение по умолчанию	Имя	Номер	Значение по умолчанию
K0	0x 01A0	Не определено	K16	0x 01B0	Не определено
K1	0x 01A1	Не определено	K17	0x 01B1	Не определено
K2	0x 01A2	Не определено	K18	0x 01B2	Не определено
K3	0x 01A3	Не определено	K19	0x 01B3	Не определено
K4	0x 01A4	Не определено	K20	0x 01B4	Не определено
K5	0x 01A5	Не определено	K21	0x 01B5	Не определено
K6	0x 01A6	Не определено	K22	0x 01B6	Не определено
K7	0x 01A7	Не определено	K23	0x 01B7	Не определено
K8	0x 01A8	Не определено	K24	0x 01B8	Не определено
K9	0x 01A9	Не определено	K25	0x 01B9	Не определено
K10	0x 01AA	Не определено	K26	0x 01BA	Не определено
K11	0x 01AB	Не определено	K27	0x 01BB	Не определено
K12	0x01AC	Не определено	K28	0x 01BC	Не определено
K13	0x01AD	Не определено	K29	0x 01BD	Не определено
K14	0x 01AE	Не определено	K30	0x 01BE	Не определено
K15	0x 01AF	Не определено	K31	0x 01BF	0x0000 0000

Таблица 13 – Группа регистров циклического буфера целочисленного АЛУ типа J (J-IALU)

Имя	Номер	Значение по умолчанию
JB0	0x 01C0	Не определено
JB1	0x 01C1	Не определено
JB2	0x 01C2	Не определено
JB3	0x 01C3	Не определено
JL0	0x 01C4	Не определено
JL1	0x 01C5	Не определено
JL2	0x 01C6	Не определено
JL3	0x 01C7	Не определено
резервные	0x 01C8– 0x 01DA	Не определено
JUSP	0x 01DB	Не определено
резервные	0x 01DC– 0x 01DF	Не определено

Таблица 14 – Группа регистров циклического буфера K-IALU

Имя	Номер	Значение по умолчанию
KB0	0x 01E0	Не определено
KB1	0x 01E1	Не определено
KB2	0x 01E2	Не определено
KB3	0x 01E3	Не определено
KL0	0x 01E4	Не определено
KL1	0x 01E5	Не определено
KL2	0x 01E6	Не определено
KL3	0x 01E7	Не определено
резервные	0x 01E8– 0x 01FA	Не определено
KUSP	0x 01FB	Не определено
резервные	0x 01FC– 0x 01FF	Не определено

6.7.1 Регистры статуса целочисленного АЛУ (J31/JSTAT и K31/KSTAT)

Регистры статуса JSTAT (для J-IALU) и KSTAT (для K-IALU) хранят состояния флагов и обновляются в результате выполнения различных команд целочисленного АЛУ. При записи возможно обращение к регистру JSTAT как J31. При использовании в качестве операнда в арифметических, логических и функциональных операциях J31 рассматривается как ноль. Регистры J31/JSTAT (K31/KSTAT) (значение сброса = 0x0000 0000) описаны ниже (Таблица 15).

Таблица 15 – Регистр состояния целочисленного АЛУ

Бит	Имя	Назначение
0	Z	Признак нулевого результата операции
1	N	Признак отрицательного результата операции
2	V	Признак переполнения результата операции
3	C	Признак переноса
4	-	Резервные. Всегда 0

6.8 Группы регистров устройства управления

Группа содержит 32 регистра, однако используются не все регистры. Часть регистров зарезервирована. Все регистры группы выполняют специальные функции. Особенностью данной группы является то, что регистры доступны только как однословные.

Таблица 16 – Группа регистров устройства управления

Имя	Описание	Номер	Значение по умолчанию
CJMP	Вычисленный адрес перехода	0x 0340	Не определено
-	-	0x 0341	-
RETI	Возврат из прерывания	0x 0342	Не определено
RETIB	Псевдоним RETI, для вложенных прерываний	0x 0343	Не определено
RETS	Возврат из исключительной ситуации	0x 0344	Не определено
DBGE	Возврат из эмуляции	0x 0345	Не определено
-	-	0x 0346– 0x 0347	-
LC0	Счетчик циклов 0	0x 0348	Не определено

LC1	Счетчик циклов 1	0x 0349	Не определено
-	-	0x 034A– 0x 034F	-
IVSW	Адрес-вектор для программных исключений	0x 0350	Не определено
-	-	0x 0351– 0x 0353	-
FLAGREG	Регистр управления флагами процессора	0x 0354	0x0000 0000
FLAGREGST	Установка бит регистра флага	0x 0355	Не определено
FLAGREGCL	Сброс бит регистра флага	0x 0356	Не определено
-	-	0x 0357	-
SQCTL	Регистр управления	0x 0358	0x0000 0204
SQCTLST	Установка бит регистра управления	0x 0359	Не определено
SQCTLCL	Сброс бит регистра управления	0x 035A	Не определено
SQSTAT	Регистр статуса, только чтение	0x 035B	0x0000 FF04
SFREG	регистр специальных статических флагов	0x 035C	0x0000 0000
-	-	0x 035D	-
EXT_FUN	Включение расширенных функций	0x 035E	-
-	-	0x 035F	-

6.8.1 Регистр управления флагом (FLAGREG)

Регистр FLAGREG – 32-битный регистр, который контролирует направление контакта флага (вход или выход) и обеспечивает значение на контакте (1 или 0), когда контакт сконфигурирован как выход. Регистр FLAGREG (значение сброса = 0x0000 0000) описывается ниже (Таблица 17).

Таблица 17 – Биты регистра управления флагом

Бит	Имя	Назначение
3:0	FLAGx_EN	Направление флага 0 – прием 1 – выдача
7:4	FLAGx_OUT	Значение флага при выдаче
31–8	-	Не используются и всегда равны нулю

Контакт FLAG0 использует биты 0 и 4, FLAG1 использует биты 1 и 5, FLAG2 использует биты 2 и 6, FLAG3 использует биты 3 и 7.

6.8.2 Регистр управления SQCTL

Устройство управления последовательностью команд контролируется значением регистра SQCTL. После сброса значение регистра равно 0x0000_0204. Подробное описание всех бит регистра приведено ниже (Таблица 18).

Таблица 18 – Биты регистра управления SQCTL

Бит	Имя	Назначение
0	-	
1	-	
2	GIE	Глобальное разрешение прерываний: 1 – прерывания разрешены; 0 – запрещены

3	SWIE	Разрешение генерации исключительных ситуаций (программных прерываний): 1 – разрешено; 0 – запрещено
4	-	
5	-	
6	-	
7	KUSP	Выбор указателя стека в режиме супервизора (NMOD=1) привключенном режиме работы с раздельными указателями (USP=1): 0 – используется регистр KSP; 1 – используется регистр USP
8	DBGEN	Разрешение отладки: 1 – разрешено; 0 – запрещено
9	NMOD	Режим работы: 0 – пользователь; 1 – супервизор
10	TRCBEN	Включение буфера трасс: 0 – буфер трасс выключен и хранит старое значение; 1 – включен и следит за исполнением переходов
11	TRCBEXEN	Разрешение генерации исключительной ситуации от буфера трасс: 0 – запрещено; 1 – разрешено
12	DF_IEEE	Разрешение использования команд обработки данных с плавающей запятой двойной точности: 0 – используется расширенная точность; 1 – код команды расширенной точности соответствует коду команды двойной точности
13	BYTE_ON	Разрешение использования кодов команды TRAP 32-63 в качестве кодов префикса для создания новых команд: 0 – команды TRAP 32-63 используются для генерации исключений; 1 – команды TRAP 32-63 используются в качестве префикса.
14	USP	Разрешение использования двух указателей стеков (регистр JK27): 0 – регистр JK27 один для пользователя и супервизора; 1 – для пользователя и супервизора используются разные регистры
31-15	-	Зарезервированы и не используются. При чтении всегда равны нулю

Биты 12, 13, 14 регистра управления соответствуют дополнительным возможностям процессора, и их включение должно быть дополнительно разрешено (см. бит EXT_MODE регистра SQSTAT, а также регистр EXT_FUN).

6.8.3 Регистр управления (SQCTLST). Установка бит

Бит SQCTLST является псевдонимом для SQCTL. При записи по данному адресу ‘1’ в любом бите записываемых данных устанавливает соответствующий бит в SQCTL, тогда как ‘0’ в записываемых данных не меняет значение бита.

6.8.4 Регистр управления (SQCTLCL). Сброс бит

Бит SQCTLCL является псевдонимом для записи в SQCTL. При записи по данному адресу ‘0’ в любом бите записываемых данных сбрасывает

соответствующий бит в SQCTL, тогда как ‘1’ в записываемых данных не меняет значение бита.

6.8.5 Регистр статических флагов (SFREG)

Статические флаги используются как статические копии условий. При желании сохранить значение условия до того, как другая команда поменяет его значение, можно скопировать значение в регистр SFREG и использовать его позже как условие. Все статические флаги условий группируются в регистре SFREG. Регистр SFREG после сброса равен нулю и описание его разрядов приведено ниже (Таблица 19).

Таблица 19 – Биты регистра статических флагов

Бит	Имя	Назначение
0	GSCF0	Статический флаг целочисленных АЛУ
1	GSCF1	Статический флаг целочисленных АЛУ
2	XSCF0	Статический флаг вычислительного модуля X
3	XSCF1	Статический флаг вычислительного модуля X
4	YSCF0	Статический флаг вычислительного модуля У
5	YSCF1	Статический флаг вычислительного модуля У
31-6	-	Не используются и всегда равны нулю

6.8.6 Регистр включения расширенных функций (EXT_FUN)

Регистр состоит из одного бита, значение которого может быть прочитано в регистре SQSTAT, бит 30 – EXT_MODE. Этот бит по сбросу устанавливается в 1(все расширенные функции включены) и меняет свое значение на противоположное всякий раз после выполнения команды 0x8B5E1A3C. Этот код суть команда пересылки из регистра EXT_FUN в этот же регистр EXT_FUN (EXT_FUN=EXT_FUN).

6.8.7 Регистр статуса устройства управления (SQSTAT)

Данный регистр доступен только для чтения и содержит информацию о текущем статусе устройства управления. Подробное описание разрядов регистра приведено ниже (Таблица 20).

Таблица 20 – Биты регистра SQSTAT

Бит	Имя	Назначение
1:0	MODE	Текущий режим работы процессора: 00 – пользователь 01 – супервизор 11 – эмулятор Другие состояния невозможны
2	IDLE	Процессор в состоянии ожидания прерывания (если 1)
7:3	SPVCMD	Значение параметра последней выполненной команды TRAP

11:8	EXCAUSE	Код последней исключительной ситуации: 0000 – команда TRAP 0001 – точка наблюдения или буфер трасс 0010 – команда плавающей запятой 0011 – ошибочная линия команд 0100 – невыровненный доступ 0101 – попытка записи защищенного регистра 0110 – переполнения счетчика статистики 0111 – ошибочный доступ в IALU 1000 – x 1001 – x 1010 – x 1011 – x 1100 – x 1101 – x 1110 – x 1111 – не было ошибок с момента сброса
15:12	EMCAUSE	Код ситуации вызвавшей переход в режим эмулятора: 1111 – не было перехода в режим эмулятора после сброса 0000 – команда EMUTRAP 0001 – запрос JTAG 0010 – сработала точка наблюдения Все другие коды зарезервированы и не могут присутствовать
16	FLAG0	Состояние внешнего контакта FLAG0
17	FLAG1	Состояние внешнего контакта FLAG1
18	FLAG2	Состояние внешнего контакта FLAG2
19	FLAG3	Состояние внешнего контакта FLAG3
20	EXE_ISR	Флаг устанавливается в 1 при: - переходе процессора в состояние обработки прерывания; - записи данных в регистр RETIB. Установленный флаг запрещает прерывания. Флаг сбрасывается: - командой выхода из прерывания RTI; - командой RDS (если нет обработки исключительной ситуации); - чтением регистра RETIB. Сброшенный флаг означает возможность обработки нового прерывания процессором
21	EXE_SWI	Признак того что процессор находится в состоянии обработки исключительной ситуации (когда равен 1)
22	EMUL	Признак того что процессор находится в режиме эмулятора (когда равен 1)
23	ISR_MODE	Флаг устанавливается в 1 при переходе процессора в состояние обработки прерывания. Флаг сбрасывается: - командой выхода из прерывания RTI; - командой RSD (если нет обработки исключиткльной ситуации). Установленный флаг означает состояние обработки прерывания в режиме супервизора.
24	BTBEN	
25	-	BTBLK
26	-	
27	-	
28	BTBEN	Буфер предсказания переходов включен (если 1) или выключен (если 0)
29	-	BTBLK

30	EXT_MODE	Флаг включения расширенных операций процессора. После аппаратного сброса значение 1 (расширенные операции разрешены)
31	I_LOCK	Флаг блокировки прерываний (1 – прерывания блокированы)

6.9 Группа регистров устройства защиты памяти

Процессор имеет несколько режимов работы и особенностью режима пользователя является то, что для него ограничен доступ к некоторым аппаратным ресурсам. Однако внутренняя и внешняя память одинаково доступны как в режиме супервизора, так и в режиме пользователя. В ряде приложений желательно защитить код или данные системы от разрушения некорректным поведением пользовательской программы. Для этого в процессоре имеется группа регистров, с помощью которой можно описать отдельные регионы внутренней памяти и способы доступа к ним. Также эта группа регистров содержит регистры управления кэш-памятью. Все регистры данной группы доступны только как однословные. Запись к регистрам разрешена только если включен режим использования расширенных функций процессора (бит регистра EXT_FUN).

Подробное описание регистров приведено в разделе 7 «Архитектура кэш-памяти процессора».

Таблица 21 – Группа регистров устройства защиты памяти

Имя	Описание	Номер	Значение по умолчанию
Группа 0x1E			
MS0_C	Регистр управления кэшированием данных MS0	0x03C0	0
MS0_WT	Регистр управления сквозной записью MS0	0x03C1	0
MS0_CI	Регистр управления кэшированием команд MS0	0x03C2	0
		0x03C3 – 0x03C7	
MS1_C	Регистр управления кэшированием данных MS1	0x03C8	0
MS1_WT	Регистр управления сквозной записью MS1	0x03C9	0
MS1_CI	Регистр управления кэшированием команд MS1	0x03CA	0
		0x03CB – 0x03CF	
SDR_C	Регистр управления кэшированием данных SDRAM	0x03D0	0
SDR_WT	Регистр управления сквозной записью SDRAM	0x03D1	0
SDR_CI	Регистр управления кэшированием команд SDRAM	0x03D2	0
		0x03D3 – 0x03DF	
Группа 0x1F			
PU0	Регистр защиты 0	0x03E0	Не определено
PU1	Регистр защиты 1	0x03E1	Не определено
PU2	Регистр защиты 2	0x03E2	Не определено
PU3	Регистр защиты 3	0x03E3	Не определено
PU4	Регистр защиты 4	0x03E4	Не определено
PU5	Регистр защиты 5	0x03E5	Не определено
PU6	Регистр защиты 6	0x03E6	Не определено
PU7	Регистр защиты 7	0x03E7	Не определено
PU_SR	Регистр состояния	0x03E8	0

Имя	Описание	Номер	Значение по умолчанию
-	-	0x03E9 – 0x03FB	0
PU_CR	Регистр управления	0x03FC	0
IDC_CR	Управление кэш-памятью	0x03FD	0
-	-	0x03FE – 0x03FF	0

6.9.1 Регистры защиты (PUx)

Каждый регистр защиты позволяет описать определенную область памяти и способы доступа к ней. Работа регистра защиты включается соответствующим битом в регистре управления. Подробное описание бит регистров защиты приведено ниже (Таблица 22).

Таблица 22 – Регистр PU

Бит	Имя	Назначение
10:0	STA	Начальный адрес модуля памяти. Соответствует физическому адресу памяти образуемому как {11'b0,START_A[10:0],10'b0}
11	-	
22:12	ENDA	Конечный адрес модуля памяти. Соответствует физическому адресу памяти образуемому как {11'b0,END_A[10:0],10'b11_1111_1111}
23	-	
25:24	JK_AP	Режим доступа к модулю со стороны шин J и K. 00 – полный доступ 01 – супервизор чтение и запись, пользователь чтение 10 – всем только чтение 11 – супервизор только чтение
27:26	I_AP	Режим доступа к модулю со стороны шины I. 00 – полный доступ 01 – только супервизор 10 – доступ запрещен 11 – доступ запрещен
29:28	H_AP	Режим доступа к модулю со стороны шины S (хост, DMA). 00 – чтение и запись 01 – только чтение 10 – доступ запрещен 11 – доступ запрещен
31:30	-	Резерв

Поля STA и ENDA используются для сравнения со значениями адресных шин J, K, I, S. При этом в сравнении участвуют только биты 20:10 указанных шин. Проверка выполняется только при доступе к внутренней памяти. Так для K шины доступа попадание в некоторый модуль означает истинность следующего выражения:

Hit_PU = (K[20:10]>=STA) && (K[20:10]<=ENDA).

Минимальный размер модуля равен странице из 1 К слов. Модуль может рассматриваться как множество из 1 К страниц.

При нарушении прав доступа к модулю памяти вырабатывается исключительная ситуация. Исключение составляет контроль доступа со стороны шины S. Нарушение прав доступа будет вызывать блокировку записи и установку

флага ошибки. Исключительная ситуация вырабатываться не будет. Необходимо отметить, что исключительная ситуация всегда форсируется после выполнения команды. Поэтому в случае срабатывания защиты по доступу со стороны шины I, процессор выполнит одну линию команд из защищенной области, прежде чем перейдет на обработку исключительной ситуации. Разобраться какое событие вызвало срабатывание исключительной ситуации можно анализируя регистр состояния устройства управления, а также регистр состояния модуля защиты.

6.9.2 Регистр состояния (PU_SR)

Регистр хранит информацию о событии, которое вызвало срабатывание модуля защиты. После сброса значение регистра равно нулю. Подробное описание бит регистра состояния приведено в разделе 7 «Архитектура кэш-памяти процессора».

6.9.3 Регистр управления (PU_CR)

Регистр осуществляет включение в работу всех функций модуля защиты, а также функций управления кэш-памятью. После сброса значение регистра равно нулю. Подробное описание бит регистра управления приведено в разделе 7 «Архитектура кэш-памяти процессора».

6.9.4 Регистр управления кэшированием данных

Регистр используется для задания атрибута кэшируемости для отдельных страниц банков внешней памяти. Как известно процессор может адресовать три банка внешней памяти: MS0, MS1, SDRAM. При этом модули MS0, MS1 имеют деление на страницы по 128 К слов, а модуль SDRAM имеет деление на страницы по 512 К слов. Каждый бит регистра соответствует странице памяти. Так 0-й бит нулевой странице, 1-й первой и т. д. Итого можно определить до 32•128 = 4 М слов памяти модулей MS0, MS1 и до 16 М слов динамической памяти.

6.9.5 Регистр управления сквозной записью

Аналогично управлению кэшируемости страниц, имеется возможность для каждой страницы указать бит стратегии записи при попадании в кэш: 0 – запись только в кэш, 1 – запись в кэш и во внешнюю память.

6.9.6 Регистр управления кэшированием команд

Аналогично управлению кэшируемостью страниц для данных имеется возможность для каждой страницы указать бит кэшируемости при обращении за командами: 0 – не кэшируется, 1 – кэшируется.

Более подробно о функциях регистров управления кэшами см. приведено в разделе 7 «Архитектура кэш-памяти процессора».

6.10 Группы регистров отладки

Группы отладки описываются ниже (Таблица 23). Доступ к регистрам отладки возможет только как к отдельным словам. К регистрам отладки может быть применена только команда пересылки «регистр-регистр» или командам немедленной загрузки данных. Эти регистры не могут быть загружены из памяти или

сохранены напрямую в память. После каждой записи в регистры группы отладки происходит перезапуск конвейера процессора.

Таблица 23 – Группа регистров отладки

Имя	Описание	Адрес	Значение по умолчанию
WP0CTL	Управление точкой наблюдения 0	0x 0360	0x0000 0000
WP1CTL	Управление точкой наблюдения 1	0x 0361	0x0000 0000
WP2CTL	Управление точкой наблюдения 2	0x 0362	0x0000 0000
-	-	0x 0363	-
WP0STAT	Состояние точки наблюдения 0; (RO)	0x 0364	0x0000 0000
WP1STAT	Состояние точки наблюдения 1; (RO)	0x 0365	0x0000 0000
WP2STAT	Состояние точки наблюдения 2; (RO)	0x 0366	0x0000 0000
-	-	0x 0367	-
WP0L	Нижний адрес точки наблюдения 0	0x 0368	Не определено
WP0H	Верхний адрес точки наблюдения 0	0x 0369	Не определено
WP1L	Нижний адрес точки наблюдения 1	0x 036A	Не определено
WP1H	Верхний адрес точки наблюдения 1	0x 036B	Не определено
WP2L	Нижний адрес точки наблюдения 2	0x 036C	Не определено
WP2H	Верхний адрес точки наблюдения 2	0x 036D	Не определено
WPDR	Регистр данных точки наблюдения 1	0x 036E	0
WPMR	Регистр маски точки наблюдения 1.	0x 036F	0
CCNT0	Нижняя часть счетчика циклов	0x 0370	0
CCNT1	Верхняя часть счетчика циклов	0x 0371	0
PRFM	Маска монитора производительности	0x 0372	0x0000 0000
PRFCNT	Счетчик монитора производительности	0x 0373	0x0000 0000
TRCBMASK	Маска буфера трассировки (RO)	0x 0374	0x0000 0000
TRCBPTR	Указатель буфер трассировки (RO)	0x 0375	0x0000 0000
-	-	0x 0376 – 0x 037B	-
TRCBVAL	Допустимый буфер трассировки (RO)	0x 037C	0x0000 0000
-	-	0x 037D – 0x 037F	-
TRCB31–0	Буфер трассировки (RO)	0x 0140 – 0x015F	0x0000 0000

Точка наблюдения представляет собой механизм описания условия, при выполнении которого модуль отладки формирует запрос к процессору на обработку исключительной ситуации или на переход в режим эмуляции.

6.10.1 Регистр управления точкой наблюдения (WPxCTL)

У каждой из трех точек наблюдения есть регистр управления (WP0CTL, WP1CTL и WP2CTL), который используется для определения действий точки наблюдения. Значением сброса регистров WPxCTL является 0x0000 0000. Каждая из точек наблюдения осуществляет мониторинг определенных шин процессорного ядра: точка 0 – шину I, точка 1 – шины J и K, точка 2 – шину S.

Подробное описание разрядов регистра WP0CTL приведено в таблице 24.

Таблица 24 – Регистр WP0CTL

Бит	Имя	Назначение
1:0	OPMODE	Режим работы: 00 – выключена 01 – включена. Анализ совпадения адреса 10 – включена. Анализ попадания в интервал 11 – включена. Анализ выхода вне интервала

Бит	Имя	Назначение
3:2	EXTYPE	Действия при выполнении условия: 00 – нет действий 01 – программное исключение 10 – переход в режим эмулятора 11 – резерв
4	SSTP	Режим пошагового выполнения команд: 1 – включен 0 – выключен
15:5	-	Зарезервировано
31:16	WP0CNT	Счетчик количества срабатываний точки наблюдения до генерации исключительной ситуации

Подробное описание разрядов регистра WP1CTL приведено в таблице 25.

Таблица 25 – Регистр WP1CTL

Бит	Имя	Назначение
1:0	OPMODE	Режим работы: 00 – выключена 01 – включена. Анализ совпадения адреса 10 – включена. Анализ попадания в интервал 11 – включена . Анализ выхода вне интервала
3:2	EXTYPE	Действия при выполнении условия: 00 – нет действий 01 – программное исключение 10 – переход в режим эмулятора 11 – резерв
4	READ	Мониторинг циклов чтения: 1 – включен 0 - выключен
5	WRITE	Мониторинг циклов записи: 1 – включен 0 – выключен
6	Jbus	Мониторинг шины J: 1 – включен 0 – выключен
7	Kbus	Мониторинг шины K: 1 – включен 0 – выключен
15:8	-	Зарезервировано
31:16	WP1CNT	Счетчик количества срабатываний точки наблюдения до генерации исключительной ситуации

Подробное описание разрядов регистра WP2CTL приведено в таблице 26.

Таблица 26 – Регистр WP2CTL

Бит	Имя	Назначение
1:0	OPMODE	Режим работы: 00 – выключена 01 – включена. Анализ совпадения адреса 10 – включена. Анализ попадания в интервал 11 – включена. Анализ выхода вне интервала
3:2	EXTYPE	Действия при выполнении условия: 00 – нет действий 01 – программное исключение 10 – переход в режим эмулятора 11 – резерв

Бит	Имя	Назначение
4	READ	Мониторинг циклов чтения: 1 – включен 0 – выключен
5	WRITE	Мониторинг циклов записи: 1 – включен 0 – выключен
15:6	-	Зарезервировано
31:16	WP2CNT	Счетчик количества срабатываний точки наблюдения до генерации исключительной ситуации

6.10.2 Регистры состояния точки наблюдения (WPxSTAT)

У каждой из трех точек наблюдения есть регистр состояния (WP0STAT, WP1STAT и WP2STAT). Значениями сброса регистров WPxSTAT являются 0x0000 0000. Подробное описание разрядов регистра приведено ниже (Таблица 27).

Таблица 27 – Регистр WPxSTAT

Бит	Имя	Назначение
15:0	WPxCNT	Текущее значение счетчика точки наблюдения. Счетчик вычитает 1 каждый раз при срабатывании точки
17:16	EX	Состояние точки наблюдения: 00 –выключена 01 – активна 10 – резерв 11 – переполнился счетчик
31:18	-	Всегда 0

6.10.3 Регистры указателя адреса точки наблюдения (WPxL/WPxH)

Указатели адреса точки наблюдения являются 32-битными указателями, определяющими адрес или диапазон адресов точки наблюдения. После сброса их значение не определено.

6.10.4 Регистр маски монитора производительности (PRFM)

Регистр маски монитора производительности (PRFM) определяет, какие события в процессоре отслеживаются и подсчитываются счетчиком монитора производительности (PRFCNT). Значением сброса регистра PRFM является 0x0000 0000. Подробное описание разрядов регистра приведено ниже (Таблица 28).

Таблица 28 – Регистр PRFM

Бит	Имя	Назначение
7:0	-	Резерв
8	Jexe	Подсчет команд выполненных на линии конвейера J 1 – включен 0 – выключен
9	Kexe	Подсчет команд выполненных на линии конвейера K 1 – включен 0 – выключен
10	Xexe	Подсчет команд выполненных на линиях конвейера X1 и X2 1 – включен 0 – выключен

Бит	Имя	Назначение
11	Yexe	Подсчет команд выполненных на линиях конвейера У1 и У2 1 – включен 0 – выключен
12	Sexe	Подсчет команд выполненных на линии конвейера S 1 – включен 0 – выключен
15:13	-	Резерв
16	STALL	Подсчет команд остановов конвейера вызванных ожиданием данных или другими блокирующими конвейер ситуациями 1 – включен 0 – выключен
17	BTB_true	Подсчет количества правильных предсказаний буфера переходов 1 – включен 0 – выключен
18	ABORT	Подсчет количества программных исключительных ситуаций 1 – включен 0 – выключен
19	Uexe	Подсчет количества линий команд выполненных в режиме пользователя 1 – включен 0 – выключен
20	BTB_false	Подсчет количества ошибочных предсказаний буфера переходов 1 – включен 0 – выключен
21	BTB_load	Подсчет количества загрузок в буфер переходов 1 – включен 0 – выключен
31:22	-	Резерв

Все события, которые отслеживаются монитором производительности, суммируются по «ИЛИ» и при их наступлении происходит увеличение счетчика PRFM на 1. Логичным является разрешение подсчета одного условия из многих.

6.10.5 Регистр счетчика монитора производительности (PRFCNT)

Назначение счетчика производительности – увеличивать свое значение на 1 каждый раз, когда происходит отслеживаемое событие. Отслеживаемое событие определяется регистром маски монитора производительности (PRFM). Регистр очищается после сброса.

6.10.6 Регистры счетчика циклов (CCNTx)

Длина счетчика циклов равна 64 разряда, однако доступен он как два универсальных 32-разрядных регистра – CCNT0 и CCNT1. После сброса значение счетчика равно нулю.

В связи с тем, что счетчик 64 бита, а доступ возможен только к частям регистра, необходимо соблюдение специальной процедуры чтения и записи регистра. При чтении сначала считывается нижняя часть (CCNT0), а затем верхняя часть (CCNT1). При чтении нижней части верхняя часть копируется в буфер и это гарантирует ее корректное значение. При записи сначала пишется нижняя часть (она попадает в буфер), а затем верхняя. При записи верхней полное 64-битное значение попадает в счетчик.

6.10.7 Регистры указателя и буфера трассировки (TRCBx/TRCBPTR)

Каждый раз, когда процессор выполняет переход на непоследовательно исполняемую команду, адрес непоследовательно выбранной команды (адрес перехода) записывается в один из регистров буфера трассировки. Первая запись осуществляется в буфер трассировки 0, вторая – в 1 и далее циклично. Указатель буфера трассировки определяет последний записанный буфер трассировки.

Таблица 29 – Группа регистров буфера трассировки

Имя	Описание	Адрес	Значение по умолчанию
TRCB0	Буфер трассировки 0; только чтение	0x 0140	0x0000 0000
TRCB1	Буфер трассировки 1; только чтение	0x 0141	0x0000 0000
TRCB2	Буфер трассировки 2; только чтение	0x 0142	0x0000 0000
TRCB3	Буфер трассировки 3; только чтение	0x 0143	0x0000 0000
TRCB4	Буфер трассировки 4; только чтение	0x 0144	0x0000 0000
TRCB5	Буфер трассировки 5; только чтение	0x 0145	0x0000 0000
TRCB6	Буфер трассировки 6; только чтение	0x 0146	0x0000 0000
TRCB7	Буфер трассировки 7; только чтение	0x 0147	0x0000 0000
TRCB8	Буфер трассировки 8; только чтение	0x 0148	0x0000 0000
TRCB9	Буфер трассировки 9; только чтение	0x 0149	0x0000 0000
TRCB10	Буфер трассировки 10; только чтение	0x 014A	0x0000 0000
TRCB11	Буфер трассировки 11; только чтение	0x 014B	0x0000 0000
TRCB12	Буфер трассировки 12; только чтение	0x 014C	0x0000 0000
TRCB13	Буфер трассировки 13; только чтение	0x 014D	0x0000 0000
TRCB14	Буфер трассировки 14; только чтение	0x 014E	0x0000 0000
TRCB15	Буфер трассировки 15; только чтение	0x 014F	0x0000 0000
TRCB16	Буфер трассировки 16; только чтение	0x 0150	0x0000 0000
TRCB17	Буфер трассировки 17; только чтение	0x 0151	0x0000 0000
TRCB18	Буфер трассировки 18; только чтение	0x 0152	0x0000 0000
TRCB19	Буфер трассировки 19; только чтение	0x 0153	0x0000 0000
TRCB20	Буфер трассировки 20; только чтение	0x 0154	0x0000 0000
TRCB21	Буфер трассировки 21; только чтение	0x 0155	0x0000 0000
TRCB22	Буфер трассировки 22; только чтение	0x 0156	0x0000 0000
TRCB23	Буфер трассировки 23; только чтение	0x 0157	0x0000 0000
TRCB24	Буфер трассировки 24; только чтение	0x 0158	0x0000 0000
TRCB25	Буфер трассировки 25; только чтение	0x 0159	0x0000 0000
TRCB26	Буфер трассировки 26; только чтение	0x 015A	0x0000 0000
TRCB27	Буфер трассировки 27; только чтение	0x 015B	0x0000 0000
TRCB28	Буфер трассировки 28; только чтение	0x 015C	0x0000 0000
TRCB29	Буфер трассировки 29; только чтение	0x 015D	0x0000 0000
TRCB30	Буфер трассировки 30; только чтение	0x 015E	0x0000 0000
TRCB31	Буфер трассировки 31; только чтение	0x 015F	0x0000 0000

6.10.8 Регистр данных точки наблюдения 1 (WPDR)

Точка наблюдения 1 имеет дополнительные возможности по мониторингу обмена данными ядра процессора с памятью. Выше описывался механизм срабатывания исключительной ситуации в случае совпадения адресов точки наблюдения с адресами шин J и K. Однако имеется возможность дополнить

сравнение адресов еще и сравнением данных. При чтении имеется возможность контролировать данные, которые считывает процессор. Если дополнительно к совпадению адресов происходит совпадение прочитанных данных – точка срабатывает. Также можно контролировать и процесс записи данных. Необходимые для контроля данные записываются в регистр WPDR. Необходимо отметить, что шина чтения или записи имеет разрядность 128 бит. Разрядность регистра данных только 32 бита. Поэтому сравнение выполняется только для 32-разрядного слова шины данных определяемого младшими битами шины адреса независимо от того, какое действие выполняется: чтение или запись 64-х или 128-разрядного слова.

6.10.9 Регистр маски точки наблюдения 1 (WPMR)

При использовании шины данных в описании точки наблюдения 1 не все биты шины данных могут понадобиться при анализе читаемых или записываемых данных. Регистр маски позволяет установить, какие биты регистра данных WPDR должны участвовать в сравнении. Если бит установлен в 1 – соответствующий ему бит данных сравнивается с битом шины данных. После сброса регистр маски равен нулю и данные не участвуют в работе точки наблюдения.

7 Архитектура кэш-памяти процессора

Процессор имеет встроенные раздельные кэш команд и кэш данных для кэширования данных внешней памяти. В разделе «Ядро процессора» была подробно описана структура конвейера процессора. Отметим основные стадии конвейера, которые задействуются при работе с внутренней памятью (Рисунок 12).

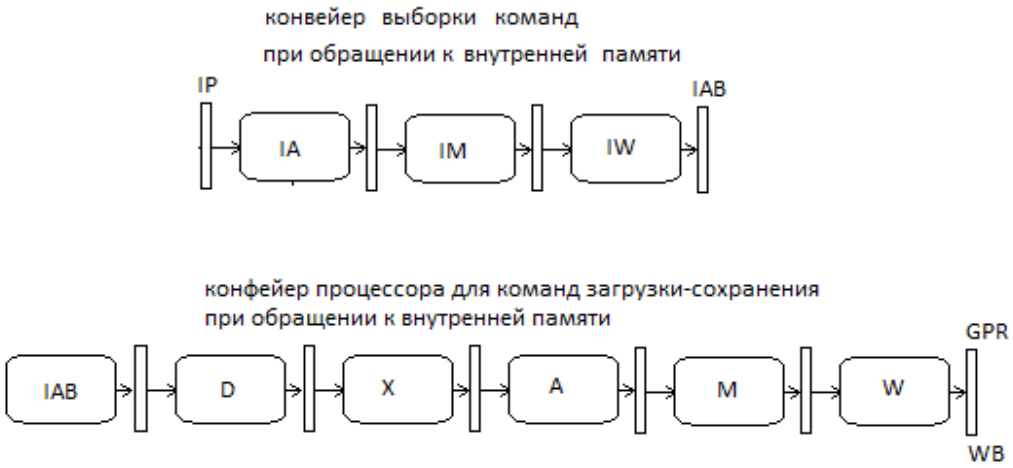


Рисунок 12 – Стадии конвейера процессора

Источником адреса для конвейера выборки команд является указатель на команду (IP). На стадии IA адрес анализируется и при обращении к внутренней памяти направляется в выбранный банк внутренней памяти. На стадии IM происходит чтение накопителя банка памяти и прочитанные данные загружаются в буфер. На стадии IW происходит передача данных из буфера памяти в буфер IAB. Работа вышеописанных стадий команд инициируется только буфером IAB: если в буфере имеется свободное место – формируется запрос на чтение команды.

Буфер команд IAB является источником команд для конвейера обработки. Для команд загрузки (чтения) данных из памяти и для команд сохранения (записи) данных в памяти стадии конвейера представлены на Рисунке 12:

- На стадии D происходит декодирование команды и определение операндов источников операции;
- На стадии X происходит вычисление адреса, по которому будет происходить обращение к памяти;
- На стадии A выполняется анализ адреса и определение факта обращения во внутреннюю память;
- При чтении на стадии M выполняется чтение данных из накопителя выбранного банка памяти в буфер;
- На стадии W происходит передача данных из буфера памяти в регистр-приемник процессорного ядра. Если выполняется операция записи, на стадии W происходит запись данных в буфер записи WB.

При обращении к внутренней памяти каждая стадия конвейера выполняется за один такт частоты процессора.

Кэш команд и кэш данных процессора включаются в работу, когда на стадии IA (A) обнаруживается обращение к внешней памяти процессора. В этом случае на стадиях конвейера выборки команд IM и IW, а также на стадиях конвейера выборки данных M и W выполняются другие операции.

Рассмотрим конвейер выборки команд (Рисунок 13).

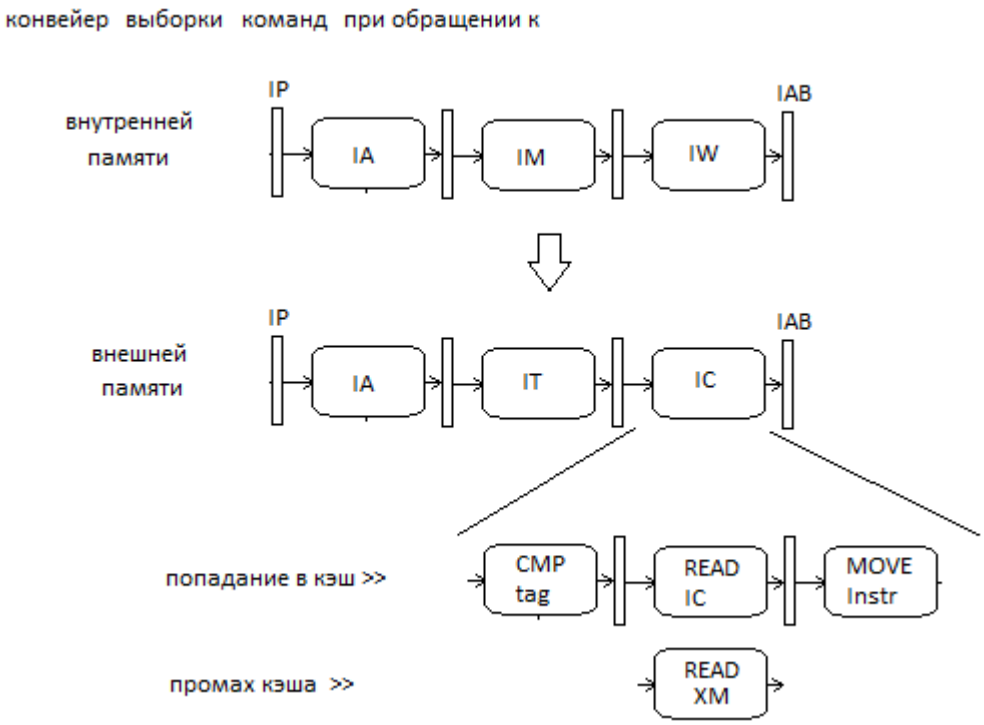


Рисунок 13 – Конвейер выборки команд из внешней памяти

В случае выборки команды из внешней памяти на стадии IT выполняется чтение линии памяти тэгов. Индекс линии определяется адресом чтения. Прочитанное значение линии тэгов передается в буфер. На стадии IC выполняются следующие действия:

1 Сравнение тэгов кэш-памяти с полем тега адреса команды;
2 *При совпадении* тегов – чтение линии команд из накопителя кэш-памяти или
При несовпадении тегов – обращение к внешней памяти.
Действия, выполняемые на данном этапе зависят от того кэшируемый адрес или некэшируемый.
3 Прочитанная линия команды передается в буфер IAB.

Каждое из перечисленных действий, в случае попадания в кэш, занимает один такт. Таким образом стадия IC будет всегда выполнятся за три такта при попадании в кэш. При промахе кэша количество тактов будет зависеть от соотношения частот процессорного ядра и внешней памяти, от типа памяти и ее характеристик быстродействия. Поскольку быстродействие конвейера определяется быстродействием его самой медленной стадии, то скорость выполнения линейного кода команды при попадании в кэш-команд будет в три раза медленнее, чем выполнение того же кода из внутренней памяти. Однако на самом деле при линейном выполнении кода выборка линий команд будет только в два раза медленнее. Это связано с тем, что устройство управления успевает отправить два запроса на чтение команд прежде чем поступит сигнал об отсутствии готовности команды.

На Рисунке 14 показана временная диаграмма чтения команд с учетом конвейера кэш-памяти команд. Предполагается, что каждый раз происходит попадание в кэш.

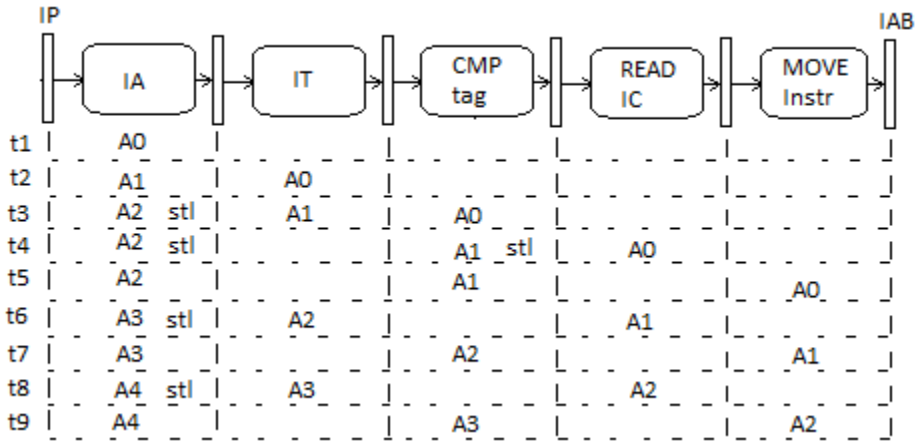


Рисунок 14 – Конвейер команд при попадании в кэш

Из Рисунка 14 видно, что в такте t3 устройство управления получает сигнал отсутствия готовности линии команд, и запрос A2 останавливает свое продвижение. Однако запросы A0 и A1 продолжают движение по конвейеру. В такте t4 видно, что в момент чтения линии команд по запросу A0, запрос A1 останавливается до завершения чтения запроса A0. Начиная с такта t5 возникает ситуация, которая будет повторяться. В данном такте линия команд A0 передается в IAB, и конвейер команд продвигается. В такте t6 запрос A3 останавливается (нет готовности линии команд A1), но запросы A1 и A2 могут продолжать движение. Таким образом, в буфер IAB линии команд будут поступать со скоростью равной половине частоты ядра, т.е. по сравнению с внутренней памятью наблюдается снижение скорости чтения команд в два раза. Дополнительный такт ожидания будет возникать при каждом изменении последовательной выборки команд. Также дополнительные потери производительности будут при выполнении команд переходов. Это связано с тем, что буфер предсказания переходов BTB не выполняет предсказаний переходов для адресов внешней памяти.

При чтении или записи данных, находящихся во внешней памяти, конвейер обработки данных будет иметь вид как на Рисунке 15. Действия, выполняемые при чтении данных, практически аналогичны действиям при чтении команд. В случае обращения во внешнюю память на стадии DT выполняется чтение линии памяти тэгов. Индекс линии определяется адресом чтения-записи. Прочитанное значение линии тэгов передается в буфер.

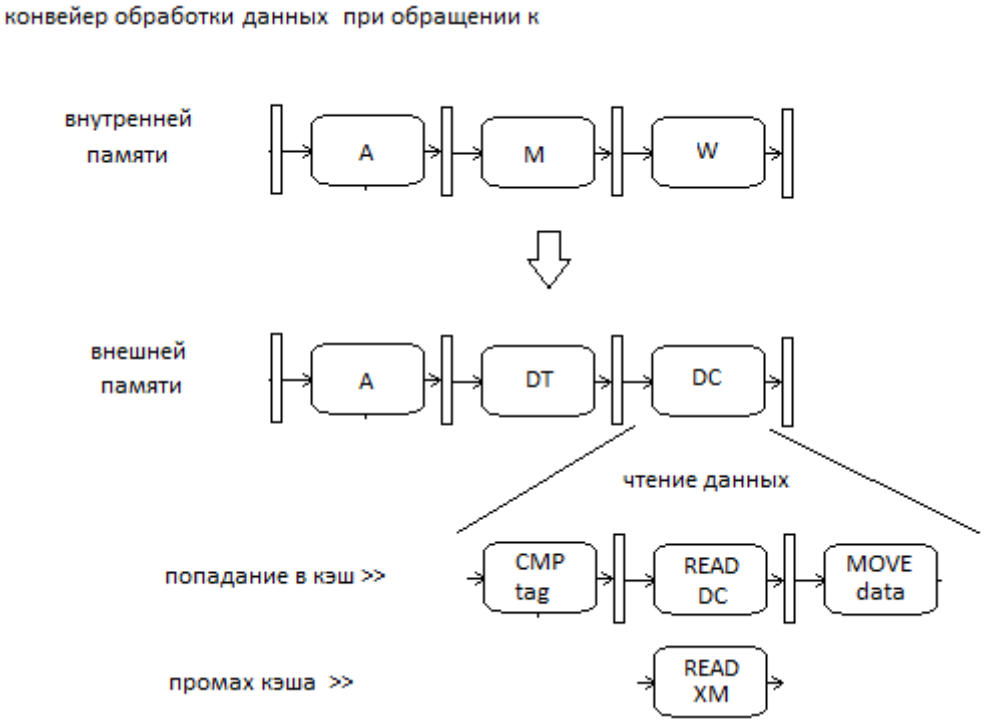


Рисунок 15 – Конвейер обработки данных

На стадии DC выполняются три действия:

1 Сравнение тэгов кэш-памяти данных с полем тега адреса данных:
   – Если выполняется *команда записи*, на данном этапе выполнение команды завершается. При попадании в кэш данные записываются в накопитель, а в случае промаха, данные записываются в выходной буфер SOC интерфейса.
   – Если выполняется *команда чтения* данных, необходимо выполнение следующих действий.
2 *При совпадении тегов* происходит чтение линии команд из накопителя кэш-памяти
   или
   При несовпадении тегов происходит обращение к внешней памяти. Действия, выполняемые на данном этапе зависят от того кэшируемый адрес или некэшируемый.
3 Прочитанные данные передаются в регистр-приемник команды загрузки.

Каждое из перечисленных трех действий, в случае попадания в кэш, занимает один такт. Для команды записи данная стадия всегда будет занимать один такт независимо от попадания в кэш. Для команды чтения данных стадия DC будет всегда выполнятся за три такта при попадании в кэш. При промахе кэша количество тактов будет зависеть от соотношения частот процессорного ядра и внешней памяти, от типа памяти и ее характеристик быстродействия.

Таким образом при работе с данными внешней памяти, в случае их размещении в кэш памяти, при чтении всегда будет два дополнительных такта останова конвейера на каждой операции загрузки. Исключением из этого правила будет ситуация, когда в одной линии команд есть два обращения к внешней памяти (от модулей JALU и KALU). Здесь ситуация похожа на кэш команд: два запроса чтения будут обрабатываться друг за другом, и вместо 4-х дополнительных тактов останова будет 3 такта.

Режим выполнения команд из внешней памяти, а также режим обработки данных внешней памяти процессором рассматриваются как неосновные. Поэтому

кэш команд и кэш данных не оптимально встроены в конвейер процессора. Для получения максимальной скорости при работе с кэш необходимо увеличение длины конвейера процессора. Однако это привело бы к снижению производительности программ при работе с внутренней памятью. Режим работы с внутренней памятью рассматривается как основной.

7.1 Кэш команд

Кэш команд (далее кэш) имеет размер 8 К 32-разрядных слов и организован в виде 128 линий, каждая из которых имеет 4 входа. Структурная схема кэша команд приведена на Рисунке 16. Ассоциативность кэша равна 4. Каждой линии входа соответствуют четыре квадрослова команд. Каждому квадрослову соответствует свой бит достоверности.

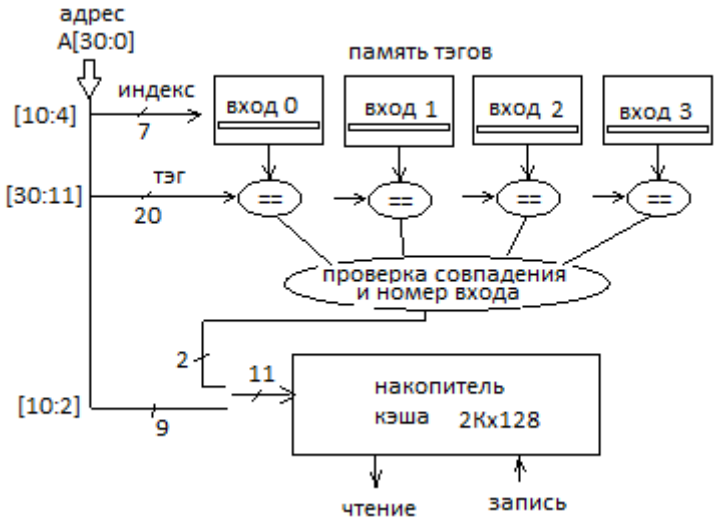


Рисунок 16 – Структура кэша команд

Таким образом 32-разрядный адрес команды при обращении к кэшу имеет следующий вид:

- Биты 1:0 не используются.
- Биты 3:2 указывают на номер квадрослова текущей линии, т.е. выбирают одно из 4-х квадрослов линии.
- Биты 10:4 образуют индекс линии памяти тэгов. Каждая линия имеет 4 входа, это значит, что имеется 4 банка памяти тэгов, и из каждого банка считывается тэг с указанным номером.
- Биты 30:11 образуют тэг, который сравнивается со значением поля тэга каждого банка (входа). По результатам сравнения принимается решение о промахе или попадании в кэш.
- Бит 31 не используется, т.к. всегда равен 0.

Структура одной записи тэговой памяти представляет собой 20-разрядное поле тэга и 4 бита достоверности для каждого из 4-х квадрослов (всего 24 бита).

После чтения тэговой памяти образуются 4 записи для сравнения. Если поле тэга текущего адреса равно одному из полей прочитанных тэгов, и установлен соответствующий запрашиваемому квадрослову бит достоверности, то принимается решение о попадании в кэш, и команда считывается из кэша. Если совпадения не было, принимается решение о промахе, и запрашиваемое квадрослово считывается из внешней памяти и загружается в кэш. Также в тэговую память записывается значение тэга, загруженного квадрослова, и устанавливается соответствующий бит

достоверности. Новое значение будет записано в один из 4-х входов (банков) тэговой памяти, в зависимости от выбранного алгоритма замещения тэгов в линии. В данном случае реализован самый простой вариант циклического замещения. Это значит, что кроме памяти тэгов имеется 128 счетчиков (каждый размером 2 бита), которые отслеживают номер банка для записи нового значения. После записи значение соответствующего счетчика увеличивается.

В кэше могут отображаться команды из трех внешних типов (банков) памяти. Каждый банк памяти условно разбивается на 32 страницы. Для динамической памяти размер страницы равен 512 К слов, для статической памяти – 128 К слов. Каждой странице может быть установлен атрибут кэшируемости, т.е. возможность загружать в кэш или нет.

Также для кэша предусмотрены два варианта загрузки в случае промаха:

- загрузка одного квадрослова;
- загрузка двух смежных квадрослов (отличающихся битом 2 адреса).

7.2 Кэш данных

Структура кэша данных (далее кэша) схожа со структурой кэша команд и имеет такой же размер накопителя – 8 К слов, 128 линий по 4 входа каждая. Но в отличие от кэша команд содержимое кэша данных может быть изменено. Для этого требуется введение дополнительных бит в поле тэга для отображения факта модификации. Для каждого квадрослова имеется специальный грязный бит (dirty bit) указывающий на то, что одно из слов или все слова квадрослова были изменены.

Кэш данных обычно поддерживает одну из двух стратегий записи:

- обратную запись (write-back) – данные в случае попадания в кэш записываются только в кэш;
- сквозную запись (write-throw) – данные пишутся и в кэш и во внешнюю память.

Для сквозной записи «грязные биты» всегда равны нулю. В случае обратной записи возникает проблема, когда линия входа с установленными «грязными битами» должна быть замещена новыми данными. В этом случае старые данные линии должны быть откопированы обратно во внешнюю память (copy-back). Поле «грязных битов» позволяет копировать только модифицированные квадрослова.

Обычно стратегия обратной записи дает более высокую скорость обработки. Но в ряде случаев важно иметь достоверную копию данных и во внешней памяти (например, для видеобуфера или для контроллера DMA). В этом случае предпочтительнее использовать стратегию сквозной записи. Для каждой страницы банка памяти может быть задан не только бит кэшируемости, но и бит сквозной записи. Таким образом, одни участки памяти могут поддерживать стратегию обратной записи, а другие – сквозной записи.

Кроме типов write-back и write-throw к кэшу данных применимы понятия:

- read-allocate – стратегия подразумевает действия кэша в случае промаха при чтении данных – в данном случае в кэш загружается линия данных;
- write-allocate – стратегия write-allocate подразумевает действия кэша в случае промаха при записи данных – в данном случае в кэш сначала загружается линия данных, а затем выполняется запись в линию.

В процессоре поддерживается только стратегия read-allocate. Стратегия write-allocate не поддерживается, и в случае промаха при записи, данные сразу же отправляются во внешнюю память (никаких загрузок в кэш не выполняется).

Аналогично кэшу команд для кэша данных может быть установлена стратегия подзагрузки сразу двух квадрослов в случае промаха.

Операции с кэш-памятью данных контролируются битами C (кэшируемые данные) и WT (сквозная запись). Биты определяются с помощью специальных регистров, описание которых приведено в подразделе 7.5 «Модуль управления защитой внутренней памяти и управления кэш-памятью». При обращении к кэшу возможно попадание в кэш (HIT) или промах (MISS). Попадание означает, что данные для заданного адреса находятся в кэше. Промах означает отсутствие данных с заданным адресом в кэше. Одна линия кэша одного из четырех входов кэш-памяти имеет атрибуты: 20 бит тэга, 4 бита достоверности (V бит), 4 бита модификации (Db бит). Каждому квадрослову соответствует свой бит достоверности и бит модификации. С кэшем данных кроме операций чтения и записи возможны операции очистки и сброса бит достоверности (см. подраздел 7.3 «Обеспечение когерентности кэшей и внешней памяти»).

Возможные ситуации работы с кэш-памятью приведены в Таблица 30, где также показано состояние бит до и после операции.

Таблица 30

Операция	Status	V	Db	Тэг	Данные	Действия*	V	Db	Тэг	Данные
Сброс процессора		x	x	x	x	F0	0	x	x	x
Чтение (C==1)	miss	0	x	x	x	F1	1	0	Ty	Ty
Чтение (C==1)	miss	1	0	Tx	Dx	F1	1	0	Ty	Dy
Чтение (C==1)	miss	1	1	Tx	Dx	F2 затем F1	1	0	Ty	Dy
Чтение	hit	1	x	Tx	Dx	F3	1	x	Tx	Dx
Запись	miss	x	x	x	x	Нет операций	x	x	x	x
Запись (WT==0)	hit	1	x	Tx	Dx	F4	1	1	Tx	Dy
Запись (WT==1)	hit	1	x	Tx	Dx	F5	1	0	Tx	Dy
Очистка	miss	x	x	x	x	Нет операций	x	x	x	x
Очистка	hit	1	0	x	x	Нет операций	1	0	x	x
Очистка	hit	1	1	Tx	Dx	F2	1	0	Tx	Dx
Сброс линии	miss	x	x	x	x	Нет операций	x	x	x	x
Сброс линии	hit	1	x	x	x	F6	0	x	x	x

Обозначения в таблице:

- x, Tx, Dx – данные в кэше до и после выполнения операции;
- Ty, Dy – данные в кэше после выполнения операции;
- F0– очистка всех бит достоверности;
- F1– загрузка блока (одного или двух квадрослов) данных в выбранную линию;

- F2– обратная запись (copy-back) активной линии в буфер записи. Из всех 4-х квадрослов во внешнюю память передаются только те из них, для которых бит модификации установлен в 1;
- F3– чтение данных активной линии в процессор;
- F4– запись данных в активную линию. Для соответствующего квадрослова устанавливается в 1 его бит модификации;
- F5– запись данных в активную линию. Для соответствующего квадрослова устанавливается в 0 его бит модификации. Записываемые данные также транслируются во внешнюю память;
- F6– очистка бита достоверности выбранной линии.

Как упоминалось в описании кэша команд, для каждой линии имеется свой 2-х битовый указатель на номер входа. После загрузки линии в текущий вход, значение указателя увеличивается на 1. Для случая чтения и состояния miss в левой части таблицы приводится информация о линии того входа, куда будет загружена новая информация.

7.3 Обеспечение когерентности кэшей и внешней памяти

При использовании кэш-памяти возникает проблема обеспечения когерентности данных в случае, когда к внешней памяти имеет доступ какое-то из дополнительных устройств кроме процессора. Таким устройством является контроллер прямого доступа к памяти (контроллер DMA). Такая же проблема может возникать в случае модификации программного кода.

В процессоре реализованы команды управления, позволяющие поддержать программную реализацию обеспечения когерентности данных.

В процессоре реализованы следующие команды управления для кэша данных:

- 0000 – очистка линии с адресом A;
- 0001 – очистка линии по номеру линии и номеру входа;
- 0010 – очистка и сброс бит достоверности линии с адресом A;
- 0011 – очистка и сброс бит достоверности линии по номеру линии и номеру входа;
- 0100 – нет операции;
- 0101 – сброс бит достоверности одного входа линии по номеру линии и номеру входа;
- 0110 – сброс бит достоверности всех входов линии по номеру линии;
- 0111 – сброс бит достоверности всего кэша данных.

Таким образом, если необходимо копировать значение из кэша данных по конкретному адресу A, следует очистить биты 3:0 адреса, в поле 3:0 адреса записать номер команды (код смотри выше), а также записать полученный адрес в специальный регистр блока управления кэш-памятью IDC_CR.

Если такой адрес присутствует в кэше данных и имеет установленные «грязные биты», произойдет обратное копирование данных во внешнюю память. Одновременно с копированием можно сбросить биты достоверности.

В процессоре реализованы следующие команды управления для кэша команд:

- 1010 – сброс бит достоверности линии с адресом A;
- 1101 – сброс бит достоверности одного входа линии по номеру линии и номеру входа;

- 1110 – сброс бит достоверности всех входов линии по номеру линии;
- 1111 – сброс бит достоверности всего кэша команд.

Таким образом, если необходимо удалить команды из кэша команд по конкретному адресу A, следует очистить биты 3:0 адреса, в поле 3:0 адреса записать номер команды (код смотри выше), а также записать полученный адрес в специальный регистр IDC_CR. Если такой адрес присутствует в кэше команд, произойдет сброс бит достоверности. Регистр IDC_CR доступен по записи и чтению. При записи информация, записываемая в регистр, содержит код выполняемой операции, а также может содержать передаваемый параметр (адрес, индекс). При чтении регистра IDC_CR значение имеет только бит 0 регистра. Все другие биты при чтении имеют значение ноль. Нулевой разряд регистра выполняет функцию флага занятости регистра IDC_CR. После записи данный флаг устанавливается в 1 и удерживается равным 1 до тех пор, пока команда не будет передана в кэш данных или кэш команд на исполнение. Поэтому, перед записью в регистр IDC_CR может потребоваться предварительное чтение и анализ флага занятости. Например, следующий код позволяет выполнить обратное копирование всех модифицированных линий кэша данных во внешнюю память:

```
        lc1 = 128*4;; // полное количество линий кэша данных
        j5 = 0x0001;; // команда обратного копирования и начальный индекс
rep_cln:
        IDC_CR = j5;; // обратное копирование если линия модифицирована
        j5 = j5+0x10;;  // переход к следующей линии
wait_cln_fin:
        j0 = IDC_CR;; // чтение флага готовности
        j0 = j0 and 1;; // проверка флага
        if njeq, jump wait_cln_fin (NP);;  // ожидание если флаг установлен
        if nlc1e, jump rep_cln;; // очистка следующей линии
```

Скорость выполнения данной процедуры будет зависеть в первую очередь от количества модифицированных линий в кэше данных. Если количество таких линий достаточно велико, поток данных из кэша во внешнюю память будет определяющим в скорости. И, например, если модифицированных линий нет вообще, время выполнения процедуры составит около 128*4*6 = 3072 такта.

С учетом аппаратных модификаций, возможна следующая процедура очистки:

```
        lc1 = 128*4;; // полное количество линий кэша данных
        j5 = 0x0001;;  // команда обратного копирования и начальный индекс
        rep_cln:    IDC_CR = j5;; // обратное копирование если линия модифицирована
        j5 = j5+0x10;;      // переход к следующей линии
        if nlc1e, jump rep_cln;; // очистка следующей линии
```

В этом случае минимальное время будет в два раза меньше. Данный алгоритм требует, чтобы между командами записи в IDC_CR было как минимум две линии команд. Однако, в случае, если количество модифицированных линий в кэше велико, оба алгоритма будут приблизительно равны по времени, т.к. в данном случае определяющим фактором будет скорость обмена с внешней памятью.

7.4 Эффективность кэша команд и кэша данных

При использовании кэша команд команды поступают в буфер со скоростью половины частоты процессора. Одновременно могут считываться четыре команды, что обеспечивает достаточный темп в случае, когда за один такт выполняется меньше 4-х команд одной линии. Если в каждом такте выполняются все четыре команды одной линии, и нет задержек на конвейере команд, скорость составит только половину от максимальной. В случае ветвлений кэш команд будет менее эффективен, чем внутренняя память. При ветвлении из-за большей длины конвейера (5 вместо 3) формируется дополнительная задержка на загрузку первой команды перехода. Если переходы очень часты или тело циклов очень короткое, могут возникнуть дополнительные потери. В случае работы с внешней памятью не используется буфер предсказания переходов, что снижает быстродействие.

Для кэша данных запись данных при попадании в кэш или при промахе не вносит дополнительных задержек. При чтении данных из кэша будет дополнительная задержка. В случае чтения из внутренней памяти (стадии конвейера A, M, W) данные готовы на стадии W и могут поступать в регистр. В случае работы с внешней памятью и в случае попадания в кэш, на стадии W обнаруживается попадание в кэш, и формируется запрос на чтение кэша. В следующем такте происходит чтение данных из кэша в буфер (конвейер данных процессора остановлен). В следующем такте данные из буфера записываются в регистр приемника (конвейер остановлен). После записи конвейер может продолжить движение. Таким образом, при чтении данных образуются два дополнительных такта остановки конвейера для случая, когда нет зависимости по данным, т.е. считываемые данные не требуются в следующем такте для обработки. Если такая зависимость есть, образуется три дополнительных такта по сравнению с внутренней памятью.

Ассоциативность кэшей равна 4. Это означает, что данные могут храниться по четырем адресам с различными тэгами, но одинаковыми индексами. Например, можно максимально эффективно одновременно работать с четырьмя буферами данных во внешней памяти, каждый объемом 2К слов, размещенных по адресам 0x4000_0000, 0x4000_0800, 0x4000_1000, 0x4000_1800. Однако, если потребуется пятый буфер одновременно с четырьмя предыдущими, эффективность использования кэша будет очень низкой, т.к. буферы будут по очереди вытеснять друг друга из кэша.

Кэш эффективен в случае, когда необходимо повторное использование данных. Также он может быть полезен при однократном использовании данных. Например, при чтении слова загружается квадрослово. Это значит, что оставшиеся три слова могут быть считаны из кэша. При установке бита подзагрузки сразу двух квадрослов, эффективность увеличивается.

7.5 Модуль управления защитой внутренней памяти и управления кэш-памятью

В процессор добавлен модуль, который содержит набор регистров для управления кэш-памятью, а также управления защитой внутренней памяти. Данный модуль использует регистры групп 0x1E и 0x1F. Набор регистров приведен в Таблице 31. Доступ к регистрам устройства защиты памяти выполняется с помощью обычных команд пересылки IALU.

Внимание! Доступ по записи возможен только в режиме супервизора и при установленном в 1 бите EXT_MODE регистра SQSTAT.

Таблица 31 – Регистры модуля защиты памяти

Номер	Название	Назначение
Группа 0x1E		
0x03C0	MS0_C	Регистр управления кэшированием данных MS0
0x03C1	MS0_WT	Регистр управления сквозной записью MS0
0x03C2	MS0_CI	Регистр управления кэшированием команд MS0
0x03C3 – 0x03C7	-	
0x03C8	MS1_C	Регистр управления кэшированием данных MS1
0x03C9	MS1_WT	Регистр управления сквозной записью MS1
0x03CA	MS1_CI	Регистр управления кэшированием команд MS1
0x03CB – 0x03CF	-	
0x03D0	SDR_C	Регистр управления кэшированием данных SDRAM
0x03D1	SDR_WT	Регистр управления сквозной записью SDRAM
0x03D2	SDR_CI	Регистр управления кэшированием команд SDRAM
0x03D3 – 0x03DF	-	
Группа 0x1F		
0x03E0	PU0	Регистр конфигурации защиты модуля памяти 0
0x03E1	PU1	Регистр конфигурации защиты модуля памяти 1
0x 03E2	PU2	Регистр конфигурации защиты модуля памяти 2
0x03E3	PU3	Регистр конфигурации защиты модуля памяти 3
0x03E4	PU4	Регистр конфигурации защиты модуля памяти 4
0x03E5	PU5	Регистр конфигурации защиты модуля памяти 5
0x03E6	PU6	Регистр конфигурации защиты модуля памяти 6
0x03E7	PU7	Регистр конфигурации защиты модуля памяти 7
0x03E8	PU_SR	Регистр состояния
0x03E9 – 0x03FB	-	
0x03FC	PU_CR	Регистр управления
0x03FD	IDC_CR	Регистр управления операциями с кэш памятью
0x03FE – 0x03FF	-	

7.5.1 Регистр защиты памяти PU

Регистр управляет защитой одного модуля внутренней памяти. Размер модуля и параметры защиты задаются программно. При программировании регистра размер защищаемого модуля памяти задается количеством страниц размером 1 К слов. Назначение разрядов регистра приведено в Таблице 32.

Таблица 32 – Регистр защиты PU

Номер бита	Название	Назначение
10-0	STA	Начальный адрес модуля. Соответствует физическому адресу памяти образуемому как {11'b0,START_A[10:0],10'b0}.
11	-	
22-12	ENDA	Конечный адрес модуля. Соответствует физическому адресу памяти образуемому как {11'b0,END_A[10:0],10'b11_1111_1111}
23	-	

Номер бита	Название	Назначение
25-24	JK_AP	Режим доступа к модулю со стороны шин J и K. 00 – полный доступ 01 – супервизор чтение и запись, пользователь чтение 10 – всем только чтение 11 – супервизор только чтение
27-26	I_AP	Режим доступа к модулю со стороны шины I. 00 – полный доступ 01 – только супервизор 10 – доступ запрещен 11 – доступ запрещен
29-28	H_AP	Режим доступа к модулю со стороны шины S (хост, DMA). 00 – чтение и запись 01 – только чтение 10 – доступ запрещен 11 – доступ запрещен
31-30	-	

Поля STA и ENDA используются для сравнения со значениями адресных шин J, K, I, S. В сравнении участвуют только биты 20:10 указанных шин. Проверка выполняется только при доступе к внутренней памяти. Так для K-шины попадание в некоторый модуль означает истинность следующего выражения:

Hit_PUx = (K[20:10]>=STAx) && (K[20:10]<=ENDAx).

Минимальный размер модуля равен странице из 1 К слов. Модуль может рассматриваться как множество из 1 К страниц. При нарушении прав доступа к модулю памяти вырабатывается исключительная ситуация. Исключение составляет контроль доступа со стороны шины S. Нарушение прав доступа будет вызывать блокировку записи и установку флага ошибки. Исключительная ситуация вырабатываться не будет. Исключительная ситуация всегда форсируется после выполнения команды. Поэтому в случае срабатывания защиты по доступу со стороны шины I, процессор выполнит одну линию команд из защищенной области, прежде чем перейдет на обработку исключительной ситуации. Определить событие, вызвавшее срабатывание исключительной ситуации, можно анализируя регистр состояния устройства управления, а также регистр состояния модуля защиты.

7.5.2 Регистр состояния PU_SR

Регистр состояния хранит информацию о событии, которое вызвало срабатывание модуля защиты. После сброса значение регистра равно нулю. Назначение разрядов регистра приведено в таблице 33.

Таблица 33 – Регистр управления PU_SR

Номер	Название	Назначение
0	JPF	Срабатывание защиты при доступе со стороны шины J
1	KPF	Срабатывание защиты при доступе со стороны шины K
2	IPF	Срабатывание защиты при доступе со стороны шины I
3	HPF	Срабатывание защиты при доступе со стороны шины S (хост, DMA)
4-31	-	Всегда равны нулю

Если какой-то флаг в регистре состояния установлен, значит при доступе со стороны соответствующей шины произошло срабатывание модуля защиты, что

вызвало генерацию исключительной ситуации. При возникновении исключительной ситуации из регистра состояния SQSTAT можно узнать, что ситуация была вызвана срабатыванием защиты по правам доступа. При чтении регистра PU_SR можно определить, была ли данная ситуация вызвана срабатыванием модуля защиты. Если регистр PU_SR равен нулю, значит сработала защита в модулях J или K целочисленного АЛУ. Регистр PU_SR автоматически сбрасывается после выполнения чтения.

7.5.3 Регистр управления PU_CR

Регистр управляет включением различных функций процессора. Назначение разрядов регистра приведено в таблице 34.

Таблица 34 – Регистр управления PU_CR

Номер	Название	Назначение
0	DC_ON	1 – включение кэш памяти данных
1	IC_ON	1 – включение кэш памяти команд
2	EN_2dQW	1 – грузить 2 квадрослова данных при промахе 0 – загрузка одного квадрослова
3	EN_2iQW	1 – грузить 2 квадрослова команд при промахе 0 – загрузка одного квадрослова
4	SOC_speed_up	0 – разрешает выполнение команды на стадии W одновременно с загрузкой данных с СОК шины. 1 – запрещает выполнение команды на стадии W одновременно с загрузкой данных с СОК шины. Сначала загружаются данные, затем начинает движение конвейер
5-7	-	Зарезервированы
8	PSD_fun	Управление генерацией исключительной ситуации при загрузке регистров устройства управления и модуля отладки из внешней памяти. 0 – разрешена генерация. 1 – исключительная ситуация только при нарушении прав доступа
9	SP_byte	Управление автоматическим сдвигом указателя стека и фрэйма при работе с байтами и короткими. 0 – функция выключена, 1 – функция включена
10	MIN_MAX	Включение функции поиска номера максимума-минимума 0 – выключено 1 – включено
11	LC_sup	Выключение анализа выхода из цикла 0 – анализ работает 1 – анализ выключен 0 – разрешение использования процедуры анализа завершения цикла. Позволяет сократить время выхода из цикла. 1 – при реализации циклов с помощью LC0, LC1 будут потери в скорости при выполнении последнего предсказания.
12-15		Зарезервированы
16	PU0_EN	Разрешение работы модуля защиты 0. 1 – разрешено 0 – запрещено
17	PU1_EN	Разрешение работы модуля защиты 1. 1 – разрешено 0 – запрещено

Номер	Название	Назначение
18	PU2_EN	Разрешение работы модуля защиты 2. 1 – разрешено 0 – запрещено
19	PU3_EN	Разрешение работы модуля защиты 3. 1 – разрешено 0 – запрещено
20	PU4_EN	Разрешение работы модуля защиты 4. 1 – разрешено 0 – запрещено
21	PU5_EN	Разрешение работы модуля защиты 5. 1 – разрешено 0 – запрещено
22	PU6_EN	Разрешение работы модуля защиты 6. 1 – разрешено 0 – запрещено
23	PU7_EN	Разрешение работы модуля защиты 7. 1 – разрешено 0 – запрещено
24-29	-	Зарезервированы
31-30	USOC_EN	Режим доступа к периферийным устройствам SOC-шины (адреса 0x8000000 и выше) в режиме пользователя 0 0 – запрещен любой доступ 0 1 – запрещена запись, но разрешено чтение 1 x – разрешен любой доступ По сбросу любой доступ запрещен

Бит SOC_speed_up позволяет ускорить процесс загрузки данных с СОК-шины или внешней памяти. Команда чтения регистра СОК-шины тратит ориентировочно 10 тактов ядра на загрузку данных. В это время процессор простаивает. Бит SOC_speed_up позволяет управлять операциями в 10-м такте загрузки, т.е. в момент, когда данные грузятся непосредственно в регистр процессора:

- если бит равен 0, возможно выполнение операций процессором одновременно с загрузкой, если нет конфликта по записи;
- если бит равен 1, процессор продолжит работу только после записи.

В результате процессор может простаивать девять тактов вместо десяти. При работе с кэшем данных использование данного бита позволяет сократить число тактов останова конвейера процессора при чтении данных из кэша: вместо двух дополнительных тактов ожидания, будет только один такт. Загрузка данных из кэша в регистр-приемник будет происходить одновременно с выполнением команд следующей линии.

В процессоре доступ к устройствам SOC-шины возможен также с помощью обычных команд загрузки и сохранения регистров по определенному адресу. Адресное пространство периферийных устройств от 0x8000_0000 до 0xFFFF_FFFF. Доступ ко всем периферийным устройствам возможен также в режиме пользователя, т.к. устройство защиты памяти защищает только внутреннюю оперативную память. Наибольшую опасность в данном случае представляет некорректное поведение пользовательской программы, способной привести к потере работоспособности всей системы.

В связи с этим, в модуле защиты в регистре управления выделены специальные биты (USOC_EN ==PU_CR[31:30]), которые запрещают доступ по записи и по чтению к адресному пространству периферийных устройств в режиме

пользователя, независимо от того, разрешены программные исключения или нет. Коды: 00 – запрет доступа, 01 – чтение разрешено, 1x – полный доступ.

После сброса значение регистра PU_CR равно нулю.

7.5.4 Регистры управления кэшированием данных MS0_C, MS1_C, SDR_C

Регистры используются для задания атрибута кэшируемости (C бит) для отдельных страниц банков внешней памяти. Процессор может адресовать три банка внешней памяти: MS0, MS1, SDRAM (четыре банка). При этом модули MS0 и MS1 имеют деление на страницы по 128 К слов, а модуль SDRAM (банк 0) имеет деление на страницы по 512 К слов. Каждый бит регистра соответствует странице памяти. Так 0-й бит соответствует нулевой странице, 1-й бит – первой и т.д. Итого можно определить до 32*128 = 4 М слов памяти модулей MS0, MS1 и до 16 М слов динамической памяти. Для динамической памяти регистр управляет только банком 0. Другие три области динамической памяти всегда считаются кэшируемыми и используют стратегию обратной записи. Таким образом для памяти MS0, MS1 индекс страницы берется из бит 21:17 адреса слова памяти. Для динамической памяти индекс страницы – из бит 23:19 адреса слова. Если адрес динамической памяти превышает размер 15 М слов, это всегда соответствует кэшируемому адресу.

7.5.5 Регистр управления сквозной записью MS0_WT, MS1_WT, SDR_WT

Аналогично управлению кэшируемостью страниц существует возможность для каждой страницы указать бит стратегии записи (WT бит) при попадании в кэш:

0 – запись только в кэш;

1 – запись в кэш и во внешнюю память.

7.5.6 Регистр управления кэшированием команд MS0_CI, MS1_CI, SDR_CI

Аналогично управлению кэшируемостью страниц для данных существует возможность для каждой страницы указать бит кэшируемости при обращении к командам:

0 – не кэшируется;

1 – кэшируется.

7.6 Кэш память и операции с байтами и короткими словами во внешней памяти

Процессор поддерживает на аппаратном уровне работу с байтами и короткими словами во внешней памяти. Как и в случае с внутренней памятью, адрес байта или короткого слова перед обращением к памяти предварительно преобразуется в адрес слова, и затем выполняется операция. Следовательно, кэш данных всегда будет хранить в тэге адрес слова данных.

Рассмотрим случай, когда в регистре j2 хранится значение 0x4000_4000. В случае байтового доступа с использованием регистра j2 кэш данных и внешняя память будут работать с адресом 0x1000_1000. При обращении к короткому слову рабочий адрес будет равен 0x2000_2000. Для кэша данных и для внешней памяти это будут три разных адреса. Для соответствия между адресами слов, байт и коротких слов необходимо строго придерживаться правил формирования адресов и правил преобразования адресов при операциях с указателями. Если какой-то указатель на слово имеет значение A, то соответствующий данному слову указатель

на короткое слово должен иметь значение A*2, а соответствующий указатель на байт значение A*4. Для области MS0, MS1 это означает, что адреса коротких слов будут начинаться с 0x6vvv_vvvv и 0x7vvv_vvvv. Для адресов динамической памяти возникает сложность, т.к. для адреса слова 0x4vvv_vvvv адрес байта выходит за границы 32-разрядного формата. Поэтому динамическая память имеет дополнительное адресное пространство 0x1vvv_vvvv, которое соответствует стандартному адресному пространству. При его использовании (в регистре SYSCON должен быть установлен соответствующий бит) адреса коротких слов будут равны 0x2vvv_vvvv, а адреса байт 0x4vvv_vvvv. При операциях с указателями разных типов должен выполняться соответствующий сдвиг. При обращении к словам во внешней памяти адрес 0x4000_0000 соответствует той же ячейке памяти, что и адрес 0x1000_0000. Однако в кэше данных эти адреса будут разными.

7.7 Алгоритм очистки кэша команд

Для обеспечения когерентности кэша команд возможно использование специальных команд очистки линий кэша команд по значению адреса или по значению индекса. При этом необходимо соблюдать ряд требований к потоку команд на конвейере процессора.

Далее приведен пример кода для удаления блока инструкций из кэша команд. В таких случаях обычно задается стартовый адрес блока и конечный адрес блока. В начале текущий адрес блока равен стартовому. После очистки к текущему адресу прибавляется значение 16 и проверяется выход за пределы блока. Если завершения блока нет, процесс повторяется. Код программы следующий:

#1: IDC_CR = текущий_адрес_очистки; увеличение адреса очистки на 16;

#2: проверка конца блока (в JALU);;

#3: команда перехода на команду #1 если конца нет;; (NP)

Таблица 35 – Команда очистки кэша с использованием адреса

Такт	IA	IM	IW	P	D	X	A	M	W
1					#1				
2					*#2*	#1			
3					*#3*	*#2*	#1		
4	Очистка конвейера команд					*#3*	*#2*	#1	
5	#1							*#2*	#1
6	#x1	#1							*#2*
7		#x1	#1						
8			#x1	#1					
9				#x1	#1				
10					*#2*	#1			
11					*#3*	*#2*	#1		
12	Очистка конвейера команд					*#3*	*#2*	#1	
13	#1							*#2*	#1
14	#x1								*#2*

Программа очистки кэша исполняется из внутренней памяти и не использует кэш команд. Из временной диаграммы Таблицы 35 видно, что команда #1 на стадии W запускает дополнительный процесс #x1, который требует четыре последующих такта:

Такт 6 – передача адреса в кэш команд;

Такт 7 – чтение линии памяти тэгов;

Такт 8 – проверка совпадения тэгов с адресом;

Такт 9 – в случае совпадения, запрос к памяти тегов на очистку бита достоверности.

Имеются ограничения на параллельные действия со стороны процессора. В момент, когда специальная команда #x1 читает память тегов (такт 7) на конвейере процессора не должен происходить переход. Иначе переход сбрасывает содержимое конвейера выборки команд, а заодно и команду #x1 на стадии IM.

Если в момент запроса на очистку бита достоверности (такт 9) имеется запрос к памяти тегов со стороны следующей специальной команды, то следующая специальная команда будет приостановлена. Этот факт нужно учитывать при записи значения в регистр IDC_CR – слишком плотный поток команд может вызвать пропуск отдельных команд. По сути специальная команда требует для своего исполнения один в случае промаха или два такта в случае попадания. В связи с этим для команды перехода цикла рекомендуется использовать опцию (NP).

8 Встроенный интерфейс (SOC-интерфейс)

При работе с внутренней памятью ядро процессора использует три внутренние шины I, J, K. Однако для работы с периферийными устройствами процессор использует специальный встроенный интерфейс, который осуществляет передачу запроса от внутренних шин ядра на специальную шину периферийных устройств (SOC – шина). Данная шина соединяет периферийные модули (внешний порт, DMA, порты связи, порт JTAG, контроллер прерываний и др.) с системой памяти через SOC-интерфейс и шину S типа. SOC-шина имеет шину данных шириной 128 бит и адресную шину шириной 32 бита. Она работает с частотой равной половине частоты ядра. Все данные, передаваемые между внутренней памятью или ядром и периферийными модулями, проходят через SOC-интерфейс и SOC-шину.

SOC-интерфейс представляет собой модуль процессора, выполняющий связующую роль между ядром и внутренними периферийными устройствами. Со стороны ядра к интерфейсу подключены четыре шины J-, K-, I- и S-типов, а с другой стороны одна SOC-шина периферийных устройств. SOC-шина имеет пропускную способность 128 бит, и ее тактовый генератор (SOCCLK) работает на частоте, равной половине частоты ядра (CCLK). Периферийными устройствами, которые могут быть ведущими на SOC-шине являются (Рисунок 17):

- SOC-интерфейс;
- контроллер DMA.

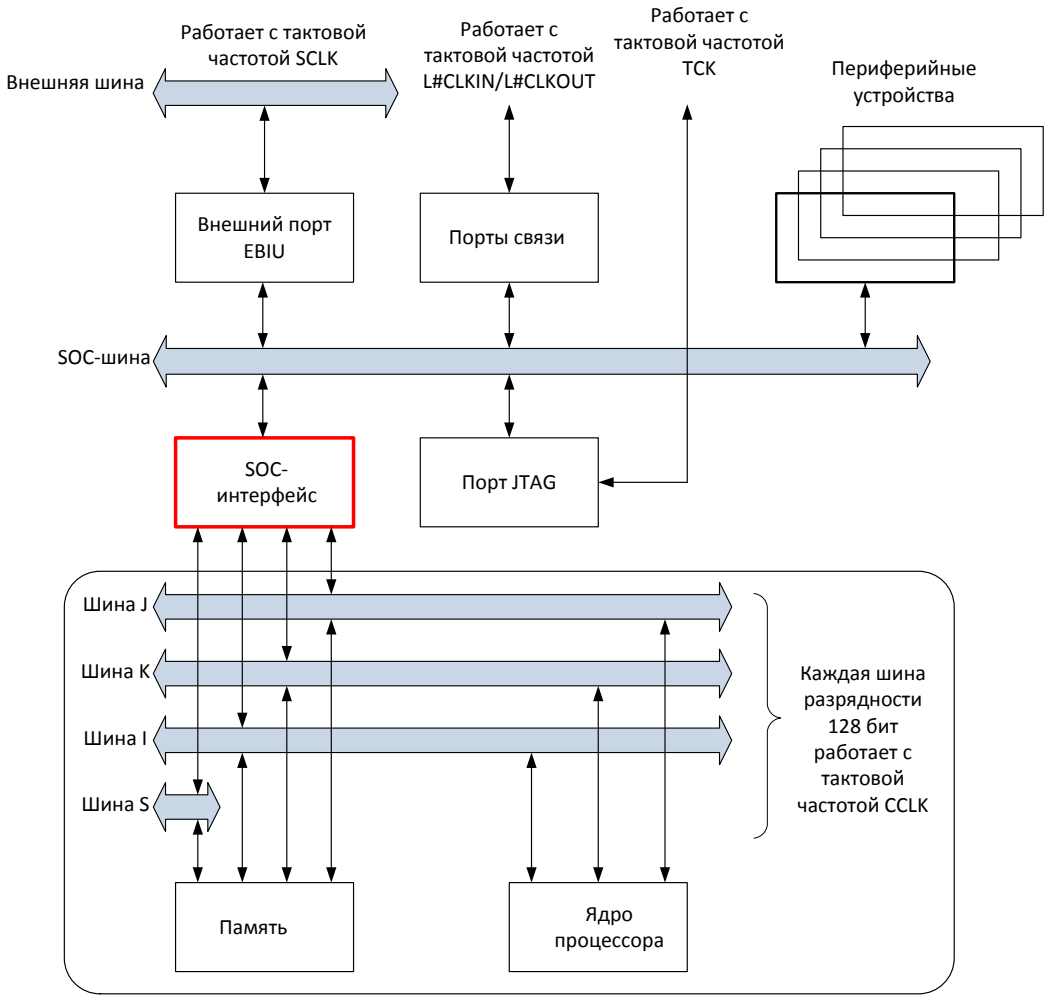


Рисунок 17 **–** Подключение периферийных модулей

Все внутренние периферийные устройства подключены к SOC-шине и являются ведомыми устройствами.

Структура SOC-интерфейса показана на Рисунок 18, а архитектура шин, соединяющих различные модули SOC-шины, приведена на Рисунок 19.

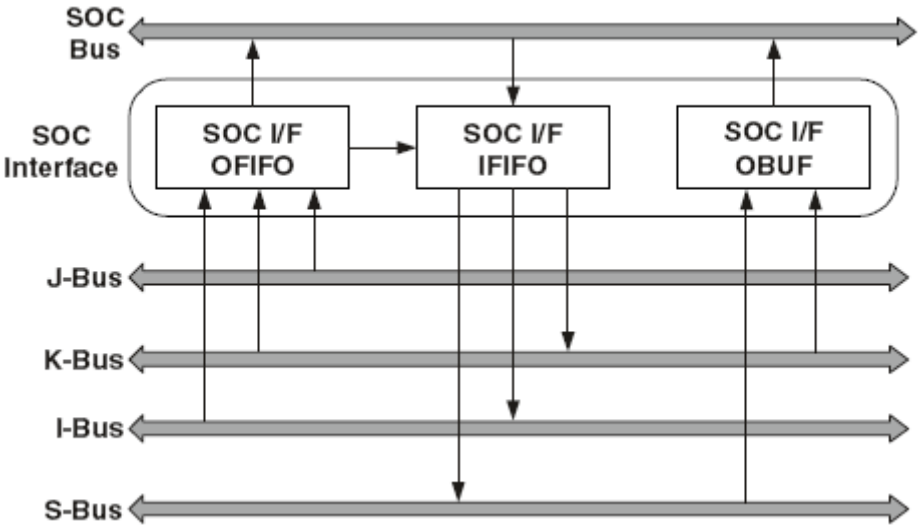


Рисунок 18 – Поток данных через SOC-интерфейс

В связи с тем, что на SOC-шине может быть несколько ведущих устройств, при каждом доступе к шине выполняется арбитраж. При этом доступ к периферийным устройствам возможен со стороны процессора и со стороны контроллера DMA. Со стороны процессора представителем является SOC-интерфейс и при этом он имеет два источника запросов: выходное FIFO интерфейса (OFIFO) и выходной буфер (OBUF). Приоритеты доступа при арбитраже фиксированы. Наивысший приоритет имеет выходной буфер интерфейса OBUF, который может возвращать прочитанное значение в контроллер DMA или в порт связи.

SOC-интерфейс состоит из трех устройств FIFO, через которые проходят все потоки данных. Если процессор обращается во внешнюю память, запрос из SOC-интерфейса поступает сразу во внешний интерфейс. Если шины процессорного ядра посылают в SOC-интерфейс запрос на чтение, прочитанные данные должны вернуться обратно в интерфейс. Для возвращаемых данных используется входное IFIFO. Также в данный буфер помещаются запросы всех внешних ведущих устройств, которые обращаются к внутренней памяти ядра. Если модуль IFIFO получил прочитанные данные, он пересылает их во внутренний регистр ядра. Если же был получен запрос на запись со стороны ведущего устройства, записываемые данные пересылаются по адресу-приемнику (используется шина S). В случае, если запрос на чтение получен от внешнего ведущего устройства, выполняется чтение внутренней памяти (используется шина S) и прочитанные данные помещаются в выходной буфер OBUF. Таким образом, в OBUF попадают только считываемые внешними ведущими устройствами данные ядра.

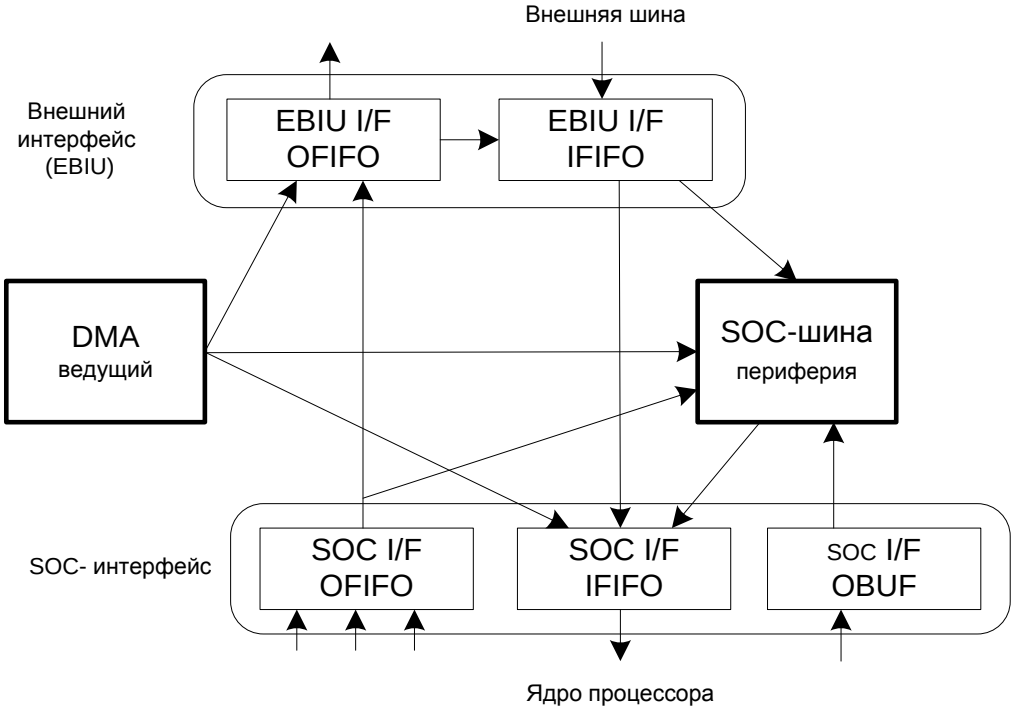


Рисунок 19 – Структура соединений модулей SOC-шины

8.1 Транзакции SOC OFIFO

Все запросы ядра по шинам J, K, I, которые не попадают во внутреннюю память, помещаются в выходное OFIFO. Это могут быть следующие запросы:

- Запись из ядра процессора по шинам J и K в устройства SOC-шины или внешнее адресное пространство;
- Запросы чтения по шинам J и K из устройств SOC-шины или внешнего адресного пространства;
- Запрос чтения команды по шине I из внешней памяти или из регистра инструкций эмулятора (EMUIR).

Все запросы ядра поступают в OFIFO с частотой ядра (CCLK). Считывание запросов из буфера выполняется с частотой SOC-шины (SOCCLK). На выходе буфера осуществляется проверка, к какому ресурсу выполняется запрос (внешняя память или внутренняя периферия), и далее запрос пересылается в пункт назначения. Если выполняется запрос к периферии, он имеет наименьший приоритет в арбитраже доступа. Если выполняется запрос к внешней памяти, также происходит арбитраж доступа к выходному FIFO внешнего интерфейса. За доступ к внешнему ресурсу OFIFO соревнуется с контроллером DMA (Рисунок 19). SOC-интерфейс в этом арбитраже всегда имеет наивысший приоритет.

С точки зрения конвейера процессора команда завершается, когда запрос на запись или на чтение помещается в SOC OFIFO. Если это операция записи, она считается выполненной и конвейер ядра продолжает движение, а если операция чтения, конвейер останавливается до момента возвращения данных. При этом неважно имеется зависимость по данным или нет. При чтении команд из внешней памяти после помещения двух запросов в OFIFO конвейер команд останавливается до момента получения прочитанной команды. При этом конвейер обработки команд может продолжать выбирать команды из буфера выравнивания и отправлять их на исполнение.

Буфер OFIFO имеет размер 8 записей. Это означает, что он может принять до 8-ми запросов без остановки конвейера процессора. При этом имеется следующие особенности:

- Если по шинам J и K одновременно выполняются две записи, процессору потребуется один такт, чтобы поместить их в OFIFO. При этом одна транзакция (J) помещается сразу в OFIFO, а вторая (K) во временный буфер. Из временного буфера запрос пересылается в OFIFO в следующем такте.
- Если процессору не требуется доступ к OFIFO в следующем такте, задержек конвейера не произойдет.

Запросы в OFIFO помещаются с частотой ядра, а считываются с частотой SOC-шины, т.е. в два раза меньшей. Поэтому в случае интенсивного потока записи во внешнюю память возможно заполнение буфера и это вызовет приостановку конвейера процессора.

8.2 Транзакции SOC IFIFO

Входной буфер IFIFO SOC-интерфейса обрабатывает два типа транзакций:

- Запросы чтения ядра процессора. SOC IFIFO принимает прочитанные данные, полученные в результате транзакций чтения, которые были переданы через OFIFO;
- Запросы чтения или записи внутренней памяти ядра от DMA.

В первом случае возвращаемые данные записываются в регистр ядра, используя шину K. Во втором случае IFIFO использует шину S для доступа к внутренней памяти. В случае операции чтения внутреннего ресурса, прочитанные данные помещаются в выходной буфер OBUF.

Имеется три источника запросов к IFIFO (Рисунок 19):

- DMA;
- внешний интерфейс;
- SOC-шина.

Наивысший приоритет имеет SOC-шина, т.к. ее запрос означает чтение регистра периферии по запросу ядра, далее идет запрос внешнего интерфейса и наименьший приоритет у контроллера DMA. Запрос от внешнего интерфейса означает возвращение данных в ответ на запрос чтения внешней памяти ядром. Приоритеты доступа расставлены так, чтобы запросы ядра получали наивысший приоритет. В случае чтения это минимизирует простои процессора.

8.3 Транзакции SOC OBUF

Выходной буфер OBUF SOC-интерфейса задействуется только в операциях чтения внутренних ресурсов (памяти) контроллером DMA. Данные из OBUF могут пересылаться в DMA или порты связи.

8.4 Программирование SOC интерфейса

Чтение процессором внешней памяти вызывает передачу запроса от ядра через несколько доменов синхронизации. Все это может приводить к длительным остановам процессора. Для ускорения работы процессора с внешней памятью

используется кэш-память данных и кэш-память команд. Процессор может организовывать работу с внешней памятью, как с использованием кэш-памяти, так и без.

Архитектурно SOC-шина представляет собой простой интерфейс к регистрам периферийных устройств, в котором циклы чтения или записи выполняются в течение одного такта. Если какое-либо из устройств не готово выдать (принять) данные в течение одного такта, оно может выставлять сигнал ожидания завершения операции.

9 Таймеры общего назначения

Процессор имеет два 64-битных таймера общего назначения: таймер 0 и таймер 1. Доступ к таймерам возможен как по номеру регистра (с использованием команд пересылки), так и при помощи адреса в адресном пространстве периферийных устройств (с использованием команд загрузки и сохранения). Таймеры принадлежат группе регистров контроллера прерываний (группа 0x3A). Номера регистров и адреса в адресном пространстве периферийных устройств приведены ниже (Таблица 36).

Таблица 36 – Регистры модуля таймеров

Имя	Описание	Номер/адрес	Значение по умолчанию
INTCTL	Управление прерыванием	0x074E / 0x8000_034E	0
TIMER0L	Текущее значение таймера 0, младшие разряды	0x 0750 / 0x8000_0350	Не определено
TIMER0H	Текущее значение таймера 0, старшие разряды	0x 0751/ 0x8000_0351	Не определено
TIMER1L	Текущее значение таймера 1, младшие разряды	0x 0752/ 0x8000_0352	Не определено
TIMER1H	Текущее значение таймера 1, старшие разряды	0x 0753 / 0x8000_0353	Не определено
TMRIN0L	Значение инициализации таймера 0, младшие разряды	0x 0754/ 0x8000_0354	Не определено
TMRIN0H	Значение инициализации таймера 0, старшие разряды	0x 0755/ 0x8000_0355	Не определено
TMRIN1L	Значение инициализации 1, младшие разряды	0x 0756 / 0x8000_0356	Не определено
TMRIN1H	Значение инициализации 1, старшие разряды	0x 0757 / 0x8000_0357	Не определено

Таймеры – это автономно работающие счетчики, имеющие следующие функции:

- загрузка начальным значением;
- декремент значения до нуля;
- индикация завершения счета;
- автоматическая перезагрузка;
- повторение цикла.

Индикатор завершения счета (такт после вычитания счетчика, когда он был равен 1) является прерыванием. Каждый таймер содержит два 64-битных регистра: начальное значение (TMRINxH/L) счета и текущее значение (TIMERxH/L) счетчика. Регистр TMRINxH/L доступен для чтения и записи, тогда как TIMERxH/L предназначен только для чтения. Оба регистра имеют размер 64 бита, но каждый доступен только как две 32-разрядные половины: старшая и младшая. Доступные 32-битные регистры:

- TMRIN0H;
- TMRIN0L;
- TMRIN1H;
- TMRIN1L;

- TIMER0H;
- TIMER0L;
- TIMER1H;
- TIMER1L.

При этом нет буферизации при считывании или записи различных частей регистра. Регистр TIMERx выполняет функцию счетчика и обновляется каждый такт частоты SOCCLK. Если счетчик в текущем такте имеет значение 1, в следующем такте в него загружается начальное значение из регистра TMRINx.

Каждый из таймеров имеет бит разрешения счета, который хранится в регистре управления INTCTL. Установка бита TMRxRN в 1 разрешает счет, а сброс бита в 0 останавливает счет.

9.1 Регистр управления прерываниями (INTCTL)

Регистр INTCTL – 32-битный регистр, который управляет чувствительностью внешних входов прерываний IRQ3-0 (прерываний чувствительных к фронту или уровню) и обеспечивает управление стартом и остановкой таймеров. Подробное описание разрядов регистра приведено ниже (Таблица 37). Следует обратить внимание на то, что регистр INTCTL может быть модифицирован программой начальной загрузки в зависимости от того, какой режим загрузки был выбран.

Таблица 37 – Регистр INTCTL

Бит	Имя	Назначение
0	IRQ0_EDGE	Запрос прерывания по входу IRQ0 вызывается: 0 – фронтом сигнала (из высокого в низкий); 1 – уровнем (низкий)
1	IRQ1_EDGE	Запрос прерывания по входу IRQ1 вызывается: 0 – фронтом сигнала (из высокого в низкий); 1 – уровнем (низкий)
2	IRQ2_EDGE	Запрос прерывания по входу IRQ2 вызывается: 0 – фронтом сигнала (из высокого в низкий); 1 – уровнем (низкий)
3	IRQ3_EDGE	Запрос прерывания по входу IRQ3 вызывается: 0 – фронтом сигнала (из высокого в низкий); 1 – уровнем (низкий)
4	TMR0RN	Разрешение работы таймера 0: 0 – таймер остановлен; 1 – таймер работает
5	TMR1RN	Разрешение работы таймера 1: 0 – таймер остановлен; 1 – таймер работает
31:6	-	Не используются и при чтении всегда равны нулю

9.2 Операции таймера

После сброса таймеры остановлены, т.к. биты TMRxRN в регистре INTCTL сбрасываются. Для запуска и работы таймера необходимо установить соответствующий бит в TMRxRN. После установки бита значение в TMRINxH/L копируется в регистр TIMERxH/L, и таймер начинает обратный отсчет.

Программы могут останавливать таймер, сбрасывая бит TMRxRN, и запускать его, устанавливая бит TMRxRN. Загрузка нового значения в регистр TMRINxH/L не

вызывает немедленной загрузки этого же значения в регистр счетчика TIMERxH/L. Новое начальное значение будет загружено в счетчик только в двух ситуациях:

- при изменении бита TMRxRN из 0 в 1, т.е. при включении таймера;
- при работе таймера в момент достижения счетчиком значения 0.

При работающем таймере в момент достижения счетчиком нуля, таймер выполняет следующие действия:

- генерирует два прерывания с высоким и низким приоритетом;
- загружает в счетчик начальное значение;
- начинает обратный отсчет.

Таймеры имеют биты запроса прерывания в регистрах ILATx (INT_TIMER0LP, INT_TIMER1LP, INT_TIMER0HP или INT_TIMER1HP). Прерывание от таймера импульсное и после установки может быть сброшено при обслуживании данного запроса.

Для управления таймерами используйте следующую процедуру:

1. Если таймер работает, отключите его сбросом бита TMRxRN в регистре INTCTL;
2. Если прерывание должно произойти, установите высокий или низкий приоритет прерывания таймера, выполнив следующие действия:
   - назначьте адрес(а) вектора прерываний для программы, загрузив регистр(ы) вектора прерываний (IVTIMERxH/LP);
   - разрешите прерывания таймеров, установив бит INT_TIMERxH/LP в регистре IMASK;
   - разрешите глобальные прерывания, установив бит SQCTL_GIE в регистре SQCTL;
3. Загрузите начальное значение в регистры TMRINxH/L;
4. Запустите таймер, установив бит TMRxRN в регистре INTCTL.

10 Таймеры с функцией Захвата/ШИМ

Все блоки таймеров выполнены на основе 32-битного перезагружаемого счетчика, который синхронизируется с выхода 16-битного предделителя. Перезагружаемое значение хранится в отдельном регистре. Счет может быть прямой, обратный или двунаправленный (сначала прямой до определенного значения, а затем обратный).

Каждый из двух таймеров процессора содержит 32-битный счетчик, 16-битный предделитель частоты и 4-канальный блок захвата/сравнения. Их можно синхронизировать системной синхронизации, внешними сигналами или другими таймерами.

Помимо составляющего основу таймера счетчика, в каждый блок таймера также входит четырехканальный блок захвата/сравнения. Данный блок выполняет, как стандартные функции захвата и сравнения, так и ряд специальных функций. Таймеры имеют в своем составе 4 канала схем захвата, ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Каждый из таймеров может генерировать прерывания и запросы DMA.

Особенности:

- 32-битный вверх, вниз, вверх/вниз счетчик;
- 16-разрядный программируемый предварительный делитель частоты;
- до четырех независимых 32-битных каналов захвата на один таймер. Каждый из каналов захвата может захватить (скопировать) текущее значение таймера при изменении некоторого входного сигнала. В случае захвата имеется дополнительная возможность генерировать прерывание и/или запрос DMA;
- четыре 32-битных регистра сравнения (совпадения), которые позволяют осуществлять непрерывное сравнение, с дополнительной возможностью генерировать прерывание и/или запрос DMA при совпадении;
- имеется до четырёх внешних выводов, соответствующих регистрам совпадения со следующими возможностями:
  - сброс в НИЗКИЙ уровень при совпадении;
  - установка в ВЫСОКИЙ уровень при совпадении;
  - переключение (инвертирование) при совпадении;
  - при совпадении состояние выхода не изменяется;
  - переключение при некотором условии.

Внешние выводы блоков таймеров приведены в таблице 38

Таблица 38 – Внешние выводы блоков таймеров

Обозначение вывода (основное)	Обозначение вывода блоков таймеров	Тип вывода	Функциональное назначение
PA[22]	GTMR0 CH0o	I/O	Таймер 0. Выход 0 ШИМ (+)
PA[23]	GTMR0 nCH0o	I/O	Таймер 0. Выход 0 ШИМ (-)
PA[24]	GTMR0 CH1o	I/O	Таймер 0. Выход 1 ШИМ (+)
PA[25]	GTMR0 nCH1o	I/O	Таймер 0. Выход 1 ШИМ (-)
PA[26]	GTMR0 CH2o	I/O	Таймер 0. Выход 2 ШИМ (+)
PA[27]	GTMR0 nCH2o	I/O	Таймер 0. Выход 2 ШИМ (-)
PA[28]	GTMR0 CH3o	I/O	Таймер 0. Выход 3 ШИМ (+)
PA[29]	GTMR0 nCH3o	I/O	Таймер 0. Выход 3 ШИМ (-)

Обозначение вывода (основное)	Обозначение вывода блоков таймеров	Тип вывода	Функциональное назначение
PA[30]	GTMR0 BRK	I/O	Таймер 0. Блокировка выходов
PA[31]	GTMR0 ETR	I/O	Таймер 0. Универсальный регистратор событий
PB[0]	GTMR0 CH0i	I/O	Таймер 0. Вход 0 захвата
PB[1]	GTMR0 CH1i	I/O	Таймер 0. Вход 1 захвата
PB[2]	GTMR0 CH2i	I/O	Таймер 0. Вход 2 захвата
PB[3]	GTMR0 CH3i	I/O	Таймер 0. Вход 3 захвата
PB[8]	GTMR1 CH0o	I/O	Таймер 1. Выход 0 ШИМ (+)
PB[9]	GTMR1 nCH0o	I/O	Таймер 1. Выход 0 ШИМ (-)
PB[10]	GTMR1 CH1o	I/O	Таймер 1. Выход 1 ШИМ (+)
PB[11]	GTMR1 nCH1o	I/O	Таймер 1. Выход 1 ШИМ (-)
PB[12]	GTMR1 CH2o	I/O	Таймер 1. Выход 2 ШИМ (+)
PB[13]	GTMR1 nCH2o	I/O	Таймер 1. Выход 2 ШИМ (-)
PB[14]	GTMR1 CH3o	I/O	Таймер 1. Выход 3 ШИМ (+)
PB[15]	GTMR1 nCH3o	I/O	Таймер 1. Выход 3 ШИМ (-)
PB[16]	GTMR1 BRK	I/O	Таймер 1. Блокировка выходов
PB[17]	GTMR1 ETR	I/O	Таймер 1. Универсальный регистратор событий
PB[18]	GTMR1 CH0i	I/O	Таймер 1. Вход 0 захвата
PB[19]	GTMR1 CH1i	I/O	Таймер 1. Вход 1 захвата
PB[20]	GTMR1 CH2i	I/O	Таймер 1. Вход 2 захвата
PB[21]	GTMR1 CH3i	I/O	Таймер 1. Вход 3 захвата

10.1 Функционирование

Таймер предназначен для того, чтобы подсчитывать циклы периферийной тактовой частоты Fdts или какие-либо внешние события и произвольно генерировать прерывания, запросы DMA или выполнять другие действия. Значения таймера, при достижении которых будут выполнены те или иные действия, задаются восемью регистрами совпадения. Кроме того, в процессоре имеются четыре входа захвата, чтобы захватить значение таймера при изменении некоторого входного сигнала, с возможностью генерировать прерывание или запрос DMA.

Структурная схема блока Таймер представлена ниже (Рисунок 20).

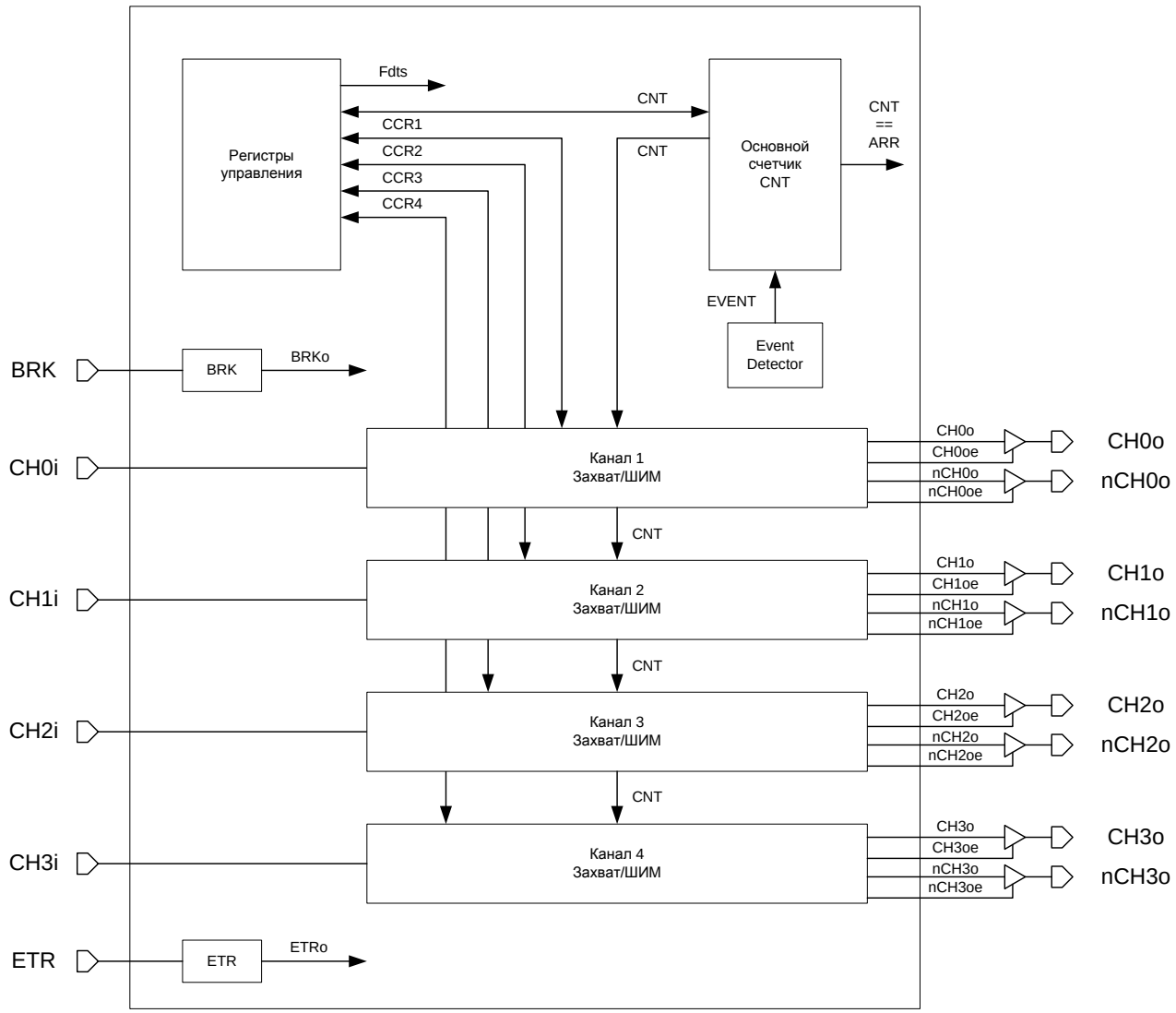


Рисунок 20 – Структурная схема таймера

Таймер содержит основной 32-битный счетчик CNT, блок регистров управления и четыре канала схем Захвата/ШИМ.

Таймер позволяет работать в режимах:

- таймер;
- расширенный таймер, с объединением нескольких таймеров;
- схема захвата;
- схема ШИМ.

10.1.1 Инициализация таймера

Перед началом работы с таймерами в первую очередь должны быть включены тактовые сигналы. Параметры задаются в блоке «Сигналы сброса и тактовой частоты».

Для задания тактовой частоты блока необходимо установить бит разрешения тактирования блока (бит 19 для таймера 0, бит 20 для таймера 1 регистра CFG8). Каждый таймер тактируется одной единственной частотой SOC-шины.

После подачи тактового сигнала на блок таймера можно приступать к работе с ним.

10.1.2 Режим таймера

Таймеры построены на базе 32-битного счетчика, объединенного с 16-битным предварительным делителем. Скорость счета таймера зависит от значения, находящегося в регистре делителя.

Счетчик может считать вверх, вниз или сначала вверх и затем вниз.

Базовый блок таймера включает в себя:

- основной счетчик таймера (CNT);
- делитель частоты при счете основного счетчика (PSC);
- основание счета основного счетчика (ARR).

Сигналом для изменения CNT может служить как внутренняя частота TIM_CLK, так и события в других счетчиках, либо события на линиях TxCHO данного счетчика.

Чтобы запустить работу основного счетчика необходимо задать:

- начальное значение основного счетчика таймера – CNT;
- значение предварительного делителя счетчика – PSG, при этом основной счетчик будет считать на частоте CLK= CLK/(PSG+1);
- значение основания счета для основного счетчика ARR;
- режим работы счетчика CNTRL:
  - выбрать источник события переключения счетчика EVENT_SEL;
  - режим счета основного счетчика CNT_MODE (значения 00 и 01 при тактировании внутренней частотой, значения 10 и 11 при тактировании внешними сигналами);
  - направление счета основного счетчика DIR;
- разрешить работу счетчика CNT_EN.

По событиям совпадения значения основного счетчика со значением нуля или значением основания счета генерируется прерывание и запрос DMA, которые могут быть замаскированы.

10.2 Режимы счета

Счет вверх: CNT_MODE = 00, DIR = 0.

```
GTIMERx->CNTRL  = 0x00000000;//Режим инициализации таймера

//Настраиваем работу основного счетчика

GTIMERx->CNT = 0x00000004;//Начальное значение счетчика

GTIMERx->PSG = 0x00000000;//Предделитель частоты

GTIMERx->ARR = 0x00000013;//Основание счета



//Разрешение работы таймера.

GTIMERx->CNTRL = 0x00000001;//Счет вверх по TIM_CLK.
```

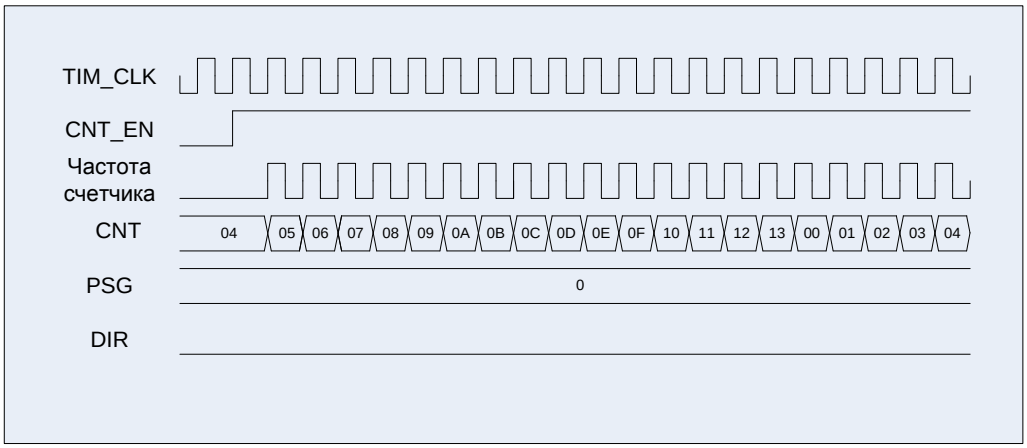


Рисунок 21 – Диаграммы работы таймера, счет вверх от 0 до 0x0013, стартовое значение 0x0004

Счет вниз: CNT_MODE = 00, DIR = 1

```
GTIMERx->CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
GTIMERx->CNT = 0x00000004;//Начальное значение счетчика
GTIMERx->PSG = 0x00000000;//Предделитель частоты
GTIMERx->ARR = 0x00000013;//Основание счета

//Разрешение работы таймера.
GTIMERx->CNTRL = 0x00000009;//Счет вниз по TIM_CLK.
```

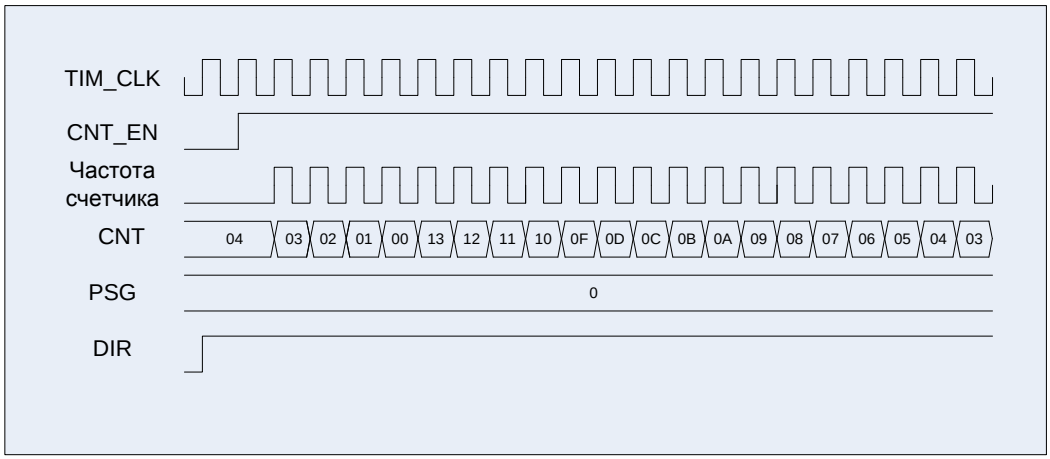


Рисунок 22 – Диаграммы работы таймера, счет вниз от 0x0013 до 0, стартовое значение 0x0004

Счет вверх/вниз: CNT_MODE = 01, DIR = 0

```
GTIMERx->CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
GTIMERx->CNT = 0x00000004;//Начальное значение счетчика
GTIMERx->PSG = 0x00000000;//Предделитель частоты
GTIMERx->ARR = 0x00000013;//Основание счета

//Разрешение работы таймера.
GTIMERx->CNTRL = 0x00000041;//Счет вверх/вниз по TIM_CLK.
```

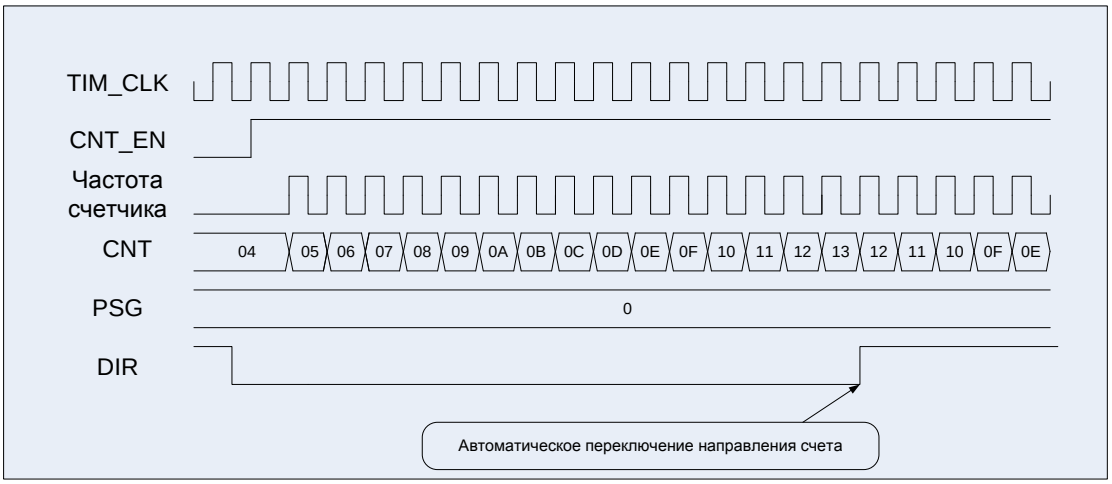


Рисунок 23 – Диаграммы работы таймера, счет вверх/вниз, сначала вверх

Счет вверх/вниз: CNT_MODE = 01, DIR = 1

```
GTIMERx->CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
GTIMERx->CNT = 0x00000004;//Начальное значение счетчика
GTIMERx->PSG = 0x00000000;//Предделитель частоты
GTIMERx->ARR = 0x00000013;//Основание счета

//Разрешение работы таймера.
GTIMERx->CNTRL = 0x00000049;//Счет вверх/вниз по TIM_CLK.
```

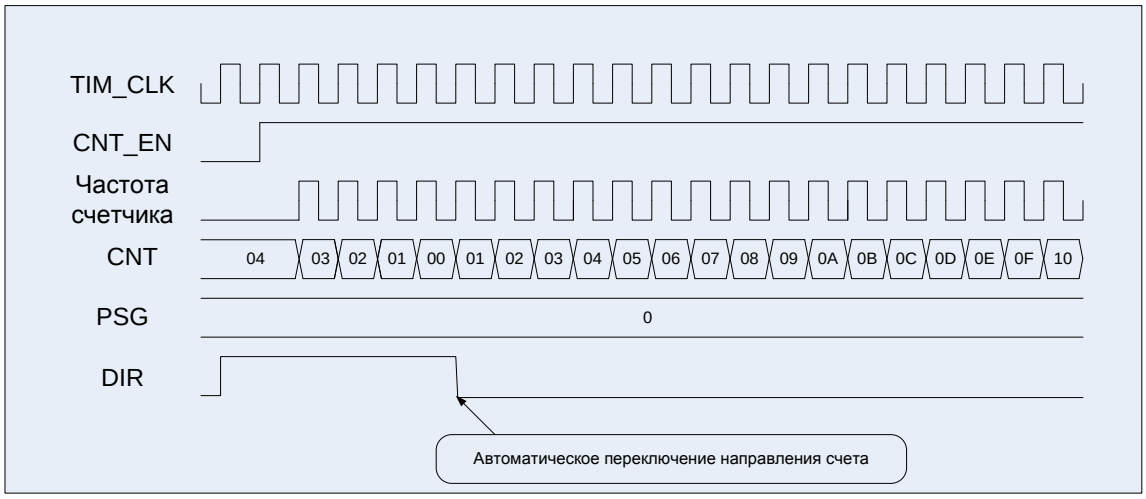


Рисунок 24 – Диаграммы работы таймера, счет вверх/вниз, сначала вниз

10.3 Источник событий для счета

Источники событий для счета:

- внутренний тактовый сигнал (TIM_CLK);
- события в других счетчиках (CNT==ARR в таймере X);
- внешний тактовый сигнал режим 1: События на линиях TxCHO данного счетчика;
- внешний тактовый сигнал режим 2: События на входе ETR данного счетчика.

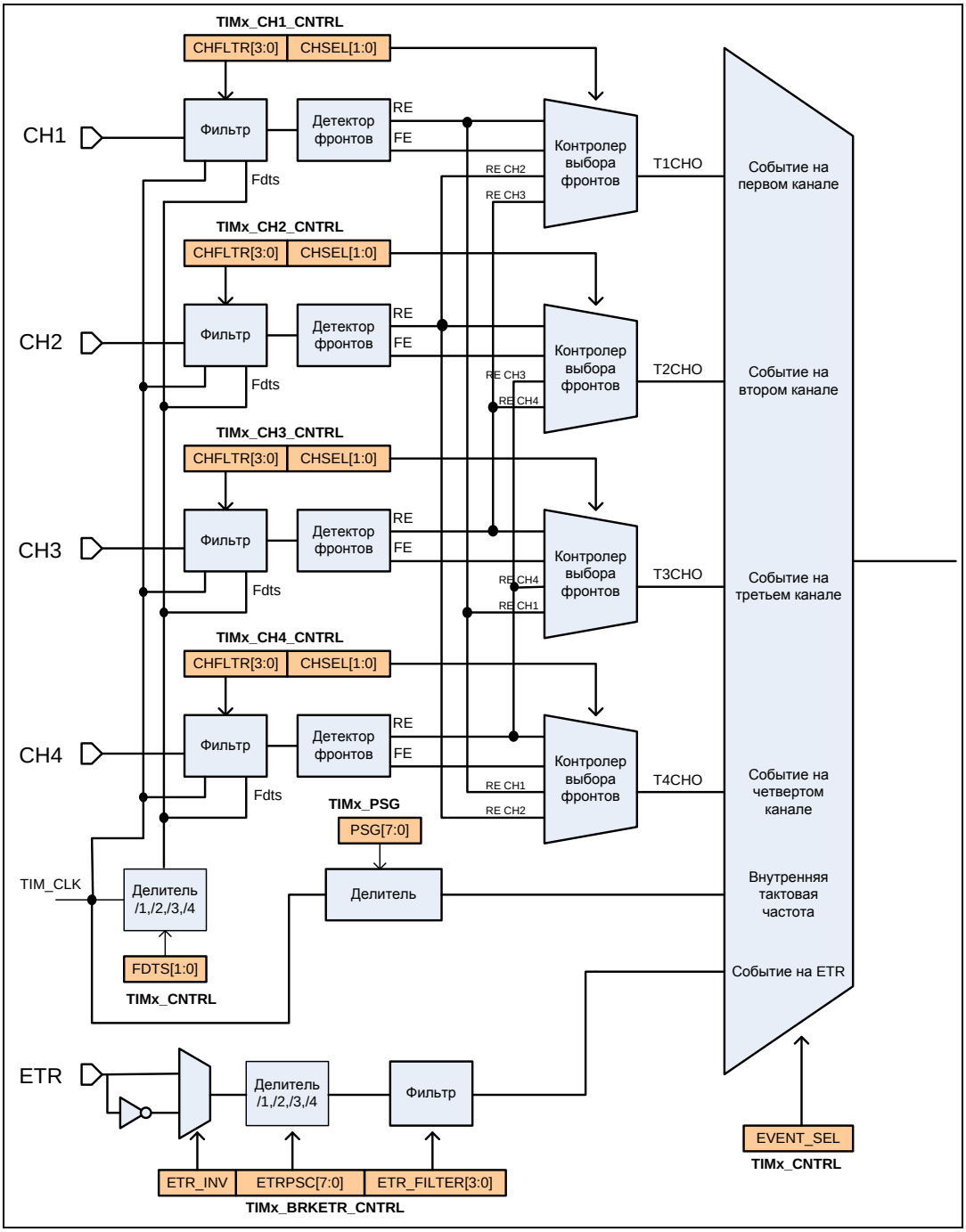


Рисунок 25 – Структурная схема формирования события для счета

10.3.1 Внутренний тактовый сигнал (TIM_CLK)

Режим выбирается, когда CNT_MODE = 0x, EVENT_SEL = 0000. Для запуска режима необходимо задать начальное значение основного счетчика, значение

предварительного делителя основного счетчика, основание счета для основного счетчика и задать режим работы в регистре CNTRL.

Значения регистров CNT, PSG и ARR можно изменять даже во время работы счетчика, при этом их значения вступят в силу по CNT = ARR или CNT = 0, в зависимости от направления счета. Значение регистра основания счета (ARR) может вступить в силу мгновенно после его записи в регистр при условии, что установлен флаг ARRB_EN = 1 (регистр CNTRL). Если значение предварительного делителя основного счетчика не равно нулю, то счетный регистр делителя будет инкрементироваться по каждому импульсу сигнала TIM_CLK до тех пор, пока не достигнет значения, находящегося в регистре делителя. Далее счетный регистр делителя сбрасывается в ноль, содержимое основного счетчика таймера измениться на 1 и снова начинается счет.

Поле DIR определяет, в какую сторону будет меняться значение счетчика: DIR = 0 – счетчик считает вверх (Рисунок 26), DIR = 1 – счетчик считает вниз (Рисунок 27). Если CNT_MODE = 00, то направление счета определяется полем DIR, если CNT_MODE = 01, счетчик считает вверх/вниз с автоматическим изменением DIR (Рисунок 28).

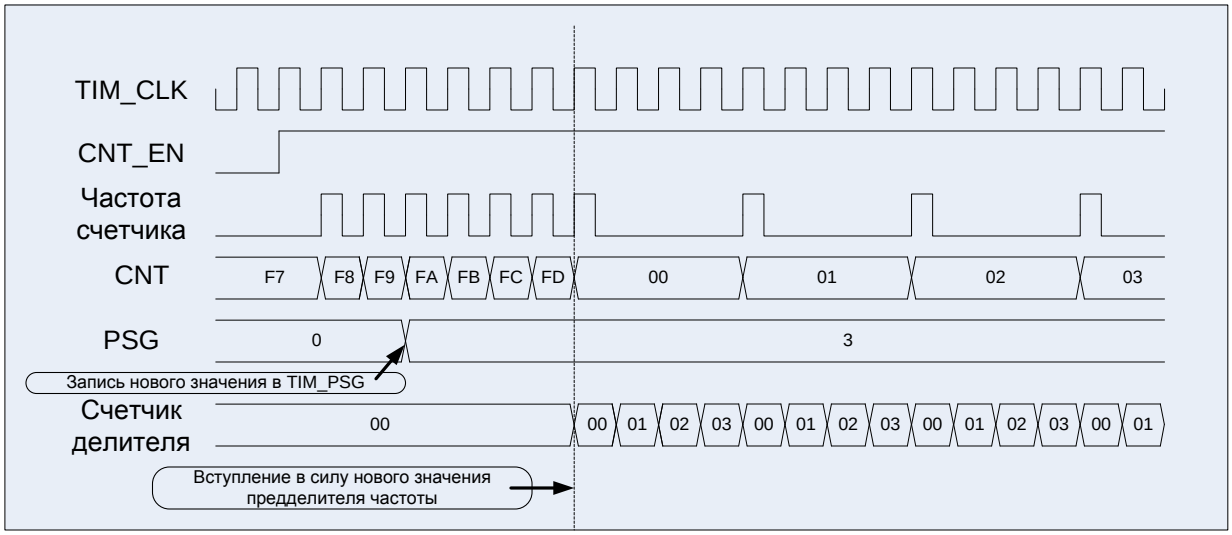


Рисунок 26 – Диаграммы работы счетчика: счет вверх

(CNT_MODE = 00, EVENT_SEL = 0000, DIR = 0)

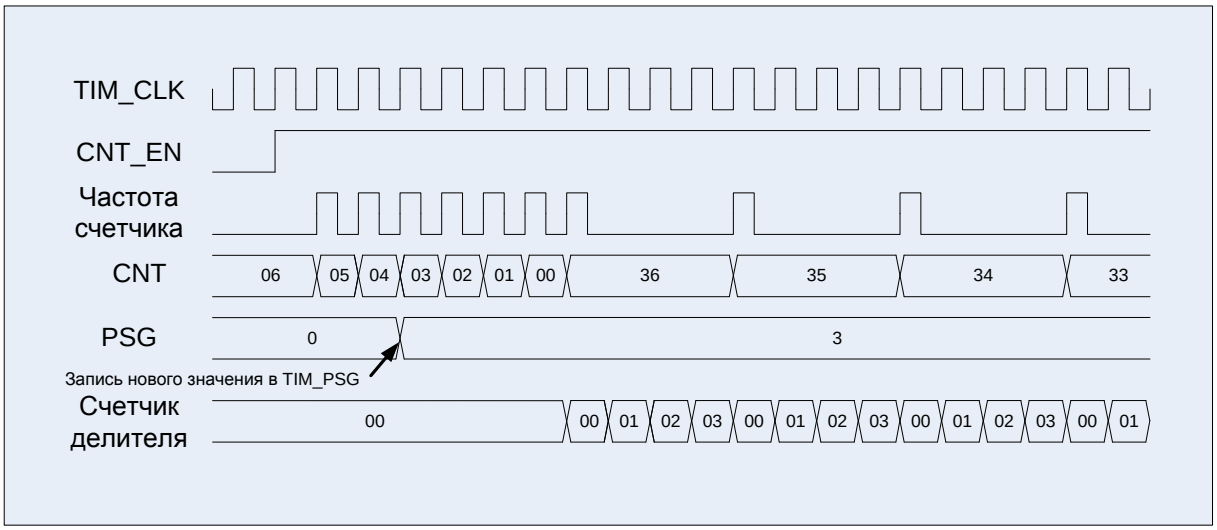


Рисунок 27 – Диаграммы работы счетчика: счет вниз

(CNT_MODE = 00, EVENT_SEL = 0000, DIR = 1)

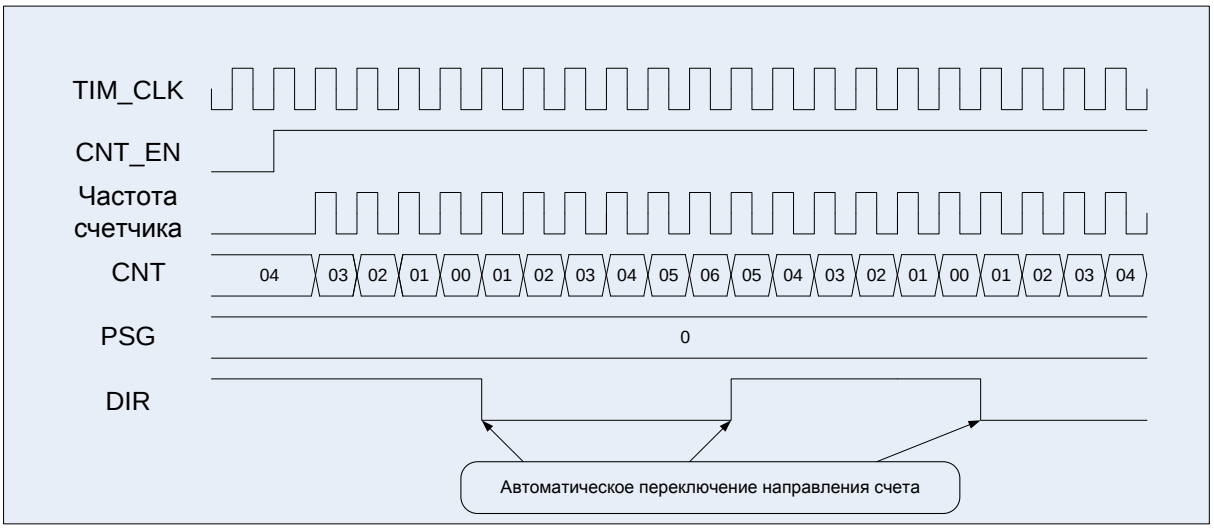


Рисунок 28 – Диаграммы работы счетчика: счет вниз/вверх

(CNT_MODE = 01, EVENT_SEL = 0000, DIR = 1)

10.3.2 События в других счетчиках (CNT==ARR в таймере X)

Каждый из блоков таймеров полностью независим друг от друга, но у них предусмотрена возможность синхронизированной друг с другом работы. Это позволяет создавать более сложные массивы таймеров, которые работают полностью автономно и не требуют написания какого-либо кода программы для выполнения сложных временных функций.

У каждого таймера имеются входы запуска от других трех таймеров, а также внешние входы, связанные с выводами блоков захвата/сравнения.

У каждого из блоков таймеров имеется выход запуска, который соединен с входами других трех таймеров. Синхронизация таймеров возможна в нескольких различных режимах. Ниже показан пример каскадного соединения счетчиков. Следует иметь в виду, что это общая схема каскадного включения. В процессоре реализовано только два таймера – GTIMER0 и GTIMER1.

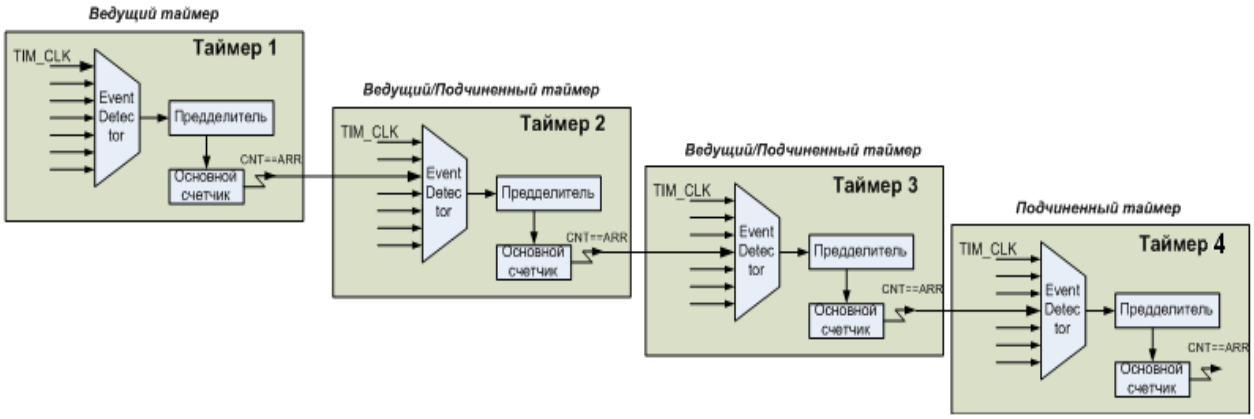


Рисунок 29 – Пример каскадного соединения таймеров

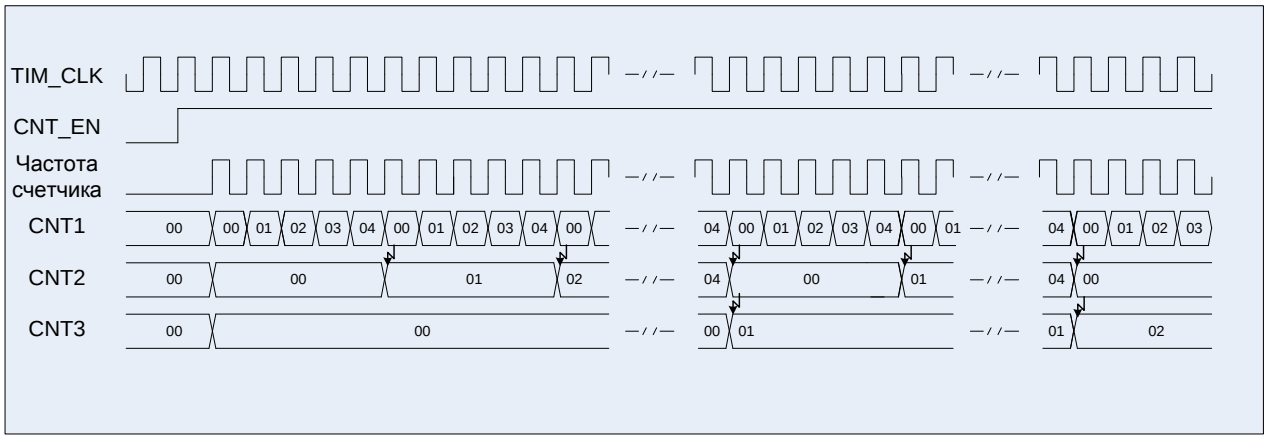


Рисунок 30 – Диаграммы работы трех таймеров в каскаде

DIR_1, DIR_2, DIR_3 = 0;
EVENT_SEL_1 = 0000, EVENT_SEL_2 = 0001, EVENT_SEL_3 = 0010;
CNT_MODE_1, CNT_MODE_2, CNT_MODE_3 = 00;

10.3.3 Внешний тактовый сигнал режим 1. События на линиях TxCHO данного счетчика

Этот режим выбирается, когда EVENT_SEL = 01xx в регистре CNTRL. Счетчик может считать по положительному фронту или по отрицательному фронту на выбранном входе или по положительному фронту на других каналах (Рисунок 31). На входе сигнала стоит фильтр, с помощью которого можно контролировать длительность сигнала, для фильтрации можно использовать как сигнал TIM_CLK, при этом может быть идентифицированная длительность 1, 2, 4, 8 TIM_CLK, также можно при фильтровании использовать производную от TIM_CLK частоту FDTS. Частота семплирования данных задается в регистре CNTRL в поле FDTS.

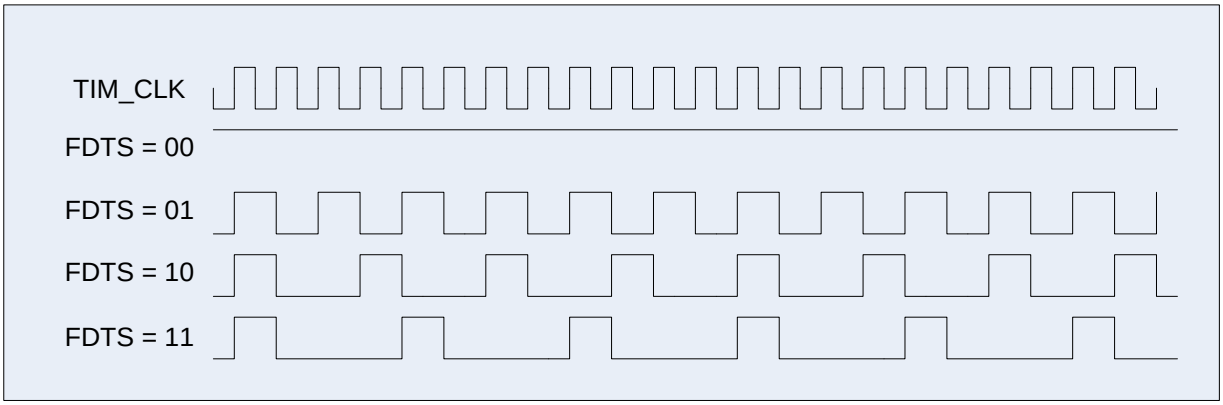


Рисунок 31 – Диаграммы возможных частот семплирования данных (FDTS)

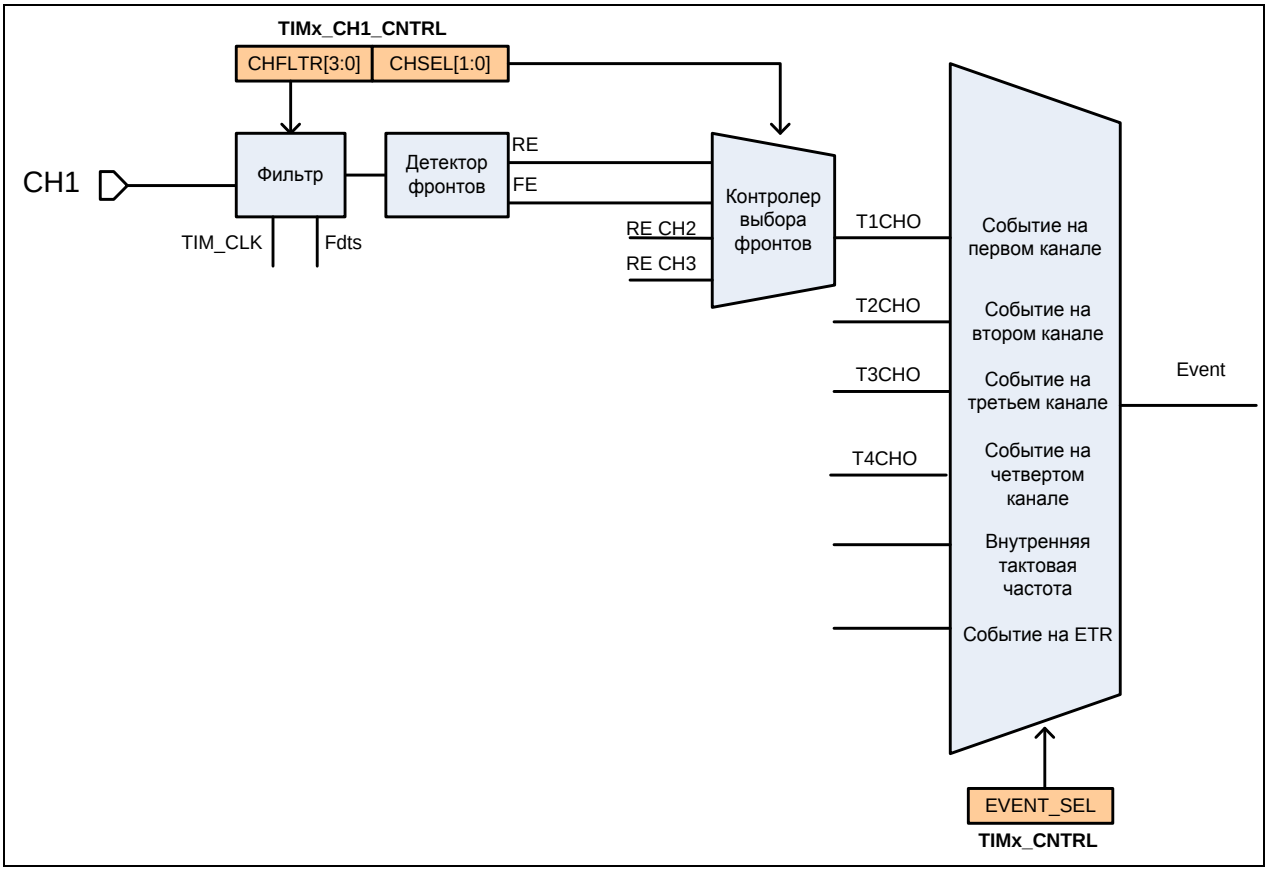


Рисунок 32 – Тактирование с входа первого канала

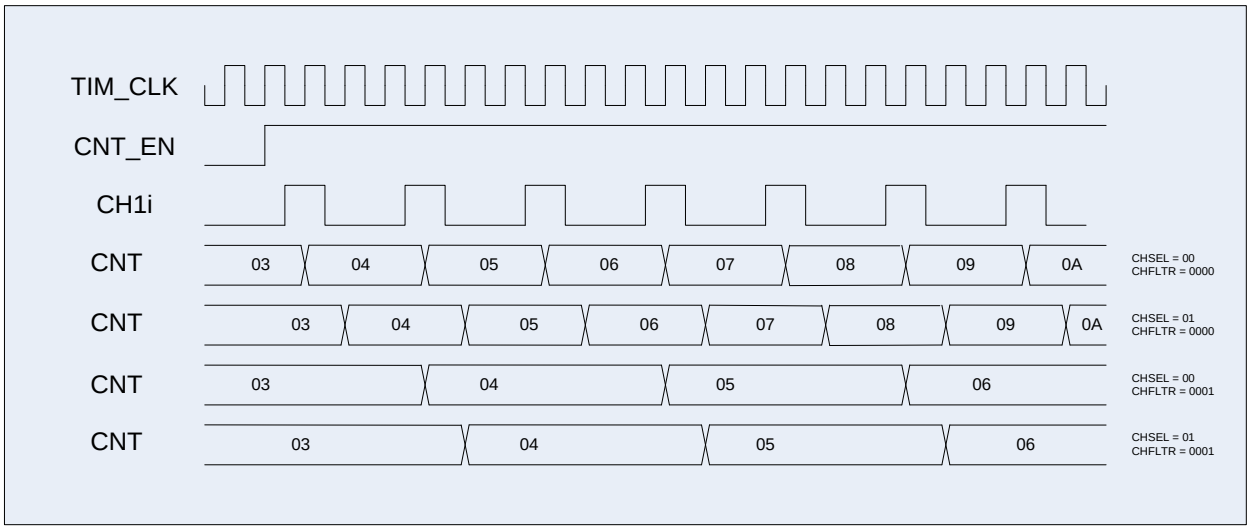


Рисунок 33 – Диаграмма внешнего тактирования с разными вариантами фильтра

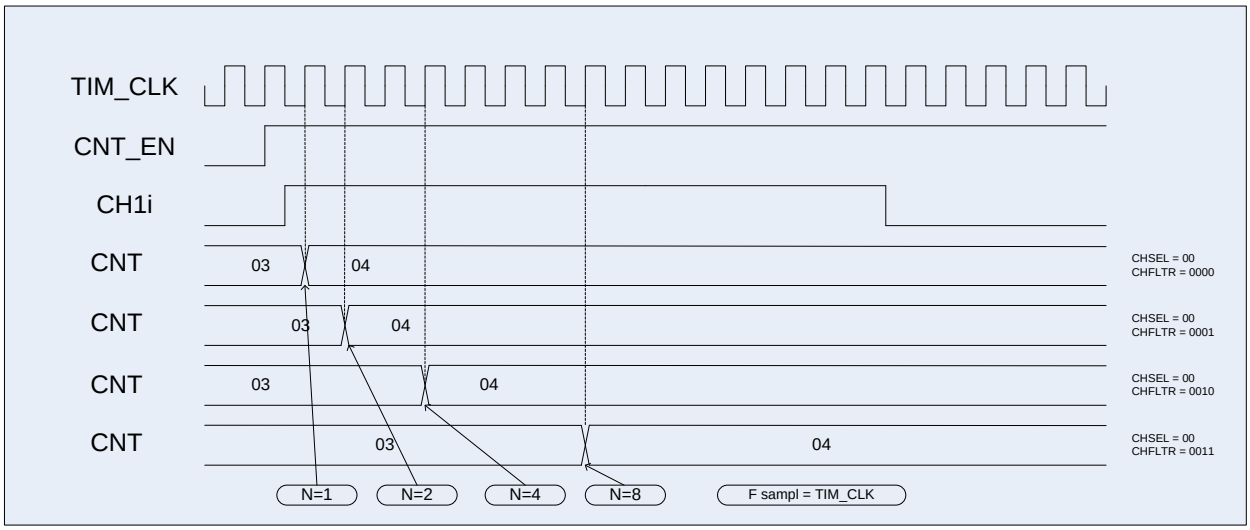


Рисунок 34 – Диаграмма внешнего тактирования с разными вариантами фильтра

10.3.4 Внешний тактовый сигнал режим 2. События на входе ETR данного счетчика

Этот режим выбирается, когда EVENT_SEL = 1000 в регистре CNTRL. В регистре BRKETR_CNTRL можно настроить коэффициент деления 2, 4 или 8 (ETRPSC) данного входа тактовой частоты, а также использовать инверсию входа.

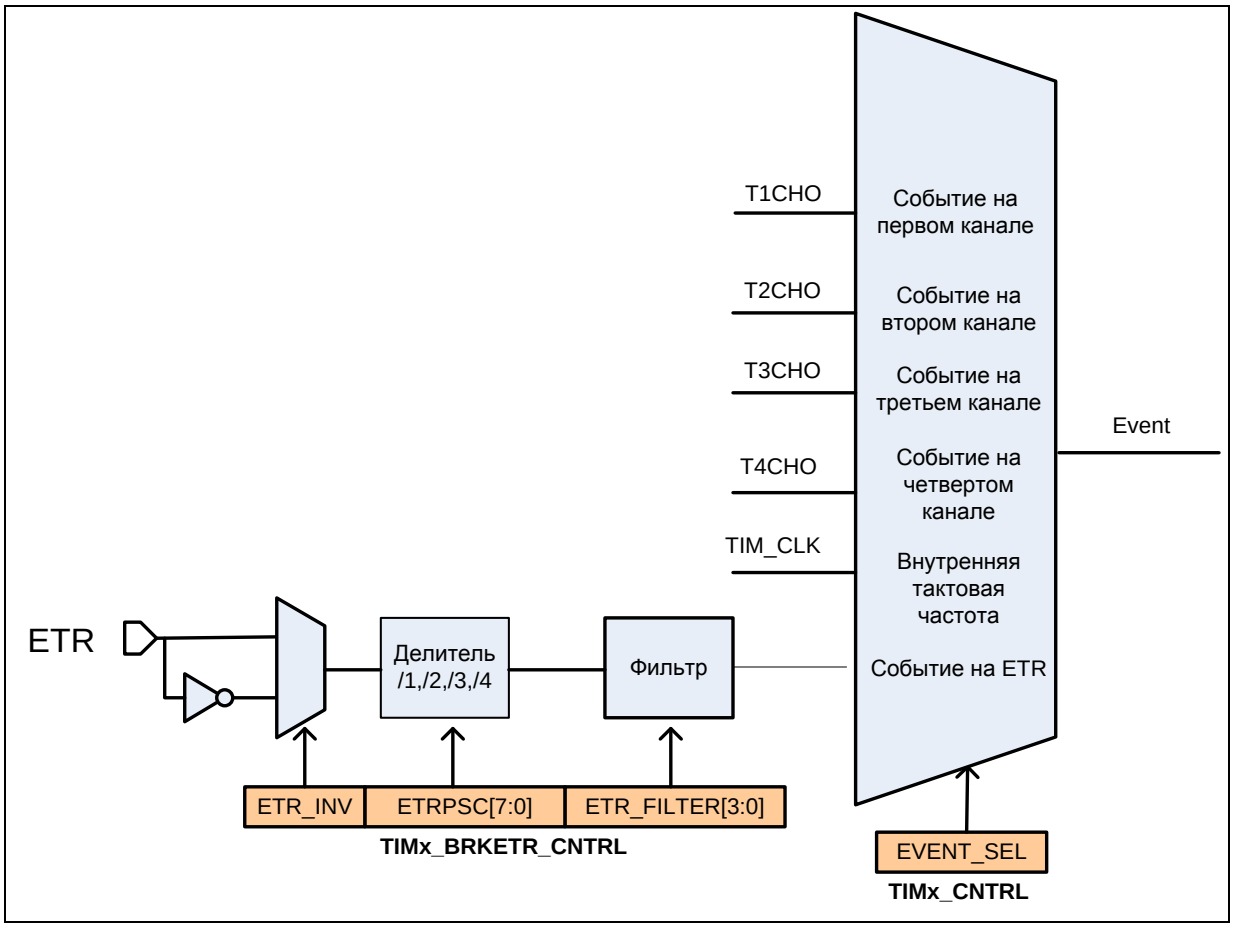


Рисунок 35 – Схема тактирования сигналом со входа ETR

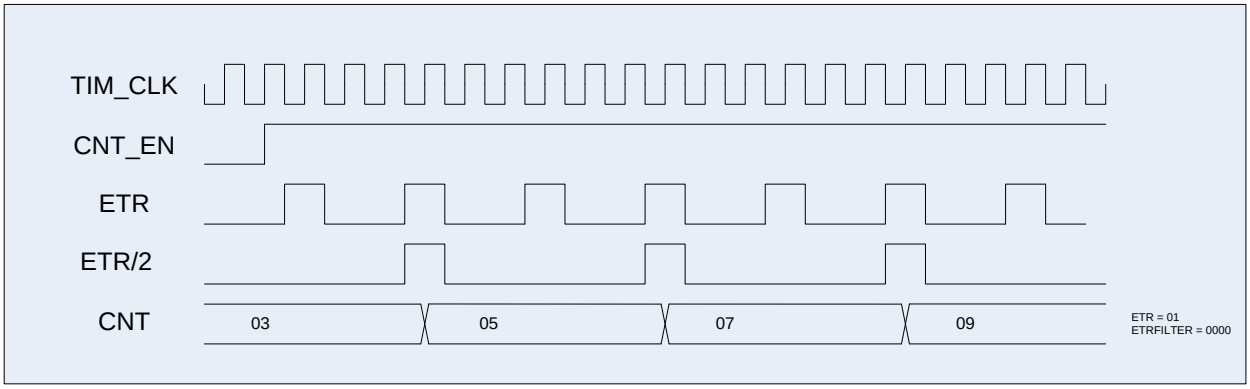


Рисунок 36 – Диаграмма тактирования сигналом со входа ETR

10.4 Режим захвата

Структурная схема блока Захвата представлена на рисунке ниже (Рисунок 37).

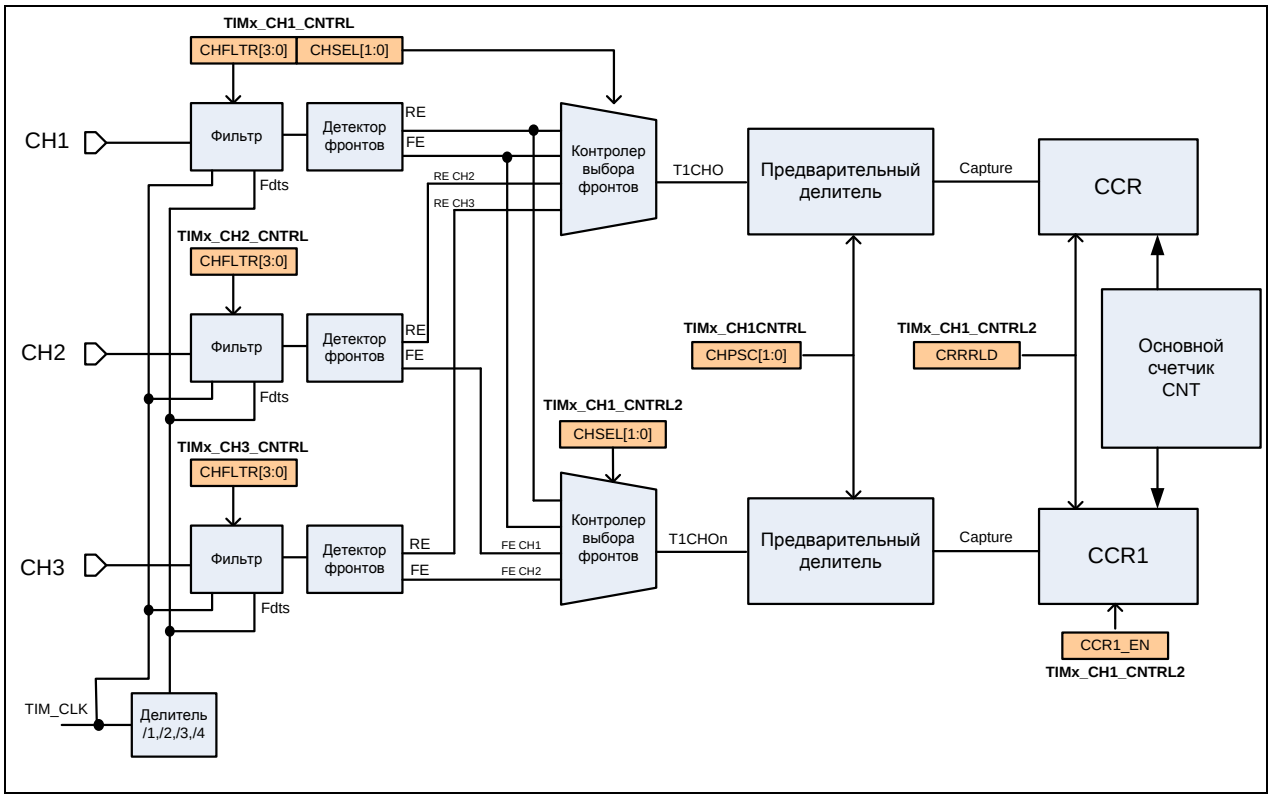


Рисунок 37 – Структурная схема блока захвата на примере канала 1

Для включения режима захвата для определенного канала необходимо в регистре управления каналом CHy_CNTRL записать 1 в поле CAPnPWM. Для регистрации событий по линии Chxi используется схема регистрации событий. Входной сигнал фиксируется в Таймере с частотой Fdts, или TIM_CLK. Также вход может быть настроен на прием импульсов заданной длины за счет конфигурирования блока FILTER. На выходе блока Фильтр вырабатывается сигнал положительного перепада и отрицательного перепада. На блоке MUX производится выбор используемого для Захвата сигнала, между положительным фронтом канала, отрицательным фронтом канала и положительными и отрицательными фронтами сигналов от других каналов. После блока MUX предварительный делитель может быть использован для фиксации каждого события, каждого второго, каждого

четвертого и каждого восьмого события. Выход предварительного делителя является сигналом Capture для регистра CCR, и Capture1 для регистра CCR1 при этом в регистры CCR и CCR1 записывается текущее значение основного счетчика CNT.

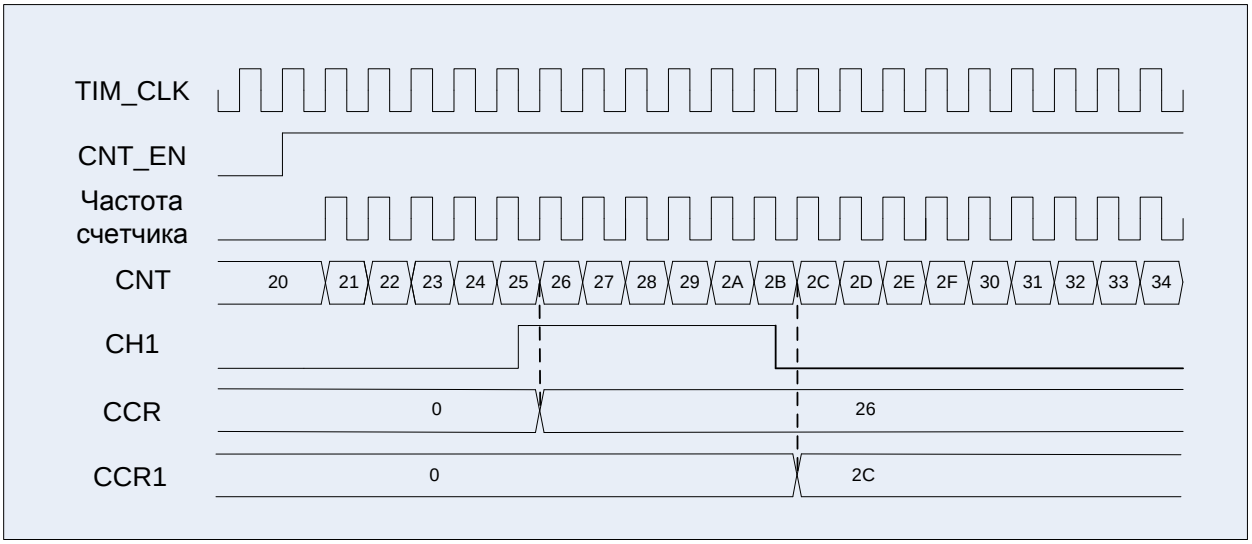


Рисунок 38 – Диаграмма захвата события со входа первого канала

На рисунке показан пример захвата значения основного счетчика в регистр CCR по положительному фронту на входе канала, а в регистр CCR1 по отрицательному фронту на входе канала. В регистре IE можно разрешить выработку прерываний по событию захвата на определенном канале, а в регистре DMA_RE можно разрешить формирование запросов DMA.

10.5 Режим ШИМ

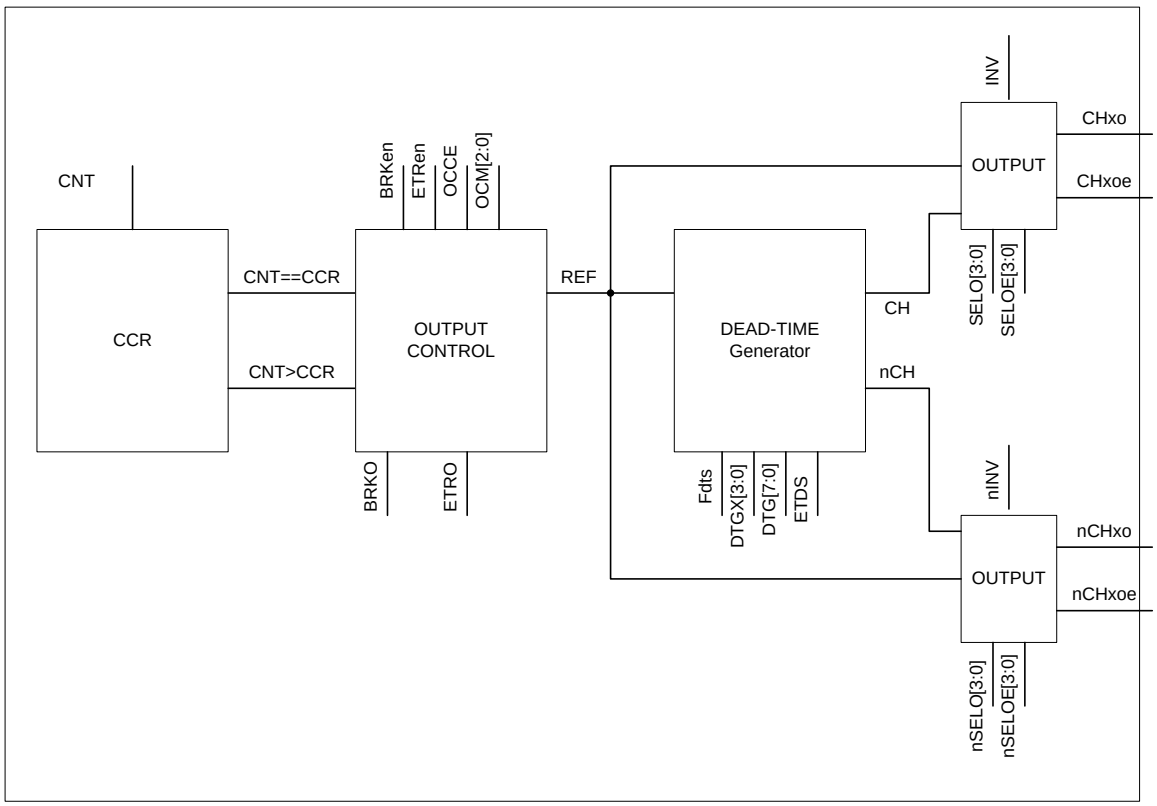


Рисунок 39 – Структурная схема блока сравнения

Для включения режима сравнения для определенного канала необходимо в регистре управления каналом CHy_CNTRL записать 0 в поле CAPnPWM. При работе в режиме ШИМ выходной сигнал может формироваться на основании сравнения значения в регистре CCR и основного счетчика CNT или регистров CCR, CCR1 и значения основного счетчика CNT. Полученный сигнал может без изменения выдаваться на выводы ChxO и nCHxO. Либо с применением схемы DEAD TIME Generator формируются управляющие сигналы с мертвой зоной. У каждого канала есть два выхода: прямой и инверсный. Для каждого выхода формируется как сигнал для выдачи, так и сигнал разрешения выдачи, т.е. если выход канала должен всегда выдавать тот или иной уровень, то на выводе разрешения выдачи ChxOE (для прямого) и на CHxNOE (для инверсного) должны формироваться «1». Если канал работает на вход (например, режим захвата), то там всегда должен быть «0» для прямого канала. Сигналы OE работают по тем же принципам, что и просто выходные уровни, но у них есть собственные сигналы разрешения вывода SELOE и nSELOE, в которых можно выбрать постоянный уровень, либо формируемый на основании REF.

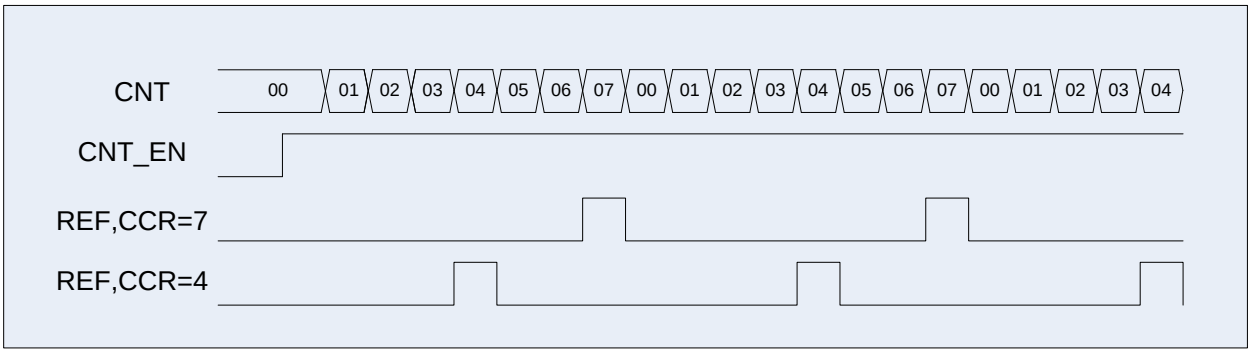


Рисунок 40 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

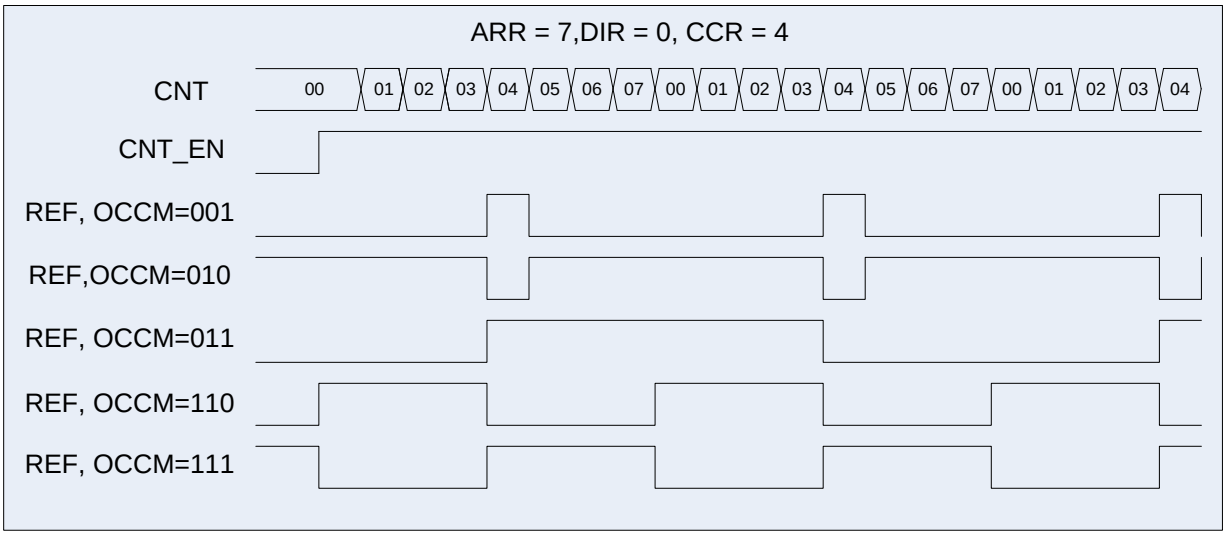


Рисунок 41 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN=0

Сигнал REF может быть очищен с использованием внешнего сигнала с входа ETR или внешнего триггерированного по PCLK сигнала с входа BRK.

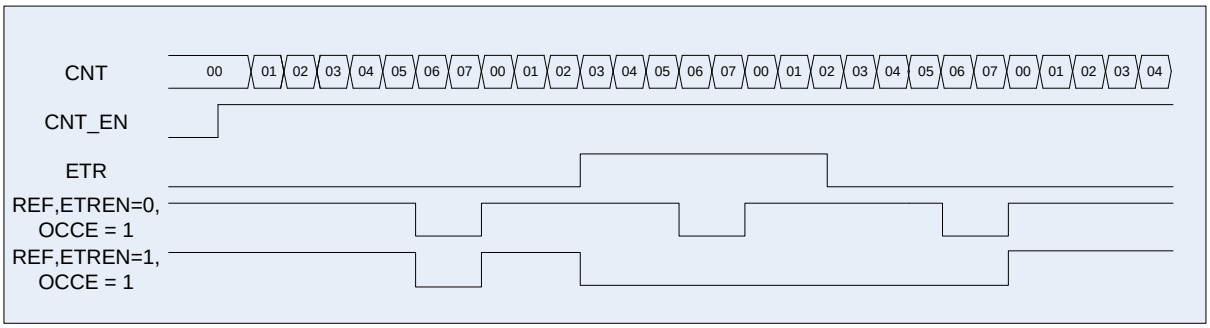


Рисунок 42 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 0

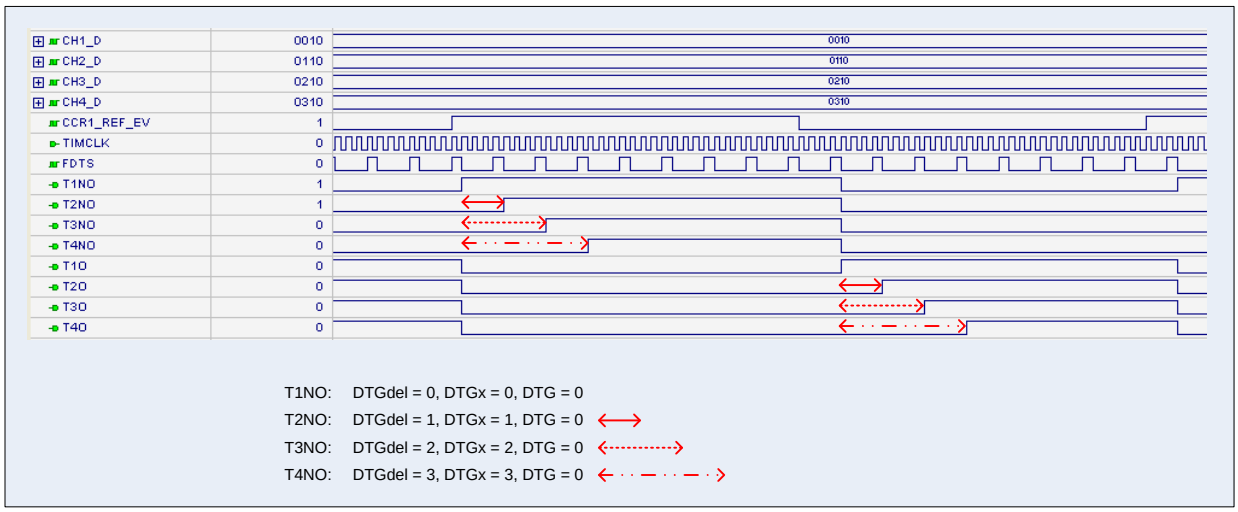


Рисунок 43 – Диаграмма работы схемы DTG

Если CCR1_EN = 1, тогда значение основного счетчика CNT сравнивается со значениями регистров CCR и CCR1, и в зависимости от запрограммированного формата выработки сигнала REF (регистры управления каналами таймера CHy_CNTRL поле OCCM) будет формироваться сигнал соответствующей формы.

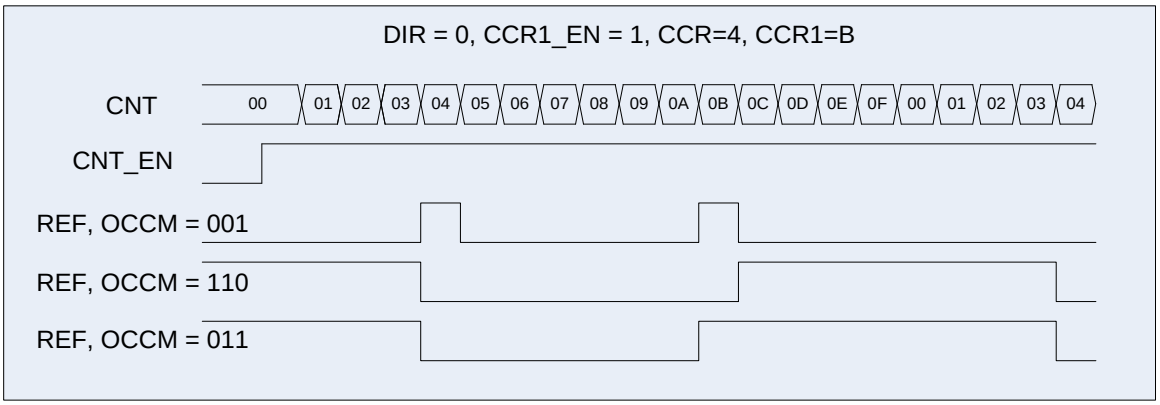


Рисунок 44 – Диаграмма работы схемы в режиме ШИМ, CCR1_EN = 1

При записи новых значений CCR и CCR1, если установлен бит CRRRLD, то регистры CCR1 и CCR получат новые значения только при CNT = 0, иначе запись осуществляется немедленно. Факт окончания записи обозначается взведением флага WR_CMPL.

10.6 Примеры

10.6.1 Обычный счетчик

```
GTIMERx->CNTRL  = 0x00000000;
//Настраиваем работу основного счетчика
GTIMERx->CNT = 0x00000000;//Начальное значение счетчика
GTIMERx->PSG = 0x00000000;//Предделитель частоты
GTIMERx->ARR = 0x0000000F;//Основание счета

GTIMERx->IE = 0x00000002;//Разрешение генерировать прерывание при CNT = ARR

GTIMERx->CNTRL = 0x00000001;//Счет вверх по TIM_CLK. Разрешение работы таймера.
```

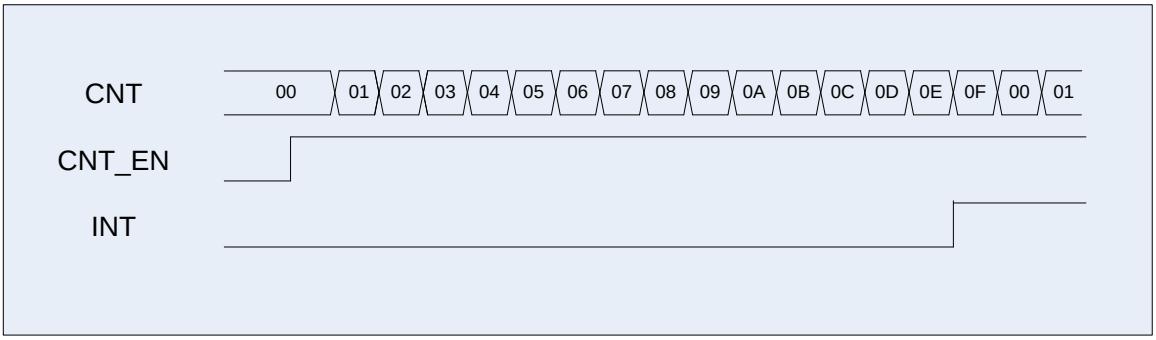


Рисунок 45 – Режим обычного счетчика

10.6.2 Режим захвата

```
GTIMERx->CNTRL  = 0x00000000;//Режим инициализации таймера
//Настраиваем работу основного счетчика
GTIMERx->CNT = 0x00000000;//Начальное значение счетчика
GTIMERx->PSG = 0x00000000;//Предделитель частоты
GTIMERx->ARR = 0x000000FF;//Основание счета

GTIMERx->IE = 0x00001E00;//Разрешение генерировать прерывание
                         //по переднему фронту на выходе CAP по всем каналам
//Режим работы каналов – захват
GTIMERx->Chy_CNTRL[0] = 0x00008000;
GTIMERx->Chy_CNTRL[1] = 0x00008002;
GTIMERx->Chy_CNTRL[2] = 0x00008001;
GTIMERx->Chy_CNTRL[3] = 0x00008003;
```

//Режим работы выхода канала – канал на выход не работает

GTIMERx->Chy_CNTRL1[0]= 0x00000000;

GTIMERx->Chy_CNTRL1[1]= 0x00000000;

GTIMERx->Chy_CNTRL1[2]= 0x00000000;

GTIMERx->Chy_CNTRL1[3]= 0x00000000;

GTIMERx->CNTRL = 0x00000001;//Счет вверх по TIM_CLK. Разрешение работы таймера.

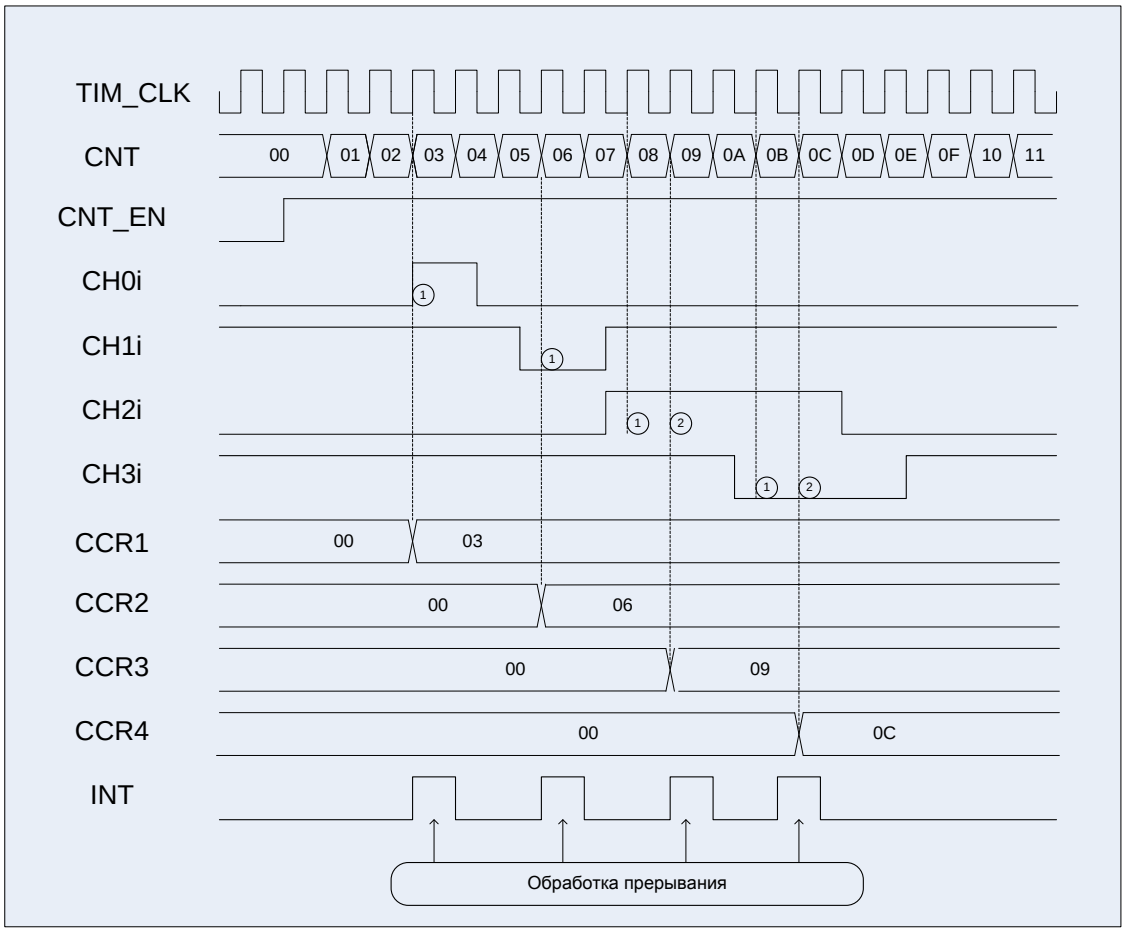


Рисунок 46 – Диаграммы примера работы в режиме захвата

10.6.3 Режим ШИМ

GTIMERx->CNTRL = 0x00000000;//Режим инициализации таймера

//Настраиваем работу основного счетчика

GTIMERx->CNT = 0x00000000;//Начальное значение счетчика

GTIMERx->PSG = 0x00000000;//Предделитель частоты

GTIMERx->ARR = 0x00000010;//Основание счета

```
GTIMERx->IE = 0x000001E0;//Разрешение генерировать прерывание
                         //по переднему фронту на выходе REF по всем каналам
//Режим работы каналов – ШИМ
GTIMERx->Chy_CNTRL[0] = 0x00000200;
GTIMERx->Chy_CNTRL[1] = 0x00000200;
GTIMERx->Chy_CNTRL[2] = 0x00000400;
GTIMERx->Chy_CNTRL[3] = 0x00000600;

//Режим работы выхода канала – канал на выход не работает
GTIMERx->Chy_CNTRL1[0]= 0x00000099;
GTIMERx->Chy_CNTRL1[1]= 0x00000099;
GTIMERx->Chy_CNTRL1[2]= 0x00000099;
GTIMERx->Chy_CNTRL1[3]= 0x00000099;

//Разрешение работы таймера.
GTIMERx->CNTRL = 0x00000001;//Счет вверх по TIM_CLK.
```

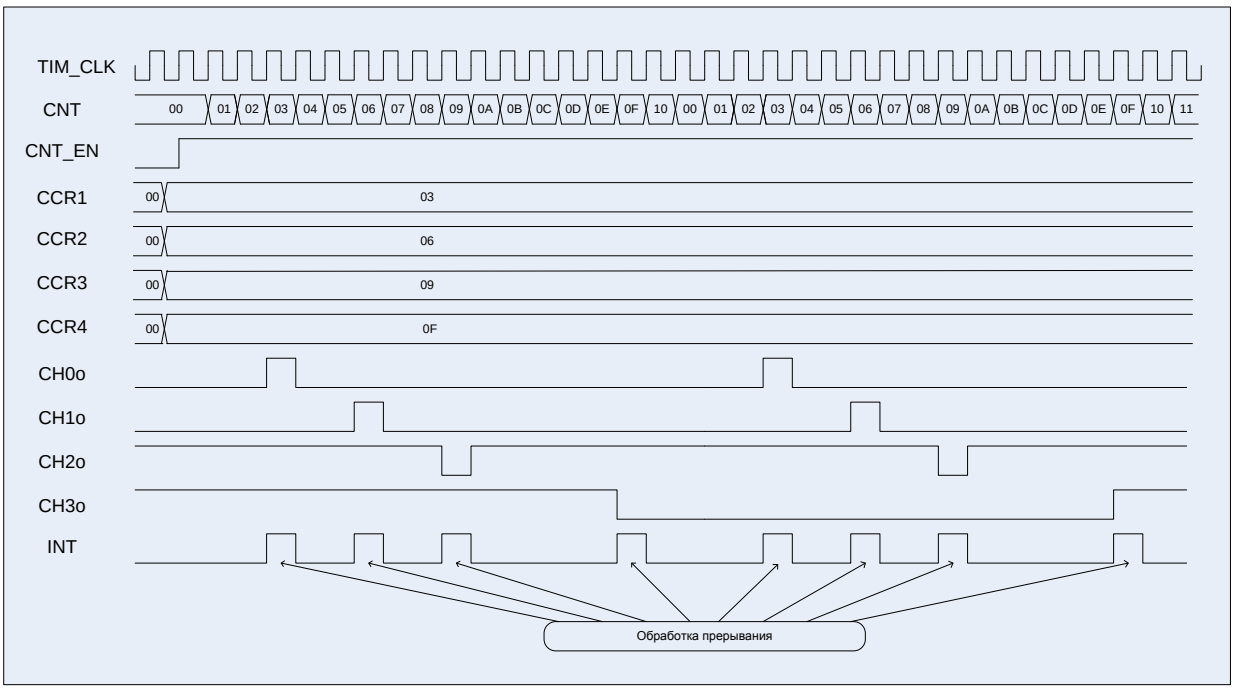


Рисунок 47 – Диаграммы примера работы в режиме ШИМ

10.7 Описание регистров блока таймера

Таблица 39 – Базовые адреса и смещения регистров управления таймера

Адрес	Название	Описание
0x8000_0400	GTIMER0	Контроллер GTIMER0
0x8000_0440	GTIMER1	Контроллер GTIMER1
Смещение		
0x00	CNT[31:0]	Основной счетчик таймера
0x01	PSG[15:0]	Делитель частоты при счете основного счетчика
0x02	ARR[31:0]	Основание счета основного счетчика
0x03	CNTRL[11:0]	Регистр управления основного счетчика
0x04	CCR1[31:0]	Регистр сравнения, захвата для 1 канала таймера
0x05	CCR2[31:0]	Регистр сравнения, захвата для 2 канала таймера
0x06	CCR3[31:0]	Регистр сравнения, захвата для 3 канала таймера
0x07	CCR4[31:0]	Регистр сравнения, захвата для 4 канала таймера
0x08	CH1_CNTRL[15:0]	Регистр управления для 1 канала таймера
0x09	CH2_CNTRL[15:0]	Регистр управления для 2 канала таймера
0x0A	CH3_CNTRL[15:0]	Регистр управления для 3 канала таймера
0x0B	CH4_CNTRL[15:0]	Регистр управления для 4 канала таймера
0x0C	CH1_CNTRL1[15:0]	Регистр управления 1 для 1 канала таймера
0x0D	CH2_CNTRL1[15:0]	Регистр управления 1 для 2 канала таймера
0x0E	CH3_CNTRL1[15:0]	Регистр управления 1 для 3 канала таймера
0x0F	CH4_CNTRL1[15:0]	Регистр управления 1 для 4 канала таймера
0x10	CH1_DTG[15:0]	Регистр управления DTG для 1 канала таймера
0x11	CH2_DTG[15:0]	Регистр управления DTG для 2 канала таймера
0x12	CH3_DTG[15:0]	Регистр управления DTG для 3 канала таймера
0x13	CH4_DTG[15:0]	Регистр управления DTG для 4 канала таймера
0x14	BRKETR_CNTRL[15:0]	Регистр управления входом BRK и ETR
0x15	STATUS[15:0]	Регситр статуса таймера
0x16	IE[15:0]	Регистр разрешения прерывания таймера
0x17	DMA_RE[15:0]	Регистр разрешения запросов DMA от прерываний таймера
0x18	CH1_CNTRL2[15:0]	Регистр управления 2 для 1 канала таймера
0x19	CH2_CNTRL2[15:0]	Регистр управления 2 для 2 канала таймера
0x1A	CH3_CNTRL2[15:0]	Регистр управления 2 для 3 канала таймера
0x1B	CH4_CNTRL2[15:0]	Регистр управления 2 для 4 канала таймера
0x1C	CCR11[31:0]	Регистр сравнения, захвата 1 для 1 канала таймера
0x1D	CCR21[31:0]	Регистр сравнения, захвата 1 для 2 канала таймера
0x1E	CCR31[31:0]	Регистр сравнения, захвата 1 для 3 канала таймера
0x1F	CCR41[31:0]	Регистр сравнения, захвата 1 для 4 канала таймера
0x20	DMA_RE1[15:0]	Регистр разрешения запросов DMA от прерываний канала 1 таймера
0x21	DMA_RE2[15:0]	Регистр разрешения запросов DMA от прерываний канала 2 таймера
0x22	DMA_RE3[15:0]	Регистр разрешения запросов DMA от прерываний канала 3 таймера
0x23	DMA_RE4[15:0]	Регистр разрешения запросов DMA от прерываний канала 4 таймера

10.7.1 CNT

Таблица 40 – Основной счетчик таймера CNT

Номер	31…0
Доступ	R/W
Сброс	0
	CNT[31:0]

Таблица 41 – Описание бит регистра CNT

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…0	CNT[31:0]	Значение основного счетчика таймера

10.7.2 PSG

Таблица 42 – Делитель частоты при счете основного счетчика PSG

Номер	31..16	15..0
Доступ	R/W	R/W
Сброс	0	0
	-	**PSG[15:0]**

Таблица 43 – Описание бит регистраPSG

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...16	-	Зарезервировано
15…0	PSG[15:0]	Значение предварительного делителя счетчика Основной счетчик считает на частоте CLK = TIM_CLK/(PSG+1)

10.7.3 ARR

Таблица 44 – Основание счета основного счетчика ARR

Номер	31…0
Доступ	R/W
Сброс	0
	ARR[31:0]

Таблица 45 – Описание бит регистра ARR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…0	ARR[31:0]	Основание счета для основного счетчика CNT = [0…ARR]

10.7.4 CNTRL

Таблица 46 – Регистр управления основного счетчика CNTRL

Номер	31…12	11…8	7…6	5…4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0000	00	00	0	0	0	0
	-	**EVENT SEL [3:0]**	**CNT MODE [1:0]**	**FDTS [1:0]**	**DIR**	**WR CMPL**	**ARRB EN**	**CNT EN**

Таблица 47 – Описание бит регистра CNTRL

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…12	-	Зарезервировано
11...8	EVENT_SEL [3:0]	Биты выбора источника событий 4’b0000 – всегда “0” 4’b0001 – CNT == ARR в таймере 1 4’b0010 – CNT == ARR в таймере 2 4’b0011 – CNT == ARR в таймере 3 4’b0100 – событие на первом канале 4’b0101 – событие на втором канале 4’b0110 – событие на третьем канале 4’b0111 – событие на четвертом канале 4’b1000 – событие переднего фронта ETR 4’b1001 – событие заднего фронта ETR 4’b1010 – CNT == ARR в таймере 4
7..6	CNT_MODE [1:0]	Режим счета основного счетчика 2’b00 – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при PSG = 0 2’b01 – счетчик вверх/вниз с автоматическим изменением DIR при CNT = 0 или CNT = ARR 2’b1x – счетчик вверх при DIR=0 счетчик вниз при DIR=1 при EVENT = 1
5...4	FDTS[1:0]	Частота семплирования данных FDTS 2’b00 – каждый TIM_CLK 2’b01 – каждый второй TIM_CLK 2’b10 – каждый третий TIM_CLK 2’b11 – каждый четвертый TIM_CLK
3	DIR	Направление счета основного счетчика 0 – вверх, от 0 до ARR 1 – вниз, от ARR до 0
2	WR_CMPL	Окончание записи, при задании нового значения регистров CNT, PSG и ARR 1 – данные не записаны и идет запись 0 – новые данные можно записывать
1	ARRB_EN	Разрешение мгновенного обновления ARR 0 – ARR будет перезаписан в момент записи в ARR 1 – ARR будет перезаписан при завершении счета CNT
0	CNT_EN	Разрешение работы таймера 0 – таймер отключен 1 – таймер включен

10.7.5 CCRy

Таблица 48 – Регистр сравнения/захвата для ‘y’ канала таймера CCRy

Номер	31…0
Доступ	R/W
Сброс	0
	CCR[31:0]

Таблица 49 – Описание бит регистра CCRy

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	CCR[31:0]	Значение CCR, с которым сравнивается CNT при работе в ШИМ режиме. Значение CNT, при котором произошел факт захвата события, в режиме захвата

10.7.6 CCRy1

Таблица 50 – Регистр сравнения/захвата для ‘y’ канала таймера CCRy1

Номер	31…0
Доступ	R/W
Сброс	0
	CCR1[31:0]

Таблица 51 – Описание бит регистра CCRy1

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…0	CCR1[31:0]	Значение CCR1, с которым сравнивается CNT при работе в ШИМ режиме. Значение CNT, при котором произошел факт захвата события, в режиме захвата

10.7.7 CHy_CNTRL

Таблица 52 – Регистр управления для ‘y’ канала таймера Chy_CNTRL

Номер	31…16	15
Доступ	U	R/W
Сброс	0	0
	-	**CAP nPWM**

Номер	14	13	12	11…9	8	7…6	5…4	3…0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	000	0	00	00	0000
	WR CMPL	**ETREN**	**BRKEN**	**OCCM [2:0]**	**OCCE**	**CHPSC [1:0]**	**CHSEL [1:0]**	**CHFLT R[3:0]**

Таблица 53 – Описание бит регистра Chy_CNTRL

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15	CAP nPWM	Режим работы канала Захват или ШИМ 1 – канал работает в режиме Захват 0 – канал работает в режиме ШИМ
14	WR CMPL	Флаг окончания записи, при задании нового значения регистра CCR 1 – данные не записаны и идет запись 0 – новые данные можно записывать
13	ETREN	Разрешения сброса по выводу ETR 0 – запрещен сброс 1 – разрешен
12	BRKEN	Разрешение сброса по выводу BRK 0 – запрещен сброс 1 – разрешен
11…9	OCCM[2:0]	Формат выработки сигнала REF в режиме ШИМ Если CCR1_EN = 0: 000 – всегда 0 001 – 1, если CNT = CCR 010 – 0, если CNT = CCR 011 – переключение REF, если CNT =CCR 100 – всегда 0 101 – всегда 1 110 – 1, если DIR= 0 (счет вверх), CNT<CCR, иначе 0 0, если DIR= 1 (счет вниз), CNT<CCR, иначе 1 111 – 0, если DIR= 0 (счет вверх), CNT<CCR, иначе 1 1, если DIR= 1 (счет вниз), CNT<CCR, иначе 0 Если CCR1_EN = 1: 000 – всегда 0; 001 – 1, если CNT = CCR или CNT = CCR1; 010 – 0, если CNT = CCR или CNT = CCR1; 011 – переключение REF, если CNT =CCR или CNT =CCR1; 100 – всегда 0; 101 – всегда 1; 110 – 1, если DIR = 1 (счет вниз), CCR1 < CNT < CCR, иначе 0; 0, если DIR= 0 (счет вверх), CCR < CNT < CCR1, иначе 1; 111 – 0, если DIR = 1 (счет вниз), CCR1 < CNT < CCR, иначе 1; 1, если DIR = 0 (счет вверх), CCR< CNT< CCR1, иначе 0;
8	OCCE	Разрешение работы ETR 0 – запрет ETR 1 – разрешение ETR
7…6	CHPSC[1:0]	Предварительный делитель входного канала 00 – нет деления 01 – /2 10 – /4 11 – /8
5…4	CHSEL[1:0]	Выбор события по входному каналу 00 – положительный фронт 01 – отрицательный фронт 10 – положительный фронт от других каналов: Для первого канала от 2 канала Для второго канала от 3 канала Для третьего канала от 4 канала Для четвертого канала от 1 канала

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
		11 – положительный фронт от других каналов: Для первого канала от 3 канала Для второго канала от 4 канала Для третьего канала от 1 канала Для четвертого канала от 2 канала
3…0	CHFLTR[3:0]	Сигнал зафиксирован: 0000 – в 1 триггере на частоте TIM_CLK 0001 – в 2 триггерах на частоте TIM_CLK 0010 – в 4 триггерах на частоте TIM_CLK 0011 – в 8 триггерах на частоте TIM_CLK 0100 – в 6 триггерах на частоте FDTS/2 0101 – в 8 триггерах на частоте FDTS/2 0110 – в 6 триггерах на частоте FDTS/4 0111 – в 8 триггерах на частоте FDTS/4 1000 – в 6 триггерах на частоте FDTS/8 1001 – в 8 триггерах на частоте FDTS/8 1010 – в 5 триггерах на частоте FDTS/16 1011 – в 6 триггерах на частоте FDTS/16 1100 – в 8 триггерах на частоте FDTS/16 1101 – в 5 триггерах на частоте FDTS/32 1110 – в 6 триггерах на частоте FDTS/32 1111 – в 8 триггерах на частоте FDTS/32

10.7.8 CHy_CNTRL1

Таблица 54 – Регистр управления 1 для ‘y’ канала таймера Chy_CNTRL1

Номер	31…13	12	11…10	9…8	7…5	4	3…2	1…0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	00	00	0	0	00	00
	-	**NINV**	**NSELO [1:0]**	**NSELOE [1:0]**	-	**INV**	**SELO [1:0]**	**SELOE [1:0]**

Таблица 55 – Описание бит регистра Chy_CNTRL1

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...13	-	Зарезервировано
12	NINV	Режим выходной инверсии инверсного канала 0 – выход не инвертируется 1 – выход инвертируется
11...10	NSELO[1:0]	Режим работы выхода инверсного канала 00 – всегда на выход выдается 0, канал на выход не работает 01 – всегда на выход выдается 1, канал всегда работает на выход 10 – на выход выдается сигнал REF 11 – на выход выдается сигнал с DTG
9…8	NSELOE[1:0]	Режим работы инверсного канала на выход 00 – всегда на OE выдается 0, канал на выход не работает 01 – всегда на OE выдается 1, канал всегда работает на выход 10 – на OE выдается сигнал REF, при REF = 0 вход, при REF = 1 выход. 11 – на OE выдается сигнал с DTG, при CHn = 0 вход, при CHn = 1 выход
7…5	-	Зарезервировано

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
4	INV	Режим выходной инверсии прямого канала 0 – выход не инвертируется 1 – выход инвертируется
3…2	SELO[1:0]	Режим работы выхода прямого канала 00 – всегда на выход выдается 0, канал на выход не работает 01 – всегда на выход выдается 1, канал всегда работает на выход 10 – на выход выдается сигнал REF. 11 – на выход выдается сигнал с DTG.
1…0	SELOE[1:0]	Режим работы прямого канала на выход 00 – всегда на OE выдается 0, канал на выход не работает 01 – всегда на OE выдается 1, канал всегда работает на выход 10 – на OE выдается сигнал REF, при REF = 0 вход, при REF = 1 выход. 11 – на OE выдается сигнал с DTG, при CH = 0 вход, при CH = 1 выход

10.7.9 CHy_CNTRL2

Таблица 56 – Регистр управления 2 для ‘y’ канала таймера Chy_CNTRL2

Номер	31…4	3	2	1…0
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	00
	-	**CRRRLD**	**CCR1_EN**	**CHSEL[1:0]**

Таблица 57 – Описание бит регистра Chy_CNTRL2

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...4	-	Зарезервировано
3	CRRRLD	Разрешение обновления регистров CCR и CCR1 0 – обновление возможно в любой момент времени 1 – обновление будет осуществлено только при CNT = 0
2	CCR1_EN	Разрешение работы регистра CCR1 0 – CCR1 не используется 1 – CCR1 используется
1…0	CHSEL1[1:0]	Выбор события по входному каналу для CAP1 00 – положительный фронт по Chi 01 – отрицательный фронт по Chi 10 – отрицательный фронт от других каналов Для первого канала от 2 канала Для второго канала от 3 канала Для третьего канала от 4 канала Для четвертого канала от 1 канала 11 – отрицательный фронт от других каналов Для первого канала от 3 канала Для второго канала от 4 канала Для третьего канала от 1 канала Для четвертого канала от 2 канала

10.7.10 CHy_DTG

Таблица 58 – Регистр CHy_DTG управления DTG

Номер	31..16	15…8	7…5	4	3…0
Доступ	U	R/W	U	R/W	R/W
Сброс	0	00000000	000	0	0000
	-	**DTG[7:0]**	-	**EDTS**	**DTGx [3:0]**

Таблица 59 – Описание бит регистра CHy_DTG

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...16	-	Зарезервировано
15…8	DTGx[7:0]	Основной делитель частоты Задержка DTGdel = DTGx*(DTG+1).
7…5	-	Зарезервировано
4	EDTS	Частота работы DTG 0 – TIM_CLK 1 – FDTS
3…0	DTG [3:0]	Предварительный делитель частоты DTG

10.7.11 BRKETR_CNTRL

Таблица 60 – Регистр BRKETR_CNTRL управления входом BRK и ETR

Номер	31…8	7…4	3…2	1	0
Доступ	U	R/W	R/W	R/W	R/W
Сброс		0000	00	0	0
	-	**ETR FILTER [3:0]**	**ETR PSC [1:0]**	**ETR INV**	**BRK INV**

Таблица 61 – Описание бит регистра BRKETR_CNTRL

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...8	-	Зарезервировано
7…4	ETR FILTER[3:0]	Цифровой фильтр на входе ETR. Сигнал зафиксирован: 0000 – в 1 триггере на частоте TIM_CLK 0001 – в 2 триггерах на частоте TIM_CLK 0010 – в 4 триггерах на частоте TIM_CLK 0011 – в 8 триггерах на частоте TIM_CLK 0100 – в 6 триггерах на частоте FDTS/2 0101 – в 8 триггерах на частоте FDTS/2 0110 – в 6 триггерах на частоте FDTS/4 0111 – в 8 триггерах на частоте FDTS/4 1000 – в 6 триггерах на частоте FDTS/8 1001 – в 8 триггерах на частоте FDTS/8 1010 – в 5 триггерах на частоте FDTS/16 1011 – в 6 триггерах на частоте FDTS/16 1100 – в 8 триггерах на частоте FDTS/16 1101 – в 5 триггерах на частоте FDTS/32 1110 – в 6 триггерах на частоте FDTS/32 1111 – в 8 триггерах на частоте FDTS/32

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
3…2	ETRPSC[1:0]	Асинхронный предделитель внешней частоты 00 – без деления 01 – /2 10 – /4 11 – /8
1	ETR INV	Инверсия входа ETR 0 – без инверсии 1 – инверсия
0	BRK INV	Инверсия входа BRK 0 – без инверсии 1 – инверсия

10.7.12 STATUS

Таблица 62 – Регистр статуса таймера STATUS

Номер	31...17	16…13	12…9	8…5
Доступ	U	U	R/W	R/W
Сброс	0	0	0	0
	-	**CCR CAP1 EVENT [3:0]**	**CCR REF EVENT [3:0]**	**CCR CAP EVENT [3:0]**

Номер	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	BRK EVENT	**ETR FE EVENT**	**ETR RE EVENT**	**CNT ARR EVENT**	**CNT ZERO EVENT**

Таблица 63 – Описание бит регистра STATUS

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…17	-	Зарезервировано
16…13	CCR CAP1 EVENT[3:0]	Событие переднего фронта на входе CAP1 каналов таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
12…9	CCR REF EVENT[3:0]	Событие переднего фронта на выходе REF каналов таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
8…5	CCR CAP EVENT[3:0]	Событие переднего фронта на входе CAP каналов таймера 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события. Бит 0 – первый канал Бит 3 – четвертый канал
4	BRK EVENT	Триггерированное по PCLK состояние входа BRK, 0 – BRK == 0 1 – BRK == 1 Сбрасывается записью 0, при условии наличия 0 на входе BRK
3	ETR FE EVENT	Событие заднего фронта на входе ETR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.
2	ETR RE EVENT	Событие переднего фронта на входе ETR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием, приоритет у нового события.
1	CNT ARR EVENT	Событие совпадения CNT с ARR 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса CNT и ARR не изменили состояния, то флаг повторно не взводится.
0	CNT ZERO EVENT	Событие совпадения CNT с нулем 0 – нет события 1 – есть событие Сбрасывается записью 0, если запись одновременно с новым событием совпадения, приоритет у нового события. Если с момента совпадения до момента программного сброса CNT не изменил состояния, то флаг повторно не взводится.

10.7.13 IE

Таблица 64 – Регистр разрешения прерывания таймера IE

Номер	31...17	16…13	12…9	8…5
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-	CCR CAP1 EVENT IE [3:0]	CCR REF EVENT IE [3:0]	CCR CAP EVENT IE [3:0]

Номер	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	BRK EVENT IE	**ETR FE EVENT IE**	**ETR RE EVENT IE**	**CNT ARR EVENT IE**	**CNT ZERO EVENT IE**

Таблица 65 – Описание бит регистра IE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31…17	-	Зарезервировано
16...13	CCR CAP1 EVENT IE [3:0]	Флаг разрешения прерывания по событию переднего фронта на выходе CAP1 каналов таймера 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
12…9	CCR REF EVENT IE[3:0]	Флаг разрешения прерывания по событию переднего фронта на выходе REF каналов таймера 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
8…5	CCR CAP EVENT IE [3:0]	Флаг разрешения прерывания по событию переднего фронта на выходе CAP каналов таймера 0 – нет прерывания 1 – прерывание разрешено Бит 0 – первый канал Бит 3 – четвертый канал
4	BRK EVENT IE	Флаг разрешения по триггерированному по PCLK состоянию входа BRK, 0 – нет прерывания 1 – прерывание разрешено
3	ETR FE EVENT IE	Флаг разрешения прерывания по заднему фронту на входе ETR 0 – нет прерывания 1 – прерывание разрешено
2	ETR RE EVENT IE	Флаг разрешения прерывания по переднему фронту на входе ETR 0 – нет прерывания 1 – прерывание разрешено
1	CNT ARR EVENT IE	Флаг разрешения прерывания по событию совпадения CNT и ARR 0 – нет прерывания 1 – прерывание разрешено
0	CNT ZERO EVENT IE	Флаг разрешения прерывания по событию совпадения CNT и нуля 0 – нет прерывания 1 – прерывание разрешено

10.7.14 DMA_RE

Таблица 66 – Регистр DMA_RE разрешения запросов DMA от прерываний таймера

Номер	31...17	16…13	12…9	8…5
Доступ	U	R/W	R/W	R/W
Сброс	0	0	0	0
	-	**CCR CAP1 EVENT RE[3:0]**	**CCR REF EVENT RE[3:0]**	**CCR CAP EVENT RE[3:0]**

Номер	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	BRK EVENT RE	**ETR FE EVENT RE**	**ETR RE EVENT RE**	**CNT ARR EVENT RE**	**CNT ZERO EVENT RE**

Таблица 67 – Описание бит регистра DMA_RE

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...17	-	Зарезервировано
16…13	CCR CAP1 EVENT RE [3:0]	Флаг разрешения запроса DMA по событию переднего фронта на выходе CAP1 каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
12…9	CCR REF EVENT RE[3:0]	Флаг разрешения запроса DMA по событию переднего фронта на выходе REF каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
8…5	CCR CAP EVENT RE [3:0]	Флаг разрешения запроса DMA по событию переднего фронта на выходе CAP каналов таймера 0 – нет запроса DMA 1 – запрос DMA разрешен Бит 0 – первый канал Бит 3 – четвертый канал
4	BRK EVENT RE	Флаг разрешения по триггерированному по PCLK состоянию входа BRK, 0 – нет запроса DMA 1 – запрос DMA разрешен
3	ETR FE EVENT RE	Флаг разрешения запроса DMA по заднему фронту на входе ETR 0 – нет запроса DMA 1 – запрос DMA разрешен
2	ETR RE EVENT RE	Флаг разрешения запроса DMA по переднему фронту на входе ETR 0 – нет запроса DMA 1 – запрос DMA разрешен

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
1	CNT ARR EVENT RE	Флаг разрешения запроса DMA по событию совпадения CNT и ARR 0 – нет запроса DMA 1 – запрос DMA разрешен
0	CNT ZERO EVENT RE	Флаг разрешения запроса DMA по событию совпадения CNT и нуля 0 – нет запроса DMA 1 – запрос DMA разрешен

11 Порты общего назначения GPIO

Большинство внешних выводов процессора представляют собой универсальные входы-выходы, которые могут быть перепрограммированы пользователем. Каждый бит портов GPIO программируется индивидуально.

Процессор имеет несколько портов ввода-вывода общего назначения. Часть из них имеет альтернативные функции ввода-вывода и может быть использована различными интерфейсами процессора. Процессор имеет 128 выводов общего назначения, которые могут быть индивидуально сконфигурированы как вход или выход. Также дополнительно имеется 22 вывода общего назначения, которые всегда сконфигурированы как выход.

11.1 Порты PA, PB, PC

Специальный модуль GPIO объединяет три самостоятельных 32-разрядных порта. Регистры этого модуля подключены к шине периферийных устройств и имеют базовый адрес 0x80001000 (порт A), адрес 0x80001040 (порт B), адрес 0x80001080 (порт C). Некоторые регистры имеют четыре адреса для работы: загрузки, установки, сброса и инверсии бит (суффиксы _LD, _SET, _CLR и _INV, соответственно). Краткое описание регистров порта приведено ниже (Таблица 68). После сброса значения регистров PiDDR, PiIMR, PiALT и PiPUR равно 0. Значения всех остальных регистров после сброса не определено.

Таблица 68 – Регистры управление портами PA, PB, PC

Номер	Имя	Тип	Назначение
0	PiDR_LD	R/W	Регистр данных для выдачи информации. Загрузка значения
1	PiDR_SET	R/W	Регистр данных для выдачи информации. 1 - установка регистра в 1 0 - начение регистра не меняется Чтение регистра возвращает значение регистра PIDR
2	PiDR_CLR	R/W	Регистр данных для выдачи информации. 1 - установка регистра в 0 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PIDR
3	PiDR_INV	R/W	Регистр данных для выдачи информации. 1 - инверсия содержимого регистра 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PIDR
4	PiDDR_LD	R/W	Регистр направления выдачи информации. Загрузка значения 0 – прием; 1 – выдача.
5	PiDDR_SET	R/W	Регистр направления выдачи информации: 1 - установка регистра в 1 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PIDDR
6	PiDDR_CLR	R/W	Регистр направления выдачи информации: 1 - установка регистра в 0 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PIDDR

Номер	Имя	Тип	Назначение
7	PiDDR_INV	R/W	Регистр направления выдачи информации: 1 - инверсия содержимого регистра 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PIDDR
8	PiPEIE_LD	R/W	Регистр разрешения прерывания по положительному (из 0 в 1) изменению внешней линии порта. Загрузка значения 1 - прерывание по изменению из 0 в 1 разрешено 0 - прерывание по изменению из 0 в 1 запрещено
9	PiPEIE_SET	R/W	Регистр разрешения прерывания по положительному (из 0 в 1) изменению внешней линии порта. 1 - установка регистра в 1 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiPEIE
10	PiPEIE_CLR	R/W	Регистр разрешения прерывания по положительному (из 0 в 1) изменению внешней линии порта. 1 - установка регистра в 0 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiPEIE
11	PiPEIE_INV	R/W	Регистр разрешения прерывания по положительному (из 0 в 1) изменению внешней линии порта. 1 - инверсия содержимого регистра 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiPEIE
12	PiNEIE_LD	R/W	Регистр разрешения прерывания по отрицательному (из 1 в 0) изменению внешней линии порта. Загрузка значения 1 – прерывание по изменению из 1 в 0 разрешено 0 – прерывание по изменению из 1 в 0 запрещено
13	PiNEIE_SET	R/W	Регистр разрешения прерывания по отрицательному (из 1 в 0) изменению внешней линии порта. 1 - установка регистра в 1 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiNEIE
14	PiNEIE_CLR	R/W	Регистр разрешения прерывания по отрицательному (из 1 в 0) изменению внешней линии порта. 1 - установка регистра в 0 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiNEIE
15	PiNEIE_INV	R/W	Регистр разрешения прерывания по отрицательному (из 1 в 0) изменению внешней линии порта. 1 - инверсия содержимого регистра 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiNEIE
16	PiINVR_LD	R/W	Регистр инверсного значения внешней линии порта, при котором регистрируется прерывание. Загрузка значения 1 – прерывание регистрируется при значении внешней линии 0 0 – прерывание регистрируется при значении внешней линии 1
17	PiINVR_SET	R/W	Регистр инверсного значения внешней линии порта, при котором регистрируется прерывание. 1 - установка регистра в 1 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiNVR

Номер	Имя	Тип	Назначение
18	PiINVR_CLR	R/W	Регистр инверсного значения внешней линии порта, при котором регистрируется прерывание. 1 - установка регистра в 0 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiNVR
19	PiINVR_INV	R/W	Регистр инверсного значения внешней линии порта, при котором регистрируется прерывание. 1 - инверсия содержимого регистра 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiNVR
20	PiIMR_LD	R/W	Регистр маскирования прерывания внешней линии порта. Загрузка значения 1 – прерывание разрешено 0 – прерывание запрещено
21	PiIMR_SET	R/W	Регистр маскирования прерывания внешней линии порта. 1 - установка регистра в 1 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiIMR
22	PiIMR_CLR	R/W	Регистр маскирования прерывания внешней линии порта. 1 - установка регистра в 0 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiIMR
23	PiIMR_INV	R/W	Регистр маскирования прерывания внешней линии порта. 1 - инверсия содержимого регистра 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiIMR
24	PiALT_LD	R/W	Регистр выбора функции порта. Загрузка значения 1 - альтернативная функция вывода включена 0 - вывод общего назначения
25	PiALT _SET	R/W	Выбор функции порта. 1 - установка регистра в 1 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiALT
26	PiALT _CLR	R/W	Регистр выбора функции порта. 1 - установка регистра в 0 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiALT
27	PiALT _INV	R/W	Регистр выбора функции порта. 1 - инверсия содержимого регистра 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiALT
28	PiPUR_LD	R/W	Регистр включения резистора подтяжки «к питанию» или «к земле» на внешней линии порта. Загрузка значения 0 – подтяжка включена 1 – подтяжка выключена
29	PiPUR_SET	R/W	Регистр включения резистора подтяжки «к питанию» или «к земле» на внешней линии порта. 1 - установка регистра в 1 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiPUR

Номер	Имя	Тип	Назначение
30	PiPUR_CLR	R/W	Регистр включения резистора подтяжки «к питанию» или «к земле» на внешней линии порта. 1 - установка регистра в 0 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiPUR
31	PiPUR_INV	R/W	Регистр включения резистора подтяжки «к питанию» или «к земле» на внешней линии порта. 1 - инверсия содержимого регистра 0 - значение регистра не меняется Чтение регистра возвращает значение регистра PiPUR
32	PiPXD	R	Значение внешней линии порта
33	PiINTREQ	R	Запросы прерывания по входам порта
34	-	-	Зарезервировано
35	PiECLR	R/W	Сброс запросов прерываний по фронту (запись 1 вызывает сброс)
Примечание – Порты A, B и C имеют идентичные регистры. Символом “i” в имени регистра обозначено одно из трех значений: A, B, C			

Каждая линия 32-разрядного порта может быть сконфигурирована как выход или как вход, может сгенерировать прерывание по положительному или/и отрицательному фронту сигнала или по уровню (выбор уровня задается битом инверсии входа). Все запросы прерываний собираются в один общий запрос, который поступает в контроллер прерывания.

Каждый бит порта может иметь альтернативную функцию. Если бит регистра PiALT имеет нулевое значение, соответствующая линия порта работает под управлением регистров порта. Если бит равен 1, линия порта переходит под управление заданного устройства процессора. Возможные альтернативные функции портов приведены в таблице описания выводов.

Виды прерываний от внешних линий портов разрешаются/запрещаются независимо регистрами PiPEIE и PiNEIE. Если в любом из регистров PiPEIE или PiNEIE установлена 1, будут разрешены соответствующее прерывание по переходу из 0 в 1 (фронт), или из 1 в 0 (срез), или оба. Разрешенное прерывание по фронту или срезу запрещает прерывание по уровню. Итоговое прерывание может быть заблокировано значением 0 бит регистра маски PiIMR.

11.2 Порты PE и PD

Порты PE и PD предназначены для организации интерфейса к внешней памяти. Если нет необходимости подключать внешнюю память к данному интерфейсу, тогда шина адреса и шина данных могут быть использованы как порты общего назначения. При этом порт PE представляет собой 22-разрядный однонаправленный выходной порт, а порт PD – 32-разрядный двунаправленный порт общего назначения. Порт PE имеет только регистр данных. Порт PD имеет регистр данных и регистр направления. Регистры портов подключены к шине обмена периферийных устройств и имеют базовый адрес 0x8000_0090. Регистры портов представлены ниже (Таблица 69).

Таблица 69 – Регистры управления портами PE, PD

Номер	Имя	Тип	Описание
0	PD_LOAD	R/W	Регистр данных порта PD. Загрузка всех 32 бит

Номер	Имя	Тип	Описание
1	PD_SET	R/W	Регистр данных порта PD. 1 – установка регистра в 1 0 – значение регистра не меняется Чтение регистра возвращает значение регистра PD
2	PD_CLEAR	R/W	Регистр данных порта PD. 1 – установка регистра в 0 0 – значение регистра не меняется Чтение регистра возвращает значение регистра PD
3	PD_DIR	R/W	Регистр направления порта PD. 1 – выдача данных 0 – прием данных
4	PE_LOAD	R/W	Регистр данных порта PE. Загрузка всех 22 бит
5	PE_SET	R/W	Регистр данных порта PE. 1 – установка регистра в 1 0 – значение регистра не меняется Чтение регистра возвращает значение регистра PE
6	PE_CLEAR	R/W	Регистр данных порта PE. 1 – установка регистра в 0 0 – значение регистра не меняется Чтение регистра возвращает значение регистра PE
7	-		Зарезервировано
8	PX_ALT	R/W	Управление альтернативными функциями портов PE и PD 1 – альтернативная функция включена 0 – альтернативная функция выключена
9	PD_PIN	R	Чтение внешних контактов порта PD
10	PD_PU	R/W	Управление резисторами доопределения до питания внешних выводов PD[31:0] 0 – резистор подключен 1 – резистор отключен

11.3 Регистр PX_ALT

Данный регистр задает различные конфигурации для выводов портов PE, PD. Назначение бит регистра приведено ниже (Таблица 70).

Таблица 70 – Регистр управления альтернативными функциями

Бит	Имя	Тип	Назначение
0	PDB[0]	R/W	1 – порт PD[7:0] находится под управлением контроллера внешней памяти. 0 – порт PD[7:0] находится под управлением PD_LOAD[7:0], PD_SET[7:0], PD_CLEAR[7:0] и PD_DIR[7:0], т.е. порт общего назначения.
1	PDB[1]	R/W	1 – порт PD[15:8] находится под управлением контроллера внешней памяти. 0 – порт PD[15:8] находится под управлением PD_LOAD[15:8], PD_SET[15:8], PD_CLEAR[15:8] и PD_DIR[15:8], т.е. порт общего назначения.
2	PDB[2]	R/W	1 – порт PD[23:16] находится под управлением контроллера внешней памяти. 0 – порт PD[23:16] находится под управлением PD_LOAD[23:16], PD_SET[23:16], PD_CLEAR[23:16] и PD_DIR[23:16], т.е. порт общего назначения.

Бит	Имя	Тип	Назначение
3	PDB[3]	R/W	1 – порт PD[31:24] находится под управлением контроллера внешней памяти. 0 – порт PD[31:24] находится под управлением PD_LOAD[31:24], PD_SET[31:24], PD_CLEAR[31:24] и PD_DIR[31:24], т.е. порт общего назначения.
4	PEB[0]	R/W	1 – порт PE[7:0] находится под управлением внешнего контроллера внешней памяти. 0 – порт PE[7:0] находится под управлением PE_LOAD[7:0], PE_SET[7:0], PE_CLEAR[7:0], т.е. порт общего назначения.
5	PEB[1]	R/W	1 – порт PE[15:8] находится под управлением внешнего контроллера внешней памяти. 0 – порт PE[15:8] находится под управлением PE_LOAD[15:8], PE_SET[15:8], PE_CLEAR[15:8], т.е. порт общего назначения.
6	PEB[2]	R/W	1 – порт PE[21:16] находится под управлением внешнего контроллера внешней памяти. 0 – порт PE[21:16] находится под управлением PE_LOAD[21:16], PE_SET[2:16], PE_CLEAR[21:16], т.е. порт общего назначения.
7	PDXF0	R/W	Управление подключением периферийных устройств (Таблица 71)
8	PDXF1	R/W	Управление подключением периферийных устройств (Таблица 71)
31-9	-	-	Не используются

Конфигурация для разрядов с 0-го по 31-й порта PD следующая: если биты PDB[3:0] равны нулю – это порт общего назначения (каждый байт порта управляется своим битом), а если какой-то бит PDB[3:0] равен 1 – соответствующие ему разряды порта выполняют альтернативную функцию. При этом для двух младших байт (биты с 0-го по 15-й) альтернативной функцией является только функция внешней шины данных для подключения внешней памяти. *Биты PDB[2] и PDB[3] управляют различными данными, но их значение всегда должно быть одинаковыми (два нуля или две единицы).*

Функции для разрядов с 16-го по 31-й порта PD более сложные и дополнительно зависят от битов 7 и 8 регистра PX_ALT. Возможные конфигурации для старших 16 битов приведены ниже (Таблица 71).

Таблица 71 – Функции порта PD[31:16] (биты PDB[3:2]=11b)

Номер	PX_ALT[8:7]	Функция
16-23	x0	Шина данных внешней памяти. Вход-выход Биты с 16-го по 23-й
	01	NAND флэш-память. Шина данных. Вход-выход Биты с 0-го по 7-й
	11	МКПД интерфейс Входы – 16, 17, 21, 22 Выходы – 18, 19, 20, 23
24-30	x0	Шина данных внешней памяти. Вход-выход Биты с 24-го по 30-й
	01	NAND флэш-память. Шина управления Выходы
	11	МКПД интерфейс Выходы – 24, 25
31	x0	Шина данных внешней памяти. Вход-выход Бит 31-й
	x1	Вход Для контроллера NAND флэш-памяти – вход готовности

После сброса регистр альтернативных функций равен нулю и порты PE, PD являются портами общего назначения. При этом состояние портов следующее:

- порт PD находится в отключенном состоянии (вход);
- значение порта PE равно 0.

При необходимости использования внешней памяти необходимо записать в регистр PX_ALT нужную конфигурацию (задать разрядность шины данных и шины адреса).

При необходимости подключения внешней NAND памяти или при использовании интерфейса МКПД, также в регистре PX_ALT необходимо задать соответствующую конфигурацию. При использовании данных контроллеров шина данных может иметь максимальную разрядность 16 бит.

Включение режима 16-разрядной внешней шины данных происходит посредством установки бита SYSCON[19] регистра управления. Установка данного бита оказывает влияние только на контроллер внешней шины, т.к. в случае 16-разрядной шины контроллер будет выполнять два последовательных чтения (или две записи) при обращении к внешней памяти. При этом конфигурация шины данных и шины адреса не зависит от бита SYSCON[19]. Можно, например, использовать только восемь младших разрядов шины данных и работать с ними как с 32-разрядной шиной.

11.4 Регистр FLAGREG

В архитектуре вычислительного ядра процессора предусмотрен специальный регистр управления флагами FLAGREG, который осуществляет управление специальными внешними контактами. Если для бит с 0-го по 3-й порта PC задать альтернативную функцию, выводы PC[3:0] переходят под управление регистра FLAGREG и начинают выполнять функции флагов FLAG3–0.

Каждый контакт может быть сконфигурирован как вход или выход индивидуально. При конфигурации как выход, программы могут использовать выводы флагов для сигналов событий или условий для любого внешнего устройства. При конфигурации как входы, программы могут считывать состояние внешних контактов, используя регистр FLAGREG или использовать значение входа как условие в условных командах.

Для подготовки и использования флагов следует использовать следующую процедуру:

1. С помощью битов FLAGx_EN регистра FLAGREG сконфигурируйте каждый из выводов FLAG3–0 как вход (FLAGx_EN=0) или выход (FLAGx_EN=1).
2. С помощью битов FLAGx_OUT регистра FLAGREG выберите значения выхода для каждого вывода FLAG3–0 (для тех контактов, которые сконфигурированы как выход).
3. Установите в единицу биты с 0-го по 3-й в регистре альтернативных функций порта PC. С этого момента выводы PC[3:0] переходят под управление регистра флагов.
4. Прочитайте биты FLGx в регистре SQSTAT для определения входного значения для каждого вывода FLAG3–0, или используйте входные значения флагов (FLAGx_IN) в условных командах.

Выводы флагов (FLAG3–0) позволяют процессору посылать сигналы внешним устройствам или получать входные сигналы от них. Отметим, что аналогичные функции по управлению внешними флагами можно реализовать и при помощи

регистров порта PC. Отличие будет только в том, что управление с помощью регистра FLAGREG будет происходить быстрее, т.к. регистр FLAGREG относится к группе регистров устройства управления и тактируется клоком ядра CCLK.

Таблица 72 – Альтернативные функции портов PE, PD

Обозначение				Тип	Выполняемая функция
Функция 1	**Функция 2**	**Функция 3**	**GPIO**		
XTO			PE[22]	O	Синхронизация, выход для подключения резонатора
A[21]			PE[21]	O	Шина адреса внешнего интерфейса
A[20]			PE[20]	O	Шина адреса внешнего интерфейса
A[19]			PE[19]	O	Шина адреса внешнего интерфейса
A[18]			PE[18]	O	Шина адреса внешнего интерфейса
A[17]			PE[17]	O	Шина адреса внешнего интерфейса
A[16]			PE[16]	O	Шина адреса внешнего интерфейса
A[15]			PE[15]	O	Шина адреса внешнего интерфейса
A[14]			PE[14]	O	Шина адреса внешнего интерфейса
A[13]			PE[13]	O	Шина адреса внешнего интерфейса
A[12]			PE[12]	O	Шина адреса внешнего интерфейса
A[11]			PE[11]	O	Шина адреса внешнего интерфейса
A[10]			PE[10]	O	Шина адреса внешнего интерфейса
A[9]			PE[9]	O	Шина адреса внешнего интерфейса
A[8]			PE[8]	O	Шина адреса внешнего интерфейса
A[7]			PE[7]	O	Шина адреса внешнего интерфейса
A[6]			PE[6]	O	Шина адреса внешнего интерфейса
A[5]			PE[5]	O	Шина адреса внешнего интерфейса
A[4]			PE[4]	O	Шина адреса внешнего интерфейса
A[3]			PE[3]	O	Шина адреса внешнего интерфейса
A[2]			PE[2]	O	Шина адреса внешнего интерфейса
A[1]			PE[1]	O	Шина адреса внешнего интерфейса
A[0]			PE[0]	O	Шина адреса внешнего интерфейса
DATA[31]	NF_RDY	MIL2_RTAP	PD[31]	I/O	Шина данных внешнего интерфейса
DATA[30]	NF_CS[2]	MIL2_RTA4	PD[30]	I/O	Шина данных внешнего интерфейса
DATA[29]	NF_CS[1]	MIL2_RTA3	PD[29]	I/O	Шина данных внешнего интерфейса
DATA[28]	NF_CS[0]	MIL2_RTA2	PD[28]	I/O	Шина данных внешнего интерфейса
DATA[27]	NF_WE	MIL2_RTA1	PD[27]	I/O	Шина данных внешнего интерфейса
DATA[26]	NF_RE	MIL2_RTA0	PD[26]	I/O	Шина данных внешнего интерфейса
DATA[25]	NF_ALE	MIL2_OU2X	PD[25]	I/O	Шина данных внешнего интерфейса
DATA[24]	NF_CLE	MIL2_OU2N	PD[24]	I/O	Шина данных внешнего интерфейса
DATA[23]	NF_D[7]	MIL2_OU2P	PD[23]	I/O	Шина данных внешнего интерфейса
DATA[22]	NF_D[6]	MIL2_IN2N	PD[22]	I/O	Шина данных внешнего интерфейса
DATA[21]	NF_D[5]	MIL2_IN2P	PD[21]	I/O	Шина данных внешнего интерфейса
DATA[20]	NF_D[4]	MIL2_OU1X	PD[20]	I/O	Шина данных внешнего интерфейса
DATA[19]	NF_D[3]	MIL2_OU1N	PD[19]	I/O	Шина данных внешнего интерфейса
DATA[18]	NF_D[2]	MIL2_OU1P	PD[18]	I/O	Шина данных внешнего интерфейса
DATA[17]	NF_D[1]	MIL2_IN1N	PD[17]	I/O	Шина данных внешнего интерфейса
DATA[16]	NF_D[0]	MIL2_IN1P	PD[16]	I/O	Шина данных внешнего интерфейса
DATA[15]			PD[15]	I/O	Шина данных внешнего интерфейса
DATA[14]			PD[14]	I/O	Шина данных внешнего интерфейса
DATA[13]			PD[13]	I/O	Шина данных внешнего интерфейса
DATA[12]			PD[12]	I/O	Шина данных внешнего интерфейса
DATA[11]			PD[11]	I/O	Шина данных внешнего интерфейса
DATA[10]			PD[10]	I/O	Шина данных внешнего интерфейса
DATA[9]			PD[9]	I/O	Шина данных внешнего интерфейса
DATA[8]			PD[8]	I/O	Шина данных внешнего интерфейса
DATA[7]			PD[7]	I/O	Шина данных внешнего интерфейса

Обозначение				Тип	Выполняемая функция
Функция 1	**Функция 2**	**Функция 3**	**GPIO**		
DATA[6]			PD[6]	I/O	Шина данных внешнего интерфейса
DATA[5]			PD[5]	I/O	Шина данных внешнего интерфейса
DATA[4]			PD[4]	I/O	Шина данных внешнего интерфейса
DATA[3]			PD[3]	I/O	Шина данных внешнего интерфейса
DATA[2]			PD[2]	I/O	Шина данных внешнего интерфейса
DATA[1]			PD[1]	I/O	Шина данных внешнего интерфейса
DATA[0]			PD[0]	I/O	Шина данных внешнего интерфейса

12 Прерывания

Процессор имеет контроллер прерываний, который принимает и обслуживает запросы от внешних портов и от внутренних периферийных устройств. Возможно задание векторных прерываний и программных прерываний, задаваемых пользователем. Прерывания могут быть вложенными. При обработке прерывания выполняется переход по вектору прерывания, значение которого хранится в таблице векторов прерываний.

Процессор поддерживает обработку прерываний, являющихся индикаторами различных внешних или внутренних событий. Прерывания делятся на программные и аппаратные.

Для каждого запроса прерывания существует:

- регистр в таблице векторов прерываний (группы регистров 0x38 и 0x39);
- бит в регистре защелки прерывания ILAT;
- бит в регистре маски прерывания IMASK;
- бит в регистре приоритета прерывания PMASK.

Номера битов в регистрах одинаковы и соответствуют номеру прерывания. Этот же номер имеет и регистр вектора прерывания в таблице векторов.

Программные прерывания вызываются специальными командами или определенными ситуациями, которые возникают при выполнении команд.

Аппаратные прерывания вызываются различными событиями, которые происходят в периферийных устройствах, или входным сигналом на выводах внешних прерываний.

Прерывание может быть вызвано аппаратными ошибками в процессоре, такими как:

- Ошибка в работе контроллера DMA;
- Ошибка порта связи.

Прерывания могут быть чувствительны к фронту или уровню сигнала:

- *Прерывания чувствительные к фронту* фиксируются при их появлении установкой определенного флага в регистре ILAT. Флаги хранятся до тех пор, пока не будут обслужены или сброшены по команде. Если флаг не сбрасывается во время обработки, новое событие не обнаруживается и запрос игнорируется.
- *Прерывания чувствительные к уровню* должны поддерживать активный уровень запроса до начала обработки, в противном случае они не видны. Если запрос продолжается после обработки, он считается новым запросом прерывания.

Чувствительные к уровню прерывания обычно появляются как некоторый активный результат в определенном доступном регистре и аннулируются чтением этого регистра или записью в него неактивного значения. Чувствительные к фронту прерывания инициируются событием (например, истечением таймера).

Каждое из прерываний процессора имеет регистр вектора прерываний, который является адресом процедуры, обслуживающей данное прерывание. Весь регистровый файл называется Таблица Вектора Прерываний (IVT). Каждый регистр таблицы является 32-битным для того, чтобы подключать процедуры обработки прерываний из внешней или внутренней памяти.

12.1 Группы регистров контроллера прерываний

Данные группы регистров относятся к векторам прерываний, управлению прерываниями и регистрам статуса (Таблица 73, Таблица 74, Таблица 75). Все регистры контроллера прерываний доступны только как однословные.

12.1.1 Группы регистров векторов прерываний

Данная группа регистров доступна как из поля команды пересылки, с использованием номера группы и номера регистра в группе, так и с помощью команд загрузки и сохранения по адресу в адресном пространстве периферийных устройств. Базовый адрес таблицы векторов прерываний в адресном пространстве периферийных устройств имеет значение 0x8000_0300. Смещение вектора по отношению к базовому адресу, а также номер регистра с учетом его группы приведены ниже (Таблица 73). Номер регистра образуется конкатенацией номера группы (старшие 6 бит) и номера регистра в группе (младшие 5 бит).

Таблица 73 – Группа А регистров таблицы вектора прерываний

Имя	Описание	Номер	Смещение
IVKERNEL	Прерывание ядра VDK	0x0700	0
IVGPIO	Прерывание от портов общего назначения PA, PB, PC	0x0701	1
IVTIMER0LP	Низкий приоритет таймера #0	0x0702	2
IVTIMER1LP	Низкий приоритет таймера #1	0x0703	3
IVUART0	UART 0	0x0704	4
IVUART1	UART 1	0x0705	5
IVLINK0	Порт связи #0	0x0706	6
IVLINK1	Порт связи #1	0x0707	7
-	Зарезервирован	0x0708	8
-	Зарезервирован	0x0709	9
IVNANDC	NAND флэш контроллер	0x070A	10
IVMIL0	МКПД 0	0x070B	11
IVMIL1	МКПД 1	0x070C	12
IVDIGC	Цифровой коррелятор	0x070D	13
IVDMA0	Регистр DMA #0	0x070E	14
IVDMA1	Регистр DMA #1	0x070F	15
IVDMA2	Регистр DMA #2	0x0710	16
IVDMA3	Регистр DMA #3	0x0711	17
IVARINC_RX	Приемник ARINC	0x0712	18
IVARINC_TX	Передатчик ARINC	0x0713	19
IVSPI1	Контроллер SPI1	0x0714	20
IVSPI2	Контроллер SPI2	0x0715	21
IVDMA4	Регистр DMA #4	0x0716	22
IVDMA5	Регистр DMA #5	0x0717	23
IVDMA6	Регистр DMA #6	0x0718	24
IVDMA7	Регистр DMA #7	0x0719	25
IVI2C	Контроллер I2C	0x071A	26
IVGTMR0	Таймер 0 с функцией Захвата/ШИМ	0x071B	27
IVGTMR1	Таймер 1 с функцией Захвата/ШИМ	0x071C	28
IVDMA8	Регистр DMA #8	0x071D	29
IVDMA9	Регистр DMA #9	0x071E	30
IVDMA10	Регистр DMA #10	0x071F	31

Таблица 74 – Группа B регистров таблицы вектора прерываний

Имя	Описание	Номер	Смещение
IVDMA11	Регистр DMA #11	0x0720	32
IVADDA0	UP\DOWN модуль 0	0x0721	33
IVADDA1	UP\DOWN модуль 1	0x0722	34
IVADDA2	UP\DOWN модуль 2	0x0723	35
IVADDA3	UP\DOWN модуль 3	0x0724	36
IVDMA12	Регистр DMA #12	0x0725	37
IVHOST	Хост-интерфейс	0x0726	38
IVLCD	LCD интерфейс	0x0727	39
-	Зарезервирован	0x0728	40
IVIRQ0	Вывод регистра IRQ0	0x0729	41
IVIRQ1	Вывод регистра IRQ1	0x072A	42
IVIRQ2	Вывод регистра IRQ2	0x072B	43
IVIRQ3	Вывод регистра IRQ3	0x072C	44
IVSPI	Контроллер SPI 0	0x072D	45
IVSSI0	Контроллер SSI 0	0x072E	46
IVSSI1	Контроллер SSI 1	0x072F	47
VIRPT	Векторный регистр VIRPT	0x0730	48
IVVCAM	Видеокамера	0x0731	49
IVBUSLK	Вектор блокировки шины	0x0732	50
IVH264	H264 декодер	0x0733	51
IVTIMER0HP	Высокий приоритет таймера 0	0x0734	52
IVTIMER1HP	Высокий приоритет таймера 1	0x0735	53
IVALARM	RTC ALARM	0x0736	54
IVTIC	RTC TIC	0x0737	55
IVDOG	RTC WATCHDOG	0x0738	56
IVHW	Аппаратная ошибка	0x0739	57
-	-	-	58-63

12.1.2 Группа регистров управления контроллером прерываний

В задачу регистров управления прерывания входит прием запроса на прерывание, его обработка, генерация номера прерывания для передачи в процессор. Список регистров приведен ниже (Таблица 75). Регистры доступны как по номеру, так и по адресу в пространстве адресов периферийных устройств.

Таблица 75 – Группа регистров управления прерываниями

Имя	Описание	Номер\Адрес	Значение после сброса
ILATL	ILAT, младшие разряды	0x0740 \0x8000_0340	0
ILATH	ILAT старшие разряды	0x0741 \0x8000_0341	0
ILATSTL	ILAT младшие разряды, установка	0x0742 \0x8000_0342	Не определено
ILATSTH	ILAT старшие разряды, установка	0x0743 \0x8000_0343	Не определено
ILATCLL	ILAT младшие разряды, сброс	0x0744 \0x8000_0344	Не определено
ILATCLH	ILAT старшие разряды, сброс	0x0745 \0x8000_0345	Не определено
PMASKL	PMASK младшие разряды	0x0746 \0x8000_0346	0
PMASKH	PMASK старшие разряды	0x0747\0x8000_0347	0

Имя	Описание	Номер\Адрес	Значение после сброса
IMASKL	IMASK младшие разряды	0x0748 \0x8000_0348	0
IMASKH	IMASK старшие разряды	0x0749\0x8000_0349	0
-	-	-	-
INTCTL	Управление прерыванием	0x074E\0x8000_034E	0

12.1.3 Регистр управления прерываниями (INTCTL)

Регистр INTCTL – 32-битный регистр, который управляет чувствительностью внешних входов прерываний IRQ3-0 (прерываний чувствительных к фронту или уровню) и обеспечивает управление стартом и остановкой таймеров. Подробное описание разрядов регистра приведено ниже (Таблица 76).

Таблица 76 – Регистр INTCTL

Бит	Имя	Назначение
0	IRQ0_EDGE	Запрос прерывания по входу IRQ0 вызывается: 0 – фронтом сигнала(из высокого в низкий); 1 – уровнем (низкий)
1	IRQ1_EDGE	Запрос прерывания по входу IRQ1 вызывается: 0 – фронтом сигнала(из высокого в низкий); 1 – уровнем (низкий)
2	IRQ2_EDGE	Запрос прерывания по входу IRQ2 вызывается: 0 – фронтом сигнала(из высокого в низкий); 1 – уровнем (низкий)
3	IRQ3_EDGE	Запрос прерывания по входу IRQ3 вызывается: 0 – фронтом сигнала(из высокого в низкий); 1 – уровнем (низкий)
4	TMR0RN	Разрешение работы таймера 0: 0 – таймер остановлен; 1 – таймер работает
5	TMR1RN	Разрешение работы таймера 1: 0 – таймер остановлен; 1 – таймер работает
31:6	-	Не используются и при чтении всегда равны нулю

Описание регистров таймера и способов работы с таймерами приведено в подразделе «Таймеры».

12.1.4 Регистры защелки прерываний (ILATL/ILATH)

Все запросы прерываний, которые поступают в контроллер прерываний, фиксируются (запоминаются) в регистре ILAT. Регистр ILAT – это 64-битный регистр доступный только для чтения как два 32-битных регистра ILATH и ILATL:

- ILATL – младшая часть ILAT (биты 31:0);
- ILATH – старшая часть ILAT (биты 63:32).

Каждый бит соответствует одному прерыванию и устанавливается при появлении этого прерывания. Порядок битов запросов прерываний устанавливается в соответствии с приоритетами прерываний: 0 – самый низкий приоритет.

Регистр ILAT имеет псевдонимы для упрощения процедур установки и сброса бит регистра. Для установки используются регистры ILATSTL или ILATSTH, а для сброса ILATCLL или ILATCLH. Регистры-псевдонимы доступны только по записи. При установке записываемое значение 1 приводит к установке бита, ноль не

изменяет значение. При сбросе запись нуля вызывает сброс, а запись единицы не изменяет значение.

12.1.5 Регистры маскирования прерываний (IMASKL/IMASKH)

Каждый бит запроса прерываний, зафиксированный в регистре ILAT, имеет соответствующий ему бит разрешения обслуживания прерывания (или бит маскирования прерывания). Значение 1 в бите маски разрешает прохождение соответствующего ему запроса прерывания в ILAT для дальнейшей обработки.

Регистр IMASK – 64-битный регистр, доступный как два 32-битных регистра IMASKL и IMASKH. После сброса значение IMASK всегда равно 0.

12.1.6 Регистры приоритетного маскирования прерываний (PMASKL/PMASKH)

Регистр PMASK – 64-битный регистр, доступный как два 32-битных регистра PMASKL и PMASKH. Регистры доступны только для чтения и отображают состояние запросов прерываний, которые поступают на приоритетный шифратор контроллера прерываний.

Каждый бит регистра запроса прерываний соответствует определенному источнику прерываний и имеет соответствующий ему (с тем же номером) бит в регистре маскирования IMASK и бит в регистре PMASK. Назначения бит этих регистров приведены ниже (Таблица 77 и Таблица 78).

Таблица 77 – Биты регистров ILATL, IMASKL, PMASKL

Бит	Имя	Описание
0	INT_KERNEL	Прерывание ядра
1	INT_GPIO	Порты PA, PB, PC
2	INT_TIMER0LP	Низкий приоритет таймера #0
3	INT_TIMER1LP	Низкий приоритет таймера #1
4	INT_UART0	UART0
5	INT_UART1	UART1
6	INT_LINK0	Порт связи #0
7	INT_LINK1	Порт связи #1
8	-	Зарезервирован
9	-	Зарезервирован
10	INT_NANDC	NAND контроллер
11	INT_MIL0	МКПД 0
12	INT_MIL1	МКПД 1
13	INT_DIGC	Цифровой коррелятор
14	INT_DMA0	DMA #0
15	INT_DMA1	DMA #1
16	INT_DMA2	DMA #2
17	INT_DMA3	DMA #3
18	INT_ARINC_RX	Приемник ARINC
19	INT_ARINC_TX	Передатчик ARINC
20	INT_SPI1	Контроллер SPI1
21	INT_SPI2	Контроллер SPI2
22	INT_DMA4	DMA #4
23	INT_DMA5	DMA #5
24	INT_DMA6	DMA #6
25	INT_DMA7	DMA #7
26	INT_I2C	Контроллер I2C
27	INT_GTMR0	Таймер 0 с функцией Захвата/ШИМ
28	INT_GTMR1	Таймер 1 с функцией Захвата/ШИМ

Бит	Имя	Описание
29	INT_DMA8	DMA #8
30	INT_DMA9	DMA #9
31	INT_DMA10	DMA #10

Таблица 78 – Биты регистров ILATH, IMASKH, PMASKH

Бит	Имя	Описание
0	INT_DMA11	DMA #11
1	INT_ADDA0	Модуль UP\DOWN 0
2	INT_ADDA1	Модуль UP\DOWN 1
3	INT_ADDA2	Модуль UP\DOWN 2
4	INT_ADDA3	Модуль UP\DOWN 3
5	INT_DMA12	DMA #12
6	INT_HOST	Хост-интерфейс
7	INT_LCD	Контроллер LCD
8	-	-
9	INT_IRQ0	Вывод IRQ0
10	INT_IRQ1	Вывод IRQ1
11	INT_IRQ2	Вывод IRQ2
12	INT_IRQ3	Вывод IRQ3
13	INT_SPI0	Контроллер SPI 0
14	INT_SSI0	Контроллер SSI 0
15	INT_SSI1	Контроллер SSI 1
16	INT_VIRPT	Векторный регистр VIRPT
17	INT_VCAM	Контроллер видеокамеры
18	INT_BUSLOCK	Блокировка шины
19	INT_H264	Декодер H264
20	INT_TIMER0HP	Высокий приоритет таймера 0
21	INT_TIMER1HP	Высокий приоритет таймера 1
22	INT_ALARM	Прерывание ALARM от RTC
23	INT_TIC	TIC прерывание от RTC
24	INT_WDOG	Прерывание от сторожевого таймера
25	INT_HWERR	Аппаратная ошибка
31:26	-	-

12.2 Операции с прерываниями

Операциями с прерываниями являются следующие виды операций:
- установка запросов прерываний;
- обработка прерываний или исключений;
- возврат из прерывания.

Контроллер прерываний включает следующий регистры установки и управления:
- регистр управления прерываниями (INTCTL);
- регистры маски прерываний (IMASK) – IMASKH и IMASKL;
- регистры макси приоритета (PMASK) – PMASKH и PMASKL;
- регистры защелки прерываний (ILAT) – ILATH и ILATL.

Наличие запроса прерывания отражается активным (равен 1) битом в регистре ILAT. Если соответствующий ему бит в регистре маски IMASK установлен, запрос прерывания поступает на дальнейшую обработку. Регистр приоритетов

PMASK хранит запросы прерываний, которые процессор принял к обработке. Приоритетом прерывания является номер бита в регистре PMASK (от 0 до 63).

Регистр приоритетов позволяет блокировать низкоприоритетные запросы прерываний до момента окончания обработки более высокоприоритетного. Если прерывание разрешено, и его приоритет выше текущего обрабатываемого прерывания (или нет обработки прерываний в данный момент), контроллер прерываний формирует активный уровень запроса прерывания к процессору, сопровождая его номером прерывания и адрес-вектором процедуры обслуживания данного прерывания. Если бит GIE (бит глобального разрешения прерываний) в регистре SQCTL установлен, процессор прерывает текущий ход программы и переходит к выполнению процедуры обработки прерывания.

Как только прерывание начинает обслуживаться, соответствующий данному прерыванию бит в регистре PMASK устанавливается в 1. Данная установка запрещает прохождение запросов прерываний с приоритетом ниже обрабатываемого. При этом процедура обработки прерывания может разрешить обработку более высокоприоритетных прерываний. Как только обработка прерывания заканчивается, соответствующий ему бит в PMASK сбрасывается. Это происходит при выполнении команд RTI или RDS. При этом команда RTI означает завершение процедуры обработки, а команда RDS означает, что обработка текущего прерывания может быть прервана другим прерыванием, т.е. возможны вложенные прерывания.

Активный запрос прерывания вызывает установку соответствующего бита в регистре ILAT. Существует возможность программно установить бит в этом регистре, т.е. приложение может эмулировать прерывание посредством записи в регистр ILAT. Запись по адресам ILATSTL или ILATSTH обновляет регистр ILAT значением логического «ИЛИ» старого содержимого и нового записываемого. В результате, каждый установленный бит в записанных данных устанавливает соответствующий бит в регистре ILAT. Установка бита прерывания вызывает в процессоре предположение о возникновении соответствующего прерывания. Биты прерывания могут быть также сброшены записью в адреса сброса ILATCLL или ILATCLH. Такая запись применяет оператор AND к записываемым данным и старым данным регистра ILAT. В этом случае нулевой бит во входных данных сбрасывает соответствующий бит в ILAT, тогда как установленный бит остается неизменным. Таким образом, приложение может сбросить ждущее прерывание до его исполнения.

Операции сброса или установки бит в ILAT должны выполняться при следующих условиях:

- незадействованные (резервные) биты не могут быть установлены. При записи по адресам установки или сброса резервные биты должны быть нулями;
- биты программного исключения и эмуляторного исключения не могут быть установлены через ILATST. Для того чтобы вызвать такие исключения следует использовать команды TRAP и EMUTRAP;
- прерывания, которые запускаются уровнем, не должны изменяться;
- записи в ILATL и ILATH не должны предприниматься (т.е. только сброс или установка);
- прерывание должно быть сброшено в то время, когда оно запрещено, иначе оно может быть обслужено до того, как будет сброшено;
- адреса установки и сброса работают с одинарными словами.

12.3 Программа обработки аппаратного прерывания

Действия, выполняемые при обработке прерываний можно разделить на две группы:

- действия, выполняемые контроллером прерываний;
- действия, выполняемые процессором.

Рассмотрим последовательность обработки запроса прерывания на примере прерывания от таймера 0. Действия контроллера прерываний будут следующие:

- Когда таймер включен и его счетчик имеет в текущем такте SOCCLK значение 1, вырабатывается активный уровень запроса прерывания. Активный запрос существует только один такт, которого достаточно, чтобы в следующем такте биты 2 и 52 регистра ILAT установились в 1. Бит 2 соответствует запросу прерывания от таймера 0 с низким уровнем приоритета, а бит 52 – с высоким уровнем.
- При ожидании прерывания с высоким уровнем приоритета бит 52 в регистре IMASK должен быть установлен в 1. Логическое «И» между битом запроса прерывания и битом маски равно 1, следовательно, прерывание может быть принято к обслуживанию.
- Далее выполняется анализ регистра PMASK. Если биты регистра с 52-й по 63-й равны нулю, в обработке нет прерываний равного или более высокого приоритета, чем данное. В противном случае придется подождать, пока не будет обслужено более высокоприоритетное прерывание.
- Выполнения описанных выше условий достаточно чтобы контроллер прерываний сформировал запрос прерывания к процессору. Кроме активного уровня запроса на линии, контроллер передает в процессор номер прерывания (52), а также извлекает из таблицы векторов прерываний значение регистра с номером 52 и передает его в процессор. Это значение является адресом процедуры обработки данного прерывания.

Когда запрос прерывания поступает в процессор (в устройство управления), первым действием процессора является проверка бита глобального разрешения прерываний GIE регистра управления SQCTL. Если бит равен 1 – прерывания разрешены и процессор выполняет следующие действия:

- Запоминает номер и адрес-вектор обработки пришедшего прерывания.
- Выполняет сброс всего конвейера чтения команд вместе с буфером IAB.
- Запускает чтение команд по адрес-вектору прерывания.
- Конвейер обработки команд продолжает и заканчивает обработку всех команд, которые находились на конвейере в момент сброса конвейера чтения команд.
- Когда последняя команда конвейера обработки завершает свое выполнение, процессор повторно проверяет бит разрешения прерываний GIE и, если прерывания по-прежнему разрешены, первая команда процедуры обработки прерывания поступает из буфера команд на конвейер обработки.
- При достижении первой командой последней стадии конвейера (W) происходит формирование сигнала о том, что данное прерывание взято на обработку. Этот факт приводит к установке бита 52 регистра PMASK и сброс бита 52 в регистре ILAT. Установленный бит в регистре PMASK запрещает все запросы прерывания уровнем ниже 53 (т.е. новое прерывание от таймера 0 также запрещено). Вместе с этим процессор

запоминает в специальном регистре RETI адрес следующей команды, которая должна быть выполнена при выходе из прерывания. Дополнительно процессор устанавливает внутренний флаг EXE_ISR, который запрещает все прерывания. Флаг EXE_ISR указывает на то, что процессор находится в состоянии обработки прерывания. Флаг доступен для чтения как 20-й бит регистра SQSTAT.

- Переход на обработку прерывания вызывает установку режима супервизора (если до этого был режим пользователя).
- Сброс бита 52 в регистре ILAT означает, что бит стал доступен для приема нового запроса прерывания от таймера 0.
- Далее процессор выполняет все команды, которые соответствуют процедуре обслуживания данного прерывания. Для выхода из процедуры обработки используется команда RTI.
- Выполнение команды RTI приводит к сбросу бита 52 в регистре PMASK, что разрешает прохождение всех прерываний, которые ранее блокировались. Также данная команда вызывает сброс флага EXE_ISR, что снимает блокировку прерываний и включает функции бита GIE. Процессор переходит к выполнению команд по адресу из регистра RETI.

Процедура, описанная выше, соответствует типичной ситуации обработки при отсутствии каких-либо аномалий. Однако существуют некоторые особенности:

- Если перед повторной проверкой разрешения прерываний (после полного освобождения конвейера обработки) оказывается, что прерывания запрещены (это может быть сделано командой сбрасывающей бит GIE), тогда все действия по обработке прерывания отменяются и конвейер перезапускается с адреса, следующего за адресом последней выполненной команды.
- Если запрос прерывания из контроллера пришел в процессор, но бит GIE равен нулю, прерывание будет отложено. В это время в контроллере прерываний может появиться запрос с более высоким приоритетом, чем данное. Контроллер прерываний обработает его и передаст в процессор номер и адрес-вектор нового прерывания. Запрос от таймера 0 в этом случае будет отложен до момента окончания обработки нового прерывания.
- Возможна ситуация, когда запрос пришел из контроллера, но в это же время процессор выполнял сброс бита запроса прерывания в регистре ILAT. Эта ситуация приведет к тому что запрос из контроллера прерываний будет стабильным в течение одного или нескольких тактов. Тем не менее, в такой ситуации запрос может быть обработан. Если снятие запроса произошло после или во время сброса конвейера чтения команд, это означает, что запрос будет обслужен. Для надежного выполнения подобных операций необходимо предварительно запрещать прерывания битом GIE, а затем производить сброс в контроллере прерываний.

Обычно процедура обработки прерывания начинается с сохранения контекста процессора, т.е. сохранения на стеке всех регистров процессора, которые будут использованы программой обработки прерывания. Это необходимо для того, чтобы после обработки прерывания восстановить прежнее состояние процессора и дать возможность прерванной программе корректно возобновить работу. Факт обработки прерывания отражается в установке флага EXE_ISR и это запрещает все прерывание, в том числе и более высокоприоритетные, чем данное. Если время обработки прерывания значительно и это может повредить обработке запроса более

приоритетного прерывания, программа может разрешить обслуживание более высокоприоритетных прерываний. Для этого необходимо обязательно сохранить на стеке содержимое регистра RETI, который хранит адрес возврата из прерывания. Для сохранения содержимого RETI используется специальный псевдоним этого регистра – RETIB.

Выполнение команды **j0 = RETIB;;** вызовет копирование регистра RETI в регистр j0 и сброс флага EXE_ISR. Сам регистр j0 должен до этого быть сохранен на стеке. После выполнения указанной команды процессор имеет возможность обслуживать более высокоприоритетные запросы, чем запрос 52. Описываемая ситуация вносит коррективы в последовательность действий при выходе из обработки прерывания. Перед выходом мы имеем ситуацию, когда содержимое регистра возврата RETI неопределенно (новые прерывания могли изменить его). Поэтому необходимо взять наш адрес возврата из регистра j0 (или другого места, где он хранился) и вернуть обратно в RETI. Однако выполнение обычной пересылки **RETI = j0;;** будет некорректным, т.к. после этой команды возможно возникновение прерывания и адрес возврата может быть потерять. Снова следует использовать регистр-псевдоним RETIB. Пересылка **RETIB = j0;;** пересылает адрес возврата в RETI и устанавливает флаг EXE_ISR. Это запрещает прерывания и позволяет безопасно восстановить весь контекст, а затем выполнить финишную команду RTI.

В процедуре, описанной выше, разрешения прерываний с более высоким приоритетом, чем текущее. Разрешаются прерывания с приоритетом от 53 и выше, но прерывание с приоритетом 52 запрещено. Это означает, что, обрабатывая запрос от таймера 0, имеется возможность принять новый запрос прерывания от таймера, т.к. бит 52 в регистре ILAT был сброшен и имеет возможность зафиксировать новое событие, но не обслуживать его (т.к. бит 52 в регистре PMASK установлен). Также нет возможности обслуживать запросы с приоритетом ниже 52.

Разрешить обслуживание запросов с приоритетом ниже 52 можно с помощью команды RDS. Выполнение команды **j0 = RETIB;;** разрешает только прерывания выше 52-го, т.к. бит 52 в регистре PMASK установлен. Выполнение команды RDS вызывает следующие действия процессора:

- сбрасывается в ноль текущий самый высокоприоритетный установленный бит в регистре PMASK;
- сбрасывается флаг EXE_ISR;
- выполняется переход в режим работы, который предшествовал обработке прерывания;
- выполняется перезапуск конвейера в новых условиях.

Перед выполнением команды RDS необходимо сохранить адрес возврата. Если прежним режимом является режим пользователя с соответствующими ограничениями, и возврат в этот режим нежелателен, нужно сначала сохранить регистр SQCTL, и затем установить в нем бит режима супервизора. В этом случае после команды RDS останется режим супервизора.

Вышеописанные действия позволяют реализовать вложенные прерывания. Выход из подобных прерываний выполняется с использованием регистра RETIB.

Описанные действия относятся к обработке любого аппаратного прерывания, а не только 52-го.

Команда возврата из прерывания RTI является командой перехода и может выполняться с опцией предсказания или без. Использование опции предсказания позволяет ускорить выход из прерывания на несколько тактов процессора. Необходимо помнить, что содержимое регистра RETI должно быть корректно в момент выполнения предсказания, т.е. в момент передачи в память адреса линии команд содержащей RTI. Если регистр RETI загружается непосредственно перед командой RTI или в нескольких тактах до RTI, предпочтительнее использовать

команду RTI без предсказания перехода, т.к. в данной случае предсказание будет ошибочным, и все выполненные действия будут отменены.

12.4 Особенности сохранения контекста процессора

Во время выполнения программы процессор может находиться в различных режимах. Выделяют два основных режима работы:

- режим супервизора (ядра или системный);
- режим пользователя.

Режим супервизора может иметь несколько подрежимов (или видов):

- выполнение задачи системы;
- обработка прерывания;
- обработка программной исключительной ситуации;
- отладка.

Подрежим можно определить в результате анализа регистра состояния SQSTAT. Для каждого подрежима процессор имеет соответствующие регистры связи:

- CJMP-регистр связи для выполнения вызовов процедур при нормальном ходе программы как в режиме супервизора, так и в режиме пользователя;
- RETI-регистр связи при возникновении прерывания;
- RETS-регистр связи при возникновении программной исключительной ситуации;
- DBGE-регистр связи при вызове эмулятора.

Регистры связи RETI, RETS, DBGE используются только при входе в исключительную ситуацию и при выходе из нее. При нормальном ходе программы для связи используется регистр CJMP. В процессоре используется один общий для всех режимов регистр-указатель стека. Также в качестве указателя стека может быть использован любой от 0-го до 30-го регистр IALU, но по умолчанию компилятор использует в качестве указателя стека регистр J27/K27. На аппаратном уровне процессор поддерживает раздельные регистры J27/K27 для режима супервизора (KSP или SP) и режима пользователя (USP).

Режим работы с двумя указателями стеков возможен только при установленном бите SQCTL[14]. Если бит SQCTL[14] равен нулю, процессор работает с общим для всех режимов указателем стека. В режиме ядра регистр j27/k27 соответствует KSP, а также имеется возможность чтения и записи указателя USP через резервный регистр номер 27 группы регистров базы и длины циклической адресации модулей IALU. В режиме пользователя j27/k27 доступен только как USP. Выбор указателя стека осуществляется следующим образом:

- если процессор находится в режиме обработки прерывания – используется KSP;
- если процессор находится в режиме обработки исключительной ситуации – используется KSP;
- если процессор находится в режиме эмулятора – используется KSP;
- если процессор исполняет прикладную задачу и бит SQCTL[9] равен нулю – используется USP;
- если процессор исполняет прикладную задачу и бит SQCTL[9] равен единице – выбор указателя определяется битом SQCTL[7]: если бит равен нулю – KSP, иначе – USP.

Таким образом, в режиме супервизора возможна работа как с указателем ядра, так и с указателем пользователя.

Основные трудности в выборе указателя возникают в момент смены режима, т.к. из-за наличия конвейера фазы чтения и записи регистра разнесены во времени, и различным моментам времени могут соответствовать различные режимы.

Процесс смены режима посредством модификации бита SQCTL[9] наиболее безопасный, т.к. каждый раз после записи регистра SQCTL происходит перезапуск конвейера. Это гарантирует, что команды, следующие за сменой режима, извлекаются и выполняются в требуемом режиме.

Смена режима посредством программной исключительной ситуации (установка SWI_mode) безопасна во время входа в обработчик, т.к. возникновение исключительной ситуации приводит к сбросу конвейера, его очистке и последующему переходу на программу обработки. Безопасный выход из обработчика должен быть реализован соответствующим образом. Выход из обработчика (сброс SWI_mode) осуществляется посредством выполнения команды RTI. Данная команда должна исполняться с опцией RTI (NP). Это гарантирует сброс конвейера после смены режима и последующее корректное использование указателя стека (а также работу устройства защиты).

Смена режима посредством аппаратного прерывания (установка ISR_mode) безопасна во время входа в обработчик. Это гарантируется аппаратурой процессора. Признак прерывания движется по конвейеру одновременно с командами обработчика и осуществляет выбор корректного указателя стека, если первые команды обработчика используют указатель. Безопасный выход из прерывания (сброс ISR_mode) должен быть реализован соответствующим образом. Выход осуществляется посредством выполнения команды RTI. Данная команда должна исполняться с опцией RTI (NP). Это гарантирует сброс конвейера после смены режима и последующее корректное использование указателя стека (а также работу устройства защиты).

Сброс режима обработки аппаратного прерывания или программного прерывания может быть выполнен не только командой RTI, но и командой RDS. Опасность использования команды RDS при работе с разными указателями стека, а также с устройством защиты, связано с тем, что выполнение команды RDS приводит к возврату в режим, предшествовавший прерыванию. При работе в режиме раздельных указателей стека после выполнения команды RDS будет выполнен перезапуск конвейера. До выполнения команды RDS будет доступен указатель стека супервизора, после – это может быть либо указатель стека супервизора или пользователя.

Поскольку возможно использование сразу двух указателей стека, сохранение контекста может быть ускорено, если стеки размещаются в различных банках памяти. Это позволяет за такт сохранить восемь регистров процессора. Если использование двух стеков неудобно, можно использовать факт разбиения каждого банка внутренней памяти на две половины, одна из которых – четные квадрослова, вторая – нечетные. В этом случае регистр К27 может хранить копию указателя J27, и при сохранении контекста возможна работа сразу с парой квадрослов. Это позволяет сохранять и восстанавливать сразу восемь регистров.

12.5 Обработка программных прерываний (исключений)

Исключения – программные прерывания, которые вызваны исполняемым программным кодом. При этом могут быть специальные команды (TRAP, EMUTRAP) или ситуации, которые возникают при выполнении команд. К таким ситуациям относятся аномалии при вычислении с плавающей запятой, доступ к длинным словам по невыровненному адресу и др.

Обработка программного прерывания отличается от аппаратного следующим:

- Бит GIE не имеет значения при обработке исключений. Для программных прерываний используется специальный бит глобального разрешения SWIE.
- Часть программных прерываний имеет индивидуальные биты разрешения. Это касается исключений, которые возникают в вычислительных модулях X и У. Другие исключения не имеют специальных бит для разрешения, кроме глобального.
- Программное прерывание происходит сразу за выполнением команды, инициировавшей это событие.
- Для хранения адреса возврата используется специальный регистр RETS, а не регистр RETI.
- Возникновение исключения приводит к установке 62-го бита регистра PMASK, что означает блокировку всех аппаратных прерываний.
- В качестве адреса-вектора процедуры обработки исключений используется содержимое регистра IVSW.
- Возникновение исключения приводит к установке специального бита EXE_SWI. Бит виден в 21-м разряде регистра SQSTAT. Бит может быть установлен только при возникновении исключительной ситуации. Бит может быть сброшен только при выходе из обработки исключения или командой RDS. Других механизмов установки или сброса данного бита нет.
- Установка бита EXE_SWI приводит к переходу в режим супервизора (если ранее был режим пользователя), а также к запрещению всех программных и аппаратных прерываний.
- Если при обработке исключительной ситуации возникло новое исключение, оно не будет обработано.

Последовательность действий процессора, выполняемых при возникновении исключительной ситуации, следующая:

- В поле SQSTAT[11:8] процессор сохраняет код исключительной ситуации. Если ситуация была вызвана командой TRAP, в поле SQSTAT[7:3] дополнительно сохраняется параметр данной команды. Все это позволяет идентифицировать источник исключения.
- Сбрасывается весь конвейер процессора и начинается чтение команд по адресу из регистра IVSW.
- В регистре RETS сохраняется адрес возврата, а именно адрес первой команды линии следуемой за линией команд вызвавшей исключение.
- Бит PMASK[62] устанавливается в 1, запрещая все аппаратные прерывания.
- Устанавливается внутренний флаг EXE_SWI, переводя процессор в режим супервизора и запрещая все аппаратные и программные прерывания.
- Процессором выполняются все команды процедуры обработки исключительной ситуации.
- Перед выходом из прерывания программа копирует содержимое регистра RETS в регистр RETI с помощью команды RETI = RETS;;. Регистр RETIB, в данном случае, не должен использоваться.
- Выход из обработчика осуществляется с помощью команды RTI.

При возврате из программного прерывания используется регистр RETI, поэтому последовательность выхода может быть такой:

[j31+temporary address] = RETI;;

RETI = RETS;;
RTI (NP); RETI = [j31+temporary address];;

Таким образом видно, что программное прерывание при своем возникновении создает очень строгие ограничения в поведении процессора, также как и аппаратное. Однако возникновение аппаратного прерывания не запрещает обработку программного прерывания, если оно возникнет при выполнении команд аппаратного обработчика. При обработке программного прерывания можно разрешить аппаратные прерывания, выполнив команду RDS. Выполнение команды RDS приведет к следующему:

- произойдет сброс бита PMASK[62], т.е. все аппаратные прерывания в контроллере прерываний станут доступны.
- произойдет сброс бита EXE_SWI. Аппаратные прерывания станут управляться битом GIE, а программные – SWIE. Произойдет возврат к предыдущему режиму работы.

При использовании команды RDS необходимо заранее учесть влияние последующих событий на корректное исполнение программного обработчика.

Программное прерывание может быть вложенным в аппаратное, если биты EXE_ISR и EXE_SWI равны 1. В подобной ситуации команды RTI или RDS оказывают влияние только на бит EXE_SWI, и их исполнение означает переход на уровень аппаратного обработчика.

12.6 Обработка прерываний эмулятора

Прерывания эмулятора – события, которые приводят к переходу процессора в режим эмуляции, когда все его действия управляются посредством команд через интерфейс JTAG. Подобное прерывание может быть вызвано:

- командой EMUTRAP;
- срабатыванием точки наблюдения;
- аппаратным запросом из JTAG интерфейса.

Обработка данного типа прерываний имеет следующие особенности:

- Специальный бит глобального разрешения DBGEN. Биты GIE и SWIE не влияют.
- Прерывания от точек наблюдения и из JTAG имеют дополнительные биты разрешения.
- Прерывание происходит сразу за командой, которая покинула последнюю стадию конвейера W в момент возникновения запроса.
- Для хранения адреса возврата используется специальный регистр DBGE.
- В качестве адреса-вектора процедуры обработки используется фиксированный адрес 0x001F_03A4.
- Возникновение исключения приводит к установке специального бита EMUL. Бит виден в 22-м разряде регистра SQSTAT. Бит может быть установлен только при возникновении данной исключительной ситуации. Бит может быть сброшен только при выходе из обработки исключения командой RTI. Других механизмов установки или сброса данного бита не предусмотрено.
- Установка бита EMUL приводит к переходу в режим эмулятора. Также это приводит к запрещению всех программных и аппаратных прерываний.

Последовательность действий процессора, выполняемых при возникновении прерывания эмулятора, следующая:

- В поле SQSTAT[15:12] процессор сохраняет код исключительной ситуации, что позволяет идентифицировать источник исключения.
- Сбрасывается весь конвейер процессора и начинается чтение команд по адресу 0x001F_03A4. При этом изменения адреса не происходит. Данный адрес соответствует регистру команд интерфейса JTAG, т.е. процессор все время исполняет те команды, которые ему выставляет JTAG интерфейс.
- В регистре DBGE сохраняется адрес возврата.
- Устанавливается внутренний флаг EMUL, переводя процессор в режим эмулятора и запрещая все аппаратные и программные прерывания.
- Процессором выполняются все команды, которые предоставляет ему JTAG интерфейс.
- Перед выходом из прерывания программа копирует содержимое регистра DBGE в регистр RETI с помощью команды RETI = DBGE;;. Регистр RETIB, в данном случае, не должен использоваться.
- Выход из обработчика осуществляется с помощью команды RTI.

При возврате из программного прерывания используется регистр RETI, поэтому последовательность выхода может быть такой:

```
[j31+temporary address] = RETI;;
RETI = DBGE;;
RTI (NP); RETI = [j31+temporary address];;
```

Прерывание эмулятора накладывает серьезные ограничения на перечень команд, которые должны поставляться интерфейсом JTAG. Например, запрещены любые команды перехода и т.д. При переходе в режим эмулятора блокируются некоторые аппаратные ресурсы. Например, таймеры останавливаются и не меняют свое значение до момента выхода из режима эмуляции.

12.7 Источники прерываний процессора

Таблица 79 – Таблица векторов прерываний

Приоритет	Прерывание	Описание	Срабатывание
63 (самый высокий)	-	Зарезервирован	-
62	-	Зарезервирован	-
61	-	Зарезервирован	-
60	-	Зарезервирован	-
59	-	Зарезервирован	-
58	-	Зарезервирован	-
57	HWERR	Аппаратная ошибка	По уровню
56	WDOG	RTC.Сторожевой таймер	По фронту
55	TIC	RTC.Интервальный таймер	По фронту
54	ALARM	RTC.Будильник	По фронту
53	TIMER1H	Высокий приоритет таймера 1	По фронту
52	TIMER0H	Высокий приоритет таймера 0	По фронту
51	H264	Декодер H264	По уровню
50	BUSLOCK	Блокировка шины	По фронту
49	VCAM	Контроллер видеокамеры	По уровню
48	VIRPT	Векторное прерывание	По фронту

Приоритет	Прерывание	Описание	Срабатывание
47	SSI1	Контроллер I2S_1	По уровню
46	SSI0	Контроллер I2S_0	По уровню
45	SPI0	Контроллер SPI 0	По уровню
44	IRQ3	Вывод IRQ3	По фронту или по уровню
43	IRQ2	Вывод IRQ2	По фронту или по уровню
42	IRQ1	Вывод IRQ1	По фронту или по уровню
41	IRQ0	Вывод IRQ0	По фронту или по уровню
40	-	-	-
39	LCD	Контроллер ЖКИ	По уровню
38	HOST	Хост-интерфейс	По фронту
37	DMA12	DMA канал 12	По фронту
36	ADDA3	UP\DOWN 3	По фронту
35	ADDA2	UP\DOWN 2	По фронту
34	ADDA1	UP\DOWN 1	По фронту
33	ADDA0	UP\DOWN 0	По фронту
32	DMA11	DMA канал 11	По фронту
31	DMA10	DMA канал 10	По фронту
30	DMA9	DMA канал 9	По фронту
29	DMA8	DMA канал 8	По фронту
28	GTMR1	Таймер 1 с функцией Захвата/ШИМ	По уровню
27	GTMR0	Таймер 0 с функцией Захвата/ШИМ	По уровню
26	I2C	Контроллер I2C	По уровню
25	DMA7	DMA канал 7	По фронту
24	DMA6	DMA канал 6	По фронту
23	DMA5	DMA канал 5	По фронту
22	DMA4	DMA канал 4	По фронту
21	INT_SPI2	Контроллер SPI2	По уровню
20	INT_SPI1	Контроллер SPI1	По уровню
19	ARINC_TX	ARINC передатчик	По уровню
18	ARINC_RX	ARINC приемник	По уровню
17	DMA3	DMA канал 3	По фронту
16	DMA2	DMA канал 2	По фронту
15	DMA1	DMA канал 1	По фронту
14	DMA0	DMA канал 0	По фронту
13	DIGC	Цифровой коррелятор	По фронту
12	MIL1	МКПД 1	По фронту
11	MIL0	МКПД 0	По фронту
10	NANDC	NAND флэш	По уровню
9	-	Зарезервирован	-
8	-	Зарезервирован	-
7	LINK1	Запрос порта связи 1	По уровню
6	LINK0	Запрос порта связи 0	По уровню
5	UART1	UART 1	По уровню
4	UART0	UART 0	По уровню
3	TIMER1L	Низкий приоритет таймера 1	По фронту
2	TIMER0L	Низкий приоритет таймера 0	По фронту
1	GPIO	Порты PA,PB,PC	По уровню
0 (самый низкий)	KERNEL	Ядро	По фронту

12.7.1 Прерывания истечения таймера

Возможны четыре прерывания от таймеров: два для каждого таймера, одно – с высоким приоритетом и одно – с низким. Два приоритетов для одного таймера дают возможность пользователю выбрать уровень приоритета.

Регистры вектора прерываний таймера: IVTIMER0HP, IVTIMER1HP, IVTIMER0LP и IVTIMER1LP. В случае прерывания от таймера, устанавливаются оба бита приоритетов (высокий и низкий) в ILAT (например, для таймера 0 это биты 2 и 52).

Сброс или установка бита запроса прерывания от таймера в регистре ILAT с помощью адресов ILATSTx или ILATCLx оказывает влияние сразу на оба бита таймера, т.е. нельзя разграничить запрос низкого от запроса высокого приоритета.

12.7.2 Прерывания на обслуживание порта связи

Существуют два канала портов связи, которые обычно работают с выделенными им каналами DMA. Регистры вектора обслуживания порта связи: IVLINK0, IVLINK1. Когда данные поступают в буфер приемника порта связи, и соответствующий ему канал DMA не инициализирован, порт связи формирует запрос прерывания на обслуживание.

12.7.3 Прерывания завершения работы каналов DMA

Каждый из 13-ти каналов DMA может генерировать прерывание. Регистры вектора: IVDMA0, IVDMA1, IVDMA2, IVDMA3, IVDMA4, IVDMA5, IVDMA6, IVDMA7, IVDMA8, IVDMA9, IVDMA10, IVDMA11 и IVDMA12. Если бит разрешения прерывания в регистре управления передачей DMA установлен, канал DMA генерирует прерывание по завершении передачи блока.

12.7.4 Прерывания от внешних источников

Порты общего назначения PA, PB, PC могут формировать запрос на прерывание от каждого из 96-ти внешних контактов. Каждый запрос может быть запрограммирован индивидуально. Однако все запросы от портов общего назначения имеют один вектор прерывания IVGPIO. Существует четыре внешних прерывания общего назначения IRQ3–0, которые могут быть обработаны особым образом. Эти внешние входы соответствуют контактам 3-0 порта PC. Для них существуют отдельные вектора прерывания (IRQ3–0): IVIRQ0, IVIRQ1, IVIRQ2 и IVIRQ3. Когда активируется один из входов, запускается прерывание (если они разрешены). Тип прерывания по фронту или уровню программируется в регистре INTCTL.

12.7.5 Векторное прерывание

Векторное прерывание – прерывание общего назначения для использования другим мастером. Другое ведущее устройство или процессор могут записывать адрес в этот регистр. Запись вызывает установку запроса прерывания, а записанное значение в регистре – адрес обработки прерывания. Регистр вектора прерывания – VIRPT.

12.7.6 Прерывание от захвата шины

Прерывание захвата шины происходит, когда установлен бит захвата шины в регистре BUSLOCK, и процессор становится хозяином шины. Регистр вектора захвата шины – IVBUSLK. Данное прерывание сохранено для совместимости с процессором 1967ВН028. В процессоре 1967ВН044 внешняя шина всегда принадлежит процессору, поэтому выполнение процедуры захвата шины не требуется. Однако установка бита BUSLOCK приведет к генерации прерывания.

12.7.7 Прерывание от аппаратной ошибки

Прерывание аппаратной ошибки указывает на одну из возможных ошибок:

- Ошибка в DMA. При инициализации некоторого канала обнаружена ошибочная ситуация.
- Широковещательное чтение из внешней памяти (аналогично 1967ВН028). Бит 16 в SYSTAT активен, если произошла данная ошибка.
- Доступ к отключенной SDRAM. Бит 18 в SYSTAT активен, если произошло данное событие.
- Обращение к несуществующему адресному пространству. Бит 19 в SYSTAT активен, если произошло данное событие.
- Ошибка порта связи. В одном из регистров LSTATx в полях RER или TER активна ошибка.

Регистр вектора прерываний аппаратных ошибок – IVHW.

12.7.8 Прерывания RTC

Модуль часов реального времени RTC может формировать три запроса прерывания. Прерывание ALARM происходит при совпадении счетчика времени с заданным значением, что позволяет реализовать функции будильника. Прерывание TIC является периодическим и формируется через заданный интервал времени. Данное прерывание может быть полезно при работе ОС. Прерывание сторожевого таймера формируется в момент, когда счетчик сторожевого таймера принимает значение 1. Это служит напоминанием системе о необходимости выполнить повторную инициализацию сторожевого таймера. Если повторная инициализация не выполняется в течение определенного времени, сторожевой таймер инициирует сброс всей системы.

12.7.9 Прерывания от периферийных устройств

В процессоре имеется ряд периферийных устройств, которые могут формировать запросы прерываний к процессору. Подробное описание порядка формирования и обслуживания прерывания приведено в главах, посвященных каждому из периферийных устройств.

12.7.10 Программные исключения

Программные исключения – прерывания, появляющиеся во время выполнения программного кода. Если происходит исключение и бит SWIE равен нулю – исключение теряется. Вектор прерываний программных исключений – IVSW.

Тип исключительной ситуации можно определить по номеру в регистре состояния процессора устройства управления. Соответствие номера типу исключительной ситуации приведено ниже (Таблица 80). Используя номер исключительной ситуации в регистре состояния, можно выполнить соответствующую обработку возникшей исключительной ситуации. При этом может понадобиться дополнительная информация, доступная из других источников или полученная в результате анализа кода командной строки, вызвавшей исключение.

Таблица 80 – Программные исключения

Номер	Имя	Описание события
0	trap	Вызывается выполнением команды TRAP. Дополнительный номер исключения можно получить из регистра состояния
1	Wp_abort	Срабатывание точки наблюдения в модуле отладки
2	I0	Исключительная ситуация в модулях ХУ
3	UNDEF	Неопределенная команда или сочетание команд
4	I1	Невыровненный адрес при доступе к длинным словам
5	I3+prot	Нарушение прав доступа к памяти
6	PRFMC	Переполнение счетчика событий
7	I2	Чтение из области 0x0C00_0000 – 0x0FFF_FFFF
8		
9	H264	Ошибочная ситуация в модуле H264
10		
11		
12		
13		
14		
15		

13 Прямой доступ к памяти

Встроенный контроллер DMA осуществляет передачу данных без участия процессора. Данные и код могут быть загружены в процессор через DMA-передачу, которая может быть осуществлена между:

- внутренней памятью процессора и внешней памятью, внешней периферией, внутренней памятью;
- внешней памятью и внешними периферийными устройствами;
- внешней памятью и портами связи или между двумя портами связи.

Внешние выводы контроллера приведены в таблице 82

Таблица 81 – Внешние выводы контроллера DMA

Обозначение вывода (основное)	Обозначение вывода контроллера	Тип вывода	Функциональное назначение
PB[4]	nDMAR[0]	I/O	Запрос канала 0 контроллера DMA
PB[5]	nDMAR[1]	I/O	Запрос канала 1 контроллера DMA
PB[6]	nDMAR[2]	I/O	Запрос канала 2 контроллера DMA
PB[7]	nDMAR[3]	I/O	Запрос канала 3 контроллера DMA
PB[16]	nDMAR[1]	I/O	Запрос канала 1 контроллера DMA
PB[22]	nDMAR[4]	I/O	Запрос канала 4 контроллера DMA
PB[23]	nDMAR[5]	I/O	Запрос канала 5 контроллера DMA
PB[24]	nDMAR[6]	I/O	Запрос канала 6 контроллера DMA
PB[25]	nDMAR[7]	I/O	Запрос канала 7 контроллера DMA
PB[26]	nDMAR[2]	I/O	Запрос канала 2 контроллера DMA
PB[27]	nDMAR[8]	I/O	Запрос канала 8 контроллера DMA
PB[28]	nDMAR[9]	I/O	Запрос канала 9 контроллера DMA
PB[29]	nDMAR[10]	I/O	Запрос канала 10 контроллера DMA
PB[30]	nDMAR[11]	I/O	Запрос канала 11 контроллера DMA
PB[31]	nDMAR[3]	I/O	Запрос канала 3 контроллера DMA

DMA контроллер имеет 13 каналов. Четыре канала позволяют выполнять универсальные пересылки типа «память-память». Восемь каналов выполняют пересылки типа «память-периферийное устройство». При этом четыре канала работают с устройствами-приемниками, а другие четыре – с устройствами-передатчиками. Один канал (12-й) является специализированным и может работать только с модулем цифрового коррелятора.

Прямой доступ к памяти (direct memory access, DMA) – механизм передачи данных без исполнения команд в ядре процессора. Встроенный контроллер DMA избавляет процессор от необходимости передачи данных между внутренней памятью и внешними устройствами\памятью или между внутренними периферийными устройствами и внешней или внутренней памятью. Он позволяет ядру процессора указать каналу команду передачи данных и вернуться к нормальной работе, тогда как контроллер DMA выполнит передачу данных в фоновом режиме.

Модуль DMA может быть ведущим или ведомым на шине SOC. При доступе к регистрам DMA со стороны процессорного ядра, модуль DMA выступает в роли ведомого устройства. При работе канала DMA по пересылке данных модуль DMA выступает в роли ведущего устройства и соревнуется в доступе к различным ресурсам с другими ведущими устройствами. Ресурсы, к которым посылает свои

запросы контроллер DMA, приведены ниже (Рисунок 19). Это интерфейсы внутренней или внешней памяти, а также устройства SOC-шины.

В данном подразделе используются следующие термины:

- *Блок управления передачей (TCB)* – блок из четырех слов, который определяет набор параметров для DMA-операций.
- *Регистр TCB* – 128-разрядный регистр, который содержит TCB.
- *Загрузка цепочки TCB* – процесс, при котором контроллер DMA загружает новый TCB из памяти и автоматически инициализирует регистр TCB.
- *Транзакция канала DMA* – последовательность действий канала по пересылке одного операнда.
- *Операнд канала* – порция данных (одно, два или четыре слова), размер которой задается регистром управления и которой оперирует канал в одной транзакции.
- *Режим квитирования* – режим работы канала, в котором инициатором одной транзакции выступает запрос от устройства.
- *Выровненный адрес* – означает, что для двойного слова (64 бита) младший бит адреса всегда равен 0, а для квадрослова (128 бит) два младших разряда адреса должны быть равны нулю.

13.1 Архитектура контроллера DMA

Контроллер DMA включает в себя 13 каналов, предназначенных для выполнения различных типов передачи:

- каналы с 0 по 3 являются универсальными и позволяют задавать произвольный источник и приемник, которые имеют адрес в карте памяти процессора.
- каналы с 4 по 7 жестко закреплены за внутренними периферийными устройствами, которые выполняют функции передатчиков данных (выдача данных на внешние контакты). Т.е. источник данных этих каналов DMA программируемый (внутренняя или внешняя память). Для определения приемника существует два режима работы. При выборе первого режима работы приемник данных и инициатор запросов DMA один и тот же и однозначно определяется значением в регистре DMACFGL. При выборе второго режима работы инициатор запросов определяется регистром DMACFGL, приемник данных – специальным регистром DCA (подробнее см. описание ниже). Каналы с 4 по 7 могут выполнять только пересылки память-устройство.
- каналы с 8 по 11 жестко закреплены за внутренними периферийными устройствами, которые выполняют функции приемников данных (прием информации с внешних контактов). Т.е. приемник данных программируемый (внутренняя или внешняя память). Для определения источника данных существует два режима работы. При выборе первого режима работы источник данных и инициатор запросов DMA один и тот же и однозначно определяется значением в регистре DMACFGH. При выборе второго режима работы инициатор запросов определяется регистром DMACFGH, источник данных – специальным регистром DCA (подробнее см. описание ниже). Каналы с 8 по 11 могут выполнять только пересылки устройство-память.
- канал 12 может быть использован совместно с цифровым коррелятором и позволяет организовать передачу данных от 16 каналов коррелятора во внутреннюю память процессора.

Передачи DMA характеризуются направлением потока данных от источника к приемнику. Если в роли источников или приемников выступает память, они описываются регистром ТСВ. Универсальные каналы с 0-го по 3-й имеют два ТСВ-регистра для описания источника и приемника, каналы с 4-го по 12-й имеют – по одному регистру ТСВ и регистр DCA. Рисунок 48 показывает блок-схему DMA-контроллера.

DMA поддерживает так называемые цепочки операций, когда одна запрограммированная в канале пересылка после завершения работы вызывает автоматическое чтение из внутренней памяти новой команды пересылки без участия процессора. Эта техника в первую очередь используется для одного DMA-канала, но в некоторых случаях может использоваться и для разных каналов.

Процессор также имеет четыре контакта внешнего запроса DMA (nDMAR3–0), которые позволяют внешним устройствам ввода\вывода запрашивать DMA-сервис. В ответ на внешний запрос по входу nDMARi, соответствующий i-й канал контроллера DMA выполняет передачу в соответствии с его инициализацией. Для инициализации канала программа должна записать в блок управления передачей (ТСВ) командное слово.

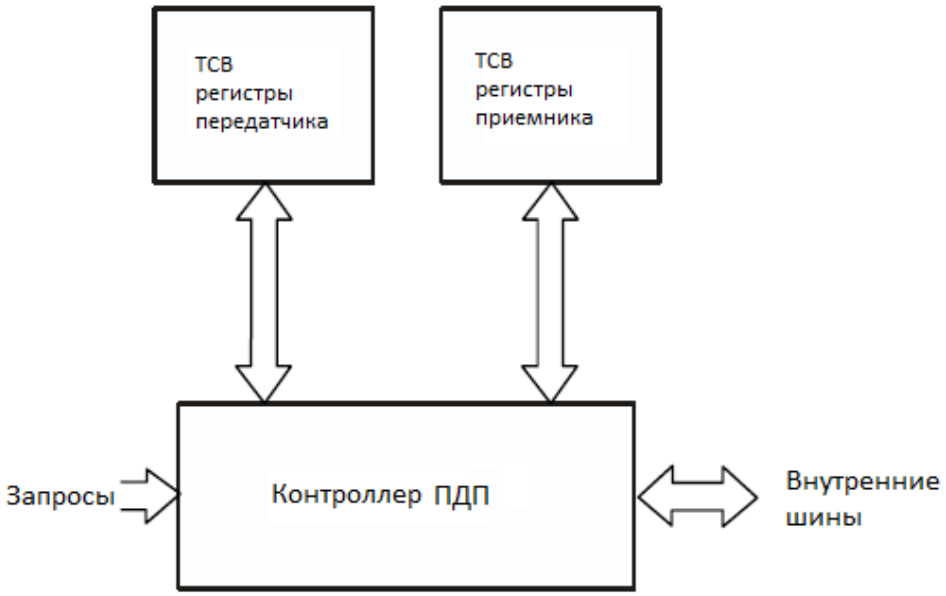


Рисунок 48 – Блок-схема DMA

Блок управления передачей ТСВ является регистром счетверенного слова (128 бит), который содержит информацию, необходимую для работы канала DMA. Вид регистра ТСВ представлен ниже (Рисунок 49). В одном канале может присутствовать приемник и передатчик, только приемник или только передатчик. Названия ТСВ передатчика и приемника имеют внутриканальный смысл: ТСВ передатчика передает данные ТСВ приемнику. Сам же ТСВ передатчик берет данные из источника информации, а ТСВ приемника передает принятые данные в пункт назначения.

В случае ТСВ передатчика, четыре слова содержат данные адреса источника информации (DI), количество слов для передачи (DX/DY), приращение адреса (DX/DY), контрольные разряды (DP).

В случае ТСВ приемника эти четыре слова содержат адрес назначения (DI), количество слов, которое нужно получить (DX/DY), приращение адреса (DX/DY), контрольные разряды (DP).

Рисунок 49 – Регистры блока управления передачей DMA

Одна передача (пересылка или транзакция) при работе DMA означает копирование порции данных (операнда) из источника данных в приемник данных. Размер порции данных описывается в регистре управления DP и может быть равен одному, двум или четырем 32-разрядным словам. В регистре DX задается количество слов для передачи. Количество транзакций будет равно количеству слов, деленному на размер порции данных. После каждой транзакции канал осуществляет изменение адреса на величину приращения, задаваемую в регистре DX(DY). Канал DMA может поддерживать двухмерные пересылки данных, когда адрес изменяется по горизонтальной составляющей X и по вертикальной составляющей У. Поэтому в ТСВ присутствуют описатели DX и DY. Обычный режим передачи – одномерный, используется только регистр DX. Регистр DY используется только в двухмерном режиме.

Особенность дополнительного DCA регистра адреса устройства по сравнению, например, с каналами 0-3 состоит в том, что это не полноценный ТСВ регистр, а 64-разрядный регистр, хранящий адрес устройства и некоторую управляющую информацию. Обращение к паре регистров возможно только как к длинному слову. Младший (четный) регистр содержит адрес устройства. Старший регистр содержит управляющую информацию.

Аналогично ТСВ регистру, введем обозначения – DCA:DI и DCA:DP. DCA:DI определяет адрес периферийного устройства. Невозможны никакие модификации этого адреса в процессе выполнения обмена. Он всегда постоянный. Невозможны изменения значения этого регистра при выполнении цепочки операций. Также введены ограничения на разрядность регистра DCA:DI: для каналов 4-7 разрядность регистра 16 бит (старшие 16 бит 32-разрядного адреса предполагаются равными 0x8000), для каналов 8-11 разрядность регистра 32 бита. Регистр DCA:DP использует только три бита – с 31 по 29-й. Любое ненулевое значение в этих разрядах (рекомендуется использовать значение 001b) вызывает включение нового регистра в использование вместо старой функции. После сброса новый регистр управления равен нулю и его функции выключены.

Контроллер поддерживает словную адресацию памяти, и минимальной адресуемой единицей данных для него является 32-разрядное слово.

Соответствие между номером канала и его характеристиками приведено ниже (Таблица 82).

Таблица 82 – Особенности программирования каналов контроллера

Номер канала	Источник данных	Приемник данных	Инициирование транзакции
0-3	Программируется в ТСВ передатчика	Программируется в ТСВ приемника	Программируется. Может использовать запрос устройства

Номер канала	Источник данных	Приемник данных	Инициирование транзакции
4-7	Программируется в ТСВ	Режим 1. Периферийное устройство – инициатор транзакции. Номер устройства задается в регистре конфигурации DMACFGx. Режим 2. Адрес устройства задается в специальном регистре DCA	Всегда только по запросу периферийного устройства. Номер устройства задается в регистре конфигурации DMACFGx
8-11	Режим 1. Периферийное устройство – инициатор транзакции. Номер устройства задается в регистре конфигурации DMACFGx. Режим 2. Адрес устройства задается в специальном регистре DCA	Программируется в ТСВ	Всегда только по запросу периферийного устройства. Номер устройства задается в регистре конфигурации DMACFGx
12	Запись данных цифровым коррелятором в регистр канала	Программируется в ТСВ приемника	Всегда при наличии данных во внутреннем буфере

Каналы контроллер DMA не однотипны. Каналы 0-3 позволяют выполнять пересылки: память-память, память-устройство, устройство-память. Каналы 4-7 позволяют выполнять только пересылки память-устройство, т.е. работают только с устройствами-передатчиками. Каналы 8-11 позволяют выполнять только пересылки устройство-память, т.е. работают только с устройствами-приемниками. При необходимости работы с внутренними устройствами с помощью каналов 0-3 возникают дополнительные ограничения. Каналы 0 и 2 могут быть запрограммированы только для работы с устройствами-передатчиками, а каналы 1 и 3 только с устройствами-приемниками.

Контроллер DMA может быть ведомым на SOC-шине устройств, и его внутренние ресурсы могут быть доступны процессору посредством чтения и записи регистров DMA. Доступ к регистрам контроллера DMA может быть выполнен как с помощью команд пересылки процессора (используя номер регистра), так и с помощью команд загрузки и сохранения (используя адрес в карте памяти).

13.1.1 Регистры управления и статуса

Базовый адрес регистров контроллера в карте памяти периферийных устройств равен **0x8000_0000**. В таблицах описания регистров указано смещение относительно базового адреса.

Общее описание группы регистров управления и статуса приведено ниже (Таблица 83).

Таблица 83 – Группа регистров управления контроллер DMA

Имя	Описание	Адрес-смещение	Значение по умолчанию
DCNT	Управление контроллер DMA	0x 0060	0x0000 0000
-	-	0x 0061–0x 0063	-
DCNTST	Управление контроллер DMA, биты установки	0x 0064	Не определено

Имя	Описание	Адрес-смещение	Значение по умолчанию
-	-	0x 0065–0x 0067	-
DCNTCL	Управление контроллер DMA, биты сброса	0x 0068	Не определено
-	-	0x 0069–0x 006B	-
DSTAT	Статус контроллер DMA	0x 006C	0x0000 0000
-	-	0x 006D–0x 006F	-
DSTATCL	Статус контроллер DMA, биты сброса	0x 0070	Не определено
-	-	0x 0071–0x 0077	-
DMACFGL	Задание номера устройства Для каналов с 0 по 7	0x0078	0
DMACFGH	Задание номера устройства Для каналов с 8 по 15	0x0079	0
-	-	0x 007A–0x 007F	-

Особенно отметим функции регистров DMACFGx (DMACFGL, DMACFGH). В процессоре имеется несколько периферийных устройств, способных поддерживать работу совместно с контроллером DMA. Каждому устройству в контроллере DMA ставится в соответствие номер. Для того чтобы некоторому каналу DMA поставить в соответствие выбранное устройство, в определенном поле регистра DMACFGx осуществляется программирование номера устройства. Для каждого канала в регистре конфигурации отведено четыре бита, что позволяет задать 16 источников запросов на обмен (биты 0-3 регистра DMACFGL соответствуют каналу 0, биты 4-7 каналу 1, биты 8-11 каналу 2 и т. д). Регистр DMACFGH обслуживает каналы с 8 по 11. После сброса значение регистра равно нулю и соответствует начальному варианту работы контроллер DMA. Значение в поле регистра соответствует номеру источника запроса.

Имеются две группы источников запросов: передатчики (Таблица 84) и приемники (Таблица 85).

Таблица 84 – Устройства-передатчики

Номер	Обслуживаемый запрос	Размер данных для обмена
0	Источник запроса определяется номером канала КПДП: 0 – nDMAR0 1 – nDMAR1 2 – nDMAR2 3 – nDMAR3 4 – LINK 0 5 – LINK 1 6 – 7 – 8 – LINK 0 9 – LINK 1 10 – 11 –	-
1	Готовность передатчика UART_1 принять данные	32 бита
2	Готовность передатчика UART_2 принять данные	32 бита
3	Готовность передатчика SPI0 принять данные	32 бита
4	Готовность LCD контроллера принять данные	128 бит
5	Готовность передатчика SSI0 контроллера принять данные	32 бита

Номер	Обслуживаемый запрос	Размер данных для обмена
6	Готовность передатчика SSI1 контроллера принять данные	32 бита
7	Готовность буфера NANDF контроллера принять данные для записи	32 или 128 (определяется контроллером NANDF)
8	Готовность модуля UP\DOWN1 принять данные	128 бит
9	Готовность модуля UP\DOWN0 принять данные	128 бит
10	Готовность модуля UP\DOWN2 принять данные	128 бит
11	Готовность модуля UP\DOWN3 принять данные	128 бит
12	Готовность передатчика SPI1 принять данные	
13	Готовность передатчика SPI2 принять данные	
14	H264_RQ1	128 бит
15	H264_RQ0	128 бит
16	Внешний запрос nDMAR[4] (каналы 4-7) или nDMAR[8] (каналы 8-11)	
17	внешний запрос nDMAR[5] (каналы 4-7) или nDMAR[9] (каналы 8-11)	
18	Внешний запрос nDMAR[6] (каналы 4-7) или nDMAR[10] (каналы 8-11)	
19	Внешний запрос nDMAR[7] (каналы 4-7) или nDMAR[11] (каналы 8-11)	
20	Запрос 0 от таймера 0 с функцией Захвата/ШИМ	
21	Запрос 1 от таймера 0 с функцией Захвата/ШИМ	
22	Запрос 2 от таймера 0 с функцией Захвата/ШИМ	
23	Запрос 3 от таймера 0 с функцией Захвата/ШИМ	
24	Запрос 4 от таймера 0 с функцией Захвата/ШИМ	
25	Запрос 0 от таймера 1 с функцией Захвата/ШИМ	
26	Запрос 1 от таймера 1 с функцией Захвата/ШИМ	
27	Запрос 2 от таймера 1 с функцией Захвата/ШИМ	
28	Запрос 3 от таймера 1 с функцией Захвата/ШИМ	
29	Запрос 4 от таймера 1 с функцией Захвата/ШИМ	
30	Запрос от таймера 0 контроллера прерываний	
31	Запрос от таймера 1 контроллера прерываний	

Использование номера запроса от 16 и выше требует определения адреса в дополнительных регистрах каналов.

Далее рассмотрена ситуация при работе с несколькими каналами 0-3, некоторые из которых работают по запросам таймера:

1 Все каналы имеют одинаковый приоритет (не важно – высокий или стандартный) – тогда, каналы, работающие по запросам таймеров, начнут передавать данные только после того, как закончат работу все остальные каналы. При этом, в случаи с каналами 0-3 запросы копятся, и поэтому по завершению работы каналов без запросов, каналы с запросами отправят столько данных сколько запросов накопилось за время их простоя. А затем, каналы с запросами таймеров начнут работать непосредственно по самим запросам таймера. Стоит отметить, что после включения каналов с запросами таймеров, даже во время простоя, статус у этих каналов все равно равен 1.

2 Каналы с запросами имеют высокий приоритет, все остальные – стандартный. В этом случае, каналы с запросами от таймеров работают параллельно с остальными каналами.

Таблица 85 – Устройства-приемники

Номер	Обслуживаемый запрос	Размер данных для обмена
0	Источник запроса определяется номером канала КПДП: 0 – nDMAR0 1 – nDMAR1 2 – nDMAR2 3 – nDMAR3 4 – LINK 0 5 – LINK 1 6 – 7 – 8 – LINK 0 9 – LINK 1 10 – 11 –	
1	Готовность приемника UART_1 выдать данные	32 бита
2	Готовность приемника UART_2 выдать данные	32 бита
3	Готовность приемника SPI0 выдать данные	32 бита
4	Готовность контроллера видеокамеры выдать данные	128 бит
5	Готовность приемника SSI0 контроллера выдать данные	32 бита
6	готовность приемника SSI1 контроллера выдать данные	32 бита
7	Готовность буфера NANDF контроллера выдать данные	32 или 128 (определяется контроллером NANDF)
8	Готовность модуля UP\DOWN1 выдать данные	128 бит
9	Готовность модуля UP\DOWN0 выдать данные	128 бит
10	Готовность модуля UP\DOWN2 выдать данные	128 бит
11	Готовность модуля UP\DOWN3 выдать данные	128 бит
12	Готовность приемника SPI1 выдать данные	
13	Готовность приемника SPI2 выдать данные	
14	H264_RQ1	128 бит
15	H264_RQ0	128 бит
16	Внешний запрос nDMAR[4] (каналы 4-7) или nDMAR[8] (каналы 8-11)	
17	Внешний запрос nDMAR[5] (каналы 4-7) или nDMAR[9] (каналы 8-11)	
18	Внешний запрос nDMAR[6] (каналы 4-7) или nDMAR[10] (каналы 8-11)	
19	Внешний запрос nDMAR[7] (каналы 4-7) или nDMAR[11] (каналы 8-11)	
20	Запрос 0 от таймера 0 с функцией Захвата/ШИМ	
21	Запрос 1 от таймера 0 с функцией Захвата/ШИМ	
22	Запрос 2 от таймера 0 с функцией Захвата/ШИМ	
23	Запрос 3 от таймера 0 с функцией Захвата/ШИМ	
24	Запрос 4 от таймера 0 с функцией Захвата/ШИМ	
25	Запрос 0 от таймера 1 с функцией Захвата/ШИМ	
26	Запрос 1 от таймера 1 с функцией Захвата/ШИМ	
27	Запрос 2 от таймера 1 с функцией Захвата/ШИМ	
28	Запрос 3 от таймера 1 с функцией Захвата/ШИМ	
29	Запрос 4 от таймера 1 с функцией Захвата/ШИМ	
30	Запрос от таймера 0 контроллера прерываний	
31	Запрос от таймера 1 контроллера прерываний	

Каждое из устройств имеет для передачи и (или) для приема встроенный буфер FIFO. При работе с устройствами с помощью каналов 0-3 необходимо задать источник запроса с помощью регистра конфигурации, а также обязательно прописать адрес источника или приемника в соответствующем регистре канала. При работе с каналами 4-11 программируется только номер источника запроса. Особенным образом должны обрабатываться запросы H264_RQ[1:0]. Модули, формирующие эти запросы, находятся в процессорном ядре и имеют адрес в адресном пространстве внутренней памяти. Поэтому данные запросы могут обрабатываться только с помощью каналов DMA с 0-го по 3-й.

13.1.2 Группа регистров TCB каналов 0-3 контроллер DMA

Доступ к регистрам данной группы осуществляется только как к счетверенным словам, т.е. операции выполняются сразу с четырьмя регистрами.

Таблица 86 – Группа регистров TCB каналов 0-3

Имя	Описание	Адрес-смещение	Значение по умолчанию
DCS0:DI	канал 0, адрес источника	0x 0000	0
DCS0:DX	канал 0, модификатор X источника	0x 0001	0x01000004
DCS0:DY	канал 0, модификатор У источника	0x 0002	0
DCS0:DP	канал 0, управление источником	0x 0003	0
DCD0:DI	канал 0, адрес приемника	0x 0004	0
DCD0:DX	канал 0, модификатор X приемника	0x 0005	0x01000001
DCD0:DY	канал 0, модификатор У приемника	0x 0006	0
DCD0:DP	канал 0, управление приемником	0x 0007	0
DCS1:DI	канал 1, адрес источника	0x 0008	0
DCS1:DX	канал 1, модификатор X источника	0x 0009	0x01000004
DCS1:DY	канал 1, модификатор У источника	0x 000A	0
DCS1:DP	канал 1, управление источником	0x 000B	0
DCD1:DI	канал 1, адрес приемника	0x 000C	0
DCD1:DX	канал 1, модификатор X приемника	0x 000D	0x01000001
DCD1:DY	канал 1, модификатор У приемника	0x 000E	0
DCD1:DP	канал 1, управление приемником	0x 000F	0
DCS2:DI	канал 2, адрес источника	0x 0010	0
DCS2:DX	канал 2, модификатор X источника	0x 0011	0x01000004
DCS2:DY	канал 2, модификатор У источника	0x 0012	0
DCS2:DP	канал 2, управление источником	0x 0013	0
DCD2:DI	канал 2, адрес приемника	0x 0014	0
DCD2:DX	канал 2, модификатор X приемника	0x 0015	0x01000001
DCD2:DY	канал 2, модификатор У приемника	0x 0016	0
DCD2:DP	канал 2, управление приемником	0x 0017	0
DCS3:DI	канал 3, адрес источника	0x 0018	0
DCS3:DX	канал 3, модификатор X источника	0x 0019	0x01000004
DCS3:DY	канал 3, модификатор У источника	0x 001A	0
DCS3:DP	канал 3, управление источником	0x 001B	0
DCD3:DI	канал 3, адрес приемника	0x 001C	0
DCD3:DX	канал 3, модификатор X приемника	0x 001D	0x01000001
DCD3:DY	канал 3, модификатор У приемника	0x 001E	0
DCD3:DP	канал 3, управление приемником	0x 001F	0

Особенности при работе с несколькими каналами 0-3, некоторые из которых работают по запросам таймера:

1. В случае, когда все каналы имеют одинаковый приоритет (высокий или стандартный), каналы, работающие по запросам таймеров, начнут передавать данные только после того, как закончат работу все остальные каналы. При этом, в случаи с каналами 0-3 запросы копятся, и по завершении работы каналов без запросов, каналы с запросами отправят столько данных сколько запросов накопилось за время их простоя. Затем, каналы с запросами таймеров, начнут работать непосредственно по самим запросам таймера. После включения каналов с запросами таймеров, даже во время простоя, статус у этих каналов все равно равен 1.
2. В случае, когда каналы с запросами имеют высокий приоритет, все остальные – стандартный, каналы с запросами от таймеров, работают параллельно с остальными каналами.

13.1.3 Группа регистров ТСВ каналов 4-7 контроллер DMA

Доступ к регистрам DCx осуществляется только как к счетверенным словам, т.е. операции выполняются сразу с четырьмя регистрами. Доступ к регистрам DCAx осуществляется только как к двойным словам, т.е. операции выполняются сразу с двумя регистрами.

Таблица 87 – Группа регистров ТСВ каналов 4-7

Имя	Описание	Адрес-смещение	Значение по умолчанию
DC4:DI	канал 4, адрес источника	0x 0020	Не определено
DC4:DX	канал 4, модификатор X источника	0x 0021	Не определено
DC4:DY	канал 4, модификатор У источника	0x 0022	Не определено
DC4:DP	канал 4, управление источником	0x 0023	0
DC5:DI	канал 5, адрес источника	0x 0024	Не определено
DC5:DX	канал 5, модификатор X источника	0x 0025	Не определено
DC5:DY	канал 5, модификатор У источника	0x 0026	Не определено
DC5:DP	канал 5, управление источником	0x 0027	0
DC6:DI	канал 6, адрес источника	0x 0028	Не определено
DC6:DX	канал 6, модификатор X источника	0x 0029	Не определено
DC6:DY	канал 6, модификатор У источника	0x 002A	Не определено
DC6:DP	канал 6, управление источником	0x 002B	0
DC7:DI	канал 7, адрес источника	0x 002C	Не определено
DC7:DX	канал 7, модификатор X источника	0x 002D	Не определено
DC7:DY	канал 7, модификатор У источника	0x 002E	Не определено
DC7:DP	канал 7, управление источником	0x 002F	0
DCA4:DI	Канал 4, адрес устройства-приемника	0x0030	Не определено
DCA4:DP	Канал 4, включение	0x0031	0
DCA5:DI	Канал 5, адрес устройства-приемника	0x0032	Не определено
DCA5:DP	Канал 5, включение	0x0033	0
DCA6:DI	Канал 6, адрес устройства-приемника	0x0034	Не определено
DCA6:DP	Канал 6, включение	0x0035	0
DCA7:DI	Канал 7, адрес устройства-приемника	0x0036	Не определено
DCA7:DP	Канал 7, включение	0x0037	0
		0x0038 – 0x003F	

Устройство всегда находится на СОК-шине внутренних периферийных устройств.

Значение 001b в разрядах DCA:DP[31:29] вызывает включение регистра DCA (регистр задания адреса периферийного устройства по которому можно выполнить запись данных). После сброса регистр DCA управления равен нулю и его функции выключены. Направление передачи данных при работе каналов 4-7 показано на Рисунке 50.

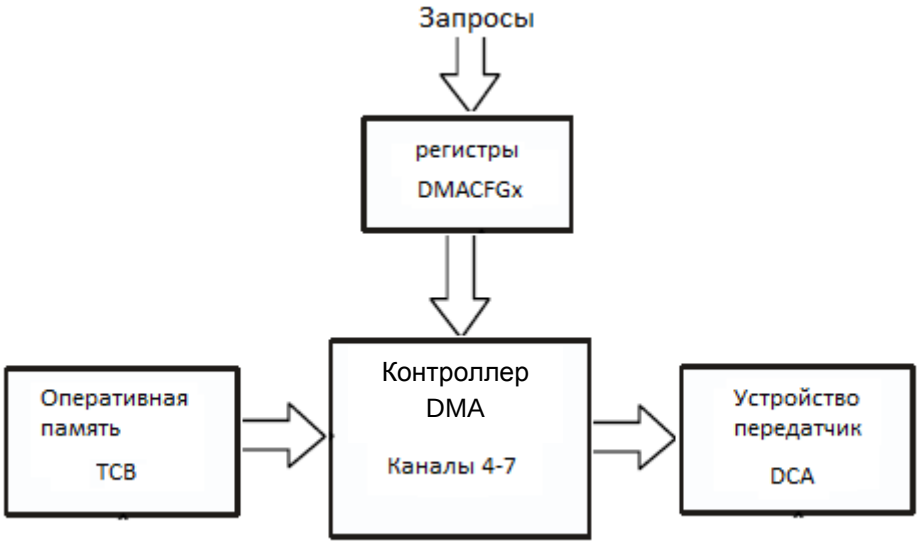


Рисунок 50 – Передача данных в каналах 4-7

Используя запросы от таймеров или иных источников, можно запрограммировать в каналах 4-7 произвольные пересылки из любого типа памяти в любое **внутреннее** устройство SOC-шины.

Если необходимо обращение к устройству-приемнику на внешней шине должны использоваться только каналы 0-3.

13.1.4 Группа регистров ТСВ каналов 8-11

Доступ к регистрам DCx осуществляется только как к счетверенным словам, т.е. операции выполняются сразу с четырьмя регистрами. Доступ к регистрам DCAx осуществляется только как к двойным словам, т.е. операции выполняются сразу с двумя регистрами.

Таблица 88 – Группа регистров ТСВ каналов 8-11

Имя	Описание	Адрес-смещение	Значение по умолчанию
DC8:DI	канал 8, адрес приемника	0x 0040	0
DC8:DX	канал 8, модификатор X приемника	0x 0041	0
DC8:DY	канал 8, модификатор У приемника	0x 0042	0
DC8:DP	канал 8, управление приемником	0x 0043	0
DC9:DI	канал 9, адрес приемника	0x 0044	0
DC9:DX	канал 9, модификатор X приемника	0x 0045	0
DC9:DY	канал 9, модификатор У приемника	0x 0046	0
DC9:DP	канал 9, управление приемником	0x 0047	0

DC10:DI	канал 10, адрес приемника	0x 0048	0
DC10:DX	канал 10, модификатор X приемника	0x 0049	0
DC10:DY	канал 10, модификатор У приемника	0x 004A	0
DC10:DP	канал 10, управление приемником	0x 004B	0
DC11:DI	канал 11, адрес приемника	0x 004C	0
DC11:DX	канал 11, модификатор X приемника	0x 004D	0
DC11:DY	канал 11, модификатор У приемника	0x 004E	0
DC11:DP	канал 11, управление приемником	0x 004F	0
DCA8:DI	Канал 8, адрес устройства-источника	0x0050	0
DCA8:DP	Канал 8, включение	0x0051	0
DCA9:DI	Канал 9, адрес устройства- источника	0x0052	0
DCA9:DP	Канал 9, включение	0x0053	0
DCA10:DI	Канал 10, адрес устройства- источника	0x0054	0
DCA10:DP	Канал 10, включение	0x0055	0
DCA11:DI	Канал 11, адрес устройства- источника	0x0056	0
DCA11:DP	Канал 11, включение	0x0057	0

После сброса регистр DCA:DP каналов 8-11 равен нулю, и новые функции выключены. Если записать в регистр DCA:DP ненулевую информацию, то канал 8-11 будет использовать в качестве адреса источника значение из регистра DCA:DI.

Значение, записываемое в регистр управления DCA:DP имеет, кроме функции включения, дополнительный смысл. Если поле управление равно 4, обращение происходит к устройству на СОК-шине (если адрес соответствует СОК-шине) или на внешнюю шину процессора. Если значение поля равно 1, то обращение выполняется к памяти ядра процессора. Таким образом, каналы 8-11 могут выполнять пересылку «устройство_на_внешней_шине – память». Значение регистра DCA:DI в процессе работы канала не изменяется. Направление передачи данных при работе каналов 8-11 показано на Рисунке 51.

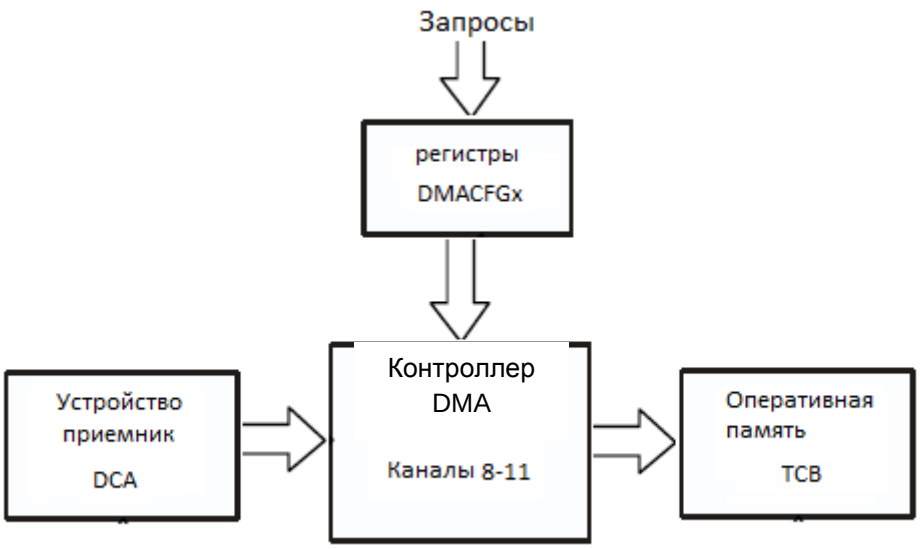


Рисунок 51 – Передача данных в каналах 8-11

Если каналы 4-11 завершают свою работу в нормальном режиме, т.е. по завершении происходит очистка управляющего поля регистра DSP, DDP, то и управляющее поле дополнительного регистра DCA:DP также сбрасывается.

В противном случае необходима дополнительная очистка регистра. Запись в регистр DCA никак каналом не отслеживается и не влияет на последующее включение-выключение канала.

Последовательность использования функций каналов 4-11 должна быть следующей:

1. При выключенном канале необходимо записать в регистр конфигурации DMACFGx номер источника запроса. Используются 5 бит номера. Старшие биты расширения номера хранятся в разрядах 27-16 регистра DMACFGH. По одному для каждого из каналов 11-0.
2. В регистре DCA выбранного канала необходимо записать адрес устройства и значение управляющего кода.
3. Включить канал в работу стандартным образом, т.е. записать активное значение TCB канала.

Назначение разрядов регистра DMACFGx каналам контроллера описано в Таблице 89. Поле H относится к регистру DMACFGH, а поле L – к регистру DMACFGL.

Таблица 89 – Назначение разрядов регистра DMACFGx

Канал	Биты DMACFGx
0	H[16].L[3:0]
1	H[17].L[7:4]
2	H[18].L[11:8]
3	H[19].L[15:12]
4	H[20].L[19:16]
5	H[21].L[23:20]
6	H[22].L[27:24]
7	H[23].L[31:28]
8	H[24].H[3:0]
9	H[25].H[7:4]
10	H[26].H[11:8]
11	H[27].H[15:12]

13.1.5 Группа регистров TCB канала 12

Таблица 90 – Группа регистров TCB канала 12

Имя	Описание	Адрес-смещение	Значение по умолчанию
DC12:DI	Канал 12, адрес приемника	0x 0058	0x0000 0000
DC12:DX	Канал 12, модификатор X приемника	0x 0059	0x0100 0001
DC12:DY	Канал 12, модификатор У приемника	0x 005A	0x0000 0000
DC12:DP	Канал 12, управление приемником	0x 005B	0x5380 0000

Канал 12 также называется AutoDMA-каналом в связи с тем, что при выполнении записи данных в него ведущим устройством, он автоматически выполняет пересылку принятых данных в запрограммированный в него приемник. Для выполнения операций записи в канал 12 используются определенные регистры.

13.1.6 Группа регистров AutoDMA

Регистры данной группы относятся к каналу 12 контроллера DMA и используются для реализации режима «подчиненный» (slave) DMA. Данные регистры могут быть доступны через адресное пространство периферийных устройств.

Таблица 91 – Группа регистров AutoDMA

Имя	Описание	Адрес-смещение	Значение по умолчанию
AUTODMA0:0	Регистр AutoDMA 0, разряды данных 31–0	0x 03E0	Не определено
AUTODMA0:1	Регистр AutoDMA 0, разряды данных 63–32	0x 03E1	Не определено
AUTODMA0:2	Регистр AutoDMA 0, разряды данных 95–64	0x 03E2	Не определено
AUTODMA0:3	Регистр AutoDMA 0, разряды данных 127–96	0x 03E3	Не определено
-	-	0x 03E4– 0x 03FF	-

13.2 Настройка DMA-передач

DMA-операции могут программироваться ядром процессора или внешним хост-устройством. Операция программируется записью в регистры TCB. Регистр TCB – регистр счетверенных слов, описывающий передачу DMA-блока. DMA-канал настраивается записью счетверенного слова в каждый из регистров TCB. Каждый регистр должен быть загружен начальным адресом для блока, приращением адреса, количеством слов и управляющей информацией.

Если TCB запрограммирован на генерацию прерывания, запрос прерывания формируется, когда блок данных полностью передан.

Регистры TCB могут быть доступны только как счетверенные слова.

13.2.1 Регистры блока управления передачей (TCB)

Каждый TCB-регистр имеет длину 128 бит и разделен на четыре 32-битных регистра:

- регистр индекса (DI);
- регистр X количества и приращения (DX);
- регистр Y количества и приращения (DY);
- регистр управления и указателя цепочки (DP).

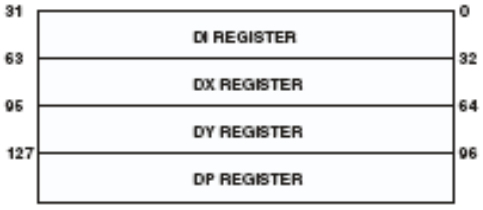


Рисунок 52 – Регистры блока управления передачей

Эти четыре регистра образуют счетверенное слово TCB, которое может быть загружено как выровненное счетверенное слово из внутренней памяти посредство цепочки или с помощью ядра.

Для инициирования нового обмена, после завершения текущего, программа должна записать новые параметры в регистры TCB.

Таблица 92 – Каналы DMA и связанные регистры TCB

Канал DMA	Регистры TCB	Описание
0	DCS0 DCD0	TCB регистр источника канала 0 TCB регистр назначения канала 0
1	DCS1 DCD1	TCB регистр источника канала 1 TCB регистр назначения канала 1
2	DCS2 DCD2	TCB регистр источника канала 2 TCB регистр назначения канала 2
3	DCS3 DCD3	TCB регистр источника канала 3 TCB регистр назначения канала 3
4	DC4	TCB регистр источника канала 4
5	DC5	TCB регистр источника канала 5
6	DC6	TCB регистр источника канала 6
7	DC7	TCB регистр источника канала 7
8	DC8	TCB регистр назначения канала 8
9	DC9	TCB регистр назначения канала 9
10	DC10	TCB регистр назначения канала 10
11	DC11	TCB регистр назначения канала 11
12	DC12	TCB регистр назначения канала 12

Регистр DI

32-разрядный регистр DI содержит начальный адрес блока данных и используется для прямого доступа к памяти. Он может указывать на адрес внешней памяти или внутренней памяти.

Регистр DX

32-разрядный регистр DX имеет два поля:

- DXM – младшие биты (с 0-го по 15-й или с 0-го по 21-й) хранят значение модификатора адреса (DX_MODIFY), которое используется для изменения адреса после каждой транзакции.
- DXC – старшие биты (с 16-го по 31-й или с 22-го по 31-й) хранят значение количества слов (DX_COUNT) в блоке данных который необходимо передать.

Например, если необходимо передать 16 слов данных, непрерывно размещенных в памяти, порциями по 4 слова в каждой передаче, параметр DXC будет равен 16, а значение DXM будет равно 4. Длина операнда (порции данных) в регистре DP устанавливается равной счетверенному слову.

Существуют ограничения, которые следует принимать во внимание при программировании TCB:

- указатель DI должен содержать адрес, выровненный на границе операнда, определяемого в регистре DP;
- значение DXM должно быть кратно размеру операнда;
- значение DXC должно быть кратным длине операнда.

Выбор длины полей модификатора и количества определяется типом обмена, задаваемым полем TY в регистре DP.

Регистр DY

32-разрядный регистр DY используется вместе с регистром DX, когда двумерный режим DMA разрешен. Регистр DY содержит два поля:

- DYM – младшие биты (с 0-го по 15-й или с 0-го по 21-й) хранят значение модификатора адреса (DY_MODIFY), которое используется для изменения адреса после завершения передачи блока по координате X. Модификация Y – разница между адресом последнего элемента данных в строке и адресом первого элемента в следующей строке.
- DYC – старшие биты (с 16-го по 31-й или с 22-го по 31-й) хранят значение количества попыток (DY_COUNT) передач по координате X.

Если двумерный (2D) режим DMA запрещен, регистр DY загружается значением 0 или любым другим допустимым значением.

Выбор длины полей модификатора и количества определяется типом обмена, задаваемым полем TY в регистре DP.

Регистр DP

Данный регистр задает режим работы передатчика\приемника канала. Подробное описание разрядов регистра приведено ниже (Таблица 93).

Таблица 93 – Описание разрядов регистра DP

Бит	Имя	Назначение
18:0	CHPT	Указатель цепочки. Эти разряды включают в себя разряды 20–2 адреса внутренней памяти, где находится значение следующего регистра TCB. Разряды 1–0 адреса не указываются, поскольку адрес должен быть выровненным на границе квадрослова, т.е биты равны нулю. Например, если содержимое следующего TCB сохраняется в памяти 0x87128–0x8712B тогда поле CHPT будет содержать b0010_0001_1100_0100_1010 (0x21C4A)
21:19	CHTG	Канал-приемник следующей цепочки. Это поле имеет значение *только для каналов 4-11* и указывает в какой канал DMA будет загружено новое значение TCB: 000 – канал 8 001 – канал 9 010 – канал 10 011 – канал 11 100 – канал 4 101 – канал 5 110 – канал 6 111 – канал 7
22	CHEN	Разрешение загрузки следующей цепочки: 1 – разрешено 0 – запрещено При разрешении загрузки после окончания передачи всего блока данных канал выполнит чтение новой TCB из внутренней памяти и загрузит ее в канал назначения

Бит	Имя	Назначение
23	DRQ	Разрешение анализа запроса от периферийного устройства: 1 – разрешено 0 – запрещено Если разрешено, канал выполняет одну транзакцию только по запросу. Если запрещено, канал выполняет передачу всего блока без ожидания запроса. Бит имеет значение для каналов 0-3 и позволяет им анализировать запросы от устройств. Для каналов 4-11 бит не имеет значения, т.к. они всегда работают только по запросу
24	INT	Генерация запроса прерывания после окончания работы канала: 1 – разрешено 0 – запрещено
26:25	LEN	Длина передаваемых данных (операнда) в одном цикле обмена: 00 – резерв 01 – слово 32 бита 10 – длинное слово 64 бита 11 – квадрослово 128 бит Длина операнда накладывает ограничения на адрес источника или приемника – они всегда должны быть выравнены. Также накладываются ограничение на модификаторы и количество передаваемых слов в X
27	2D	Включение режима 2-х мерной пересылки: 1 – двумерная пересылка 0 – одномерная пересылка
28	PR	Приоритет циклов обмена: 1 – высокий 0 – обычный Уровень приоритета используется во время арбитража каналов между собой. Также уровень приоритета оказывает влияние на контакт DPA внешней шины
31:29	TY	Тип обмена (выбор источника или приемника): 000 – канал выключен 001 – линк порт (только в случае передачи линк-линк) 010 – внутренняя память (16\16) 011 – внутренняя память (22\10) 100 – внешняя память (16\16) 101 – зарезервировано 110 – загрузочное EPROM. nBMS контакт 111 – внешняя память (22\10)

Ограничения при программировании регистра DP

Неправильное задание конфигурации TCB может вызвать генерацию ошибки каналом. На использование *регистра управления TCB* накладываются *следующие ограничения*:

- Установка длины операнда (LEN). В каналах 0-3 поле LEN должно быть одинаковым в обоих TCB.
- Если поле TY источника или получателя TCB имеет тип 6 (загрузочная EPROM), поле LEN должно быть 0x1 (стандартное слово).

Задание полей количества (DX и DY) имеет следующие ограничения:

- Для каналов 0-3 общее количество слов передатчика должно равняться количеству слов приемника.
- Для обычного режима обмена количество слов равно DXC.

- Для двухмерного режима количество слов равно DXC•DYC. При этом бит 2D может быть установлен в одном из TCB и сброшен в другом (для каналов с 0-го по 3-й).
- Нет ограничений для регистра количества в DMA-каналах 4-11. Во всех случаях, если канал не простаивает, DXC не может быть нулевым. Если 2D режим установлен, DYC не может быть 0 или 1

Из поля TY регистра управления видно, что можно задать семь типов обмена. Однако в зависимости от номера канала существует *ограничения в выборе типа*. Разрешенные комбинации (отмечены +) типов в каналах 0-3 приведены ниже (Таблица 94).

Таблица 94 – Допустимые комбинации передач каналов с 0 по 3

Тип передатчика	Тип приемника						
	1	2	3	4	5	6	7
1							
2				+		+	+
3				+		+	+
4		+	+	+		+	+
5							+
6		+	+				
7		+	+	+	+	+	+

Все другие комбинации при их установке в каналах 0-3 вызовут генерацию ошибки канала.

Для каналов 4-7 также существует ограничение в выборе типа обмена: в качестве источника данных могут быть выбраны только внутренняя (тип 2, 3) или внешняя (тип 4, 7) память.

Ограничения в выборе типа для каналов 8-11: в качестве приемника данных могут быть выбраны только внутренняя (2, 3), внешняя (4, 7) память или порт связи (тип 1).

Для канала 12 единственным типом может быть только внутренняя память (тип 2, 3).

Ограничения запросов DMA

В каналах 0-3 бит DRQ должен быть одинаковым в обоих регистрах TCB.

Ограничения выравнивания

Если поле LEN равно 2 (двойное слово), следующие поля должны быть четными (бит 0 очищен):

- DI;
- DXM;
- DYM (при необходимости);
- DXC.

Если поле LEN равно 3 (счетверенное слово) значения этих же полей должны быть кратны 4 (биты 1–0 очищены).

Для типа 6 (загрузочная EPROM) данные регистры должны быть все кратны 4 (биты 1–0 очищены).

Ограничения диапазона адресов

Следующая связь должна существовать в любой момент работы TCB между полем TY в регистре DP и адресом:

- TY = 1 (канал связи) ⇒ адрес – один из регистров передачи порта связи.
- TY = 2 или 3 (внутренняя память) ⇒ адрес – в диапазоне внутренней памяти и не в регистрах.
- TY = 4 или 7(внешняя память) ⇒ адрес – в диапазоне внешней памяти (биты 31–22 отличны от нуля).

13.2.2 Регистр статуса DMA (DSTAT/DSTATC)

Регистр статуса DSTAT предназначен только для чтения и отражает текущее состояние каналов контроллера. Регистр имеет разрядность 64 бита и два адреса доступа. Второму адресу соответствует имя DSTATCL регистра. При чтении регистра DSTATCL все коды ошибок в регистре статуса очищаются, и состояние канала изменяется на состояние выключено. Регистры статуса DMA (DSTAT, DSTATCL) могут быть доступны только как двойное или счетверенное слово. В регистре состояния каждому каналу контроллера отведено 3 бита. С помощью этих бит кодируется состояние канала. Возможные значение состояния канала:

- 000 – выключен;
- 001 – активен (работает);
- 010 – завершил работу;
- 011 – резерв;
- 100 – ошибка. Инициализация активного канала;
- 101 – ошибка. Запрещенная конфигурация канала;
- 110 – резерв;
- 111 – ошибка. Некорректный адрес.

Назначение разрядов регистра состояния приведено ниже (Таблица 95).

Таблица 95 – Биты регистра состояний

Бит	Имя	Описание
2:0	CH0	Состояние канала 0
5:3	CH1	Состояние канала 1
8:6	CH2	Состояние канала 2
11:9	CH3	Состояние канала 3
14:12	CH4	Состояние канала 4
17:15	CH5	Состояние канала 5
20:18	CH6	Состояние канала 6
23:21	CH7	Состояние канала 7
31:24	-	-
34:32	CH8	Состояние канала 8
37:35	CH9	Состояние канала 9
40:38	CH10	Состояние канала 10
43:41	CH11	Состояние канала 11
49:44	-	-
52:50	CH12	Состояние канала 12
63:53	-	-

Активный статус DMA устанавливается, когда канал DMA включен (когда поле TY регистра DPx установлено).

Статус завершения DMA устанавливается, когда последняя транзакция DMA завершена.

Статус ошибки DMA устанавливается, если обнаружено запрещенное или ошибочное состояние.

В случае, если обнаруживается ошибочное состояние, DMA может генерировать аппаратное прерывание, если оно включено, и биты статуса могут быть очищены только чтением из регистра DSTATC. Регистры DSTAT и DSTATC должны читаться как сдвоенное или счетверенное слово. Чтение этих регистров как стандартное слово недопустимо.

13.2.3 Регистр управления DMA

Биты регистра управления выполняют единственную функцию – приостановка работы соответствующего им канала DMA. Подробное описание разрядов регистра управления приведено ниже (Таблица 96).

Таблица 96 – Биты регистра управления

Бит	Имя	Описание
0	DMA0_P	Пауза (если 1) для канала 0 DMA
1	DMA1_P	Пауза (если 1) для канала 1 DMA
2	DMA2_P	Пауза (если 1) для канала 2 DMA
3	DMA3_P	Пауза (если 1) для канала 3 DMA
4	DMA4_P	Пауза (если 1) для канала 4 DMA
5	DMA5_P	Пауза (если 1) для канала 5 DMA
6	DMA6_P	Пауза (если 1) для канала 6 DMA
7	DMA7_P	Пауза (если 1) для канала 7 DMA
9:8		-
10	DMA8_P	Пауза (если 1) для канала 8 DMA
11	DMA9_P	Пауза (если 1) для канала 9 DMA
12	DMA10_P	Пауза (если 1) для канала 10 DMA
13	DMA11_P	Пауза (если 1) для канала 11 DMA
15:14		
16	DMA12_P	Пауза (если 1) для канала 12 DMA
29:17		-
30	nDMAR_ALT	Если бит 30 регистра установить в 1, то запросы nDMAR[3:1] используют альтернативные входы согласно Таблице 1
31	PMOD	Управление режимом паузы

Регистр управления DMA имеет три адреса:

- DCNT – регулярный адрес для чтения и записи.
- DCNTST – адрес для установки бит. Доступен только по записи. Запись 1 устанавливает соответствующий бит, а запись 0 не изменяет его значения.
- DCNTCL – адрес для очистки бит. Доступен только по записи. Запись 0 сбрасывает значение бита, а запись 1 не изменяет его значение.

Назначение бита паузы состоит в том, что с его установкой в 1 соответствующий канал DMA приостанавливает свою работу и возобновляет ее только после того, как бит станет равен 0.

Назначение бита PMOD состоит в том, что, если установить бит паузы при установленном бите PMOD регистра управления, блокировки СОК-шины не будет.

Вместо этого в регистре состояния выбранного канала будет отражаться код 3 до момента завершения всех начатых транзакций. Это позволяет отследить момент завершения операций каналом и затем безопасно перепрограммировать его. Отсутствие блокировки СОК шины положительно влияет на оперативность действий процессора в случае прерывания.

Регистры управления DMA (DCNT, DCNTST, DCNTCL) могут быть доступны как нормальное, удлиненное и счетверенное слово. Начальное значение DCNT после сброса равно 0.

Управление битом паузы каналами 4-12 довольно простое, т.к. установка бита паузы для них означает остановку генерации транзакций каналом. При этом уже запущенные каналами транзакции спокойно завершаются.

Управление битом паузы для каналов 0-3 более сложное. Это связано с тем, что транзакции этих каналов состоят из двух самостоятельных частей: транзакции чтения данных из источника в буфер DMA и затем транзакции записи данных из буфера по адресу приемника. Канал может запускать до 8-ми транзакций чтения, прежде чем хотя бы одна транзакция записи выполнится. Бит паузы влияет только за запуск транзакций чтения. Но установка бита паузы не означает, что канал остановил работу. Это будет сделано, только когда все запущенные транзакции чтения загрузят в буфер данные, и затем выполнятся все соответствующие им транзакции записи. Изменение настроек канала после установки бита паузы приведет к потере прочитанных данных, т.к. операции записи могут быть сброшены. Чтобы этого не произошло, после установки бита паузы для каналов 0-3 оценивается наличие у канала DMA незавершенных транзакций записи, и если таковые имеются – канал генерирует сигнал блокировки доступа к SOC-шине со стороны ядра. Таким образом, если процессор сразу после записи бита паузы пытается произвести операцию на SOC-шине, он может быть остановлен на некоторое время. Данное время может быть значительным, если канал работает с внешней памятью. Это обстоятельство нужно учитывать при работе с каналами 0-3. По возможности не стоит использовать бит паузы для каналов 0-3, если источником данных для каналов является внешняя память и при этом необходима высокая скорость реакции на прерывание.

Установка бита PMOD регистра управления в 1 отразится на поведении контроллера DMA при установке бита паузы. Если установить бит паузы при установленном бите PMOD регистра управления, то блокировки SOC-шины не произойдет. Вместо этого в регистре состояния выбранного канала будет отражаться код 3 до момента завершения всех начатых транзакций. Это позволяет отследить момент завершения операций каналом и затем безопасно перепрограммировать его. Отсутствие блокировки SOC-шины положительно влияет на оперативность действий процессора в случае прерывания.

Пример

Рассмотрим пример использования каналов 4-11 для организации пересылок. Допустим, необходимо переслать массив из 8-ми чисел из одной области внутренней оперативной памяти в другую область внутренней оперативной памяти, используя при этом в качестве инициатора пересылки таймер 0 процессора 1986. Задача решается с использованием каналов 4 и 8, а также с использованием интерфейса SPI1.

Обозначения:

```
#define base_DST_ch74   0x80000030
#define base_SRC_ch118  0x80000050
#define base_DMACFG     0x80000078
```

```
Наш массив для пересылки
.var spm[8] = {0x11111111, 0x11111122,0x11111333,0x11114444,
               0x11155555, 0x11666666,0x17777777,0x88888888};
```

Предположим, что таймер 0 включен, работает и с определенным периодом формирует запросы к каналам DMA. Поскольку каналы DMA не включены и не настроены на прием запросов от таймера, то никаких событий не происходит. Включим SPI1 интерфейс для передачи 32-разрядных данных в тестовом режиме с включенной обратной связью.

```
                j8 = base_SPI1;;

        xr0 = 0x1001f;;
        [j8+0] = xr0;; // cr1
        xr0 = 0x0d;;   //
        [j8+1] = xr0;; // cr2
        xr0 = 0xff;;
        [j8+3] = xr0;; // st
```

Определим источники запросов для активации каналов 4 и 8. Канал 4 будем использовать для пересылки данных из памяти в передатчик SPI1 по запросу таймера 0 (запрос 0). Канал 8 будем использовать для пересылки данных от приемника SPI1 во внутреннюю память. Итого код активации канала 4 равен 20 (см. Таблицу 84), а код активации канала 8 равен 12 (см. Таблицу 85). Формируем 64-разрядный код и пишем его в регистр конфигурации:

```
             j8 = base_DMACFG;;
        xr0 = 0x00040000;;   // GTMR0 req
              xr1 = 0x0010000c;;
        l[j8+0] = xr1:0;;
```

Определяем устройство в которое канал 4 будет записывать данные для передачи. Адрес устройства есть буфер данных SPI1. Активируем регистр DCA записав в поле активации значение 001b.

```
              j8 = base_DST_ch74;;
        xr0 = 0x80000362;;   // dest  SPI1
              xr1 = 0x20000000;;// код активации
        l[j8+0] = xr1:0;;
```

Определяем устройство, из которого канал 8 будет читать данные. Отметим, что в данном случае можно было бы обойтись без новых функций, т.к. источник запроса и обслуживающее устройство – одно и то же устройство. Но будем использовать регистр DCA. Запишем адрес-источник данных – это буфер данных приемника SPI1. Код активации равен 100b, что означает обращение к устройству на SOC-шине.

```
        j8 = base_SRC_ch118;;
        xr0 = 0x80000362;; // src  SPI1
              xr1 = 0x80000000;;// код активации
        l[j8+0] = xr1:0;;
```

Включаем каналы «стандартным» образом. Включаем канал 4 для передачи. Адрес источника – это массив SPM. Передача идет 32-разрядными словами. Количество слов равно 8.

```
XR0 = spm;; // address
XR1 = 0x00080001;;
XR2 = 0x00000000;;
XR3 = 0x42080000;;

DC4    = XR3:0;;
```

Включаем канал 8 «стандартным» образом. Адрес приемника равен 0x40000. Количество слов равно 8.Обмен идет 32-разрядными словами.

```
XR0 = 0x00040000;; // internal address
  XR1 = 0x00800001;;
  XR2 = 0x00000000;;
  XR3 = 0x53080000;;

  DC8    = XR3:0;;
```

Ожидаем некоторый промежуток времени, пока пересылка закончится. Это можно сделать, анализируя регистр состояния DMA или по прерыванию от канала 8. Проверяем, что данные успешно переданы.

```
  lc1 = 8;;

  j0 = spm;;
  j1 = 0x40000;;


rpp_dd:
  j2 = [j0+=1];;
  j3 = [j1+=1];;
  comp(j2,j3);;
  if njeq, jump error (NP);;
  if nlc1e, jump rpp_dd;;
```

13.3 Операции DMA контроллера

13.3.1 DMA управление каналами 4-11

Внутренние периферийные устройства процессора способны использовать 12 каналов DMA для обработки приема и передачи данных. Для передачи из внутренней или внешней памяти в устройства используются каналы 0, 2, 4-7. При использовании каналов 4-7 выбор периферийного устройства для требуемого канала выполняется с помощью соответствующих каналу бит регистра конфигурации DMACFGx. Регистры источника TCB программируются адресом внутренней/внешней памяти DI, инкрементом адреса DXM, количеством слов для передачи DXC и управляющими разрядами DP. По окончании программирования TCB автоматически начинается передача и продолжается до завершения передачи блока. При этом транзакции всегда инициируются устройством в момент, когда оно готово принять новую порцию данных. Данный процесс повторяется, пока блок данных не передается полностью.

Для передачи из устройств во внутреннюю или внешнюю память, используются каналы 1, 3, 8-11. При использовании каналов 8-11 выбор периферийного устройства для требуемого канала выполняется с помощью соответствующих каналу бит регистра конфигурации DMACFGx. Регистры приемника TCB программируются адресом внутренней/внешней памяти DI, инкрементом адреса DXM, количеством слов для передачи DXC и управляющими разрядами DP. По окончании программирования TCB автоматически начинается передача. При этом транзакции канала инициируются периферийным устройством в момент, когда у него есть принятые данные. После генерации запроса, принятые данные считываются соответствующим каналом DMA и отсылаются по адресу получателя. Данный процесс повторяется, пока блок данных не передается полностью.

Количество слов для передачи или приема определяется как количество DXC или количество DXC умноженное на количество DYC (если режим 2D установлен).

13.3.2 DMA управление каналами 0-3

Особенностью каналов 0-3 является возможность выполнения передачи память-память. В связи с этим каналы 0-3 имеют два регистра ТСВ. Для передачи из внутренней во внешнюю память в регистры источника ТСВ записывается адрес внутренней памяти DI, приращение адреса DXM, количество слов для передачи DXC и управляющие разряды DP. Регистры ТСВ приемника заполняются адресом внешней памяти DI, приращением адреса DXM, количеством слов для передачи DXC и управляющими разрядами DP. При передаче в обратном направлении источник и приемник меняются местами.

После окончания настройки передача начинается автоматически и продолжается до полной передачи блока. Передачи выполняются порциями (по одному операнду на транзакцию). При этом канал может инициировать транзакции непрерывно.

Каналы 0-3 также могут выполнять пересылки устройство-память или память-устройство. В этом случае в ТСВ канала программируются адреса памяти и устройства. Основное отличие в такой передаче – это инициирование обмена по запросу устройства. Для этого в регистре конфигурации DMACFGx для заданного канала осуществляется программирование номера устройства. Выбранное устройство будет выполнять инициирование обменов. Для каналов 0-3 возможно задание внешнего источника запроса с помощью выводов nDMAR3–0, т.е. обмен может выполняться с устройством на внешней шине.

Устройство ввода-вывода, в отличие от памяти, сообщает каналу DMA о готовности к передаче данных. Устройство-источник готово, если у него есть данные для записи. Приемник готов, если у него есть место в буфере записи. Для передачи данных между внутренней и внешней памятью могут использоваться данные размером 32, 64 и 128 бит. При доступе к внешней памяти контроллер DMA использует те же интерфейсные сигналы, что и ядро процессора.

13.3.3 DMA управление каналом 12

Канал 12 использует режим работы, в котором контроллер DMA выступает в роли ведомого устройства, а цифровой смеситель (ЦС) – в роли ведущего. Для ЦС канал 12 виден как буфер приема информации с индикацией готовности. Когда ЦС имеет данные для передачи, он анализирует флаг готовности буфера приема канала 12 и записывает в буфер свои данные. Как только в буфере канала появляются данные, канал выполняет их пересылку во внутреннюю память в соответствии с запрограммированным в нем ТСВ. Для передачи данных между ЦС и внутренней памятью могут использоваться внутренние данные размером только 128 бит.

Канал 12 включен, если поле TY в регистре DP не равно нулю. Канал может поддерживать цепочки операций.

Если происходит запись в канал 12, когда он выключен и не занят загрузкой нового ТСВ цепочки, записанные данные теряются, происходит прерывание аппаратной ошибки и ошибка отражается в регистре SYSTAT.

Если запись в канал происходит, когда он выключен, но занят загрузкой ТСВ новой цепочки, данные помещаются в буфер и передаются по назначению, когда загрузка ТСВ для канала завершается. В этом случае нет генерации ошибки. В ТСВ канала программируется только базовый адрес буфера приемника. Смещение относительно базы передает сам ЦС вместе с соответствующей ему порцией данных. Канал 12 формирует итоговый адрес, суммируя значения базы и смещения.

Канал 12 работает с ЦС, только если установлен бит DRQ в регистре управления TCB. Если данный бит сброшен, канал 12 может выполнять обмен в случае, когда в его регистр AutoDMA производит запись ведущее устройство.

13.4 DMA передача

13.4.1 Адресация памяти

Контроллер DMA поддерживает словную адресацию памяти. Это означает, что минимальной адресуемой единицей является 32-разрядное слово, и адрес в 32-разрядном индексном регистре есть адрес слова.

Благодаря 32-разрядному регистру адреса DMA может получать доступ к полному адресному пространству процессора. Каждый канал включает регистр индекса DI и поле модификации DXM в своем регистре TCB, которые используются для адресации буфера данных в памяти. 32-битный регистр TCB DI содержит начальный адрес блока и должен инициализироваться стартовым адресом буфера данных. Поле модификации DXM содержит значение со знаком (т.е. положительное или отрицательное) размером 16 или 22 бита и определяет величину изменения адреса после выполнения каждой транзакции. Измененное значение адреса используется в следующей транзакции. Само значение адреса в регистре DI не указывает на адресацию внутренней или внешней памяти. Чтобы указать, какому типу памяти соответствует адрес, используется дополнительное поле TY в регистре управления DP блока TCB. Если в результате модификации, адрес выйдет за границы существующей памяти или сменит область внутренней памяти на внешнюю (или наоборот), контроллер DMA все равно будет обращаться к области памяти в соответствии с полем TY.

Передаваемые данные могут быть стандартными, сдвоенными или счетверенными словами. Это указывается в поле LEN регистра управления DP.

Каждый канал DMA имеет поле количества DXC в регистре DX, которое должно быть инициализировано количеством слов для передачи, независимо от длины данных (32, 64 или 128 бит). Счетчик количества уменьшается после каждой передачи DMA в данном канале. Значения уменьшения равны 1, 2 или 4 и определяются полем LEN. Когда количество достигает значения ноль, DMA завершается и может быть сгенерировано прерывания для информирования процессора.

Если поле количества DXC в TCB DX инициализируется нулем, DMA передача на этом канале возможна и будет соответствовать выполнению передачи максимально возможного количества данных. Это происходит из-за того, что первая передача начинается до того, как проверяется значение количества.

Значение количества анализируется после вычитания из него 1. Надежным способом отключения DMA-канала является сброс поля TY в соответствующем управляющем регистре.

13.4.2 Приоритеты каналов DMA

Система приоритетов используется при выборе канала DMA для обслуживания, т.к. в одном такте может поступить активный запрос более чем от одного канала DMA. В зависимости от бита PR в регистре DP блока TCB канал может относиться к группе каналов с высоким приоритетом или к группе с низким приоритетом. Внутри каждой группы каналов используется фиксированная система приоритетов между каналами. Описание каналов DMA в порядке уменьшения приоритетов приведено ниже (Таблица 97).

Приоритет устанавливается исходя из следующих соображений:

- Входящие данные имеют более высокий приоритет, чем исходящие. Приемники устройств имеют более высокий приоритет, чем передатчики.
- Каналы устройств имеют более высокий приоритет, чем каналы общего назначения.
- Каналы общего назначения дополнительно имеют циклически изменяемый приоритет в последовательности 3-2-1-0.

Контроллер DMA анализирует запросы от всех каналов внутри двух групп и в течение одного такта определяет канал с самым высоким приоритетом. Выбранный канал внутри высокоприоритетной группы имеет преимущество над каналом обычной группы. Выбранному каналу в текущем такте предоставляется право выполнить транзакцию. В следующем такте арбитраж повторяется.

Загрузка цепочки ТСВ получает приоритет канала. Например, передача данных канала с более высоким приоритетом производится перед загрузкой цепочки ТСВ с более низким приоритетом.

Таблица 97 – Приоритеты каналов DMA

Канал	Описание	Приоритет
Канал 12	Интерфейс ЦС	Наиболее высокий
Канал 11	Устройства-приемники	
Канал 10		
Канал 9		
Канал 8		
Канал 7	Устройства-передатчики	
Канал 6		
Канал 5		
Канал 4		
Канал 3	Каналы общего назначения	
Канал 2		
Канал 1		
Канал 0		Наиболее низкий

13.4.3 Циклически присваиваемый приоритет

Для каналов 0-3 сделана дополнительная модификация – циклический приоритет внутри их группы. Это означает, что после каждого цикла арбитража приоритет каналов 0-3 меняется циклически. Приоритеты каналов после сброса следующие:

- канал 3 (самый высокий);
- канал 2;
- канал 1;
- канал 0 (самый низкий).

Каждый раз, если одному из каналов 0-3 предоставляется возможность выполнить транзакцию, приоритет каналов меняется. Только что работавший канал получает высочайший приоритет, остальные приоритеты расставляются как описано выше, но по циклу. Например, если канал 1 в текущем такте выполняет транзакцию, в следующем такте приоритет каналов следующий:

- канал 1 (самый высокий);

- канал 0;
- канал 3;
- канал 2 (самый низкий).

Если следующую транзакцию выполняет канал 0, приоритеты следующие:

- канал 0 (самый высокий);
- канал 3;
- канал 2;
- канал 1 (самый низкий).

Приоритеты каналов 0-3 остаются неизменными при выборе каналов с более высоким приоритетом (4-12). По окончании транзакции канала с более высоким приоритетом, порядок приоритетов в каналах 0-3 сохраняется прежним.

В примерах выше подразумевалось, что все четыре канала не устанавливают бит приоритета в TCB или все устанавливают, т.е. принадлежат одной группе. Если один или более канал установлен на высокий приоритет, циклический алгоритм применяется для каждой группы в отдельности. Например, если после сброса у канала 2 установлен бит приоритета, приоритеты следующие:

- канал 2 (установлен бит);
- канал 3 (самый высокий);
- канал 1;
- канал 0 (самый низкий).

Если канал 1 запрашивает доступ и получает его, схема приоритетов следующая:

- канал 2 (установлен бит);
- канал 1 (самый высокий);
- канал 0;
- канал 3 (самый низкий).

Если в течение работы канала 1, канал 2 генерирует запрос, он получает доступ и работа канала 1 прерывается. После окончания передачи каналом 2 схема приоритетов для каналов с более низкими приоритетами не меняется, и канал 1 выигрывает арбитраж и продолжает передачу. Если более одного канала имеют установленный бит приоритета, арбитраж между этими каналами проходит циклически.

13.4.4 Приоритет контроллер DMA на внутренних шинах процессора

Канал DMA, выигравший арбитраж приоритетов среди всех каналов внутри контроллера, получает возможность выполнить транзакцию и генерирует запрос к одному из трех ресурсов системы (Рисунок 19):

- SOC IFIFO – входной буфер SOC-интерфейса для доступа к внутренней памяти.
- EBIU OFIFO – выходной буфер внешнего интерфейса для доступа к внешней памяти.
- SOC-шина периферийных устройств.

При арбитраже запросов к SOC IFIFO контроллер DMA имеет самый низкий приоритет. Первыми получают доступ данные, считываемые процессором с SOC-шины, или запрос от EBIU IFIFO внешнего интерфейса.

При арбитраже запросов к EBIU OFIFO контроллер DMA также имеет самый низкий приоритет. Первым получает доступ ядро процессора. При этом не имеет значения высокоприоритетный запрос DMA или нет.

При арбитраже запросов к SOC-шине периферийных устройств DMA имеет более высокий приоритет, чем ядро процессора, но ниже чем возвращаемые из ядра или внешней памяти данные. Данные, возвращаемые из ядра или внешней памяти, являются результатом пересылки DMA типа память-устройство. Канал DMA формирует запрос к SOC-шине только при выполнении пересылки устройство-память.

13.4.5 Цепочка DMA

Традиционный алгоритм работы DMA заключается в программировании процессором канала DMA, выполнение каналом передачи данных и генерации прерывания после завершения работы. В ответ на запрос прерывания процессор может запрограммировать канал на новую передачу.

Использование цепочек в DMA позволяет контроллеру автоматически инициализировать себя следующей передачей блока данных после завершения передачи текущего блока. При этом участие процессора не требуется, и он экономит время, которое потратил бы на обработку прерывания и программирование нового обмена. Используя механизм цепочки, могут быть организованы множественные операции DMA с различными атрибутами и устройствами ввода-вывода.

В первую очередь для поддержки цепочки используются биты регистра TCB DP:

- Бит CHEN разрешения цепочки.
- Поле CHPT указателя на цепочку. Это адрес в памяти, который содержит следующий TCB для загрузки (где содержится адрес, инкремент, количество слов для передачи и управляющие разряды для следующего передаваемого блока).
- Поле CHTG приемника нового TCB. Прочитанное из памяти значение TCB загружается в канал DMA указанный в данном поле.

Таким образом, в каждом TCB канала имеется механизм для загрузки в регистр TCB нового значения командного слова после окончания работы. Из-за ограничения размера указателя хранение значений цепочек возможно только во внутренней памяти процессора.

Существует несколько ограничений на цепочки DMA между каналами:

- Для каналов 0-3 возможна поддержка только одноканальной цепочки, т.е. поле CHTG не имеет значения, и новый TCB цепочки может быть загружен только в вызвавший ее канал. При этом в обоих TCB канала бит CHEN должен быть установлен, т.к. для работы этих каналов нужна загрузка двух TCB.
- Для каналов DMA 4-11 возможна межканальная цепочка, когда один канал может запускать работу в другом канале. Например, канал 8, приняв блок данных и загрузив его в память, может инициировать канал 4 на передачу принятых данных.

Включение и выключение цепочек

Передачи DMA запускаются записью счетверенного слова в регистр TCB (каналы 0-3 требуют загрузки обоих регистров TCB). Цепочка образуется при

установленном бите разрешения CHEN, и, если указатель цепочки CHPT является разрешенным адресом.

Поле CHPT в TCB DP может быть загружено:

- при инициализации канала;
- во время работы канала (вставка цепочки).

В первом варианте пользователь заранее знает, что необходимо выполнить несколько разнообразных передач, и программирует цепочку операций, загружая первый TCB в канал, а все последующие TCB сохраняя во внутренней памяти.

Во втором случае пользователь запускает передачу одного блока, но в процессе выполнения программы (возможно, другим процессом) возникает необходимость в передаче нового блока данных. В этом случае программа определяет, что требуемый канал активен и устанавливает для данного канала бит паузы. Это вызывает приостановку работы канала и позволяет программе:

- Если цепочка операций в активном TCB не включена – организовать вставку цепочки путем установки бита CHEN в 1 и загрузки поля указателя CHPT необходимым значением. Значение TCB новой передачи сохраняется во внутренней памяти.
- Если цепочка операций включена – добавить в конец (или в любое место) списка цепочек свою запись.

После выполнения описанных действий процессор может сбросить бит паузы и канал продолжит работу. После сброса бита паузы приостановленный активный канал выполняет повторный контроль своего состояния и, в случае корректной работы, продолжает работу.

Генерация прерывания в цепочке операций

Пользователь может сам выбрать режим прерывания при работе цепочки. Если необходима генерация прерывания при передаче конкретного блока, бит INT в регистре TCB блока нужно установить в 1. Прерывание произойдет после передачи данного блока. Если необходимо сгенерировать прерывание только после окончания работы всей цепочки, бит INT должен быть установлен только в последнем TCB цепочки.

TCB и загрузка цепочек

Во время загрузки TCB цепочки, регистры TCB канала DMA загружаются значениями, полученными из внутренней памяти. TCB хранится в четырех последовательных словах выровненного счетверенного слова. Чтение нового TCB цепочки инициируется после передачи всего блока данных. Для каналов 0-3 необходимо чтение 2-х TCB. При необходимости чтения цепочки канал формирует запрос на чтение TCB со своим приоритетом, т.е. чтение цепочки каналом имеет такой же приоритет, как и чтение данных.

13.4.6 Двухмерный DMA

Данный раздел описывает изменения в работе, которые происходят при включении режима двухмерного DMA в процессоре. Режим двухмерного DMA включается установкой бита 2D в TCB DP регистре. Этот режим использует внешние и внутренние адреса. Преимущество двухмерной адресации состоит в перераспределении данных из одномерной строки (вектора) в представление в виде двухмерного массива (матрицы) с координатами X и Y. Измерения X и Y

определяются в регистрах TCB передатчика и/или приемника. При этом измерение X определяет количество слов в строке массива, а измерение Y – количество строк.

- Измерения массива передатчика и приемника могут отличаться. Главное требование – итоговое число слов в источнике и получателе должно быть равно.
- Двухмерный массив памяти может быть перемещен в одномерный массив и наоборот, если итоговое число слов в источнике и получателе равно.
- Двухмерный блок в памяти может быть передан в порты связи, а блок, полученный из порта связи или из AutoDMA, может быть размещен в памяти как двухмерный массив.

13.4.7 Организация канала двухмерного DMA

В режиме двухмерного DMA адресация к массиву может быть выполнена на любом DMA-канале, приемнике или передатчике. Регистр индекса (DI) загружается адресом первого элемента массива и обновляется после каждой передачи операнда на значение модификатора DXM, пока поле количества DXC не достигнет нуля. Величина модификатора DXM в регистре DX содержит смещение, добавляемое к текущему значению адреса для перехода на следующий элемент в измерении X (следующая колонка).

Для режима 2D поле количества DXC в регистре DX имеет дополнительную буферизацию (внутренний буфер CIX). При загрузке TCB, поле количества DXC и буфер CIX загружаются одинаковым значением. В процессе передачи по измерению X количество в DXC уменьшается до нуля и буфер CIX используется для перезагрузки количества в DXC. Можно сказать, что измерение X работает в режиме 2D так же, как в обычном режиме, только после передачи всей строки X происходит перезагрузка значения количества.

Регистр DY в TCB используется для задания количества строк массива и для перехода от одной строки к другой. В процессе передачи текущей строки к адресу элемента прибавляется значение DXM и осуществляется переход к следующему элементу строки (элементы не обязательно последовательны). При передаче последнего элемента строки (поле DXC достигает значения ноль) к адресу последнего элемента прибавляется модификатор DYM (модификатор DXM не добавляется). Это позволяет перейти к первому элементу следующей строки. Значение модификации DYM в регистре DY является целочисленным числом со знаком, что позволяет уменьшать или увеличивать адрес.

Значение количества строк в DYC уменьшается на 1 и, если это была не последняя строка, количество DXC перезагружается из CIX.

Когда поле количества DYC достигает нуля, DMA-передача блока окончена, образуя матрицу размером X на Y.

Пример задания двухмерной передачи

Матрица с количеством колонок COL и количеством строк ROW. Каждый элемент матрицы имеет размер TXS (он определяется полем LEN в DP и может быть равен 1, 2 или 4). Элементы матрицы размещаются в памяти последовательно, причем сначала элементы первого столбца, затем второго и т.д. Например, необходимо выполнить передачу массива построчно. Следующая формула может быть применена для расчета параметров DX и DY.

```
DX = TXS*COL << 16; // количество слов данных в строке
DX |= TXS*ROW; // шаг от текущего элемента строки к следующему
```

DY = ROW << 16; // количество строк
DY |= (-TXS*(((COL-1)*ROW)-1))&0xFFFF; // переход к следующей строке

Операция «И» с 0xFFFF использовалась для того чтобы ограничить длину отрицательного модификатора размеров в 16 бит перед операцией логического ИЛИ с количеством строк.

Двухмерные DMA операции

Рассмотрим подробно действия, которые происходят внутри канала при двухмерной передаче. Все действия происходят в течение одного такта, но имеют причинно-следственную связь:

- Берется текущий адрес в регистре DI и запускается транзакция передачи операнда длиной LEN (чтение или запись).
- При подтверждении отправки транзакции происходит вычитание из DXC числа слов в операнде. Результат вычитания анализируется.
- Если результат вычитания не равен нулю, к содержимому регистра DI прибавляется значение модификации DXM. Результат вычитания помещается в поле количества DXC, а новое значение адреса в DI. На этом действия канала в текущей транзакции закончены.
- Если результат вычитания равен нулю, происходит вычитание единицы из поля количества DYC. Результат вычитания анализируется.
- Если результат вычитания количества DYC равен нулю – канал завершает работу.
- Если результат вычитания не равен нулю, в регистр количества DXC загружается начальное значение из буфера CIX, к значению регистра DI прибавляется значение модификатора DYM, результат вычитания загружается в поле количества DYC. Процесс повторяется.

Особенность двухмерного DMA (и любого DMA) в том, что первая передача начинается до анализа количества. Это означает, что DMA не может быть выключен установкой в ноль количества в регистре DXC или DYC. Для отключения двухмерного DMA необходимо очистить бит 2D в регистре управления DP. Для отключения канала DMA, необходимо очистить поле TY в регистре DP.

13.4.8 Прерывания DMA

Прерывание окончания DMA генерируется, если бит INT в регистре канала DP установлен.

DMA генерирует прерывание, когда количество передач в активном канале DMA доходит до нуля и передача заканчивается. Когда назначением DMA является внутренняя память, окончание передачи означает, что последний элемент данных попал в буфер SOC IFIFO.

Когда назначением DMA является внешняя память, окончание передачи означает, что последний элемент данных попал в буфер EBIU OFIFO.

Если назначением передачи является устройство-передатчик, окончание передачи означает, что последний элемент данных был загружен в буфер передатчика.

Видно, что генерация прерывания для случая, когда пункт назначения данных в памяти, не означает, что последние данные уже находятся в памяти. Это произойдет с задержкой. Для случая внутренней памяти задержка может достигать:

- 32-х тактов процессора (если запись идет в блок памяти, куда одновременно обращаются с чтением все три внутренних процессорных шины);
- 16-ти тактов (если запись идет в блок памяти без конфликтов).

Минимальная задержка 3 такта. Процессор не в состоянии за это время обработать запрос прерывания и попытаться прочитать последнее слово переданных данных.

Задержка для внешней памяти зависит от частоты и настроек системной шины.

Каждый канал DMA имеет собственный бит запроса прерывания в контроллере прерываний. Прерывания DMA фиксируются в регистре ILAT и разрешаются в регистре IMASK. Приоритет запросов прерываний от каналов DMA фиксированный и определен в контроллере прерываний.

Опрос регистра DSTAT может использоваться как альтернатива прерываниям для определения окончания одиночной последовательности DMA. Если включены цепочки, опрос DSTAT нельзя использовать, т.к. следующая последовательность DMA может быть на подходе в момент возврата статуса по результатам опроса.

13.4.9 Запуск и окончание последовательностей DMA

Данный подраздел рассматривает запуск, приостановку, возобновление и окончание последовательностей DMA.

13.4.9.1 Запуск последовательности DMA

Последовательность DMA для каналов 0-3 запускается в одном из случаев:
- включен канал DMA, бит запроса DRQ равен нулю;
- включен канал x DMA, установлен бит запроса DRQ и получен запрос от выбранного устройства.

Последовательность DMA для каналов 4-7 запускается в случае, когда:
- включен канал DMA и имеется запрос от устройства-передатчика на прием данных в буфер передатчика. Бит запроса DRQ не имеет значения.

Последовательность DMA для каналов 8-11 запускается в случае, когда:
- включен канал DMA и имеется запрос от устройства-приемника о наличии данных в буфере приемника. Бит запроса DRQ не имеет значения.

Последовательность DMA для канала 12 запускается в случае, когда:
- Включен канал DMA и имеются данные в буфере канала. Буфер канала может принять до 4-х операндов. Бит запроса DRQ не имеет значения.

Для запуска новой последовательности DMA после завершения текущей, программа должна записать новые параметры в регистры TCB. Запись активного TCB в активный TCB регистр вызывает ошибку.

13.4.9.2 Приостановка последовательности DMA

Последовательность операций DMA в некотором канале приостанавливается, когда соответствующий ему бит паузы в регистре DCNT установлен. Установка бита паузы может произойти в любой момент работы канала: канал может работать, загружать цепочку или завершить работу. Поэтому после установки бита паузы

пользователь должен перед продолжением работы проанализировать состояние канала.

Если момент установки паузы пришелся на загрузку цепочки, бит паузы не остановит загрузку цепочки, и до момента загрузки цепочки поле TY регистра TCB будет читаться как 0. Хотя биты состояния в этот момент будут показывать, что канал активен.

Если момент установки бита паузы пришелся на завершение работы каналом, поле TY регистра TCB будет равно нулю, и регистр состояния укажет, что канал не активен.

Установка бита паузы в момент нормальной работы канала (пересылка данных) вызывает останов операций передачи. При этом каналы с 4 по 12 немедленно прекращают операции пересылки. Каналы с 0 по 3 немедленно прекращают операции чтения данных из источника, но продолжают выполнять незавершенные операции записи.

Внимание! При наличии незавершенных операций записи после установки бита паузы SOC-шина становится недоступной для ядра процессора.

Разблокировка доступа произойдет только после завершения операций записи.

Установка бита паузы позволяет записать новое активное значение TCB (поле TY не равно нулю) в активный регистр TCB. Такая запись приводит к генерации аппаратной ошибки. Однако, в случае паузы такая запись разрешена. Если случай записи без бита паузы рассматривается как сбой в программе, запись с установленным битом паузы является осознанной последовательностью действий по смене режима работы канала. Это позволяет отложить задачу, выполняемую каналом, и запрограммировать канала на более приоритетную задачу. После ее завершения состояние канала может быть восстановлено.

Имеется особенность в перепрограммировании каналов с 0 по 3. Если новое активное TCB имеет поле TY такое же, как активный регистр TCB канала, данная запись рассматривается только как попытка модификации регистра DP (например, попытка вставить цепочку). В этом случае при записи обновляется только значение регистра DP. Все другие значения неизменны. Поэтому для каналов с 0 по 3, для задания нового обмена необходимо:

- сохранить текущие параметры регистра TCB;
- записать в регистр TCB значение с полем TY равным нулю;
- записать активное значение в TCB регистр.

13.4.9.3 Возобновление последовательности DMA

Последовательность операций DMA в некотором канале возобновляется, когда соответствующий ему бит паузы в регистре DCNT сбрасывается в ноль. Для каналов с 4 по 12 это означает немедленное возобновление работы в соответствии с командой TCB. Для каналов с 0 по 3 сброс бита паузы может привести к процедуре проверки правильности конфигурации TCB, если во время паузы были записи в TCB регистры.

13.4.9.4 Окончание последовательности DMA

Последовательность DMA заканчивается в одном из случаев:

- Регистры количества равны нулю и цепочка заканчивается.
- Канал отключен сбросом поля TY в одном из TCB и DP регистрах. При этом такая запись неактивного TCB может происходить в любой момент. Для каналов с 0 по 3 могут быть запущены до 8-ми транзакций чтения во внутреннюю или внешнюю память, и после отключения канала все эти

данные поступят во внутренний буфер контроллера DMA и будут утеряны, если канал выключен. Если к этому времени канал будет включен для новой задачи, оставшиеся от предыдущей задачи данные вызовут ошибку в работе канала, т.к. обнаружится запись данных, которые не читались.

13.5 Каналы DMA общего назначения

Четыре канала DMA с 0 по 3 являются каналами общего назначения, т.к. обладают двумя регистрами TCB и позволяют программировать произвольный источник и приемник данных. Регистры TCB каналов определяют следующее:

- канал включен, если оба поля TY в TCB DP не равны нулю;
- приоритет канала высокий, если бит PR установлен в одним из регистров TCB DP;
- прерывание DMA включено, если бит INT установлен в одном из регистров TCB DP;
- режим запроса DMA включен, если бит DRQ установлен в одном из регистров TCB DP;
- поле LEN должно быть одинаковым в обоих регистрах;
- бит CHEN должен быть одинаковым в обоих регистрах;
- передача внутренней и внешней памяти устанавливается в поле TY регистров TCB DP, что позволяет эффективно передавать данные между внутренней памятью процессора и внешней памятью или устройством;
- поле CHTG не имеет смысла для этих каналов. В случае цепочки каждый регистр TCB загружается новое значение в свой канал в соответствии с его полем CHPT.

Передача DMA между внутренней памятью процессора и внешней памятью требует от одного канала DMA генерации адресов для обоих типов памяти. Внешний адрес генерируется в соответствии с данными в регистре TCB внешней памяти. Адрес внутренней памяти генерируется в соответствии с данными в регистре TCB внутренней памяти. Если бит 2D в TCB DP установлен в одном из регистров TCB, адрес этого TCB также обновляется в соответствии со значением модификатора в регистре DY. Передачи внутри внешней памяти и внутренней памяти поддерживаются. Также поддерживается передача между внешней памятью и внешними устройствами.

Поле TY в регистре DP указывает тип передачи. Ранее приводилось описание допустимых комбинаций типов в одном канале (Таблица 85). Каждый канал с 0 по 3 имеет регистр TCB передатчика и приемника, где передатчик считывает данные из памяти и передает их приемнику, а приемник отсылает данные в память.

Большинство параметров TCB не имеют отношения к типу памяти, а являются описателями блока памяти (который может размещаться в любом месте) или описывают поведение канала во время работы и при завершении обмена.

Описатели блока памяти:

- регистры DX и DY задают размеры блока и расположение элементов блока в памяти;
- бит 2D определяет, рассматривается блок данных как линейный или как двухмерный;
- биты LEN определяют длину операнда блока.

Описатели поведения канала во время работы и при завершении обмена:

- бит PR указывает, что канал имеет высокий приоритет;

- бит DRQ определяет активацию транзакции по запросу или без него;
- бит INT разрешает формирование запроса прерывания к процессору по окончании обмена;
- бит CHEN разрешает цепочку операций;
- поле CHPT хранит адрес цепочки;
- поле CHTG определяет приемник цепочки.

Рассмотренные выше параметры не зависят от типа обмена. Поэтому далее при описании различных вариантов обмена будут рассматриваться параметры, которые зависят от выбранного типа памяти.

13.5.1 Передача из внешней памяти во внутреннюю память

Для передачи данных из внешней памяти во внутреннюю необходимо запрограммировать TCB передатчика для чтения внешней памяти, а TCB приемника – для записи во внутреннюю память. TCB передатчика программируется следующим образом:

- Регистр DI

Инициализируется начальным адресом блока данных во внешней памяти.

- Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память. Выбор между типом 4 и 7 влияет только на длину полей DXC, DXM, DYC, DYM.

TCB приемника данных описывается аналогичным образом со следующими особенностями, связанными с выбором внутренней памяти:

- Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память. Выбор между типом 2 и 3 влияет только на длину полей DXC, DXM, DYC, DYM приемника.

- Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти.

Все другие параметры TCB для передатчика и приемника описывают блоки памяти и поведение канала. В описанных выше параметрах имеются ограничения, основные из которых:

- поле LEN должно быть одинаковым в обоих TCB;
- бит CHEN должен быть одинаковым в обоих TCB.

Выбор между типами обменов 2 и 3 или 4 и 7 зависит от размера величины модификации адреса. Если при переходе от одного элемента к другому необходимо изменить адрес на величину более чем 32768 (в положительную сторону или отрицательную), недостаточно 16 бит поля DXM для хранения такого значения. Поэтому следует использовать тип обмена 3 или 7 место 2 или 4, чтобы поле модификации было 22 бита. В этом случае уменьшается размер поля количества.

13.5.2 Передача из внутренней памяти во внешнюю память

В случае передачи из внутренней памяти во внешнюю память нет никаких особенностей. По сравнению с режимом, описанным в подразделе «Передача из

внешней памяти во внутреннюю память», TCB передатчика и приемника просто меняются местами.

TCB передатчика:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память. Выбор между типом 2 и 3 влияет только на длину полей DXC, DXM, DYC, DYM приемника.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти.

TCB приемника:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память. Выбор между типом 4 и 7 влияет только на длину полей DXC, DXM, DYC, DYM приемника.

– Регистр DI

Инициализируется начальным адресом блока данных во внешней памяти.

13.5.3 Передача из внешней памяти во внешнюю память

Подобный тип передачи выполняется в случае, когда в TCB приемника и передатчика поля типа TY равны 4 или 7. Адреса в регистрах DI должны указывать на блоки данных во внешней памяти.

13.5.4 Передача из внутренней памяти во внутреннюю память

Данный тип передачи программируется следующим образом.

TCB передатчика:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память. Выбор между типом 2 и 3 влияет только на длину полей DXC, DXM, DYC, DYM приемника.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти.

TCB приемника:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память. Выбор между типом 2 и 3 влияет только на длину полей DXC, DXM, DYC, DYM приемника.

– Регистр DI

Инициализируется начальным адресом блока данных внутренней памяти.

Необходимо отметить, что пересылки во внутренней памяти с помощью DMA выполняются как минимум вдвое медленнее, чем такие же пересылки, выполненные ядром процессора. Однако полезным может быть тот факт, что пересылка

выполняется в фоновом режиме и процессор может выполнять в это время другую работу.

13.5.5 Передача из устройства ввода во внутреннюю или внешнюю память

Данный тип передачи программируется следующим образом.

ТСВ передатчика:

- Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память. Выбор между типом 4 и 7 влияет только на длину полей DXC, DXM, DYC, DYM приемника.
Бит DRQ должен быть установлен, что указывает на инициирование обмена только по запросу от устройства.

- Регистр DI

Инициализируется адресом буфера периферийного устройства. При этом если бит 31-й адреса равен 1, это адрес внутреннего периферийного устройства на SOC-шине. Если бит равен нулю, это внешнее устройство.

ТСВ приемника:

- Регистр DP

Поле TY должно быть равно 2 (3) или 4 (7), что указывает на внутреннюю или внешнюю память. Выбор влияет только на длину полей DXC, DXM, DYC, DYM приемника.
Бит DRQ должен быть установлен, что указывает на инициирование обмена только по запросу от устройства.

- Регистр DI

Инициализируется начальным адресом блока данных внутренней или внешней памяти.

Регистр конфигурации DMACFGx задает номер периферийного устройства, которое будет формировать сигналы запроса на обмен.

13.5.6 Передача из внутренней или внешней памяти в устройство вывода

Данный тип передачи программируется следующим образом.

ТСВ передатчика:

- Регистр DP

Поле TY должно быть равно 2 (3) или 4 (7), что указывает на внутреннюю или внешнюю память. Выбор типа влияет только на длину полей DXC, DXM, DYC, DYM приемника.
Бит DRQ должен быть установлен, что указывает на инициирование обмена только по запросу от устройства.

- Регистр DI

Инициализируется начальным адресом блока данных во внутренней или внешней памяти.

ТСВ приемника:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память. Выбор между типом 4 и 7 влияет только на длину полей DXC, DXM, DYC, DYM приемника.

Бит DRQ должен быть установлен, что указывает на инициирование обмена только по запросу от устройства.

– Регистр DI

Инициализируется адресом буфера периферийного устройства. При этом если бит 31-й адреса равен 1, это адрес внутреннего периферийного устройства на SOC-шине. Если бит равен нулю, это внешнее устройство.

Регистр конфигурации DMACFGx задает номер периферийного устройства, которое будет формировать сигналы запроса на обмен.

Канал DMA не контролирует корректность соответствия номера устройства в регистре DMACFGx адресу буфера устройства в регистре ТСВ. Периферийное устройство выступает только в роли инициатора транзакции. Однако пересылка данных определяется только ТСВ источника и приемника. Поэтому по инициативе периферийного устройства с помощью каналов 0-3 можно выполнять любой вариант пересылки, который можно запрограммировать в канале. Отличие каналов 0 и 2 от каналов 1 и 3 только в том, что они имеют различные источники запросов обмена. При этом нужно помнить, что, если какое-то устройство сформировало запрос на обмен, для формирования следующего запроса устройство должно быть кем-то обслужено, т.е. из источника должны быть прочитаны данные, а в приемник записаны.

13.6 Каналы DMA с 4 по 11 для работы с устройствами

Четыре канала DMA определяют передачи от устройств к внутренней памяти или внешней памяти. Другие четыре канала определяют передачи из внутренней памяти или внешней памяти в устройства.

Особенностью каналов DMA является то, что все они имеют только один регистр ТСВ. Для каналов с 4 по 7 это ТСВ передатчика, который читает данные из внешней или внутренней памяти и отправляет данные в устройство-передатчик. Для каналов с 8 по 11 это ТСВ приемника, который берет данные из устройства-приемника и пересылает их во внутреннюю или внешнюю память. Наличие только одного ТСВ регистра в каналах 4-11 не позволяет определить вторую сторону пересылки (источник или приемник). Для разрешения данной проблемы и используется дополнительный регистр DMACFGx.

Как в случае каналов общего назначения, большинство параметров ТСВ не имеют отношения к устройству и описывают только блок данных или поведение канала во время работы или при завершении передачи. Поле TY регистра ТСВ указывает на тип передачи.

13.6.1 Передача из устройства во внутреннюю \ внешнюю память

Для переноса данных от устройства во внутреннюю или внешнюю память должны использоваться только каналы с 8 по 11. Приемник ТСВ этих каналов должен быть запрограммирован соответствующим образом.

Для передачи во внутреннюю память:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти куда будут пересылаться данные из порта связи.

Для передачи во внешнюю память:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память.

– Регистр DI

Инициализируется начальным адресом блока данных во внешней памяти, куда будут пересылаться данные из порта связи.

Для выбранного канала в регистре DMACFGH должен быть задан номер устройства, с которым работает канал. Биты регистра DMACFGH имеют следующее соответствие: биты 0-3 каналу 8, биты 4-7 каналу 9, биты 8-11 каналу 10, биты 12-15 каналу 11. Передача всегда инициируется устройством. Как только буфер устройства-приемника загружается принятыми данными, посылается запрос к каналу DMA, и он выполняет пересылку данных в блок данных описываемый регистром TCB.

Каналы DMA 4-11 имеют одну особенность при организации цепочки операций: если в универсальном канале новое значение TCB может быть загружено только в регистр того же канала, для каналов 4-11 можно указать номер канала-приемника, следующего TCB. Правда номер канала-приемника лежит в диапазоне от 4 до 11. Поэтому после окончания работы один канал может инициировать работу другого канала.

13.6.2 Внутренняя/внешняя память – устройство-передатчик

Для пересылки данных из внутренней или внешней памяти в устройство должны использоваться только каналы с 4 по 7. Передатчик TCB этих каналов должен быть запрограммирован соответствующим образом.

Для передачи из внутренней памяти:

– Регистр DP

Поле TY должно быть равно 2 или 3, что указывает на внутреннюю память.

– Регистр DI

Инициализируется начальным адресом блока данных во внутренней памяти, откуда будут читаться данные для порта связи.

Для передачи из внешней памяти:

– Регистр DP

Поле TY должно быть равно 4 или 7, что указывает на внешнюю память.

– Регистр DI

Инициализируется начальным адресом блока данных во внешней памяти, откуда будут читаться данные для порта связи

Для выбранного канала в регистре DMACFGL должен быть задан номер устройства, с которым работает канал. Биты регистра DMACFGL имеют следующее соответствие: биты 16-19 каналу 4, биты 20-23 каналу 5, биты 24-27 каналу 6, биты 28-31 каналу 7.

Передача всегда инициируется устройством-передатчиком. Как только буфер передатчика имеет возможность принять данные - посылается запрос к каналу DMA, и он выполняет пересылку данных из блока данных, описываемого регистром TCB, в буфер передатчика.

13.6.3 Пересылка между портами связи

Контроллер DMA поддерживает возможность пересылки данных между портами связи без использования промежуточных буферов в памяти. Понятно, что передача может идти от приемника порта связи в передатчик порта связи.

В процессоре имеется два порта связи. Поскольку инициатором передачи в данном случае выступает приемник порта связи, такие передачи поддерживаются только каналами DMA с 8 по 9. Отличием такой пересылки от пересылки в память является необходимость дополнительного контроля готовности передатчика канала связи принять данные от приемника. Особенности программирования TCB приемника каналов с 8 по 9 состоят в следующем:

- Регистр DP

Поле TY должно быть равно 1, что указывает на порт связи.

- Регистр DI

Инициализируется адресом передатчика порта связи, который будет принимать данные. Адрес 0x1F04A0 для порта связи 0, 0x1F04A8 для порта связи 1.На самом деле в данном адресе важны только биты 3 и 4. С их помощью кодируется номер передатчика порта связи. Все другие биты безразличны.

- Модификатор DXM должен быть сброшен.

Передача всегда инициируется готовностью приемника канала связи и готовностью передатчика канала связи. Как только буфер приемника загружается принятыми данными, посылается запрос к каналу DMA. Канал DMA анализирует состояние буфера передатчика, в который он запрограммирован записать данные. Если буфер передатчика канала связи пуст – происходит пересылка.

13.7 Канал 12

Канал 12 поддерживает возможность пересылки данных от ведущего устройства во внутреннюю память процессора. В качестве ведущего устройства может выступать только модуль цифрового смесителя (ЦС). Блок данных в TCB приемника описывается стандартным образом. Параметры работы канала и завершения работы канала определяются стандартным образом. Единственным ограничением в TCB является то, что тип обмена TY должен быть равен 2 или 3, т.е. приемник данных всегда внутренняя память.

Канал 12 имеет буфер данных и запрашивает выполнение транзакции, как только в буфере появляются данные. Размер буфера данных составляет четыре 128-разрядных слова. Требованием в работе канала является соответствие размера данных, записываемых в буфер, размеру операнда, который определяется полем LEN в регистре TCB.

Канал принимает данные в буфер только, когда он активен. Если происходит запись в буфер при неактивном канале – генерируется флаг ошибки.

13.8 Пример

Рассмотрим пример использования каналов 4-11 для организации пересылок с учетом новых возможностей. Допустим, необходимо переслать массив из 8-ми чисел из одной области внутренней оперативной памяти в другую область внутренней оперативной памяти, используя при этом в качестве инициатора пересылки таймер 0 процессора 1986. Решим задачу с использованием каналов 4 и 8, а также с использованием интерфейса SPI1.
Введем обозначения:

```
#define base_DST_ch74   0x80000030
#define base_SRC_ch118  0x80000050
#define base_DMACFG     0x80000078
```

Наш массив для пересылки

```
.var spm[8] = {0x11111111, 0x11111122,0x11111333,0x11114444,
               0x11155555, 0x11666666,0x17777777,0x88888888};
```

Предполагаем, что таймер 0 у нас уже включен, работает и с определенным периодом формирует запросы к каналам DMA. Поскольку каналы DMA не включены и не настроены на прием запросов от таймера, то никаких событий не происходит.
Включим SPI1 интерфейс для передачи 32-разрядных данных в тестовом режиме с включенной обратной связью.

```
                j8 = base_SPI1;;

        xr0 = 0x1001f;;
        [j8+0] = xr0;; // cr1
        xr0 = 0x0d;;   //
        [j8+1] = xr0;; // cr2
        xr0 = 0xff;;
        [j8+3] = xr0;; // st
```

Определим источники запросов для активации каналов 4 и 8. Канал 4 будем использовать для пересылки данных из памяти в передатчик SPI1 по запросу таймера 0 (запрос 0). Канал 8 будем использовать для пересылки данных от приемника SPI1 во внутреннюю память. Итого код активации канала 4 равен 20 (Таблица 84), а код активации канала 8 равен 12 (Таблица 85). Формируем 64-разрядный код и пишем его в регистр конфигурации

```
            j8 = base_DMACFG;;
      xr0 = 0x00040000;;   // GTMR0 req
            xr1 = 0x0010000c;;
      l[j8+0] = xr1:0;;
```

Определяем устройство, в которое канал 4 будет записывать данные для передачи. Адрес устройства есть буфер данных SPI1. Активируем регистр DCA, записав в поле активации значение 001b.

```
      j8 = base_DST_ch74;;
xr0 = 0x80000362;;   // dest  SPI1
      xr1 = 0x20000000;;// код активации
l[j8+0] = xr1:0;;
```

Определяем устройство, из которого канал 8 будет читать данные. Отметим, что в данном случае можно было бы обойтись без новых функций, т.к. источник запроса и обслуживающее устройство это одно и тоже устройство. Но будем использовать регистр DCA. Запишем адрес-источник данных – это буфер данных приемника SPI1. Код активации равен 100b что означает обращение к устройству на СОК-шине.

```
j8 = base_SRC_ch118;;
xr0 = 0x80000362;; // src  SPI1
      xr1 = 0x80000000;;// код активации
l[j8+0] = xr1:0;;
```

Включаем каналы «стандартным» образом. Включаем канал 4 для передачи. Адрес источника – это массив SPM. Передача идет 32-разрядными словами. Количество слов равно 8.

```
XR0 = spm;; // address
XR1 = 0x00080001;;
XR2 = 0x00000000;;
XR3 = 0x42080000;;

DC4    = XR3:0;;
```

Включаем канал 8 «стандартным» образом. Адрес приемника равен 0x40000. Количество слов равно 8. Обмен идет 32-разрядными словами.

```
XR0 = 0x00040000;; // internal address
  XR1 = 0x00800001;;
  XR2 = 0x00000000;;
  XR3 = 0x53080000;;

  DC8    = XR3:0;;
```

Ожидаем некоторый промежуток времени, пока пересылка закончится. Это можно сделать, анализируя регистр состояния DMA либо по прерыванию от канала 8.Проверяем, что данные успешно переданы.

```
    lc1 = 8;;

  j0 = spm;;
j1 = 0x40000;;


  rpp_dd:
j2 = [j0+=1];;
j3 = [j1+=1];;
```

```
        comp(j2,j3);;
    if njeq, jump error (NP);;
     if nlc1e, jump rpp_dd;;
```

14 Интерфейс внешней памяти

Большинство выводов процессора являются универсальными, т.е. они индивидуально могут быть запрограммированы как вход или выход. Вместе с тем процессор может быть сконфигурирован так, что набор внешних контактов будет образовывать внешний порт, который обеспечивает доступ к внешней памяти, устройствам ввода-вывода, отображаемыми на пространство памяти.

Внешний порт процессора обеспечивает связь с внешней памятью и периферией. Внешняя память и периферия включены в унифицированное адресное пространство процессора. Внутренние шины процессора мультиплексируются на внешний порт, создавая шину внешней системы с одной 22-битной шиной адресов и одной 16\32-битной шиной данных. Внешняя память и устройства могут иметь разрядность 16 или 32 бита. Процессор автоматически упаковывает внешние данные в слова длиной 32, 64 или 128 бит. Встроенное декодирование адресных линий (для генерации сигналов выбора блоков памяти) обеспечивает адресацию устройств внешней памяти. Специальные линии контроля также формируются для упрощения адресации страничного режима динамической памяти.

Процессор использует синхронный конвейер при обменах на внешней шине. Это позволяет осуществлять связь с синхронной DRAM. Опция допускает асинхронную связь для медленных устройств.

Внешние данные могут быть доступны по каналам DMA или через ядро. При доступе посредством ядра задержка чтения может быть значительной, что может вызывать длительные остановы процессора.

Программируемое количество тактов ожидания позволяет работать с периферией, имеющей различные требованиями на время доступа.

Внешние выводы внешней памяти приведены в Таблице 98

Таблица 98 – Внешние выводы

Обозначение вывода (основное)	Обозначение вывода памяти	Тип вывода	Функциональное назначение
PC[7]	SCLK	O	Внешний интерфейс. Синхросигнал
PC[8]	SD_CKE	I/O	Интерфейс SDRAM. Разрешение синхронизации
PC[9]	MSSD[0]	I/O	Интерфейс SDRAM. Активация модуля 0
PC[10]	MSSD[1]	I/O	Интерфейс SDRAM. Активация модуля 1
PC[11]	MSSD[2]	I/O	Интерфейс SDRAM. Активация модуля 2
PC[12]	MSSD[3]	I/O	Интерфейс SDRAM. Активация модуля 3
PC[13]	SD_nCAS	I/O	Интерфейс SDRAM. Выбор колонки
PC[14]	SD_nRAS	I/O	Интерфейс SDRAM. Выбор строки
PC[15]	SD_nWE	I/O	Интерфейс SDRAM. Признак записи
PC[16]	SD_DQM	I/O	Интерфейс SDRAM. Маска
PC[17]	SD_A10	I/O	Интерфейс SDRAM. Бит 10 адреса
PC[18]	nMS[1]	I/O	Интерфейс статической памяти. Выбор типа 1
PC[19]	nMS[0]	I/O	Интерфейс статической памяти. Выбор типа 0
PC[20]	nBMS	I/O	Интерфейс статической памяти. Выбор внешней памяти начальной загрузки
PC[21]	nRD	I/O	Интерфейс статической памяти. Строб чтения
PC[22]	nWR	I/O	Интерфейс статической памяти. Строб записи
PC[23]	ACK	I/O	Интерфейс статической памяти. Сигнал готовности
PE[0]	A[0]	O	Внешний интерфейс. Шина адреса
PE[1]	A[1]	O	Внешний интерфейс. Шина адреса

Обозначение вывода (основное)	Обозначение вывода памяти	Тип вывода	Функциональное назначение
PE[2]	A[2]	O	Внешний интерфейс. Шина адреса
PE[3]	A[3]	O	Внешний интерфейс. Шина адреса
PE[4]	A[4]	O	Внешний интерфейс. Шина адреса
PE[5]	A[5]	O	Внешний интерфейс. Шина адреса
PE[6]	A[6]	O	Внешний интерфейс. Шина адреса
PE[7]	A[7]	O	Внешний интерфейс. Шина адреса
PE[8]	A[8]	O	Внешний интерфейс. Шина адреса
PE[9]	A[9]	O	Внешний интерфейс. Шина адреса
PE[10]	A[10]	O	Внешний интерфейс. Шина адреса
PE[11]	A[11]	O	Внешний интерфейс. Шина адреса
PE[12]	A[12]	O	Внешний интерфейс. Шина адреса
PE[13]	A[13]	O	Внешний интерфейс. Шина адреса
PE[14]	A[14]	O	Внешний интерфейс. Шина адреса
PE[15]	A[15]	O	Внешний интерфейс. Шина адреса
PE[16]	A[16]	O	Внешний интерфейс. Шина адреса
PE[17]	A[17]	O	Внешний интерфейс. Шина адреса
PE[18]	A[18]	O	Внешний интерфейс. Шина адреса
PE[19]	A[19]	O	Внешний интерфейс. Шина адреса
PE[20]	A[20]	O	Внешний интерфейс. Шина адреса
PE[21]	A[21]	O	Внешний интерфейс. Шина адреса
XTO	A[22]	O	Внешний интерфейс. Шина адреса
PD[0]	D[0]	I/O	Внешний интерфейс. Шина данных
PD[1]	D[1]	I/O	Внешний интерфейс. Шина данных
PD[2]	D[2]	I/O	Внешний интерфейс. Шина данных
PD[3]	D[3]	I/O	Внешний интерфейс. Шина данных
PD[4]	D[4]	I/O	Внешний интерфейс. Шина данных
PD[5]	D[5]	I/O	Внешний интерфейс. Шина данных
PD[6]	D[6]	I/O	Внешний интерфейс. Шина данных
PD[7]	D[7]	I/O	Внешний интерфейс. Шина данных
PD[8]	D[8]	I/O	Внешний интерфейс. Шина данных
PD[9]	D[9]	I/O	Внешний интерфейс. Шина данных
PD[10]	D[10]	I/O	Внешний интерфейс. Шина данных
PD[11]	D[11]	I/O	Внешний интерфейс. Шина данных
PD[12]	D[12]	I/O	Внешний интерфейс. Шина данных
PD[13]	D[13]	I/O	Внешний интерфейс. Шина данных
PD[14]	D[14]	I/O	Внешний интерфейс. Шина данных
PD[15]	D[15]	I/O	Внешний интерфейс. Шина данных
PD[16]	D[16]	I/O	Внешний интерфейс. Шина данных
PD[17]	D[17]	I/O	Внешний интерфейс. Шина данных
PD[18]	D[18]	I/O	Внешний интерфейс. Шина данных
PD[19]	D[19]	I/O	Внешний интерфейс. Шина данных
PD[20]	D[20]	I/O	Внешний интерфейс. Шина данных
PD[21]	D[21]	I/O	Внешний интерфейс. Шина данных
PD[22]	D[22]	I/O	Внешний интерфейс. Шина данных
PD[23]	D[23]	I/O	Внешний интерфейс. Шина данных
PD[24]	D[24]	I/O	Внешний интерфейс. Шина данных
PD[25]	D[25]	I/O	Внешний интерфейс. Шина данных
PD[26]	D[26]	I/O	Внешний интерфейс. Шина данных
PD[27]	D[27]	I/O	Внешний интерфейс. Шина данных
PD[28]	D[28]	I/O	Внешний интерфейс. Шина данных
PD[29]	D[29]	I/O	Внешний интерфейс. Шина данных
PD[30]	D[30]	I/O	Внешний интерфейс. Шина данных
PD[31]	D[31]	I/O	Внешний интерфейс. Шина данных

14.1 Внешняя память

Интерфейс внешней памяти позволяет организовать подключение различных типов памяти к процессору. Основными типами памяти являются асинхронная или синхронная статическая память (SRAM), а также синхронная динамическая память (SDRAM).

Контроллер SDRAM расположен на кристалле процессора. Он управляет всеми контрольными сигналами SDRAM (RAS, CAS, SDWE, SDCKE, SDQM) и поддерживает инициализацию и регенерацию микросхем SDRAM. Максимальная производительность SDRAM при передаче данных равна одной передаче за один тактовый цикл. Производительность может удерживаться близко к максимальной, если последовательные доступы выполняются к той же странице

Процессор поддерживает также протокол медленного устройства. Этот протокол следует использовать для устройств не критичных в плане производительности.

Внешняя шина включает в себя шину данных, с пропускной способностью 32 или 16 разрядов, 23-разрядную шину адреса и управляющие сигналы. Процессор всегда ведущий на внешней шине.

В подразделе «Порты общего назначения GPIO» отмечалось, что интерфейс внешней памяти реализуется на базе портов общего назначения PE и PD. Определяющим конфигурационным регистром является регистр PX_ALT (Таблица 70). Для реализации полнофункционального внешнего интерфейса (32-разрядная шина данных и 23-разрядная шина адреса) в регистр PX_ALT необходимо записать значение 0_0111_1111b. Это соответствует тому, шина данных и шина адреса полностью контролируются модулем интерфейса внешней памяти. Линии управления внешнего интерфейса используют порт PC и также нуждаются в их конфигурировании. Регистр PC_ALT позволяет задать необходимую конфигурацию в зависимости от типа используемой памяти.

После того, как модуль внешнего интерфейса получает контроль над внешними выводами микросхемы, он может быть дополнительно сконфигурирован под различные настройки путем записи в конфигурационные регистры SYSCON и SDRCON. Данные регистры задают параметры непосредственно для контроллера внешнего интерфейса. Регистры контроллера внешнего интерфейса доступны как регистры периферийных устройств на SOC-шине. Базовый адрес регистров 0x8000_0080. Список доступных регистров приведен ниже (Таблица 99). Номер регистра образуется конкатенацией номера группы (старшие 6 бит) и номера регистра в группе (младшие 5 бит).

Таблица 99 – Группа регистров интерфейса шины

Имя	Описание	Номер\смещение	Значение после сброса
SYSCON	Регистр конфигурации внешней шины	0x0480 \ 0	0x0008_0067
BUSLOCK	Регистр блокировки внешней шины	0x0483 \ 3	0
SDRCON	Регистр конфигурации SDRAM	0x0484 \ 4	0
SDRSRF	Запрос режима selfrefresh	0x0485 \ 5	0
SYSTAT	Регистр статуса системы; только чтение	0x0486 \ 6	0
SYSTATCL	Регистр статуса системы; адрес для сброса ошибок. Только чтение	0x0487 \ 7	0
SD_REF	16-ти разрядный регистр периода регенерации SDRAM	Адрес 0x8000009C	0x0000_00FF

14.1.1 Особенности внешней шины

- Ширина шины данных: 16 или 32 разряда;
- Конвейерные передачи с программируемым числом этапов конвейера;
- Программируемые состояния простоя (IDLE);
- Протокол поддерживает циклы ожидания, добавляемые ведомым устройством с использованием вывода ACK;
- Интерфейс EPROM и FLASH: 8-разрядная шина данных с фиксированным числом циклов ожидания, поддержкой чтения и записи;
- SDRAM-интерфейс;
- Поддержка медленных устройств ввода/вывода;
- Поддержка передач через DMA для внешних устройств ввода/вывода в режиме квитирования.

14.1.2 Внешние контакты ввода/вывода интерфейса шины

Ниже представлено описание основных выводов внешнего интерфейса, используемых для реализации обмена со всеми типами памяти. Буква «**n**» в начале названия вывода указывает на низкий активный уровень сигнала.

- **ADDR22-0**

Шина адреса. Выдает на эти линии адрес устройства, к которому выполняется доступ.

- **DATA31–0**

Шина данных с тремя состояниями. Процессор выдает данные на эти выводы или принимает данные или команды с помощью этих выводов. Имеется возможность задания размера шины (16 или 32 бита) для работы с различными устройствами системы.

- **nRD**

Чтение памяти. nRD активируется всякий раз, когда процессор осуществляет чтение из подчиненного устройства системы. Активный уровень низкий. Не используется при доступе к SDRAM.

- **nWR**

Запись слова. nWR активируется в случае, когда происходит запись во внешнее устройство. nWR изменяется одновременно с выводами ADDR. Активный уровень низкий. Не используется при доступе к SDRAM.

- **ACK**

Подтверждение готовности. Процессор анализирует вход ACK во время доступа к устройствам. Внешние подчиненные устройства могут деактивировать (установить в низкий уровень) ACK чтобы добавить состояние ожидания во время доступа к ним. ACK используется подчиненными устройствами на фазе данных. Высокий уровень на линии ACK означает готовность устройства, низкий – требование подождать. ACK не используется при доступе к SDRAM.

- **nBMS**

Выбор загрузочной памяти. nBMS активизируется, когда процессор выполняет доступ к загрузочной памяти (EPROM или флэш). nBMS изменяется одновременно с выводами ADDR. Активный уровень низкий.

– **nMS1-0**

Выбор памяти. nMS0 или nMS1 активируются в то время, когда процессор желает получить доступ к банкам памяти 0 или 1 соответственно. Выходы nMS1–0 являются декодированными сигналами адреса памяти и изменяются параллельно с выводами ADDR. Когда линии ADDR31–27 равны 0b00110, nMS0 активирован. Когда линии ADDR31–27 равны 0b00111, nMS1 активирован. Активный уровень низкий.

– **MSSD3–0**

Выборка памяти SDRAM. MSSD0, MSSD1, MSSD2 или MSSD3 активны (один из четырех), всякий раз, когда процессор выполняет доступ к пространству памяти SDRAM. MSSD3–0 являются выводами декодированного адреса памяти и активны в то время, когда процессор выполняет командный цикл SDRAM. Некоторые команды активируют сразу все четыре линии.

– **nRAS**

Выборка адреса строки. Во время чтения или записи SDRAM низкий уровне сигнала RAS указывает на то, что на шине адреса находится достоверный адрес строки. В остальных случаях доступа к SDRAM он определяет тип операции для выполнения в соответствии со спецификацией SDRAM.

– **nCAS**

Выборка адреса столбца. Во время чтения или записи SDRAM низкий уровне сигнала CAS указывает на то, что на шине адреса находится достоверный адрес столбца. В остальных случаях доступа к SDRAM он определяет тип операции для выполнения в соответствии со спецификацией SDRAM.

– **DQM**

Маска слова данных SDRAM. При высоком уровне сигнала переводит буферы данных DQ SDRAM в третье состояние. При чтении всегда равен нулю. При операциях записи DQM является активным (равным 1), для запрещения записи слова.

– **SDA10**

Вывод разряда 10 адреса SDRAM. SDRAM использует отдельный вывод адресной линии номер 10 из-за того, что она используется в специальных операциях и может задействоваться в то время, когда процессор выполняет на внешней шине транзакцию не связанную с SDRAM.

– **SDCKE**

Разрешение тактового сигнала SDRAM. Вход синхронизации микросхемы динамической памяти может быть постоянно активным, но микросхема будет управляться данным синхросигналом, только если на линии SDCKE высокий уровень. Низкий уровень сигнала SDCKE используется для перевода памяти в режим саморегенерации.

– **nSDWE**

Разрешение записи SDRAM. При низком уровне сигнала во время активного CAS, SDWE указывает на выполнение записи SDRAM. При высоком уровне сигнала во время активного CAS, SDWE указывает на выполнение чтения SDRAM. В остальных операциях SDRAM, SDWE определяет тип операции для выполнения в соответствии со спецификацией SDRAM.

14.1.3 Структура интерфейса внешней памяти

Передаваемые на внешней шине данные могут быть стандартными словами (32 разряда), двойными словами (64 разряда) или счетверенными словами (128 разрядов). Адрес этих слов должен быть выровнен по их размеру. Поскольку ширина шины (16 или 32 разряда) может быть меньше размера передачи, передача данных распределяется по нескольким циклам внешней шины.

Внешний порт (EBIU) всегда является мастером на внешней шине, а также мастером и подчиненным на внутренних шинах процессора. Рисунок 19 показывает структуру внутренних шин, соединяющих внешний порт с внутренними модулями процессора.

В структуре внешнего порта два главных элемента передачи данных:

- выходное FIFO (OFIFO) передачи запросов на внешнюю шину;
- входное FIFO (IFIFO) входных запросов с внешней шины или возвращаемых данных с внешней шины.

Буфер IFIFO используется в случае передачи данных во внутренний приемник в ответ на чтение ядром процессора или контроллером DMA устройств на внешней шине. В этой ситуации внешний интерфейс работает как мастер на внешней шине и возвращает прочитанные данные. Далее данные отправляются на SOC-шину (контроллер DMA или устройства) или в ядро процессора (в IFIFO SOC-интерфейса).

Выходной буфер OFIFO внешнего порта используется для всех транзакций, направленных во внешнее адресное пространство. Это могут быть запросы процессорного ядра или контроллера DMA.

Потоки данных, которые имеют место при различных вариантах передач, представлены на рисунке 53. Возможны операции чтения и операции записи. Операции, выполняемые ядром или контроллером DMA.

Если операция является операцией записи, она завершается:

- для ядра процессора в момент записи данных в SOC OFIFO;
- для контроллера DMA в момент записи в EBIU OFIFO.

Если выполняется операция чтения, данные проходят следующие этапы:

- При чтении ядром процессора, запрос помещается в SOC OFIFO, далее он перемещается в EBIU OFIFO, где читается и анализируется внешним интерфейсом. Выполняется цикл чтения внешнего устройства и прочитанные данные размещает в EBIU IFIFO, откуда они пересылаются в SOC IFIFO. SOC интерфейс принятые данные отправляет в регистр назначения.
- При чтении контроллером DMA, запрос помещается в EBIU OFIFO, где затем читается и анализируется внешним интерфейсом. Выполняется цикл чтения внешнего устройства и прочитанные данные размещает в EBIU IFIFO, откуда они пересылаются в буфер контроллера или буфера передатчиков устройств.

При доступе к ресурсам со стороны многих устройств используется приоритет доступа. EBIU IFIFO участвует в доступе к SOC-шине и к SOC-IFIFO. При доступе к SOC-шине приоритет следующий:

- SOC OBUF (наивысший);
- EBIU IFIFO;
- процессорное ядро (низший).

При доступе к SOC IFIFO приоритет следующий:
- чтение ядром регистра SOC шины;
- EBIU IFIFO;
- контроллер DMA.

Приоритет устройств фиксированный и не зависит от приоритета транзакции DMA или приоритета транзакции внешнего мастера.

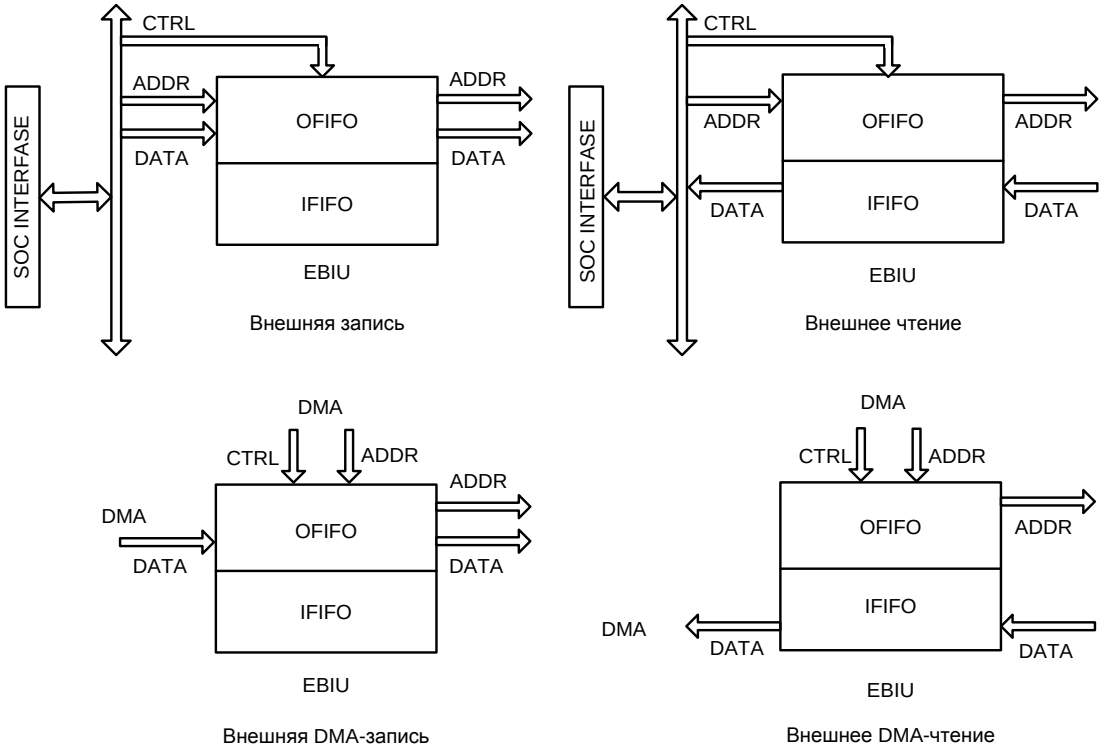


Рисунок 53 – Передача данных через интерфейс внешней шины (EBIU)

14.1.4 Программирование регистра SYSCON

Регистр SYSCON является регистром конфигурации контроллера внешнего интерфейса и должен быть запрограммирован до того, как начнется передача данных на шине (если значения по умолчанию должны изменяться). В режиме пользователя доступ к SYSCON невозможен. Описание разрядов регистра приведено в Таблице 100.

Таблица 100 – Регистр SYSCON

Бит	Имя	Назначение
0	BNK0IDLE	Вставка пустого цикла между операциями доступа к банку памяти MS0: 1 – вставить цикл; 0 – нет
2:1	BNK0WAIT	Количество внутренних циклов ожидания при обращении к банку MS0: 00 – 0 циклов 01 – 1 цикл 10 – 2 цикла 11 – 3 цикла

Бит	Имя	Назначение
4:3	BNK0PIPE	Глубина конвейера банка памяти MS0: 00 – 1 цикл 01 – 2 цикла 10 – 3 цикла 11 – 4 цикла
5	BNK0SLOW	Тип протокола обмена для банка MS0: 1 – медленный (асинхронный) 0 – синхронный (конвейерный)
6	BNK1IDLE	Вставка пустого цикла между операциями доступа к банку памяти MS1: 1 – вставить цикл 0 – нет
8:7	BNK1WAIT	Количество внутренних циклов ожидания при обращении к банку MS1: 00 – 0 циклов 01 – 1 цикл 10 – 2 цикла 11 – 3 цикла
10:9	BNK1PIPE	Глубина конвейера банка памяти MS1: 00 – 1 цикл 01 – 2 цикла 10 – 3 цикла 11 – 4 цикла
11	BNK1SLOW	Тип протокола обмена для банка MS1: 0 – медленный (асинхронный) 1 – синхронный (конвейерный)
17:12		Зарезервировано
18	-	Всегда 0
19	MEMWIDTH	Ширина шины данных при обращении к внешней памяти 1 – 16 бит 0 – 32 бита
21:20		Зарезервировано
23:22		Всегда 0
24		Режим работы входного ФИФО внешнего интерфейса 0 – нормальный режим 1 – ускорение передачи данных Бит можно только установить. Сброс только аппаратно.
25		Режим работы выходного буфера внешнего интерфейса 0 – нормальный режим 1 – ускорение передачи данных Бит можно только установить. Сброс только аппаратно
26		Всегда 0
27	XSDM	Модификация адресного пространства. 0 – стандартное адресное пространство. 1 – модифицированное адресное пространство в котором адреса с 0x10000000 по 0x2FFFFFFF соответствуют адресному пространству динамической памяти. Используется для поддержки байтовой адресации внешней памяти
31:28		Всегда 0

Ширина шины данных

Физически шина данных внешнего интерфейса имеет 32 линии. Однако это не значит, что подключаемая память или внешнее устройство обязательно должны иметь такую же разрядность шины данных. С помощью регистра PX_ALT можно задать ширину шины данных равной 32, 16 или 8 бит. Контроллер внешнего

интерфейса будет управлять тем количеством выводов шины данных, значение которых определено в регистре PX_ALT. Вместе с этим сам контроллер поддерживает алгоритм работы с 32-разрядной или 16-разрядной внешней шиной данных. Это означает, что если в регистре SYSCON[19] задана разрядность шины данных 32 бита, то, несмотря на установки в регистре PX_ALT, интерфейс будет вести себя так, как будто бы шина имеет разрядность 32 бита. Если в регистре SYSCON[19] установлена разрядность шины данных 16 бит, все словные операции интерфейс будет превращать в две последовательные операции по 16 бит.

Протокол медленного устройства для шины

Для настройки протокола медленного устройства устанавливается в 1 разряд протокола медленного устройства BNK#SLOW (разряд 5 для банка 0, разряд 11 для банка 1). В случае выбора протокола медленного устройства некоторые поля (поле глубины конвейера PIPE, разряд IDLE) являются безразличными и не используются. Поле ожидания определяет число внутренних циклов ожидания при медленных доступах. Если число внутренних ожиданий равно нулю, механизм внешних ожиданий (анализ входа АСК) не может использоваться для данных транзакций.

Конвейерный протокол для шины

Для выбора конвейерного протокола бит BNK#SLOW (HOSTSLOW) соответствующего банка памяти устанавливается в 0, а поле PIPE устанавливается на заданную глубину конвейера (0b00 для одного цикла, 0b01 для двух циклов, 0b10 для трех циклов и 0b11 для четырех циклов). Так определяется глубина конвейера только для транзакций чтения. Глубина конвейера для транзакций записи всегда равна одному циклу. Разряд IDLE устанавливается, если на шине есть несколько подчиненных устройств в одном банке памяти, с целью предотвращения конфликта на шине данных между различными подчиненными устройствами при последовательном чтении. Если банк памяти содержит только одно подчиненное устройство, разряд IDLE сбрасывается. Поле внутреннего ожидания безразлично при конвейерных транзакциях.

Начальные значения для работы шины

После сброса регистр SYSCON имеет начальное значение равное 0x80067. Программа начальной загрузки меняет содержимое регистра SYSCON в зависимости от выбранного способа загрузки (см. раздел «Начальный старт процессора» и текст загрузочной программы). Пользователь может не менять данное значение, если оно соответствует конфигурации его внешней шине. Начальное значение определяет:

- банк 0: медленный протокол, три состояния ожидания, включен холостой режим;
- банк 1: конвейерный протокол, конвейер глубиной 1 цикл, нет состояний ожидания, включен холостой режим;

В зависимости от выбранного способа загрузки BOOT[2:0] регистр SYSCON принимает значения, указанные в таблице 92.

Таблица 101 – Значение регистра SYSCON в зависимости от режима загрузки

BOOT[2:0]	Значение регистра SYSCON
000	0x809e7
001	0x9e7
010 – 110	0x80067

14.1.5 Интерфейс конвейерного протокола

Процессор использует конвейерный протокол для соединения с быстрой синхронной памятью, подключаемой к nMS1-0. Конвейерный протокол обеспечивает передачу данных со скоростью частоты внешней шины. При конвейерном доступе адрес и управляющие разряды выдаются одновременно в одном такте, а данные лишь спустя несколько циклов: от одного до четырех в зависимости от направления передачи и установленных параметров. Процессор может выдавать адрес следующей транзакции еще до прихода данных предыдущей транзакции.

Управляющие сигналы, используемые в конвейерных транзакциях:

- **nRD** – если активен указывает на транзакцию чтения;
- **nWR**– транзакция записи, если один из этих сигналов активный;
- **ACK** – управляется адресуемым подчиненным устройством в течение цикла данных. Если имеет высокий уровень, подчиненное устройство готово завершить цикл данных, в противном случае мастер шины должен ждать готовности подчиненного устройства.

Базовая конвейерная транзакция

Временная диаграмма базовой конвейерной транзакции представлена ниже (Рисунок 54). Во время адресного цикла, адрес передается вместе с сигналами RD или WR.

Цикл данных может начинаться с задержкой от одного до четырех тактов, в соответствии с параметрами, указанным в регистре конфигурации для подчиненного устройства, и направлением транзакции. Задержка между циклом адреса и циклом данных есть глубина конвейера. В примере глубина конвейера равна двум (Рисунок 54). В цикле данных данные передаются в соответствии с направлением транзакции (чтение или запись). Если подчиненное устройство готово к завершению транзакции, оно активирует сигнал ACK (равен 1) и принимает или выставляет данные. Если подчиненное устройство не готово, оно деактивирует сигнал ACK (равен 0) и задерживает данные.

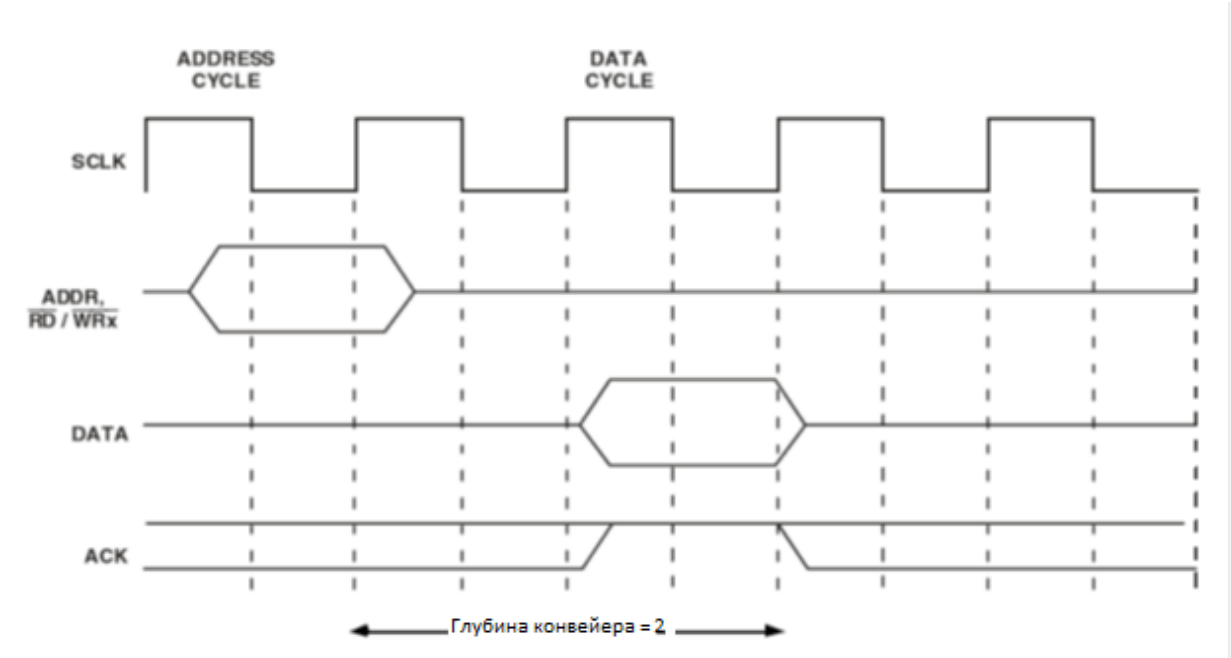


Рисунок 54 – Основные конвейерные передачи

Глубина конвейера программируется для каждого из банков памяти, а также зависит от типа транзакции (чтение или запись). Для записи глубина конвейера всегда равна 1, т.е. данные выставляются процессором на шину в следующем такте после выдачи адреса. При чтении из внешних банков памяти глубина конвейера программируется пользователем и может быть до четырех циклов. Глубина конвейера может быть выбрана индивидуально для каждого их банков в регистре SYSCON после сброса.

Одиночные транзакции занимают различное число циклов в зависимости от размера транзакции и ширины внешней шины:

- транзакция стандартного слова: один цикл;
- транзакция двойного слова на 32-разрядной шине: 2 цикла;
- транзакция двойного слова на 16-разрядной шине: 4 цикла;
- транзакция счетверенного слова на 32-разрядной шине: 4 цикла;
- транзакция счетверенного слова на 16-разрядной шине: 8 циклов;

Тип транзакции определяется сигналами RD и WR во время цикла адреса.

Циклы ожидания

В транзакциях чтения, если подчиненное устройство вовремя готово к циклу данных выполняемой транзакции, подчиненное устройство активирует сигнал ACK в цикле данных. Если подчиненное устройство не готово, оно деактивирует сигнал ACK в цикле данных и держит его неактивным (равным 0) пока не будет готово продолжить транзакцию. ACK также может быть деактивирован после транзакции записи, если подчиненное устройство не может вовремя отправить данные в место назначения.

Сигнал ACK может быть выставлен в любом цикле данных в многоцикловых транзакциях. Нет никаких ограничений на продолжительность удерживания ACK в нуле.

Рисунок 55 демонстрирует использование сигнала ACK. В этом примере, данные DA1 не готовы вовремя и выставляются на один такт позже. Сигнал ACK деактивируется во время цикла данных DA1 и активируется в следующем цикле, чтобы указать на достоверные данные. Из-за этого цикла ожидания:

- мастер берет с шины данные на один цикл позже;
- адресный цикл, следующий за сигналом ACK (AB2) растягивается на один цикл;
- все подчиненные устройства с приостановленными транзакциями во время цикла, следующего за сигналом ACK, замедляются на число циклов, которые были деактивированы сигналом ACK. Например, между адресным циклом AB0 и циклом данных DB0 проходит пять циклов, хотя глубина конвейера равна только четырем.

При этом адресные и управляющие сигналы на линиях в цикле, где ACK был в нуле, должны защелкиваться подчиненными устройствами.

Транзакции записи используют циклы ожидания ACK не так, как транзакции чтения. В случае записи подчиненное устройство активирует сигнал ACK, пока у него имеется свободное место в буфере приема.

Ниже представлен пример, в котором адресный цикл AA3 заставляет подчиненное устройство деактивировать сигнал ACK (Рисунок 56). Это может быть вызвано тем, что подчиненное устройство приняло 128-разрядное слово в буфер, и

он занят для следующей транзакции. Остановка удлиняет конвейер транзакции на два цикла. Циклы, которые были остановлены ACK, начинаются спустя один такт после деактивации сигнала ACK – например, AB2 и DB1 дополнены двумя циклами.

При этом все подчиненные устройства принимают адрес AB1, а устройство, которому адресованы данные DB0, также принимает эти данные во внутренний буфер.

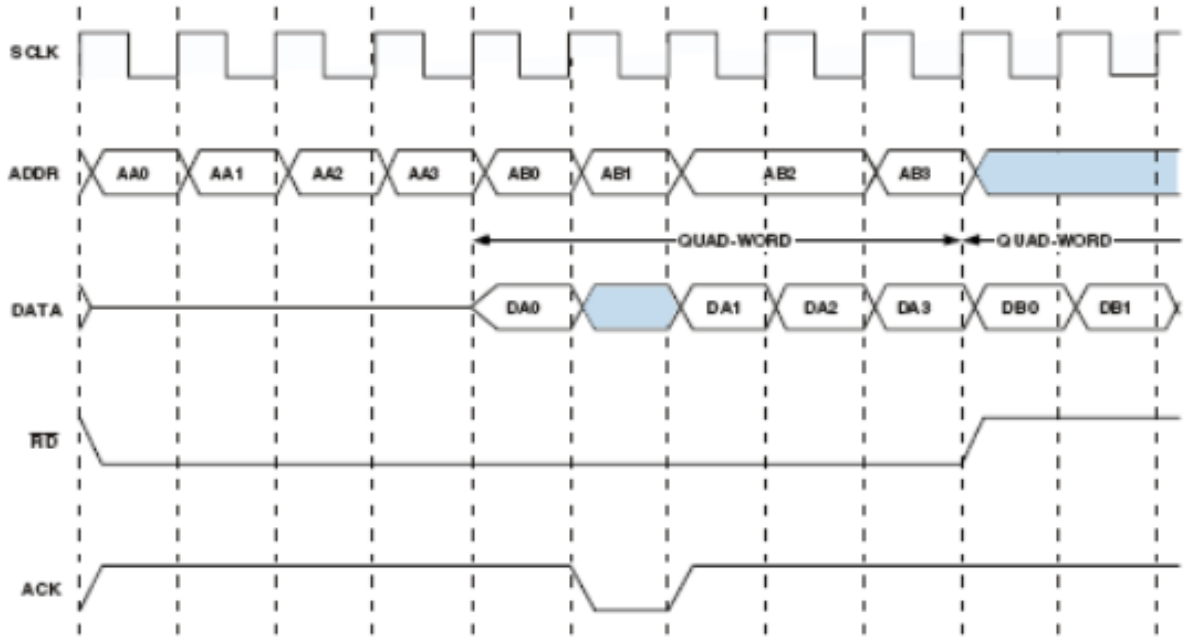


Рисунок 55 – Доступ к пакетному чтению с последующим пакетным чтением

Транзакции AB0 и AB1 адресуют другое подчиненное устройство (Рисунок 56). Если первое подчиненное устройство переводит сигнал ACK на низкий уровень и следующие транзакции соответствуют другому подчиненному устройству, новое подчиненное устройство транзакции не может управлять сигналом ACK до тех пор, пока первое подчиненное устройство не завершит транзакцию.

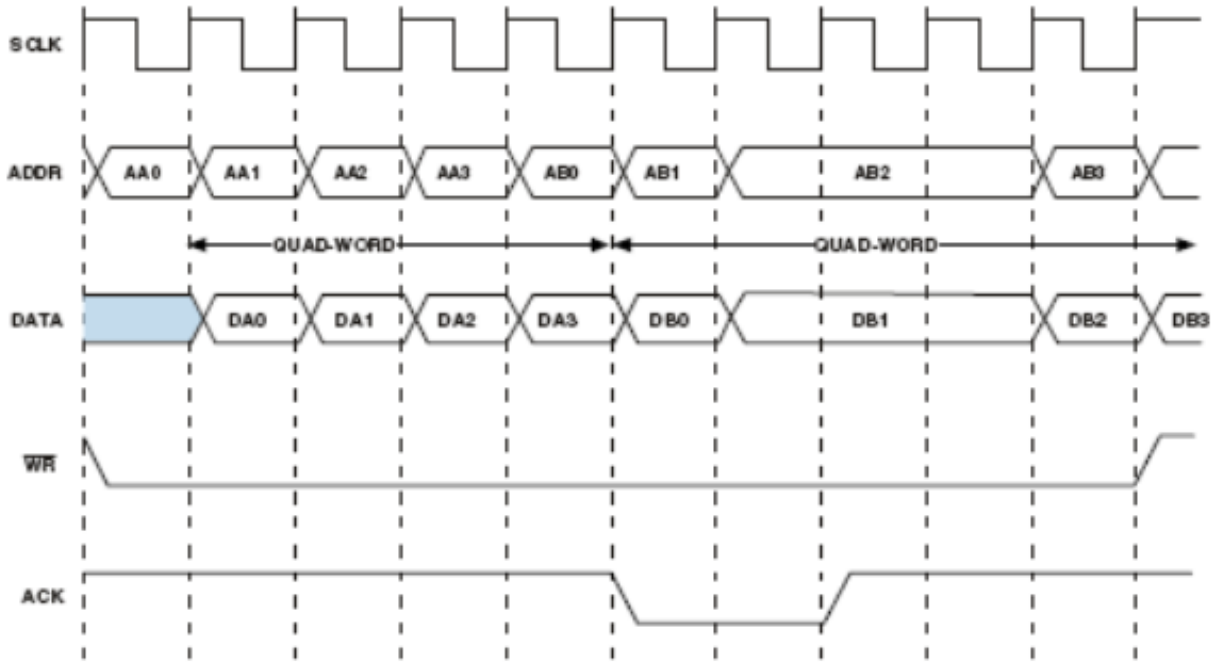


Рисунок 56 – Доступ к пакетной записи с последующей пакетной записью

14.1.6 Интерфейс протокола медленного устройства

Процессор поддерживает протокол медленных устройств, который может быть использован для простых устройств. Этот протокол может быть сконфигурирован для адресных пространств, принадлежащих банку 0 или банку 1. Протокол устанавливается посредством программирования соответствующих разрядов в регистре SYSCON:

- бит SLOW равен 1;
- биты PIPE глубины конвейера равны нулю;
- циклы ожидания WAIT программируются в соответствии с требованиями системы;
- цикл IDLE (холостой цикл) равен 1.

Целью таких установок является прямое соединение с простыми, медленными, некритическими модулями памяти или периферийными устройствами. Протокол может работать с синхронными и асинхронными устройствами. Базовый протокол с конфигурацией «без циклов ожидания» представлен ниже (Рисунок 57, Рисунок 58).

В первом цикле выдается адрес и активируется линия выбора соответствующего банка памяти.

Во втором цикле активируются сигналы RD или WR (в зависимости от типа транзакции). Если транзакция записи, в этом цикле процессор выдает данные. В случае транзакции чтения в этом цикле подчиненное устройство начинает выдавать данные. В случае чтения, в конце цикла процессор защелкивает данные с шины во внутреннем регистре.

В третьем цикле процессор продолжает удерживать адрес и данные (если выполняется запись), но деактивирует сигналы RD или WR. В случае чтения устройство перестает выдавать данные на шину.

Для конфигурации «без циклов ожидания», циклы ожидания не могут быть добавлены деактивирующим сигналом ACK.

Одно из ключевых требований при организации протокола медленного устройства – гарантированное время удержания адреса и данных. Из временной диаграммы видно, что процессор обеспечивает один такт предустановки адреса и один такт удержания адреса и данных. Дополнительные такты предустановки данных могут обеспечиваться внутренними программируемыми циклами ожидания или внешним сигналом ACK.

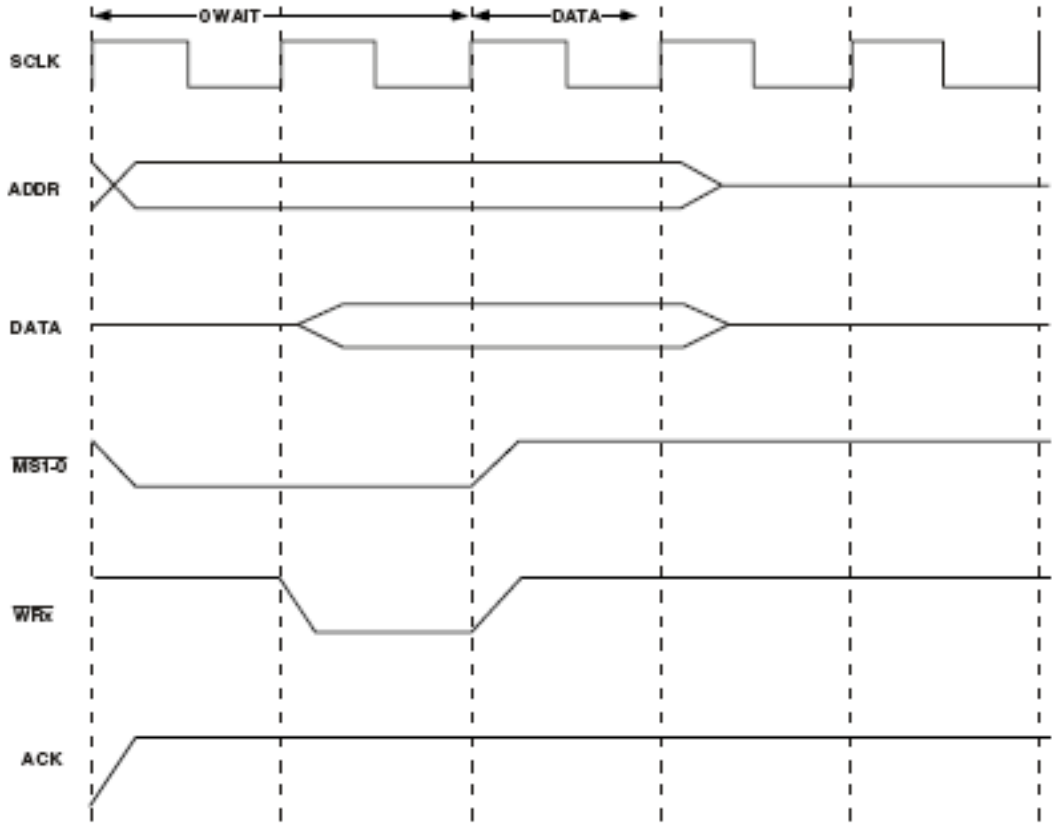


Рисунок 57 – Медленный протокол. Запись с нулевым временем ожидания

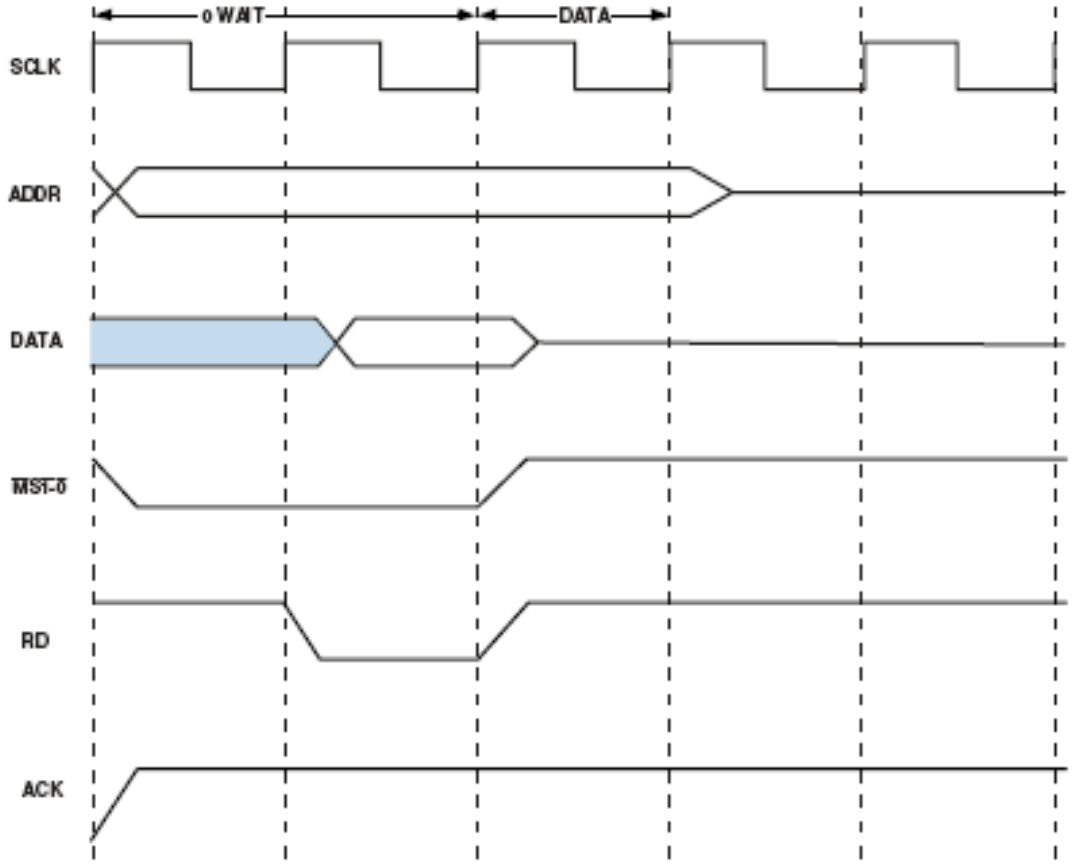


Рисунок 58 – Медленный протокол. Чтение с нулевым временем ожидания

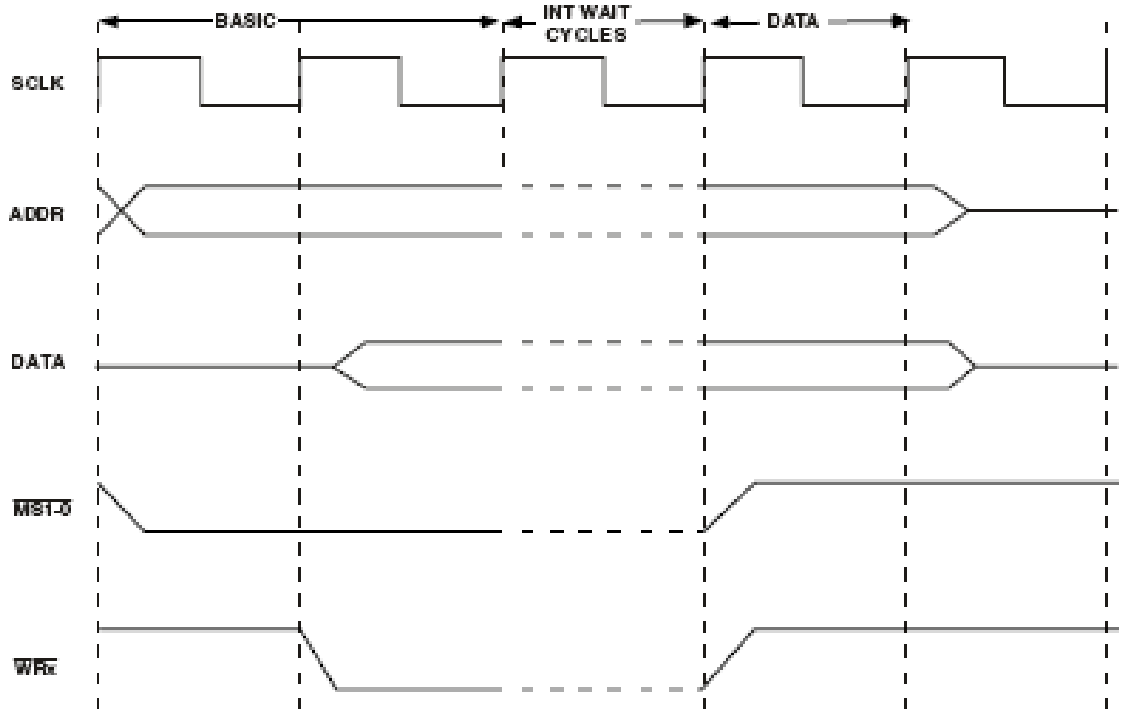


Рисунок 59 – Медленный протокол. Запись с циклами ожидания

В протоколе медленного устройства при необходимости могут быть вставлены циклы ожидания. Регистр SYSCON включает в себя поле задания некоторого числа внутренних циклов ожидания. Значения могут быть от нуля до трех.

Рисунок 59 и Рисунок 60 показывают медленные транзакции с одним или более циклами ожидания. Ситуация похожа на транзакцию без циклов ожидания, но второй цикл (цикл в котором активированы RD или WR) повторяется в соответствии с числом циклов внутреннего ожидания.

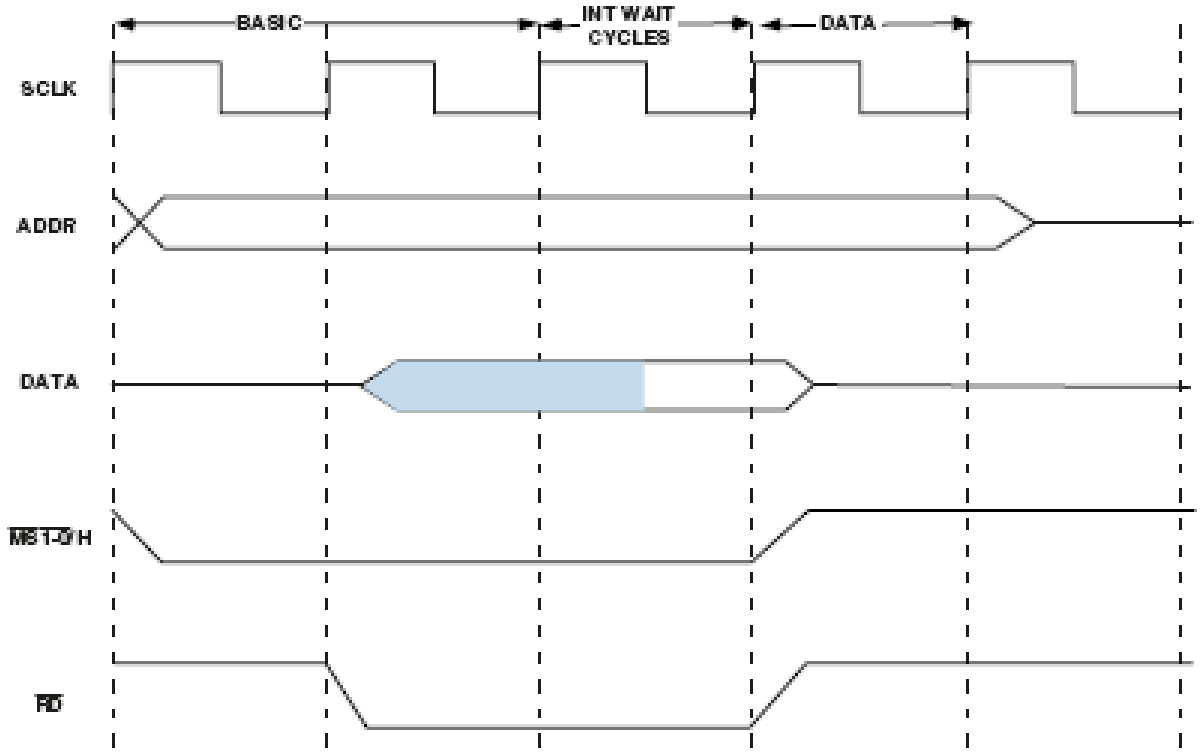


Рисунок 60 – Медленный протокол. Чтение с циклами ожидания

Дополнительные циклы внешнего ожидания могут быть вставлены путем деактивации сигнала ACK. Это может быть сделано только тогда, когда в регистре SYSCON поле WAIT не равно нулю. Для того, чтобы добавить циклы внешнего ожидания, сигнал ACK должен быть деактивирован в течение одного или более циклов перед последним циклом ожидания (см. Рисунок 61).

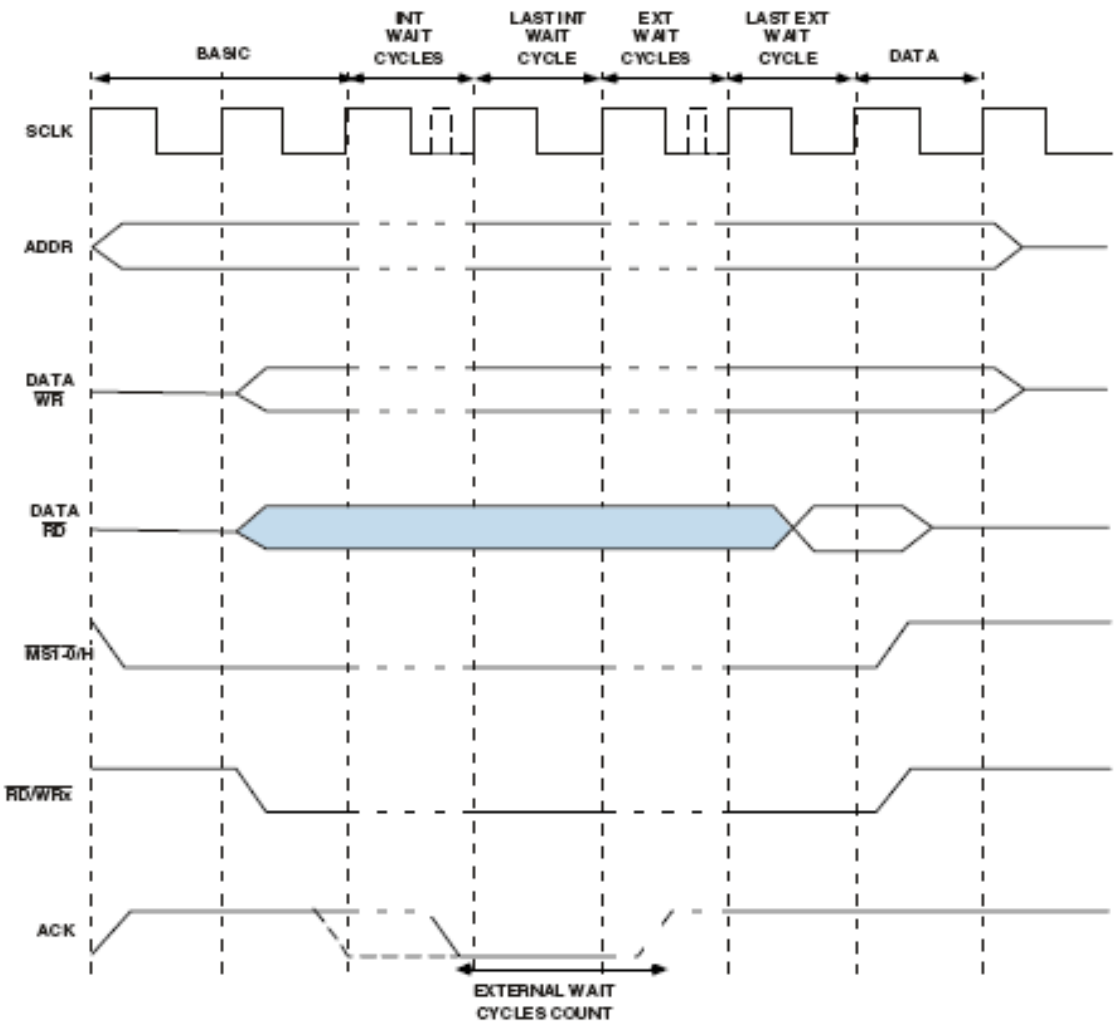


Рисунок 61 – Медленный протокол. Запись/чтение с внешними циклами ожидания

14.1.7 Интерфейс EPROM

Процессор имеет возможность работать с внешней 8-разрядной перепрограммируемой памятью типа EPROM. Во время сброса процессор может быть сконфигурирован для загрузки из внешнего EPROM. В этом случае программа загружается из EPROM во внутреннюю память автоматически как часть последовательности сброса. Для этой цели используется канал 0 DMA.

Поскольку разрядность шины данных памяти EPROM составляет 8 бит, внешний интерфейс процессора упаковывает принимаемые байты в 32-разрядные данные.

Процессор может работать с EPROM не только во время сброса. Имеется возможность доступа и в процессе выполнения программы. Однако в работе с EPROM имеется одна особенность – память EPROM не является частью адресного пространства процессора. Процессор поддерживает словную адресацию внешней памяти, а память EPROM имеет байтовую организацию. В связи с этим для работы с EPROM можно использовать только канал общего назначения DMA. Для работы с EPROM тип обмена в ТСВ канала устанавливается равным 6. При чтении EPROM, обмен может происходить только между EPROM и внутренней памятью. Внешний интерфейс определяет, что текущий запрос есть запрос DMA и его тип равен 6. Это позволяет интерфейсу выбрать для обмена протокол медленного устройства с фиксированным числом циклов внутреннего ожидания равным 16. При выполнении

внешней транзакции процессор активизирует линию BMS для выбора EPROM. Для передачи данных используются биты с 0 по 7-й шины данных.

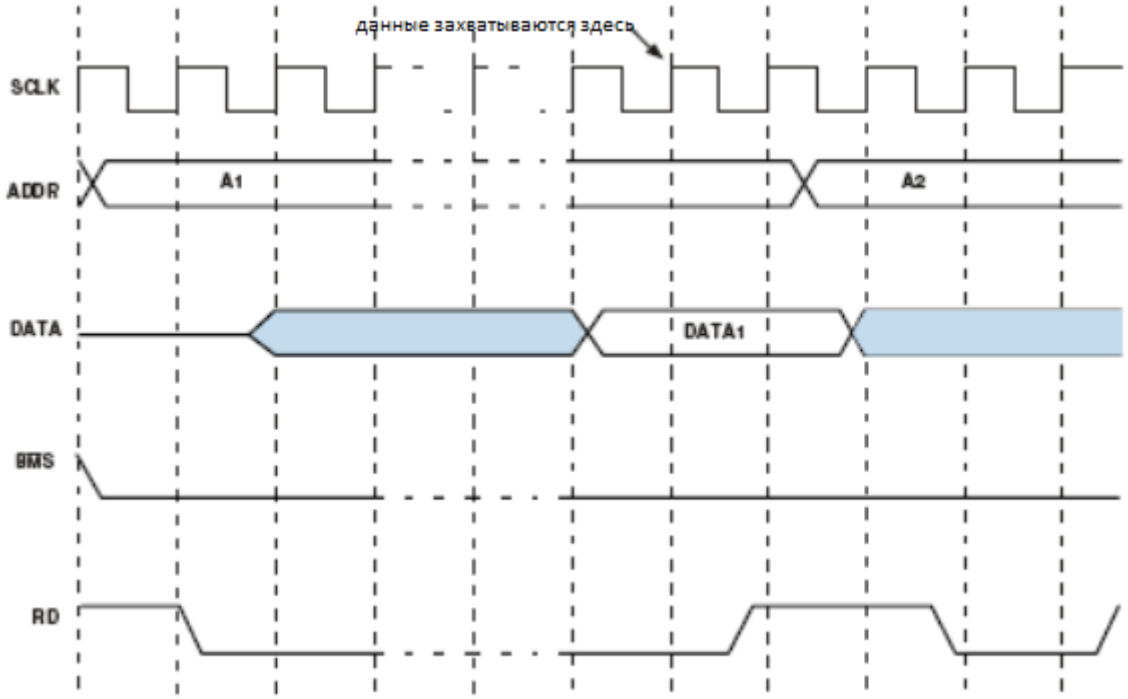


Рисунок 62 – Доступ к загрузке EPROM – 16 циклов ожидания

Возможно, как чтение внешней EPROM, так и операция записи для программирования памяти. Во время чтения интерфейс упаковывает байты в слова и передает их во внутреннюю память. Операция записи идентична чтению, но с одним отличием – при записи нет распаковки слова в байты. Процессор может выдавать на внешнюю шину только 32-разрядные слова, а EPROM может принимать только младший байт. Поэтому при записи программист сам должен подготовить информацию в удобном для записи виде. При записи EPROM источником данных может быть как внутренняя, так и внешняя память.

Во внутреннем или внешнем ОЗУ процессора данные должны быть организованы так, как показано ниже (Рисунок 63). Слева исходные данные для программирования. Справа показаны распакованные в слова байты для передачи во внешнюю EPROM.

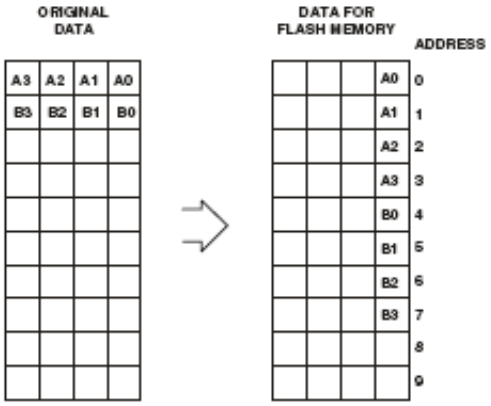


Рисунок 63 – Подготовка данных для записи в 8-разрядную EPROM

При выполнении обмена с EPROM, внешний интерфейс всегда устанавливается размер шины данных 32 разряда, несмотря на установки в

SYSCON. При этом адрес EPROM не должен быть в используемом адресном пространстве SDRAM или банков MS1 и MS0. Лучше всего старшие биты адреса установить в ноль. При чтении внешний интерфейс автоматически устанавливает размер данных равным квадрослову. Это позволяет произвести четыре цикла чтения, взять из них по одному байту и сформировать одно слово. Канал DMA при этом должен выполнять обмен словами, но увеличивать адрес на 4. Таким образом, первое считывание будет по адресам 0, 1, 2, 3, а второе по адресам 4, 5, 6, 7. Такой способ используется при чтении.

При записи используется размер данных, который задает канал DMA. По сути при записи внешний интерфейс работает с EPROM так как будто эта память имеет разрядность 32 бита и передает ей слова данных. Основной хитростью здесь является предварительная подготовка данных для записи в памяти.

14.1.8 Интерфейс SDRAM

Процессор имеет выделенное адресное пространство для подключения синхронной динамической памяти SDRAM. Четыре банка в этом адресном пространстве выбираются линиями выборки MSSD3–0. SDRAM доступно в адресном диапазоне с 0x4000 0000 до 0x7FFF FFFF.

Процессор поддерживает стандартную (3,3 В) и пониженной мощности (2,5 В) SDRAM. Память работает с использованием частоты внешней шины SCLK (системная частота). Все входы защелкиваются и все выходы корректны при положительном (из 0 в 1) перепаде частоты. Синхронный интерфейс позволяет выполнять передачу данных каждый такт. SDRAM поддерживает несколько типов пакетных доступов в зависимости от инициализации регистра режима SDRCON. Каждый тип доступа предваряется запуском соответствующей команды в SDRAM. Специальные режимы SDRAM должны инициализироваться в регистре SDRCON.

SDRAM имеет внутреннюю организацию в виде двух или четырех банков. Выводы выбора банка SDRAM определяют, к какому банку происходит обращение. SDRAM имеет программируемый параметр задержки чтения, который должен инициализироваться приложением в соответствии с типом устройства и рабочей тактовой частотой.

Для соответствия требуемым временным характеристикам SDRAM, процессор может выполнять конвейерную передачу адреса и управляющих команд SDRAM. Для этого используется бит глубины конвейера в регистре SDRCON.

Микросхемы SDRAM поставляются различными производителями. Каждый поставщик имеет свои временные требования касательно параметров задержки команд ACT - PRE (tRAS) и PRE - ACT (tRP). Для поддержки всех основных поставщиков и различных уровней скорости, регистр SDRCON программируем для того, чтобы разработчик мог удовлетворить временные требования микросхемы SDRAM.

Такими параметрами являются битовые поля: CAS задержка, PRE - RAS задержка, RAS - PRE задержка.

Наилучшая скорость обмена с SDRAM достигается при использовании блочного обмена, когда процессор выполняет несколько последовательных чтений или записей в один банк памяти. Интерфейс процессора имеет возможность отслеживания и поддержки пакетного доступа.

Особенностью работы динамической памяти является необходимость регенерации всех ячеек памяти в течение заданного интервала времени. Это гарантирует сохранность информации. Процессор имеет возможность программировать частоту регенерации для соответствия временным требованиям микросхемы.

При описании интерфейса SDRAM используются следующие определения:

- Команда активации банка ACT
  Активирует выбранный банк и фиксирует в нем новую строку. Должна использоваться перед командой чтения или записи.

- Длина пакета (Burst Length)
  Определяет количество слов, которые поступают на вход или выход SDRAM после детектирования команды чтения или записи. Микросхема памяти всегда программируется для длины пакета равного полной странице.

- Тип пакетирования (Burst Type)
  Определяет порядок, в котором SDRAM отправляет или принимает пакетные данные после детектирования команды чтения или записи. Процессор поддерживает только последовательный доступ.

- Задержка CAS (CAS Latency)
  Задержка, в тактах, между тем, когда SDRAM определяет команду чтения и когда данные поступают на выходные выводы. Величина запаздывания определяется уровнем скорости устройства и тактовой частотой шины. Приложение должно программировать этот параметр в регистре SDRCON.

- Маскирование ввода-вывода DQM данных
  Вывод DQM используется для маскирования контроллером операций записи.

- Регистр SDRCON
  Регистр, который содержит программируемые конфигурационные параметры для возможности работы контроллера SDRAM с микросхемой SDRAM.

- Регистр режима (Mode register)
  Регистр конфигурации SDRAM, который содержит определяемые пользователем параметры, согласованные с регистром SDRCON. Этот регистр находится внутри микросхемы памяти.

- Размер страницы (Page Size)
  Процессор поддерживает 1024-, 512-, и 256-словные размеры страниц. Размер страницы может быть запрограммирован в регистре SDRCON.

- Команда подзарядки (Precharge)
  Подзаряжает активный банк.

- Период регенерации (Refresh Rate)
  Программируемая величина в регистре SDRCON. Период регенерации позволяет приложениям координировать скорость SCLK с требуемой частотой регенерации SDRAM.

- Саморегенерация (Self-Refresh)
  Состояние микросхемы памяти, в котором она использует внутренний таймер и периодически инициирует автоматические регенерации памяти без внешних команд. Этот режим работы обеспечивает SDRAM режим низкого потребления.

– tRAS
Требуемая задержка между запуском команды активации строки банка и запуском команды подзарядки. Величина зависит от производителя и задается в регистре SDRCON.

– tRC
Требуемая задержка между последовательными командами активации одного банка. Данный параметр зависит от производителя и определяется как tRC = tRP + tRAS. Процессор фиксирует значение этого параметра, так что это непрограммируемая опция.

– tRCD. RAS - CAS задержка
Требуемая задержка между командой ACT и началом первой операции записи или чтения. Данный параметр зависит от производителя и определяется как tRCD = CL. Процессор использует фиксированное значение этого параметра, так что это непрограммируемая опция.

– tRP
Требуемая задержка между запуском команды подзарядки и командой активации. Данный параметр зависит от производителя и определяется в SDRCON.

Рисунок 64 показывает интерфейс контроллера SDRAM между внутренним ядром процессора и внешним устройством SDRAM.

Процессор обычно генерирует адрес внешней памяти, который затем активизирует соответствующую линию выбора SDRAM памяти (MSSD3–0), а также указывает на тип операции обмена: чтение или запись. Эта информация анализируется контроллером SDRAM. Внутренняя 32-битная шина адреса мультиплексируется контроллером SDRAM для создания соответствующего сигнала выбора памяти, адреса строки, адреса колонки и банка в SDRAM.

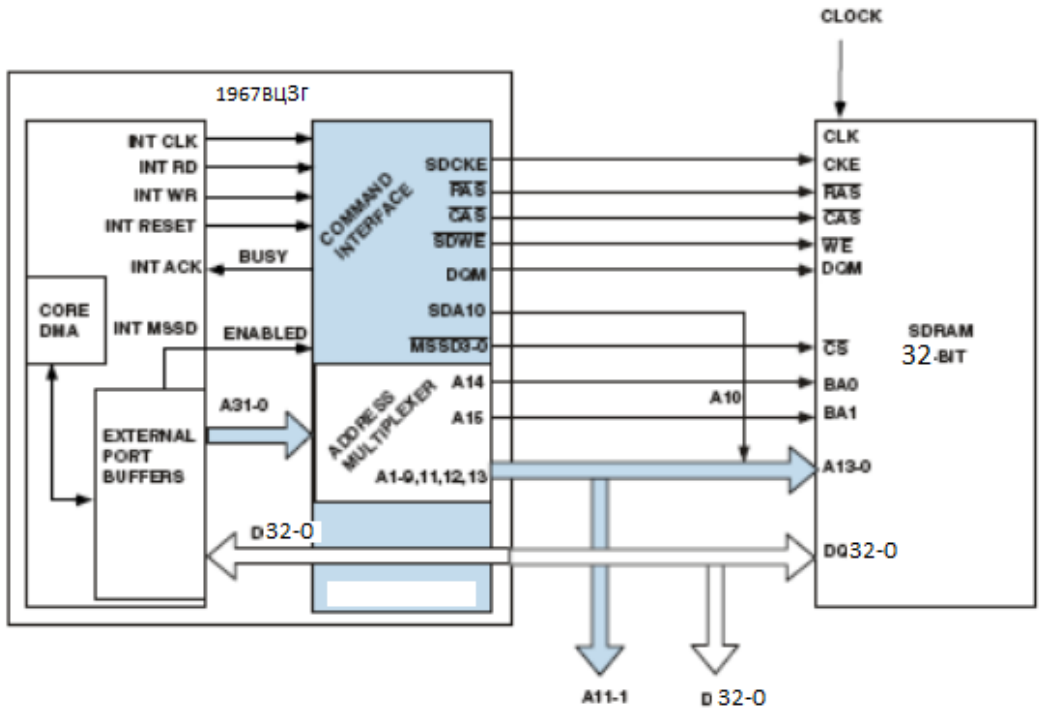


Рисунок 64 – Интерфейс контроллера SDRAM
(для конфигурации с 32-разрядной шиной)

Контакты ввода/вывода интерфейса SDRAM

Интерфейс SDRAM порта внешней шины использует для обмена с SDRAM памятью общую шину данных и часть шины адреса. Для управления микросхемой памяти используются дополнительные линии:

- **MSSD3–0;**
- **RAS;**
- **CAS;**
- **DQM;**
- **SDA10;**
- **SDCKE;**
- **SDWE.**

Назначение линий было описано ранее. Выводы данных и адреса SDRAM могут соединяться напрямую с выводами данных и адреса процессора. Ширина шины данных может быть выбрана 16 или 32 разряда при программировании регистра SYSCON.

Подключения линий адреса:

- разряды 9-0 адреса SDRAM соединяются с выводами ADDR9–0 процессора;
- разряд 10 адреса SDRAM соединяется с выводом SDA10 процессора;
- разряды 15-11 адреса SDRAM соединяются с выводами ADDR15–11 (столько разрядов сколько требуется) процессора.

Выводы выбора банка

Внутренняя организация микросхемы динамической памяти включает в себя два или четыре банка. Для выбора банка микросхема имеет специальные выводы BS1-0. Эти выводы подключаются к адресным линиям процессора. Соединение выводов адреса процессора в качестве линий выбора банка для устройства SDRAM меняется в зависимости от следующих факторов:

- рабочее напряжение, используемое для SDRAM (стандартно 3,3 В или 2,5 В в SDRAM с пониженной мощностью);
- число банков.

Для двух банков BS равен ADDR11 или любой адресной линии в диапазоне ADDR15–11. Для четырех банков BS1–0 равен ADDR12–11 или BS1–0 может быть равен другой паре адресов в диапазоне ADDR15–11.

Для SDRAM с пониженной мощностью использование линий адресов в качестве сигналов выбора банка ограничено ADDR15-14. Для двух банков может быть использована любая из двух адресных линий ADDR14 или ADDR15. Для четырех банков должны использоваться ADDR15–14.

Физическое соединение внутреннего адреса процессора и адреса микросхемы SDRAM

Выше отмечалось, что SDRAM может подключаться к шине данных шириной 16 бит или 32 бита. Выбор подключения зависит от объема памяти, которую пользователь хочет установить в системе, и от желаемой скорости обмена.

В Таблицах 102, 103 и 104 приведено выравнивание данных внешнего порта и соединения SDRAM для 32-разрядной системной шины.

В таблицах приведено соответствие между внешними контактами адреса процессора и контактами адреса микросхемы SDRAM, а также указано какие разряды внутреннего 32-битного адреса выдаются на эти контакты в разных фазах доступа (т.е. в фазе активации банка и выбора активного ряда (строки)), а также в фазе чтения или записи при выборе колонки (номера слова в строке).

При 16-разряной шине чтение (RD) и запись (WR) для передачи данных всегда используются два такта: сначала младшая часть слова, а затем старшая. Основным отличием при 16-разрядной шине данных при выдаче адреса является то, что 32-разрядный внутренний адрес сдвигается влево на один разряд перед выдачей на внешние контакты. Самый младший бит адреса в первом такте обмена всегда равен нулю, а во втором такте равен 1.

Таблица 102 – Размер страницы 256 слов, 32-разрядная шина

Физический вывод процессора Процессор	Внутренний адрес активного цикла банка	Внутренний адрес периода выборки столбца	Физический вывод SDRAM
A0	8	0	A0
A1	9	1	A1
A2	10	2	A2
A3	11	3	A3
A4	12	4	A4
A5	13	5	A5
A6	14	6	A6
A7	15	7	A7
A8	16	8	A8
A9	17	9	A9
A10	несущественный	несущественный	NC
SDA10	18	0	A10/AP
A11	19	19	A11 или банк
A12	20	20	A12 или банк
A13	21	21	A13 или банк
A14	22	22	A14 или банк
A15	23	23	A15 или банк

Таблица 103 – Размер страницы 512 слов, 32-разрядная шина

Физический вывод процессора Процессор	Внутренний адрес активного цикла банка	Внутренний адрес периода выборки столбца	Физический вывод SDRAM
A0	9	0	A0
A1	10	1	A1
A2	11	2	A2
A3	12	3	A3
A4	13	4	A4
A5	14	5	A5
A6	15	6	A6
A7	16	7	A7
A8	17	8	A8
A9	18	9	A9
A10	несущественный	несущественный	NC
SDA10	19	нулевой	A10/AP
A11	20	20	A11 или банк
A12	21	21	A12 или банк
A13	22	22	A13 или банк
A14	23	23	A14 или банк
A15	24	24	A15 или банк

Таблица 104 – Размер страницы 1 K слов, 32-разрядная шина

Физический вывод процессора Процессор	Внутренний адрес активного цикла банка	Внутренний адрес периода выборки столбца	Физический вывод SDRAM
A0	10	0	A0
A1	11	1	A1
A2	12	2	A2
A3	13	3	A3
A4	14	4	A4
A5	15	5	A5
A6	16	6	A6
A7	17	7	A7
A8	18	8	A8
A9	19	9	A9
A10	несущественный	несущественный	NC
SDA10	20	нулевой	A10/AP
A11	21	21	A11 или банк
A12	22	22	A12 или банк
A13	23	23	A13 или банк
A14	24	24	A14 или банк
A15	25	25	A15 или банк

Программирование параметров SDRAM

Микросхемы SDRAM могут быть получены от разных поставщиков и в зависимости от поставщика, микросхемы могут иметь различные требования к последовательности включения и временным параметрам. С целью возможности работы с большим количеством поставщиков, в процессоре предусмотрен регистр SDRCON в котором можно запрограммировать некоторые параметры SDRAM. Описание разрядов регистра приведено ниже (Таблица 105).

Таблица 105 – Регистр SDRCON

Бит	Имя	Назначение
0	SDREN	Включение контроллера SDRAM 1 –включен 0 – выключен
2:1	CAS	Задержка CAS 00 -1 цикл 01 – 2 цикла 10 – 3 цикла 11 – резерв
3	PIPE	Дополнительный конвейер: 1 – используется 0 – нет
5:4	PAGE	Размер страницы: 00 – 256 слов 01 – 512 слов 10 – 1024 слов 11 – резерв
6	-	

Бит	Имя	Назначение
8:7	REF	Период регенерации памяти (в тактах SCLK) 00 – 1100 циклов 01 – 1850 циклов 10 – 2200 циклов 11 – 3700 циклов
10:9	PRC2RAS	Задержка от подзаряда до строба RAS (такты SCLK) 00 – 2 01 – 3 10 – 4 11 – 5
13:11	RAS2PRC	Задержка от строба RAS до команды подзаряда: 000 – 2 **……** 110 – 8
14	INIT	Выбор последовательности инициализации: 1 – команда MRS после регенерации памяти 0 – команда MRS перед регенерацией памяти
15	EMREN	Использование дополнительного регистра режима: 1 – разрешено 0 – не используется
31:16	-	Всегда 0

Регистр SDRCON хранит информацию о конфигурации интерфейса SDRAM. Ниже подробно описывается назначение каждого из параметров регистра.

Включение контроллера SDRAM

Разряд SDREN должен быть установлен, если в системе есть SDRAM. В противном случае, он должен быть сброшен. Любой доступ к SDRAM, пока этот разряд сброшен, вызывает аппаратное прерывание ошибки доступа. Установка данного бита включает в работу контроллер SDRAM внешнего интерфейса. При включении контроллер сразу же начинает процедуру инициализации внешней динамической памяти

Выбор значения задержки CAS Latency

Значение задержки CAS определяет задержку в тактовых циклах системы (SCLK), между временем, когда SDRAM обнаруживает команду чтения и временем, когда он выдает данные на его внешние выводы. Этот параметр позволяет процессору знать, в каком такте после передачи в SDRAM команды чтения он может принять прочитанные данные.

Задержка CAS не используется в циклах записи.

Опция буферизации SDRAM. Глубина конвейера

Ранее были описаны ситуации, при которых пользователю может понадобиться использовать буферизацию адресных и управляющих сигналов. Связано это с максимальной нагрузочной способностью выводов процессора и требуемой скоростью обмена. В случае использование буферизации в конвейере чтения данных появляется дополнительный цикл. Чтобы данный цикл был учтен контроллером SDRAM и используется бит PIPE. Шина данных не буферизируется. Поэтому при установленном разряде PIPE (1), контроллер SDRAM задерживает данные в цикле записи в течение одного такта.

В случае чтения контроллер SDRAM осуществляет захват данных на один цикл позже.

Ниже приведен пример одного процессора, в котором интерфейс SDRAM соединяется с множеством банков SDRAM (Рисунок 65). Микросхема SDRAM имеет шину данных шириной 4 бита. Для обеспечения 32-разрядной шины данных используется восемь микросхем. Чтобы уменьшить нагрузку на внешние выводы процессора используется буферный регистр, который состоит из двух регистров A и B, каждый из которых запоминает одинаковую информацию, но передает ее на свои четыре микросхемы памяти

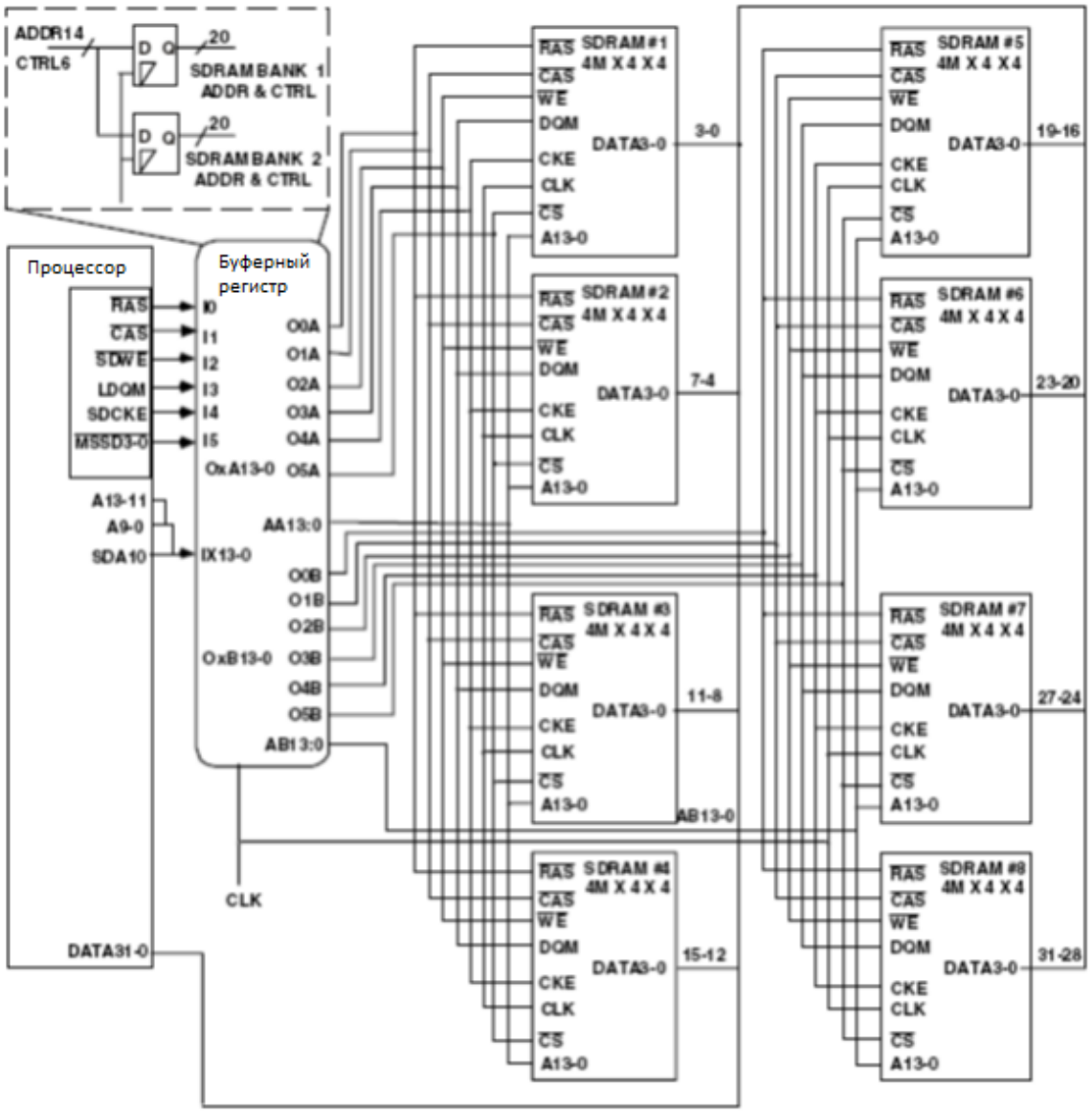


Рисунок 65 – Однопроцессорная система со стандартными устройствами SDRAM (32-разрядная шина)

Выбор размера страницы SDRAM

Размер страницы определяется конкретным типом микросхемы SDRAM. Разряды PAGE определяют размер страницы банков SDRAM, в количестве слов. Знание размера страницы необходимо контроллеру SDRAM для отслеживания ситуаций обращения к одной и той же странице. Это позволяет существенно увеличить скорость обмена. Также размер страницы влияет на распределение разрядов внутреннего адреса между адресами строки, столбца и выбора банка памяти.

Период регенерации памяти

Память SDRAM требует, чтобы в течение определенного интервала времени все ее ячейки были обновлены (регенерированы). Это гарантирует сохранность информации. В связи с этим контроллер SDRAM должен периодически посылать в микросхемы памяти команды регенерации. Период подачи команд программируется. Период внутри контроллера задается **в тактах клока внешней шины** в регистре SD_REF. По сбросу отдельный счетчик обнуляется. Каждый раз при достижении нулевого значения счетчик выставляет запрос контроллеру SDRAM на команду регенерации памяти, загружает значение, записанное в регистре SD_REF и декрементирует свое значение каждый фронт синхросигнала внешней шины.

Задержка от подзаряда до строба RAS

Протокол обмена с динамической памятью требует, чтобы при доступе к новому ряду банка или перед выполнением регенерации памяти, выполнялся подзаряд (закрытие) текущей активной строки. Для подзаряда используется команда **PRE**. После выполнения подзаряда невозможно сразу перейти к активизации следующего ряда, т.к. спецификация микросхемы требует, чтобы между подзарядом и активизацией нового ряда была обеспечена некоторая минимальная задержка. Значение этой задержки, выраженное числом системных тактовых циклов (SCLK), и программируется в поле PRC2RAS.

Выбор задержки между RAS и предварительным зарядом

Аналогично предыдущему параметру, существует требование минимального интервала времени между подачей команды RAS (активизация ряда банка) и подачей команды PRE подзаряда банка. Данный интервал задается числом системных тактовых циклов SCLK. Значение поля RAS2PRC может изменяться от 0 до 6, это соответствует числу циклов SCLK от 2 до 8. Если, например, поле RAS2PRC равно 011 это означает, что минимальный интервал равен пяти циклам. Это не означает, что после команды RAS минимум через пять тактов должна быть команда PRE. Это означает, что если процессор выполнил команду RAS, а затем по каким-то причинам (например, регенерация) ему необходимо выполнить подзаряд, процессор должен проверить прошло ли с момента команды RAS как минимум пять тактов. Тогда он может подавать команду подзаряда.

Выбор последовательности инициализации SDRAM

Для того чтобы микросхема SDRAM корректно работала, она должна быть предварительно проинициализирована. Процессор обеспечивает две наиболее часто используемые последовательности инициализации.

Бит последовательности инициализации INIT в регистре SDRCON выбирает режим включения SDRAM. Когда бит установлен (1), контроллер SDRAM последовательно генерирует: команду PRE, восемь циклов регенерации и команду MRS (установка регистра режима). Когда бит INIT сброшен (0), контроллер SDRAM работает в следующем порядке: команда PRE, команда MRS и восемь циклов регенерации.

Разрешение регистра расширенного режима (EMR)

Бит EMREN в регистре SDRCON разрешает программирование регистра расширенного режима. Когда бит установлен (1), контролер SDRAM формирует цикл

EMRS, предшествующий команде MRS. Когда бит сброшен (0), последовательность инициализации происходит так, как описано в главе выше.

Бит разрешающий EMR должен быть установлен только во время обмена данными с устройствами SDRAM пониженной мощности (2,5 В). Именно в таких микросхемах используется дополнительный регистр режима.

Включение после сброса

После сброса содержимое регистра SDRCON равно нулю и контроллер SDRAM выключен. Как только приложение пользователя выполняет запись в регистр SDRCON данных с установленным битом включения контроллера SDRAM, контроллер тут же инициирует выбранную последовательность включения динамической памяти. Последовательность инициализации определяется разрядом INIT и разрядом регистра расширенного режима в регистре SDRCON. Программный сброс не вызывает сброса контроллера SDRAM и не инициирует повторно последовательность включения.

14.2 Команды контроллера SDRAM

Этот раздел описывает каждую команду, которую контроллер SDRAM использует для управления интерфейсом SDRAM. Ниже представлен обзор различных команд, используемых встроенным контроллером:

- **MRS**
  (установка регистра режима). Инициализирует рабочие параметры SDRAM во время последовательности включения.

- **PRE**
  (подзаряд). Осуществляет подзаряд банка.

- **ACT**
  (активация банка). Активирует страницу в запрашиваемом банке.

- **Read**
  Чтение из SDRAM.

- **Write**
  Запись в SDRAM.

- **REF**
  (регенерация). Переводит SDRAM в режим регенерации.

- **SREF**
  (саморегенерация). Помещает SDRAM в режим саморегенерации, в котором память управляет своими операциями регенерации изнутри.

- **NOP**
  (нет операций). Передается после чтения или записи с целью разрешения пакетных операций или с целью установления режима ожидания для различных доступов SDRAM. Во время этой команды MSSD3–0 деактивирован.

14.2.1 Команда установки регистра режима (MRS)

Установка регистра режима является частью последовательности инициализации SDRAM. Каждая микросхема SDRAM имеет внутри регистр режима (количество регистров режима может достигать четырех) и команда MRS позволяет передать данные, используя разряды адреса ADDR13–0, в микросхему для записи в регистр. Таким образом, MRS инициализирует рабочие параметры SDRAM.

Регистр режима (регистр номер 0) имеет разрядность 13 бит и команда MRS инициализирует следующие параметры:

- Биты 2:0 – Burst length. Значение 0b111, что соответствует полной странице.
- Бит 3 – Burst type. Значение 0, что соответствует последовательному изменению адреса.
- Биты 6:4 – CAS Latency. Значение параметра определяется в соответствии с программированием регистра SDRCON[2:1].

Все оставшиеся биты регистра режима программируются нулевым значением.

При выполнении команды MRS, SDRAM должна находиться в состоянии подзаряда во всех своих банках.

Передача команды осуществляется соответствующей комбинацией значений на линиях управления. Эти значения приведены ниже (Таблица 106). Происходит запись сразу во все микросхемы памяти.

Таблица 106 – Состояние выводов во время выполнения команды MRS

Разряд	Состояние
MSSD3–0	0 (все)
CAS	0
RAS	0
SDWE	0
SDCKE	1
A14 == A30	0
A15 == A31	0 – MRS 1 – EMRS

Если в микросхеме памяти имеется дополнительный регистр режима и для инициализации требуется его запись, в регистре SDRCON должны быть сделаны соответствующие установки (бит EMREN должен быть равен 1). Во время последовательности инициализации контроллер SDRAM запишет нулевое значение в дополнительный регистр режима. Выполняется это с помощью той же команды MRS, но бит 15 адреса (и бит 31) будет установлен в 1, что соответствует дополнительному регистру режима. С помощью адресных линий 14 и 15 можно задать номер дополнительного регистра режима как 01 или 10.

14.2.2 Команда подзаряда (PRE)

Особенность работы микросхемы памяти SDRAM состоит в том, что в некоторых ситуациях необходимо выполнять подзаряд (предварительный заряд) внутренних шин банков памяти. Команда PRE (Таблица 107) исполняется в двух ситуациях:

- Прерывание цикла чтения или записи. Точный момент, когда данная команда генерируется контроллером SDRAM, зависит от

последовательности выполняемых транзакций и адресов, по которым выполняется доступ.

- Предварительная зарядка активного банка. При выполнении команды на линии A10 всегда высокий уровень, что соответствует подзаряду всех банков микросхемы. Однако подзаряжается только активный банк. Алгоритм работы контроллера SDRAM такой, что он поддерживает активным только один банк.

Передача команды PRE возможна в следующих случаях:

- Во время последовательной инициализации SDRAM.
- Перед командой ACT (доступ к новой странице). Доступ к новой странице требует, чтобы активная страница была закрыта. Это делается путем выполнения команды подзаряда.
- Перед циклом регенерации. Выполнение команды регенерации требует, чтобы все банки памяти были подзаряжены.
- Перед тем, как процессор освобождает внешнюю шину. Это гарантирует однозначные условия начала работы нового мастера шины.

Передача команды PRE в микросхему памяти требует, чтобы в течение определенного времени к микросхеме не посылались команды активации. Данное время программируется в регистре SDRCON в соответствии с характеристиками микросхемы памяти.

Таблица 107 – Состояние выводов во время выполнения команды PRE

Вывод	Состояние
MSSD3–0	0 (все)
SDWE	0
RAS	0
CAS	1
SDCKE	1
SDA10	1

14.2.3 Команда выбора активного банка (ACT)

Команда ACT (Таблица 108) посылается контроллером перед любым чтением или записью страницы, которая в данный момент не является активной. Перед командой ACT идет команда PRE, если в каком-то банке имеется другая активная страница. Команда ACT открывает доступ к странице SDRAM в некотором банке, и данная страница остается открытой до тех пор, пока не будет закрыта следующей командой предварительной зарядки PRE.

Таблица 108 – Состояние вывода во время выполнения команды ACT

Вывод	Состояние
MSSD3–0	0 (один из четырех)
CAS	1
RAS	0
SDWE	1
SDCKE	1

14.2.4 Команда чтения (Read)

Команда Read (Таблица 109) выполняет чтение данных из активной страницы памяти SDRAM. Если чтение страницы выполняется первый после активации командой ACT, между командами ACT и Read должна быть задержка, которая определяется параметром tRCD. Если это уже не первое чтение, отслеживание данной задержки не выполняется. Команда Read задает начальный адрес слова в странице, с которого начинается чтение. Микросхема памяти будет использовать переданный адрес как стартовый и каждый такт увеличивать его, если необходимо выполнить последовательное чтение блока памяти. После подачи команды Read данные появляются на внешних выводах микросхемы через время равное задержке CAS Latency.

Одна транзакция чтения занимает различное количество циклов шины в зависимости от размера операнда и ширины внешней шины:

- чтение одинарного слова на 32-разрядной шине – 1 цикл;
- чтение одинарного слова на 16-разрядной шине – 2 цикла;
- чтение двойного слова на 32-разрядной шине – 2 цикла;
- чтение двойного слова на 16-разрядной шине – 4 цикла;
- чтение счетверенного слова на 32-разрядной шине – 4 цикла;
- чтение счетверенного слова на 16-разрядной шине – 8 циклов.

Если после команды Read на SDRAM не поступает какая-то другая команда, SDRAM продолжает последовательное чтение данных, наращивая внутренний адрес. Этот режим чтения называется «страничный режим» или «пакет». Такой режим удобен, когда транзакция чтения длится более одного цикла.

Когда транзакция полностью завершена, SDRAM может продолжить работу по нескольким путям:

- если есть следующая транзакция чтения SDRAM на той же странице, новая транзакция начинается с подачи команды Read;
- если нет новых транзакций SDRAM, контроллер посылает команду BSTOP. Эта команда прерывает последовательное чтение страницы;
- если после чтения приходит транзакция записи SDRAM на той же странице, команда BSTOP также останавливает чтение и запись начинается после приема данных;
- если есть доступ SDRAM к другой странице, поступает команда BSTOP и после нее посылается команда закрытия активной страницы PRE.

Таблица 109 – Состояние вывода во время выполнения команды Чтение

Вывод	Состояние
MSSD3–0	0 (один из четырех)
CAS	0
RAS	1
SDWE	1
SDCKE	1

Ниже приведена временная диаграмма использования команды Read для случая ширины шины 32 бита (Рисунок 66). Задержка CAS Latency равна двум.

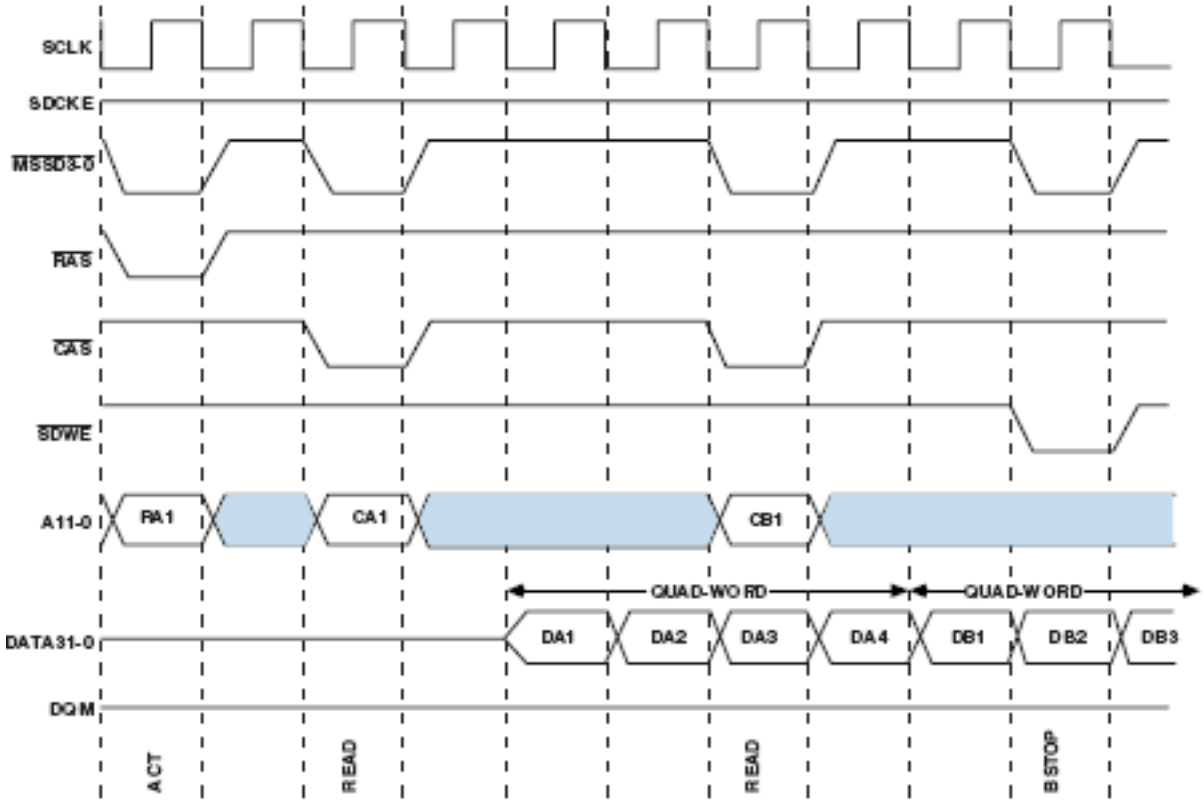


Рисунок 66 – Шина 32 разряда
(пакетное чтение с последующим пакетным чтением на той же странице)

14.2.5 Команда записи (write)

Команда Write (Таблица 110) выполняет запись данных в активную страницу. Если это первая запись после активизации страницы, перед подачей команды записи должен быть соблюден интервал времени tRCD. Команда записи устанавливает начальный адрес слова страницы, с которого начнется запись. Как и в случае чтения, если после записи нет других команд, на следующем такте будет выполняться запись по последовательно увеличенному адресу. Если запись нужно остановить, контроллер SDRAM должен передать команду BSTOP.

Одна транзакция записи занимает разное количество циклов в зависимости от размера операнда транзакции и ширины внешней шины:

- запись одного слова на 32-разрядной шине – 1 цикл;
- запись одного слова на 16-разрядной шине – 2 цикла;
- запись двойного слова на16-разрядной шине – 4 цикла;
- запись двойного слова на 32-разрядной шине – 2 цикла;
- запись счетверенного слова на 16-разрядной шине – 8 циклов;
- запись счетверенного слова на 32-разрядной шине – 4 цикла.

Одновременно с подачей начального адреса, процессор выставляет на шину данных и записываемые данные. Последовательная запись блока данных называется «страничный режим» или «пакет». При этом в последующих циклах нет необходимости подавать адрес, достаточно только передавать следующие данные.

Когда транзакция записи завершена, контроллер SDRAM может продолжить работу по нескольким путям:

- если нет новых транзакций SDRAM, поступает команда BSTOP, которая указывает на завершение записи блока данных;

- если следующая транзакция есть тоже запись в эту же страницу, контроллер может подавать новую команду записи с новым начальным адресом и данными;
- если следующая транзакция – это чтение SDRAM на той же странице, команда BSTOP останавливает транзакцию записи и транзакция чтения начинается с нового цикла;
- если есть доступ SDRAM к другой странице, поступает команда BSTOP и далее подаются команды закрыть текущую страницу (PRE) и активизировать следующую. При этом соблюдаются все необходимые задержки, запрограммированные в регистре конфигурации.

Таблица 110 – Состояние вывода во время выполнения команды Запись

Вывод	Состояние
MSSD3–0	0 (один из четырех)
CAS	0
RAS	1
SDWE	0
SDCKE	1

Ниже приведена временная диаграмма использования команды Write для случая ширины шины 32 бита (Рисунок 67). Задержка CAS Latency равна двум.

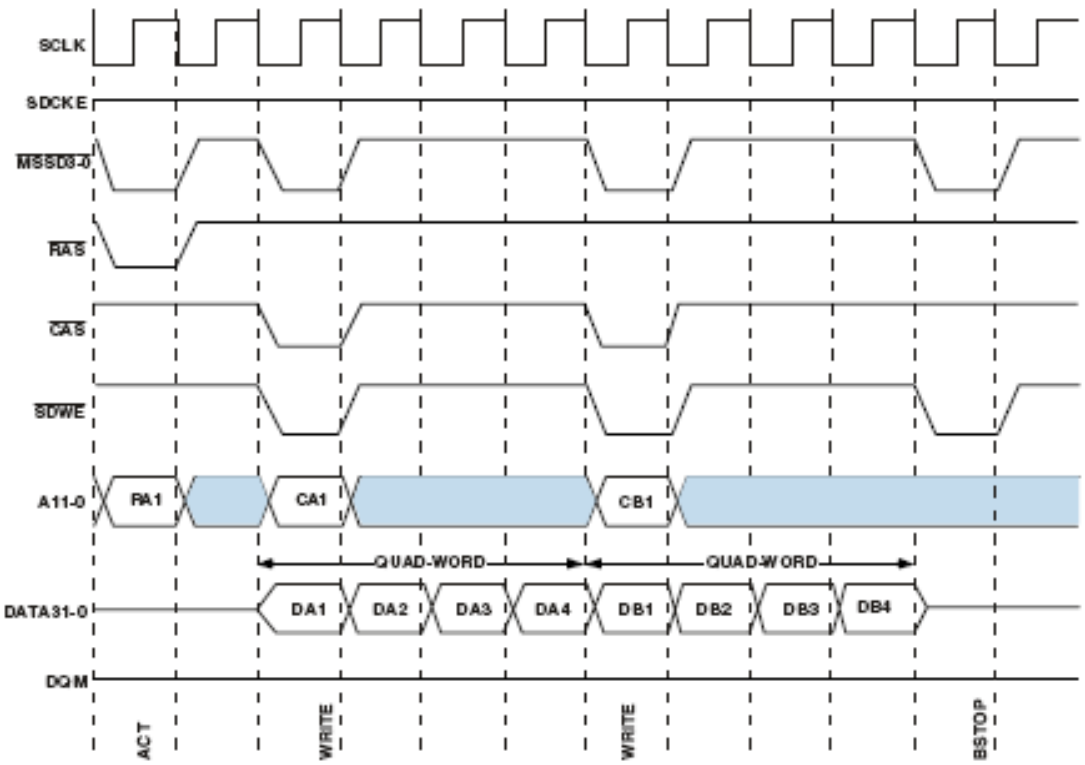


Рисунок 67 – Шина 32 разряда
(пакетная запись с последующей пакетной записью на той же странице)

14.2.6 Команда регенерации (REF)

Память SDRAM требует периодической регенерации своих ячеек (строк) для сохранности информации. Команда REF (Таблица 111) является запросом к SDRAM для выполнения транзакции регенерации. Эта команда генерируется автоматически процессором и вызывает регенерацию данных по адресу, который находится внутри

SDRAM. Перед выполнением команды REF контроллер SDRAM выполняет проверку наличия активных страниц и, если они имеются – выполняет команду предварительной зарядки (PRE) для активного банка.

Следующая команда активации (ACT) выдается контроллером только после минимальной задержки, равной tRC, где:

$$t_{RC} = t_{RP} + t_{RAS} + 1$$

Частота регенерации (период подачи команд регенерации) устанавливается в регистре SDRCON.

Таблица 111 – Состояние выводов во время выполнения команды REF

Вывод	Состояние
MSSD3–0	0(все)
CAS	0
RAS	0
SDWE	1
SDCKE	1

14.2.7 Команда саморегенерации (SREF)

Описанная выше команда регенерации REF требует от процессора непрерывно отслеживать период повторения циклов регенерации и своевременно посылать команды REF. Эта команда может выполняться на фоне выполнения транзакций процессором к другим типам памяти (не SDRAM). Также существует режим, когда память отвечает за процедуру регенерации и самостоятельно поддерживает данные достоверными. Этот режим называется режимом саморегенерации и процессор переводит SDRAM в этот режим путем подачи команды SREF (Таблица 112). В данном режиме память недоступна для использования.

При выполнении команды саморегенерации должны быть выполнены все те же временные требования, что и при выполнении команды регенерации.

Перед выполнением команды SREF контроллер подзаряжает все банки микросхем. После команды SREF память выполняет операции регенерации самостоятельно без внешнего контроля. После выхода из режима саморегенерации контроллер SDRAM ждет количество циклов tRC перед выполнением команды активизации банка (ACT).

Память переводится в режим саморегенерации установкой 0-го бита регистра SDRSRF в единичное значение. Данный режим может быть использован как составная часть режима пониженного энергопотребления системы. Для выхода из режима саморегенерации необходимо очистить нулевой бит регистра SDRSRF.

Таблица 112 – Состояние вывода во время выполнения команды SREF

Вывод	Состояние
MSSD3–0	0 (все)
CAS	0
RAS	0
SDWE	1
SDCKE	0

Режим саморегенерации поддерживается тем, что на линии SDCKE постоянно низкий уровень, что запрещает использование входа синхронизации.

14.3 Вспомогательные регистры интерфейса внешней шины

В процессоре 1967ВН044 функции данных регистров ограничены.

14.3.1 Регистр управления блокировкой шины (BUSLOCK)

Регистр BUSLOCK определяет состояние запроса блокировки шины. В этом 32-разрядном регистре определен только один нулевой бит. Все другие биты (с 1 по 31) не используются. Запись значения 1 в нулевой бит вызывает запрос на блокировку шины со стороны процессора. В процессоре 1967ВН044 внешний интерфейс всегда является мастером шины, и нет необходимости в регистре блокировки.

14.3.2 Регистр статуса системы (SYSTAT)

Регистр SYSTAT предназначен только для чтения и указывает на состояние некоторых параметров системы.

Таблица 113 – Регистр SYSTAT

Бит	Имя	Назначение
9:0	-	Всегда 0
10		Всегда 1
11	-	Всегда 0
12	-	Бит 0 регистра CFG1
13	MRSCOMP	Признак окончания процедуры инициализации динамической памяти: 1 – завершена. Память готова к использованию 0 – нет
14	BUSLKACT	Признак захвата шины: 1 – текущий процессор стал мастером на шине и удерживает ее 0 – нет захвата шины
15	-	Всегда 0
16	BROADREADERR	Ошибка доступа к адресному пространству 0x0C….: 1 – была попытка чтения из указанного пространства 0 – нет ошибки
17	AUTODMAERR	Признак ошибки в работе каналов AutoDMA: 1 – была попытка записи в выключенный канал 0 – нет ошибки
18	SDRAMERR	Признак ошибки доступа к SDRAM: 1 – была попытка доступа неинициализированной внешней динамической памяти. 0 – нет ошибки
19	MPREADERR	Признак ошибки чтения процессором запрещенного адресного пространства: 1 – ошибка. 0 – нет ошибки
31:20	-	Всегда 0

Программы могут считывать регистр SYSTAT, используя имена SYSTAT или SYSTATCL (сброс SYSTAT):

- SYSTAT – чтение без изменений в содержимом регистра;
- SYSTATCL – разряды 19–16 будут сброшены после чтения.

15 Последовательный хост-интерфейс

Процессор содержит синхронный последовательный интерфейс, который может быть использован для загрузки информации в процессор внешним ведущим устройством, а также для анализа состояния процессора посредством чтения его ресурсов.

Интерфейс подключен к периферийной шине и имеет базовый адрес 0x8000_005C.

Внешние выводы хост-интерфейса приведены в таблице 114.

Таблица 114 – Внешние выводы

Обозначение вывода (основное)	Обозначение вывода для Хост-интерфейса	Тип вывода	Функциональное назначение
PC[4]	HCLK	I/O	Хост интерфейс. Синхросигнал
PC[5]	HDI	I/O	Хост интерфейс. Вход данных
PC[6]	HDO	I/O	Хост интерфейс. Выход данных

15.1 Регистры интерфейса

Краткое описание регистров интерфейса приведено в Таблица 115.

Таблица 115 – Регистры хост-интерфейса

Номер\ смещение	Обозначение регистра	R/W	Описание	Состояние по сбросу
0x00	-	R	Неопределенное значение	0
0x01	-	R	Неопределенное значение	0
0x02	MCR	R	Управление запросом прерываний к процессору	0
0x03	-	R	Неопределенное значение	0

Единственный доступный для процессора регистр хост-интерфейса MCR можно прочитать только посредством чтения квадрослова по адресу 0x8000_005C. Регистр может использоваться как один из возможных способов передачи информации от хоста к процессору.

15.2 Аппаратная реализация интерфейса

Для организации обмена синхронный интерфейс использует три линии: линию синхронизации HCLK и линии данных HDI, HDO. Эти линии функционально совмещены с линиями задания конфигурации BOOT[2:0].

Во время сброса системы линии BOOT выполняют функцию задания варианта начального старта системы, а после завершения сброса они могут быть использованы для управления интерфейсом. Для реализации интерфейса необходимо использовать шесть линий:

- питание;
- общий;
- сброс nRST;
- BOOT.0/HCLK;
- BOOT.1/HDI;
- BOOT.2/HDO.

Начальная последовательность действий ведущего при этом будет следующей:

- установить на входе nRST низкий уровень (сброс системы) и задать на входах BOOT[2:0] необходимую начальную конфигурацию;
- после некоторой задержки, установить на nRST высокий уровень, сохраняя на входах BOOT[2:0] старое значение (обеспечивая время удержания);
- передать контроль над выводами BOOT[2:0] ведущему устройству.

15.2.1 Протокол обмена по последовательному интерфейсу

- Информация принимается каналом по положительному фронту (переход из 0 в 1) синхросигнала HCLK. Выдача информации осуществляется также по положительному фронту.
- Обмен всегда осуществляется 32-разрядными словами. Одна посылка состоит, как минимум из 34 бит: старт бит (0), 32 бита данных (старший бит первый) и стоп-бит (значение 1). Примеры обмена данными приведены на рисунках 68 и 69.
- После сброса процессора хост-интерфейс некоторое время недоступен. Ведущее устройство после сброса процессора должно посылать сигнал синхронизации и значение 1 на входе данных HDI. Как только интерфейс включается (на него перестает действовать внутренний сброс процессора), на выходной линии данных появится активное значение 0 и через несколько тактов синхронизации – значение 1. После этого интерфейс готов к работе.
- Посредством интерфейса возможно выполнение операций чтения и записи, а также внутренних операций.

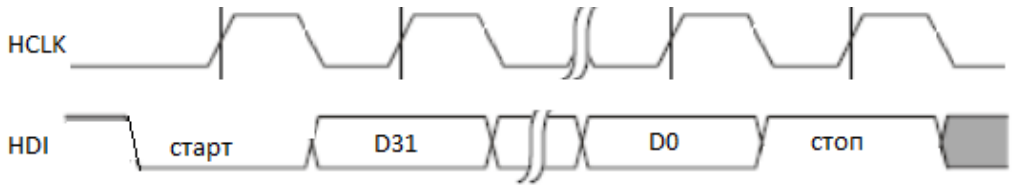


Рисунок 68 – Прием информации контроллером интерфейса

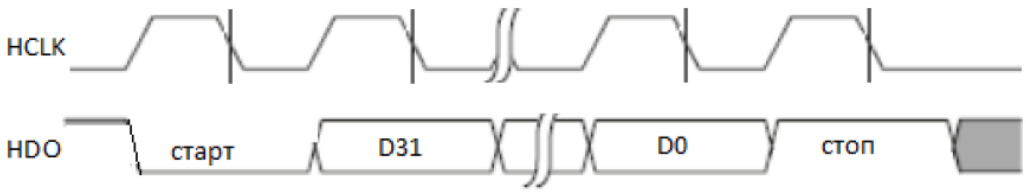


Рисунок 69 – Передача информации контроллером интерфейса

15.2.2 Стандартный формат обмена

Имеется два формата команд: стандартный и короткий. В стандартном формате начальный адрес обмена передается отдельным словом, а в коротком он упакован в управляющее слово. Для выполнения операции записи в стандартном формате выполняется следующая процедура обмена:

1 передается управляющее слово;
2 передается начальный адрес;
3 передаются данные.

Структура управляющего слова приведена ниже (Таблица 116).

Таблица 116 – Структура управляющего слова стандартного формата

Номер	Название	Описание
31:28	TYPE	Задают тип операции 00x0 – чтение 01x0 – запись 10xx – внутренняя операция 11xx – резерв
27:20	DDP	Поле управления обменом
19:17	-	
16	MDF	1 – разрешение модификации адреса после каждого обмена
15:0	CNT	Количество 32-разрядных слов данных при обмене

Для выполнения операции чтения стандартного формата выполняется следующая процедура обмена:

1 передается управляющее слово;
2 передается начальный адрес;
3 принимаются данные.

При выполнении внутренней операции имеется возможность загрузить разряды 29:0 управляющего слова в специальный регистр управления интерфейсом MCR. В настоящее время используется только бит 0 регистра управления. При записи в него значения 1 происходит генерация запроса прерывания к процессору, т.е. имеется возможность послать запрос на прерывания, не выполняя процедуры обмена данными.

15.2.3 Короткий формат обмена

Для выполнения операции записи в коротком формате выполняется следующая процедура обмена:

1 передается управляющее слово;
2 передается слово данных.

Структура управляющего слова приведена в Таблица 117.

Таблица 117 – Структура управляющего слова короткого формата

Номер	Название	Описание
31:28	TYPE	Задают тип операции: 00x1 – чтение 01x1 – запись 10xx – внутренняя операция 11xx – резерв
27:20	DDP	Поле управления обменом
19	A[31]	Бит 31 адреса. Используется для доступа к регистрам периферии
18:16	A[20:18]	Биты адреса 20:18 при обращении к внутренней памяти. Все другие биты равны нулю
15:0	A[15:0]	Младшие биты адреса

При использовании короткого формата возможна запись или чтение только одного слова (в соответствии с размером) данных. Адрес не изменяется. Имеется возможность доступа ко всем ресурсам процессора.

Для выполнения операции чтения в коротком формате выполняется следующая процедура обмена:

1 передается управляющее слово;
2 принимаются данные.

При использовании короткого формата нет прямого указания на количество передаваемых слов данных. Количество передаваемых 32-разрядных слов определяется значением поля LEN (см. таблицу 118). Для двойного слова это 2 слова, а для квадрослова это 4 слова.

15.2.4 Поле управления обменом DDP

Биты TYPE управляющей посылки задают только общую информацию о типе обмена. Если выполняется чтение или запись данных, то дополнительная информация о параметрах обмена кодируется в поле DDP.

Таблица 118 – Структура поля управления обменом DDP

Номер	Название	Описание
7:5	TY	Выбор источника либо приемника: 000 – нет операции (выключено) 001 – запрещено 010 – внутренняя память 011 – регистры процессорного ядра 100 – внешняя память и регистры периферии 101 – запрещено 110 – внешний EPROM 111 – запрещено
4	PR	Приоритет обмена: 1 – высокий 0 – обычный
3	-	Резерв. Всегда должен быть равен 0
2:1	LEN	Длина передаваемых данных в одном цикле обмена: 00 – резерв 01 – слово 32 бита 10 – длинное слово 64 бита 11 – квадрослово 128 бит
0	INT	Генерация запроса прерывания после окончания работы канала: 1 – разрешено 0 – запрещено

Если формат обмена стандартный, то принимается следующее слово, которое рассматривается как начальный адрес источника либо приемника. Если формат обмена короткий, то начальный адрес формируется как {A[31], 10'b0, A[20:18], 2'b00, A[15:0]}. Отметим, что биты TY поля DDP имеют более высокий приоритет при декодировании адреса. Если, например, адрес имеет значение 0xA5A5, а биты TY равны 100, то будет выполнено обращение к внешней памяти по адресу 0xA5A5.

Бит MDF стандартной посылки используется для указания модификации адреса после выполнения одного цикла обмена. Если этот бит равен нулю, то данные все время будут читаться или записываться по одному и тому же адресу. Если бит MDF равен 1, то после чтения или записи одной порции данных (размер порции определяется битами LEN), будет выполнена модификация адреса (увеличение адреса на значение 1, 2 или 4 в соответствии со значением поля LEN).

Количество 32-разрядных слов обмена определяется значением поля CNT стандартной посылки, либо значением поле LEN короткой посылки. После каждой передачи порции данных, значение внутреннего счетчика слов уменьшается от

начального (значение CNT) на величину 1, 2 или 4 в соответствии со значением поля LEN.

Бит PR задает приоритет интерфейса при арбитраже в сравнении с каналами контроллера ПДП и процессором. Желательно использовать значение PR равное 1, т.к. в случае низкого приоритета хост интерфейс может быть приостановлен работой более высокоприоритетных каналов контроллера ПДП. Особенно это важно во время записи блока данных во внешнюю память.

Бит запроса прерывания INT позволяет сгенерировать запрос прерывания к процессору после окончания обмена данными. Хост-интерфейс имеет соответствующий бит запроса прерываний в контроллере прерываний (бит 6 регистра ILATH, IMASKH), а также регистр вектора прерываний IVHOST.

Обращаем внимание на разрешенные комбинации в разрядах TY. Использование запрещенных комбинаций приведет к непредсказуемому поведению интерфейса.

Также, как и при программировании каналов контроллера ПДП, необходимо соблюдать правила выравнивания адресов при работе с длинными словами и квадрословами. Значение количества данных CNT должно быть кратно 2-м при работе с длинными словами и кратно 4-м при работе с квадрословами.

15.2.5 Внутренняя операция интерфейса

При выполнении внутренней операции содержимое принятого управляющего слова (биты с 0-го по 29-й) записываются в регистр MCR. Нулевой бит регистра MCR может формировать запрос прерывания к процессору. Данное прерывание анализируется в контроллере прерываний как запрос номер 6 регистра ILATH. Процессор может прочитать значение регистра MCR. Чтение выполняется посредством чтения квадрослова по адресу 0x8000_005C и последующем использовании второго слова прочитанного квадрослова. Отметим, что запрос прерывания от хост-интерфейса к процессору можно также сформировать и при выполнении операций чтения либо записи. Особенно это удобно делать при выполнении операции записи. В этом случае процессору может быть передана информация и после этого сформирован запрос прерывания. Такой способ позволяет передать процессору больше информации, чем через регистр MCR.

15.2.6 Вход синхронизации HCLK

Вход синхронизации используется для задания тактового импульса обмена по последовательному интерфейсу. Прием и выдача информации осуществляется по положительному фронту HCLK. Отметим, что вход HCLK во время сброса выполняет функцию задания варианта начального старта (BOOT[0]). Также этот вывод является входом-выходом порта общего назначения PC[4]. Во время сброса требуемый уровень на входе HCLK должен задаваться резистором (либо внешним ведущим устройством). После сброса вход HCLK может управляться ведущим устройством (при его наличии), либо иметь постоянное значение. Дело в том, что любое переключение на выводе PC[4] рассматривается интерфейсом как подача частоты HCLK. Поэтому данный вывод не может произвольно изменяться. Если пользователь процессора не предусматривает подключения внешнего ведущего устройства к хост-интерфейсу, т.е. хост-интерфейс не планируется использовать, то желательно заблокировать вход HCLK посредством установки бита H_OFF (бит 13) в регистре CFG1 (см. раздел 29 «Модуль управления синхронизацией и энергопотреблением»). После этого вывод HCLK(PC[4]) может использоваться без ограничений для реализации других функций.

15.2.7 Вход данных HDI

Вход используется для передачи процессору управляющих команд и данных. Во время сброса вход выполняет функцию задания варианта начального старта(BOOT[1]). Также этот вывод является входом-выходом порта общего назначения PC (бит 5). Во время сброса требуемый уровень на входе HDI должен задаваться резистором (либо внешним ведущим устройством). После сброса вход HDI может управляться ведущим устройством (при его наличии), либо выполнять функции порта общего назначения PC[5]. При обмене с интерфейсом ведущее устройство всегда посылает 32-разрядные данные с дополнительными старт и стоп битами. Если нет необходимости передавать информацию по входу HDI, на входе должен удерживаться высокий уровень. При наличии частоты синхронизации HCLK высокий уровень гарантирует отсутствие входных данных. Интерфейс обнаруживает наличие входных данных при принятии старт бита. После этого внутренний счетчик отсчитывает 32 такта данных. Стоп-бит нужен как дополнительный такт синхронизации для выполнения внутренних операций. Значение стоп-бита не контролируется. Однако для приема следующего слова, во входном потоке должна появиться 1 и только после неё будет детектирован старт бит.

15.2.8 Выход данных HDO

Выход данных HDO используется для передачи данных от процессора к ведущему устройству. В момент сброса процессора он выполняет функцию задания варианта начального старта (BOOT[2]), поэтому во время сброса данный вывод находится в отключенном состоянии (т.е. настроен на прием информации с внешнего контакта). Если на входе HCLK имеются переключения, то они рассматриваются как наличие частоты синхронизации от ведущего устройства. В этом случае данные с входа HDI записываются в приемный сдвиговый регистр. Если после завершения внутреннего сигнала сброса будет принято не менее 32-х единичных значений с входа HDI, то интерфейс включится, и выход HDO станет активным выходом. Для задания варианта начального старта на выводе HDO может использоваться резистор, подключенный к шине общий. Таким образом, на выводе HDO будет удерживаться уровень нуля до тех пор, пока интерфейс не включится и не начнет выдавать активную единицу. При чтении внутренней ячейки памяти процессора, внешнее хост-устройство заранее не знает сколько тактов понадобится для считывания значения. Поэтому ведущее устройство после передачи команды чтения должно непрерывно подавать сигналы синхронизации HCLK и удерживать на входе HDI высокий уровень. Как только данные поступят в буфер интерфейса, на выводе HDO появится значение 0 (старт бит), и за ним будут переданы 32 бита данных. После них будет выдан стоп бит. Выдача данных на вывод HDO осуществляется автоматически, как только в буфере появляются данные.

15.2.9 Алгоритм работы внутренней машины состояний

Внутренний автомат интерфейса работает с тактовой частотой SOC-шины (SOCCLK) и постоянно анализирует поступающую информацию. Обработка информации осуществляется в соответствии со следующим алгоритмом:

Состояние S0 Ожидание управляющего слова. Если слово принято – запись слова во внутренний регистр HCR и переход в состояние S1.

Состояние S1 Анализ регистра HCR. Если стандартный формат чтения – переход в состояние S4. Короткий формат чтения – переход в состояние S7. Если стандартный формат записи – переход в состояние S8. Короткий формат записи – переход в состояние S10. Если внутренняя операция – переход в состояние S2. В случае резервной операции – зависание машины состояний (!).

Состояние S2 Выполнение внутренней операции. Перезапись регистра HCR в регистр MCR. Переход в исходное состояние S0.

Состояние S4 Ожидание адреса для выполнения чтения. После приема адреса – запись управляющей информации (начальный адрес, счетчик слов, управляющее слово) в канала обмена. Переход в состояние S6 - ожидания выполнения операции чтения. Если в момент записи в канал обмена последний находится в активном состоянии, устанавливается флаг ошибки и канал выключается.

Состояние S6 Ожидание завершения работы канала. Канал завершит работу только после чтения всего блока данных. При этом моментом окончания работы канала является факт загрузки последнего слова в буфер и начало его передачи в ведущее устройство. После передачи последнего слова данных - переход в исходное состояние S0.

Состояние S7 Запись управляющей информации (начальный адрес, счетчик слов, управляющее слово) в канал обмена. Переход в состояние S6 - ожидания выполнения операции чтения. Если в момент записи в канал обмена последний находится в активном состоянии, устанавливается флаг ошибки и канал выключается.

Состояние S8 Ожидание адреса для выполнения операции записи. После приема адреса – запись управляющей информации (начальный адрес, счетчик слов, управляющее слово) в канал обмена. Переход в состояние S9 - ожидания приема данных для записи. Если в момент записи в канал обмена последний находится в активном состоянии, устанавливается флаг ошибки и канал выключается.

Состояние S9 Ожидание данных для выполнения операции записи. После приема данных – запись принятого слова в буфер канала. Ожидание нового слова данных либо завершения работы канала. Канал инициирует запись данных в процессор только если в буфере данных достаточно данных для запрограммированной записи. После записи последнего слова данных - переход в исходное состояние S0.

Состояние S10 Запись управляющей информации (начальный адрес, счетчик слов, управляющее слово) в канал обмена. Переход в состояние S9 - ожидания данных для выполнения операции записи. Если в момент записи в канал обмена последний находится в активном состоянии, устанавливается флаг ошибки и канал выключается.

Машина состояний интерфейса требует корректного завершения всех операций. У ведущего процессора нет возможности анализировать состояние интерфейса, и в случае различных аномалий возвращать его функционирование в начальное состояние. Только сброс всего процессора является выходом из возможной ошибочной ситуации.

15.2.10 Доступ к регистрам ядра

Выше было отмечено, что в случае, когда биты TY поля DDP равны 011’b, хост устройство может осуществлять доступ к внутренним регистрам вычислительного ядра процессора. Отметим, что это единственный способ получить возможность доступа к регистрам для внешнего устройства. Для доступа к регистрам используются только разряды адреса с 0-го по 10-й. Биты с 5-го по 10-й образуют номер группы регистров, а биты с 0-го по 4-й номер регистра в группе. Номера групп при доступе со стороны хост-устройства совпадают с номерами групп, которые используются системой команд процессора для выполнения пересылок. Используемые номера групп приведены в таблице 119.

Таблица 119 – Группы регистров вычислительного ядра

Номер группы	Название группы
0	Регистры общего назначения вычислительного модуля X: XR[0-31]
2	Регистры общего назначения вычислительного модуля Y: YR[0-31]
10	Регистры модуля отладки
12	Регистры общего назначения J модуля IALU : J[0-31]
13	Регистры общего назначения K модуля IALU : K[0-31]
14	Дополнительные регистры J модуля IALU : JB[0-3], JL[0-3]
15	Дополнительные регистры K модуля IALU : KB[0-3], KL[0-3]
26	Регистры устройства управления
27	Регистры модуля отладки
30	Регистры модуля защиты и управления кэш-памятью
31	Регистры модуля защиты и управления кэш-памятью

15.2.11 Ограничения интерфейса

При режиме старта BOOT[2:0]==000 процессор переходит в режим ожидания прерывания. Хост-устройство может выполнить загрузку данных в память процессора и запустить процессор на выполнение программы. Последовательный хост-интерфейс позволяет ведущему устройству осуществлять полный контроль процессора. Передача данных по последовательному интерфейсу выполняется с частотой HCLK, а все внутренние операции процессора в контроллере интерфейса с частотой SOC-шины. Чтобы не было проблем с переполнением приемного буфера интерфейса желательно иметь частоту HCLK не выше частоты SOC-шины. Особое внимание необходимо уделить начальным операциям, выполняемым после сброса, т.к. в этом случае процессор работает на входной частоте синхронизации, и она может быть недостаточно высокой. В этом случае можно анализировать параметры частот процессора и менять частоту обмена HCLK. Особое внимание нужно уделять операциям записи блока данных в процессор, т.к. в этом случае внутренние операции должны успевать выполниться до приема новой порции данных.

Операции чтения и записи интерфейса независимы. Интерфейс всегда начинает выдачу данных на выход HDO, если в его буфер поступают данные. Таким образом, выдачу данных можно вести одновременно с приемом новой информации. Единственное ограничение - это совместное использование буфера интерфейса, как для операций чтения, так и для операций записи. Поэтому операция записи может начинаться только после завершения чтения последнего слова данных.

Для корректной работы интерфейса рекомендуется не начинать новой операции до момента полного завершения предыдущей.

Операции чтения области регистров внутренних периферийных устройств запрещены. При выполнении таких операций интерфейс будет заблокирован. См. описание ошибки 0009 в документе “1967ВН044 Errata Notice”.

16 Порты связи

Процессор имеет два LINK-порта связи для обеспечения 8\4\1-битного приема и 8\4\1-битной передачи в мультипроцессорных системах. Оба LINK-порта могут быть сконфигурированы как один 16-ти битный порт для одновременного приема и передачи данных (полный дуплекс).

Порты связи имеют следующие характеристики:

- тактовая скорость связи равна 1/2 от выходной частоты собственной PLL;
- данные порта связи упакованы в 128-битные слова для DMA передачи во внутреннюю и внешнюю память;
- каждый порт связи имеет собственные буферные регистры;
- передачи каждого порта контролируются подтверждающим протоколом;
- порты связи поддерживают полный дуплекс и передачу в\из внешних портов или других каналов связи.

Порты связи предназначены для использования при двухточечной связи между процессорами в системе, но могут использоваться в качестве интерфейса для коммуникации с любыми другими устройствами, разработанными для работы по тому же протоколу.

Порты связи процессора используют схему LVDS (дифференциальные сигналы низкого напряжения). Данные выдаются и принимаются как на фронте, так и на срезе тактового сигнала. Каждый из портов имеет канал приема и передачи, и способен работать в полностью дуплексном режиме.

Управление передачей ядром процессора выполняется посредством механизмов прерывания и опроса. Управление передачей контроллером DMA выполняется через назначенные DMA-каналы приема/передачи. При этом все каналы DMA портов поддерживают цепочки операций, а также могут быть использованы при загрузке процессора во время старта.

Внешние выводы портов связи приведены в Таблице 115

Таблица 120 – Внешние выводы

Обозначение вывода (основное)	Обозначение вывода для портов связи	Тип вывода	Функциональное назначение
PC[24]	L0ACKO	I/O	LINK порт 0. Выход разрешения передачи
PC[25]	L0ACKI	I/O	LINK порт 0. Вход разрешения передачи
PC[26]	L0BCMPO	I/O	LINK порт 0. Выход окончания блока
PC[27]	L0BCMPI	I/O	LINK порт 0. Вход разрешения передачи
PC[28]	L1ACKO	I/O	LINK порт 1. Выход разрешения передачи
PC[29]	L1ACKI	I/O	LINK порт 1. Вход разрешения передачи
PC[30]	L1BCMPO	I/O	LINK порт 1. Выход окончания блока
PC[31]	L1BCMPI	I/O	LINK порт 1. Вход разрешения передачи
L0DATOP[0]	-	-	LINK порт 0. Выход данных
L0DATON[0]	-	-	LINK порт 0. Выход данных

L0DATOP[1]	-	-	LINK порт 0. Выход данных
L0DATON[1]	-	-	LINK порт 0. Выход данных
L0DATOP[2]	-	-	LINK порт 0. Выход данных
L0DATON[2]	-	-	LINK порт 0. Выход данных
L0DATOP[3]	-	-	LINK порт 0. Выход данных
L0DATON[3]	-	-	LINK порт 0. Выход данных
L0DATOP[4]	-	-	LINK порт 0. Выход данных
L0DATON[4]	-	-	LINK порт 0. Выход данных
L0DATOP[5]	-	-	LINK порт 0. Выход данных
L0DATON[5]	-	-	LINK порт 0. Выход данных
L0DATOP[6]	-	-	LINK порт 0. Выход данных
L0DATON[6]	-	-	LINK порт 0. Выход данных
L0DATOP[7]	-	-	LINK порт 0. Выход данных
L0DATON[7]	-	-	LINK порт 0. Выход данных
L0CLKOP	-	-	LINK порт 0. Выход синхронизации
L0CLKON	-	-	LINK порт 0. Выход синхронизации
L0DATIP[0]	-	-	LINK порт 0. Вход данных
L0DATIN[0]	-	-	LINK порт 0. Вход данных
L0DATIP[1]	-	-	LINK порт 0. Вход данных
L0DATIN[1]	-	-	LINK порт 0. Вход данных
L0DATIP[2]	-	-	LINK порт 0. Вход данных
L0DATIN[2]	-	-	LINK порт 0. Вход данных
L0DATIP[3]	-	-	LINK порт 0. Вход данных
L0DATIN[3]	-	-	LINK порт 0. Вход данных
L0DATIP[4]	-	-	LINK порт 0. Вход данных
L0DATIN[4]	-	-	LINK порт 0. Вход данных
L0DATIP[5]	-	-	LINK порт 0. Вход данных
L0DATIN[5]	-	-	LINK порт 0. Вход данных
L0DATIP[6]	-	-	LINK порт 0. Вход данных
L0DATIN[6]	-	-	LINK порт 0. Вход данных
L0DATIP[7]	-	-	LINK порт 0. Вход данных
L0DATIN[7]	-	-	LINK порт 0. Вход данных
L0CLKIP	-	-	LINK порта 0. Вход синхросигнала(+)
L0CLKIN	-	-	LINK порта 0. Вход синхросигнала(-)
L1DATOP[0]	-	-	LINK порт1. Выход данных
L1DATON[0]	-	-	LINK порт1. Выход данных
L1DATOP[1]	-	-	LINK порт1. Выход данных
L1DATON[1]	-	-	LINK порт1. Выход данных
L1DATOP[2]	-	-	LINK порт1. Выход данных
L1DATON[2]	-	-	LINK порт1. Выход данных
L1DATOP[3]	-	-	LINK порт1. Выход данных
L1DATON[3]	-	-	LINK порт1. Выход данных
L1DATOP[4]	-	-	LINK порт1. Выход данных
L1DATON[4]	-	-	LINK порт1. Выход данных
L1DATOP[5]	-	-	LINK порт1. Выход данных
L1DATON[5]	-	-	LINK порт1. Выход данных
L1DATOP[6]	-	-	LINK порт1. Выход данных
L1DATON[6]	-	-	LINK порт1. Выход данных
L1DATOP[7]	-	-	LINK порт1. Выход данных
L1DATON[7]	-	-	LINK порт1. Выход данных
L1CLKOP	-	-	LINK порт 1. Выход синхросигнала(+)
L1CLKON	-	-	LINK порт 1. Выход синхросигнала(-)
L1DATIP[0]	-	-	LINK порт 1. Вход данных
L1DATIN[0]	-	-	LINK порт 1. Вход данных
L1DATIP[1]	-	-	LINK порт 1. Вход данных
L1DATIN[1]	-	-	LINK порт 1. Вход данных

L1DATIP[2]	-	-	LINK порт 1. Вход данных
L1DATIN[2]	-	-	LINK порт 1. Вход данных
L1DATIP[3]	-	-	LINK порт 1. Вход данных
L1DATIN[3]	-	-	LINK порт 1. Вход данных
L1DATIP[4]	-	-	LINK порт 1. Вход данных
L1DATIN[4]	-	-	LINK порт 1. Вход данных
L1DATIP[5]	-	-	LINK порт 1. Вход данных
L1DATIN[5]	-	-	LINK порт 1. Вход данных
L1DATIP[6]	-	-	LINK порт 1. Вход данных
L1DATIN[6]	-	-	LINK порт 1. Вход данных
L1DATIP[7]	-	-	LINK порт 1. Вход данных
L1DATIN[7]	-	-	LINK порт 1. Вход данных
L1CLKIP	-	-	LINK порт 1. Вход синхросигнала(+)
L1CLKIN	-	-	LINK порт 1. Вход синхросигнала(-)

16.1 Архитектура портов связи

Порты связи подключены к SOC-шине как периферийное устройство (Рисунок 70).

Порт связи состоит из двух частей – передатчик и приемник. Каналы передачи и приема имеют буфер, как показано ниже (Рисунок 71). Буферные регистры, подключенные к SOC-шине, доступны программисту как регистры LBUFTX и LBUFRX. Они являются 128-разрядными, отображаемыми в памяти универсальными регистрами. Регистры дополнительных буферов и сдвиговые регистры являются программно недоступными.

Дополнительная буферизация позволяет конвейеризировать операции приема-передачи порта связи и операции контроллера DMA. Также имеет значение, что сдвиговые регистры приема-передачи имеют собственную частоту работы, которая отличается от частоты SOC-шины.

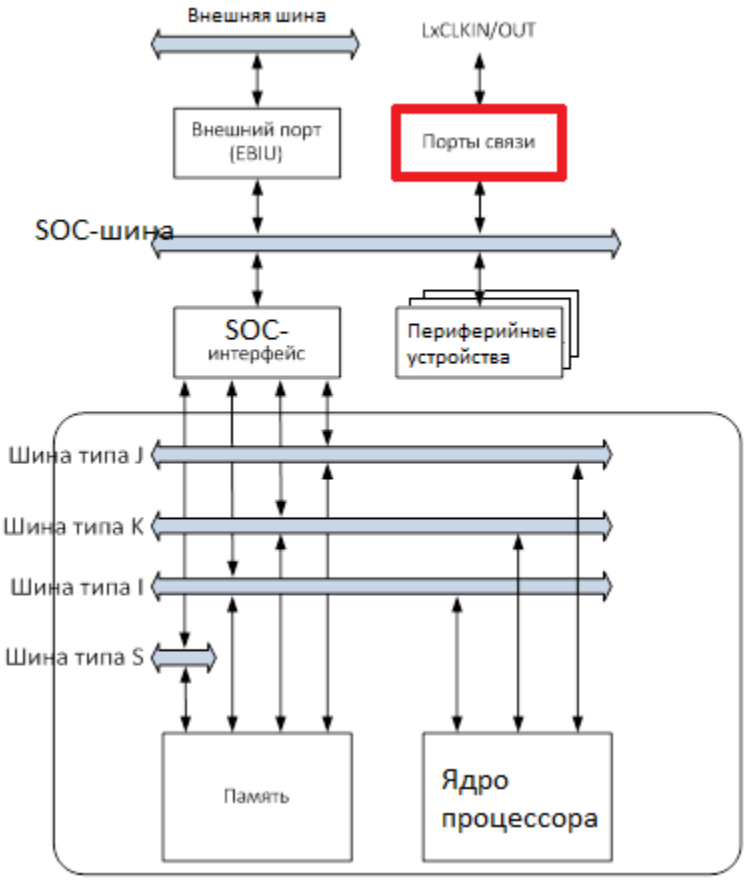


Рисунок 70 – Подключение портов связи на кристалле

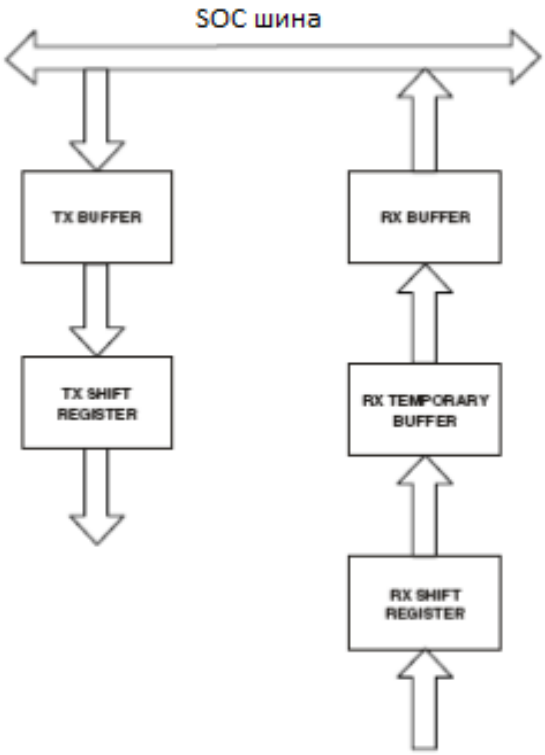


Рисунок 71 – Архитектура порта связи

16.2 Внешние выводы портов связи

Для выполнения обмена данными с внешними устройствами порты связи используют внешние выводы микросхемы, которые описаны ниже (Таблица 121). В таблице приведен перечень выводов одного порта связи. Знак ‘x’ в имени сигнала указывает на порт связи – 0 или 1. В набор внешних выводов порта связи входят как дифференциальные выводы, так и обычные выводы. Для каждого дифференциального вывода существует пара линий: P и N. На линии P передается истинное значение сигнала, а на линии N противоположное. Раздельные шины данных для приема и для передачи имеют разрядность 8 бит каждая. При этом имеется режим передачи с использованием только одного разряда шины данных из восьми.

Таблица 121 – Описание выводов – порты связи

Сигнал	Описание
LxDATO7–0P	Шина данных 7–0 передача LVDS P
LxDATO7–0N	Шина данных 7–0 передача LVDS N
LxCLKOUTP	Тактовый генератор передачи LVDS P
LxCLKOUTN	Тактовый генератор передачи LVDS N
LxACKI	Входной сигнал подтверждения передатчика. Используя этот сигнал приемник указывает передатчику, что можно продолжать передачу
nLxBCMPO	Завершение блока. Когда передача выполняется с использованием DMA, данный сигнал указывает приемнику, что передаваемый блок закончился. Во время сброса выводы L1BCMPO, L2BCMPO и L3BCMPO используются как входы для конфигурирования процессора. Перед началом инициализации порта, порт связи выдает сигнал высокого уровня на nLxBCMPO (деактивирован), указывая приемнику, что порт подсоединен
LxDATI7–0P	Данные 7–0 прием LVDS P
LxDATI7–0N	Данные 7–0 прием LVDS N
LxCLKINP	Тактовый генератор приема LVDS P
LxCLKINN	Тактовый генератор приема LVDS N
LxACKO	Выходной сигнал подтверждения приемника. Используя этот сигнал приемник указывает передатчику, что можно продолжать передачу.
nLxBCMPI	Завершение блока. Когда передача выполняется с использованием DMA, данный сигнал указывает приемнику, что передаваемый блок закончился. После сброса сигнал высокого уровня на nLxBCMPI указывает на то, что к приемнику порта подключен внешний передатчик

Каждый порт связи имеет два независимых канала, один для приема и второй для передачи, которые могу работать одновременно. Канал передачи передает данные на другое устройство, а канал приема получает данные с другого устройства. Каждый канал осуществляет обмен данными, используя до восьми бит данных и используя сигналы LxCLKOUTP/N, LxACKI, LxCLKINP/N и LxACKO для управления передачей данных.

Сигналы nLxBCMPI и nLxBCMPO используются для информирования о том, что текущая передача блока данных завершена. Порты связи должны соединяться так, как показано на Рисунках 72 и 73.

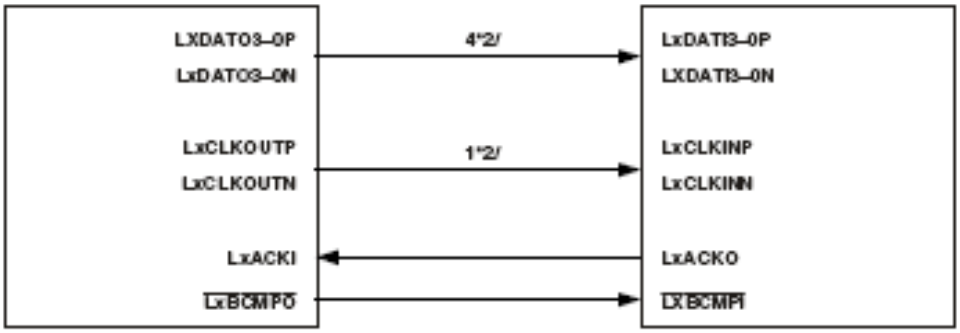


Рисунок 72 – Конфигурация порта связи (4-разрядный режим)

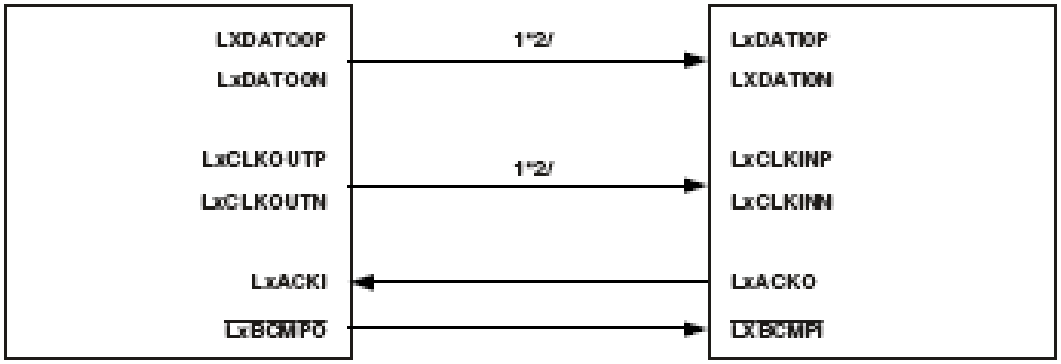


Рисунок 73 – Конфигурация порта связи (1-разрядный режим)

Порты связи процессора являются тактируемыми высокоскоростными LVDS портами данных. LVDS является стандартом для дифференциальной передачи сигналов между удаленными элементами. LVDS обеспечивает более высокую частоту, более высокий уровень устойчивости к шумам, более низкое энергопотребление и меньшее число электромагнитных помех.

Передача сигналов LVDS требует дифференциального завершения. Рисунок 74 показывает процесс передачи через порт от одного процессора к другому. Внешние резисторы согласования (RT) 100 Ом должны быть установлены на плате как можно ближе к выводам микросхемы. Линии на плате должны быть по возможности одинаковыми для того, чтобы поддерживать одинаковое время задержки для всех выводов данных и тактовых сигналов.

Линии несимметричных сигналов (LxACKI, LxACKO, nLxBCMPI и nLxBCMPO) не так критичны, но их задержки должны были близки к задержкам соответствующих им дифференциальных сигналов.

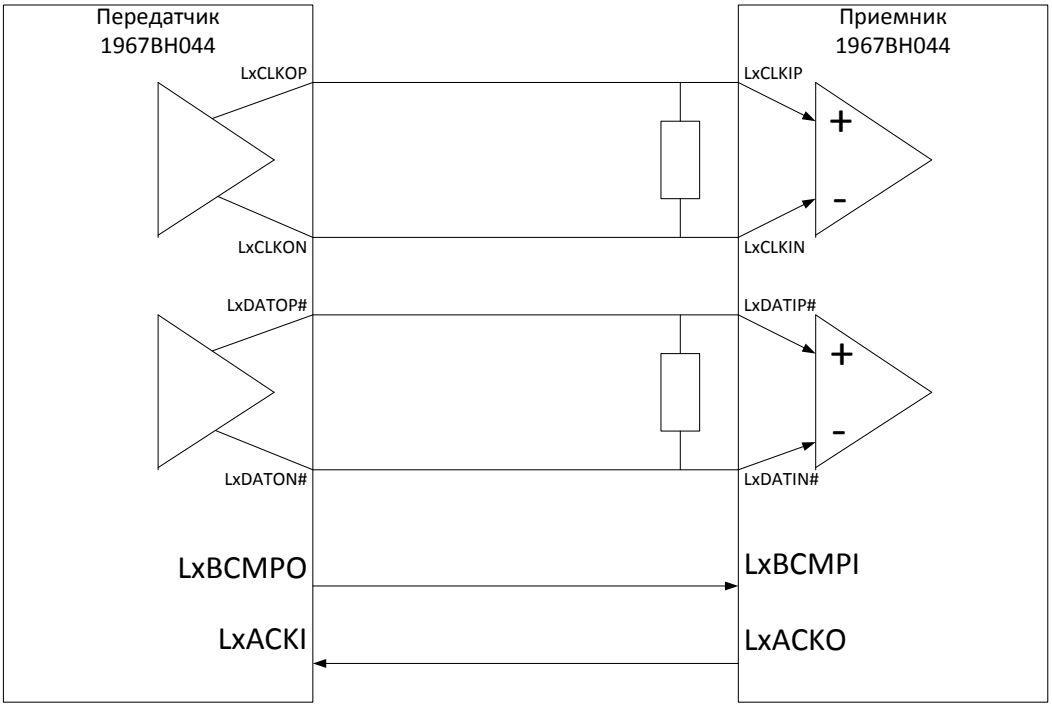


Рисунок 74 – Конфигурация передачи между процессорами

16.3 Группа регистров портов связи

Регистры буфера портов связи доступны как счетверенные слова. Две группы регистров данного типа описаны в таблицах 122 и 123. Регистры управления и статуса доступны как одинарные слова. Базовый адрес группы регистров буфера приема/передачи портов связи 0x8000_00A0 (Таблица 122), базовый адрес группы регистров статуса и контроля портов связи 0x8000_00E0 (Таблица 123).

Таблица 122 – Группа регистров буфера приема/передачи порта связи

Имя	Смещение	Тип	Значение после сброса	Описание
LBUFTX0:0	0x00	R/W	-	Данные передатчика порта связи 0
LBUFTX0:1	0x01	R/W	-	Данные передатчика порта связи 0
LBUFTX0:2	0x02	R/W	-	Данные передатчика порта связи 0
LBUFTX0:3	0x03	R/W	-	Данные передатчика порта связи 0
LBUFRX0:0	0x04	R	-	Данные приемника порта связи 0
LBUFRX0:1	0x05	R	-	Данные приемника порта связи 0
LBUFRX0:2	0x06	R		Данные приемника порта связи 0
LBUFRX0:3	0x07	R	-	Данные приемника порта связи 0
LBUFTX1:0	0x08	R/W	-	Данные передатчика порта связи 1
LBUFTX1:1	0x09	R/W	-	Данные передатчика порта связи 1
LBUFTX1:2	0x0A	R/W	-	Данные передатчика порта связи 1
LBUFTX1:3	0x0B	R/W	-	Данные передатчика порта связи 1
LBUFRX1:0	0x0C	R	-	Данные приемника порта связи 1
LBUFRX1:1	0x0D	R	-	Данные приемника порта связи 1
LBUFRX1:2	0x0E	R	-	Данные приемника порта связи 1
LBUFRX1:3	0x0F	R	-	Данные приемника порта связи 1

Таблица 123 – Группа регистров статуса и контроля порта связи

Имя	Смещение	Тип	Значение после сброса	Описание
LRCTL0	0x00	R/W	0x0	Регистр управления приемником порта связи 0
LRCTL1	0x01	R/W	0x0	Регистр управления приемником порта связи 1
-	0x02 - 0x03		-	Зарезервировано
LTCTL0	0x04	R/W	0x0	Регистр управления передатчиком порта связи 0
LTCTL1	0x05	R/W	0x0	Регистр управления передатчиком порта связи 1
-	0x06 - 0x0B		-	Зарезервировано
LRDLY0	0x0C	R/W	0x0	Регистр управления задержками LVDS-приемников порта связи 0
LRDLY1	0x0D	R/W	0x0	Регистр управления задержками LVDS-приемников порта связи 1
-	0x0E - 0x0F		-	Зарезервировано
LRSTAT0	0x10	R	0x0	Регистр состояния приемника порта связи 0
LRSTAT1	0x11	R	0x0	Регистр состояния приемника порта связи 1
-	0x12 - 0x13		-	Зарезервировано
LTSTAT0	0x14	R	0x2	Регистр состояния передатчика порта связи 0
LTSTAT1	0x15	R	0x2	Регистр состояния передатчика порта связи 1
-	0x16 - 0x17		-	Зарезервировано
LRSTATC0	0x18	R	0x0	Регистр сброса состояния приемника порта связи 0
LRSTATC1	0x19	R	0x0	Регистр сброса состояния приемника порта связи 1
LTSTATC0	0x1A	R	0x2	Регистр сброса состояния передатчика порта связи 0
LTSTATC1	0x1B	R	0x2	Регистр сброса состояния передатчика порта связи 1
-	0x1C - 0x1F		-	Зарезервировано

16.3.1 Регистр сброса состояния приемника/передатчика порта связи

Чтение регистра возвращает значение регистра состояния передатчика/приемника с автоматическим обнулением некоторых бит регистра статуса приемника/передатчика.

16.4 Прием и передача данных

Обмен данными осуществляется посредством записи в буфер передачи и чтением из буфера приема. Длина передаваемых данных всегда равна 128 битам. Все данные, записанные в буфер передачи, копируются в сдвиговый регистр, как только он становится пустым, и после этого передаются. Приемник разрешает продолжение приема данных, только когда его сдвиговый регистр пуст или, когда в его буферных регистрах есть достаточно места для приема данных из сдвигового регистра, когда прием этого счетверенного слова будет завершен. После того, как

целое счетверенное слово получено, приемник перемещает данные из сдвигового регистра в буфер приема, когда он свободен.

Если приемник не готов принять следующее слово данных, он использует линию подтверждения LxACK0 для приостановки работы передатчика.

В таблицах (Таблица 122, Таблица 123) приведены имена регистров, которые пользователь может использовать при программировании портов связи.

16.5 Связь с DMA

Каждый порт связи может соединяться с двумя каналами DMA. Один канал используется для передачи данных, в то время как второй используется для приема данных. Оба канала DMA связаны через интерфейс с внутренней памятью, внешней памятью или другими буферами портов связи. Передатчики портов связи с 0 по 1-й соединены соответственно с 4-го по 5-й каналами DMA. Приемники портов связи с 0 по 1-й соединены соответственно с 8-го по 9-й каналами DMA. Каналы DMA с 4-го по 5-й называются каналами передачи, т.к. передают данные передатчикам порта связи, а каналы с 8 по 9-й называются каналами приема, т.к. принимают данные от приемников порта связи.

Передатчик порта связи формирует сервисный запрос к каналу передачи DMA, когда буферный регистр LBUFTX пуст и канал DMA включен. Приемник порта связи формирует запрос к каналу приема DMA, когда он записывает счетверенное слово данных в буферный регистр LBUFRX и канал DMA включен.

Канал-приемник DMA (с 8 по 9) может также использоваться для транзитных передач посредством записи данных из приемника своего порта связи в буферный регистр передатчика любого другого порта связи. Канал DMA при этом отслеживает, что буферный регистр передатчика свободен.

16.6 Завершение блочной передачи

Функция завершения передачи блока позволяет передатчику информировать приемник о том, что блочная передача завершена.

Поскольку канал передачи DMA точно знает, сколько слов необходимо передать и может отследить момент передачи последнего слова, то вместе с записью последнего слова в буфер передатчика, он формирует сигнал завершения передачи блока и передает его в передатчик вместе с данными. Порт связи активизирует nLxBCMPO, когда он передает последнее счетверенное слово и в регистре управления LTCTL установлен разряд BCMPE.

Когда вход nLxBCMPI активен, то принимающий порт связи передает эту информацию каналу DMA вместе с запросом. Это позволяет каналу приемнику DMA отследить завершение приема и закончить свою работу.

16.7 Прерывания портов связи

Порты связи имеют специализированные прерывания для управления потоком данных, когда порт связи осуществляет передачу с использованием ядра процессора (в отличие от передачи с использованием каналов DMA). Прерывание от приемника порта связи является активным только тогда, когда канал DMA для порта связи не включается. Порт приема активизирует прерывание, когда счетверенное слово, получаемое портом связи, ожидает в регистре LBUFRX. Прерывание приема порта связи является чувствительным по уровню и поэтому, если канал DMA, взаимодействующий с портом связи, становится активным после разрешения

прерывания порта связи, прерывание приема порта связи будет деактивировано и будет сформирован запрос к DMA.

16.8 Инициализация после сброса и загрузка через порт связи

Старт процессора может быть выполнен посредством загрузки начального кода программы через порты связи. В этом режиме порт работает как подчиненное устройство, настроенное на прием данных. После сброса все каналы DMA порта приема могут настраиваться для передачи блока из 256 слов во внутреннюю память по адресам с 0 до 255 и настраиваются для формирования прерывания по окончании передачи блока. Соответствующие вектора прерываний для каналов DMA устанавливаются в нулевой адрес.

При сбросе или запрещении портов связи через регистры управления, содержимое буферов порта связи сбрасывается.

16.9 Протокол передачи данных порта связи

Порт связи осуществляет передачу данных через 1-, 4- или 8-разрядную шину данных по каждому направлению (TX или RX), с использованием трех типов контрольных сигналов (Таблица 121). В одноразрядном режиме для передачи используются выводы LxDATI0P/N и LxDATO0P/N.

Каждый порт связи имеет два независимых канала, которые могут работать одновременно. Канал передачи передает данные на внешнее устройство, канал приема получает данные с внешнего устройства. Каждый канал осуществляет передачу данных с использованием одного, четырех или восьми разрядов данных, используя сигналы LxCLKOUTP/N, LxACKI, LxCLKINP/N и LxACKO для управления передачей данных. Сигналы nLxBCMPI и nLxBCMPO используются для оповещения о завершении текущей передачи блока.

Существует несколько общих правил, которые применяются в протоколе порта связи (Рисунок 75 – Рисунок 82). Знание этих правил позволяет понять примеры, приведенные на диаграммах. Общие правила таковы:

- первые данные передаются на нарастающем (из 0 в 1) фронте тактового сигнала порта связи (условно LXCLKOUTP);
- последние данные передаются на спадающем (из 1 в 0) фронте тактового сигнала порта связи (условно LXCLKOUTP);
- LXCLKOUTP переводится в низкий уровень, когда порт связи находится в состоянии простоя.

Минимальный объем передачи данных через порт равен счетверенному слову. Счетверенное слово может быть передано:

- в течение 8 тактовых циклов, когда используется 8 разрядов шины данных;
- в течение 16 тактовых циклов, когда используются четыре разряда шины данных;
- в течение 64 циклов, когда используется только один разряд данных.

На рисунках ниже показаны случаи обмена для различных шин данных.

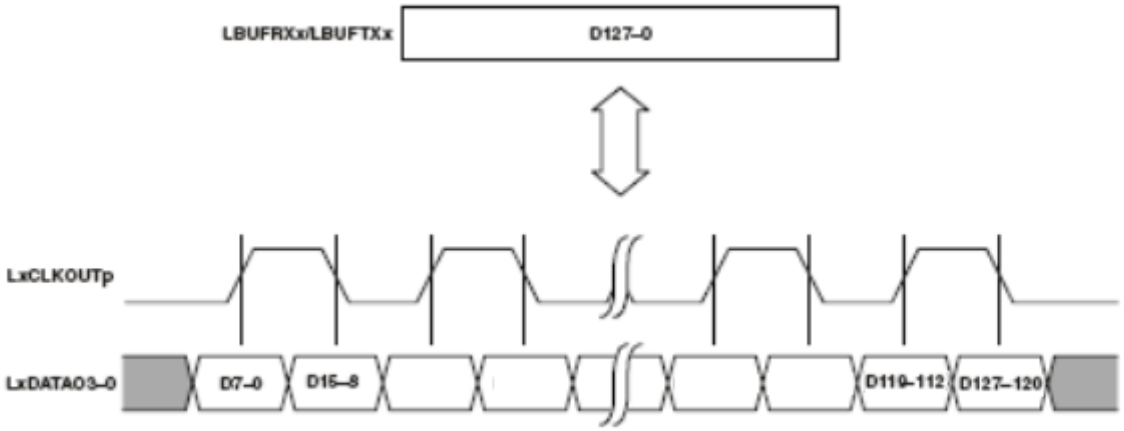


Рисунок 75 – Шина данных 8 разрядов

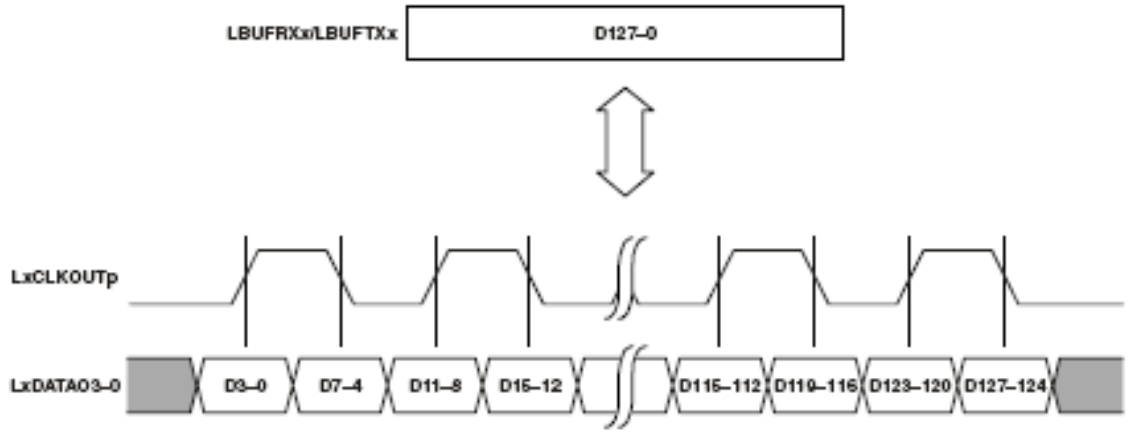


Рисунок 76 – Шина данных 4 разряда

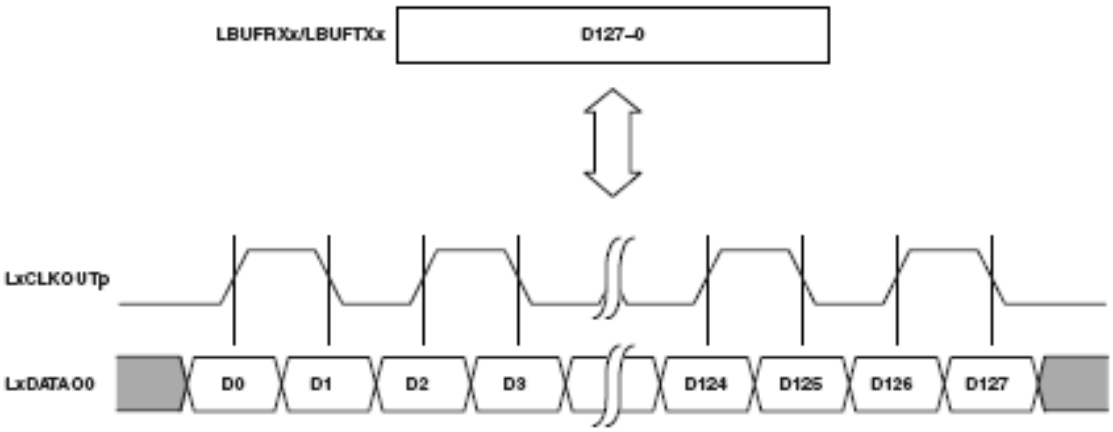


Рисунок 77 – Шина данных 1 разряд

Передача данных начинается, когда на LxACKI высокий уровень сигнала, что обозначает, что буфер приема свободен. Как показано на рисунках, первые биты данных достоверны до первого нарастающего фронта LxCLKOUTP и следующие за ними биты данных достоверны до спадающего фронта тактового сигнала.

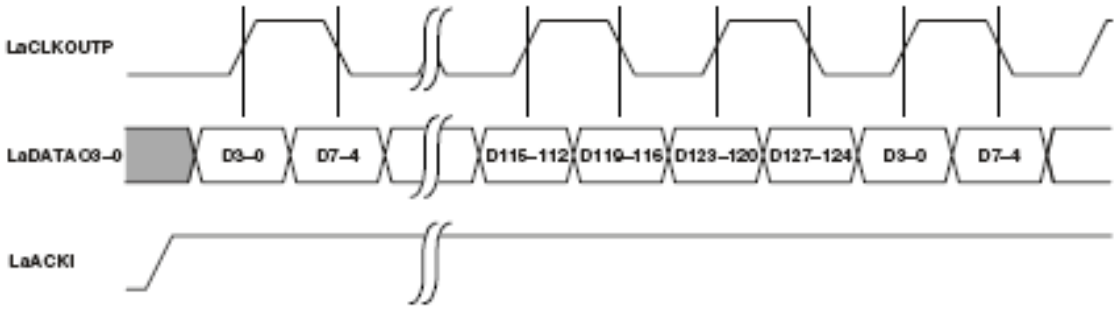


Рисунок 78 – Порт связи “a” передает слова порту связи “b” одно за другим

Рисунок 79 иллюстрирует состояние порта связи “b”, когда он не готов принимать больше данных. В этом случае после передачи текущего слова данных передатчик будет ждать пока сигнал подтверждения не примет высокий уровень.

На последовательность передачи влияет установка разрядов верификации данных. Это разряд RVERE в регистре LRCTL и разряд TVERE в регистре LTCTL. Если установлен разряд TVERE, байт контрольной суммы отправляется после последнего байта в счетверенном слове. За байтом контрольной суммы всегда следует «пустой» байт. Отправка байта контрольной суммы (биты Vx) и «пустого» байта (Xx) выполняется в течение двух циклов для шины 4-разрядных данных и восемь циклов для шины 1-разрядных данных. Рисунок 80 иллюстрирует последовательность передачи для случая 1-разрядной шины данных.

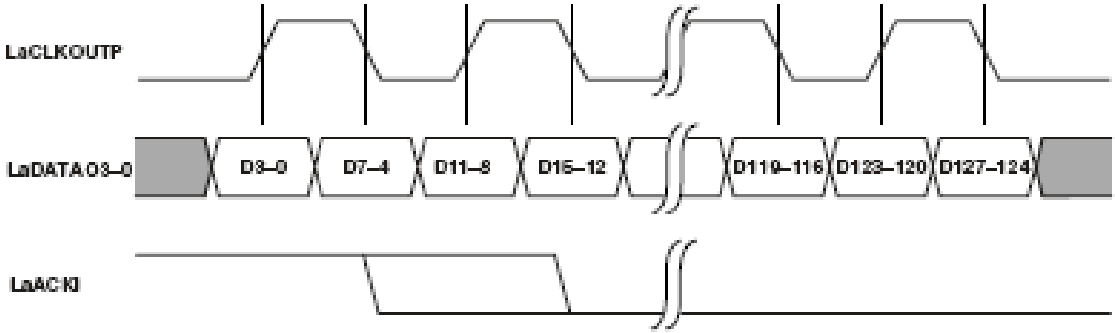


Рисунок 79 – Порт связи “a” передает порту связи “b” (приемник заполнен)

Сообщения между различными устройствами в системе обычно имеют различную длину. В некоторых случаях блоки данных также имеют различную длину. При этом знание длины блока не всегда доступно для принимающего устройства и не может быть передано передающим устройством. В этом случае на помощь приходят управляющие сигналы nLxBCMPO и nLxBCMPI.

Передающий порт связи указывает порту связи приема, что блок завершен с использованием выходного сигнала nLxBCMPO передатчика, который связан с входным сигналом приемника nLxBCMPI.

Когда приемник распознает этот сигнал, он передает информацию каналу DMA о том, что блок данных завершен. В результате этого канал DMA игнорирует то, что его счетчик слов еще не достиг значения нуля, и работает так, как при завершении блока.

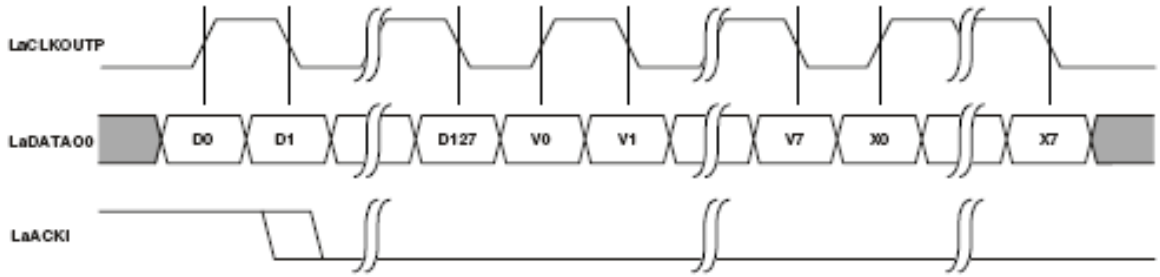


Рисунок 80 – Порт связи “a” передает порту связи “b” с верификацией

Сигнал nLxBCMPO указывает на завершение блока переходом на низкий уровень после первого нарастающего фронта LxCLKOUT в последнем счетверенном слове блока. На nLxBCMPO устанавливается высокий уровень сигнала (неактивный) перед последним спадающим фронтом сигнала LxCLKOUT того же счетверенного слова. Сигнал является неактивным, если разряд TBCMPE сбрасывается в регистр LTCTL или, когда передачи порта связи выполняются под управлением ядра процессора.

Рисунок 81 показывает, как порт связи “a” передает сигнал порту связи “b” о том, что текущий буфер завершен.

Рисунок 81 – Порт связи “a” передает порту связи “b” с завершением блока

Порт связи “a” передает сигналы порту связи “b” о том, что текущий буфер заполнен (Рисунок 82). Новое счетверенное слово следует за завершенным блоком.

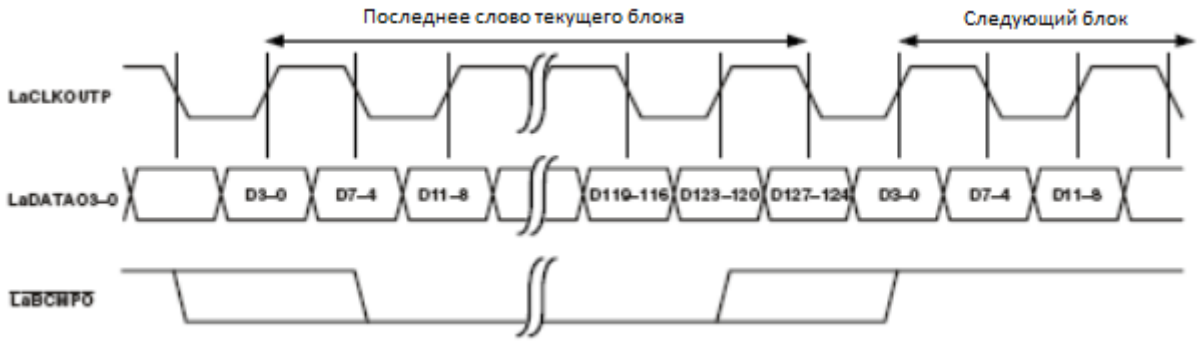


Рисунок 82 – Порт связи “a” передает порту связи “b” с завершением блока (новый блок из порта связи “a” следует немедленно после завершения первого блока)

16.10 Задержки передачи через порт связи

Порты связи должны быть в состоянии преодолевать задержки между источником и пунктом назначения. Задержки для различных типов сигналов (LxCLKOUT и LxDAT7–O) должны соответствовать допускам, указанным в техническом описании. Существует два типа ограничений для задержек в системе:

- задержка от LxCLKOUT в передатчике к LxCLKIN в приемнике плюс задержка от LxACKO в приемнике к LxACKI в передатчике должны быть меньше чем половина периода передачи счетверенного слова.
- максимальная разница между задержкой трассировки nLxBCMPO и задержкой трассировки тактового сигнала (и данных) составляет одну треть периода передачи счетверенного слова.

Первое ограничение связано со следующим фактором. Если передатчик отправляет данные, а приемник передает сигнал о том, что он не готов принять больше данных через LxACKO, передатчик должен сначала обработать LxACKI до того, как он отправит следующие данные. Если LxACKI не проанализирован вовремя, данные будут утеряны приемником.

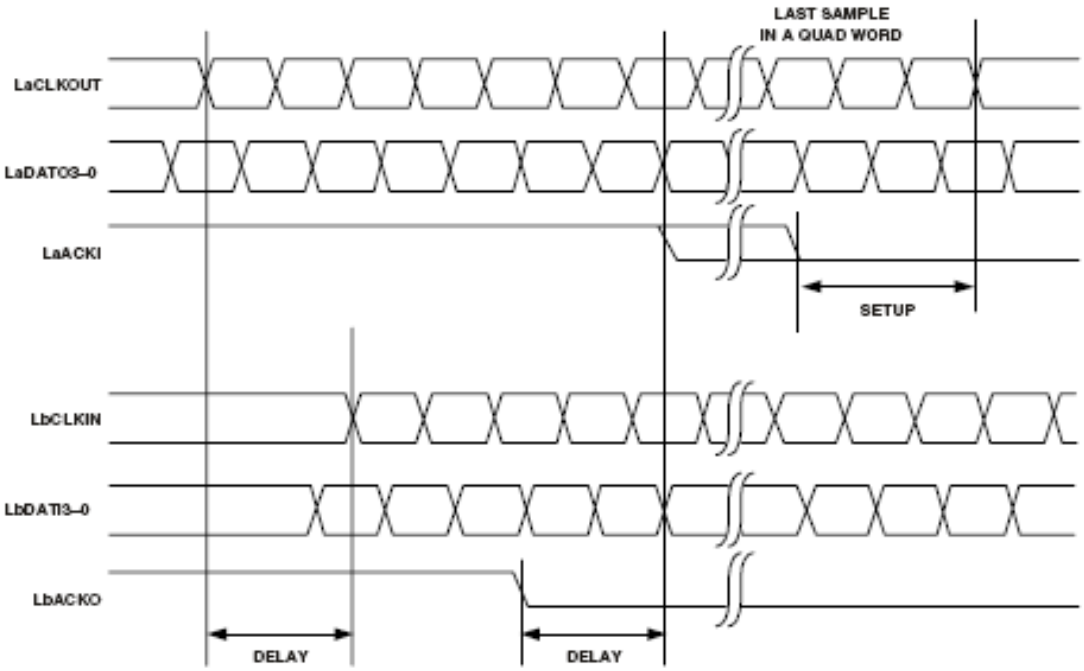


Рисунок 83 – Порт связи “a” передает порту связи “b” с задержкой

16.11 Механизмы определения ошибок порта связи

Порты связи поддерживают функции обнаружения ошибок в процессе работы. Когда порт связи обнаруживает состояние ошибки, он продолжает работу в соответствии со следующим алгоритмом:

- Если обнаружена ошибка истечения времени, устанавливаются соответственно разряды TTER (ошибка истечения времени передачи) в регистр LTSTAT или RTER (ошибка истечения времени приема) в регистр LRSTAT. Если ошибки истечения времени разрешены в регистрах LRCTL или LTCTL, активируется аппаратное прерывание.

- Если обнаружено переполнение данных в приемнике, разряд ROVER устанавливается в LRSTAT. Если разрешена ошибка переполнения данных приема в LRCTL, активируется аппаратное прерывание.
- Если верификация разрешена в LRCTL и обнаружена неверная контрольная сумма, разряд RCSER устанавливается соответственно в LRSTAT и активируется аппаратное прерывание.
- Если активное значение записывается в LRCTL или LTCTL и значение в регистре управления является активным, перезапись регистра предотвращается, разряд RWER устанавливается в LRSTAT или разряд TWER устанавливается в LTSTAT соответственно, активируется аппаратное прерывание.

Поля в LRSTAT и LTSTAT, указывающие на ошибку, могут быть очищены с помощью чтения из соответствующих очищенных регистров LRSTATC и LTSTATC.

16.11.1 Время ожидания передачи через порт связи

Если в передатчике LxACKI деактивирован в течение периода 2048 тактовых циклов и в передатчике есть данные, готовые к передаче, происходит ошибка времени ожидания (time-out) и разряд TTER устанавливается в LTSTAT. На линии LxACKI имеется слабый резистор, понижающий логический уровень. В случае если нет подключения приемника, вход будет в неактивным состоянии. Данная ошибка позволяет обнаружить отсутствие приемника вообще или сбой в его работе. Также эта ошибка позволяет предотвратить возможное зависание системы в случае ожидания готовности передачи.

16.11.2 Время ожидания приемника

Если принимающий буфер приемника полон и нет доступа для чтения в течение 2048 тактовых циклов, происходит ошибка окончания времени ожидания и разряд RTER устанавливается в LRSTAT. Данная ошибка информирует процессор о возможной некорректной работе программы обслуживания канала связи.

16.11.3 Ошибка верификации порта связи

Передатчик имеет возможность формировать байт контрольной суммы и отправлять его вместе с передаваемым счетверенным словом. Передача контрольного байта разрешается специальным битом регистра управления. Контрольный байт рассчитывается как сумма всех байтов данных, которые были переданы. 128-разрядное слово состоит из 16 байтов и контрольный байт отправляется в конце передачи каждых 16 байтов. Приемник принимает первые 16 байт данных, производит вычисление собственной контрольной суммы и сравнивает ее с принятым значением. Если два байта различаются, разряд RCSER устанавливается в регистр LSTAT и формируется аппаратное прерывание.

Алгоритм контрольной суммы следующий:

- Checksum = младший байт от суммы (B0 + B1 + … + B15).

16.11.4 Ошибка записи приема/передачи через порт связи

Запись активного значения в регистр управления приемника или передатчика является допустимой только при выключенном приемнике или передатчике. Запись активного значения в активный управляющий регистр вызывает ошибку записи. Таким образом для того, чтобы поменять некоторые параметры обмена необходимо предварительно выключить соответствующий приемник или передатчик и затем включить его снова с требуемым значением.

16.12 Регистр управления приемником порта связи (LRCTLx)

Регистры LRCTLx определяют параметры приема для порта связи. Подробное описание разрядов регистра приведено ниже (таблица 124).

Таблица 124 – Регистр LRCTL

Бит	Имя	Назначение
0	REN	Бит включения приемника: 1 – включен 0 – выключен
1	RVERE	Разрешение контроля при приеме 1 – разрешено 0 – запрещено
2	RTOE	Разрешение прерывания в случае обнаружения ошибки time out
3	RBCMPE	Разрешение анализа входа nLxBCMPI : 1 – разрешено 0 – запрещено
5:4	RDSIZE	Размер шины данных приема: 11 – 16 бит 10 – 8 бит 01 – 4 бита 00 – 1 бит
6	ROVRE	Разрешение прерывания при переполнении буфера приемника: 1 – разрешено 0 – запрещено
7	-	Зарезервировано
8	-	Зарезервировано
9	GPS_CLK_EN	Разрешение генерации клока GPS
10	-	Зарезервировано
11	DATA_SRC	Выбор приемника данных: 0 – буфер порта связи 1 – модуль UP-DOWN
13:12	ADCW	Режим для ADC 8 бит шины данных 00 – 16 бит 01 – 14 бит 10 – 12 бит 11 – 10 бит
14	COMPL	Режим представления данных 0 – входные данные представлены в дополнительном коде и не преобразуются для дальнейшей обработки 1 – входные данные представлены в прямом коде и преобразуются в дополнительный для дальнейшей обработки
15	ADC_DDR	Режим ADC DDR 0 – четные биты передаются по фронту, нечетные по срезу 1 – первая половина бит передается по фронту, вторая по срезу

Бит	Имя	Назначение
16	RX_CLK_SRC	Источник клока: 0 – L0CLKI, 1 – L1CLKI
31:17	-	Зарезервировано

Регистр не может изменяться в процессе выполнения операции обмена. Запись активного значения в контрольный регистр допускается, только когда регистр имеет неактивное значение (REN сброшен). С целью изменения настроек во время работы порта связи, должно быть записано неактивное значение, чтобы прекратить работу порта, после чего следует записать новое активное значение. Игнорирование данного правила может вызвать аппаратное прерывание ошибки и новое значение не будет записано.

Выбор разрядности шины данных для обмена осуществляется в соответствии со следующим правилом:

- если бит 2 регистра CFG_APB модуля CMU установлен в 1, разрядность шины данных приемников портов связи равна 4 бита;
- если бит 2 регистра CFG_APB модуля CMU сброшен в 0, разрядность шины данных приемников портов связи определяется согласно значению бит RDSIZE.

16.13 Регистр управления передатчиком порта связи (LTCTLx)

Регистр устанавливает параметры передачи для порта связи. Значением сброса для регистров LTCTLx является 0. Подробное описание разрядов регистра приведено ниже (таблица 125).

Таблица 125 – Регистр LTCTL

Бит	Имя	Назначение
0	TEN	Бит включения передатчика: 1 – включен 0 – выключен
1	TVERE	Разрешение формирования контрольной суммы при передаче: 1 – разрешено 0 – запрещено
2	TTOE	Разрешение прерывания в случае обнаружения ситуации time out
3	TBCMPE	Разрешение формирования выхода LxBCMPO: 1 – разрешено 0 – запрещено
5:4	TDSIZE	Размер шины передачи: 11 – 16 бит 10 – 8 бит 01 – 4 бита 00 – 1 бит
8:6	-	Зарезервировано
10:9	TXC	Источник синхросигнала передатчика: 00, 10 – внутренний генератор (частота формируется синтезатором тактовой частоты LINK_PLL) 01 – синхросигнал приемника 11 – инвертированный синхросигнал приемника
11	TX_DATA_DST	Выбор источника данных для передачи: 0 – передатчик порта связи 1 – модуль UP-DOWN

Бит	Имя	Назначение
12	TX_COMPL	Режим представления данных 0 – выходные данные представлены без изменения 1 – каждые 16 бит выходных данные рассматриваются как независимые и модифицируются инверсией знакового бита
13	TX_DAC_MODE	Режим внешнего DAC 0 – нормальная работа. Сигнал LxACKI анализируется 1 – работа с непрерывным потоком данных. Сигнал LxACKI игнорируется
31:14	-	Зарезервировано

Регистр не может изменяться в процессе выполнения операции обмена. Запись активного значения в контрольный регистр допускается, только когда регистр имеет неактивное значение (TEN сброшен). С целью изменения настроек во время работы порта связи, должно быть записано неактивное значение, чтобы прекратить работу порта, после чего следует записать новое активное значение. Игнорирование данного правила может вызвать аппаратное прерывание ошибки и новое значение не будет записано.

16.14 Регистр управления задержками приемника порта связи (LRDLYx)

Регистр устанавливает индивидуальные задержки на каждый бит приемника порта связи. Меняя задержки для каждого бита, можно снижать влияние внешней линии связи для обеспечения устойчивого приема. В зависимости от варианта микросхемы, напряжения питания и температуры (PVT), величина единичной задержки может находиться в пределах от 224 до 346 пс. Полная задержка определяется как произведение единичной задержки на десятичное значение из DxDLY.

Таблица 126 – Регистр LRDLY

Бит	Имя	Назначение
2:0	D0_DLY	Задержка бита 0 данных
5:3	D1_DLY	Задержка бита 1 данных
8:6	D2_DLY	Задержка бита 2 данных
11:9	D3_DLY	Задержка бита 3 данных
14:12	D4_DLY	Задержка бита 4 данных
17:15	D5_DLY	Задержка бита 5 данных
20:18	D6_DLY	Задержка бита 6 данных
23:21	D7_DLY	Задержка бита 7 данных
26:24	CLK_DLY	Задержка синхросигнала

16.15 Регистр состояния приемника порта связи (LRSTATx)

Регистры LRSTATx указывают статус приемника порта связи. Значением сброса для регистров LRSTATx является 0x0000 0000. Подробное описание разрядов регистра приведено в Таблице 127.

Таблица 127 – Регистр LRSTAT

Бит	Имя	Назначение
0	RSTAT	Состояние буфера приемника: 0 – буфер пустой 1 – буфер не пустой

Бит	Имя	Назначение
1	RSTAT_ADD	Состояние промежуточного буфера приемника (вспомогательный бит, не является битом готовности данных для считывания программой) 0 – вспомогательный буфер пустой 1 – вспомогательный буфер не пустой
2	RTER	1 – приемник обнаружил ситуацию time out 0 – нет ошибки Бит может быть сброшен чтением регистра LRSTATCx
3	RWER	1 – приемник обнаружил ошибку записи 0 – нет ошибки Бит может быть сброшен чтением регистра LRSTATCx
4	RCSER	1 – обнаружена ошибка контрольной суммы 0 – нет ошибки Бит может быть сброшен чтением регистра LRSTATCx
5	ROVER	1 – ошибка переполнения приемника 0 – нет ошибки Бит может быть сброшен чтением регистра LRSTATCx
6	RINIT	Состояние инициализации приемника: 1 – проинициализирован 0 – не инициализировался
31:7	-	Всегда ноль

16.16 Регистр состояния передатчика порта связи (LTSTATx)

Регистры LTSTATx указывают статус передатчика порта связи. Значением сброса регистров LTSTATx является = 0x0000 0002. Подробное описание разрядов регистра приведено ниже (Таблица 128).

Таблица 128 – Регистр LTSTAT

Бит	Имя	Назначение
0	TVACANT	1 – буфер может принять данные 0 – буфер не может принять данные
1	TEMPTY	1 – передатчик пуст 0 – передатчик занят Бит может быть использован при выключении передатчика.
2	TTER	1 – ошибка передачи time out 0 – нет ошибки Бит может быть сброшен чтением регистра LTSTATCx
3	TWER	1 – ошибка записи 0 – нет ошибки Активное значение было записано в регистр управления во время, когда передатчик был включен. Бит может быть сброшен чтением регистра LTSTATCx
31:4	-	Всегда ноль

16.17 Последовательность включения портов связи

В силу особенностей примененных схемотехнических решений при работе с портами связи необходимо соблюдать следующую последовательность их программирования:

1 при настройке каналов необходимо в первую очередь включить передатчик (бит TEN регистров LTCTLx), т.к. в момент включения

аналоговой части LVDS-передатчика возможно появление ложного импульса;

2 включение приемника необходимо производить как минимум в два этапа. Сначала необходимо записать в регистры LRCTLx все настройки, за исключением бита REN, а последней транзакцией подтвердить все настройки этих регистров и установить бит REN в единицу.

17 Последовательный асинхронный интерфейс UART

Процессор имеет два интерфейса UART. Интерфейс имеет гибкую систему программирования, позволяющую задать различные скорости обмена, длину посылки, контроль четности, а также различные ситуации прерываний. Интерфейс имеет встроенные FIFO глубиной 8 байт для приема и передачи. Возможности интерфейса позволяют ему работать с контроллером DMA.

Внешние выводы интерфейса UART приведены в Таблице 129

Таблица 129 – Внешние выводы

Обозначение вывода (основное)	Обозначение вывода для интерфейса UART	Тип вывода	Функциональное назначение
PA[0]	U0_TXD	I/O	Интерфейс UART0. Выход передатчика
PA[1]	U0_RXD	I/O	Интерфейс UART0. Вход приемника
PA[2]	U1_TXD	I/O	Интерфейс UART1. Выход передатчика
PA[3]	U1_RXD	I/O	Интерфейс UART1. Вход приемника

Интерфейсы имеют различные базовые адреса в карте памяти процессора: UART_0 – **0x80000100**, UART_1 – **0x80000120**. Каждый из интерфейсов имеет набор регистров, приведенный ниже (Таблица 130).

Таблица 130 – Регистры интерфейса

Номер регистра	Обозначение регистра	Описание	Состояние по сбросу
0	UARTDR	Регистр данных	-
1	URXSTAT	Регистр состояния приемника	0
2	UCR_load	Регистр управления загрузка	0
3	UCR_set	Регистр управления установка	0
4	UCR_clear	Регистр управления сброс	0
5	UBitRate	Регистр управления скоростью обмена	0
6	UFLAG	Регистр флагов	-
7	UINTM	Регистр запросов прерываний (маскированный)	0
8	UINT	Регистр запросов прерываний (немаскированный)	0

Условная структурная схема интерфейса приведена ниже (Рисунок 84). Регистры интерфейса позволяют пользователю задать требуемый режим работы: формат посылки, скорость обмена, источник прерываний и др. Рассмотрим более подробно назначение регистров интерфейса.

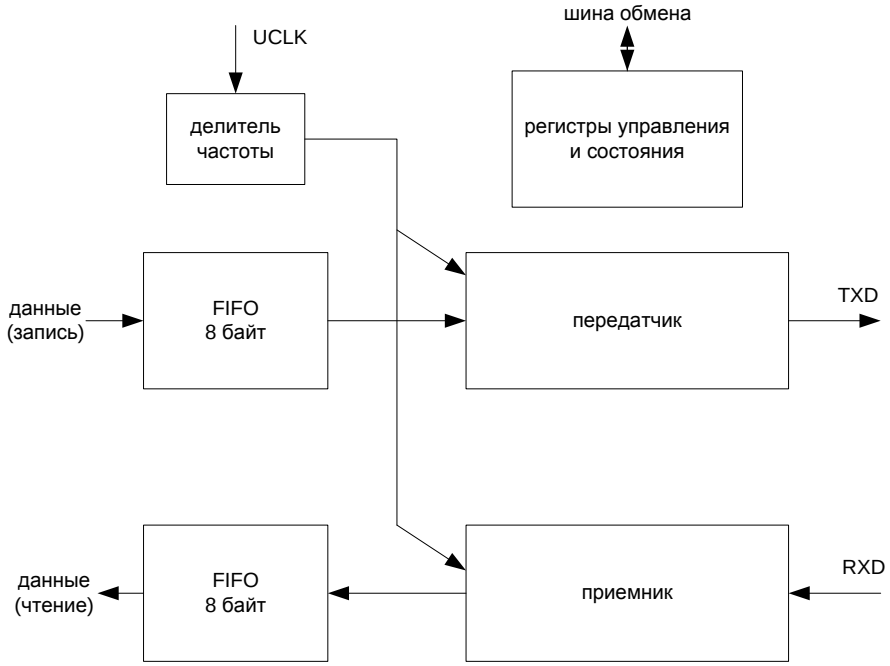


Рисунок 84 – Структурная схема UART

17.1 Регистр данных UARTDR

Регистр *UARTDR* представляет собой буфер данных передатчика при записи и буфер данных приемника при чтении. В качестве такого буфера используется FIFO емкостью 8 байт. Это позволяет выполнить единовременную загрузку 8 байт данных в буфер передатчика или обслуживать приемник только в случае заполнения буфера приемника не менее, чем наполовину. Это сокращает количество прерываний процессора на обслуживание интерфейса. Буфер приемника содержит не только байты данных, но и соответствующие им признаки ошибок приема-передачи. Во время чтения байта данных, происходит копирование соответствующих ему признаков в регистр состояния RXSTAT. Несмотря на то, что интерфейс подключен к 32-разрядной шине, при обмене данными используются только 8 младших бит шины.

17.2 Регистр состояния RXSTAT

Таблица 131 – Регистр RXSTAT

Бит	Обозначение	Тип	Описание
0	FRAME	R	Ошибка в формате посылки 0 – нет ошибки 1 – ошибка
1	PARITY	R	Ошибка четности 0 – нет ошибки 1 – ошибка
2	OVERRUN	R	Переполнение буфера приемника 0 – нет переполнения 1 – переполнение. Бит очищается после чтения регистра.
3	ERROR	R	Устанавливается в 1, если одна из выше описанных ошибок обнаруживается.
7:4		-	Не используется

Бит FRAME устанавливается в 1, если во время приема не был обнаружен Стоп-бит, т.е. посылка не завершена корректно. Бит PARITY устанавливается в 1, если в принятом байте обнаружена ошибка четности\нечетности. Бит OVERRUN устанавливается в 1, если после приема очередного байта обнаруживается, что буфер приемника уже полон и нет места для записи принятого байта. В этом случае происходит потеря принятого байта и установка признака переполнения. Если после чтения текущего байта данных и его регистра состояния обнаруживается, что бит переполнения, установлен, то еще, по крайней мере, 7 байт данных в буфере являются достоверными.

17.3 Регистры управления скоростью обмена UBitRate

Интерфейс содержит в себе программируемый делитель частоты обмена разрядностью 16 бит.

Таблица 132 – Регистр UBitRate

Бит	Обозначение	Тип	Описание
15:0	BR	R/W	Биты делителя

Входной частотой интерфейса является внешний вход синхронизации XTI. Внутри интерфейса она может дополнительно делиться на величину, заданную в регистре управления скоростью обмена, увеличенную на 1. Так, если значение равно 0x03, делитель будет делить входную частоту на 4. Эта частота будет использована приемником для сканирования принимающей линии. При этом предполагается, что длительность одного бита принимаемой посылки определяется битом UHBRE регистра UCR и может составлять 4 или 16 тактов базовой частоты. Таким образом, реальная скорость приема – передачи по интерфейсу равна (XTI / (***UBitRate*** + 1)) / (16 или 4).

17.4 Регистр управления интерфейсом UCR

Таблица 133 – Регистр UCR

Бит	Обозначение	Тип	Описание
0	UARTEN	R/W	Разрешение работы UART интерфейса 0 – выключен 1 – включен
1	UTXDIS	R/W	Управление передатчиком 0 – включен 1 – выключен
2	URXDIS	R/W	Управление приемником 0 – включен 1 – выключен
3	DMAONERR	R/W	Управление запросом к контроллеру DMA во время обнаружения ошибки приема 0 – обнаружение ошибки не влияет формирование запроса к контроллеру DMA на обслуживание. 1 – формирование запроса к контроллеру DMA на обслуживание запрещено при обнаружении ошибки.
4	UTXFDIS	R/W	Управление буфером передатчика 0 – FIFO включено 1 – FIFO выключено

Бит	Обозначение	Тип	Описание
5	URXFDIS	R/W	Управление буфером приемника 0 – FIFO включено 1 – FIFO выключено
7:6	UHBRE	R/W	Управление длительностью одного бита посылки 00 – 1/16 clock 01 – 1/4 clock 1x – 1/4 clock, выключение внутреннего делителя
8	BREAK	R/W	Если равен 1, линия TXD передатчика устанавливается в высокий уровень
9	PRTEN	R/W	Управление проверкой и формированием бита четности 0 – запрещено 1 – разрешено
10	EVENPRT	R/W	Выбор проверки на четность\нечетность 0 – дополнение посылки до четного 1 – дополнение посылки до нечетного
11	XSTOP	R/W	Количество Стоп-битов (передача) 0 – один стоп-бит 1 – два стоп-бита приемник не проверяет количество принятых стоп-бит.
12	UFIFOEN	R/W	Управление буфером FIFO 0 – выключено (глубина буфера равна 1 слову) 1 – включение FIFO приемника и передатчика
14:13	WRDLEN	R/W	Выбор длины передаваемых данных 00 – 8 bits 01 – 7 bits 10 – 6 bits 11 – 5 bits
15	INV	R/W	Инверсия TXD линии когда 1
16	TXINT	R/W	Управление разрешением прерывания от флага TXINT 0 – запрещено 1 – разрешено
17	RXINT	R/W	Управление разрешением прерывания от флага RXINT 0 – запрещено 1 – разрешено
18	RXERRINT	R/W	Управление разрешением прерывания от флага RXERRINT 0 – запрещено 1 – разрешено
19	MSINT	R/W	Управление разрешением прерывания от флага MSINT 0 – запрещено 1 – разрешено
20	UDINT	R/W	Управление разрешением прерывания от флага UDINT 0 – запрещено 1 – разрешено
21	UTXEINT	R/W	Управление разрешением прерывания от флага UTXEINT 0 – запрещено 1 – разрешено
22	URXTINT	R/W	Управление разрешением прерывания от флага URXTINT 0 – запрещено 1 – разрешено
30:23	RSV	-	Не используется
31	LBM	R/W	Тестовый режим работы интерфейса при котором выход передатчика коммутируется на вход приемника (когда ==1)

Бит UARTEN разрешает работу интерфейса. Он используется одновременно с битами UTXDIS и URXDIS, которые индивидуально управляют передатчиком и приемником соответственно.

Бит DMAONERR позволяет управлять интерфейсом во время его работы с контроллером DMA. В случае если при приеме данных возникнет ошибка, то при установленном бите DMAONERR произойдет блокировка запроса к контроллеру DMA, что исключит прием ошибочных данных.

Бит UTXFDIS используется только для управления выключением FIFO передатчика. При установленном бите FIFO выключен.

Бит URXFDIS используется только для управления выключением FIFO приемника. При установленном бите FIFO выключен.

Биты UHBRE определяют количество тактов частоты синхронизации (полученной после деления входной частоты блока UART на значение UBRCR) необходимых для приема (передачи) одного бита данных. Значение 00 задает режим, в котором длительность одного бита передаваемой (принимаемой) информации равна 16 периодам частоты внутреннего делителя. Этот режим обеспечивает надежный прием в условиях помех. Значения 01, 10, 11 задают режим, в котором длительность одного бита передаваемой (принимаемой) информации равна 4 периодам частоты внутреннего делителя. В случае значений 10 или 11, значение коэффициента внутреннего делителя частоты равно 1 вне зависимости от значения в регистре ***UBitRate***.

Таким образом, реальная скорость обмена интерфейса UART определяется как входная частота XTI, деленная на запрограммированный коэффициент в регистре ***UBitRate***, затем деленная на 16 или 4 в передатчике (приемнике) интерфейса. Максимальная скорость обмена составляет XTI /4.

Бит BREAK. Установка данного бита приводит к установке высокого уровня на линии TXD передатчика.

Бит PRTEN разрешает (если равен 1) автоматическую генерацию дополнительного бита паритета при передаче байта данных. При приеме, установка этого бита разрешает проверку паритета принятого байта данных и генерацию флага об ошибке.

Бит EVENPRT позволяет задать тип проверки: четное или нечетное количество единичных бит в посылке. Если этот бит равен нулю, дополнительный бит четности дополняет количество единичных бит в посылке до четного количества. К примеру, если передается значение 0x15, бит четности равен 1. Если передается значение 0x55, то бит четности равен 0. Соответственно, если значение бита EVENPRT равно единице, осуществляется дополнение до нечетного количества единичных бит.

Бит XSTOP позволяет задать количество Стоп-битов в одиночной посылке. Значение 0 соответствует одному стоп-биту, а значение 1 – двум стоп-битам. Этот бит используется только передатчиком. Приемник не проверяет количество принятых стоп-битов больше одного. Увеличение количества стоп-битов до двух иногда полезно в случаях, когда существует расхождение в скорости обмена передающей и принимающей сторон.

Бит UFIFOEN позволяет управлять буферами приемника и передатчика. Если этот бит, равен 0, флаги состояния буферов формируются так, как будто размер буфера равен одному байту. При установленном бите, размер буфера равен 8.

Биты WDRLEN позволяют запрограммировать количество реальных информационных бит в посылке. Таким образом, посылка может быть длиной от 7 бит (старт, 5 бит данных, стоп) до 12 бит (старт, 8 бит данных, бит четности и два стоповых).

17.5 Регистр состояния интерфейса UFLAG

Таблица 134 – Регистр UFLAG

Бит	Обозначение	Тип	Описание
0			
1			
2			
3	UBUSY	R	Передатчик занят Бит равен 1 во время передачи данных, включая стоп-биты
4	URXFE	R	Состояние буфера приемника. 0 – буфер приемника хранит данные 1 – буфер приемника пуст
5	UTXFF	R	Состояние буфера передатчика 0 – буфер передатчика может принять новые данные 1 – буфер передатчика заполнен полностью
7:6			

Бит UBUSY показывает состояние передатчика в текущий момент времени. Единичное значение бита указывает на процесс передачи информации. Флаг может быть использован для определения момента, когда передатчик полностью закончил работу.

Бит URXFE является признаком состояния FIFO приемника. Значение флага не зависит от того включено FIFO или нет. Единичное значение флага указывает на то, что буфер приемника пуст.

Бит UTXFF является признаком состояния FIFO передатчика. Значение флага зависит от того включено FIFO или нет. Если FIFO выключено, единичное значение флага указывает на то, что в буфере передатчика имеется байт данных для передачи. Если FIFO включено, единичное значение флага указывает на то, что буфер передатчика заполнен полностью.

17.6 Регистры управления прерываниями UINT, UINTM

Интерфейс имеет регистр состояния запросов прерывания UINT. Каждый бит регистра состояния имеет соответствующий бит разрешения прерывания. Каждый из интерфейсов системы имеет только один вход запроса прерывания в контроллере прерываний. Поэтому на программиста возлагается задача определения, какая именно ситуация вызвала запрос на прерывание.

Таблица 135 – Регистр UINT

Бит	Обозначение	Тип	Описание
0	TXINT	R	Запрос прерывания от передатчика
1	RXINT	R	Запрос прерывания от приемника
2	RXERRINT	R	Ошибка приема во время работы с контроллером DMA
3	MSINT	R/W	Запрос прерывания от модема
4	UDINT	R	Запрос прерывания от приемника в состоянии «выключен»
5	UTXEINT	R/W	Запрос прерывания от буфера передатчика при отсутствии в нем данных
6	URXTINT	R/W	Запрос прерывания от приемника при обнаружении ситуации “time-out”
7	RSV	-	Не используется

Бит TXINT – запрос прерывания от передатчика. Он отражает состояние FIFO передатчика. Если FIFO выключено, запрос на прерывание активен (равен 1) если буфер пуст. Если же FIFO включено, запрос на прерывание активен, если буфер заполнен на половину или менее.

Бит RXINT – запрос прерывания от приемника. Запрос может быть активен в следующих случаях: FIFO приемника выключено и в нем имеются принятые данные, FIFO приемника включено и заполнено принятыми данными на половину или больше, FIFO приемника содержит данные и не поступило новых данных за время более чем период, равный передаче 32 бит данных.

Бит RXERRINT отражает информацию о возможных ошибках при приеме данных. Наиболее эффективно данный бит может быть использован при работе интерфейса с контроллером DMA. В этом случае, если контроллер DMA прочитает данные, при приеме которых были обнаружены ошибки, то интерфейс автоматически выработает прерывание, что позволит процессору правильно обработать возникшую ситуацию. Данный бит автоматически блокирует генерацию запроса к контроллеру DMA на обслуживание (при разрешении в регистре управления).

Бит MSINT отражает запрос на прерывание от линий состояния модема. Изменение состояния любого из входов DCD, DSR, CTS вызывает установку бита MSINT.

Бит UDINT позволяет определить ситуацию, когда приемник интерфейса выключен, но на линии приемника RXD обнаружен старт бит. Это приводит к установке запроса на прерывание.

Бит UTXEINT активен в случае, когда интерфейс включен и буфер передатчика пуст.

Бит URXTINT отражает наличие ситуации в приемнике, когда нет приема за период равный передаче 32 бит данных. Этот флаг может быть использован для определения ситуации конца пакета данных.

Биты регистра состояния, помеченные как R/W, могут быть сброшены программно. Это осуществляется записью 1 в соответствующий бит регистра состояния. Запись нуля не изменяет значения бита.

Необходимо отметить следующую деталь – при чтении регистра UINTM происходит чтение значения регистра состояния, маскированное значением разрешения. Для чтения оригинального значения регистра состояния используется регистр UINT.

17.7 Программирование интерфейса UART

Перед началом работы с интерфейсом UART необходимо разрешить альтернативную функцию для соответствующих битов портов общего назначения. Необходимо заранее установить требуемую скорость обмена и значения соответствующих коэффициентов деления в конкретном интерфейсе. После программирования скорости обмена необходимо задать другие параметры обмена (длину посылки, контроль ошибок, ситуации прерываний) и включить интерфейс. Интерфейсы UART_0, UART_1 могут работать с контроллером DMA.

18 Последовательный синхронный интерфейс SPI

Интерфейс включает сигнал синхронизации, разрешения обмена, вход и выход данных. Он имеет гибкую систему программирования, позволяющую задать различные скорости обмена, длину посылки, а также различные ситуации прерываний. Интерфейс имеет встроенные FIFO глубиной 8 байт для приема и передачи. Возможности интерфейса позволяют ему работать с контроллером DMA.

Внешние выводы интерфейса SPI приведены в Таблице 136

Таблица 136 – Внешние выводы

Обозначение вывода (основное)	Обозначение вывода для интерфейса SPI	Тип вывода	Функциональное назначение
PA[4]	SPI0_CLK	I/O	Интерфейс SPI0. Синхросигнал
PA[5]	SPI0_DO	I/O	Интерфейс SPI0. Выход данных
PA[6]	SPI0_DI	I/O	Интерфейс SPI0. Вход данных
PA[7]	SPI0_CS[0]	I/O	Интерфейс SPI0. Выбор SPI устройства 0
PA[8]	SPI0_CS[1]	I/O	Интерфейс SPI0. Выбор SPI устройства 1
PA[9]	SPI0_CS[2]	I/O	Интерфейс SPI0. Выбор SPI устройства 2
PA[10]	SPI0_CS[3]	I/O	Интерфейс SPI0. Выбор SPI устройства 3
PA[11]	SPI0_CS[4]	I/O	Интерфейс SPI0. Выбор SPI устройства 4
PA[12]	SPI0_CS[5]	I/O	Интерфейс SPI0. Выбор SPI устройства 5
PA[13]	SPI1_CLK	I/O	Интерфейс SPI1. Синхросигнал
PA[14]	SPI1_DO	I/O	Интерфейс SPI1. Выход данных
PA[15]	SPI1_DI	I/O	Интерфейс SPI1. Вход данных
PA[16]	SPI1_CS	I/O	Интерфейс SPI1. Выбор SPI устройства
PB[0]	SPI2_CLK	I/O	Интерфейс SPI2. Синхросигнал
PB[1]	SPI2_DO	I/O	Интерфейс SPI2. Выход данных
PB[2]	SPI2_DI	I/O	Интерфейс SPI2. Вход данных
PB[3]	SPI2_CS	I/O	Интерфейс SPI2. Выбор SPI устройства

Для обмена посредством интерфейса SPI используются четыре линии: синхронизация, вход данных, выход данных, разрешение обмена. В ряде случаев используется еще дополнительная линия выбора мастера-подчиненного, однако, в данной реализации такой выбор осуществляется посредством программирования регистров управления. Направление линий интерфейса, в зависимости от выбранного режима работы, приведено в Таблице 137.

Таблица 137 – Режимы работы интерфейса

	SPI (Ведущий)	SPI (Ведомый)
SPI_CLK	Выход	Вход
SPI_DO	Выход	Вход
SPI_DI	Вход	Выход
SPI_EN	Выход	Вход

Интерфейс доступен для программирования как набор регистров, позволяющих задать нужную конфигурацию и осуществить прием-передачу данных. Регистры интерфейса подключены к шине обмена периферийных устройств. В микросхеме реализовано три независимых блока SPI, характеристики и базовые адреса которых приведены в Таблице 138

Таблица 138 – Описание

Блок	Базовый адрес регистра	Описание
SPI0	0x80000140	До шести одновременно адресуемых устройств в режиме «Ведущий», одно ведущее устройство в режиме «Ведомый»
SPI1	0x80000360	Одно устройство в режимах «Ведущий», «Ведомый»
SPI2	0x80000380	Одно устройство в режимах «Ведущий», «Ведомый»

Краткая информация о регистрах приведена в Таблице 139.

Таблица 139 – Регистры интерфейса

Номер	Имя	Назначение	По сбросу
0	SPCR0	Регистр управления 0	0
1	SPCR1	Регистр управления 1	0
2	SPDR	Регистр данных	-
3	SPSR	Регистр состояния	0
4	RX_CNT	Счетчик приемника	0

Алгоритм работы интерфейса предполагает одновременный прием и передачу данных. Интерфейс имеет встроенные буфера (FIFO) для приема и передачи данных. Размер каждого из буферов составляет восемь 32-разрядных слов. Для того чтобы осуществить обмен посредством интерфейса, необходимо:

- запрограммировать параметры обмена в регистрах управления;
- записать данные для передачи в буфер передатчика;
- прочитать принятые данные из буфера приемника.

Если необходимо только передать данные в какое-то устройство, интерфейс все равно будет принимать данные с входной линии данных. Поэтому после окончания передачи необходимо сбросить флаги состояния приемника.

Если необходимо только принять информацию, в этом случае нужно записать в буфер передатчика фиктивные данные для передачи. В этом случае количество принятых данных будет соответствовать количеству переданных.

Интерфейс SPI0 имеет возможность подключать до 6-ти внешних устройств. Каждое из устройств имеет общие линии синхронизации и данных, но индивидуальную линию выбора устройства. Номер устройства задается в регистре управления и должен быть неизменным на весь период обмена. Имеется возможность индивидуального задания неактивного уровня для каждого устройства.

Интерфейсы SPI1, SPI2 имеют возможность работы только с одним устройством, во всем остальном абсолютно идентичны интерфейсу SPI0.

Если возникает необходимость работы интерфейса в режиме «подчиненный», ведущее устройство должно осуществлять выборку интерфейса посредством входа CS1. Также если выбран вариант начальной загрузки с использованием SPI-флэш памяти, данная память должна быть подключена к CS0.

Практически количество подключаемых внешних SPI-устройств может быть и больше чем 6. В этом случае в качестве сигнала выбора устройства можно использовать любой из свободных выходов общего назначения, а в интерфейсе при осуществлении обмена задавать номер устройства 7.

Интерфейс имеет два 32-разрядные регистра управления, позволяющие задать параметры работы интерфейса. Ниже, в таблицах, приведено краткое описание разрядов регистров управления.

18.1 Регистр SPCR0

Таблица 140 – Регистр управления SPCR0

Бит	Имя	Тип	Назначение
4:0	DSS	R/W	Выбор размера данных 0-2 - Не используется 3-31 – длина данных (DSS+1)
5	-		
6	SPO	R/W	Полярность SPI_CLK 0 – SPI_CLK высокий импульс 1 – SPI_CLK низкий импульс
7	SPH	R/W	Фаза SPI_CLK 0 – SPI_CLK активен на заднем фронте 1 – SPI_CLK активен на переднем фронте
8	TWI_on	R/W	0 – стандартный SPI интерфейс 1 – 3-х проводной интерфейс с двунаправленной линией
9	TWI_rw	R/W	Чтение(0) или запись(1) в режиме TWI
10	LMSF	R/W	Выбор 0 – старший бит первый 1 – младший бит первый
11	-		
14:12	CSN		Выбор номера SPI-устройства
15	-		
27:16	SCR	R/W	Коэффициент деления частоты Скорость обмена = SOC_CLK/ (SCR + 1)
31:27	-		

Биты DSS задают количество передаваемых бит в одной посылке. Максимальное количество бит в одной посылке равно 32. Если для работы интерфейса выбрана меньшая разрядность передаваемых данных (от 4 до 32), для передачи используются только младшие значащие биты. Так, если размер одиночной посылки равен 8 бит, старшие биты (с 8 по 32) слова не будут передаваться.

Биты SPO и SPH управляют полярностью и фазой синхросигнала SPI_CLK. Ниже показаны возможные варианты обмена при передаче 4-х бит данных (Рисунок 85).

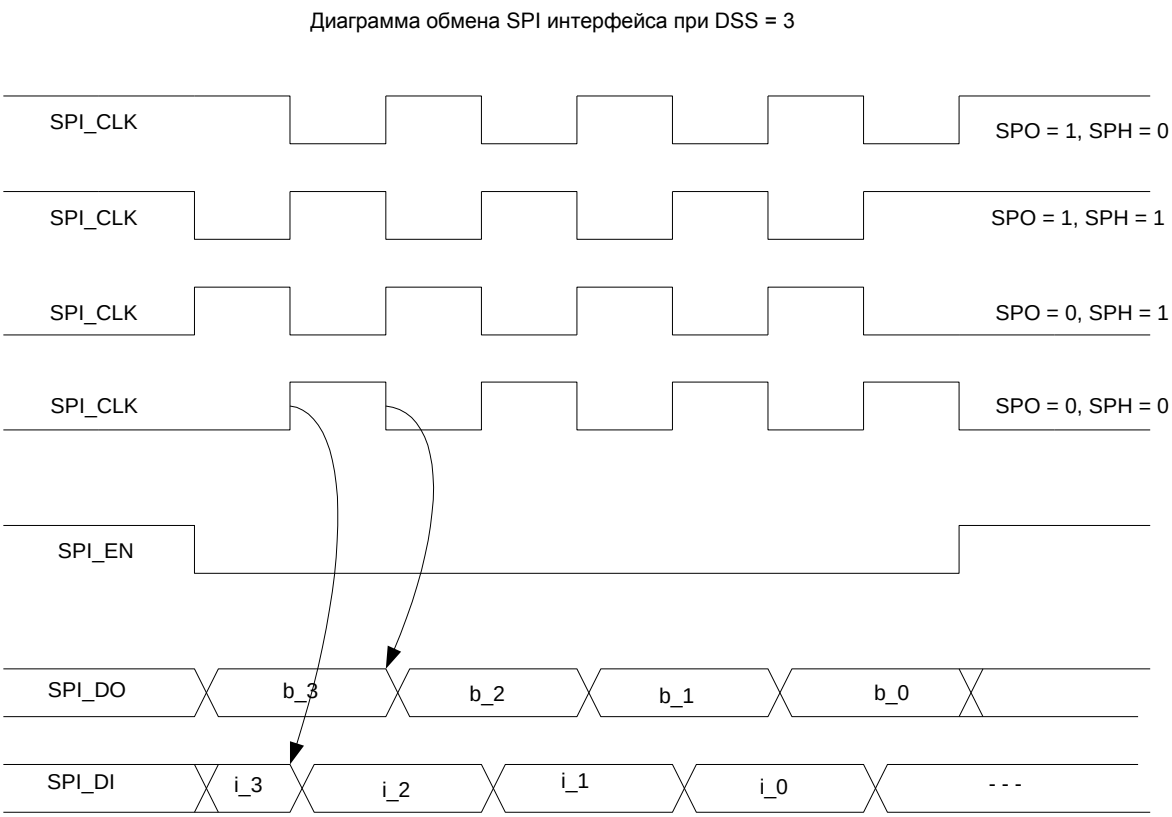


Рисунок 85 – Диаграмма обмена в режиме master

Биты SCR задают коэффициент деления для внутреннего делителя частоты. Входная частота синхронизации модуля SPI поступает с периферийной шины, затем она может дополнительно делиться на запрограммированный коэффициент деления согласно формуле:

$$SPI\_CLK = SOC\_CLK \div (SCR + 1).$$

18.2 Регистр SPCR1

Таблица 141 – Регистр управления SPCR1

Бит	Имя	Тип	Назначение
0	RIM	R/W	Управление прерыванием от приемника интерфейса 0 – запрещено 1 – разрешено
1	TIM	R/W	Управление прерыванием от передатчика интерфейс 0 – запрещено 1 – разрешено
2	LBM	R/W	Разрешение тестового режима работы 0 – обычный режим работы 1 – тестовый режим работы
3	SPE	R/W	Разрешение работы интерфейса 0 – выключен 1 – включен
4	MS	R/W	Режим работы 0 – мастер 1 – подчиненный
5	TUM	R/W	Разрешение прерывания по исчерпанию буфера передатчика при работе в режиме подчиненного

			0 – запрещено 1 – разрешено
6	ROM	R/W	Разрешение прерывания при переполнении буфера приемника 0 – запрещено 1 – разрешено
7	-	R/W	
8	R_RQM	R/W	Запрос на обслуживание со стороны FIFO приемника: 0 – FIFO загружено на половину или более 1 – в FIFO имеются данные
9	T_RQM	R/W	Запрос на обслуживание со стороны FIFO передатчика: 0 – FIFO загружено на половину или менее 1 – FIFO не заполнено полностью
10	TXO	R/W	Только передача 0 – одновременный прием-передача 1 – только передача
11	RXO	R/W	Только прием 0 – одновременный прием-передача 1 – только прием
12	CIM	R/W	Разрешение прерывания от счетчика приемника 0 – запрещено 1 – разрешено
13			
14	HOLDCS	R/W	Удержание линии CS активной после завершения обмена 0 – после обмена линия становится неактивной (работает совместно с полем DLY_T). 1 – после очередного обмена линия остается активной.
15	ROTL		Выходной уровень на выходе данных в случае работы в режиме «только прием»
21:16	CSAL	R/W	Выбор активного уровня на выходе CSi 0 – активный низкий 1 – активный высокий
23:22	-		Не используются
27:24	DLY_T	R/W	Длительность неактивного уровня CS (при HOLDCS=0) 0 – неактивного уровня нет Остальные значения – DLY_T тактов синхросигнала SPI
31:28	-		Не используются

Бит SPE разрешает работу интерфейса. Если его значение равно нулю – интерфейс выключен. Для включения интерфейса необходимо записать в данные бит значение 1.

Бит MS при включенном интерфейсе определяет режим работы интерфейса – мастер или подчиненный. В случае работы в режиме мастер (MS = 0) интерфейс самостоятельно вырабатывает сигнал синхронизации и сигнал разрешения обмена. В случае работы в режиме «подчиненный» (MS = 1), интерфейс работает под управлением внешнего синхросигнала и от внешнего сигнала разрешения работы. При работе в режиме подчиненный имеются некоторые ограничения на максимальную частоту обмена. Она должна быть как минимум в три раза ниже частоты обмена по внутренней шине периферийных устройств.

Бит RIM управляет разрешением прерывания от приемника интерфейса. Момент генерации прерывания зависит от бита R_RQM. Если бит R_RQM равен 1, то прерывание будет сформировано, если в FIFO приемника есть данные. Если бит R_RQM равен 0, то прерывание будет сформировано, если в FIFO приемника есть 4 или более слова данных, либо если в количество данных меньше 4-х, но прием данных закончен. Факт окончания приема определяется по факту отсутствия данных в буфере передатчика и отсутствии в данный момент обмена.

Бит TIM управляет разрешением прерывания от передатчика интерфейса. Момент генерации прерывания зависит от бита T_RQM. Если бит T_RQM равен 1, то прерывание будет сформировано, если FIFO передатчика заполнено не полностью. Если бит T_RQM равен 0, то прерывание будет сформировано, если в FIFO передатчика есть 4 или более свободные ячейки для записи данных.

Бит LBM позволяет перевести интерфейс в тестовый режим работы (LBM = 1), в котором выход данных передатчика подается на вход данных приемника. Это позволяет произвести проверку оборудования без подключения внешнего устройства. Для нормального режима работы необходимо устанавливать LBM = 0.

Бит TXO позволяет, при необходимости, упростить передачу информации. Как уже отмечалось выше, при обычной работе интерфейс при передаче данных осуществляет одновременный прием информации с линии входа данных. Порой эти данные не нужны, т.к. требуется выполнить только передачу. Но буфер приемника все равно заполняется. Чтобы исключить это неудобство был добавлен бит TXO, который при значении 1 выполняет блокировку приема данных и не изменяет состояние приемника.

Бит RXO позволяет, при необходимости, упростить прием информации. Как уже отмечалось выше, при обычной работе интерфейс для приема данных должен осуществить фиктивную передачу информации. Чтобы исключить это неудобство был добавлен бит RXO, который при значении 1 не требует наличия фиктивных данных в буфере передатчика при приеме данных. Для обеспечения функции «только прием» был добавлен дополнительный регистр RX_CNT.

Бит CIM разрешает генерацию запроса прерывания при значении счетчика RX_CNT равным нулю.

Бит CSAL позволяет задать активный уровень на выходе CS. Одни устройства могут активизироваться высоким уровнем, а другие – низким. Данный бит дает возможность задать активный уровень.

Бит HOLDCS позволяет осуществить удержание вывода CS в активном состоянии после завершения обмена. Бит должен быть установлен до начала обмена. После такого, как с началом обмена линия CS активизируется, она будет оставаться активной до тех пор, пока бит HOLDCS не будет сброшен. Сбросить бит можно во время осуществления последнего обмена или после его завершения. Использование данной функции позволяет осуществить передачи или прием посылки произвольной длительности.

18.3 Регистр счета принимаемых данных RX_CNT

Регистр используется только при установленным бите RXO==1, что соответствует режиму «только прием». В этом случае в регистр RX_CNT необходимо записать количество принимаемых слов данных (не более 4095). Если значение регистра не равно нулю, это автоматически формируется запрос на прием данных. После приема очередного слова осуществляется вычитание 1 из счетчика. Возможна генерация прерывания от счетчика в случае достижения им нулевого значения. Это происходит при установленном бите CIM.

18.4 Регистр данных SPDR

Регистр данных имеет разрядность 32 бит. При записи, данные попадают в верхний свободный регистр FIFO передатчика, а при чтении, считываются из нижнего регистра FIFO приемника. Размер FIFO приемника и передатчика составляет по восемь 32-разрядных слов каждый. Каждый из FIFO имеет указатели считывания и записи. Интерфейс имеет несколько флагов, которые отражают

состояние FIFO передатчика и приемника. Интерфейс подключен к 32-разрядной шине периферийных устройств.

18.5 Регистр состояния SPSR

Таблица 142 – Регистр состояния SPSR

Бит	Имя	Тип	Назначение
0	TFE	R	Флаг заполнения FIFO передатчика («пустой») 0 – в FIFO передатчика данные имеются 1 – в FIFO передатчика данные отсутствуют
1	TNF	R	Флаг заполнения FIFO передатчика («полный») 0 - FIFO передатчика заполнено полностью 1 – FIFO передатчика не заполнено
2	RNE	R	Флаг пустого FIFO приемника («не пустой») 0 – нет данных в FIFO приемника 1 – в FIFO приемника имеются данные
3	BSY	R	Занятость SPI интерфейса 0 – простаивает или выключен 1 – занят передачей-приемом
4	TFS	R	Флаг состояния FIFO передатчика 0 – буфер передатчика более чем на половину заполнен (5 или более слов записано) 1 – буфер передатчика заполнен на половину или менее (4 или меньше слов записано).
5	RFS	R	Флаг состояния FIFO приемника 0 – буфер приемника заполнен меньше чем на половину (3 или меньше слов записано) 1 – буфер приемника заполнен на половину или более (4 или больше слов записано).
6	ROR	R/W	Флаг переполнения FIFO приемника 0 – нет ошибки 1 – во время приема произошло переполнение буфера.
7	TUR	R/W	Флаг состояния FIFO передатчика (только в режиме «подчиненный») («чтение пустого буфера») 0 – нет ошибки 1 – передатчик пытался передать данные из пустого буфера
8	RFF	R	Флаг состояния FIFO приемника («полный») 0 – буфер приемника заполнен не полностью 1 – буфер приемника заполнен полностью
9	TI_REQ	R	Запрос передатчика к процессору или ПДП 0 – неактивный уровень 1 – активный уровень
10	RI_REQ	R	Запрос приемника к процессору или ПДП 0 – неактивный уровень 1 – активный уровень
31:11	RSV	-	Не используются

Биты регистра SPSR отражают состояние интерфейса во время приема-передачи данных.

Бит TFE отражает наличие данных в буфере передатчика. Значение бита 1 соответствует ситуации, когда буфер пуст. Бит может быть использован для отслеживания ситуации, когда интерфейс закончил передачу.

Бит TNF может быть использован при заполнении буфера передатчика данными. Значение бита TNF=1 означает, что буфер может принять новые данные. Значение бита равное нулю означает что, буфер заполнен полностью.

Бит RNE отражает состояние буфера приемника. Значение бита RNE=1, означает наличие данных в буфере приемника, которые могут быть прочитаны.

Бит BSY отражает занятость интерфейса приемом\передачей данных. Бит может быть использован для определения момента полного завершения работы интерфейса. Так отсутствие данных в FIFO передатчика и равенство бита BSY нулю, означает завершение работы интерфейса.

Бит TFS отражает состояние буфера передатчика. Равенство бита единице означает, что FIFO передатчика заполнено на половину или менее (т.е. в FIFO записано от 1 до 4-х слов).

Бит RFS отражает состояние буфера приемника. Равенство бита единице означает, что FIFO приемника заполнено на половину или более (т.е. в FIFO записано от 4 до 8 слов).

Бит ROR отслеживает ситуацию переполнения FIFO приемника, т.е. случая, когда в буфер приемника была произведена попытка записи новых данных в то время, когда он был заполнен. Бит может вызвать запрос прерывания процессору, если соответствующий ему бит ROM регистра управления установлен.

Бит TUR имеет смысл при работе интерфейса в режиме подчиненного и отслеживает ситуацию, когда при запросе обмена со стороны мастера, буфер передатчика пуст. При работе в режиме подчиненного обмен инициирует внешний мастер, а данные должны быть загружены в буфер предварительно. Бит может вызвать запрос прерывания процессору, если соответствующий ему бит TUM регистра управления установлен.

Биты ROR и TUR устанавливаются аппаратно, но могут быть сброшены программно. Для очистки бита необходимо произвести запись в соответствующий бит значения 1.

Бит RFF отслеживает ситуацию, когда буфер приемника заполнен полностью.

18.6 Особенности работы в режиме “slave”

Интерфейс имеет возможность работы в режиме подчиненного (“slave”) устройства. Данный режим задается при помощи установки в 1 бита MS регистра SPCR1. В данном режиме интерфейс использует вход SPI_CLK (PA[4]) как вход синхронизации, вход SPI_CS[1] (PA[8]) как вход выборки интерфейса, вход SPI_DO (PA[5]) как вход данных для приема информации и выход SPI_DI (PA[6]) для передачи данных ведущему устройству. В отличие от режима мастера, все внешние контакты интерфейса меняют направление на противоположное. Работа в режиме подчиненного устройства имеет ряд ограничений. Вход синхронизации SPI_CLK используется, так как он есть на внешнем выводе, т.е. нет никаких преобразований полярности и фазы сигнала синхронизации. Выдача данных на SPI_DI осуществляется только по отрицательному (из 1 в 0) фронту SPI_CLK. Прием данных с SPI_DO выполняется только по положительному (из 0 в 1) фронту SPI_CLK. Пример временной диаграммы представлен ниже (Рисунок 86).

Диаграмма обмена в режиме "slave" при DSS = 3

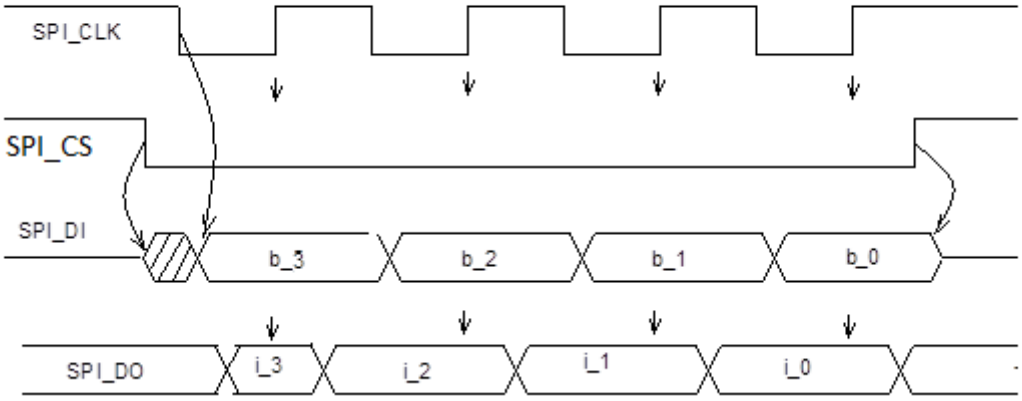


Рисунок 86 – Диаграмма обмена в режиме slave

Как следует из рисунка, выход SPI_DI становится активным, как только на входе SPI_CS появляется активный уровень. Активный уровень на входе SPI_CS может программироваться. Если бит CSAL[1] равен нулю, активным уровнем на входе SPI_CS будет низкий уровень. Если CSAL[1] равен единице – активный уровень высокий. Однако установка активного уровня на SPI_CS не означает, что на выходе SPI_DI сразу же появляются истинные данные. Только после первого отрицательного фронта на SPI_CLK на выходе SPI_DI появится достоверная информация (бит b_3). Ведущее устройство должно принять данные с линии SPI_DI по положительному фронту SPI_CLK. В этот же момент и подчиненное устройство примет данные с линии SPI_DO.

В момент приема бита b_0 подчиненное устройство зафиксирует окончание приема посылки и передаст принятые данные в FIFO приемника. При передаче данных в FIFO будет выполнена передача из домена синхронизации SPI_CLK в домен синхронизации SOC_CLK. Для корректной работы интерфейса необходимо чтобы частота СОК-шины была как минимум в 3 раза выше частоты SPI_CLK.

Поскольку момент начала передачи по интерфейсу в режиме подчиненного является случайным, данные для передачи должны быть записаны до момента начала передачи. В противном случае будет сформирован флаг ошибки. Моментом захвата данных из FIFO передатчика является первый положительный фронт сигнала SPI_CLK. К этому моменту данные для передачи должны быть записаны в FIFO. Сброс указателя на взятые из FIFO данные выполняется немного позже. Возможна ситуация, когда данные захвачены некорректно, но флаг ошибки не установится. Это может случиться, если одновременно с записью данных в FIFO приходит первый положительный фронт SPI_CLK.

Прием-передача данных в подчиненном устройстве выполняется в сдвиговом регистре тактируемым SPI_CLK. Таким образом, для обеспечения высокой скорости обмена желательно использовать большую длину посылки. В этом случае у процессора будет больше времени для обслуживания одной порции принятых данных.

19 Контроллер видеокамеры

Данный контроллер позволяет организовать прием информации от внешней видеокамеры или иного устройства, поддерживающего заданный интерфейс.

Интерфейс видеокамеры представляет собой контроллер, подключенный к шине периферийных устройств и способный принимать данные от внешней цифровой видеокамеры (или другого устройства) согласно заданному протоколу обмена. Принятые данные помещаются в буфер и могут быть переданы в запоминающие устройства процессора с помощью контроллера DMA или с помощью процессора.

Внешние выводы контроллера видеокамеры приведены в Таблице 143.

Таблица 143 – Внешние выводы контроллера

Обозначение вывода (основное)	Обозначение вывода контроллера	Тип вывода	Функциональное назначение
PA[25]	VC_CLK	I/O	Интерфейс видеокамеры. Синхросигнал
PA[26]	VC_VSYNC	I/O	Интерфейс видеокамеры. Вход вертикальной развертки
PA[27]	VC_HSYNC	I/O	Интерфейс видеокамеры. Вход горизонтальной развертки видеоинтерфейса
PA[28]	VC_D[0]	I/O	Интерфейс видеокамеры. Бит 0 данных
PA[29]	VC_D[1]	I/O	Интерфейс видеокамеры. Бит 1 данных
PA[30]	VC_D[2]	I/O	Интерфейс видеокамеры. Бит 2 данных
PA[31]	VC_D[3]	I/O	Интерфейс видеокамеры. Бит 3 данных
PB[0]	VC_D[4]	I/O	Интерфейс видеокамеры. Бит 4 данных
PB[1]	VC_D[5]	I/O	Интерфейс видеокамеры. Бит 5 данных
PB[2]	VC_D[6]	I/O	Интерфейс видеокамеры. Бит 6 данных
PB[3]	VC_D[7]	I/O	Интерфейс видеокамеры. Бит 7 данных

19.1 Регистры интерфейса

Регистры интерфейса подключены к шине обмена периферийных устройств и имеют базовый адрес **0x800001A0**. Краткое описание регистров приведено в Таблице 144.

Таблица 144 – Регистры интерфейса

Номер	Название	Назначение
0	VC_DR	128-разрядный регистр данных. Доступен только по чтению и является выходом FIFO
4	VC_SR	Регистр состояния
5	VC_CR	Регистр управления

19.1.1 Регистр управления CR

Таблица 145 – Регистр управления

Бит	Имя	Тип	Назначение
0	VCON	R/W	Включение интерфейса. При записи 1 интерфейс начинает прием данных во внутреннее FIFO.
1	VCIE	R/W	Разрешение прерывания. При записи 1 интерфейс вырабатывает прерывание если во внутреннем буфере имеются данные

Бит	Имя	Тип	Назначение
2	SMODE	R/W	Режим приема информации: 0 – видеокамера 1 – ведущее устройство
3	VPOL	R/W	Полярность сигнала VSYNC 0 – активные данные передаются при уровне VSYNC = 0 1 – активные данные передаются при уровне VSYNC = 1

19.1.2 Регистр состояния SR

Таблица 146 – Регистр состояния

Биты	имя	Описание
0	EMPTY	1 – нет данных в буфере 0 – имеются данные
1	FULL	0 – буфер заполнен не полностью 1 – буфер заполнен полностью
2	OVERF	1 – было переполнение буфера во время приема данных 0 – не было переполнения
3	UNDERF	1 – при чтении данных произошло чтение пустого буфера

19.1.3 Регистр данных DR

Посредством этого регистра происходит чтение данных после приема. Регистр имеет разрядность 128 бит и поэтому обращение к нему должно выполняться как к квадрослову. Размер буфера приема данных равен четырем 128-разрядным словам.

Интерфейс имеет внешние входы:
- DCLK – вход синхронизации;
- VSYNC – строб начала картины;
- HSYNC – строб начала линии;
- HD[7:0] – шина данных.

19.2 Режим приема «видеокамера»

Интерфейс является синхронным и осуществляет прием данных по положительному фронту сигнала синхронизации DCLK. Для начала приема информации, контроллер должен быть включен путем установки бита VCON регистра управления. После включения интерфейс начинает отслеживать стартовую ситуацию для начала приема данных. В случае режима видеокамеры для начала приема данных необходимо наличие на входе VSYNC высокого уровня. Этот факт запоминается во внутреннем флаге. В момент, когда оба входа VSYNC и HSYNC станут равным нулю (не обязательно одновременно как на рисунке), будет установлен внутренний флаг camera_ON, означающий возможность приема данных. Таким образом, к моменту прихода нового кадра (VREF активный), интерфейс готов к приему данных. Временная диаграмма для случая VPOL = 1 приведена на Рисунке 87.

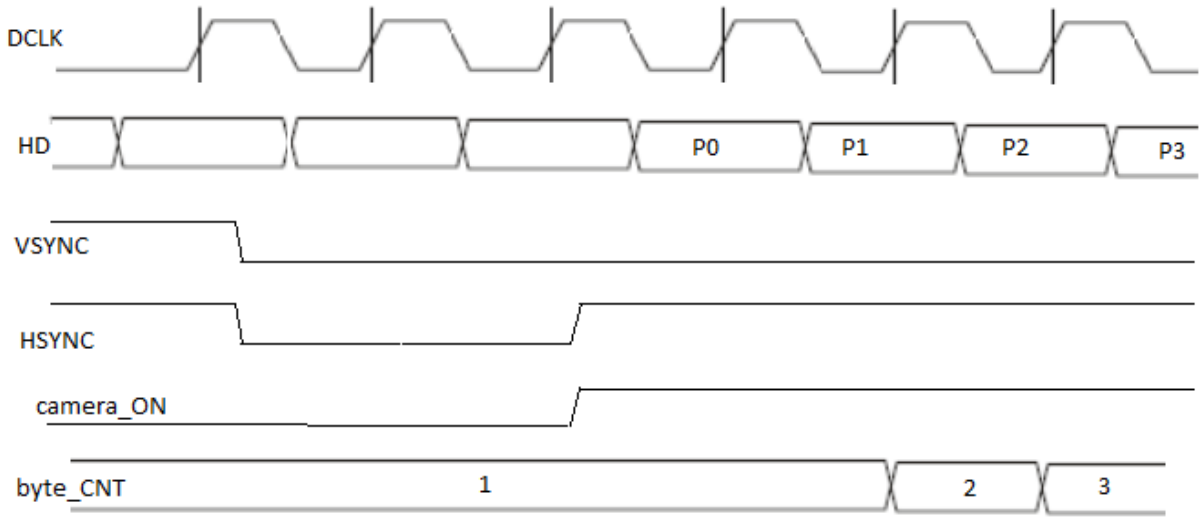


Рисунок 87 – Начало приема данных в режиме «видеокамера»

Прием данных начнется в момент, когда внутренний флаг camera_ON будет установлен и на входе HSYNC будет высокий уровень, что означает наличие данных горизонтальной строки на входе контроллера. На временной диаграмме видно расположение позиции первого принимаемого байта (P0). Каждый раз, когда HSYNC равен 1 и включен прием, происходит наращивание счетчика байт byte_CNT. Блок данных из 16 байт формирует квадрослово данных. Когда счетчик данных достигает значения 16 – квадрослово данных принято, формируется флаг готовности и принятая информация записывается в выходное FIFO. Счетчик опять начинает счет сначала. Размер выходного буфера равен 4 квадрословам. При перезаписи данных в выходное FIFO выполняется передача информации из домена синхронизации DCLK в домен синхронизации SOC-шины. Как только в выходном FIFO появляются данные, формируется запрос к контроллеру DMA на обслуживание. К DMA может прочитать данные из буфера интерфейса и переслать во внутреннюю или внешнюю память (в соответствии с настройками канала DMA). В канале контроллера DMA происходит отслеживание объема переданной информации и при необходимости может быть сформирован запрос на прерывание для анализа принятой информации. Если контроллер DMA не успевает принимать поток данных от интерфейса, возможно переполнение буфера. В этом случае будет установлен соответствующий флаг в регистре состояния.

Интерфейс устроен так, что после приема кадра необходимо выключить интерфейс и затем для принятия нового кадра опять включить его.

19.3 Режим приема «ведущее устройство»

К внешним контактам интерфейса видеокамеры может быть подключено некоторое ведущее устройство, имеющее стандартный интерфейс микропроцессорной шины (адрес, данные, сигналы выборки и записи), с целью передачи информации по определенному протоколу. Интерфейс видеокамеры поддерживает простой вариант обмена, удобный для такого подключения. Данный вариант включается при установке бита SMOD в 1. В этом случае возможно следующее подключение сигналов:

- DCLK как сигнал выборки интерфейса nVCS;
- VSYNC как сигнал адресной шины A[1];
- HSYNC как сигнал адресной шины A[0];
- HD[7:0] как младший байт шины данных D[7:0].

Сигнал выборки nVCS должен быть сформирован как логическое «или» сигналов выборки и записи, т.е. nVCS = nCS or nWE (активный уровень низкий). Дополнительно к сигналу выборки может подключаться дешифратор адреса для дополнительного выбора устройства. При таком подключении обычный цикл записи по указанному адресу вызывает загрузку байта данных в интерфейс контроллера. Пример временной диаграммы приведен ниже (Рисунок 88).

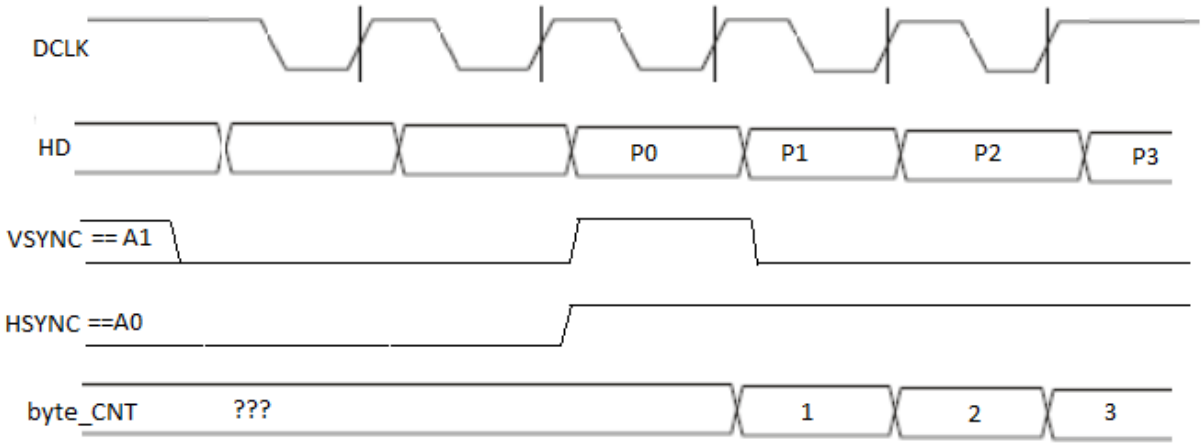


Рисунок 88 – Начало приема данных в режиме «ведущее устройство»

Для ведущего устройства передача данных в интерфейс выглядит как обычный цикл записи по адресам 0, 1, 2, 3. При этом:

- в интерфейс передаются только 8 бит данных;
- запись по адресу 0 выглядит как подача одного такта синхросигнала DCLK без выполнения каких-либо действий внутри интерфейса;
- запись по адресу 3 вызывает прием байта данных и установку счетчика принятых байт в 1. Это позволяет ведущему устройству определить начальное значение счетчика;
- запись по адресу 1 приводит к записи следующего байта и автоматическому увеличению счетчика принятых байт на 1.

После приема 16 байт выполняется передача квадрослова данных в выходное FIFO, и формируется запрос к контроллеру DMA или прерывание к процессору. Если передаваемый блок данных не кратен 16 байт, его необходимо дополнить недостающими байтами. При формировании сигнала записи (DCLK) необходимо обеспечить требуемое время предустановки входных сигналов и время удержания относительно положительного фронта строба записи.

Таким образом, в режиме «ведущее устройство», интерфейс видеокамеры может рассматриваться как ведомый порт для приема информации от ведущего устройства. В зависимости от скорости передачи информации для обслуживания интерфейса может использоваться контроллер DMA или процессор.

20 Интерфейс NAND флэш-памяти

К процессору может быть подключена внешняя NAND флэш-память.

Внешние выводы интерфейса NAND флэш-памяти приведены в Таблице 147.

Таблица 147 – Внешние выводы

Обозначение вывода (основное)	Обозначение вывода для интерфейса NAND	Тип вывода	Функциональное назначение
PA[2]	NF_RE	I/O	Интерфейс NAND Flash. Строб чтения
PA[3]	NF_WE	I/O	Интерфейс NAND Flash. Строб записи
PA[4]	NF_ALE	I/O	Интерфейс NAND Flash. Подсветка адреса
PA[5]	NF_D[0]	I/O	Интерфейс NAND Flash. Бит данных 0
PA[6]	NF_D[1]	I/O	Интерфейс NAND Flash. Бит данных 1
PA[8]	NF_CS[1]	I/O	Интерфейс NAND Flash. Выбор модуля 1
PA[9]	NF_D[2]	I/O	Интерфейс NAND Flash. Бит данных 2
PA[10]	NF_D[3]	I/O	Интерфейс NAND Flash. Бит данных 3
PA[11]	NF_D[4]	I/O	Интерфейс NAND Flash. Бит данных 4
PA[12]	NF_D[5]	I/O	Интерфейс NAND Flash. Бит данных 5
PA[13]	NF_CLE	I/O	Интерфейс NAND Flash. Подсветка команды
PA[14]	NF_D[6]	I/O	Интерфейс NAND Flash. Бит данных 6
PA[15]	NF_D[7]	I/O	Интерфейс NAND Flash. Бит данных 7
PA[17]	NF_CS[0]	I/O	Интерфейс NAND Flash. Выборка модуля 0
PA[18]	NF_RDY	I/O	Интерфейс NAND Flash. Вход готовности
PD[16]	NF_D[0]	I/O	Интерфейс NAND Flash. Бит данных 0
PD[17]	NF_D[1]	I/O	Интерфейс NAND Flash. Бит данных 1
PD[18]	NF_D[2]	I/O	Интерфейс NAND Flash. Бит данных 2
PD[19]	NF_D[3]	I/O	Интерфейс NAND Flash. Бит данных 3
PD[20]	NF_D[4]	I/O	Интерфейс NAND Flash. Бит данных 4
PD[21]	NF_D[5]	I/O	Интерфейс NAND Flash. Бит данных 5
PD[22]	NF_D[6]	I/O	Интерфейс NAND Flash. Бит данных 6
PD[23]	NF_D[7]	I/O	Интерфейс NAND Flash. Бит данных 7
PD[24]	NF_CLE	I/O	Интерфейс NAND Flash. Строб записи
PD[25]	NF_ALE	I/O	Интерфейс NAND Flash. Строб чтения
PD[26]	NF_RE	I/O	Интерфейс NAND Flash. Подсветка адреса
PD[27]	NF_WE	I/O	Интерфейс NAND Flash. Подсветка команды
PD[28]	NF_CS[0]	I/O	Интерфейс NAND Flash. Выборка модуля 0
PD[29]	NF_CS[1]	I/O	Интерфейс NAND Flash. Выборка модуля 1
PD[30]	NF_CS[2]	I/O	Интерфейс NAND Flash. Выборка модуля 2
PD[31]	NF_RDY	I/O	Интерфейс NAND Flash. Вход готовности

Специальный контроллер позволяет осуществить преобразование запроса процессора на запись или чтение данных в последовательность команд обращения к флэш-памяти. Имеется поддержка коррекции ошибок.

При использовании данного контроллера разрядность шины данных контроллера внешней памяти уменьшается до 16 бит. Процессор не имеет прямого доступа во флэш-память, и все операции осуществляются посредством программирования операций контроллера.

Возможно подключение до трех внешних банков памяти. Выбор конкретного банка определяется разрядами 31:30 регистра адреса AR:

- 00 – NF_CS 0;
- 01 – NF_CS 1;
- 10 – NF_CS 2.

Биты AR[29:0] позволяют прямо адресовать 4 Гбайт каждого банка памяти. Если подключаемая память имеет больший объем, можно использовать биты расширения ADDRH[2:0] регистра NAND_CFG. Это позволяет адресовать до 32 Гбайт каждого банка флэш-памяти. Чтение и запись в NAND флэш-память имеет свои особенности. При описании протокола обмена с флэш-памятью, будет подразумеваться работа только(!) с флэш-памятью фирмы Samsung, а также памятью с аналогичным интерфейсом от других компаний.

Контроллер флэш-памяти имеет набор регистров, позволяющих задать параметры внешней подключенной памяти и режимы работы. Регистры контроллера доступны со стороны шины периферийных устройств и имеют базовый адрес 0x80000240.

С помощью регистров конфигурации пользователь может задать направление обмена (чтение или запись), начальный адрес обмена, количество передаваемых слов, а также размер слова (int, short, char). После задания необходимых параметров контроллер самостоятельно читает данные из внешней памяти во внутренний буфер (в случае чтения) или записывает данные из внутреннего буфера во внешнюю память (в случае записи). Для передачи данных между внутренним буфером контроллера и памятью системы, может использоваться процессор или контроллер прямого доступа. Ниже показано условно-графическое обозначение модуля контроллера флэш-памяти (Рисунок 89).

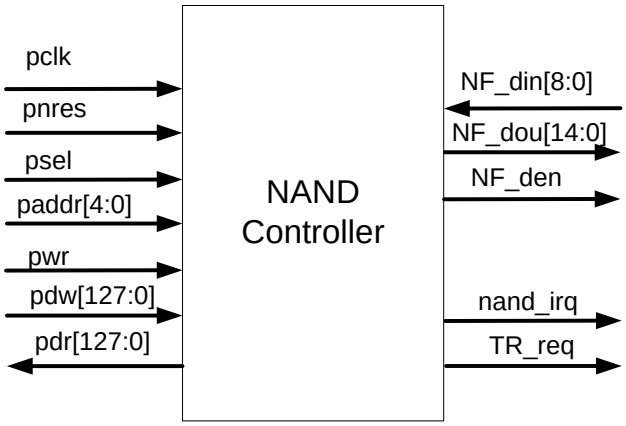


Рисунок 89 – Условное графическое обозначение контроллера

На рисунке показаны интерфейс подключения контроллера к шине периферийных устройств, посредством которого пользователь может осуществлять программирование операций обмена, а также интерфейс для взаимодействия с внешней флэш-памятью. Подробнее описание сигналов модуля приведено в Таблице 148.

Таблица 148 – Сигналы контроллера флэш-памяти

Сигнал	Направление	Активный уровень	Описание
APB интерфейс			
pclk	Вход	-	Тактовый сигнал
pnres	Вход	Низкий	Сигнал сброса
psel	Вход	Высокий	Сигнал выбора
paddr[4:0]	Вход	-	Адрес
pwr	Вход	-	Сигнал чтения или записи (низкий – чтение, высокий – запись)
pdw[127:0]	Вход	-	Данные на запись
pdr[127:0]	Выход	-	Прочитанные данные

Сигнал	Направление	Активный уровень	Описание
NAND интерфейс			
NF_din[7:0]	Вход	-	Шина входных данных
NF_din[8]	Вход	-	NF_RDY – сигнал готовности микросхемы FLASH (1 - готово)
NF_dou[7:0]	Выход	-	Шина выходных данных (команды, адреса)
NF_dou[8]	Выход	Высокий	NF_CLE – строб команды
NF_dou[9]	Выход	Высокий	NF_ALE – строб адреса
NF_dou[10]	Выход	Низкий	NF_RE – строб записи
NF_dou[11]	Выход	Низкий	NF_WE – строб чтения
NF_dou[12]	Выход	Низкий	NF_CE[0] – выборка микросхемы 0
NF_dou[13]	Выход	Низкий	NF_CE[1] – выборка микросхемы 1
NF_dou[14]	Выход	Низкий	NF_CE[2] – выборка микросхемы 2

NF_den	Выход	-	Сигнал, определяющий направление для двунаправленного порта DIO микросхемы флэш-памяти (1 – данные поступают во флэш, 0 – данные поступают из флэш). Пример:
			assign IO_bus = NF_den ? NF_dou[7:0] : 8'bz;
			assign NF_di[7:0] = IO_bus;

Сигналы запросов			
nand_irq	Выход	Высокий	Запрос на прерывание
TR_req	Выход	Высокий	Запроса на обмен с системным DMA

Контроллер может использоваться для совместной работы с контроллером DMA. Для этих целей предусмотрен выход TR_req. Изменение выхода TR_req из состояния 0 в состояние 1 сигнализирует контроллер DMA о необходимости провести обмен данными.

Соответствие сигналов управления внешней флэш-памятью контактным площадкам микросхемы приведено в Таблице 149. Предусмотрено два варианта соединения выводов контроллера с внешними выводами:

Вариант 1 – при начальной загрузке значением выводов BOOT[2:0] = 100 и по умолчанию при работе (подробнее см. GPIO.Функции порта PD);

Вариант 2 – при начальной загрузке значением выводов BOOT[2:0] = 110 и битами CFG1[18:17] (подробнее см. подраздел 29.1 «Регистр конфигурации периферийных модулей CFG1»).

Таблица 149 – Внешние контакты контроллера

Имя	Порт		Функция
	Вариант 1	Вариант 2	
NF_RDY	PD[31]	PA[18]	Вход готовности флэш-памяти
NF_CS[2]	PD[30]	–	Выход. Выбор банка 2
NF_CS[1]	PD[29]	PA[8]	Выход. Выбор банка 1
NF_CS[0]	PD[28]	PA[17]	Выход. Выбор банка 0
NF_CLE	PD[27]	PA[13]	Выход. Строб команды
NF_ALE	PD[26]	PA[4]	Выход. Строб адреса
NF_RE	PD[25]	PA[2]	Выход. Строб чтения
NF_WE	PD[24]	PA[3]	Выход. Строб записи
NF_D[7]	PD[23]	PA[15]	Шина данных. Бит 7
NF_D[6]	PD[22]	PA[14]	Шина данных. Бит 6
NF_D[5]	PD[21]	PA[12]	Шина данных. Бит 5
NF_D[4]	PD[20]	PA[11]	Шина данных. Бит 4

Имя	Порт		Функция
	Вариант 1	Вариант 2	
NF_D[3]	PD[19]	PA[10]	Шина данных. Бит 3
NF_D[2]	PD[18]	PA[9]	Шина данных. Бит 2
NF_D[1]	PD[17]	PA[6]	Шина данных. Бит 1
NF_D[0]	PD[16]	PA[5]	Шина данных. Бит 0

20.1 Регистры управления контроллером

Поведение контроллера, параметры интерфейса обмена определяются посредством регистров конфигурации контроллера.

В Таблице 150 приведен список регистров контроллера. Регистры контроллера доступны со стороны шины периферийных устройств и имеют базовый адрес 0x80000240.

Таблица 150 – Регистры контроллера флэш-памяти

Смещение	Имя	Описание	Сброс
0	IO_CFG	Регистр конфигурации временных параметров	0x00a3ffff
1	WCT_CFG	Регистр конфигурации времени ожидания	0x000000ff
2	NAND_CFG	Регистр конфигурации протокола обмена	0x0fff035b
3	WR_CFG	Регистр команд записи	0x00100080
4	RD_CFG	Регистр команд чтения	0x00003000
5-7	-	Не используются	
8	CR	Регистр управления	
9	SR	Регистр состояния	
10	AR	Регистр начального адреса обмена	
11	CNTR	Счетчик количества передаваемых слов	
12	DR	Буфер данных.	
13	ERR12	Регистр номеров ошибок 1 и 2	
14	ERR34	Регистр номеров ошибок 3 и 4	
15	SP_BUF	Регистр номера ошибочной подстраницы	

20.1.1 IO_CFG – регистр конфигурации временных параметров

Данный регистр позволяет определить основные временные параметры протокола обмена. Ниже изображены основные сигналы интерфейса обмена и основные временные параметры (Рисунок 90). Длительность стробов выражается количеством тактов частоты синхронизации, на которой работает контроллер. В качестве базовой частоты используется частота периферийной шины.

Таблица 151 – Регистр IO_CFG

Бит	Имя	Описание	Сброс
2:0	CSCA	Число периодов(+1) CLK с момента активизации NF_CS [2:0] до момента активизации сигналов nRE или nWE	111
7:3	CA	Число периодов (+1) активности сигнала nRE во время чтения памяти	11111
10:8	BWD	Число периодов (+1) активности nWE во время записи.	111
14:11	BRT	Число периодов (+1) от момента снятия одного NF_CSx до момента активизации NF_CSx другого банка, с целью избежать конфликт на шине данных	1111

17:15	BHT	Число периодов (+1) CLK в течение которых NF_CSx находится в неактивном состоянии между последовательными выборками. А также число периодов (+1) после снятия nRE или nWE до следующего активного сигнала.	111
18	MDT	Тип интерфейса. Используется совместно с VGA битом	0
20:19	-	Не используется. Значение безразлично	00
21	VGA	Тип интерфейса. Имеет значение при установленном MDT	1
22	-	Не используется. Значение безразлично	
26:23	CSKPL	Управление линиями NF_CSx между доступами к флэш-памяти 0 – линии NF_CSx становятся неактивными между выборками 1 – линии NF_CSx активны (=0) между выборками	0001
31:27	-	Не используется. Значение безразлично	

Отметим функции некоторых бит, значение которых может быть недостаточно понятно из краткого описания в таблице.

Биты MDT и VGA позволяют управлять функцией линий интерфейса: ALE, CLE, nRE (Таблица 152).

Таблица 152 – Дополнительные функции интерфейса

MDT	VGA	ALE	CLE	nRE
0	-	ALE	CLE	nRE
1	0	BA[3]	BA[2]	nRE
1	1	nWRITE	BA[2]	STB

Отметим, что строб STB = ~(nRE & nWE). Таким образом, можно запрограммировать обмен с устройствами, которые имеют более простой интерфейс обмена. Данная функция работает только в случае, когда фазы передачи команды и адреса выключены.

Биты CSKPL необходимо использовать в следующей ситуации. При чтении или записи выборка кристалла (NF_CS) активна только на период одного обмена. Поскольку в операциях с памятью всегда используется последовательный режим обмена, то в некоторых микросхемах флэш-памяти снятие выборки после очередного последовательного чтения вызовет останов наращивания адреса внутри флэш-памяти и возврат ее внутренней машины состояния в исходное состояние. Чтобы этого не произошло необходимо установить бит CSKPL = 1 для соответствующего NF_CS. В этом случае выборка микросхемы памяти будет активна постоянно, начиная с первого обращения.

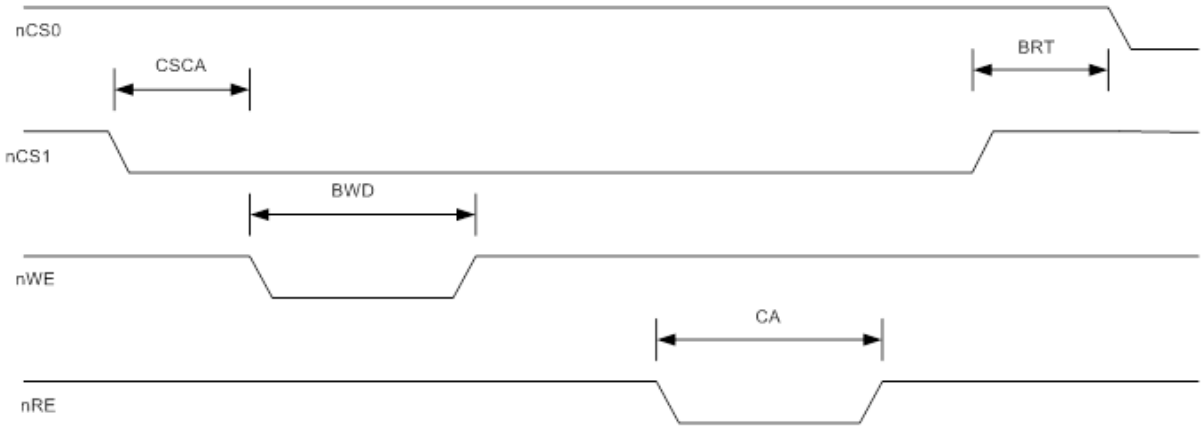


Рисунок 90 – Основные временные параметры обмена

20.1.2 WCT_CFG – регистр конфигурации времени ожидания

Регистр позволяет управлять анализом сигнала готовности внешней памяти и программировать максимальное время ожидания.

Таблица 153 – Регистр WCT_CFG

Бит	Имя	Описание	Сброс
3:0	ENWT	Разрешение анализа входа RnB во время обмена с одним из банков NF_CS[3:0]. 0 – RnB не анализируется 1 – RnB используется для анализа во время цикла обмена	1111
7:4	CCSE	Управление активностью сигнала выборки банка Эти биты позволяют снять выборку банка во время RnB = 0. Биты регистра 7:4 соответствуют NF_CS[3:0]. 0 – установить неактивный уровень если RnB =0 1 – разрешить активный уровень на NF_CSx во время RnB=0	1111
30:21	WTOC	Длительность периода “time-out” Эти биты определяют длительность периода, используемого для контроля ситуации “time-out”. Превышение длительности RnB=0 над длительностью периода WTOC•1024 приводит к ситуации “time-out” и установке бита TOEX	0
31	TOE	Разрешение контроля ситуации “time-out” 0 – запрещено 1 – разрешено	0

Биты ENWT позволяют разрешать для каждого NF_CS анализ сигнала RnB. При подключении устройств к контроллеру флэш-памяти может оказаться, что некоторые NF_CS используются для выборки устройств, которые не управляют сигналом RnB. В этом случае необходимо очистить бит ENWT соответствующего NF_CS.

Функции бит CCSE могут быть использованы в следующем случае. При чтении новой страницы или при записи флэш-памяти вырабатывается сигнал RnB=0 на продолжительное время. Если бит CCSE равен нулю для соответствующего NF_CS, то в случае RnB = 0 сигнал выборки станет неактивным на период времени, когда RnB равен нулю. Для некоторых типов микросхем это позволяет снизить мощность потребления.

Биты WTOC позволяют запрограммировать интервал времени time-out. При этом будет отслеживаться период равный WTOC•1024 тактов.

Бит TOE разрешает анализ длительности обмена.

20.1.3 NAND_CFG – регистр конфигурации протокола обмена

Позволяет задать основные параметры флэш-памяти и режимы работы.

Таблица 154 – Регистр NAND_CFG

Бит	Имя	Описание	Сброс
2:0	VOLCOL	Размер страницы 000 – 128 байта 001 – 512 байта	011

		010 – 1024 байта 011 – 2048 байта 100 – 4096 байта 101 – 8192 байта	
3	ROWBT	Число байт адреса для выборки строки 0 – 2 байта 1 – 3 байта	1
5:4	COMCLW	Число команд для записи 00 – 1 01 – 2 10 – 3 (расширенный режим) 11 – не используется	01
7:6	COMCLR	Число команд для чтения 00 – 1 01 – 2 10 – не используется 11 – не используется	01
9:8	ADRCL	Число адресных циклов 00 – 1 цикл 01 – только выбор столбца 10 – только выбор строки 11 – строка и столбец	11
10	-	Не используется	0
11	-	Не используется	0
12	ADRSC	Управление пропуском адресного цикла 0 – передавать адрес во время обмена 1 – пропустить	0
13	DATSC	Управление пропуском цикла данных 0 – передавать данные во время обмена 1 – пропустить	0
14		Не используется.	
15	COMSC	Управление пропуском цикла передачи команд 0 – передавать команды во время обмена 1 – пропустить	0
20:16	RESBF	Время задержки до анализа активности RnB Биты задают длительность (*2) периода от момента передачи последнего байта адреса до момента анализа сигнала RnB.	0x1F
27:21	RESAF	Время задержки после завершения активности RnB Биты задают длительность (*2) периода от момента, когда сигнал RnB стал неактивным до момента чтения или записи данных.	0x7F
31:28	ADDRH	Биты адреса расширения	0000

Биты VOLCOL определяют размер страницы флэш-памяти.

Бит ROWBT определяет количество байт адреса, которые необходимо передать во флэш-память для выборки страницы. Для задания номера начального байта внутри страницы используется два байта, что позволяет использовать страницы размером до 64 Кбайт.

Биты COMCLW определяют количество команд (байт команд), используемых при записи данных во флэш-память.

Биты COMCLR выполняют аналогичную функцию при чтении.

Биты ADRCL позволяют управлять передачей адреса.

Если бит ADRSC установлен, контроллер пропускает фазу передачи адреса.

Если бит DATSC установлен, контроллер пропускает фазу управления данными.

Бит COMSC управляет пропуском фазы передачи команд.

При чтении новой страницы флэш-память вырабатывает низкий уровень на выходе RnB. Его длительность равна времени считывания новой страницы из накопителя в буфер (флэш-памяти). Флэш-память изменяет значение на выходе RnB после передачи последнего байта адреса с некоторой задержкой.

Биты RESBF позволяют задержать анализ вывода RnB после передачи адреса перед началом чтения. Это позволяет избежать срабатывания неверного чтения.

Биты RESAF позволяют запрограммировать время готовности памяти после того как на линии RnB установится высокий уровень.

Биты ADDRH позволяют подключить к одному NF_CS микросхемы памяти объемом больше 4Г байт.

В одном из режимов старта системы предусмотрен старт с начальной загрузкой из флэш-памяти. После сброса контроллер настроен на работу с микросхемой, у которой страница памяти равна 2К байт. Для выбора строки будет использовано три байта адреса, а для выбора столбца два байта адреса. При передаче адреса будут переданы все 5 байт адреса и две команды чтения.

20.1.4 WR_CFG и RD_CFG – регистры конфигурации

Регистры используются для задания значения команд, используемых при чтении или записи флэш-памяти.

Таблица 155 – Регистр WR_CFG

Бит	Имя	Описание	Сброс
7:0	COMW1	первая команда записи	0x80
15:8	COMW2	вторая команда записи	0x00
23:16	COMW3	третья команда записи	0x10
31:24	-		

Таблица 156 – Регистр RD_CFG

Бит	Имя	Описание	Сброс
7:0	COMR1	первая команда чтения	0x00
15:8	COMR2	вторая команда чтения	0x30
23:16	COMR3	третья команда чтения	0x00
31:24	-		

20.1.5 CR – регистр управления

Назначение разрядов регистра приведено ниже (Таблица 157). В данном регистре программируются дополнительные параметры обмена.

Таблица 157 – Регистр CR

Бит	Название	Функция
0	EN	Разрешение работы (1)
1	RW	Чтение (0) или запись (1)
2	-	
3	SQE	Разрешение последовательного доступа (когда 0)

4	RIM	Разрешение прерывания при чтении (1-разрешение)
5	TIM	Разрешение прерывания при записи (1-разрешение)
6	CIM	Разрешение прерывания при достижении счетчика значения 0 (1-разрешение)
7	-	
9:8	SZ	Размер передаваемых данных 00 – 4 байта, 10 – два байта, 11 – байт
10	-	
11	DBSZ	Размер шины данных FIFO контроллера 0 – 32 бита 1 – 128 бит
12	ERI_EN	Разрешение прерывания в случае обнаружения ошибочной ситуации в работе контроллера, т.е. когда обнаруживается ситуация тайм-аут.
13	RnBI_EN	Разрешение прерывания в случае обнаружения положительного фронта на входе RnB, т.е. когда флаг RnB_Fрегистра состояния равен 1.
14	EIM	1 – разрешение прерывания при обнаружении ошибки ECC
15	ECC_ON	1 – включение режима ECC

После сброса значение регистра управления равно 0x0001, т.е. контроллер включен, настроен на чтение слов с размерностью шины данных в 32 бита.

Далее будут описаны типовые процедуры чтения и записи данных. Они состоят из передачи управляющих команд, адреса и блока данных. Вся эта процедура выполняется, когда бит SQE равен нулю. Если установить бит SQE в единицу, указанный протокол будет выполняться для каждого слова данных, т.е. после передачи команды, адреса и чтения 4-х байт данных (биты SZ равны 00), перед чтением следующих 4-х байт будет повторно выполнена передача команд и адреса. Данный режим очень медленный и не рекомендуется для применения.

Биты разрешения прерывания RIM, TIM, CIM могут быть использованы для организации работы контроллера вместе с процессором. Например, если выполняется чтение данных, разрешение прерывания RIM будет сигнализировать процессору о том, что во внутреннем буфере имеются данные для чтения. Если выполняется запись данных, разрешение прерывания TIM будет сигнализировать процессору о том, что во внутреннем буфере имеется место для записи данных. Прерывание CIM сигнализирует процессору о завершении обмена, т.е. во внутренний буфер или из внутреннего буфера были передано количество слов данных указанное в счетчике CNTR.

Бит RW управляет направлением передачи данных. Значение 0 определяет чтение данных из внешней памяти во внутренний буфер, а значение 1 задает передачу данных из внутреннего буфера во внешнюю память. Отметим, что чтение данных из внешней памяти инициируется только, если есть свободное место во внутреннем буфере, а запись во внешнюю память инициируется только когда есть данные во внутреннем буфере.

Бит EN разрешает работу контроллера. Если в данный бит записать ноль, произойдет очистка внутренних указателей внутреннего буфера и перевод машины состояний в исходное положение.

Биты SZ задают размер слова передаваемых данных. Размер 1 или 2 байта должен использоваться только для спецопераций.

Биты DBSZ задают размер шины данных FIFO. При смене размера шины необходимо выключить контроллер NAND флэш-памяти.

20.1.6 DR – регистр данных

Контроллер имеет FIFO данных объемом восемь 32-разрядных слов. Оно выполняет функцию буфера между системой и внешней флэш-памятью. При чтении данных из флэш-памяти, данные записываются в FIFO. При записи – данные считываются из FIFO и передаются в флэш-память. Можно установить размер шины данных FIFO. Если бит DBSZ равен нулю, размер FIFO составит 8x32 бита, а если DBSZ равен 1, размер FIFO – 2x128. В последнем случае процессор или DMA должны производить обмен квадрословами. Признаки готовности данных также формируются с учетом размерности шины данных.

Рассмотрим поведение буфера данных в зависимости от размерности шины данных FIFO.

1. Шина данных 32 бита:

В этом случае буфер имеет размер 8 слов по 32 бита. Имеются указатели на запись и чтение каждый разрядностью 3 бита. Запись в буфер увеличивает указатель записи на 1. Чтение из буфера уменьшает указатель чтения на 1.

Признак готовности буфера принимать данные – TFS активен (равен 1) в случаях, когда буфер пуст, или когда буфер не заполнен полностью, т.е. TFS==1 означает что в буфер можно записать одно слово данных.

Признак RFS (для случая, когда буфер настроен на чтение) активен (равен 1) в случаях, когда имеются данные для чтения, т.е. когда буфер не пуст. Это означает, что из буфера можно прочитать одно слово данных.

2. Шина данных 128 бит:

В этом случае буфер имеет разный вид со стороны процессора и со стороны контроллера флэш-памяти. Для процессора буфер имеет размер 2 слова по 128 бит, а для контроллера по-прежнему 8 слов по 32 бита. Указатели на запись и чтение изменяются в зависимости от направления передачи:

- **Запись**

  Запись в буфер увеличивает указатель записи на 4, т.к. в буфер помещаются сразу 4 слова данных. Чтение из буфера уменьшает указатель чтения на 1. Признак готовности буфера принимать данные TFS активен (равен 1) в случаях, когда буфер пуст или, когда буфер не заполнен полностью, т.е. TFS==1 означает, что в буфер можно записать одно квадрослово данных.

- **Чтение**

  Запись в буфер со стороны флэш-памяти увеличивает указатель записи на 1, т.к. в буфер помещается 1 слово данных. Чтение из буфера уменьшает указатель чтения на 4, т.к. считывается сразу 4 слова. Признак RFS активен (равен 1) в случаях, когда имеется хотя бы одно квадрослово данных для чтения, т.е. когда из памяти прочитаны как минимум 4 слова. Это означает, что из буфера можно прочитать одно квадрослово данных.

Бит разрядности шины данных имеет отношение только к буферу данных. Доступ к регистрам управления осуществляется только словами по 32 бита. При смене разрядности шины данных буфера **обязательно** нужно выполнить выключение контроллера (CR[0]=0). Это приведет к обнулению указателей чтения и записи и таким образом гарантирует их дальнейшую корректную работу.

Для операций с флэш-памятью, требующих чтения менее, чем 4-х слов данных необходимо использовать только 32-разрядный режим шины данных буфера.

20.1.7 AR – регистр адреса

Перед началом обмена необходимо задать адрес, начиная с которого будет производиться чтение или запись. В последующем этот адрес будет автоматически увеличиваться на 1.При записи значения в регистр адреса необходимо помнить, что данный адрес есть адрес слова. Внутри контроллера адрес расширяется AF[33:0]={AR[31:0],2'b00} двумя нулевыми битами для формирования указателя на байт. Биты AF[33:32] используются для выбора микросхемы памяти, а биты AF[31:0] прямо адресуют память. Это позволяет адресовать 4 Гбайт памяти. Внутри контроллера биты AF[31:0] дополняются битами расширения адреса AX[35:0]=={ADDRH[3:0],AF[31:0]}. Биты расширения имеют значение, если объем подключенной микросхемы памяти более 4 Гбайт.

Внутри контроллера исполнительный адрес AX делится на адрес столбца и адрес строки. Значимым параметром является адрес столбца (column). Количество бит, выделяемое для адреса столбца, зависит от размера страницы флэш-памяти. Например, для случая, когда страница равна 2К байт в качестве адреса столбца будут использованы биты AX[10:0]. Биты AX[35:11] будут адресовать строку.

20.1.8 CNTR – счетчик количества слов

Задает количество слов, которое будет прочитано из флэш-памяти или записано во флэш-память. Счетчик имеет разрядность 24 бита. При этом длина слова может быть 1, 2 или 4 байта.

20.1.9 SR – регистр состояния

Отражает значение флагов запросов прерываний. Биты регистра описаны ниже (Таблица 158).

Таблица 158 – Регистр SR

Бит	Название	Функция
0	EMPTY	FIFO данных пусто (1)
1	FULL	FIFO данных заполнено полностью (1)
2	TFS	Запрос от FIFO данных на запись новых данных процессором или контроллером прямого доступа. FIFO наполовину пусто (1)
3	RFS	Запрос от FIFO данных на чтение новых данных процессором или контроллером прямого доступа. FIFO наполовину заполнено (1)
4	IRQ	Общий запрос прерывания от контроллера. Более детально причина запроса прерывания может быть установлена согласно состояния бит 20:16 регистра. 0 – нет запроса 1 – запрос на прерывание
7:5	-	-

11:8	TOEX	Флаги состояния Time-Out. Биты отражают факт обнаружения ситуации “time-out” в процессе обмена с соответствующим NF_CSx. 0 – нет ошибки 1 – обнаружена ситуация “time-out” Биты очищаются программно, посредством записи в них 1
12	-	
13	RnB_F	Флаг изменения входа RnB из низкого уровня «0» в высокий «1» 0 – не было изменения 1 – был переход линии RnB из 0 в 1 Бит очищается программно, посредством записи в него 1
15:14	-	
16	RFS_I	Запрос на прерывание когда RFS==1 и бит RIM==1. Запрос может быть очищен посредством чтения буфера данных
17	TFS_I	Запрос на прерывание когда TFS==1 и бит TIM==1. Запрос может быть очищен посредством записи в буфер данных.
18	CNTZ_I	Запрос на прерывание когда значение счетчика становится равным нулю и бит CIM==1. Запрос может быть очищен посредством записи нового значения счетчика
19	TO_I	Запрос на прерывание когда TOEX==1 и бит ERI_EN ==1. Запрос может быть очищен посредством записи 1 в бит TOEX
20	RnBF_I	Запрос на прерывание когда RnB_F ==1 и бит RnBI_EN ==1. Запрос может быть очищен посредством записи 1 в бит RnB_F
30:21	-	-
31	RnB	Вход готовности NAND флэш-памяти. Отражает состояние линии RnB

Контроллер интерфейса содержит внутренний счетчик, который позволяет отследить ситуацию, когда обмен с памятью превышает допустимый временной интервал (ситуация time-out). Это служит защитой от блокировки работы процессора, выйти из которой можно только сбросив всю систему.

Если ситуация time-out произошла при обращении к флэш-памяти, подключенной к некоторому NF_CS выводу, соответствующий бит TOEX в регистре состояния будет установлен и цикл обмена завершится некорректно, т.е. считанные данные будут неверными или запись будет осуществлена неверно. Подобная ситуация может произойти если длительность низкого уровня на входе RnB превышает период time-out. Процесс аварийного завершения обмена может вызвать прерывание работы процессора, если соответствующее прерывание разрешено.

Сбросить биты TOEX можно только записав в них значение 1. Запись 0 в соответствующие биты не вызывает их изменения.

Бит RnB отражает состояние соответствующего выхода готовности микросхемы флэш-памяти. Бит может быть использован для определения готовности флэш-памяти.

20.2 Чтение NAND флэш-памяти

Чтение флэш-памяти всегда осуществляется в последовательном режиме и представляет собой передачу во внешнюю память команд, байт адреса и затем чтение непосредственно самих данных. Различные варианты чтения отличаются друг от друга количеством передаваемых байт команд и адреса, а также количеством принимаемых байт данных.

При чтении данных вначале происходит передача одного (или двух) байта команды, затем передаются байты адреса, после этого читаются данные. Первую команду, с которой начинается чтение, необходимо записать в поле COMR1 регистра RD_CFG.

Если размер страницы равен 512 байт, команды для доступа к первой половине страницы и ко второй половине различны (00h – для первой, 01h – для второй). Контроллер отслеживает эту ситуацию сам. Пользователь должен записать в поле COMR1 значение 00h и определить размер страницы как 512 байт. Если при чтении контроллер обнаруживает, что обращение осуществляется ко второй половине страницы, он автоматически устанавливает команду 01h вместо 00h.

После передачи команды контроллер посылает во флэш-память байты адреса. Количество адресных байт определяется в регистре конфигурации и зависит от емкости микросхемы памяти. После этого контроллер читает заданное количество байт данных последовательно и помещает их во внутренний буфер.

Передаваемое значение адреса формируется из регистра адреса AR (биты 29:0) и поля расширения адреса ADDRH (биты 2:0). Поле расширения определяет старшие биты передаваемого адреса.

Контроллер ориентирован на чтение количества байт кратных 4. Однако для выполнения специальных операций (сброс микросхемы, чтение регистра состояния микросхемы) можно выполнять более короткие операции чтения. Для чтения одного байта необходимо в счетчик слов CNTR записать значение 1 и установить размер передаваемых данных в регистре управления CR равным байту. Для чтения 2-х байт в счетчик слов записываем 1, а размер данных устанавливаем равным полуслову.

20.3 Запись в NAND флэш-память

Операция записи во многом аналогична операции чтения только направление передачи данных противоположное, т.е. из внутреннего буфера во внешнюю микросхему памяти. Команды, используемые при записи во флэш-память, должны быть записаны в поля регистра конфигурации WR_CFG. В типичном случае протокол записи данных в память состоит из передачи первого командного байта, передачи байт адреса, записи данных и передачи второго байта команды, инициирующего процесс записи внутри флэш-памяти.

Необходимо отметить, что во время программирования текущей страницы во внутреннем накопителе, флэш-память становится недоступной. Это занимает довольно продолжительное время. При выполнении операций контроллер анализирует готовность памяти и в случае обращения процессора за новыми данными в момент программирования предыдущих данных, будет осуществлена остановка работы контроллера до момента готовности флэш-памяти.

Для записи данных необходимо определить две команды: для инициализации процесса записи и для его окончания. Эти две команды записываются в поля COMW1 и COMW3 регистра конфигурации. Вначале посылается первая команда, затем байты адреса, байты данных. Во время пересылки данных во внешнюю память контроллер анализирует состояние счетчика CNTR(после пересылки каждого слова значение счетчика уменьшается на 1) и в случае равенства счетчика значению 1, контроллер отсылает последнее слово данных и за ним следует вторая команда записи. После приема второй команды записи микросхема памяти начинает процесс перезаписи принятых данных во внутренний накопитель. Флэш-память тратит значительное время для перезаписи внутреннего буфера в накопитель. На все это время память становится недоступной. Имеется возможность анализировать сигнал готовности посредством чтения регистра состояния.

20.4 Типичные процедуры работы с NAND флэш-памятью фирмы Samsung

20.4.1 Сброс машины состояния флэш-памяти

- **IO_CFG =** K, устанавливаем временные параметры соответствующие подключенной микросхеме. Если тип микросхемы неизвестен, используем максимальные значения временных параметров (т.е. худший случай)
- **NAND_CFG =** 0x3000, устанавливаем пропуск фаз передачи адреса и чтения данных, число команд чтения равно 1.
- **RD_CFG =** 0xFF, определяем первую команду чтения (COMR1).
- **CR =** 0x0301. Устанавливаем чтение байта.
- **CNTR =** 1, инициируем процесс чтения.
- происходит фиктивное чтение байта из флэш-памяти. Поскольку фазы адреса и данных будут пропущены, переданы будут только команды **FFh** во флэш-память. Тем не менее, нужно помнить, что байт данных (неизвестное значение) все равно будет загружен во внутренний буфер, и этот байт нужно вычитать, чтобы привести указатели в исходное положение.

20.4.2 Чтение ID

Согласно спецификации, для чтения ID необходимо передать команду **90h**, затем один байт адреса (значение 0) и затем прочитать два байта ID. Это можно сделать следующим образом:

- здесь и далее предполагаем, что **IO_CFG** был определен ранее
- **RD_CFG =** 0x90, определяем первую команду чтения (COMR1).
- **NAND_CFG =** 0, устанавливаем передачу только одного байта адреса, число команд чтения равно 1.
- **CR =** 0x0001. Устанавливаем чтение слова.
- **AR =** 0, адрес равен нулю
- **CNTR =** 1, инициируем процесс чтения.
- выполняется чтение слова по адресу 0. Прочитанные четыре байта и есть ID.

20.4.3 Чтение регистра состояния

Согласно спецификации, для чтения состояния необходимо передать команду **70h**, а затем прочитать байт состояния, т.е.:

- здесь и далее предполагаем, что **IO_CFG** был определен ранее
- **RD_CFG =** 0x70, определяем первую команду чтения (COMR1).
- **NAND_CFG =** 0x1000, устанавливаем пропуск адреса, число команд чтения равно 1.
- **CR =** 0x0301. Устанавливаем чтение байта.
- **CNTR =** 1, инициируем процесс чтения.
- Прочитанный байт и есть регистр состояния. При этом необходимо помнить, что перед чтением данных из внутреннего буфера необходимо убедиться в их наличии. Для этого необходимо сканировать регистр состояния SR и ожидать установки бита RFS в единицу. Это будут сигнализировать о том, что операция завершилась и данные готовы. При

чтении регистра состояния внешней памяти необходимо помнить, что если память занята (например, производит стирание блока), то она может притормозить контроллер посредством сигнала RnB. В этом случае перед чтением регистра состояния лучше выключить анализ RnB для выбранного банка памяти.

20.4.4 Очистка (erase) блока флэш-памяти

Согласно спецификации, для стирания блока флэш-памяти необходимо передать команду **60h**, передать адрес блока, пропустить запись данных и в заключение передать команду **d0h**. Таким образом, последовательность действий может быть следующей:

- **RD_CFG =** 0xd060, определяем первую(COMR1) и вторую (COMR2) команды чтения.
- **NAND_CFG =** 0x2248, устанавливаем пропуск данных, число команд чтения равно 2, передачу только адреса для выбора блока.
- **AR =** address, записываем в регистр адреса значение соответствующее номеру стираемого блока. Здесь необходимо рассчитать какие именно биты адреса регистра AR используются для указания номера блока. Если размер страницы указан как 2 Кбайт, то биты AR[8:0] будут использоваться для адресации байта внутри страницы (плюс дополнительно два нулевых бита). Биты AR[29:9] будут указывать на номер страницы. Стираемый блок может состоять из нескольких страниц. Если такое число страниц равно, например, 64, то передаваемые биты AR[14:9] безразличны. Учитываются только биты AR[29:15].
- **CR =** 0x0301. Устанавливаем чтение байта.
- **CNTR =** 1, инициируем процесс чтения.
- выполняется фиктивное чтение байта. Это приводит к передаче во флэш-память требуемой последовательности байт. После этого можно анализировать флаг готовности флэш-памяти путем чтения регистра состояния памяти или путем анализа флага RnB регистра состояния SR. Также не забываем о прочитанном байте в буфере.

20.4.5 Запись во флэш-память в последовательном режиме

Согласно спецификации, для записи данных необходимо передать команду **80h**, передать адрес, записать данные последовательно в текущую страницу, после этого необходимо передать команду **10h**, которая инициирует процесс программирования. Таким образом:

- **WR_CFG =** 0x100080, определяем первую команду записи (COMW1=80h), вторая команда записи не используется, третья команда записи (COMW3=10h).
- **NAND_CFG =** 0x035B, устанавливаем передачу полного адреса, число команд записи равно 2, размер страницы 2 Кбайт.
- **AR =** address, записываем в регистр адреса значение начального адреса слова(!) начиная с которого мы хотим произвести запись блока данных.
- **CR =** 0x0043. Устанавливаем операцию записи слова размером 4 байта и прерывание по достижению счетчика значения ноль.
- **CNTR =** К, инициируем процесс записи К*4 байт данных.
- выполняем запись последовательного блока данных во внутренний буфер, анализируя при этом его доступность для записи. Как только мы записали

последнее слово в буфер, переходим к ожиданию прерывания о завершении операции записи.

20.4.6 Чтение из флэш-памяти в последовательном режиме

Согласно спецификации, для чтения данных необходимо передать команду **00h**(**01h**, **50h**), передать адрес, прочитать данные последовательно. Таким образом:

- **RD_CFG =** 0x3000, определяем команды чтения (COMR1=00h, COMR2=30h).
- **NAND_CFG =** 0x035B, устанавливаем передачу полного адреса, число команд записи равно 2, размер страницы 2 Кбайт.
- **AR =** address, записываем в регистр адреса значение начального адреса слова(!) начиная с которого мы хотим произвести чтение блока данных.
- **CR =** 0x0001. Устанавливаем операцию чтения слова размером 4.
- **CNTR =** К, инициируем процесс чтения К*4 байт данных.
- переходим к анализу флага состояния RFS буфера данных. Как только он устанавливается в 1, производим чтение данных. После прочтения всех данных завершаем операцию.

20.5 Рекомендации по организации работы с прерываниями

Для корректной работы NAND контроллера с контроллером DMA нужно будет строго соблюдать последовательность действий. При этом данная последовательность будет разной при чтении и при записи. Исходное состояние – все выключено.

При чтении:

- программируем канал DMA. После включения он ждет импульса от NAND контроллера.
- программируем NAND контроллер и включаем обмен.
- при чтении данных из памяти в FIFO генерируется запрос к контроллеру DMA (или прерывание процессору) о том, что можно выполнить дальнейшую пересылку.
- Контроллер DMA пересылает данные.
- после чтения последнего слова NAND-контроллер может сгенерировать прерывание для процессора. Однако это не очень разумно, т.к. данные еще не переместились в память. Поэтому при чтении лучше использовать прерывание от канала DMA.
- запрос прерывания от контроллера DMA говорит о том, что все операции завершены.

При записи:

- программируем канал DMA. После включения он ждет импульса от NAND-контроллера.
- программируем NAND-контроллер и включаем обмен, сразу идет запрос к контроллеру DMA.
- контроллер DMA пересылает данные в FIFO и они передаются в память.
- после пересылки последнего слова контроллер DMA может генерировать прерывание. Но при записи лучше использовать прерывание от NAND. Оно будет говорить о том, что данные уже переданы в память.

21 Контроллер LCD-панели

Возможно подключение LCD-панелей (тип TFT) к процессору. Параметры контроллера позволяют поддерживать панели различных разрешений.

Структура контроллера LCD-панели представлена на Рисунке 91.

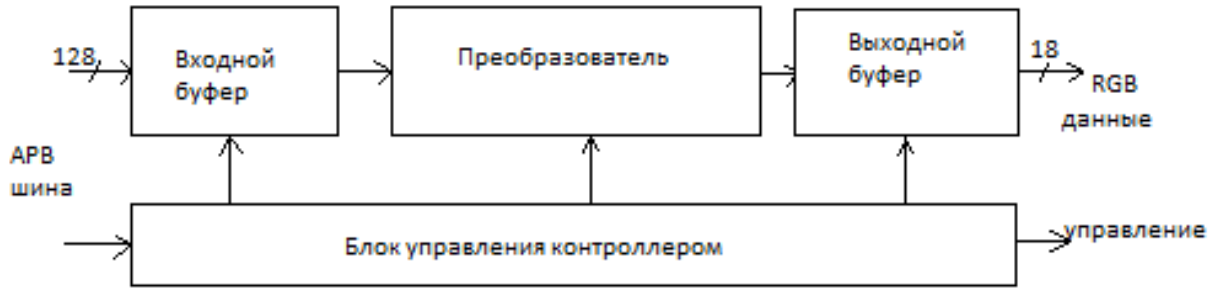


Рисунок 91 – Структура контроллера

Назначение внешних выводов контроллера приведено в Таблице 159.

Таблица 159 – Назначение внешних выводов контроллера

Обозначение вывода	Назначение вывода контроллера	Тип	Функциональное назначение
PB[4]	LC_B[0]	I/O	Интерфейс ЖКИ. Бит синего цвета 0
PB[5]	LC_B[1]	I/O	Интерфейс ЖКИ. Бит синего цвета 1
PB[6]	LC_B[2]	I/O	Интерфейс ЖКИ. Бит синего цвета 2
PB[7]	LC_B[3]	I/O	Интерфейс ЖКИ. Бит синего цвета 3
PB[8]	LC_B[4]	I/O	Интерфейс ЖКИ. Бит синего цвета 4
PB[9]	LC_B[5]	I/O	Интерфейс ЖКИ. Бит синего цвета 5
PB[10]	LC_G[0]	I/O	Интерфейс ЖКИ. Бит зеленого цвета 0
PB[11]	LC_G[1]	I/O	Интерфейс ЖКИ. Бит зеленого цвета 1
PB[12]	LC_G[2]	I/O	Интерфейс ЖКИ. Бит зеленого цвета 2
PB[13]	LC_G[3]	I/O	Интерфейс ЖКИ. Бит зеленого цвета 3
PB[14]	LC_G[4]	I/O	Интерфейс ЖКИ. Бит зеленого цвета 4
PB[15]	LC_G[5]	I/O	Интерфейс ЖКИ. Бит зеленого цвета 5
PB[16]	LC_R[0]	I/O	Интерфейс ЖКИ. Бит красного цвета 0
PB[17]	LC_R[1]	I/O	Интерфейс ЖКИ. Бит красного цвета 1
PB[18]	LC_R[2]	I/O	Интерфейс ЖКИ. Бит красного цвета 2
PB[19]	LC_R[3]	I/O	Интерфейс ЖКИ. Бит красного цвета 3
PB[20]	LC_R[4]	I/O	Интерфейс ЖКИ. Бит красного цвета 4
PB[21]	LC_R[5]	I/O	Интерфейс ЖКИ. Бит красного цвета 5
PB[22]	LC_T[0]	I/O	Интерфейс ЖКИ. Дополнительный выход 0
PB[23]	LC_T[1]	I/O	Интерфейс ЖКИ. Дополнительный выход 1
PB[24]	LC_T[2]	I/O	Интерфейс ЖКИ. Дополнительный выход 2
PB[25]	LC_T[3]	I/O	Интерфейс ЖКИ. Дополнительный выход 3
PB[26]	LC_PWM	I/O	Интерфейс ЖКИ. Выход ШИМ для управления яркостью ЖКИ
PB[27]	LC_DRDY	I/O	Интерфейс ЖКИ. Сигнал готовности данных для ЖКИ
PB[28]	LC_VSYNC	I/O	Интерфейс ЖКИ. Сигнал вертикальной синхронизации ЖКИ
PB[29]	LC_HSYNC	I/O	Интерфейс ЖКИ. Сигнал горизонтальной синхронизации ЖКИ
PB[30]	LC_CLK	I/O	Интерфейс ЖКИ. Синхросигнал

Контроллер осуществляет прием входной информации, ее преобразование и выдачу на внешнюю LCD-панель в соответствии с заданным режимом работы. Задание режима работы контроллера осуществляется посредством программирования его регистров. Регистры контроллера LCD-панели подключены к пользовательской периферийной шине и имеют базовый адрес 0x8000_0160. Перечень регистров приведен ниже (Таблица 160).

Таблица 160 – Регистры контроллера

Номер	Название	Функция
0	CTRL	Регистр управления
1	STATUS	Регистр состояния
2	HTIM	Fpline (старт, стоп)
3	VTIM	Fpframe (старт, стоп)
4	HVLEN	Размер экрана
5	HDxTIM	Горизонтальная выдача данных (старт, стоп). Дополнительное окно
6	VDxTIM	Вертикальная выдача данных (старт, стоп). Дополнительное окно
7	Vsize	Размер видеобуфера
8	FON	Данные для заполнения фоновой области
9	PXDV	Делитель пиксель клока
10	HDTIM	Горизонтальная выдача данных (старт, стоп). Основное окно
11	VDTIM	Вертикальная выдача данных (старт, стоп). Основное окно
12	PANEL_CFG	Регистр конфигурации панели
13	PWM_CR	Регистр управления ШИМ
14	SLP_PERIOD	Регистр задания интервала сна
15	-	
16	TIM_GP0	Управление сигналом GPIO_0
17	TIM_GP1	Управление сигналом GPIO_1
18	TIM_GP2	Управление сигналом GPIO_2
19	TIM_GP3	Управление сигналом GPIO_3
20	EXT_MEM_ADDR	Начальный адрес внешней памяти для выделенного канала ПДП

Управление внешней LCD панелью осуществляется посредством набора внешних выходов. Внешние выводы контроллера состоят из 26-разрядного порта общего назначение и выхода синхронизации PIX_CLK (или FPSHIFT). Назначение выводов порта общего назначение приведено в Таблице 161.

Таблица 161 – Внешние выводы управления панелью

Разряды порта	Название	Функция
17:0	fpdat	Шина данных пикселей
18	Gpio_0	Дополнительное управление панелью
19	Gpio_1	Дополнительное управление панелью
20	Gpio_2	Дополнительное управление панелью
21	Gpio_3	Дополнительное управление панелью
22	pwm	Выход ШИМ
23	drdy	Готовность данных
24	fpframe	Импульс начала фрейма
25	fpline	Импульс начала линии
26	fpshift	Пиксель-клок

Входной буфер имеет входную шину данных разрядностью 128 бит для загрузки данных. Емкость буфера составляет пять 128-разрядных слов. Входной буфер формирует запрос к контроллеру DMA на прием данных, если он не заполнен. Из входного буфера данные поступают на преобразователь порциями по 32 бита. Модуль преобразователя осуществляет формирование информации о RGB-пикселе на основании полученной информации и заданных блоком управления параметрах преобразования. На выходе преобразователя всегда данные RGB пикселя разрядностью 18 бит. Каждая цветовая компонента имеет разрядность 6 бит. Эти данные поступают в выходной буфер. Емкость выходного буфера составляет 256 пикселей. При формировании временной диаграммы управления панелью, данные из выходного буфера в нужный момент времени выдаются на внешнюю панель. В задачу преобразователя входит постоянная подкачка данных в выходной буфер по мере его освобождения. Использование выходного буфера позволяет осуществить равномерную выдачу информации на панель, даже в случаях, когда DMA контроллер может быть блокирован на некоторое время более высокоприоритетными обменами.

Регистры контроллера позволяют настроить контроллер на работу с различными типами LCD панелей. При этом возможно задание, как дополнительных внешних управляющий линий, так и настройка контроллера для работы с выбранным разрешением экрана. Регистры управления позволяют задать количество вертикальных линий панели, а также количество пикселей в линии. Дополнительно задается область выдачи данных, т.к. количество линий и пикселей в линии может превышать количество пикселей, задействованных в отображении информации. В зависимости от требований панели, контроллер может выполнять как непрерывную выдачу данных (автоматический переход на начало повторной выдачи картинки после завершения текущей выдачи), так и выдачу данных по запросу. В последнем случае контроллер после завершения выдачи картинки переходит в спящий режим и начинает новую выдачу по истечению запрограммированного интервала времени или по требованию процессора. Данный режим может использоваться, только если панель (внешнее устройство-приемник) допускает останов пиксель клока на некоторое время.

Контроллер LCD панели можно также рассматривать как параллельный синхронный 16-разрядный порт для выдачи информации с настраиваемыми параметрами выдачи данных и генерации сигналов управления.

Ниже приведена типовая временная диаграмма работы контроллера (Рисунок 92). Задача контроллера сводится к выдаче на внешние контакты одного фрейма (картины) изображения. Начало картины всегда начинается с выдачи активного уровня FPFRAME (на рисунке активный низкий уровень). Изображение состоит из некоторого множества линий. Начало каждой линии сопровождается активным уровнем FPLINE (на рисунке активный низкий уровень). Информация о пикселях текущей линии выдаются только при активном уровне на линии готовности данных DRDY (на рисунке активный высокий уровень). Значение пикселя на шине данных FPDAT выдается относительно спадающего фронта (из высокого в низкий) синхросигнала FPSHIFT. Нарастающим фронтом синхросигнала данные должны приниматься в устройстве приемнике информации.

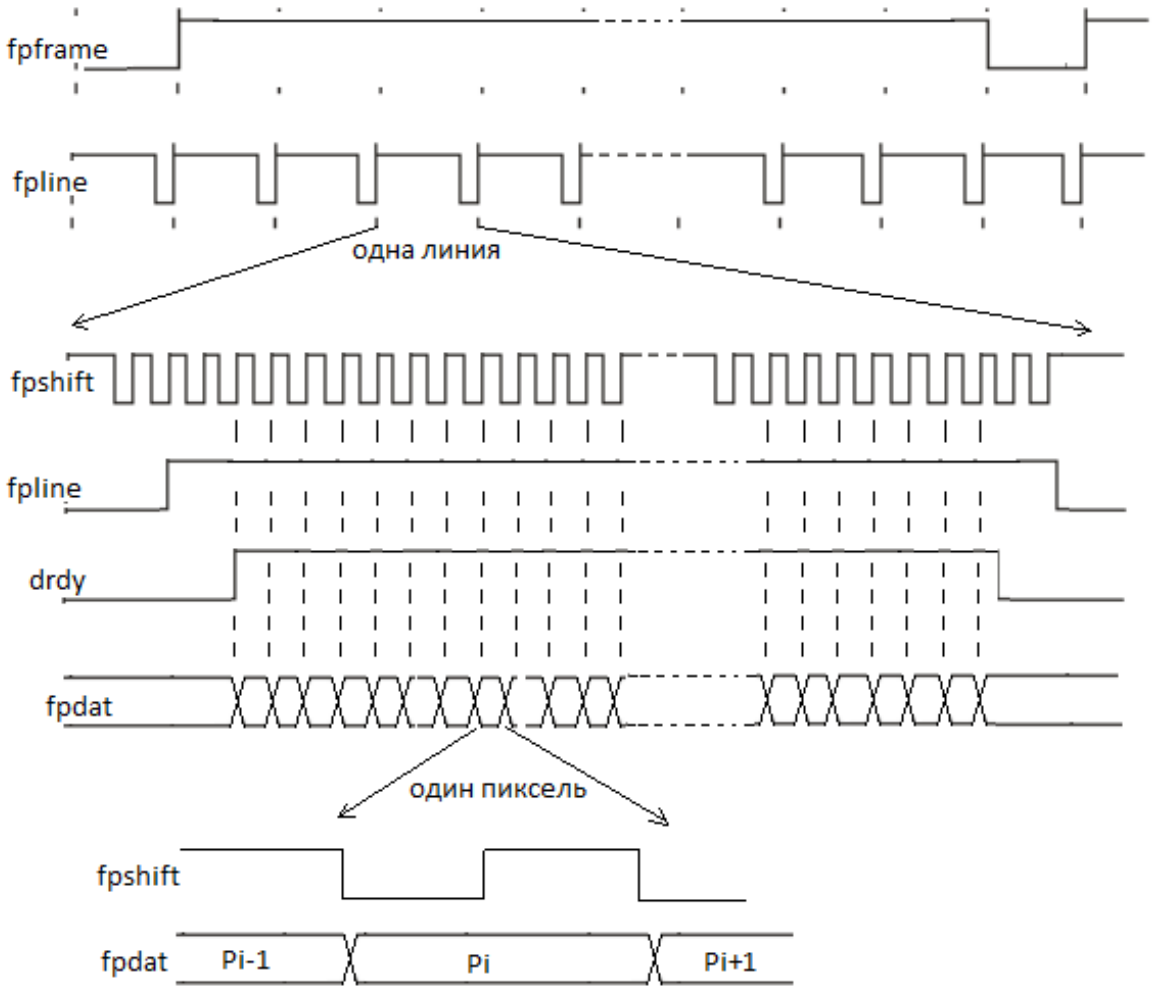


Рисунок 92 – Временная диаграмма работы контроллера

Регистры управления контроллером предназначены для задания различных временных характеристик обмена. После выдачи одного фрейма контроллер повторяет свои действия, но, возможно, уже с другими пиксельными данными.

21.1 Регистр управления CTRL

Назначение разрядов регистра управления приведено ниже (Таблица 162).

Таблица 162 – Разряды регистра управления CTRL

Бит	Имя	Функция
0	VEN	Разрешение работы контроллера: 0 – выключен 1 – включен
1	VIE	Разрешение прерывания после окончания фрейма: 0 – запрещено 1 – разрешено

2	HIE	Разрешение прерывания после окончания одной линии: 0 – запрещено 1 – разрешено
3	VBIE	Разрешение прерывания после загрузки всего видеобанка данных в выходной буфер
4	SLPIE	Разрешение прерывания после завершения сна
5	-	
6	XRQEN	Разрешение прерывания после завершения чтения DMA из внешней памяти
8:7	VBL	Количеств бит в пикселе. Зависит от выбранного режима (биты CD). См. Таблицу 163
10:9	CD	Количество бит для одного пикселя входной информации: 00 – 8 01 – 16 10 – 24 11 – 32
11	-	
12	HLDM	Останов повторной загрузки видеоданных после завершения выгрузки видеобуфера (1 – разрешено)
13	HLDV	Останов повторной генерации управляющих сигналов панели после завершения выгрузки видеобуфера: 0 – останов запрещен 1 – останов разрешен
14	-	
15	BL	Активный уровень линии готовности данных DRDY (0 – высокий, 1 – низкий)
16	-	
17	VBGR	1 – BGR режим, 0 – RGB режим
18	-	
19	-	
20	SLP_MODE	1 – спящий режим после выгрузки видеобуфера
21	SLP_PCLK	Стоп делитель пиксель клока в спящем режиме (1)
22	SLP_PXEN	Стоп разрешения работы пиксель клока в спящем режиме
23	SLP_HOLD	Останов повторной генерации управляющих сигналов панели после завершения выгрузки видеобуфера (1 – разрешено)
24	SLP_CLRF	1 – автоматический сброс флагов после задержки спящего режима 0 – ожидание программного обслуживания интерфейса
25	DMA_2QW	Разрешение режима загрузки двух квадрослов за одну транзакцию DMA: 1 – режим разрешен 0 – режим заперщен
27:26	–	Зарезервировано
28	W2W_EN	Разрешение использования регистров дополнительной активной области: 0 – запрещено 1 – разрешено
29	PXP_EN	Разрешение останова пикселькклока в случае если в выходном FIFO нет данных: 0 – запрещено. Будет генерироваться ошибка 1 – разрешено. Клок продолжит тактирование когда будут данные

Бит VEN включает контроллер. Этот бит должен быть установлен в единицу, только когда все параметры контроллера определены.

Биты VIE и HIE позволяют задать формирование запроса прерывания после окончания выдачи всей картинки (VIE) или после окончания очередной одной горизонтальной линии (HIE). Данные прерывание имеют скорее отладочную

функцию, т.к. их сложно использовать при нормальном функционировании контроллера.

Бит VBIE позволяет сформировать запрос прерывания после завершения загрузки последнего 32-разрядного слова фрейма из входного буфера в преобразователь. Генерация данного прерывания означает завершение чтения всех данных видеофрейма.

Бит SLPIE позволяет сформировать запрос прерывания к процессору после истечения периода сна контроллера.

Биты CD задают количество бит во входном слове, которые используются для кодирования информации о пикселе. Значения 0, 1, 2 и 3 соответствуют 8, 16, 24 и 32 битам.

В случае использования 32 бит, старшие 8 бит отбрасываются. Использование 32 бит равносильно режиму 24 бит, однако информация о каждом пикселе выровнена на границе слова. Из-за этого впустую теряется один байт. Режим 8 бит может использоваться для режима черно-белого изображения, т.к. в этом случае каждая цветовая RGB компонента равна значению байта. В случае использования 16-бит режима возможны несколько вариантов кодирования. Эти варианты кодируются с использованием битов VBL. Возможные варианты кодирования приведены ниже (Таблица 163). Значение VBL равное нулю (или 1) соответствует стандартному формату RGB565.

Бит DMA_2QW вводит режим загрузки двух квадрослов за одну транзакцию. Также для разрешения этого режима необходимо установить в 1 бит 31 старшего дополнительного регистра приемника DMA канала 4-7. При использовании данного режима поле «адрес устройства» дополнительного регистра канала DMA должно быть 0x80000180.

Таблица 163 – Режимы формирования пиксельных данных

CD	VBL	byte	R	G	B
11	-	-	Din[23:16]	Din[15:8]	Din[7:0]
10	-	-	Din*[23:16]	Din*[15:8]	Din*[7:0]
00	x0	0	mod3(Din[7:5])	mod3(Din[4:2])	mod2(Din[1:0])
00	x0	1	mod3(Din[15:13]	mod3Din[12:10]	mod2Din[9:8]
00	x0	2	mod3(Din[23:21]	mod3Din[20:18]	mod2Din[17:16]
00	x0	3	mod3(Din[31:29]	mod3Din[28:26]	mod2Din[25:24]
00	x1	0	mod2(Din[7:6]	mod3(Din[5:3]	mod3(Din[2:0]
00	x1	1	mod2(Din[15:14]	mod3(Din[13:11]	mod3(Din[10:8]
00	x1	2	mod2(Din[23:22]	mod3(Din[21:19]	mod3(Din[18:16]
00	x1	3	mod2(Din[31:30]	mod3(Din[29:27]	mod3(Din[26:24]
01	0x	-	{ Din[15:11],000}	{ Din[10:5],00}	{ Din[4:0],000}
01	10	-	{ Din[13:10],0000}	{ Din[8:5],0000}	{ Din[3:0],0000}
01	11	-	{ Din[14:11],0000}	{ Din[9:6],0000}	{ Din[4:1],0000}

mod2(a) = (a[1:0]==2’b11)? 8’hFF: {a[1:0], 000000},
mod3(a) = (a[2:0]==3’b111)? 8’hFF: {a[2:0], 00000}

Из всех приведенных режимов, отметим режим RGB565, как единственный случай, при котором на внешние контакты информация выдается без потерь входной информации. Данный режим позволяет рассматривать LCD порт как синхронный 16-разрядный порт.

Если используемая LCD панель допускает кратковременную остановку сигнала синхронизации, полезным будет использование бита PXP_en. Если этот бит установлен, то в случае, когда необходимо выдать новое значение данных и выходной буфер пуст, контроллер остановит сигнал синхронизации панели до

момента готовности данных в выходном буфере. Это гарантирует отсутствие сбоев при выдаче информации.

21.2 Регистр состояния STATUS

Регистр STATUS отражает текущее состояние запросов прерываний к процессору.

Таблица 164 – Регистр состояния STATUS

Бит	Имя	Функция
0	-	
1	LUINT	Флаг прерывания при ошибке чтения выходного буфера. Если флаг установлен, то при необходимости выдачи данных, буфер был пуст. Сбрасывается только при записи в данный бит значения 1
2	STA_SLEEP	Таймер сна начал отсчет. Устанавливается в момент перехода контроллера в режим сна
3	FIN_SLEEP	Таймер сна закончил отсчет. Устанавливается в момент завершения режима сна
4	VINT	Флаг завершения выдачи фрэйма
5	HINT	Флаг завершения выдачи линии
6	VBSINT	Флаг завершения чтения данных видеобуфера. Устанавливается при передаче последнего слова данных из входного буфера в преобразователь. Сбрасывается программно или аппаратно
7	-	Зарезервировано
19-8		
20	SLEEP_EXE	Режим сна (если 1). Только чтение
31-21		

Флаги регистра устанавливаются аппаратно. Сброс флагов может выполняться программно путем записи в необходимый бит значения 1. Также флаги могут сбрасываться аппаратно в момент выхода контроллера из режима сна. Флаг LUINT может быть сброшен только программно.

21.3 Регистр управления сигналом fpline (HTIM)

Этот регистр осуществляет управление импульсом начала линии и задает смещение импульса относительно начала строки, а также ширину импульса и его активный уровень. Единицей измерения является длительность PX_CLK сигнала.

Таблица 165 – Регистр состояния управления сигналом fpline (HTIM)

Бит	Имя	Функция
9:0	HPS	FPLINE начальная позиция (hps+1)
15	HPL	FPLINE активный уровень (0 – высокий, 1 – низкий)
25:16	HPW	FPLINE конечная позиция (hpw+1)

21.4 Регистр управления сигналом fpframe (VTIM)

Этот регистр осуществляет управление импульсом начала фрейма и задает смещение импульса относительно начала фрейма, а также ширину импульса и его активный уровень. Единицей измерения является одна строка.

Таблица 166 – Регистр управления сигналом fpframe (VTIM)

Бит	Имя	Функция
9:0	VPS	FPFRAME начальная позиция
15	VPL	FPFRAME активный уровень (0 – высокий, 1 – низкий)
25:16	VPW	FPFRAME конечная позиция (vpw+1)

21.5 Регистр управления размером экрана (HVLEN)

Этот регистр задает количество пикселей в одной линии и количество линий на экране. Не все пиксели линии могут быть отображаемыми на экране. Также не все линии могут быть отображаемыми на экране. Регистр HVLEN задает общее количество пикселей и линий, которое включает всю информацию (отображаемую и нет).

Таблица 167 – Регистр управления размером экрана (HVLEN)

Бит	Имя	Функция
6:0	HT	Размер по горизонтали ((ht + 1) * 8)
25:16	VT	Размер по вертикали (vt + 1)

21.6 Регистр размера видео буфера (VSIZE)

Регистр задает количество 32-разрядных слов, которые образуют видеобуфер. Контроллер использует это число, чтобы отследить момент окончания загрузки данных и сформировать запрос на прерывание. Момент формирования запроса прерывания соответствует чтению последнего слова видеобуфера из входного FIFO в преобразователь.

Таблица 168 – Регистр размера видео буфера (VSIZE)

Бит	Имя	Функция
19:0	VSIZE	Размер видеобуфера в 32-бит словах + 1
31:20	-	

После чтения последнего слова в преобразователь, происходит

- сброс бита 5 регистра управления (не используется).
- установка бита 6 регистра состояния (флаг завершения чтения видеобуфера данных).

Контроллер всегда используется совместно с выбранным для него каналом DMA. В канале DMA также задается число слов для передачи и запрос прерывания от канала DMA может быть более информативным, чем запрос прерывания от LCD контроллера на обслуживание видеобуфера.

Если бит 12 (HLDM) регистра управления установлен, после установки бита 6 регистра состояния, входное FIFO будет постоянно посылать в преобразователь признак того, что оно пусто. При этом входной буфер может формировать запрос к контроллеру DMA и если канал работает, то во входной буфер может быть загружено до пяти квадрослов данных. Данные из входного буфера не будут поступать в преобразователь до тех пор, пока бит 6 регистра состояния не будет сброшен. Бит может быть сброшен программно или аппаратно при выходе из режима сна.

21.7 Делитель для сигнала синхронизации панели (PXDV)

LCD контроллер использует для своей работы сигнал синхронизации периферийной шины SOC_CLK. Данный синхросигнал используется для формирования сигнала синхронизации пикселей PX_CLK.

PX_CLK = SOC_CLK / (P_div + 1)

Таблица 169 – Делитель для сигнала синхронизации панели (PXDV)

Бит	Имя	Функция
7:0	P_DIV	Значение делителя
8	EN_DIV	Включение делителя: 0 – выключен 1 – включен

21.8 Управление горизонтальной активной областью панели (HDTIM)

Этот регистр осуществляет управление видимой областью линии и задает смещение данной области относительно начала строки, а также ширину области. Единицей измерения является длительность PX_CLK сигнала. Как отмечалось ранее, регистр HVLEN задает общее количество пикселей в строке. Счетчик пикселей строки будет изменяться от 0 до Np, где Np это количество пикселей в строке. Регистр HDTIM задает начальный номер счетчика, после которого начинается выдача пиксельной информации, а также номер счетчика после которого прекращается выдача информации. Таким образом, будет задана активная область горизонтальной строки.

Таблица 170 – Управление горизонтальной активной областью панели (HDTIM)

Бит	Имя	Функция
9:0	HDPS	Начальная позиция активной области (hdps+1)
25:16	HDPE	Конечная позиция активной области (hdpe+1)

21.9 Управление вертикальной активной областью панели (VDTIM)

Этот регистр осуществляет управление видимой областью экрана и задает смещение данной области относительно начала экрана, а также ширину области. Единицей измерения является одна линия. Данный регистр задает номер линии, с которой начинается отображение информации на экране, а также номер линии, с которой начинается неотображаемая область. На рисунке 93 показаны функции, выполняемые регистрами при отображении информации.

Таблица 171 – Управление вертикальной активной областью панели (VDTIM)

Бит	Имя	Функция
9:0	VDPS	Номер начальной отображаемой линии
25:16	VDPE	Номер конечной отображаемой линии

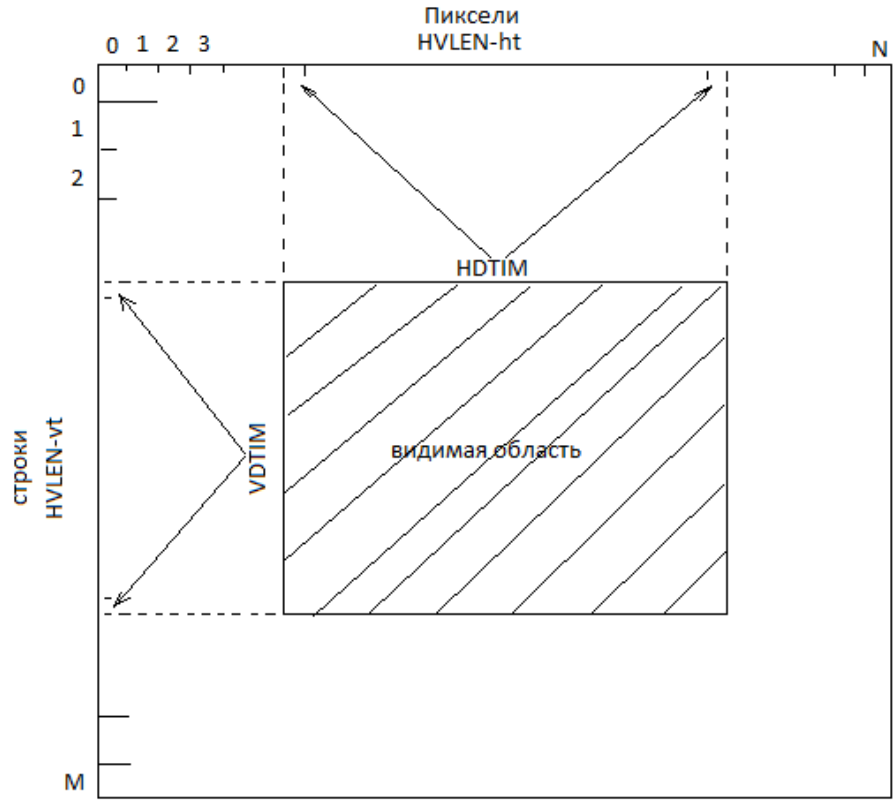


Рисунок 93 – Параметры отображаемой информации

21.10 Управление дополнительной горизонтальной активной областью панели (HDxTIM)

Регистр HDxTIM осуществляет управление дополнительной видимой областью линии и задает смещение данной области относительно начала строки, а также ширину области. Единицей измерения является длительность PX_CLK сигнала. Регистр HDxTIM используется совместно с регистром HDTIM и позволяет задать пересечение двух активных областей. Информация будет выводиться только в ту область, которая принадлежит обоим множествам. В область, которая принадлежит множеству HDTIM, но не принадлежит множеств HDxTIM, будет выводиться значение фонового регистра FON.

Таблица 172 – Управление дополнительной горизонтальной активной областью панели (HDxTIM)

Бит	Имя	Функция
9:0	HXDPS	Начальная позиция активной области (HXDPS+1)
25:16	HXDPE	Конечная позиция активной области (HXDPE+1)

21.11 Управление дополнительной вертикальной активной областью панели (VDxTIM)

Регистр VDxTIM осуществляет управление видимой областью экрана и задает смещение данной области относительно начала экрана, а также ширину области. Единицей измерения является одна линия. Регистр VDxTIM используется совместно с регистром VDTIM и позволяет задать пересечение двух активных областей. Информация будет выводиться только в ту область, которая принадлежит обоим

множествам. В область, которая принадлежит множеству строк VDTIM, но не принадлежит множеству VDxTIM, будет выводиться значение фонового регистра FON.

Таблица 173 – Управление дополнительной вертикальной активной областью панели (VDxTIM)

Бит	Имя	Функция
9:0	VXDPS	Номер начальной отображаемой линии
25:16	VXDPE	Номер конечной отображаемой линии

Регистры HDxTIM и VDxTIM могут быть использованы в случае, когда размер отображаемой информации значительно меньше разрешения панели. В этом случае данная функция позволяет значительно сократить трафик шины, т.к. передаваться будет только реально отображаемая информация.

21.12 Регистр FON

При использовании регистров задания дополнительного окна, данные выводятся только в область пересечения двух множеств: основного и дополнительного окон. В область основного окна, которая не входит в область дополнительного окна выводятся данные из регистра FON.

Таблица 174 – Регистр FON

Бит	Имя	Функция
7:0	B_Fon	Данные для B выхода (используются биты 7:2)
15:8	G_Fon	Данные для G выхода (используются биты 15:10)
23:16	R_Fon	Данные для R выхода (используются биты 23:18)

21.13 Регистр конфигурации панели (PANEL_CFG)

Таблица 175 – Регистр конфигурации панели (PANEL_CFG)

Бит	Имя	Функция
5:0	-	
6	FPSHI	Инверсия пиксель клока
8:7	-	
9	DVI_MODE	При записи всегда должен быть 0
10	DVI_CLK	При записи всегда должен быть 0
11	-	
31:12	-	Всегда ноль

21.14 Регистр управления выходом ШИМ (PWM_CR)

Регистр может быть использован для управления яркостью панели.

Таблица 176 – Регистр управления выходом ШИМ (PWM_CR)

Бит	Имя	Функция
0	CLKEN	Разрешение работы (1)
2:1	-	Должен быть 0
3	FRSH	Когда равен 1, на выходе PWM постоянный высокий уровень вне зависимости от работы внутренних счетчиков
7:4	PWM_Dv	делитель 0 – 1, 1 – 2, 2 – 4, 3 – 8, ……, 0xF - 32768

Бит	Имя	Функция
15:8	DUTY	Длительность активного уровня. 0 – всегда 0 1 – высокий для 1 периода 2 – высокий для 2 периодов …… 0xFF - – высокий для 255 периодов
23:16	RELOAD	8-бит верхнее значение. Когда счетчик достигнет это значение, он загрузит ноль и начнет отсчет заново.

В качестве базовой частоты PWM модуля используется SOC_CLK синхросигнал. Предварительно SOC_CLK может быть поделен на число, задаваемое битами PWM_DV. После чего мы будем иметь рабочую частоту PWM счетчика. Счетчик осуществляет увеличение на 1 каждый такт своей рабочей частоты. Если значение счетчика меньше значения “duty”, на внешнем выходе присутствует высокий уровень. В противном случае – низкий. При достижении счетчиком значения RELOAD, он сбросит свое значение в ноль и начнет счет заново.

21.15 Регистр управления сигналом GPIO_0, 1, 2, 3

Регистр позволяет задать требуемую временную диаграмму на выводе GPIO_0.

Таблица 177 – Регистр управления сигналом GPIO_0, 1, 2, 3

Бит	Имя	Функция
9:0	GP0_ST	GPIO_0 начальная позиция (+1)
15	HPL	GPIO_0 активный уровень (0 – высокий, 1 – низкий)
25:16	GP0_SP	GPIO_0 конечная позиция (+1)

Если значение начала импульса совпадает с его окончанием, сигнал изменяет свое значение на противоположное. Единицей измерений является пикселькл ок.

Регистры GPIO_1, 2, 3 аналогичны по функциям регистру GPIO_0. Регистры позволяют сформировать, при необходимости, дополнительные управляющие сигналы.

21.16 Организация «спящего режима»

В обычном режиме работы, после включения контроллера, выполняется процесс генерации импульсов управления панелью, а также выдача информации. После выдачи одного фрейма данных (одной картины) контроллер обычно приступает к повторной выдачи новой или старой (если видеобуфер не обновлялся) информации. При этом ряд панелей требуют непрерывной подачи сигнала синхронизации и регенерации данных. Однако существуют панели, которые после выдачи одного фрейма данных, допускают останов синхронизации и некоторый перерыв в подаче информации. В этом случае контроллер после выдачи данных может перейти в спящий режим на некоторое время. Интервал, в течение которого контроллер не выполняет передачи данных, задается с помощью регистра SLP_PERIOD. По истечении заданного периода контроллер возобновит чтение информации из видеобуфера и выдачу данных на внешний интерфейс. Переход в режим сна выполняется при завершении формирования всех управляющих сигналов одного фрейма и при установленном бите Slp_MODE.

В режиме сна контроллер:

- Загружает в счетчик интервала сна значение регистра SLP_PERIOD и начинает обратный отсчет.
- Если бит SLP_PCLK установлен в 1, выполняет останов делителя пикселькклока и таким образом останавливает синхронизацию всех внутренних счетчиков контроллера.
- Если бит SLP_PXEN установлен в 1, запрещает работу всех внутренних счетчиков, даже если делитель пикселькклока работает (т.е. бит SLP_PCLK равен 0). В этом случае активным может быть только выход fpshift (если бит SLP_PCLK равен 0). Все другие выходы управления остановлены.
- Если бит SLP_HOLD установлен в 1, запрещает изменение счетчика пикселей и устанавливает его в нулевое значение. Аналогичен действию бита HLDV.
- Если бит SLP_CLRF установлен в 1, при завершении интервала сна, контроллер выполнит очистку всех флагов, которые были установлены после выгрузки предыдущего фрейма.

Когда счетчик интервала сна уменьшает свое значение до нуля, происходит выход из режима сна и контроллер возобновляет генерацию управляющих сигналов и выдачу информации. При реализации режима сна и установленном бите SLP_CLRF, обязательно должен использоваться бит 12 (HLDM). Бит HLDM должен быть установлен в 1. Если бит SLP_CLRF установлен в 1, при выходе из режима сна также инициализируется счетчик количества слов буфера видеоданных. Однако если бит HLDM не будет установлен, то это может вызвать установку некорректного значения, т.к. к моменту окончания сна некоторое количество слов данных может быть прочитано (если канала DMA включен), обработано и размещено в выходном буфере.

Возможность использования спящего режима при работе контроллер, существенно снижает трафик к памяти процессора и сокращает потребление системы.

22 Интерфейс к аудио-кодеку AC97/I2S

Специальный интерфейс позволяет осуществить вывод аудио данных на внешний аудио-кодек типа AC97. Также данный интерфейс может работать по протоколу I2S шины и ее различных вариантов. Интерфейс имеет встроенные FIFO и может работать с DMA контроллером.

22.1 Регистры интерфейса AC97/I2S

Регистры интерфейса AC97/I2S подключены к пользовательской периферийной шине и имеют базовый адрес 0x8000_0200 (интерфейс 0) и 0x8000_0220 (интерфейс 1).

Таблица 178 – Регистры интерфейса

Номер	Название	Описание
0	SICR0	Регистр управления 0
1	SINT	
2	SICR2	Регистр управления 2
3	SISR	Регистр состояния после маскирования
4	SIRSR	Регистр состояния до маскирования
5	SIIER	Регистр разрешения прерывания
6	SIIDR	Регистр запрещения прерывания
7	SIICR	Регистр сброса флагов запросов прерывания
8	SIADR	Регистр-буфер аудио данных
9	SIMDR	Регистр-буфер модемных данных
10	ACCAR	AC канал, регистр адреса команды
11	ACCDR	AC канал, регистр данных команды
12	ACSAR	AC канал, регистр адреса состояния
13	ACSDR	AC канал, регистр данных состояния
14	ACGDR	AC канал, регистр данных GPIO
15	ACGSR	AC канал, регистр состояния GPIO
16	I2S_T_CR	Регистр управления передатчиком I2S
17	I2S_R_CR	Регистр управления приемником I2S
…	-	
28	SICR3	Регистр управления 3

22.1.1 Регистр управления SICR0

Регистр используется только для управления режимом работы AC97. Назначение разрядов регистра приведено в Таблице 179.

Таблица 179 – Биты регистра управления SICR0

Бит	Имя	Функция
0	ENB	Включение режима AC97. Включен (1) или выключен (0)
1	-	Для включения режима AC97 должен быть равен 0. Если равен 1 – запрещает включение режима AC97.
2	MODEN	Модемный кодек присутствует (1)
3	AMCCH	AMC кодек присутствует (1)
4	RST	Запись 1 вызывает сброс внешнего кодека

5	LPBK	Тестовый режим AC97 (когда равен 1)
6	BCKD	Источник синхронизации. 0 – используется внешний источник синхронизации 1 – используется внутренний источник (биты делителя REQLP)
7	MONO_DA	0 – стерео выход, 1 – моно выход
8	BIT8_DA	0 – 16 бит выход, 1 – 8 бит выход
9	MONO_AD	0 – стерео вход, 1 – моно вход
10	BIT8_AD	0 – 16 бит вход, 1 – 8 бит вход
11	TO_EN	Разрешение детектирования отсутствия клока аудиокодека. Если установлен в 1, осуществляется проверка отсутствия синхронизации на входе BITCLK (либо частота очень низкая).
12	DIV_en	Разрешение делителя клока AC97. Когда равен 1 разрешает включение внутреннего делителя клока. Значение делителя задается полем REQLP.
31-13	-	Не используются

Возможно подключение только аудио кодека, раздельных аудио и модемного кодеков, совмещенного AMC кодека.

В тестовом режиме выход данных соединен с входом данных.

22.1.2 Регистр управления SICR2

Регистр используется только для управления режимом работы AC97. Поля REQLP, EREC и ERPL используются в режиме I2S. Назначение разрядов регистра приведено в Таблице 180.

Таблица 180 – Биты регистра управления SICR2

Бит	Имя	Функция
0	EREC	Разрешение записи (1)
1	ERPL	Разрешение воспроизведения (1)
2	EINC	Разрешение приема модемных данных (1)
3	EOUT	Разрешение передачи модемных данных (1)
4	EGPIO	Разрешение приема GPIO данных (1)
5	WKUP	Запись 1 вызывает инициализацию процедуры пробуждения кодека.
6	DRSTO	Если 1 – выключает AC-link status read time-out function
22:7	REQLP	Коэффициент деления для формирования синхросигнала в режиме I2S и AC97
31:23	-	Не используются

22.1.3 Регистр управления SICR3

Регистр используется только для управления режимом работы AC97. Назначение разрядов регистра приведено в Таблице 181.

Таблица 181 – Биты регистра управления SICR3

Бит	Имя	Функция
0	COMSEL	Разрешение вторичного кодека (1)
2:1	C2ID	ID вторичного кодека
5:3	-	Биты общего назначения
31:6	-	Не используются

22.1.4 Регистр состояния SISR

Регистр используется только для управления режимом работы AC97. Режим I2S использует буфер аудио данных для своей работы. В связи с этим биты, отражающие состояние буфера аудио данных, могут быть использованы и в режиме I2S. Назначение разрядов регистра приведено в Таблице 182.

Таблица 182 – Биты регистра состояния SISR

Бит	Имя	Функция
0	DTD	Завершение передачи данных. Устанавливается в 1 после завершения передачи в кодек адреса и данных. Сбрасывается в 0 посредством записи значения 1 в 0-й бит регистра SIICR.
1	RDD	Завершение чтения данных. Устанавливается в 1 после завершения чтения из кодека адреса и данных. Сбрасывается в 0 посредством записи значения 1 в 1-й бит регистра SIICR.
2	GTD	Завершение передачи GPIO данных. Устанавливается в 1 после завершения передачи в кодек GPIO данных. Сбрасывается в 0 посредством записи значения 1 в 2-й бит регистра SIICR.
3	-	
4	BSY	Занят. 0 – интерфейс в нерабочем состоянии или выключен 1 – интерфейс передает или принимает фрейм в текущий момент.
5	ATNE	Буфер передатчика аудио данных не пуст. 0 – FIFO передатчика пусто 1 – FIFO передатчика содержит данные
6	ATNF	Буфер передатчика аудио данных заполнен не полностью. 0 – FIFO передатчика аудио данных заполнено полностью 1 – FIFO передатчика аудио данных заполнено не полностью
7	ARNE	Буфер приемника аудио данных не пуст. 0 – в FIFO приемника нет данных 1 – в FIFO приемника есть данные
8	ATFS	Запрос буфера передатчика аудио данных. 0 – FIFO передатчика аудио данных заполнено больше чем наполовину 1 – FIFO передатчика аудио данных заполнено меньше чем наполовину
9	ARFS	Запрос буфера приемника аудио данных 0 – FIFO приемника аудио данных заполнено меньше чем наполовину 1 – FIFO приемника аудио данных заполнено больше чем наполовину
10	ATUR	Отсутствие аудио данных для передачи. Устанавливается в 1 при попытке чтения аудио данных из FIFO передатчика в момент, когда оно пусто. Сбрасывается в 0 посредством записи значения 1 в 10-й бит регистра SIICR.
11	AROR	Переполнение буфера приемника аудио данных. Устанавливается в 1 при попытке записи аудио данных в FIFO приемника в момент, когда оно заполнено полностью. Сбрасывается в 0 посредством записи значения 1 в 11-й бит регистра SIICR.
12	MTNF	Буфер передатчика модема не заполнен. 0 – FIFO передатчика модемных данных заполнено полностью 1 – FIFO передатчика модемных данных заполнено не полностью

13	MRNE	Буфер приемника модема не пуст. 0 – в FIFO приемника нет данных 1 – в FIFO приемника есть данные
14	MTFS	Запрос от буфера передатчика модема. 0 – FIFO передатчика модемных данных заполнено больше чем наполовину 1 – FIFO передатчика модемных данных заполнено меньше чем наполовину
15	MRFS	Запрос от буфера приемника модема. 0 – FIFO приемника модемных данных заполнено меньше чем наполовину 1 – FIFO приемника модемных данных заполнено больше чем наполовину
16	MTUR	Отсутствие данных в буфере передатчика модема. Устанавливается в 1 при попытке чтения модемных данных из FIFO передатчика в момент когда оно пусто. Сбрасывается в 0 посредством записи значения 1 в 16-й бит регистра SIICR.
17	MROR	Переполнение буфера приемника модема. Устанавливается в 1 при попытке записи модемных данных в FIFO приемника в момент когда оно заполнено полностью. Сбрасывается в 0 посредством записи значения 1 в 17-й бит регистра SIICR.
18	RSTO	Флаг состояния «тайм-аут» Устанавливается в 1 при чтении из кодека состояния time-out. Сбрасывается в 0 посредством записи значения 1 в 18-й бит регистра SIICR.
19	CLPM	Флаг отсутствия частоты синхронизации. Равен 1 если частота кодека меньше частоты APB шины больше, чем 25 раз. Детектирует отсутствие частоты синхронизации кодека.
20	CRDYPR	Первичный кодек готов.
21	CRDYSC	Вторичный кодек готов.
22	RESU	Возобновление. Устанавливается в 1 при обнаружении sdata_in & EN_AC & CLPM. Сбрасывается в 0 посредством записи значения 1 в 22-й бит регистра SIICR
23	GINT	Флаг изменения состояния GPIO. Устанавливается в 1 при приеме GPIO[0] = 1 Сбрасывается в 0 посредством записи значения 1 в 23-й бит регистра SIICR
24	RS3V	Слот 3 принятых данных достоверен Отражает значение бита достоверности 3-го слота.
25	RS4V	Слот 4 принятых данных достоверен
26	RS5V	Слот 5 принятых данных достоверен
27	RS12V	Слот 12 принятых данных достоверен
31-28	-	Не используются

22.1.5 Регистр разрешения прерывания SIIER

Биты регистра разрешают запрос прерывания от соответствующих им разрядов регистра состояния. Запись 1 в соответствующий бит вызывает разрешение прерывания. Запись 0 не вызывает изменения разряда. Регистр используется только для управления режимом работы AC97. Режим I2S использует буфер аудио данных для своей работы. В связи с этим биты, отражающие состояние буфера аудио данных, могут быть использованы и в режиме I2S. Назначение разрядов регистра приведено в Таблице 183.

Таблица 183 – Биты регистра разрешения прерывания SIIER

Бит	Имя	Разрешение прерывание если
0	DTD	Завершению передачи данных AC97
1	RDD	Завершению приема данных AC97
2	GTD	Завершению передачи данных GPIO
7:3	-	
8	ATFS	Запросу буфера передатчика аудио данных
9	ARFS	Запросу буфера приемника аудио данных
10	ATUR	Ошибке в буфере передатчика аудио данных (отсутствие данных в момент передачи)
11	AROR	Переполнению буфера приемника аудио данных
13:12	-	
14	MTFS	Запросу буфера передатчика модемных данных
15	MRFS	Запросу буфера приемника модемных данных
16	MTUR	Ошибке в буфере передатчика модемных данных (отсутствие данных в момент передачи)
17	MROR	Переполнению буфера приемника модемных данных
18	RSTO	Флагу таймаута внешнего кодека
21:19	-	
22	RESU	Установке флага возобновления работы
23	GINT	Изменению состояния GPIO
31-24	-	
0	DTD	Завершению передачи данных AC97
1	RDD	Завершению приема данных AC97

22.1.6 Регистр запрещения прерывания SIIDR

Биты регистра запрещают запрос прерывания от соответствующих им разрядов регистра состояния. Запись 1 в соответствующий бит вызывает запрещение прерывания. Запись 0 не вызывает изменения разряда. Назначение разрядов регистра аналогично разрядам регистра SIIER.

22.1.7 Регистр адреса команды ACCAR

Регистр используется только для управления режимом работы AC97. При программировании регистров внешнего аудио-кодека необходимо задать адрес регистра и записываемые данные. Адрес регистра кодека записывается в регистр ACCAR интерфейса.

Назначение разрядов регистра приведено в Таблице 184.

Таблица 184 – Биты регистра адреса команды ACCAR

Бит	Имя	Функция
11:0	-	Не используются
18:12	IX	Индекс управляющего регистра кодека. Бит 12 всегда должен быть равен 1
19	RW	Бит чтения (1) или записи (0)
31:20	-	Не используются

22.1.8 Регистр данных команды ACCDR

Регистр используется только для управления режимом работы AC97. При программировании регистров внешнего аудио-кодека необходимо задать адрес

регистра и записываемые данные. Данные регистра кодека записывается в регистр ACCDR интерфейса.

Назначение разрядов регистра приведено в Таблице 185.

Таблица 185 – Биты регистра данных команды ACCDR

Бит	Имя	Функция
3:0	-	Не используются
19:4	CDR	16 бит записываемые данные.
31:20	-	Не используются

22.1.9 Регистр состояния адреса команды ACSAR

Регистр используется только для управления режимом работы AC97. Назначение разрядов регистра приведено в Таблице 186.

Таблица 186 – Биты регистра состояния адреса команды ACSAR

Бит	Имя	Функция
11:0	-	Не используются
18:12	SAR	Эхо индекса управляющего регистра кодека, данные которого были прочитаны
31:19	-	Не используются

22.1.10 Регистр состояния данных команды ACSDR

Регистр используется только для управления режимом работы AC97. Назначение разрядов регистра приведено в Таблице 187.

Таблица 187 – Биты регистра состояния данных команды ACSDR

Бит	Имя	Функция
3:0	-	Не используются
19:4	SDR	16 бит данных, прочитанных из кодека
31:20	-	Не используются

22.1.11 Регистр данных GPIO ACGDR

Регистр используется только для управления режимом работы AC97. Имеется возможность управлять некоторыми линиями общего назначения кодека. Данные для записи в биты GPIO кодека должны быть предварительно записаны в регистр ACGDR.

Назначение разрядов регистра приведено в Таблице 188.

Таблица 188 – Биты регистра данных GPIO ACGDR

Бит	Имя	Функция
3:0	-	Не используются
19:4	GDAT	16 бит данных, которые будут записаны в GPIO кодека
31:20	-	Не используются

22.1.12 Регистр состояния данных GPIO ACGSR

Регистр используется только для управления режимом работы AC97. Назначение разрядов регистра приведено в Таблице 189.

Таблица 189 – Биты регистра данных GPIO ACGSR

Бит	Имя	Функция
3:0	-	Не используются
19:4	SDR	16 бит GPIO данных, прочитанных из кодека
31:20	-	Не используются

22.1.13 Регистр аудио данных SIADR

Этот регистр используется для записи аудио отсчетов, передаваемых во внешний аудио-кодек при воспроизведении звука. Этому регистру соответствует FIFO глубиной восемь 32-разрядных слов. Формат записываемых данных зависит от режима МОНО или СТЕРЕО, а также от 8 или 16 бит разрешения:

- для 16-бит стерео данных младшие 16 бит регистра данных соответствуют правому каналу, а старшие – 16 бит левому;
- для 16-разрядного моно режима младшая половина хранит текущий отсчет, а старшая – следующий за ним;
- для 8-разрядного стерео: биты 7:0 – правый канал N, биты 15:8 – правый канал N+1, биты 23:16 – левый канал N, биты 31:24 – левый канал N+1;
- для 8-разрядного моно: биты 7:0 – отсчет N, биты 15:8 – отсчет N+1, биты 23:16 – отсчет N+2, биты 31:24 – отсчет N+3.

При чтении регистра данных, данные считываются из FIFO приемника. FIFO приемника имеет глубину восемь 32-битных слов.

Буфер аудио данных используется как в режиме AC97, так и в режиме I2S. Для корректного формирования данных, передаваемых по интерфейсу I2S, нужно задавать соответствующий формат передаваемых данных.

22.1.14 Регистр модемных данных SIMDR

Этот регистр используется для записи данных, передаваемых во внешний модем. Этому регистру соответствует FIFO глубиной восемь 16-разрядных слов. При чтении данные берутся из FIFO приемника модема, имеющего глубину восемь 16-бит слов. Регистр используется только для работы в режиме AC97.

22.2 Режимы работы

22.2.1 Режим работы AC97

Интерфейс имеет возможность поддержки стандартов AC97(только аудио), MC97(только модем), AMC97(аудио и модем совмещены). AC97 и AMC97 могут работать только как первичные (primary), MC97 может быть сконфигурирован как первичный и как вторичный (secondary). Протокол AC-канала определяет полнодуплексный последовательный интерфейс между внешним кодеком и контроллером. Внешний кодек (или контроллер, если нет первичного кодека в системе) генерирует 12,288 МГц частоту, поступающую в контроллер. В контроллере

входная частота делится на 256, получая, таким образом, 48 КГц звуковую частоту. Она используется контроллером и отсылается в кодек как сигнал SYNC.

Передача данных между кодеком и интерфейсом осуществляется в виде фреймов. Один фрейм содержит 13 слотов (Рисунок 94). Каждый слот (исключая нулевой) содержит 20 бит последовательных данных. Сигнал SYNC определяет начало фрейма. Он активен (высокий уровень) только в течение нулевого слота (начальный слот). Нулевой слот имеет длительность 16 бит. Хотя каждый слот имеет длительность 20 бит, только 16 бит используется.

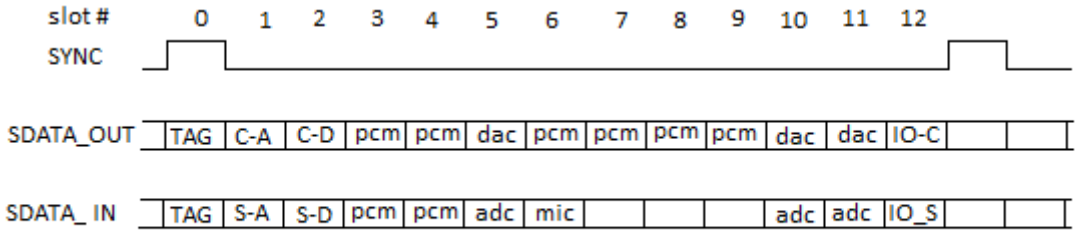


Рисунок 94 – Структура фрейма

Таким образом, один фрейм содержит 256 бит информации (20•12+16). Сигнал SYNC формируется контроллером и имеет частоту следования равную 12,288/256 = 48 КГц. Длительность высокого уровня сигнала SYNC равна 16-ти периодам сигнала синхронизации. Структура тэга фрейма и последовательность бит слотов показана ниже (Рисунок 95).

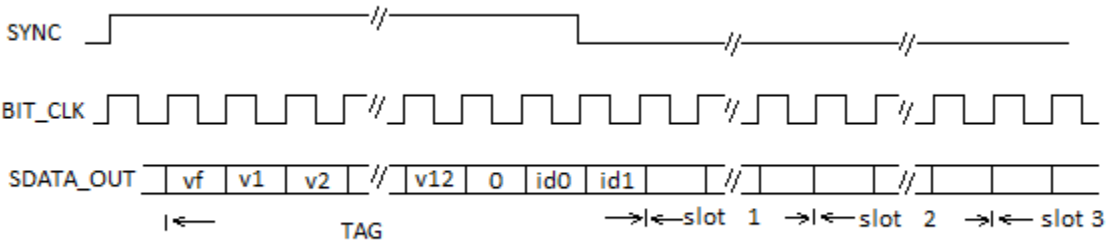


Рисунок 95 – Структура тега

Фрейм имеет бит достоверности – vf. Также имеют биты достоверности и все 12 последующих слотов. Контроллер имеет доступ к регистрам внешнего кодека посредством отсылки адреса регистра в слоте 1 и данных в слоте 2. Аудио данные передаются в 3, 4 и 5 слотах. Имеется возможность управлять GPIO выводами внешнего кодека, а также анализировать их состояние. Для этого используется последний 12 слот. Формат обмена полнодуплексный, т.е. одновременно с передачей по линии SDATA_OUT контроллер принимает данные по линии SDATA_IN. Для ознакомления с подробным описанием каждого слота необходимо изучение спецификации AC97.

Контроллер имеет возможность принимать сообщение о пробуждении (wake-up) внешнего кодека и генерировать соответствующий запрос прерывания.

Назначение выводов микросхемы для AC97 приведено в Таблице 190.

Таблица 190 – Назначение внешних выводов для блоков AC97 (интерфейс 0, интерфейс 1)

Обозначение вывода	Назначение вывода для AC97	Тип	Функциональное назначение
PA[13]	AC97_0_CLK	Вход/Выход	Синхросигнал
PA[19]	AC97_1_CLK		
PA[17]	AC97_0_SYNC	Вход/Выход	Синхронизация фрейма
PA[23]	AC97_1_SYNC		
PA[18]	AC97_0_SDI	Вход/Выход	Входные данные
PA[24]	AC97_1_SDI		
PA[15]	AC97_0_SDO	Вход/Выход	Выходные данные
PA[21]	AC97_1_SDO		
PA[14]	AC97_0_RESET	Вход/Выход	Сброс
PA[20]	AC97_1_RESET		

22.2.2 Режим работы I2S

Интерфейс имеет возможность поддержки стандарта I2S.
Назначение выводов микросхемы для I2S приведено в Таблице 190.

Таблица 191 – Назначение внешних выводов для блоков I2S (интерфейс 0, интерфейс 1)

Обозначение вывода	Назначение вывода для I2S	Тип	Функциональное назначение
PA[13]	SSI0_TCLK	I/O	Интерфейс SSI0. Синхронигнал передатчика
PA[14]	SSI0_TFS	I/O	Интерфейс SSI0. Начало кадра/выбор канала левый-правый
PA[15]	SSI0_TXD	I/O	Интерфейс SSI0. Данные передатчика
PA[16]	SSI0_RCLK	I/O	Интерфейс SSI0. Синхросигнал приемника
PA[17]	SSI0_RFS	I/O	Интерфейс SSI0. Начало кадра/выбор канала левый-правый
PA[18]	SSI0_RXD	I/O	Интерфейс SSI0. Данные приемника
PA[19]	SSI1_TCLK	I/O	Интерфейс SSI1. Синхросигнал передатчика
PA[20]	SSI1_TFS	I/O	Интерфейс SSI1. Начало кадра/выбор канала левый-правый
PA[21]	SSI1_TXD	I/O	Интерфейс SSI1. Данные передатчика
PA[22]	SSI1_RCLK	I/O	Интерфейс SSI1. Синхросигнал приемника
PA[23]	SSI1_RFS	I/O	Интерфейс SSI1. Начало кадра/выбор канала левый-правый
PA[24]	SSI1_RXD	I/O	Интерфейс SSI1. Данные приемника

Предусмотрены два управляющих регистра:
- Регистр управления передатчиком I2S_T_CR;
- Регистр управления приемником I2S_R_CR.

22.2.2.1 Регистр управления передатчиком I2S_T_CR

Таблица 192 – Биты регистра управления передатчиком I2S_T_CR

Бит	Имя	Функция
0	TEN	Разрешение работы (1)
1	MODE	0 – I2S режим 1 – DSP режим
2	SONY	Выбор стандарта I2S : SONY(1) или Philips (0) Имеет смысл только в режиме I2S
3	MS	Master (1) or slave(0) Бит определяет того, кто генерирует синхросигнал и строб начала фрейма (левый-правый)
9:4	DSS	Длина передаваемых данных – от 1 до 64 бит. Значение в битах DSS всегда на 1 меньше количества бит передаваемых данных.
10	PNIS	Определяет момент приема выбора канала: 0 – прием по положительному фронту (из 0 в 1); 1 – прием по отрицательному фронту
11	PNOS	Определяет момент выдачи данных и выбора канала: 0 – выдача по положительному фронту (из 0 в 1); 1 – выдача по отрицательному фронту.
12	SWHW	1 – указывает на необходимость перестановки 16-разрядных полуслов в одном слове
13	PACKH	Имеет смысл при длине передаваемых данных 16 бит и меньше. 1 – указывает на то, что в одном 32-разрядном слове упакованы два значения. Иначе в одном слове хранится одно значение.
14	LRSP	1 и установлен DSP-режим– указывает на то, что общее отсылаемое слово состоит из двух симметричных частей (левой и правой) которые размещаются в разных 32-бит словах. 0 – указывает на то, что передаваемая последовательность не делится на части и представляет собой непрерывный поток бит
15		

Регистр передатчика имеет длину 32 бита. Выдача бит происходит всегда «старший бит первый». При передаче данных, длина которых меньше 32 бит, данные необходимо выровнять влево. Если данные упакованы в полуслова, то при передаче данных длиной менее 16 бит их необходимо выровнять влево до границы полуслова.

В режиме I2S передатчик не может передавать данные длиной более 32 бит. При включенном передатчике передача данных инициируется импульсом начала фрейма (DSP режим) или изменением выбора канала (I2S режим). В случае если новые данные отсутствуют в буфере передатчика, происходит повторная передача последних переданных данных.

В режиме DSP можно передавать данные длиной более 32 бит. При этом эти данные могут состоять из равных половин (левой и правой), если LRSP == 1, или представлять собой непрерывный поток бит расположенных в последовательных словах.

Ниже представлен пример передачи 4-разрядных данных в режиме мастера (Рисунок 96).

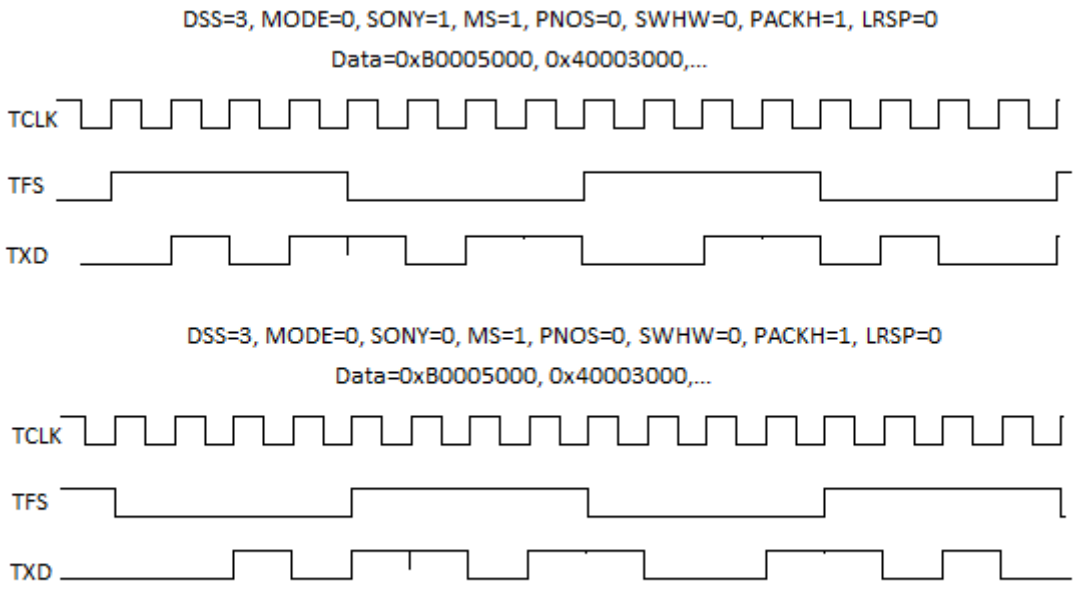


Рисунок 96 – Передача 4-разрядных данных

Передача происходит в режиме I2S (бит MODE=0). Линия TFS указывает на левый или правый канал передачи. Вся информация выдается относительно положительного фронта синхросигнала. Видим, что данные упакованы по 16 бит в одно 32-разрядное слово и выравнены по направлению к старшему биту. Основное отличие между режимами SONY и Philips состоит в том, что в режиме Philips данные выдаются с задержкой на один такт относительно импульса начала выбора канала. Если установить бит PNOS в 1, выдача сигналов выбора канала и данных будет происходить относительно отрицательного фронта синхросигнала. Пример приведен ниже (Рисунок 97).

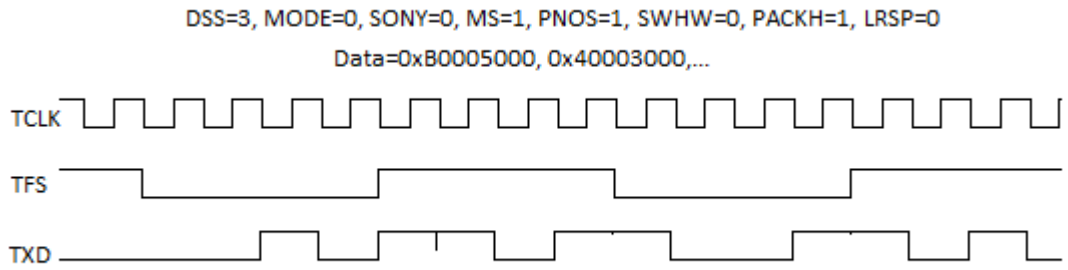


Рисунок 97 – Выдача информации по отрицательному фронту TCLK

Это позволяет более гибко адаптироваться при подключении к приемнику информации и обеспечить надежное время предустановки и удержания данных.

Установка бита SWHW в 1 вызовет перестановку полуслов в слове данных. Пример приведен ниже (Рисунок 98). Данный прием позволяет поменять местами левый и правый каналы.

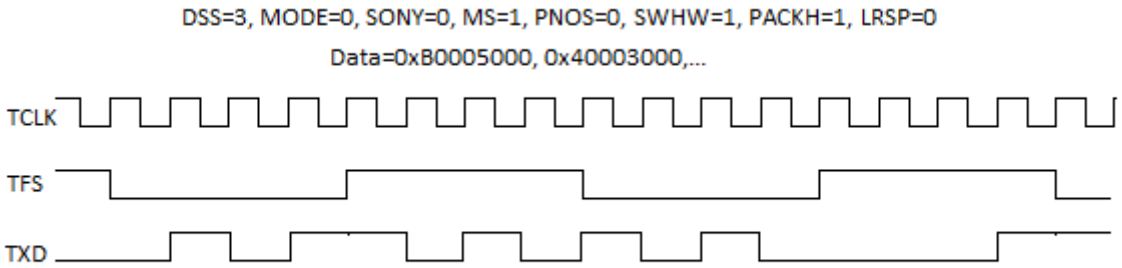


Рисунок 98 – Перестановка полуслов в слове

Интерфейс может работать в модифицированном режиме или режиме DSP. Основное отличие данного режима состоит в формировании импульса TFS. В режиме DSP он не переключается между правым и левым каналом, а импульсом информирует о начале нового фрейма. Пример приведен ниже (Рисунок 99).

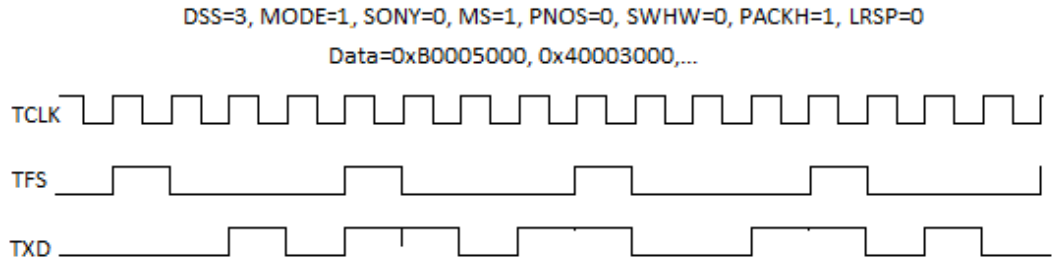


Рисунок 99 – Режим передачи DSP

Импульс на выходе TFS указывает, что в следующем такте начинается выдачи информации нового фрейма. Каждые 4 бита данных сопровождаются импульсом начала фрейма (Рисунок 99). В данном случае уже нет деления на левый и правый каналы. Однако если необходимо передать информацию, состоящую из двух частей, то возможны два способа. Первый состоит в упаковке двух половин информации в одно 32-разрядное слово и передаче как единого 32-разрядного фрейма. Второй способ может использоваться, если две порции информации находятся в последовательных словах, следующих друг за другом. В этом случае необходимо установить бит LRSP в 1. Для нашего примера передачи данных временная диаграмма представлена ниже (Рисунок 100).

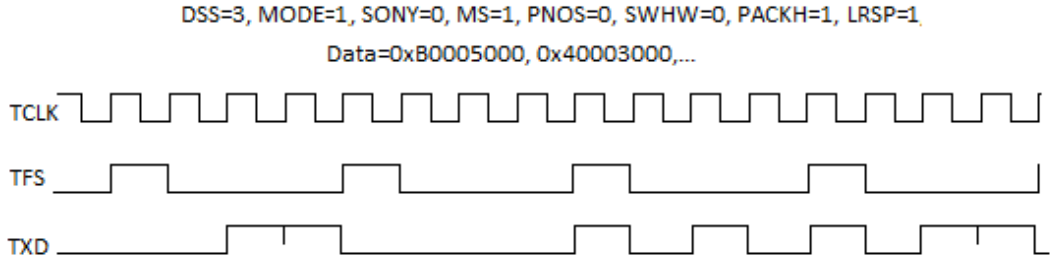


Рисунок 100 – Режим передачи смежными словами

В случае, когда информация находится в смежных словах (в нашем случае в смежных полусловах, т.к. длина данных меньше 16 бит и установлен признак PACK) количество передаваемых бит делится пополам и сначала передаются биты одной половины, а затем второй. В нашем случае передается 4 бита, поэтому сначала будут переданы два старшие бита младшего полуслова (биты 01b от значения 5), а затем два старшие бита старшего полуслова (биты 10b от значения B). Данный режим обычно используется, если длина объединенной посылки превышает 32 бита и в одном слове невозможно упаковать две передаваемые части.

Бит MS в регистре управления определяет источник формирования сигнала синхронизации и сигнала выбора канала (строб начала фрейма). Если бит установлен в 1, передатчик формирует данные сигналы. В противном случае внешнее устройство подает сигналы синхронизации и выбора канала. Последний случай обычно используется, если внешнее устройство требует формирования заданной частоты, которая не может быть получена в процессоре путем деления частоты SOC-шины на константу. Отметим, что в случае работы в режиме подчиненного (MS=0) интерфейс принимает сигнал выбора канала. В этом случае имеет значение бит PNIS, т.к. он задает момент приема достоверного значения сигнала выбора канала.

22.2.2.2 Регистр управления приемником I2S_R_CR

Таблица 193 – Биты регистра управления приемником I2S_R_CR

Бит	Имя	Функция
0	REN	Разрешение работы (1)
1	MODE	0 – I2S режим 1 – DSP режим
2	SONY	Выбор стандарта SONY(1) или Philips (0) Имеет смысл только в режиме I2S
3	MS	Master (1) or slave(0) Бит определяет того кто генерирует синхросигнал и строб начала фрейма (выбор канала левый-правый)
9:4	DSS	Длина передаваемых данных – от 1 до 64 бит
10	PNIS	Определяет момент приема данных и выбора канала. Если 0 – прием по положительному фронту (из 0 в 1) в противном случае по отрицательному
11	PNOS	Определяет момент выдачи выбора канала. Если 0 – выдача по по положительному фронту (из 0 в 1) в противном случае по отрицательному
12	SWHW	1 – указывает на необходимость перестановки 16-разрядных полуслов в одном слове
13	R_INIT	В случае приема сообщения длина которого меньше 16 бит, либо больше 16 бит но меньше 32, возможна предварительная инициализация сдвигового регистра. Регистр инициализируется значением которое зависит от бита SEXT. Функция инициализации работает если бит установлен.
14	LRSP	1 и установлен DSP режим – общее принимаемое слово состоит из двух симметричных частей (левой и правой), которые необходимо принять раздельно в разные 32-разрядные слова. 0 – принимаемая последовательность не делится на части и представляет собой непрерывный поток бит
15	SEXT	Если бит R_INIT равен 1, бит SEXT определяет значение которым инициализируется сдвиговый регистр приемника: 0 – регистр инициализируется нулями 1 – биты регистра инициализируются значением первого принимаемого бита (расширение знака)

В режиме I2S используются FIFO передатчика и приемника аудио данных. Поддержка модемного кодека невозможна.

В случае приема данных в режиме I2S, если длина данных меньше или равна 16 бит, данные будут упаковываться в 32-разрядные слова. При этом возможна перестановка полуслов в слове. В случае длины принимаемых данных больше 16 бит, данные помещаются в 32-разрядные слова. При этом достоверны только DSS+1 младших бит 32 разрядного слова. В случае приема данных, длина которых больше 32 бит, возможны два варианта приема:

1 если LRSP == 0, первые 32 бита будут размещены в первом слове, а оставшиеся биты в следующем слове;
2 если LRSP == 1, принимаемые данные делятся поровну на левую и правую половины, каждая из которых размещается в индивидуальном слове.

Прием данных и выбора слова может происходить как по отрицательному фронту синхросигнала, так и по положительному. Выбор фронта программируется.

Как приемник, так и передатчик в режиме мастер могут выдавать наружу сигнал синхронизации обмена и строб начала обмена. Для задания частоты обмена используется внутренний делитель. Базовой частотой делителя является клок периферийной шины. Значение коэффициента деления находится в регистре SICR2[22:7]. Значение коэффициента деления на 1 больше записываемого значения.

Биты разрешения работы только включают передатчик или приемник. Для того чтобы полностью включить прием или/и передачу необходимо установить соответствующие биты в регистре SICR2[1:0]. Если приемник или передатчик выключен, то все его внутренние регистры сбрасываются в ноль. Очень важным для корректного начала работы может быть необходимость установки FIFO передатчика и приемника в начальное состояние. Это можно сделать битом SICR0[4], который производит сброс всех FIFO интерфейса (указателей и флагов) в исходное состояние. Для сброса бит нужно установить, а затем программно очистить. После этого FIFO передатчика можно заполнить значением 0, если передаваемые данные еще не готовы.

23 Интерфейс I2C

Интерфейс I2C ©Philips использует для обмена две линии:
- SDA (данные);
- SCL (синхронизация).

Назначение внешних выводов микросхемы для I2C приведено в таблице 194.

Таблица 194 – Назначение внешних выводов для I2C

Обозначение вывода	Назначение вывода для I2C	Тип	Функциональное назначение
PC[11]	I2C_SDA	I/O	Интерфейс I2C. SDA
PC[12]	I2C_SCL	I/O	Интерфейс I2C. SCL

Подключение интерфейса I2C к внешним выводам определяется регистром CFG1, подробнее см. подраздел 29.1 «Регистр конфигурации периферийных модулей CFG1».

Для передачи выводов PC, указанных в таблице 194, под управление интерфейса I2C требуется только включить интерфейс битом I2C_ON регистра CR при выключенной альтернативной функции выводов порта PC.

Регистры интерфейса подключены к шине обмена периферийных устройств и имеют базовый адрес **0x800002E0**. Краткое описание регистров приведено в Таблице 195.

Таблица 195 – Регистры интерфейса I2C

Смещение	Название	Выполняемая функция регистра
0	CR	Управление
1	SR	Состояние
2	DR	Данные
3	AR	Адрес
4	VR	Делитель клока
5	CR_set	Установка бит регистра управления
6	CR_clear	Сброс бит регистра управления
7	AXR	Расширение адреса
8	INFO	Информация о внутреннем состоянии интерфейса
9	PR	Значение внешних линий SCL и SDA

Интерфейс использует в качестве базовой частоты частоту обмена по шине периферийных устройств SCLK (частота СОК-шины, равная половине процессорной частоты). Реальная частота обмена получается после деления базовой частоты на 11-разрядный коэффициент деления. Коэффициент деления задается в регистре VR. Таким образом:

$$SCL\_freq = SCLK / (4 \times VR[11:0])$$

Предположим, что частота ядра процессора 200 МГц, частота периферийной шины SCLK 100 МГц. Нам необходимо обеспечить частоту интерфейса SCL_freq равной 100 КГц. Коэффициент деления получаем делением 100 МГц на 100 КГц и дополнительным делением на 4. В результате получаем 250. В регистр VR записываем константу 250.

Интерфейс реализован таким образом, что выполнение обмена предполагает тесное взаимодействие процессора и аппаратной части интерфейса. При этом алгоритм работы зависит от режима, в котором используется интерфейс, а именно:

- работает интерфейс как мастер или как подчиненный;
- используется стандартная 7-битовая адресации или расширенная 10-битовая;
- работает ли интерфейс как единственный мастер на линиях или в мультимастерной системе.

Все особенности, приведенные выше, будут влиять на разработку программной части алгоритма обмена. В общих чертах алгоритм работы интерфейса заключается в следующем:

- пользовательская программа записывает в регистр управления некоторую команду;
- интерфейс выполняет требуемые действия и после их завершения устанавливает флаг прерывания и код состояния;
- пользовательская программа обнаруживает запрос прерывания, считывает регистр состояния и выполняет необходимые действия, после чего очищает флаг запроса прерывания;
- после очистки флага прерывания интерфейс выполняет действия, соответствующие его новому состоянию.

Более подробно особенности различных режимов работы будут описаны в соответствующих подразделах.

23.1 Регистры интерфейса

23.1.1 Регистр управления CR

Регистр задает основные конфигурационные и управляющие параметры. Описание разрядов регистра приведено в Таблице 196.

Таблица 196 – Регистр CR

Бит	Имя	Назначение
0	ABRT	Флаг аварийного (преждевременного) завершения работы интерфейса в режиме мастера. *Устанавливается аппаратно и программно, сбрасывается программно*
1	SLVE	Флаг завершения работы интерфейса в режиме подчиненного. Устанавливается в 1 при работе интерфейса в режиме подчиненного в момент получения условия “stop” при установленном бите SLVE_EN. *Устанавливается аппаратно и программно, сбрасывается программно*
2	AA	Когда равен 1: - разрешает выдачу бита подтверждения в фазе приема данных; - разрешает сравнение адреса в фазе приема адреса
3	SI	Запрос прерывания. Аппаратная установка флага в 1 сопровождается записью некоторого кода в регистр состояния. *Устанавливается аппаратно и программно, сбрасывается программно*
4	STO	Формирование “STOP” условия при работе в режиме мастера. *Устанавливается программно, сбрасывается аппаратно и программно*

5	STA	Формирование “START” условия при работе в режиме мастера
6	I2C_ON	Разрешение работы I2C интерфейса. Если бит равен 0, происходит сброс интерфейса в начальное состояние. Установка бита в 1 разрешает работу интерфейса
7	M10	Режим работы интерфейса при расширенной (10 бит) адресации. Имеет значение только при работе в режиме подчиненного. 0 – используется адрес регистра AR 1 – кроме адреса AR используется дополнительный адрес AXR
8	SLVE_EN	Разрешение генерации события при обнаружении на линиях интерфейса ситуации «СТОП». Имеет смысл только в случае, когда интерфейс работает в режиме подчиненного и активен. 0 – при работе интерфейса ситуация «СТОП» не вызывает генерации запроса прерывания к процессору. 1 – если интерфейс будучи активным подчиненным устройством (к нему в данный момент обращается мастер) обнаруживает на линиях состояние «СТОП», то происходит установка флагов IRQ и SLVE, и запись соответствующего кода в регистр состояния
9	ABN_EN	Разрешение детектирования конфликтов на линии данных при работе в мультимастерной системе. 1 – разрешено 0 – запрещено
10	TLOW_EN	1 – разрешение контролировать параметр T_low при выполнении мастером процедуры “restart” 0 – пользователь самостоятельно контролирует соблюдение параметра T_low при выполнении “restart”
11	IRQ	Запрос прерывания. Доступен по чтению и записи. Устанавливается аппаратно при аппаратной установке флагов SI, ABRT, SLVE. Сбрасывается программно
12	FREE_EN	1 – разрешена генерация прерывания если линии интерфейса готовы для обмена 0 – запрещена
13	-	
14	ABN_S_EN	Разрешение детектирования конфликтов на линиях интерфейса при работе в мультимастерной системе 1 – разрешает анализ нарушений при генерации события «рестарт» и генерацию прерывания 0 – анализ запрещен
15	ABN_P_EN	Разрешение детектирования конфликтов на линиях интерфейса при работе в мультимастерной системе 1 – разрешает анализ нарушений при генерации события «стоп» и генерацию прерывания 0 – анализ запрещен
31-16	-	Не используются. При чтении всегда ноль

Регистр управления CR доступен для записи по трем адресам (смещение относительно базы). При обращении по адресу 0 возможны чтение и загрузка регистра управления. При записи по адресу 5 возможна установка бит регистра управления, а при записи по адресу 6 – очистка бит. Установка или очистка бита происходит, если соответствующий ему бит загружаемого значения равен 1.

Бит I2C_ON управляет включением интерфейса. Запись 1 в данный бит приводит к разрешению работы интерфейса. Если необходимо вернуть интерфейс в исходное состояние, то данный бит нужно сбросить в 0.

Бит STA управляет формированием состояния «СТАРТ» на линиях обмена интерфейса. Установка данного бита приводит к изменениям на линиях SDA и SCL, которые соответствуют состоянию «СТАРТ». После формирования этого состояния на внешних линиях, интерфейс устанавливает флаг прерывания и записывает соответствующий код в регистр состояния. После этого интерфейс переходит в состояние «мастер». При работе в мультимастерной среде установка данного бита в 1 не гарантирует, что интерфейс захватит шину и станет мастером. Другой интерфейс может опередить, и тогда данный интерфейс будет находиться в состоянии ожидания готовности линий интерфейса для начала генерации состояния «СТАРТ». Если процессор получил запрос прерывания с кодом ”start”, необходимо сбросить бит STA до момента записи данных для передачи.

Бит STO управляет формированием состояния «СТОП» на линиях обмена интерфейса. Установка данного бита приводит к изменениям на линиях SDA и SCL, которые соответствуют состоянию «СТОП». После формирования этого состояния на внешних линиях интерфейс устанавливает флаг прерывания и записывает соответствующий код в регистр состояния, а также автоматически сбрасывает бит STO. Установка данного бита должна производиться, только если интерфейс работает в состоянии «мастер».

Бит AA выполняет две функции:

- во время обмена данными между мастером и подчиненным выполняется функция разрешения выдачи подтверждения (acknowledge);
- во время фазы приема адреса выполняются функции управления разрешением сравнения собственного адреса с принятым и выдачи бита подтверждения.

Бит SI является флагом прерывания, который может устанавливаться аппаратурой интерфейса. Значение бита 1 свидетельствует о том, что интерфейсом завершена очередная фаза обмена с кодом в регистре состояния и ожидается реакция со стороны пользователя.

Бит ABN_EN расширяет возможности интерфейса по анализу ситуаций сбоя при работе в мультимастерной системе. В такой системе есть риск захвата шины одновременно несколькими мастерами. В этом случае каждый из мастеров выполняет мониторинг передаваемой информации на линии и реальной информации с линии. В случае несовпадения данных, мастер автоматически отключается и устанавливает бит ABRT. Обязательная процедура анализа распространяется на передачу первого байта адреса. Если два мастера обращаются к различным устройствам, то один из них обязательно отключится. Если возможно обращение нескольких мастеров к одному устройству (адрес одинаковый), то необходимо сравнение передаваемых данных. Это сравнение является дополнительной опцией и включается установкой бита ABN_EN в 1. Сравниваются только данные при выдаче. Выдаваемые биты подтверждения при чтении не сравниваются.

Бит M10 может быть использован только в режиме подчиненного при условии, что обмен происходит с использованием расширенной 10-битовой адресации. Бит остается неизменным в течение всего обмена. Бит, установленный в 1, обязывает интерфейс рассматривать первый байт данных после стартового байта адреса, как дополнительный байт адреса, который будет сравниваться с регистром AXR. Только при их совпадении интерфейс перейдет в режим работы подчиненного и сформирует запрос прерывания к процессору. При сравнении второго байта адреса используется анализ бита направления передачи, заданного в первом принятом адресном байте. Сравнение выполняется, если бит указывает на запись. При выполнении операции чтения в режиме M10, мастер предварительно передает

первый байт адреса и признак записи, затем передается второй байт адреса. После этого выполняется рестарт и передача первого байта адреса, но уже с признаком чтения. Подчиненный отслеживает тот факт, что чтению предшествовала процедура идентификации. В противном случае подчиненный не отвечает на первый байт адреса, хотя он и совпадает.

Бит TLOW_EN может быть использован при выполнении процедуры “restart”. В этом случае после записи некоторого байта в подчиненное устройство, процессор установит бит STA и будет ожидать прерывания по событию “restart”. В случае если скорость обмена по шине низкая, а реакция на прерывания у процессора быстрая, то может возникнуть проблема, что перед выполнением “restart” на линии SCL не выдержан низкий уровень требуемой длительности. Установка данного бита в 1 позволяет на аппаратном уровне гарантировать соблюдение данного временного параметра. Бит необходимо устанавливать только в случае реальной необходимости. Если процедура обработки прерывания тратит достаточное время на обработку прерывания (больше времени T_low), использование данного бита только замедлит генерацию события “restart”.

Бит IRQ – запрос прерывания. Бит доступен для чтения и записи. Дополнительно бит устанавливается аппаратно при установке флагов SI, ABRT и SLVE. При обслуживании прерывания данный бит должен быть сброшен как можно раньше. Бит SI выполняет дополнительную функцию по удержанию линии SCL в нуле и поэтому должен быть сброшен только после обслуживания интерфейса.

23.1.2 Регистр состояния SR

Регистр отражает информацию о событии (код события), которое вызвало запрос прерывания к процессору. Описание разрядов регистра приведено в Таблице 197.

Таблица 197 – Регистр SR

Биты	Значение	Описание
4:0	0x1F	Значение после сброса
	0x01	Мастер сформировал условие “stop”. Интерфейс завершил работу в режиме мастера
	0x02	Мастер сформировал условие “start”. Интерфейс начал работу в режиме мастера
	0x03	Мастер сформировал условие “restart”. Интерфейс продолжает работу в режиме мастера
	0x04	Подчиненный обнаружил со стороны мастера обращение для записи по адресу 0. Режим широковещательной записи во все подчиненные устройства
	0x05	Подчиненный обнаружил условие «stop». Интерфейс закончил работу в режиме подчиненного
	0x09	Интерфейс, работая в состоянии «подчиненный», обнаружил на шине «свой» адрес, а также запрос на запись, после чего ответил битом подтверждения ACK = 1
	0x0B	Интерфейс, работая в состоянии «подчиненный», обнаружил на шине «свой» адрес, а также запрос на чтение, после чего ответил битом подтверждения ACK = 1
	0x0C	Мастер отослал первый байт, получил в ответ бит ACK=0 и ждет записи данных для передачи
	0x0D	Мастер отослал первый байт, получил в ответ бит ACK=1 и ждет записи данных для передачи
	0x0E	Мастер отослал первый байт, получил в ответ бит ACK=0 и ждет сброса флага прерывания для начала чтения данных из внешнего устройства

	0x0F	Мастер отослал первый байт, получил в ответ бит ACK=1 и ждет сброса флага прерывания для начала чтения данных из внешнего устройства
	0x10	«Подчиненный» принял байт данных и ответил битом подтверждения ACK = 0
	0x11	«Подчиненный» принял байт данных и ответил битом подтверждения ACK = 1
	0x12	«Подчиненный» отослал байт данных и принял бит подтверждения ACK = 0
	0x13	«Подчиненный» отослал байт данных и принял бит подтверждения ACK = 1
	0x14	Мастер отослал байт данных и принял бит АСК=0
	0x15	Мастер отослал байт данных и принял бит АСК=1
	0x16	Мастер принял байт данных и отослал бит АСК=0
	0x17	Мастер принял байт данных и отослал бит АСК=1
	0x1E	Шина свободна
5	-	Не используются. При чтении всегда ноль
6	ABRT	Бит 0 регистра управления. Информирует об аварийном (преждевременном) завершении работы мастера
7	SLVE	Бит 1 регистра управления. Информирует об обнаружении ситуации “stop” при работе подчиненного. Дополнительно в битах 4:0 устанавливается код 5
31:8	-	Резерв. При чтении всегда ноль

Значение регистра состояния устанавливается только аппаратно. Невозможно изменить это значение программно. Биты регистра доступны только по чтению. Если интерфейс выключен, значение регистра состояния равно 0x1F.

23.1.3 Регистр данных DR

Посредством этого регистра происходит запись данных для передачи и чтение данных после приема. Однако необходимо помнить, что запись новых данных в регистр возможна только тогда, когда значение регистра состояния указывает на ожидание этого действия. То же самое относится и к чтению данных. Регистр является непосредственно тем сдвиговым регистром, из которого на линию SDA передаются (старший бит первым) и принимаются (принятый бит поступает в младший бит регистра) данные.

23.1.4 Регистр адреса AR

Регистр содержит адрес устройства для работы в режиме подчиненного. При работе в расширенном режиме 10-битовой адресации регистр содержит два старших бита адреса и специальную кодовую комбинацию. Описание разрядов регистра приведено в Таблице 198.

Таблица 198 – Регистр AR

Бит	Имя	Описание
0	AZE	Разрешение доступа к «подчиненному» по нулевому адресу (когда бит равен 1)
7:1	SA	Адрес устройства в режиме «подчиненный»
31:8	-	Резерв. При чтении всегда ноль

Биты SA регистра хранят адрес устройства при функционировании интерфейса I2C в режиме подчиненного. В этом случае обращение к процессору посредством I2C интерфейса возможно только при совпадении адреса в регистре AR с принятым адресом.

Бит AZE дает дополнительную возможность при работе в режиме подчиненного. Если бит равен 1, возможно обращение к интерфейсу по адресу равному нулю.

При работе в режиме 10-битовой адресации регистр должен хранить двоичный код 1111_0aaz.

Где биты «aa» это старшие биты 10-битового адреса. Младшие 8 бит адреса должны храниться в регистре AXR.

23.1.5 Регистр делителя частоты VR

Регистр коэффициента деления внутреннего делителя частоты. Формула для получения рабочей частоты была приведена выше. Разряды регистра описаны в Таблице 199.

Таблица 199 – Регистр VR

Бит	Имя	Описание
11:0	DIV	Коэффициент деления для формирования частоты SCL
15:12	-	При чтении всегда 0
23:16	DSUP	Количество тактов предустановки бита данных относительно SCL. Реальное время предустановки будет равно DSUP+2
31:24	-	При чтении всегда 0

23.1.6 Регистр состояния линий интерфейса PR

Регистр позволяет прочитать текущее состояние линий обмена интерфейса. Регистр доступен только по чтению. Описание разрядов регистра приведено в Таблице 200.

Таблица 200 – Регистр PR

Бит	Имя	Описание
0	SDA_p	Значение вывода SDA
1	SCL_p	Значение вывода SCL
31:2	-	Всегда 0

23.1.7 Регистр адреса AXR

Регистр содержит младшие 8 бит адреса устройства для работы в режиме подчиненного и в режиме 10-битовой адресации. Доступны для чтения и записи только 8 младших бит регистра. Старшие биты при чтении всегда равны нулю.

23.1.8 Регистр INFO

Регистр доступен только для чтения и отражает состояние внутренней аппаратуры интерфейса. Биты могут быть полезны при проверке работы

интерфейса, а также (возможно) и при реализации алгоритмов обмена. Описание разрядов регистра приведено в Таблице 201.

Таблица 201 – Регистр INFO

Бит	Имя	Описание
0	I2C_ready	Линии SDA и SCL находятся в высоком уровне. 0 – на одной из линий интерфейса 0
1	SCLм	Логический уровень внутреннего сигнала синхронизации, который мастер выдает на линию SCL
2	SDA_mg	Значение линии SDA. Берется мажоритарно по трем последним отсчетам
3	SCL_mg	Значение линии SCL. Берется мажоритарно по трем последним отсчетам
4	Master	1 – интерфейс работает в режиме мастера
5	Slave	1 – интерфейс работает в режиме подчиненного
6	ST_BY	1 – интерфейс выполняет передачу (если мастер) или прием (если подчиненный) первого байта сообщения (адрес). Бит устанавливается в 1 при обнаружении ситуации “start”. Сбрасывается в ноль после завершения работы с первым байтом
7	R_W	Бит указывает на направление передачи данных при обмене. 0 – в режиме мастера указывает на передачу данных; в режиме подчиненного – на прием. 1 – обратное направление передачи. Значение бита достоверно только после приема-передачи первого байта сообщения
11:8	BIC	Внутренний счетчик, указывающий на текущий номер бита передаваемого на линиях байта сообщения
12	BUSY	1 – интерфейс занят. Бит устанавливается событием “start” и сбрасывается событием “stop”
13	STX_BY	Бит устанавливается в 1 только при режиме работы с 10-битовой адресацией. 1 – интерфейс в режиме подчиненного выполняет прием второго байта сообщения (расширенный адрес). Бит устанавливается в 1 при обнаружении ситуации совпадения принятого адреса с регистром AR. Сбрасывается в ноль после завершения приема второго байта адреса
14	SLV_EXE	1 – интерфейс работает в режиме подчиненного. Отличие данного бита от флага Slave состоит в том, что данный флаг сбрасывается в ноль при выполнении *чтения* данных из подчиненного при получении бита подтверждения ACK=0. Флаг Slave сбросится только при получении ситуации “stop”
15	SLV_XM	Бит устанавливается в 1 только при режиме работы с 10-битовой адресацией. 1 – интерфейс в режиме подчиненного принял полный 10-битовый адрес и он совпал с адресом интерфейса. Бит сбрасывается в ноль при получении ситуации “stop” или когда бит M10 регистра управления равен 0
16	M_EXE	1 – интерфейс работает в режиме мастера. Отличие от флага master состоит в том, что данный флаг не сбрасывается перед выполнением события “restart”
31:17	-	Всегда 0

23.2 Синхронизация интерфейса

Интерфейс использует для своей синхронизации частоту шины периферийных устройств. Для получения требуемых стандартных частот передачи на линии SCL используется программируемый делитель. Регистр VR задает коэффициент деления частоты. Формула расчета была приедена ранее (см. стр. 337). Делитель частоты используется только при работе в режиме мастера и формирует требуемую частоту на линии SCL. Информация с линий SCL и SDA не используется интерфейсом напрямую, а защелкивается в буфере, тактируемым частотой периферийной шины. Имеется ограничение на минимальное значение коэффициента деления, он должен быть не менее 3. При значении меньше 3 интерфейс в режиме мастера не работает. При минимальном значении 3 итоговый коэффициент деления частоты равен 12. Таким образом, теоретически максимально достижимая частота обмена:

$$SCL\_max = SCLK/12.$$

На практике достижение такой частоты обмена зависит от номинала внешнего резистора на линиях SDA и SCL. Корректная работа на максимально возможной частоте возможна, только если длительность фронта перехода из 0 в 1 на линиях SCL и SDA не превышает длительности периода частоты SCLK. В противном случае коэффициент деления необходимо увеличивать. Также имеет значение допустимая скорость обмена других устройств на шине. При необходимости подчиненное устройство должно успевать тормозить мастера.

При выдаче информации на линию SDA интерфейс гарантирует минимальное время удержания предыдущего значения данных после изменения линии SCL из 1 в 0. Это время равно четырем тактам частоты синхронизации периферийной шины SCLK.

Время предустановки данных на линии SDA до перехода линии SCL из 0 в 1 зависит от того, какой бит данных выдается:

- Если выдается *не первый бит*, время предустановки будет равно половине периода SCL минус 4 такта SCLK;
- Если выдается *первый бит*, процессор для выдачи данных сначала запишет байт данных в регистр данных DR для передачи. Через такт SCLK с момента записи эти данные появятся на линии SDA. Переход линии SCL из нуля в единицу после записи данных зависит от нескольких факторов. Например, при работе в режиме подчиненного линия SCL будет удерживаться подчиненным до момента очистки бита SI регистра управления. Если процессор оперативно среагировал на прерывание, быстро произвел запись новых данных и сбросил флаг SI, оставшееся время (с момента сброса SI до момента, когда мастер начнет выдавать 1 на линию SCL) будет составлять время предустановки данных. Если же процессор поздно среагировал на прерывание (или частота обмена высокая), и мастер уже готов выдать на SCL единицу, но подчиненный его удерживает, время предустановки будет равно количеству тактов от момента записи данных до момента сброса флага SI (минус 1). Например, если записать данные и сразу же за этим сбросить флаг, время предустановки будет равно нулю.

Возможны два способа управления временем предустановки первого бита данных:

- программный;
- аппаратный.

При программном способе рекомендуется после записи данных произвести чтение какого-нибудь регистра интерфейса, а затем сбросить флаг SI. Это даст как минимум 5 тактов предустановки. Поскольку пользователь чаще всего не может знать, сколько времени прошло с момента генерации запроса прерывания, он всегда самостоятельно должен обеспечивать требуемое время предустановки первого выдаваемого бита. Например, стандарт I2C в типовом режиме работы требует минимальное время предустановки равное 250 нс. Если процессор работает на частоте 200 МГц, чтобы обеспечить требование стандарта, необходимо после записи данных для выдачи сбросить бит SI не ранее, чем через 250нс / 5нс = 50 тактов (или 25 тактов SCLK).

Возможен также аппаратный способ управления временем предустановки первого выдаваемого на линию SDA бита. Регистр VR имеет 8-разрядное поле DSUP. Если значение этого поля не равно нулю – включается механизм автоматического обеспечения времени предустановки. Интерфейс (в режиме мастер или подчиненный) отслеживает факт того, что флаг SI установлен и произведена запись в регистр данных для передачи. Затем интерфейс начинает обратный отсчет задержки, величина которой задана в поле DSUP. Если бит SI был сброшен раньше, чем истек интервал времени удержания, интерфейс притормозит линию SCL. Если бит SI будет сброшен после истечения заданного интервала времени, линия SCL будет освобождена сразу же после сброса SI. Данный механизм позволяет освободить пользователя от необходимости следить за соблюдением времени предустановки данных.

При работе в режиме подчиненного интерфейс после приема (или передачи) «своего» байта данных сформирует для процессора запрос прерывания и будет удерживать линию SCL в нуле. Линия будет удерживаться до тех пор, пока процессор не обслужит интерфейс и не сбросит бит SI. Отметим, что случайная установка пользователем бита SI при работе интерфейса в режиме подчиненного немедленно приведет к переводу линии SCL в ноль.

При работе в режиме мастера интерфейс самостоятельно задает синхронизацию на линии SCL. При этом программируемый делитель интерфейса постоянно отслеживает, не удерживает ли какое-либо из устройств линию SCL в низком уровне. В этом случае делитель останавливается. Таким образом, подчиненное устройство может притормаживать мастера при передаче любого бита сообщения.

23.3 Режимы работы интерфейса

Интерфейс может работать в различных режимах:

- мастер в одномастерной системе;
- мастер в мультимастерной системе;
- мастер в системах с 7-ми или 10-битовой адресацией;
- подчиненный с режимом адресации 7 бит;
- подчиненный с режимом адресации 10 бит.

Для любого режима работы перед началом использования блока необходимо соответствующим образом назначить выводы общего назначения на выводы SCL и SDA посредством регистра CFG1 (см. раздел 29 «Модуль управления синхронизацией и энергопотреблением»).

В каждой из данных ситуаций будет свой особенный алгоритм работы. Поскольку интерфейс предполагает тесное программное управление, часть задачи по выполнению обмена на шине ложится на процессор.

Рассмотрим возможные примеры алгоритмов работы в различных режимах. Действия интерфейса будут выделены курсивом. Предположим, что в регистре VR заранее запрограммирован коэффициент деления частоты для получения рабочей частоты интерфейса, поэтому данный регистр далее не берется во внимание.

23.3.1 Мастер в одномастерной системе с режимом адресации 7 бит. Запись

Действия пользователя для реализации процедуры обмена данными с некоторым устройством могут быть следующими:

А) Включение интерфейса посредством установки бита I2C_ON регистра управления. Разрешение обработки запроса прерывания от интерфейса в контроллере прерываний.

Б) Анализ того, что линии SCL и SDA имеют высокий уровень. Установка бита STA для генерации события “start” на линиях интерфейса. Ожидание прерывания.

Интерфейс выполняет процедуру генерации события “start” на линиях SCL и SDA. Когда событие происходит, интерфейс устанавливает бит SI и в регистр состояния записывает код 0x02. Формируется запрос на прерывание к процессору.

В) Получение запроса на прерывание. Сброс бита IRQ.
Анализ того, что произошло событие “start”. Сброс бита STA.
Запись в регистр данных DR значения адреса устройства, к которому будет выполняться обращение. Бит 0 регистра должен быть равен нулю, что указывает на запись в устройство.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет передачу 8-ми бит данных и принимает от устройства бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x0C или 0x0D. Значение кода зависит от бита подтверждения. Формируется запрос на прерывание к процессору.

Г) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Если бит АСК=0, обнаружена ошибка, и устройства нет. Если АСК=1, устройство есть и можно передавать данные для записи.
Запись в регистр данных DR значения байта данных для передачи.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет передачу 8-ми бит данных и принимает от устройства бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x14 или 0x15. Значение кода зависит от бита подтверждения. Формируется запрос на прерывание к процессору.

Д) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Если бит АСК=0, устройство больше не может принять данные. Если АСК=1, устройство может принять следующие данные.
Предположим, что больше не нужно передавать данные. Производим установку бита STO для генерации события “stop”.

Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет генерацию события “stop”. Устанавливается бит SI, а в регистр состояния записывается код 0x01. Формируется запрос на прерывание к процессору.

Е) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Выключение интерфейса. Завершение алгоритма работы.

Каждый раз, когда устанавливается бит SI, линия синхронизации SCL удерживается в низком уровне. Как только бит SI сбрасывается, выдача синхросигнала возобновляется.

23.3.2 Мастер в одномастерной системе с режимом адресации 7 бит. Чтение

Действия пользователя для реализации процедуры обмена данными с некоторым устройством могут быть следующими:

А) Включение интерфейса посредством установки бита I2C_ON регистра управления. Разрешение обработки запроса прерывания от интерфейса в контроллере прерываний.

Б) Анализ того, что линии SCL и SDA имеют высокий уровень.
Установка бита STA для генерации события “start” на линиях интерфейса.
Ожидание прерывания.

Интерфейс выполняет процедуру генерации события “start” на линиях SCL и SDA. Когда событие происходит, интерфейс устанавливает бит SI и в регистр состояния записывает код 0x02. Формируется запрос на прерывание к процессору.

В) Получение запроса на прерывание. Сброс бита IRQ.
Анализ того, что произошло событие “start”. Сброс бита STA.
Запись в регистр данных DR значения адреса устройства, к которому будет выполняться обращение. Бит 0 регистра должен быть равен единице, что указывает на чтение данных из устройства.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет передачу 8-ми бит данных и принимает от устройства бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x0E или 0x0F. Значение кода зависит от бита подтверждения. Формируется запрос на прерывание к процессору.

Г) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Если бит АСК=0, обнаружена ошибка, и устройства нет. Если АСК=1, устройство есть и можно читать данные из устройства.
Устанавливаем бит AA в регистре управления, если необходимо будет прочитать более, чем один байт. Иначе бит AA = 0.

Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет чтение 8-ми бит данных из устройства и отсылает в устройство бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x16 (AA=0) или 0x17(AA=1). Значение кода зависит от бита подтверждения. Формируется запрос на прерывание к процессору.

Д) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Если бит АСК=0, дальнейшее чтение данных не нужно. Если АСК=1, интерфейс должен принять следующий байт данных.
Читаем принятые данные из регистра DR.
Если необходимо еще прочитать данные – сбрасываем бит SI и переходим к ожиданию прерывания. Перед чтением последнего байта обязательно нужно очистить бит AA.
Предположим, что дальнейшее чтение данных не нужно. Производим установку бита STO для генерации события “stop”.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет генерацию события “stop”. Устанавливается бит SI, а в регистр состояния записывается код 0x01. Формируется запрос на прерывание к процессору.

Е) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Выключение интерфейса. Завершение алгоритма работы.

Каждый раз, когда устанавливается бит SI, линия синхронизации SCL удерживается в низком уровне. Как только бит SI сбрасывается, выдача синхросигнала возобновляется.

В ряде случаев при обмене с внешним устройством возникает необходимость сначала записать данные в устройство (к примеру, внутренний указатель адреса), а затем перейти к чтению данных. При этом между записью и последующим чтением не должно быть состояния «СТОП». Эта задача решается следующим образом. После записи последнего байта, когда интерфейс установил флаг прерывания SI, пользователь должен установить бит STA в регистре управления и сбросить флаг прерывания SI. Подобная последовательность действий приводит к возникновению ситуации «РЕСТАРТ» и после нее можно снова передать адрес устройства с битом направления соответствующим чтению данных.

23.3.3 Подчиненный с режимом адресации 7 бит. Прием данных

Действия пользователя для реализации процедуры работы процессора как подчиненного устройства на шине I2C могут быть следующими:

А) Программирование адреса устройства в регистре AR. Установка бита AA в регистре управления для разрешения выдачи подтверждения при обнаружении совпадения адреса.
Включение интерфейса посредством установки бита I2C_ON регистра управления.

Разрешение обработки запроса прерывания от интерфейса в контроллере прерываний.
Ожидание прерывания.

Интерфейс ожидает возникновения события “start” на линиях SCL и SDA. Когда событие происходит, интерфейс устанавливает внутренний бит BUSY, ST_BY и ожидает приема первого адресного байта. После приема байта интерфейс выполняет сравнение принятого адреса с адресом в регистре AR. В случае совпадения, интерфейс устанавливает бит SI и в регистр состояния записывает код 0x09. Формируется запрос на прерывание к процессору. Интерфейс удерживает линию SCL в нуле.

Б) Получение запроса на прерывание. Сброс бита IRQ.
Анализ того, что произошло обращение к процессору со стороны мастера по записи.
Если необходимо принять только один байт, пользователь сбрасывает бит AA в ноль. Чтобы принять столько байт данных, сколько может передать мастер, бит AA должен оставаться всегда в 1.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс освобождает линию SCL и позволяет мастеру передать байт данных, после этого отвечает мастеру битом подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x10 или 0x11. Значение кода зависит от бита подтверждения. Формируется запрос на прерывание к процессору. Интерфейс удерживает линию SCL в нуле.

В) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Чтение из регистра данных DR значения байта принятых данных.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс освобождает линию SCL и позволяет мастеру передать следующий байт данных.
Процесс может повторяться многократно.

Для прекращения передачи данных, пользователю необходимо сбросить бит AA, и после приема очередного байта интерфейс ответит ACK=0. Мастер в ответ должен сформировать событие “stop”. Если мастер передал последний байт данных, событие “stop” генерируется независимо от значения бита ACK. Интерфейс в состоянии подчиненного не сопровождает наличие события “stop” генерацией прерывания процессору. Чтобы узнать о завершении обмена не по инициативе пользователя, следует при включении интерфейса установить бит SLVE_EN. Обмен завершен, если на линиях было событие “stop”. В противном случае, узнать о завершении текущего обмена можно только при следующем обращении к подчиненному устройству. Чтобы точно знать момент завершения обмена при использовании бита SLVE_EN, пользователю необходимо после получения прерывания по событию “stop” обработать прерывание и очистить флаг SLVE.

Каждый раз, когда подчиненный устанавливает бит SI, линия синхронизации SCL удерживается в низком уровне. Как только бит SI сбрасывается, разрешение

синхросигнала возобновляется. Рекомендуется стремиться к максимально быстрой реакции на прерывание, чтобы не удерживать линию SCL слишком долго. Так как длительное удерживание может быть расценено мастером, как отказ устройства, и повлечет сброс всей системы.

23.3.4 Подчиненный с режимом адресации 7 бит. Выдача данных

Действия пользователя для реализации процедуры работы процессора как подчиненного устройства на шине I2C могут быть следующими:

А) Программирование адреса устройства в регистре AR. Установка бита AA в регистре управления для разрешения выдачи подтверждения в случае обнаружения совпадения адреса.
Включение интерфейса посредством установки бита I2C_ON регистра управления.
Разрешение обработки запроса прерывания от интерфейса в контроллере прерываний.
Ожидание прерывания.

Интерфейс ожидает возникновения события “start” на линиях SCL и SDA. Когда событие происходит, интерфейс устанавливает внутренний бит BUSY, ST_BY и ожидает приема первого адресного байта. После приема байта интерфейс выполняет сравнение принятого адреса с адресом в регистре AR. В случае совпадения, интерфейс устанавливает бит SI и в регистр состояния записывает код 0x0B. Формируется запрос на прерывание к процессору. Интерфейс удерживает линию SCL в нуле.

Б) Получение запроса на прерывание. Сброс бита IRQ.
Анализ того, что произошло обращение к процессору со стороны мастера для чтения данных. Пользователь должен знать, какой байт данных хочет прочитать мастер. Нужное значение записывается в регистр данных DR для передачи.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс освобождает линию SCL и позволяет мастеру прочитать байт данных. После приема байта, мастер отвечает подчиненному битом подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x12 или 0x13. Значение кода зависит от бита подтверждения. Если бит подтверждения равен 0, интерфейс сбрасывает внутренний флаг SLV_EXE и перестает контролировать выдачу данных на линию SDA. Это позволяет мастеру сформировать событие “stop”. Формируется запрос на прерывание к процессору. Интерфейс удерживает линию SCL в нуле.

В) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Если бит АСК=1, процессор записывает в регистр данных DR следующий байт для передачи. Если бит АСК равен нулю, происходит завершение алгоритма. События “stop” можно не дожидаться.
Производим сброс бита SI регистра управления. Ожидание прерывания, если передача не завершена.

Интерфейс освобождает линию SCL и позволяет мастеру принять следующий байт данных (если чтение продолжается) или сформировать событие “stop”.

23.3.5 Мастер в одномастерной системе с режимом адресации 10 бит. Запись

Данный режим работы идентичен базовому 7-битному режиму. Отличие заключается в том, что за первым адресным байтом необходимо передать второй байт адреса. Для пользователя это будет некоторая модификация алгоритма, для аппаратной части интерфейса второй байт адреса будет выглядеть, как первый байт передаваемых данных.

23.3.6 Мастер в одномастерной системе с режимом адресации 10 бит. Чтение

Процедура чтения в 10-битовом режиме адресации имеет ряд отличий от стандартной процедуры. Также имеются некоторые особенности.

Например, необходимо прочитать два байта данных из устройства, использующего 10 бит адреса. Последовательность действий будет следующей:

А) Включение интерфейса посредством установки бита I2C_ON регистра управления. Разрешение обработки запроса прерывания от интерфейса в контроллере прерываний.

Б) Анализ того, что линии SCL и SDA имеют высокий уровень.
Установка бита STA для генерации события “start” на линиях интерфейса.
Ожидание прерывания.

Интерфейс выполняет процедуру генерации события “start” на линиях SCL и SDA. Когда событие происходит, интерфейс устанавливает бит SI и в регистр состояния записывает код 0x02. Формируется запрос на прерывание к процессору.

В) Получение запроса на прерывание. Сброс бита IRQ.
Анализ того, что произошло событие “start”. Сброс бита STA.
Запись в регистр данных DR значения адреса устройства, к которому будет выполняться обращение. Особенность адреса в том, что в старших 7-ми битах он содержит код 1111_0aa. Биты «aa» это два старших бита 10-разрядного адреса устройства. Бит направления передачи должен быть равен нулю, что указывает на запись в устройство. Это необходимо для того чтобы передать второй байт адреса.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет передачу 8-ми бит данных и принимает от устройства бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x0C или 0x0D. Значение кода зависит от бита подтверждения. Формируется запрос на прерывание к процессору.

Г) Получение запроса на прерывание. Сброс бита IRQ.

Анализ кода состояния. Если бит АСК=0, обнаружена ошибка, и устройства нет. Если АСК=1, устройство есть и можно передавать второй байт адреса. Отметим, что бит АСК=1 могут выставлять несколько устройств использующих 10-битовую адресацию, т.к. старшие два бита адреса у многих устройств могут быть одинаковыми.
Запись в регистр данных DR значения второго байта адреса для передачи.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет передачу 8-ми бит данных и принимает от устройства бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x14 или 0x15. Значение кода зависит от бита подтверждения. Формируется запрос на прерывание к процессору.

Д) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Если бит АСК=0, устройства с таким адресом нет и нужно прекратить обмен. Если АСК=1, устройство есть и с ним можно выполнять обмен данными.
Предположим, что получили АСК=1 и необходимо прочитать данные. Для этого нужно изменить направление передачи данных.
Установка бита STA для генерации события “restart” на линиях интерфейса.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет процедуру генерации события “restart” на линиях SCL и SDA. Когда событие происходит, интерфейс устанавливает бит SI и в регистр состояния записывает код 0x03. Формируется запрос на прерывание к процессору.

Е) Получение запроса на прерывание. Сброс бита IRQ.
Анализ того, что произошло событие “restart”. Сброс бита STA.
Запись в регистр данных DR значения первого байта адреса устройства, к которому будет выполняться обращение. Значение адреса должно быть тем же. Особенность в том, что бит направления передачи должен быть равен единице, что указывает на чтение из устройства. В данном случае должно ответить то устройство, для которого на предыдущих стадиях была выполнена процедура выбора.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет передачу 8-ми бит данных и принимает от устройства бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x0E или 0x0F. Значение кода зависит от бита подтверждения. Формируется запрос на прерывание к процессору.

Ж) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Если бит АСК=0, обнаружена ошибка, и устройства нет. Если АСК=1, устройство есть и можно читать данные из устройства.
Устанавливаем бит AA в регистре управления, если необходимо будет прочитать более чем один байт. Иначе бит AA = 0. В примере необходимо прочитать два байта, поэтому AA=1.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет чтение 8-ми бит данных из устройства и отсылает в устройство бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x17. Формируется запрос на прерывание к процессору.

З) Получение запроса на прерывание. Сброс бита IRQ. Анализ кода состояния.
Читаем принятые данные из регистра DR.
Сбрасываем бит AA, т.к. следующий прочитанный байт должен быть последним.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет чтение 8-ми бит данных из устройства и отсылает в устройство бит подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x16. Формируется запрос на прерывание к процессору.

И) Получение запроса на прерывание. Сброс бита IRQ. Анализ кода состояния.
Читаем принятые данные из регистра DR. Больше данные приниматься не будут, т.к. передан бит ACK=0.
Производим установку бита STO для генерации события “stop”.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс выполняет генерацию события “stop”. Устанавливается бит SI, а в регистр состояния записывается код 0x01. Формируется запрос на прерывание к процессору.

К) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Выключение интерфейса. Завершение алгоритма работы.

Каждый раз, когда устанавливается бит SI, линия синхронизации SCL удерживается в низком уровне. Как только бит SI сбрасывается, выдача синхросигнала возобновляется.

23.3.7 Подчиненный с режимом адресации 10 бит. Прием данных

В данном случае процессор на шине I2C – подчиненное устройство, к которому возможно обращение по адресу. Для идентификации устройства должен использоваться 10-битовый адрес. Алгоритм работы устройства, в случае записи в него одного байта данных, следующий:

А) Программирование адреса устройства в регистре AR. При использовании стандартной адресации в AR должен быть код 1111_0aa0, где биты «aa» это два старших бита нашего 10-битового адреса.
Запись в регистр AXR младших 8-ми бит адреса.
Установка бита M10 для задания режима работы с адресом 10 бит.
Установка бита AA в регистре управления для разрешения выдачи подтверждения в случае обнаружения совпадения адреса.

Установка бита SLVE_EN, если необходимо получить прерывание после завершения обмена при возникновении события “stop”.
Включение интерфейса посредством установки бита I2C_ON регистра управления.
Разрешение обработки запроса прерывания от интерфейса в контроллере прерываний.
Ожидание прерывания.

Интерфейс ожидает возникновения события “start” на линиях SCL и SDA. Когда событие происходит, интерфейс устанавливает внутренний бит BUSY, ST_BY и ожидает приема первого адресного байта. После приема байта интерфейс выполняет сравнение принятого адреса с адресом в регистре AR. В случае совпадения первого байта адреса, интерфейс отвечает битом подтверждения ACK=1, устанавливает признак slave, а также устанавливает бит STX_BY, указывающий на то, что интерфейс ожидает второй байт адреса.

При получении следующего байта данных интерфейс проверяет принятый байт на совпадение с регистром AXR. Если адрес не совпадает, интерфейс переходит в исходное состояние. В случае совпадения, интерфейс устанавливает бит SI и в регистр состояния записывает код 0x09. Также устанавливается внутренний флаг SLV_XM. Формируется запрос на прерывание к процессору. Интерфейс удерживает линию SCL в нуле.

Б) Получение запроса на прерывание. Сброс бита IRQ.
Анализ того, что произошло обращение к процессору со стороны мастера по записи.
В данном случае принимается только один байт, поэтому бит AA сбрасываем в ноль. Если нужно принять столько байт данных, сколько может передать мастер, то бит AA должен оставаться всегда в 1.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс освобождает линию SCL и позволяет мастеру передать байт данных, после этого отвечает мастеру битом подтверждения (в нашем примере ACK=0). Устанавливается бит SI, а в регистр состояния записывается код 0x10. Формируется запрос на прерывание к процессору. Интерфейс удерживает линию SCL в нуле. Т.к. интерфейсом передан бит ACK=0, данные больше приниматься не будут. Интерфейс сбрасывает внутренний флаг SLV_EXE, хотя флаг slave будет оставаться в 1.

В) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Чтение из регистра данных DR значения байта принятых данных.
Производим сброс бита SI регистра управления.
Т.к. мастеру уже передана информация о завершении обмена, можно не анализировать другие события. Но если установлен бит SLVE_EN, можно узнать, когда мастер сформирует событие “stop”. В этом случае необходимо перейти к ожиданию прерывания.

Интерфейс освобождает линию SCL и позволяет мастеру завершить обмен. Мастер может пытаться передавать еще байты данных, однако интерфейс будет отвечать АСК=0, не будет генерировать прерывание к процессору и не будет тормозить SCL линию. Если бит SLVE_EN установлен, интерфейс будет ждать события “stop” и сформирует запрос на прерывание.

Можно отметить следующую особенность использования бита SLVE_EN. При работе в режиме подчиненного бит SLVE_EN лучше устанавливать в момент, когда обнаружено обращение к нужному устройству по записи. Даже если на прием байта последовал ответ АСК=1, мастер может прекратить обмен по своей инициативе. Тогда мы узнаем быстрее, что текущий обмен завершен. Также существует вероятность, что мастер прервет обмен в любой момент (даже не закончив передачи байта). Бит SLVE_EN можно сбросить в ноль после приема байта данных, на который последовал ответ АСК=0 (т.е. мастер проинформирован о завершении обмена). Если бит SLVE_EN используется для того, чтобы точно знать момент завершения обмена, то после получения прерывания по событию “stop”, пользователь должен обработать прерывание и очистить флаг SLVE.

23.3.8 Подчиненный с режимом адресации 10 бит. Выдача данных

Начало процесса чтения данных из подчиненного, использующего для адресации 10 бит, аналогично началу процесса записи, описанному в предыдущем пункте. Интерфейс отследит обращение к нему со стороны мастера и сформирует для процессора запрос прерывания. Код состояния будет равен 0x09. Дальнейшие действия должны быть следующими:

Б) Получение запроса на прерывание. Сброс бита IRQ.
Анализ того, что произошло обращение к процессору со стороны мастера по записи.
Если не известно, будет мастер записывать данные или считывать, то бит AA должен быть установлен в 1. Мастер может, как передать байт данных, так и сгенерировать событие “restart”. Устанавливаем AA в 1.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс освобождает линию SCL и позволяет мастеру выполнить следующие действия. При выполнении мастером операции чтения, он сформирует на линиях интерфейса событие “restart”. Интерфейс сбросит флаги slave, SLV_EXE, но останется установленным флаг SLV_XM, что будет свидетельствовать о выполненной недавно процедуре выбора подчиненного. Интерфейс примет первый байт адреса, определит совпадение, установит, что бит направления указывает на чтение. При выполнении мастером операции чтения, интерфейс проверит бит SLV_XM и убедится, что процедура выбора предшествовала чтению. После этого интерфейс ответит АСК=1 и перейдет в состояние «подчиненый_чтение». Интерфейс устанавливает бит SI и в регистр состояния записывает код 0x0B. Формируется запрос на прерывание к процессору. Интерфейс удерживает линию SCL в нуле.

Б) Получение запроса на прерывание. Сброс бита IRQ.

Анализ того, что произошло обращение к процессору со стороны мастера для чтения данных. Пользователю необходимо знать, какой байт данных хочет прочитать мастер. Нужное значение записывается в регистр данных DR для передачи.
Производим сброс бита SI регистра управления. Ожидание прерывания.

Интерфейс освобождает линию SCL и позволяет мастеру прочитать байт данных. После приема байта, мастер отвечает подчиненному битом подтверждения. Устанавливается бит SI, а в регистр состояния записывается код 0x12 или 0x13. Значение кода зависит от бита подтверждения. Если бит подтверждения равен 0, интерфейс сбрасывает внутренний флаг SLV_EXE и перестает контролировать выдачу данных на линию SDA. Это позволяет мастеру сформировать событие “stop”. Формируется запрос на прерывание к процессору. Интерфейс удерживает линию SCL в нуле.

B) Получение запроса на прерывание. Сброс бита IRQ.
Анализ кода состояния. Если бит АСК=1, процессор записывает в регистр данных DR следующий байт для передачи. Если бит АСК равен нулю, происходит завершение алгоритма. События “stop” можно не дожидаться.
Производим сброс бита SI регистра управления. Ожидание прерывания, если передача не завершена.

Интерфейс освобождает линию SCL и позволяет мастеру принять следующий байт данных (если чтение продолжается) или сформировать событие “stop”.

Нужно отметить, что мастер может прекратить работу в любой момент времени при записи данных в подчиненное устройство (мастер контролирует линии SCL и SDA, и может сформировать “stop” во время передачи бита данных). При чтении данных из подчиненного устройства линию SDA контролирует подчиненное устройство. Однако, если подчиненный выдает бит данных равный 1, мастер может сформировать “stop” в этом случае. Подчиненный все время должен быть готов, что при возможности мастер может преждевременно (аварийно) завершить обмен. Задача интерфейса, корректно обработать ситуацию, перейти в исходное состояние, не отвлекать процессор, если для него данные события не имеют значения. При желании контролировать аномальное завершение обмена пользователь должен устанавливать бит SLVE_EN. При использовании бита SLVE_EN после получения прерывания по событию “stop”, пользователь должен обработать прерывание и очистить флаг SLVE.

23.3.9 Мастер в многомастерной системе

Стандарт обмена I2C допускает работу нескольких ведущих (мастеров) устройств на одной шине. Особенность разработки алгоритма обмена в данной ситуации состоит в необходимости учитывать непредсказуемое поведение интерфейса. Например, интерфейс может работать как мастер и в тоже время имеет адрес на шине и может работать как подчиненный. В этом случае пользователь, желая выполнить обмен с некоторым подчиненным устройством в режиме мастер, может установить бит STA, ожидать возникновения события “start”, но в ответ получить прерывание с кодом, который указывает на то, что к интерфейсу производится обращение как к подчиненному устройству.

При работе в мультимастерной системе установка бита STA не гарантирует возможность захватить шину и стать мастером. Но даже после установки бита STA получение от интерфейса прерывания с кодом события “start” не гарантирует, что в это же время другой мастер одновременно не сформировал аналогичное событие. Поэтому, став мастером и начав обмен, интерфейс во время передачи первого байта адреса всегда сравнивает то, что он выдает на линию SDA и то, что реально присутствует на линии. Сравнение происходит в момент нарастающего фронта (из 0 в 1) линии SCL. Если выдаваемое значение и значение на линии не совпадают, интерфейс принимает решение об аварийном завершении и отключается от линий обмена. При этом бит ABRT устанавливается в 1, и формируется запрос прерывания к процессору. Пользователь должен проанализировать данный бит и принять решение о повторной попытке обмена. В данном случае имеет место попытка нескольких мастеров одновременно обратиться к устройствам с различными адресами.

В случае, когда несколько мастеров одновременно обращаются к одному устройству, все биты адреса и бит направления совпадут. Тогда определить факт одновременного наличия нескольких мастеров на шине можно только, сравнивая выдаваемые биты данных. Сравнение данных включается специальным битом ABN_EN. Если он установлен, интерфейс будет анализировать выдаваемые биты данных, и, в случае аномалий, произведет отключение мастера и установится бит ABRT. В случае чтения, данные выдает подчиненное устройство, а мастера могут выдавать бит подтверждения. Нарушение бита подтверждения не контролируется интерфейсом, однако, получив прерывание, пользователь сам может определить, какой бит он выдавал и какой получил с линии. *В данной ситуации обмена проблема может возникнуть, если два мастера читают разное количество данных из подчиненного устройства. Мастер, который первым прочитал нужное ему количество данных, ответит неактивным битом подтверждения (сбросит АА перед чтением последнего байта). Однако другой мастер ответит активным битом подтверждения. В этом случае первый мастер получит код прерывания, в котором будет противоположное значение бита подтверждения. Данный факт рассматривается как наличие второго мастера, и первый мастер должен освободить интерфейс. В этом случае, при сбросе бита SI необходимо сбросить и бит включения интерфейса. Это приведет интерфейс в исходное состояние. Если интерфейс не выключать, то он продолжит работу как мастер и будет принимать данные до тех пор, пока последний мастер на шине не сформирует АСК равный нулю. В принципе, и такой алгоритм можно рассматривать как рабочий.*

При выдаче сигнала синхронизации SCL мастер всегда контролирует факт удержания подчиненным линии SCL в нуле. Поэтому, если два мастера начали одновременную работу, и их частоты не совпадают, итоговое значение линии SCL будет равно «проводному И» двух выдаваемых сигналов. Если факт задержки установки высокого уровня SCL это нормальное событие для мастера, то факт более ранней выдачи нуля на линии SCL может рассматриваться как присутствие другого мастера. В этом случае мастер, который раньше выдал на SCL ноль, может выключить других, более «медленных» мастеров. Произойдет это по следующей причине. Выдача новых данных на линию SDA привязана к реальному уровню на линии SCL, а сравнение данных внутри конкретного мастера привязано к его внутреннему клоку SCLm. Более быстрый мастер переключит линию SCL из 1 в ноль, все мастера выдадут на линию новый бит данных. Но более медленный мастер будет по-прежнему сравнивать ранее выдаваемый бит с новым значением линии SDA. Поэтому, если при выдаче есть смена значения бита данных, то медленный мастер отключится, даже если оба мастера выдают одно и то же значение. Поскольку значение 0 и значение 0xFF адреса маловероятно, то

разноскоростные мастера с большой долей вероятности обнаружат аномалию еще при выдаче адреса.

Для разрешения обнаружения конфликтов на линиях в момент генерации события “restart” используется бит ABN_S_EN. Конфликт может произойти, если два мастера одновременно обращаются к одному и тому же устройству, но один мастер для записи, а другой для чтения данных. В данном случае мастер, который пытается сгенерировать событие “restart”, освобождает линии данных и синхронизации, что позволяет другому мастеру свободно осуществлять выдачу данных.

Для разрешения обнаружения конфликтов на линиях в момент генерации события “stop” используется бит ABN_P_EN. Конфликт может произойти, например, если два мастера одновременно обращаются к одному и тому же устройству по записи, но один мастер желает завершить обмен (не смотря на бит подтверждения подчиненного), а другой пытается продолжить обмен. В данном случае мастер, который пытается сгенерировать событие “stop”, удерживает линию данных в нуле, но освобождает линию синхронизации. Здесь возможны два варианта:

1 Первому мастеру удалось сформировать событие «стоп» (если второй мастер выдавал бит данных 1), а второй мастер получит преждевременное прерывание по окончанию обмена.
2 Если второй мастер выдает бит данных 0, первый мастер отключается от шины, т.к. генерирует внутреннее событие “stop” и не проверяет, что такое же событие произошло на линиях, а второй мастер благополучно продолжает обмен.

Остаются проблемными ситуации, описанные в стандарте:

- Один мастер “restart”, а другой выдает бит данных. В этом случае очень важна скорость обмена устройств и то, какой бит данных выдает мастер. Наиболее вероятен сценарий, при котором мастер формирующий рестарт обнаружит аномалию на шине и отключится. Но не исключен вариант, что ошибку в выдаваемом бите данных обнаружит второй мастер.
- Один мастер “stop”, а другой выдает бит данных. В данном случае возможны как преждевременная генерация события стоп для второго мастера, так и благополучное завершение операций для обоих мастеров.
- Один мастер “restart”, а другой “stop”. В данной ситуации сформируется ситуация стоп, т.к. линия данных будет удерживаться в нуле, что позволит первому мастеру обнаружить конфликт.

Если при работе в многомастерной системе возможны аномалии, которые приводят к переводу интерфейса в непредсказуемое состояние, необходимо использовать системный таймер в качестве генератора максимального интервала ожидания события от интерфейса. Если запрос от таймера придет раньше, чем ответ от интерфейса, ситуацию можно рассматривать как аномальное явление. После этого необходимо выключить интерфейс, произвести анализ состояния линий интерфейса, при необходимости сбросить подчиненные устройства и затем повторить попытку обмена. Возможный алгоритм поведения следующий:

А) Обнаружение потери управления шиной (бит ABRT установлен).
Б) Установка бита FREE_EN и программирование таймера на заданный интервал времени.
В) Если первым пришел запрос прерывания от интерфейса и по флагам интерфейса видно, что обмен не был завершен корректно (отсутствовал «стоп»), генерация на линиях события «старт», а затем «стоп».
Г) Если первым пришел запрос на прерывание от таймера, выполняется программный анализ линий интерфейса. Если линия SCL удерживается в нуле, выход только в аппаратном сбросе устройств. Если линия SDA

удерживается в нуле, программная генерация синхросигнала на линии SCL, до момента, когда SDA равно единице. После этого генерация события “stop”.

23.3.10 Одновременная работа в режиме мастера и подчиненного

Возможна ситуация, когда в многомастерной системе необходимо передавать сообщения другим мастерам, а также принимать от них сообщения. В этом случае интерфейс должен самостоятельно определить, в каком режиме он должен работать в текущий момент времени. Рассмотрим пример системы, в которой имеется несколько процессоров, все они подключены посредством своих аппаратных интерфейсов к общей I2C линии и периодически обмениваются сообщениями. При включении интерфейса не задается его режим работы. Если пользователь установил бит STA и интерфейс сформировал событие “start”, то интерфейс будет работать в режиме мастера. Если же интерфейс получил с линий «чужое» событие “start”, то он – подчиненный. При этом у него должен быть запрограммирован адрес подчиненного устройства в соответствующих регистрах.

Если возникнет ситуация, когда одновременно несколько интерфейсов становятся на шине мастерами, интерфейс А (адрес подчиненного 0x20) может обращаться к интерфейсу Б (адрес подчиненного 0x22), а интерфейс Б к интерфейсу А. При передаче адреса интерфейс Б обнаружит, что передан не его адрес и отключит режим мастера (будет установлено прерывание и флаг ABRT). Однако интерфейс Б продолжит принимать данные с шины и определит, что выполняется обращение к нему как к подчиненному устройству. В данном случае пользователь интерфейса Б, начав обмен по шине как мастер должен быть готов к тому, что он потеряет управление шиной и к нему может быть выполнено обращение как к подчиненному устройству. Если такая ситуация произошла, пользователь интерфейса Б должен обработать запрос подчиненного устройства и затем повторить попытку обмена.

Отметим, что данный вариант обмена допустим только при 7-битовой адресации подчиненных устройств. При 10-битовом режиме адресации, в случае одновременного старта нескольких мастеров, они могут пропустить обращение к ним как к подчиненному устройству. При этом интерфейс мастера, который первым обнаружил аномалию на шине и отключился, может не смочь в данной транзакции ответить как подчиненное устройство. Мастер, который обращается к интерфейсу с 10-битовой адресаций может не получить активный бит подтверждения. Этот факт должен рассматриваться лишь как аномальное событие на шине. Реакцией на такое событие должна быть повторная попытка обмена. Однако начало повторной попытки для обоих мастеров лучше всего сделать через случайно сгенерированный интервал времени. Это позволит избежать одновременного захвата шины несколькими интерфейсами.

Отметим, что подобной системе лучше всего программировать скорости интерфейсов, отличающиеся друг от друга на некоторую величину. В этом случае вероятность одновременного захвата шины будет очень низкой.

23.4 Некоторые особенности работы интерфейса

В данном подразделе будут рассмотрены некоторые моменты в реализации аппаратной части интерфейса и сформулировать рекомендации для разработки более надежных алгоритмов обмена.

23.4.1 Запрос прерывания

Прерывание от I2C интерфейса анализируется по уровню. Поэтому при обработке прерывания необходимо обязательно сбрасывать бит IRQ регистра управления. Нужно гарантировать надежный интервал времени между сбросом IRQ и выходом из обработчика прерывания, чтобы исключить возможность повторной генерации прерывания.

23.4.2 Включение интерфейса

Если процессор является единственным мастером на шине I2C, включение интерфейса не представляет собой никаких проблем. Однако если процессор подключен к шине, на которой находится несколько мастеров, включение интерфейса в работу может произойти во время обмена по шине. Аппаратура интерфейса, по возможности, обеспечивает исключение возникновения различных непредвиденных ситуаций. Хотя может понадобиться и анализ состояния линий интерфейса перед включением.

23.4.3 Бит STA

Данный бит устанавливается в 1 при необходимости сформировать на линиях интерфейса событие “start” или “restart”. После получения прерывания, необходимо не забывать сбрасывать данный бит в ноль. Аппаратно данный бит не сбрасывается. После генерации события “start” или “restart” бит STA удерживает линию SDA в нуле. Чтобы обеспечить необходимое время предустановки данных, бит STA должен быть сброшен до момента записи данных для передачи. Обязательно бит STA должен быть сброшен до сброса флага SI. Если нужно сформировать событие “restart” (при установленном бите SI), необходимо убедиться, что бит STA сброшен. Затем установить бит STA и только после этого сбросить бит SI. Это гарантирует время предустановки линии SDA до фронта SCL. Возможно, для соблюдения параметра t_LOW линии SCL, необходимо будет использовать бит TLOW_EN.

Перед установкой бита STA проверка готовности линий интерфейса не обязательна. Аппаратура интерфейса самостоятельно анализирует готовность линий для генерации события.

23.4.4 Бит SLE_EN

Интерфейс, обнаружив “stop”, выдаст запрос на прерывание путем установки бит IRQ и SLVE. После этого интерфейс переходит в состояние ожидания и может детектировать факт обращения к подчиненному до того, как процессор обработает предыдущее событие “stop”. Если по каким-то причинам процессор не успел обработать событие “stop” от подчиненного до момента приема стартового адресного байта следующего обмена, то в коде состояние будет отражен факт приема стартового байта, но в тоже время будет присутствовать и установленный бит SLVE.

При предсказуемом поведении мастера и заранее оговоренном формате обмена, нет необходимости в использовании флага SLVE. При чтении из подчиненного АСК формирует мастер, и он обязан будет дать АСК = 0, чтобы информировать о завершении обмена.

Иначе подчиненный не отпустит линию SDA. В этом случае нет необходимости контролировать "стоп". При записи в типичных случаях всегда известна длина записываемых данных, и подчиненный может сам управлять окончанием обмена, выдавая АСК=0. Поэтому опять "стоп" контролировать не обязательно. В результате можно сделать вывод, что данный флаг и бит его разрешения можно использовать только для контроля каких-либо аномалий на шине обмена (если они возможны). Например, после активизации подчиненный выполняет прием или выдачу байта данных. Перед обменом байта устанавливаем SLV_EN. Байт данных передается-принимается корректно, и подчиненный обрабатывает событие. Если подчиненный видит, что этот байт был последний, он больше не контролирует шину и снимает SLV_EN (сбрасывая одновременно и другие флаги). Если вместо события об окончании приема-передачи данных будет получен флаг SLVE, рассматриваем это как аномальное завершение и отмечаем факт несостоявшегося обмена. Таким образом, бит SLVE (SLV_EN) выступает как бы предохранителем возможного непредсказуемого поведения мастера. Только в случае приема подчиненным неизвестного заранее объема блока данных, может возникнуть потребность в контроле события "стоп", чтобы узнать, что блок данных передан.

23.4.5 Мастер. Делитель клока

Интерфейс имеет 12-разрядный счетчик DIVC, который используется для задания частоты синхронизации SCL при работе в режиме мастера. Счетчик задает четверть периода частоты синхронизации и ведет счет, начиная от величины DIV (регистр VR) до 1. Затем счет повторяется. Каждый раз, когда счетчик перезагружается, специальный 2-разрядный счетчик CNTC увеличивает свое значение на 1. Старший бит счетчика CNTC и есть внутренний сигнал синхронизации SCLm, который выдается на линию SCL. Таким образом, в одном периоде SCL можно выделить четыре фазы t0, t1, t2, t3. Счетчик может останавливать свой счет при следующих условиях:

- Работая в режиме мастера и находясь в интервале t1, при значении DIVC равном 2 интерфейс проверяет, установлен ли бит SI или истек ли период предустановки данных. Если бит SI установлен или период предустановки не закончился, интерфейс останавливает счет DIVC.
- Работая в режиме мастера и находясь в интервале t2 (SCLm стал равен 1), интерфейс через 4 такта после начала интервала проверяет значение линии SCL. Если линия в нуле, счетчик DIVC останавливается. Это означает, что подчиненное устройство удерживает мастера. Как только линия SCL перейдет в высокий уровень, счетчик возобновит счет.
- Работая в режиме мастера и находясь в интервале t2 (SCLm равен 1), интерфейс сканирует линию SCL, и как только на ней установится высокий уровень, интерфейс установит внутренний флаг POS_F. Установка данного флага означает, что линия SCL была переведена в высокий уровень. На счетчик DIVC не влияет ситуация, когда в течение интервалов t2 или t3 на линии будет короткий импульс SCL=0 (длительность не более периода SCLK). Но, если на линии SCL преждевременно (до окончания интервала t3) установится нулевой уровень, то счетчик DIVC перезагрузится, а счетчик CNTC сбросится в ноль. Счет продолжится.

23.4.6 Подключение на плате

Для устойчивой работы интерфейса требуется на выводе SCL поставить RC-фильтр. Номинал резистора 100 Ом, конденсатора – 30 ÷ 56 пФ. Такие номиналы позволяют работать до скорости обмена в 400 кГц.

23.5 Примеры программирования

В данном подразделе приведены несколько примеров программирования различных действий интерфейса при реализации алгоритмов обмена.

23.5.1 Мастер. Обработка события “start”

Предположим, что интерфейс включен: запрограммирована скорость обмена, установлен бит включения. Перед установкой бита STA пользователю следует записать в регистр векторов прерываний адрес обработки события “start”, разрешить прерывание от I2C, в некоторую структуру info_I2C записывается адрес устройства, к которому будет обращение, после этого установить бит STA. Далее можно выполнять другие задачи или ожидать прерывание. При наступлении прерывания выполниться процедура:

```
.align 4;
do_restart:       q[j27+4] = k3:0;
                         k2 = k31 + base_I2C;;

                   q[j27+8] = j3:0;
                              k3 = 0x820;;
                   [k2+I2C_CRC] = k3;; // clear int & STA

                   j1 = [k2+I2C_SR];;     // get status
                   k0 = [k31 + info_I2C+0];; // device address
                   [k2+I2C_DR] = k0;;

                   comp(j1,2); k0 = k31+8;;       // check start code
                   if njeq, jump abnormal_start (NP); k1 = restart_service;;

                   [j31+base_INTV+26] = k1;;
                   [k2+I2C_CRC] = k0;   // clear SI

                    j3:0  = q[j27+8];;
                    rti (ABS) (NP); k3:0  = q[j27+4];;
```

В данной процедуре следует сохранить на стеке несколько регистров, загрузить в свободные регистры базовый адрес интерфейса и нужную информацию. Необходимо сбросить бит запроса прерывания и бит STA. Из структуры извлечь адрес устройства и записать его в регистр данных для передачи. Далее необходимо прочитать регистр состояния и убедиться, что произошло ожидаемое событие (что не является обязательным, т.к. других событий интерфейс сгенерировать не может). В регистр векторов прерываний следует записать адрес следующего обработчика, который будет обрабатывать факт передачи адреса и ответ устройства (адрес *service_Address*). После это необходимо сбросить бит SI и разрешаем интерфейсу продолжить работу. Далее следует восстановить регистры и выйти из обработчика.

Данный обработчик тратит 42 такта процессора от момента установки запроса прерывания в регистре управления, до момента сброса флага SI.

Если ожидается прерывание от I2C, а не выполняется иная работа, то вся необходимая информации (базовые адреса, нужные константы) может находиться в регистрах. В этом случае не нужно ничего сохранять на стеке, а затем восстанавливать, и время обработки может быть меньше. Например, следующая процедура:

```
.align 4;
do_xrestart:      [k2+I2C_CRC] = j2;; // clear int & STA
                  j1 = [k2+I2C_SR];;      // get status
                   [k2+I2C_DR] = k0;  j2 = j31+8;; // write address

                   [j31+base_INTV+26] = k1;;    // setup new vector
                   rti (ABS) (NP); [k2+I2C_CRC] = j2;; // clear SI
```

От момента установки запроса прерывания, до момента сброса флага SI, задержка 28 тактов. Особое внимание обратим на команду `j1 = [k2+I2C_SR]`. В данном случае нет необходимости в чтении регистра состояния, но можно использовать это чтение как задержку. Такая задержка нужна, чтобы гарантированно снялся запрос прерывания до момента выхода из прерывания. Если эту задержку убрать, будет сформирован ложный повторный запрос, т.к. процессор успеет выйти из прерывания до момента, когда еще не снят запрос прерывания (из-за более медленной шины периферийных устройств).

Запись нового вектора прерываний должна быть раньше, чем сброс флага SI. Иначе нет гарантии, что новое прерывание сработает по новому вектору.

24 ARINC контроллер

Контроллер интерфейса ARINC содержит в своем составе восемь приемников и четыре передатчика по ГОСТ 18977-79 (далее ARINC).

Каждый приемник поддерживает функцию распознавания меток (или адресов). Для каждого приемника может быть запрограммировано до 256 меток. Помимо этого, фильтрация входных данных может осуществляться не только на базе меток, но и на базе двух бит Источник/Приемник.

Каждый передатчик поддерживает однонаправленную передачу 32-х разрядных слов по двухпроводной витой паре, используя формат кодирования RZ. Доступна возможность запрограммировать 32-й бит как данные или как бит паритета. В случае формирования бита паритета, программируется его четность или нечетность.

Каждый приемник и передатчик использует собственный буфер FIFO для хранения данных. Размеры буфера FIFO варьируются от 32x32 до 256x32. Статус наполненности FIFO определяется на основе соответствующих бит статуса для каждого FIFO. Контроллер поддерживает различные скорости приема и передачи данных. Работа контроллера осуществляется на базовой частоте 1 МГц, что позволяет обнаруживать ошибки в скорости приема/передачи данных, а также в паузах между сообщениями.

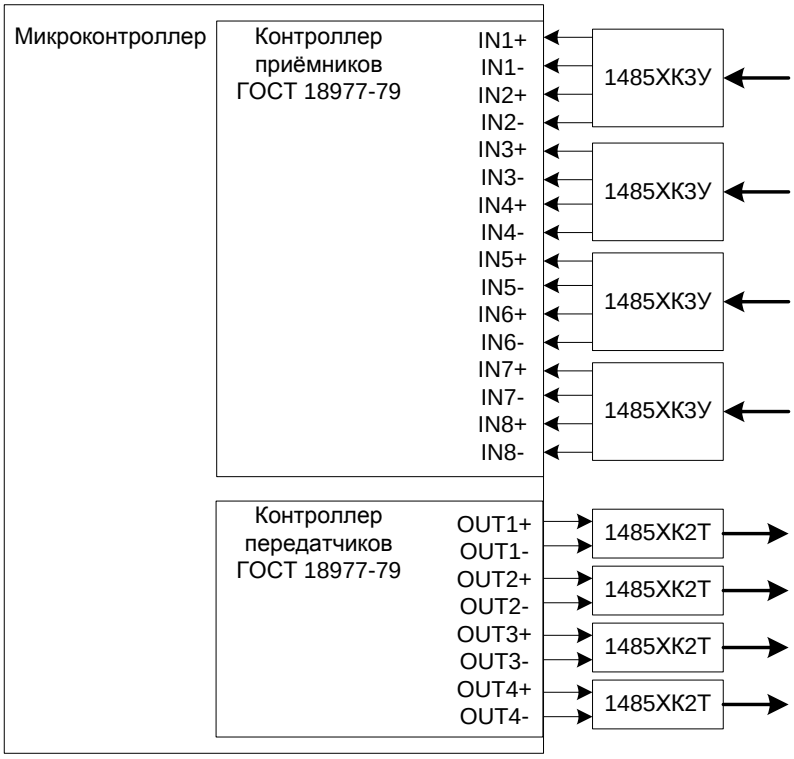


Рисунок 101

Особенности:

- симплексный режим приема/передачи со скоростями 12,5 кГц или 100 кГц;
- фильтрация входных данных на базе 256 меток и двух бит Источник/Приемник для каждого приемника;
- возможность передачи 32 бита, как данных, так и паритета;
- выбор четности/нечетности бита паритета;

- размеры буферов FIFO передатчиков: одно 256x32, три 64x32;
- размеры буферов FIFO приемников: два 256x32, четыре 64x32, два 32x32;
- возможность формирования прерываний при разных статусах наполненности буферов FIFO и при возникновении ошибок скорости передачи слова и паузы между словами;
- маскирование прерываний.

Назначение внешних выводов контроллера приведено в Таблице 202.

Таблица 202 – Назначение внешних выводов контроллера

Обозначение вывода	Назначение вывода контроллера	Тип	Функциональное назначение
PB[4]	AR_IN1P	I/O	Интерфейс ARINC. Вход приемника 1 (+)
PB[5]	AR_IN1N	I/O	Интерфейс ARINC. Вход приемника 1 (-)
PB[6]	AR_IN2P	I/O	Интерфейс ARINC. Вход приемника 2 (+)
PB[7]	AR_IN2N	I/O	Интерфейс ARINC. Вход приемника 2 (-)
PB[8]	AR_IN3P	I/O	Интерфейс ARINC. Вход приемника 3 (+)
PB[9]	AR_IN3N	I/O	Интерфейс ARINC. Вход приемника 3 (-)
PB[10]	AR_IN4P	I/O	Интерфейс ARINC. Вход приемника 4 (+)
PB[11]	AR_IN4N	I/O	Интерфейс ARINC. Вход приемника 4 (-)
PB[12]	AR_IN5P	I/O	Интерфейс ARINC. Вход приемника 5 (+)
PB[13]	AR_IN5N	I/O	Интерфейс ARINC. Вход приемника 5 (-)
PB[14]	AR_IN6P	I/O	Интерфейс ARINC. Вход приемника 6 (+)
PB[15]	AR_IN6N	I/O	Интерфейс ARINC. Вход приемника 6 (-)
PB[16]	AR_IN7P	I/O	Интерфейс ARINC. Вход приемника 7 (+)
PB[17]	AR_IN7N	I/O	Интерфейс ARINC. Вход приемника 7 (-)
PB[18]	AR_IN8P	I/O	Интерфейс ARINC. Вход приемника 8 (+)
PB[19]	AR_IN8N	I/O	Интерфейс ARINC. Вход приемника 8 (-)
PB[20]	AR_OU1P	I/O	Интерфейс ARINC. Выход передатчика 1 (+)
PB[21]	AR_OU1P	I/O	Интерфейс ARINC. Выход передатчика 1 (-)
PB[26]	AR_OU2P	I/O	Интерфейс ARINC. Выход передатчика 2 (+)
PB[27]	AR_OU2N	I/O	Интерфейс ARINC. Выход передатчика 2 (-)
PB[28]	AR_OU3P	I/O	Интерфейс ARINC. Выход передатчика 3 (+)
PB[29]	AR_OU3N	I/O	Интерфейс ARINC. Выход передатчика 3 (-)
PB[30]	AR_OU4P	I/O	Интерфейс ARINC. Выход передатчика 4 (+)
PB[31]	AR_OU4N	I/O	Интерфейс ARINC. Выход передатчика 4 (-)

24.1 Формат слова

Слова в интерфейсе ARINC всегда 32-разрядные, и включают в себя пять полей: паритет, SSM, данные, источник/приемник, метка. Биты передаются младшими разрядами вперед, за исключением метки, которая передается старшими разрядами вперед. В результате можно описать порядок следования бит по шине ARINC следующим образом:

8, 7, 6, 5, 4, 3, 2, 1, 9, 10, 11, 12, 13…32.

32	31	30	29 11	10	9	8 1
P	SSM		DATA MSB LSB	SDI		LABEL

Старший разряд всегда бит паритета. Стандартом установлено, что бит паритета должен дополнять слово до нечетного. Таким образом, количество единиц в 32-разрядном слове должно быть нечетным. Например, если биты 1-31 содержат

четное количество единиц, бит паритета должен быть установлен в единицу, с другой стороны, если биты 1-31 содержат нечетное количество единиц, то бит паритета должен быть сброшен в ноль.

Биты 31 и 30 содержат знак или статус. В контроллере эти биты рассматриваются как обычные данные и помещаются в FIFO вместе с полем данных без изменений и дополнительной обработки.

Как пример биты 31 и 30 могут кодировать следующие характеристики, представленные ниже (Таблица 203).

Таблица 203 – Биты 31 и 30

Бит		Значение
31	30	
0	0	плюс, север, восток, справа, к, выше
0	1	невычислительные данные
1	0	функциональный тест
1	1	минус, юг, запад, слева, от, ниже

Биты 10 и 9 позволяют распознать Источник/Приемник данных. Это применяется при нескольких приемниках на шине ARINC, чтобы определить, для кого из них предназначаются данные. В системе со сложной структурой эти биты могут также использоваться, чтобы определить источник передачи. В остальных случаях эти разряды используются как данные. Следует отметить, что в интерфейсе ARINC на одной витой паре может быть один передатчик и до 20 приемников. Если включена функция проверки этих бит, при их несовпадении с битами, заданными программно в контроллере, сообщение не будет помещено в FIFO.

Биты с 1 по 8 позволяют идентифицировать тип данных оставшейся части слова, следовательно, методы преобразования, применяемые к данным. Помимо этого в контроллере метки используются для фильтрации входных данных, то есть если метка в принятом сообщении не соответствует ни одной из меток определенной в памяти меток приемного канала, то данные не помещаются в FIFO. Это может служить аналогом того, что приемник не может интерпретировать метод обработки этих данных, следовательно, эти данные предназначены для другого приемника.

В случае если приемник принимает данные с неправильным битом паритета, они также не будут помещены в FIFO.

24.2 Структурная схема канала приема

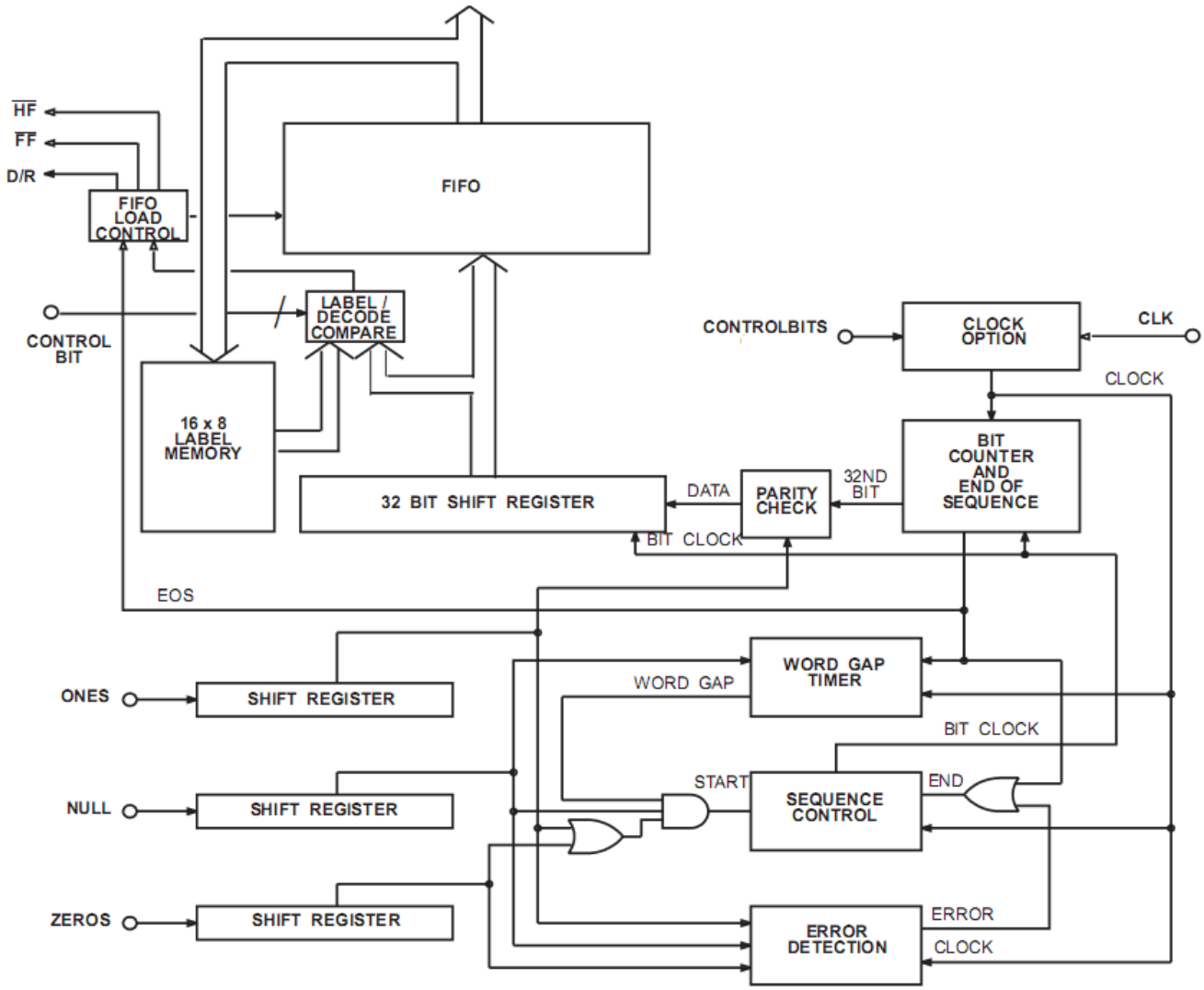


Рисунок 102 – Структурная схема канала приема

Представленная на рисунке схема работает на частоте CLK = 1 МГц, в этом случае ошибка обнаружения бита в линии не будет составлять более 0,1 %.

Сдвиговые регистры длиной 10 бит, предназначенные для обнаружения в линии трех последовательностей единиц (Ones), нулей (Zeros) и отсутствие сигнала (Null), позволяют считать данные действительными. В дополнении к этому для бит данных, One или Zero в верхних битах сдвигового регистра должны сопровождаться Null в нижних битах в переделах битового интервала. В пределах паузы между сообщениями, три последовательных бита Null должны быть сэмплированы в верхней и нижней части сдвигового регистра Null. В этом случае гарантируется минимальная ширина импульса данных.

Каждый бит данных должен быть обнаружен в пределах от 8 до 12 сэмплов. В этом случае скорость передачи считается верной.

Таймер паузы между сообщениями сэмплирует сдвиговый регистр Null каждые 10 входных тактов (или 80 тактов для скорости 12,5 кГц) после последнего полученного бита данных. Если Null обнаружен, таймер инкрементируется. Значение таймера равное трем разрешает следующий прием.

Схема паритета считает количество принятых единиц, включая бит паритета. Если результат нечетный, на выходе схемы формируется сигнал равный нулю.

После того как приняты все 32 бита логика приемника формирует сигнал конец последовательности (EOS). В зависимости от состояния бит LB_EN, SD_EN, SDI1, SDI2 регистра управления принимается решение о загрузке принятых данных

в FIFO. Если в принятом слове биты 9 и 10 не соответствует правилам или не совпала метка, слово не загружается в FIFO. Случаи, в которых происходит загрузка FIFO принятыми данными, перечислены ниже (Таблица 204).

Таблица 204 – Загрузка FIFO принятыми данными

LB_EN	Результат сравнения слова ARINC с меткой	SD_EN	Результат сравнения бит 9,10 слова ARINC с SDI1, SDI2	FIFO
0	X	0	X	Загружается
1	не совпала	0	X	Игнорируются
1	совпала	0	X	Загружается
0	X	1	не совпали	Игнорируются
0	X	1	совпали	Загружается
1	совпала	1	не совпали	Игнорируются
1	не совпала	1	совпали	Игнорируются
1	не совпала	1	не совпали	Игнорируются
1	совпала	1	совпали	Загружается

Если хотя бы одно слово загружено в FIFO, устанавливается в единицу сигнал DR, что отражается в регистре статуса контроллера. Флаг остается в неизменном состоянии, пока последнее слово не будет прочитано из FIFO и оно не будет пустым. Помимо этого применяются еще два сигнала характеризующие состояние FIFO, а именно HF означает, что FIFO наполовину полно и FF означает, что FIFO полно. Установка этих сигналов также отражается в регистре статуса. Каждый из этих флагов может быть источником прерывания, в случае если оно разрешено соответствующим битом маскирования регистра управления.

24.3 Структурная схема канала передачи

Если флаг TX_R в состоянии логической единицы, это значит что FIFO пусто и в него могут быть загружены 31- или 32-битные данные. Количество слов данных определяется размером FIFO для выбранного канала передачи. Если флаг TX_R в состоянии логического нуля, тогда только в доступные в FIFO ячейки можно загрузить данные. Если FIFO заполнено полностью, флаг FFT установлен в единицу, то FIFO игнорирует дальнейшие попытки записи в него. FIFO наполовину пусто, если установлен флаг HFT, в этом случае можно загрузить данными оставшуюся половину буфера FIFO.

В нормальном режиме работы 32 бит передаваемых данных является битом паритета. Четность или нечетность выбирается битом ODD регистра управления. Если бит разрешения паритета (EN_PAR) сброшен в ноль, тогда 32 бит передается, как бит данных из FIFO.

Если бит CH_EN установлен в единицу и FIFO передачи не пусто, начинается передача слов данных из FIFO до тех пор, пока FIFO не будет пусто или не будет сброшен бит CH_EN.

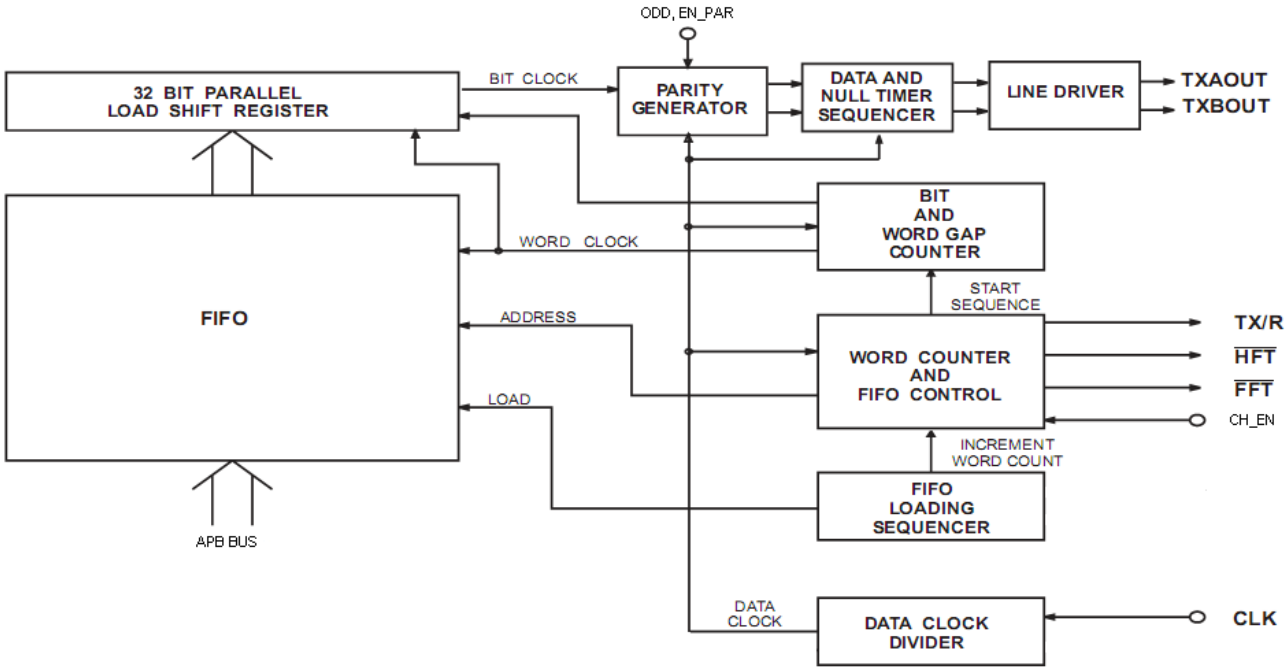


Рисунок 103 – Структурная схема канала передачи

24.4 Описание регистров приемника

Таблица 205 – Регистры приемника

Базовый адрес	Номер	Название	Состояние после сброса	Описание
0x8000_2000		ARINC429R		
+0				Канал 0
	0	RX_CR		Регистр управления приемника
	1	-		-
	2	DATA_R		FIFO принимаемых данных
	3	STATUS		Регистр состояния приемника
+4				Канал 1
+8				Канал 2
+12				Канал 3
+16				Канал 4
+20				Канал 5
+24				Канал 6
+28				Канал 7
0x8000_2400		Память приемников		Прямой доступ в память приемников
	0x0000-0x00FF	FIFO_D0		Буфер приемника канала 0
	0x0100-0x01FF	FIFO_D1		Буфер приемника канала 1
	0x0200-0x023F	FIFO_D2		Буфер приемника канала 2
	0x0240-0x027F	FIFO_D3		Буфер приемника канала 3

	0x0280-0x02BF	FIFO_D4		Буфер приемника канала 4
	0x02C0-0x02FF	FIFO_D5		Буфер приемника канала 5
	0x0300-0x031F	FIFO_D6		Буфер приемника канала 6
	0x0320-0x033F	FIFO_D7		Буфер приемника канала 7
	0x0340-0x03BF	-		-
	0x03C0-0x03FF	LABELS		Буфер меток. Для каждого канала от 0 до 7 используется 8 32-разрядных слов образующих одно 256 битовое кодовое слово метки. Каждый бит соответствует одному из 256 возможных значений метки. Номер слова – три младших бита адреса. Номер канала – биты 5..3 адреса.

Доступ к буферу данных возможен как прямым, так и косвенным путем (через указатель чтения). Доступ к меткам только прямым путем.

24.5 Регистр управления приемника RX_CR

Таблица 206 – Биты регистра управления приемника RX_CR

Бит	Имя	Значение	Описание
0	CH_EN		Разрешение работы канала 1 – прием по каналу разрешен 0 – канал приема находится в состоянии сброса
1	CLK		Скорость приема данных 1 – частота приема данных = опорная частота/80 (12,5 кГц) 0 – частота приема данных = опорная частота/10 (100 кГц)
2	LB_EN		Разрешение обнаружения меток 1 – разрешено обнаружение меток в первых 8 принятых битах 0 – обнаружение отключено, все принятые данные помещаются в FIFO
3	SD_EN		Разрешение декодирования бит данных 9 и 10 1 – разрешено сравнение бит данных 9 и 10 со значением бит SDI1 и SDI2 соответствующего канала 0 – декодирование отключено, все принятые данные помещаются в FIFO
4	DA		Бит прямого доступа 1 – память приема канала работает не в режиме FIFO, доступ к ней осуществлется в диапазоне адресов в зависимости от номера канала 0 – обычный режим работы FIFO При приеме данных из канала занесение их в память происходит в соответствии с адресом в первых восьми битах сообщения.
5	SDI1		Бит сравнения SDI1 Значение бита сравнивается с битом 9 принимаемых данных, если установлен бит SD_EN соответствующего канала

6	SDI2		Бит сравнения SDI2 Значение бита сравнивается с битом 10 принимаемых данных, если установлен бит SD_EN соответствующего канала
7	ENSYNC		Разрешение работы входов приемника в режиме данных и синхросигнала 1- разрешено 0 – запрещено При установленном бите ENSYNC для соответствующего канала вход IN_A работает как данные (D), вход IN_B как синхросигнал (SYN).
8	ENPAR		Разрешение 32 бита паритета для канала 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных Запрещено сбрасывать этот бит в ноль в штатном режиме работы контроллера
9	ODD		Выбор четности или нечетности бита паритета 1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, бит паритета устанавливается в 1, в противном случае в 0)
10	-		
11	-		
12	INTEDR		Разрешение прерывания наличие данных в FIFO 1 – разрешено прерывание, если FIFO приема данных не пусто 0 – прерывание запрещено
13	INTEER		Разрешение прерывания ошибка приема 1 – разрешено прерывания при возникновении ошибки в скорости приема или во времени паузы 4T между сообщениями (для сброса ошибки необходимо сбросить канал битом CH_EN) 0 – прерывание запрещено
14	INTEFF		Разрешение прерывания FIFO полно 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
15	INTEHF		Разрешение прерывания FIFO наполовину полно 1 – разрешено прерывание, если FIFO наполовину или более полно 0 – прерывание запрещено
23..16	DIV[7:0]		Делитель частоты ядра до 1 МГц Содержит значение, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц. Значение частоты не может быть более 255 МГц
31:24			

24.5.1 Регистр состояния приемника STATUS

Таблица 207 – Биты регистра состояния приемника STATUS

Бит	Имя	Значение	Описание
0	DR		Бит наличия данных в FIFO 0 – FIFO пусто 1 – FIFO содержит данные
1	ERR		Бит ошибки 0 – нет ошибок 1 – возникла ошибка приема
2	FF		Бит полноты FIFO 0 – FIFO не полно 1 – FIFO полно
3	HF		Бит наполненности FIFO 0 – FIFO не наполнено до половины 1 – FIFO заполнено на половину и более
4	IRQ_DR		Запрос прерывания при наличии данных в FIFO 0 – нет запроса 1 – есть запрос
5	IRQ_ERR		Запрос прерывания при наличии ошибки 0 – нет запроса 1 – есть запрос
6	IRQ_FF		Запрос прерывания при полном заполнении FIFO 0 – нет запроса 1 – есть запрос
7	IRQ_HF		Запрос прерывания при заполнении FIFO наполовину или более 0 – нет запроса 1 – есть запрос
8	IRQ_RX_CH0		Запрос прерывания от канала 0 0 – нет запроса 1 – есть запрос
9	IRQ_RX_CH1		Запрос прерывания от канала 1 0 – нет запроса 1 – есть запрос
10	IRQ_RX_CH2		Запрос прерывания от канала 2 0 – нет запроса 1 – есть запрос
11	IRQ_RX_CH3		Запрос прерывания от канала 3 0 – нет запроса 1 – есть запрос
12	IRQ_RX_CH4		Запрос прерывания от канала 4 0 – нет запроса 1 – есть запрос
13	IRQ_RX_CH5		Запрос прерывания от канала 5 0 – нет запроса 1 – есть запрос
14	IRQ_RX_CH6		Запрос прерывания от канала 6 0 – нет запроса 1 – есть запрос
15	IRQ_RX_CH7		Запрос прерывания от канала 7 0 – нет запроса 1 – есть запрос
23..16	CHAN_WP		Указатель на запись данных
31..24	DATA_RP		Указатель на чтение данных

Все приемники имеют один вход запроса прерывания в контроллере прерываний. Дополнительная информация о конкретном источнике прерываний может быть получена из регистра состояний.

24.5.2 LABEL

Буфер меток, с которыми сравниваются первые восемь принимаемых бит, если установлен LB_EN бит соответствующего канала. Размер буфера для каждого канала 256x1. Это означает что каждой из 256 кодовых комбинаций метки поставлен в соответствие один бит буфера. Процесс сравнения выполняется путем чтения соответствующего бита сразу после приема кода метки. К моменту приема всего сообщения бит из буфера должен быть получен и если его значение равно 1, это соответствует приему сообщения с заданной меткой.

24.5.3 DATA_R

FIFO принимаемых данных

В FIFO помещаются 32-разрядные данные, принимаемые из соответствующего канала. Размер FIFO для каждого канала разный:

- канал 1 – 256x32;
- канал 2 – 256x32;
- канал 3 – 64x32;
- канал 4 – 64x32;
- канал 5 – 64x32;
- канал 6 – 64x32;
- канал 7 – 32x32;
- канал 8 – 32x32.

Наличие или отсутствие данных в FIFO контролируется битами статуса DR, HF, FF соответствующего канала. При чтении данных указатель чтения автоматически инкрементируется. При записи данных в FIFO после приема указатель записи инкрементируется. Все FIFO каналов приема физически размещены в одном модуле синхронной статической памяти. Буфер каждого канала имеет смещение относительно начала памяти в соответствии с размером предыдущих буферов. Об этом нужно помнить при прямом доступе в память приемников.

Доступ к памяти

Все буферы приемников и меток расположены физически в одном модуле памяти. Поэтому при попытке обращения к буферу со стороны приемников и процессора может возникать конфликт. Он разрешается следующим образом. Если есть запрос от процессора, он имеет самый высокий приоритет и обслуживается первым. Для выбора обслуживаемого канала используется 3-х битовый счетчик, который циклически выбирает канала для обслуживания. Таким образом, в течение 8-ми тактов любой канала может быть обслужен. Запрос на доступ к памяти канала выставляет при приеме сообщения или при приеме метки. После приема сообщения у канала есть 32 такта времени ожидания на обслуживание. При приеме метки остается 22 такта ожидания. Под тактом понимается время приема одного бита сообщения, т.е. 100 КГц или 12,5 КГц. Поскольку частота шины при этом должна быть как минимум в 10 раз выше (для 100 КГц), то при работе на 1 МГц частоты шины мы имеем 220 тактов частоты шины, чтобы обслужить максимум 16 запросов (8 сообщений и 8 меток). Приостановку обслуживания запросов может вызывать доступ к памяти со стороны процессора. Однако один цикл чтения памяти процессором занимает минимум 3 такта (в действительности больше) из которых

только один есть обращение к памяти, а два других могут быть использованы для обслуживания каналов. Задержку обслуживания может вызвать обращение к памяти по записи со стороны процессора. Запись может выполняться каждый такт шины. Это нужно учитывать при работе с интерфейсом и не выполнять непрерывно записи в буфер приемника большого блока данных.

24.6 Описание регистров передатчика

Таблица 208 – Регистры передатчика

Базовый адрес	номер	Название	Состояние после сброса	Описание
0x8000_3000		ARINC429T0		Контроллер интерфейса передатчиков ARINC429
+0				Канал 0
	0	CONTROL		Регистр управления передатчика
	1	-		
	2	DATA_T		Регистр передаваемых данных
	3	STATUS		Регистр состояния передатчика
+4				Канал 1
+8				Канал 2
+12				Канал 3
0x8000_3400	0x000-0x1FF	Память передатчиков		Прямой доступ в память передатчиков
	0x000	TX0_base		Базовый адрес передатчика 0
	0x100	TX1_base		Базовый адрес передатчика 1
	0x140	TX2_base		Базовый адрес передатчика 2
	0x180	TX3_base		Базовый адрес передатчика 3
	0x1c0	FREE_base		Неиспользуемая область памяти

24.6.1 Регистр управления передатчиком CONTROL

Таблица 209 – Биты регистра управления передатчиком CONTROL

Бит	Имя	Значение	Описание
0	CH_EN		Разрешение работы канала 1 – передача по каналу разрешена 0 – канал передачи находится в состоянии сброса
1	CLK		Скорость передачи данных 1 – частота передаваемых данных = опорная чатота/80 (12,5 кГц) 0 – частота передаваемых данных = опорная частота/10 (100 кГц)
2	EN_PAR		Разрешение 32 бита паритета 1 – разрешена передача 32-м битом бита паритета 0 – разрешена передача 32-м битом бита данных

3	ODD		Выбор четности или нечетности бита паритета 1 – бит паритета формируется как дополнение до нечетности (если сумма всех разрядов данных по модулю 2 равно нулю, бит паритета устанавливается в 1, в противном случае в 0) 0 – бит паритета формируется как дополнение до четности (если сумма всех разрядов данных по модулю 2 равна единице, бит паритета устанавливается в 1, в противном случае в 0)
4	INTE_TXR		Разрешение прерывания при опустошении буфера FIFO 1 – разрешено прерывание FIFO передачи данных пусто 0 – прерывание запрещено
5	INTE_FFT		Разрешение прерывания при полном заполнении буфера FIFO 1 – разрешено прерывание при полном заполнении FIFO данных 0 – прерывание запрещено
6	INTE_HFT		Разрешение прерывания при заполнении наполовину или менее буфера FIFO 1 – разрешено прерывание FIFO наполовину или менее полно 0 – прерывание запрещено
7	ENSYNC		Разрешение работы выходов передатчика в режиме данных и синхросигнала 1 – разрешено 0 – запрещено При установленном бите ENSYNC для соответствующего канала выход OUT_A работает как данные (D), выход OUT_B как синхросигнал (SYN)
15..8	DIV[7:0]		Делитель частоты ядра до 1 МГц Содержит значение, на которое необходимо поделить частоту ядра, чтобы получить 1 МГц

24.6.2 Регистр состояния передатчика STATUS

Таблица 210 – Биты регистра состояния передатчика STATUS

Бит	Имя	Значение	Описание
0	TX_R		Флаг наличия данных в FIFO 1 – FIFO пусто 0 – FIFO содержит данные
1	BUSY		1 – передатчик выполняет передачу данных 0 – передатчик в ожидании новых данных
2	FFT		Флаг полноты FIFO 1 – FIFO полно 0 – FIFO не полно
3	HFT		Флаг наполненности FIFO канала 1 – FIFO наполнено до половины или менее 0 – FIFO наполнено более чем на половину
4	IRQ_TXR		Запрос прерывания если FIFO пусто. 0 – нет запроса 1 – есть запрос

5	-		
6	IRQ_FFT		Запрос прерывания при полном заполнении FIFO. 0 – нет запроса 1 – есть запрос
7	IRQ_HF		Запрос прерывания при заполнении FIFO наполовину или менее. 0 – нет запроса 1 – есть запрос
8	IRQ_TX_CH0		Запрос прерывания от канала 0 0 – нет запроса 1 – есть запрос
9	IRQ_TX_CH1		Запрос прерывания от канала 1 0 – нет запроса 1 – есть запрос
10	IRQ_TX_CH2		Запрос прерывания от канала 2 0 – нет запроса 1 – есть запрос
11	IRQ_TX_CH3		Запрос прерывания от канала 3 0 – нет запроса 1 – есть запрос
15..12	-		
23..16	DATA_WP		Значение указателя на запись данных
31..24	CHAN_RP		Значение указателя на чтение данных

Все передатчики имеют один вход запроса прерывания в контроллере прерываний. Дополнительная информация о конкретном источнике прерываний может быть получена из регистра состояний.

24.6.3 DATA_T

FIFO передаваемых данных

FIFO может содержать данные объемом:
- 256x32 для передачи по каналу 1;
- 64x32 для передачи по каналу 2;
- 64x32 для передачи по каналу 3;
- 64x32 для передачи по каналу 4.

Наличие или отсутствие данных в FIFO контролируется битами статуса TX_R, HFT, FFT.

FIFO доступно по чтению и записи, однако только чтение вызывает инкремент указателя чтения.

Все FIFO каналов передачи физически размещены в одном модуле синхронной статической памяти. Буфер каждого канала имеет смещение относительно начала памяти в соответствии с размером предыдущих буферов. Об этом нужно помнить при прямом доступе в память передатчиков.

25 Контроллер МКПД по ГОСТ Р 52070-2003

В микроконтроллере имеется два независимых контроллера МКПД по ГОСТ Р 52070-2003 (далее 1553), каждый из которых содержит необходимую логику и память для обработки и хранения командных слов и слов данных одного полного сообщения 1553. Каждый контроллер содержит два канала для приёма/передачи сообщений 1553: основной и резервный. В один момент времени может работать только один из каналов основной или резервный. Одновременная работа двух каналов не предусмотрена. Контроллер может работать как в режиме контролера шины, так и в режиме оконечного устройства. Для хранения входящих и исходящих командных и статусных слов, а также команд управления используются 16-разрядные регистры. Для хранения данных используется шестнадцатиразрядная память, в которой данные хранятся в области памяти, соответствующей субадресу командного слова. В каждом субадресе можно хранить только одно полное сообщение 1553. При передаче сообщения данные в память можно заносить как на «лету», так и до начала передачи. При приёме сообщения, данные можно считывать из памяти, как на «лету», так и после установки флага VALMESS.

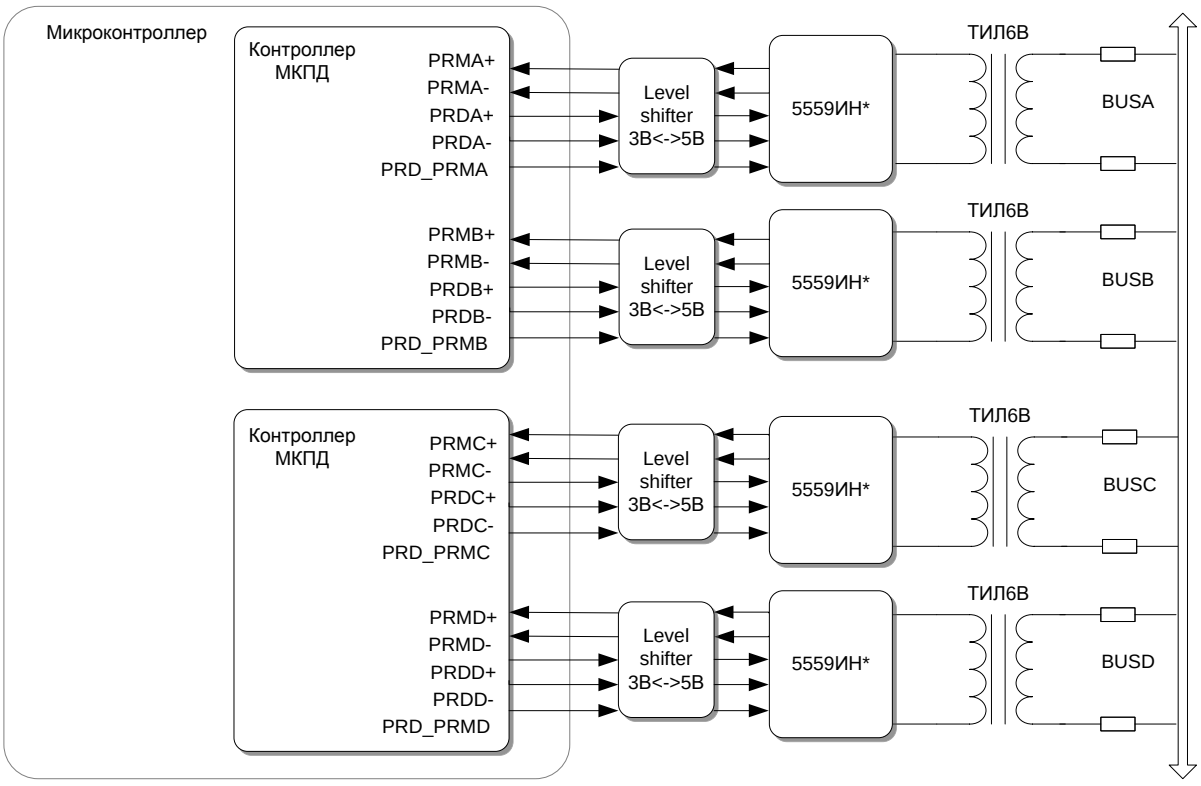


Рисунок 104 – Структурная схема контроллера интерфейса по ГОСТ Р 52070-2003

Особенности:

- поддержка основных (формат 1 – формат 6) и групповых (формат 7 – формат 10) форматов сообщений;
- поддержка режимов работы: контроллер шины, оконечное устройство, монитор;
- скорость передачи данных 1 Мбит/с в полудуплексном режиме;
- поддержка двух каналов связи: основного и резервного;
- двухпортовая память принимаемых данных 1Кх16;

- двухпортовая память передаваемых данных 1Кх16;
- возможность формирования прерываний при успешном приёме и при возникновении ошибок на шине;
- маскирование прерываний.

Назначение внешних выводов контроллера приведено в Таблице 211.

Таблица 211 – Назначение внешних выводов контроллера (интерфейс 0, интерфейс 1)

Обозначение вывода	Назначение вывода контроллера	Тип	Функциональное назначение
PA[26]	MIL0_OU1P	I/O	Интерфейс МКПД0. Выход основного канала (+)
PA[27]	MIL0_OU1N	I/O	Интерфейс МКПД0. Выход основного канала (-)
PA[28]	MIL0_OU1X	I/O	Интерфейс МКПД0. Разрешение передачи основного канала
PA[29]	MIL0_OU2P	I/O	Интерфейс МКПД0. Выход резервного канала (+)
PA[30]	MIL0_OU2N	I/O	Интерфейс МКПД0. Выход резервного канала (-)
PA[31]	MIL0_OU2X	I/O	Интерфейс МКПД0. Разрешение передачи резервного канала
PB[0]	MIL0_IN1P	I/O	Интерфейс МКПД0. Вход основного канала (+)
PB[1]	MIL0_IN1N	I/O	Интерфейс МКПД0. Вход основного канала (-)
PB[2]	MIL0_IN2P	I/O	Интерфейс МКПД0. Вход резервного канала (+)
PB[3]	MIL0_IN2N	I/O	Интерфейс МКПД0. Вход резервного канала (-)
PD[16]	MIL1_IN1P	I/O	Интерфейс МКПД1. Вход основного канала (+)
PD[17]	MIL1_IN1N	I/O	Интерфейс МКПД1. Вход основного канала (-)
PD[18]	MIL1_OU1P	I/O	Интерфейс МКПД1. Выход основного канала (+)
PD[19]	MIL1_OU1N	I/O	Интерфейс МКПД1. Выход основного канала (-)
PD[20]	MIL1_OU1X	I/O	Интерфейс МКПД1. Разрешение передачи основного канала
PD[21]	MIL1_IN2P	I/O	Интерфейс МКПД1. Вход резервного канала (+)
PD[22]	MIL1_IN2N	I/O	Интерфейс МКПД1. Вход резервного канала (-)
PD[23]	MIL1_OU2P	I/O	Интерфейс МКПД1. Выход резервного канала (+)
PD[24]	MIL1_OU2N	I/O	Интерфейс МКПД1. Выход резервного канала (-)
PD[25]	MIL1_OU2X	I/O	Интерфейс МКПД1. Разрешение передачи резервного канала

25.1 Режимы работы

Контроллер поддерживает три режима работы:
- контроллера шины (КШ);
- оконечного устройства (ОУ);
- неадресуемого монитора (М).

25.1.1 Контроллер шины

В этом режиме контроллер передаёт команды в магистраль, участвует в пересылке слов данных, принимает и контролирует ответную информацию о состоянии ОУ. Помимо этого, КШ реализует все команды управления. Для того чтобы реализовать передачу командного слова в магистраль используется регистр CommandWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр CommandWord2. Ответная информация о состоянии ОУ после приёма из магистрали хранится в регистре StatusWord1. А для сообщений формата 3 и 8 помимо этого

применяется регистр StatusWord2. Для передачи и приёма слов данных команд управления (КУ), форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой бита BCMODE и сбросом RTMODE.

25.1.2 Оконечное устройство

В этом режиме контроллер осуществляет проверку достоверности командных слов, поступающих к нему от КШ. Командное слово считается достоверным, если не возникло ошибок в магистрали при его приёме, или если поле «Адрес ОУ» соответствует коду собственного адреса ОУ или коду 11111 (групповая команда). Если командное слово определено как достоверное, то ОУ посылает в линию ответное слово (ОС) и в зависимости от поля «Приём/Передача» принимает или передаёт число данных, соответствующее полю «Число СД/Код КУ». Если же происходит приём от КШ команды управления, то ОУ реагирует в соответствии с форматами сообщений команд управления. Принятое из магистрали командное слово помещается в регистр CommandWord1, а для сообщений формата 3 и 8 принятое второе командное слово помещается в регистр CommandWord2. Ответное слово ОУ для передачи в магистраль помещается в регистр StatusWord1. Помимо этого, для сообщения формата 3, этот регистр содержит принятое ответное слово от другого ОУ. Для передачи и приёма слов данных команд управления, форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой бита RTMODE и сбросом BCMODE.

25.1.3 Монитор

В этом режиме осуществляется прослушивание магистрали и отбор необходимой информации для проведения: технического обслуживания, регистрации эксплуатационных параметров, анализа решаемых задач или обеспечения информацией резервного КШ. Монитор пассивно прослушивает выбранную шину и захватывает весь трафик на шине, но никогда не передаёт информацию на шину. Принятое из магистрали командное слово помещается в регистр CommandWord1, а для сообщений формата 3 и 8 принятое второе командное слово помещается в регистр CommandWord2. Ответное слово ОУ, принятое из магистрали, помещается в регистр StatusWord1. А для сообщений формата 3 и 8 помимо этого применяется регистр StatusWord2. Для приёма слов данных команд управления, форматы сообщений 5, 6 и 10, применяется регистр ModeData. Выбор этого режима работы осуществляется в регистре CONTROL установкой битов RTMODE и BCMODE. Для быстрой расшифровки сообщений можно применить регистр MSG. Каждому формату сообщения на магистрали соответствует определённый код в этом регистре.

25.2 Форматы сообщений

Сообщения, передаваемые по информационной магистрали, имеют формат, соответствующий форматам основных или групповых сообщений. Любые другие типы сообщений, не соответствующие ГОСТ Р 52070-2003, не поддерживаются.

Форматы основных сообщений, приведённые ниже на рисунке, используются для передачи информации предназначенной одному ОУ и предусматривают выдачу ОС. В данном случае КС – командное слово, СД – слово данных, ОС – ответное слово. Времена t1 и t2 формируются аппаратно и не могут быть изменены

программно. Пауза t2 между сообщениями, формируемая КШ, не менее 4 мкс, а пауза перед передачей ОС, формируемая ОУ, в пределах от 4 до 12 мкс. Если после ожидания 14 мкс так и не поступило ОС от ОУ, то фиксируется отсутствие ОС от ОУ и формируется соответствующий признак ошибки.

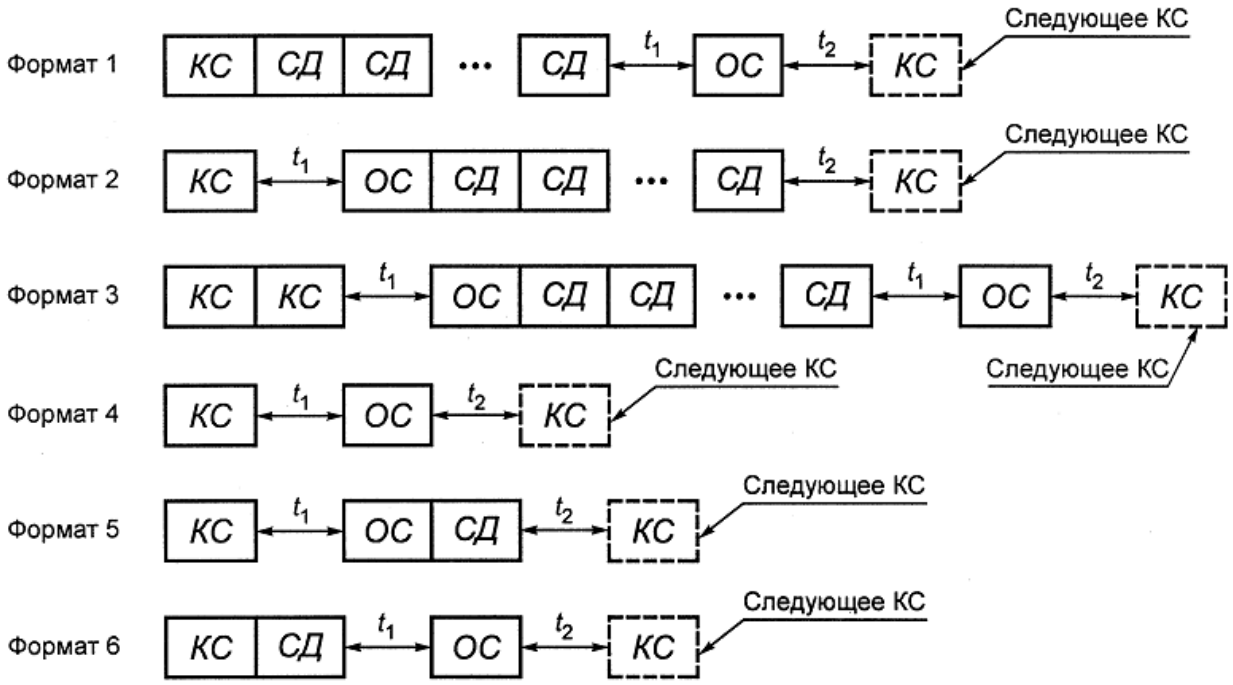


Рисунок 105 – Форматы сообщений

Время непрерывной передачи данных в линию не превышает 660 мкс, что соответствует командному слову и 32-м словам данных.

Формат 1 – передача данных от КШ к ОУ.

Формат 2 – передача данных от ОУ к КШ.

Формат 3 – передача данных от ОУ к ОУ.

Формат 4 – передача КУ.

Формат 5 – передача КУ и приём СД от ОУ.

Формат 6 – передача КУ и СД оконечному устройству.

Групповые сообщения, приведённые ниже на рисунке, начинающиеся с передачи КШ групповой команды с кодом адреса 11111, используются для передачи информации одновременно нескольким ОУ без выдачи ими ОС.

Формат 7 – передача данных (в групповом сообщении) от КШ к оконечным устройствам.

Формат 8 – передача данных (в групповом сообщении) от оконечного устройства к оконечным устройствам.

Формат 9 – передача групповой команды управления.

Формат 10 – передача групповой команды управления со словом данных.

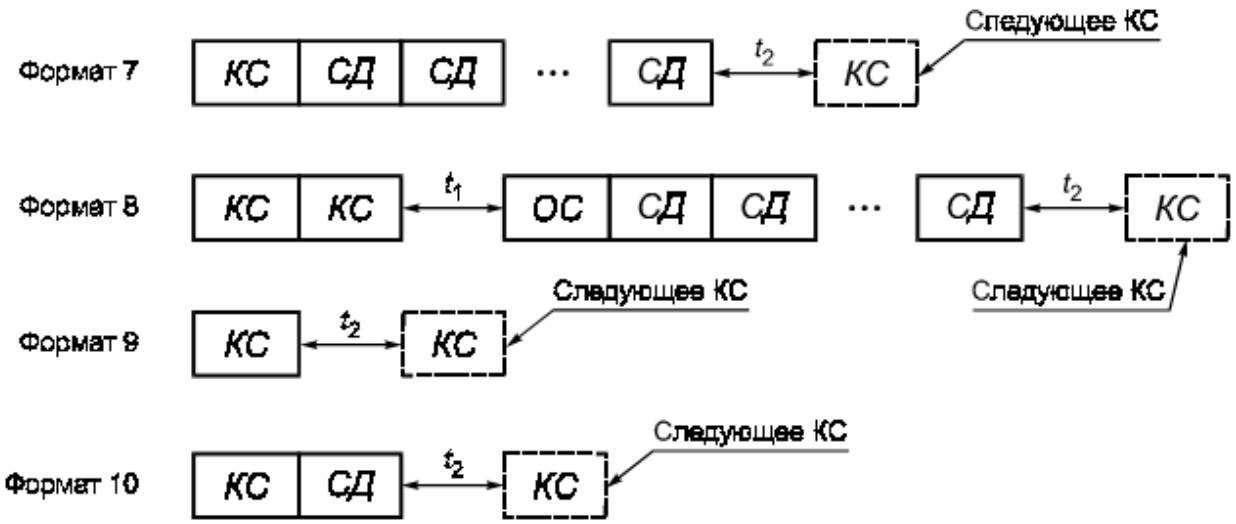


Рисунок 106 – Форматы групповых сообщений

Если ОУ в формате сообщения ОУ-ОУ получило достоверное командное слово на приём информации, то первое СД должно быть им принято через паузу не более (57±3) мкс, в противном случае формируется соответствующий признак ошибки.

25.3 Формат слов

Каждое слово начинается с сигнала пословной синхронизации (с синхросигнала) и имеет 17 информационных разрядов, включая разряд контроля по чётности. Форматы слов приведены на рисунке ниже.

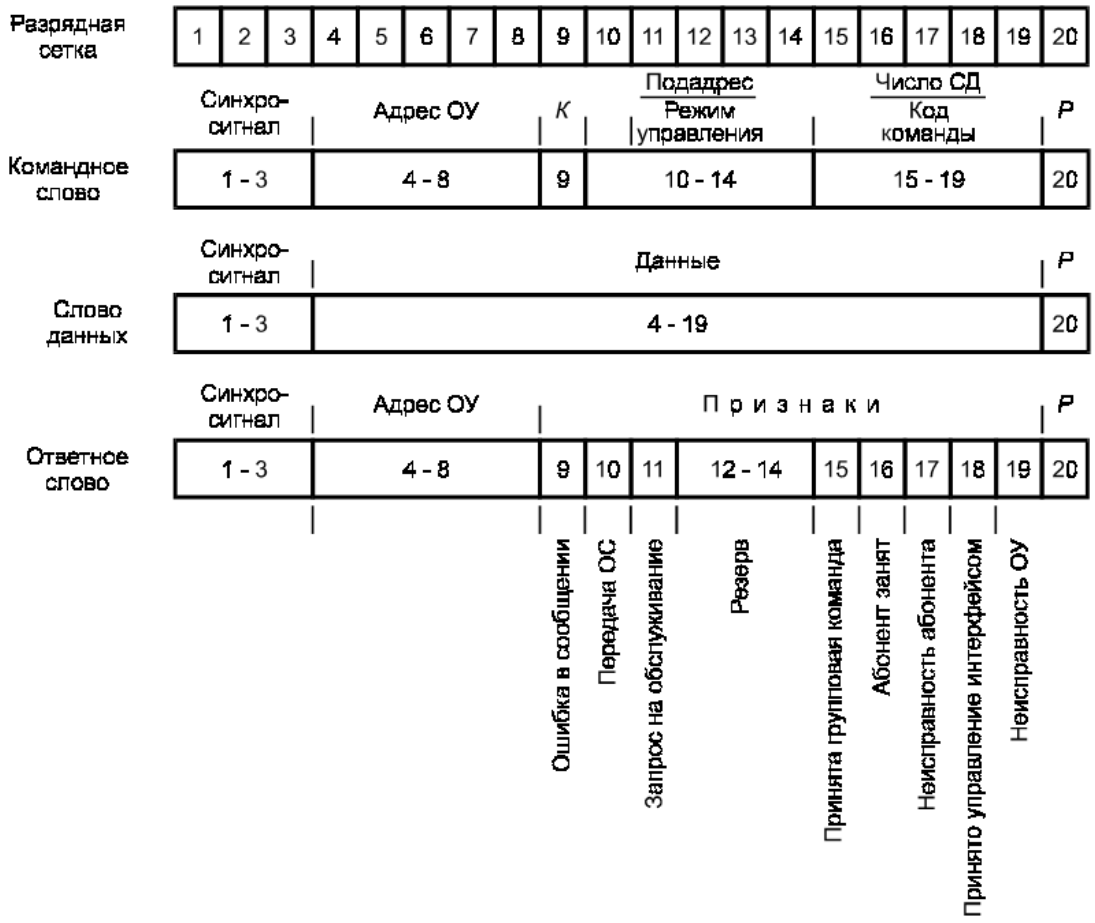


Рисунок 107 – Форматы слов

Командное слово содержит:
- синхросигнал;
- поле «Адрес ОУ»;
- разряд «Приём/Передача»;
- поле «Подадрес/Режим управления»;
- поле «Число СД/Код КУ»;
- разряд контроля по чётности.

Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала положительная, а второй – отрицательная.

Адрес ОУ содержит код адреса из диапазона кодов 00000 – 11110, которому предназначено КС. КС с кодом адреса 11111 называется групповой командой, а сообщение, содержащее групповую команду – групповым.

Разряд «Приём/Передача» указывает на действие, которое должно выполнить ОУ (принимать или передавать СД). Логический нуль означает, что ОУ должно принимать СД, а логическая единица – передавать СД.

Поле «Подадрес/Режим управления» содержит код подадреса ОУ или код признака режима управления 00000 или 11111.

Поле «Число СД/Код КУ» содержит код числа слов данных, которые должны быть переданы или приняты ОУ в связи с приёмом адресованного ему КС, или код КУ. В одном сообщении может быть передано не более 32 СД. Числовое значение двоичных кодов, обозначающих число СД, соответствует их десятичным эквивалентам, за исключением кода 00000, который соответствует числу 32.

Разряд контроля по чётности используется для контроля по чётности предшествующих ему 16 разрядов КС. Разряд принимает такое значение, чтобы сумма значений всех 17 информационных разрядов слова (включая контрольный разряд) была нечётной.

Слово данных содержит:
- синхросигнал;
- данные;
- разряд контроля по чётности.

Синхросигнал имеет длительность, составляющую три интервала времени передачи одного двоичного разряда. Полярность первой половины сигнала отрицательная, а второй – положительная.

Поле данных содержит передаваемые данные, а разряд контроля по чётности формируется так же, как в командном слове.

Ответное слово содержит:
- синхросигнал;
- поле «Адрес ОУ»;
- поле разрядов признаков состояния: ошибка в сообщении, передача ОС, запрос на обслуживание, принята групповая команда, абонент занят, неисправность абонента, принято управление интерфейсом, неисправность ОУ;
- разряд контроля по четности.

Синхросигнал аналогичен синхросигналу КС. Поле «Адрес ОУ» содержит собственный адрес ОУ. Поле разрядов признаков состояния ОУ отображает текущее

состояние ОУ. Разряд контроля по чётности формируется так же, как в командном слове.

25.4 Структурная схема в режиме КШ

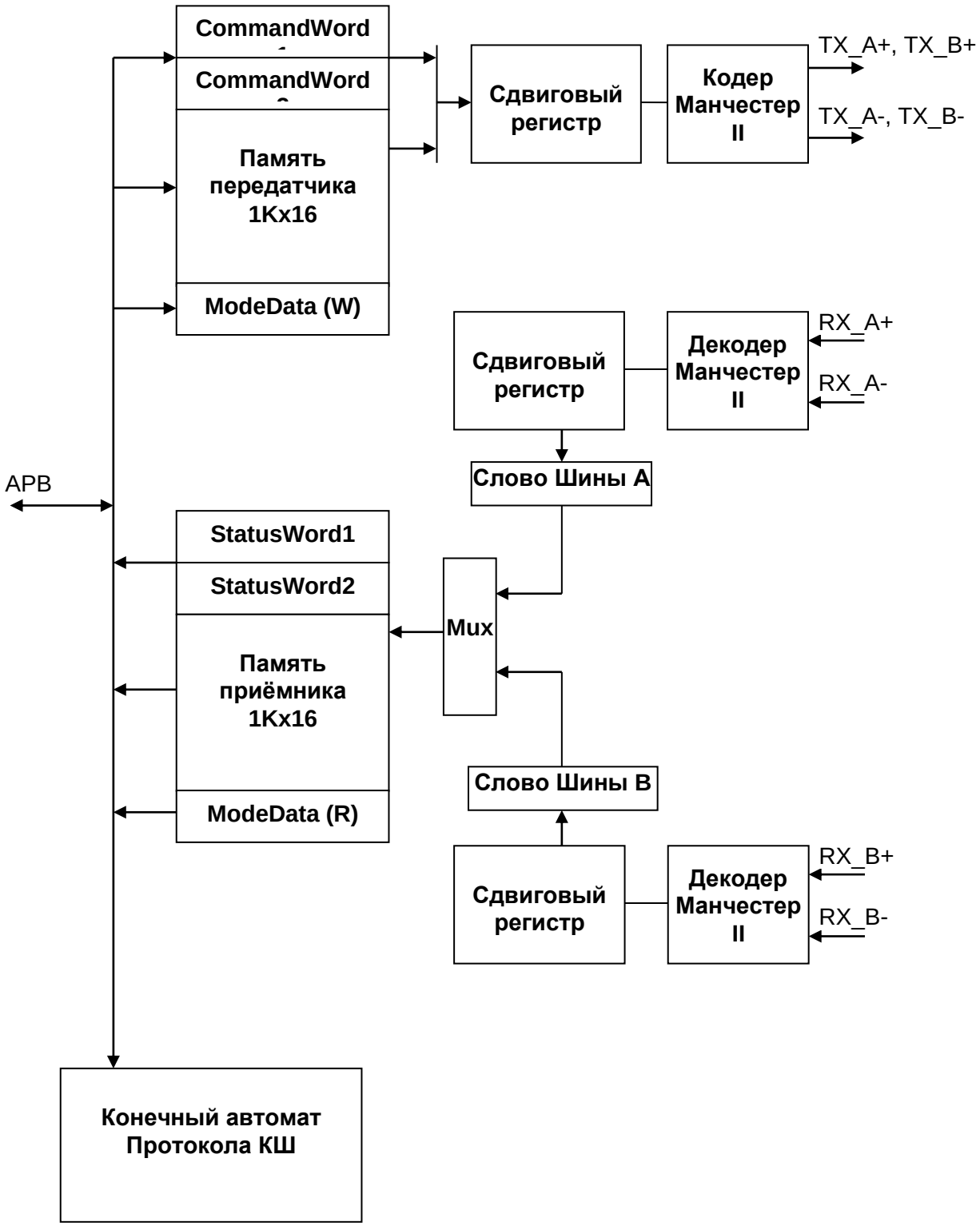


Рисунок 108 – Структурная схема работы в режиме КШ

25.5 Структурная схема в режиме ОУ

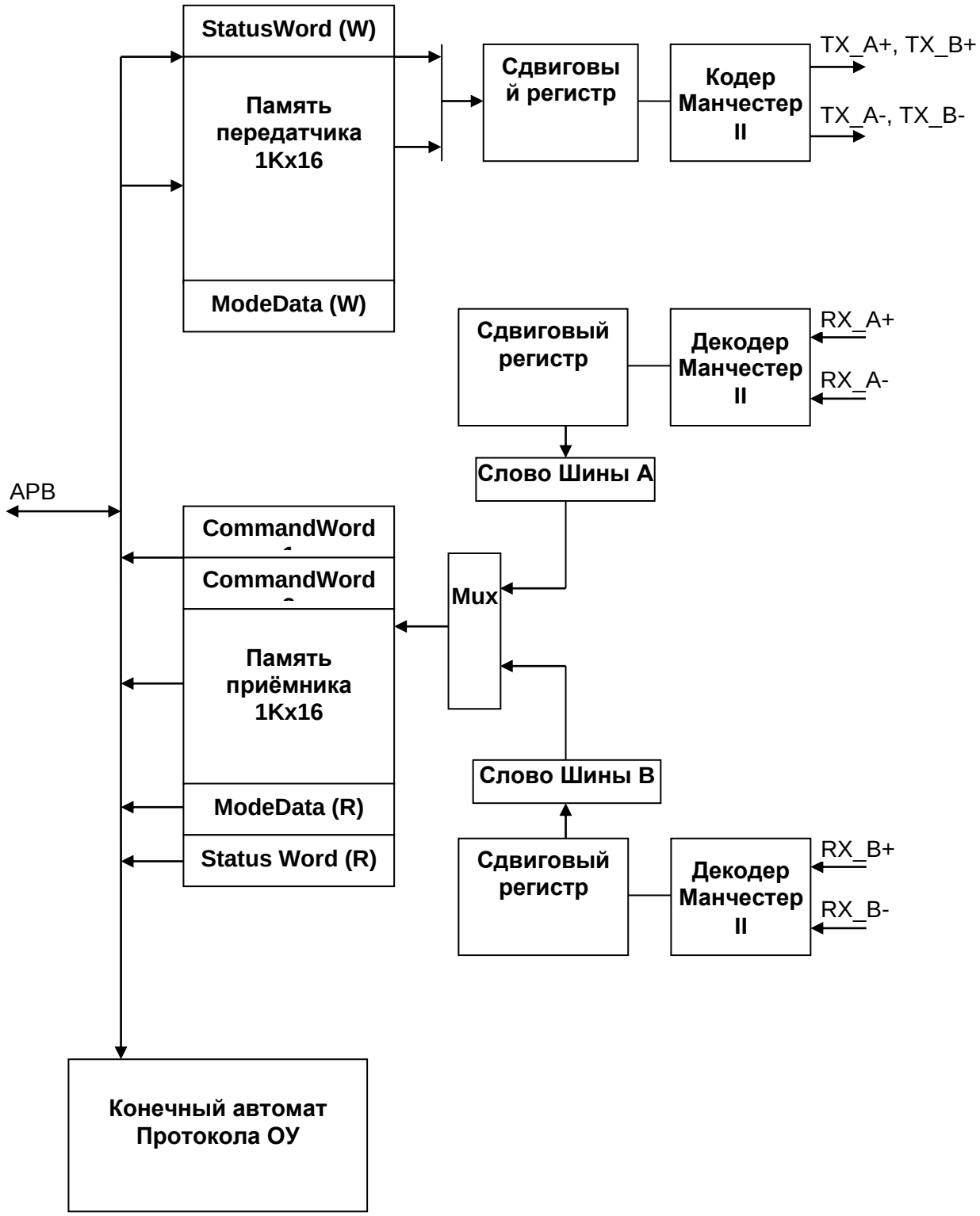


Рисунок 109 – Структурная схема работы в режиме ОУ

25.6 Структурная схема в режиме М

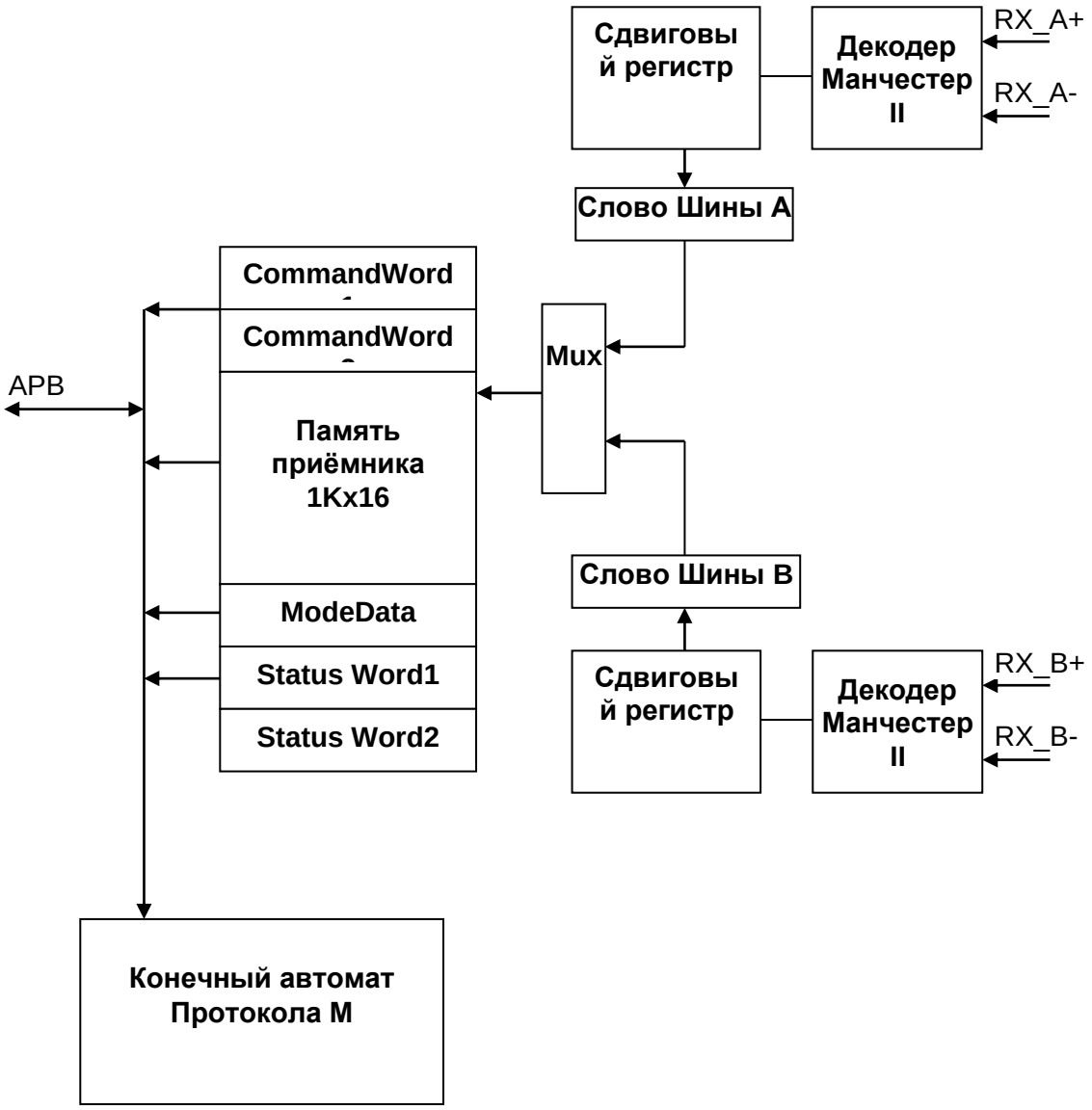


Рисунок 110 – Структурная схема работы в режиме М

25.7 Инициализация

Перед началом работы с контроллером в первую очередь необходимо сбросить контроллер, чтобы очистить все регистры сообщений. Это достигается установкой бита MR регистра CONTROL в логическую единицу. Затем бит необходимо сбросить в нуль. Далее нужно задать в регистре CONTROL значение делителя частоты DIV таким образом, чтобы при делении частоты ядра HCLK на это значение получить опорную частоту блока контроллера 1 МГц. После этого с помощью бит RTMODE и BCMODE выбирается соответствующий режим работы ОУ или КШ.

Для того чтобы выбрать, какой канал будет использован для передачи данных, основной или резервный, устанавливается соответствующий бит (TRA – основной канал, TRB – резервный канал). В режиме КШ командные слова будут передаваться только по тому каналу, который выбран. В режиме ОУ необходимо

установить оба бита, так как ОУ не может выбирать, по какому каналу ему передавать СД и ОС, и поэтому их передача происходит по тому каналу, по которому было принято КС.

Для режима ОУ в битах RTA4 – RTA0 регистра CONTROL задаётся адрес ОУ, который должен соответствовать адресу в поле «Адрес ОУ» командного слова, если идёт обращение к этому ОУ.

Пример инициализации ОУ

```
*((volatile unsigned int *)( 0x40051000))=0x00000001;  //установка  бита MR=1
*((volatile unsigned int *)( 0x40051000))= 0x00014078;
//RTMODE=1, TRB=TRA=1, RTA=1, DIV=40
```

Пример инициализации КШ

```
*((volatile unsigned int *)( 0x40051000))=0x00000001;  //установка  бита MR=1
*((volatile unsigned int *)( 0x40051000))= 0x00014014;
//BCMODE=1, TRA=1,TRB=0, RTA=0,DIV=40
```

В обоих случаях значения делителя частоты DIV = 40, что соответствует частоте ядра 40 МГц, и для получения опорной частоты контроллера необходимо 40 МГц/DIV = 1 МГц.

25.8 Приём и передача в режиме ОУ

Для того чтобы настроить контроллер в режиме ОУ, необходимо выполнить все пункты описанные в параграфе Инициализация. После этого необходимо задать ответное слово для КШ с помощью регистра StatusWord1. В режиме ОУ регистр по записи содержит предназначенное для передачи КШ ответное слово, а по чтению содержит ответное слово, полученное от передающего ОУ в транзакции ОУ-ОУ.

Пример записи ответного слова ОУ

```
*((volatile unsigned int *)(0x40051018))=0x00000800;
```

В данном случае в регистр заносятся только старшие 5 разрядов, соответствующие адресу ОУ. Остальные разряды признаки состояния ОУ можно оставить в нуле. Но в процессе работы может возникать необходимость изменять эти биты. Для этого необходимо программно устанавливать и сбрасывать эти биты, так как аппаратно они не изменяются.

Для того чтобы обеспечить формат сообщения 5, необходимо задавать слово данных, передаваемое КШ в команде управления. Для этих целей используется регистр ModeData.

Пример записи слова данных команды управления

```
*((volatile unsigned int *)(0x40051014))=0x000055AA;
```

После того как проведена инициализация, заданы ответное слово и слово данных команды управления, ОУ сразу готово к работе и может отвечать на все возможные форматы сообщений.

Так как в процессе работы ОУ в каждый момент времени требуется передача определённых СД и СД команды управления, то программно необходимо обновлять те области памяти, которые содержат эти данные. Если эти области не обновляются, то при запросе данных КШ будут переданы те данные, которые были последний раз записаны в эти области памяти. Поэтому при написании программы следует помнить и обновлять данные и слова данных команды управления.

Для хранения СД применяется адресное пространство 0x000-0xFFC (относительно базового адреса периферийного блока). Данные шестнадцатиразрядные, но обращение к ним должно быть выровнено по границе 32-разрядного слова. То есть 2 младших разряда не участвуют в формировании адреса.

Пример инициализации данных для подадреса 1

```
addon=0x80;
for(i=1;i<=32;i++)
{
*((volatile unsigned int *)( 0x40050000+addon))=i;
addon=addon+4;
}
```

Из примера становится ясно, что стартовый адрес памяти СД для подадреса 1 – 0x80, для последующих подадресов: n*0x80, где n- номер подадреса (n=1-31).

При приёме СД или слова данных команды управления, признаком обновления их значений является флаг VALMESS. После того как флаг установлен можно считать новые данные или слово данных команды управления. Но следует учитывать то, что этот флаг автоматически сбрасывается через 4 мкс, после его установки, поэтому желательно применять прерывания по установке сигнала VALMESS.

Пример чтения слова данных команды управления

```
i=*((volatile unsigned int *)(0x40051014));
```

В переменную i будет прочитано значение слова данных команды управления.

Пример чтения данных для подадреса 1

```
addon=0x80;
for(i=1;i<=32;i++)
{
        mas[i]=*((volatile unsigned int *)(0x40050000+addon));
        addon=addon+4;
}
```

Для упрощения декодирования команд управления КШ в режиме ОУ доступен регистр кодов MSG полученных сообщений. Каждому формату сообщения на магистрали соответствует определённый код в этом регистре. Чтение и последующая дешифрация этого кода упрощает процедуру декодирования сообщения и уменьшает время обработки сообщений. При использовании регистра экономится время на чтение двух командных регистров и разбор значений бит этих регистров.

25.9 Приём и передача в режиме КШ

В отличие от режима ОУ, в режиме КШ необходимо задавать не ответное слово, а командное слово или два командных слова в режиме работы ОУ-ОУ. Помимо этого, необходимо инициировать процедуру приёма или передачи данных установкой бита BCSTART. После завершения транзакции на шине этот бит автоматически сбрасывается в ноль. Поэтому для инициирования новой транзакции нужно повторно устанавливать этот бит.

Пример записи командных слов и бита BCSTART

```
*((volatile unsigned int *)(0x4005100C))=0x00000820;      //Командное слово 1
*((volatile unsigned int *)(0x40051010))=0x00000000;      //Командное слово 2
*((volatile unsigned int *)(START_ADDR_APB+0x1000))=0x00014016;  //Регистр управления
```

Как видно из примера, в командном слове 1 задаётся код слов данных 00000, что соответствует 32 СД. Данные будут передаваться от контроллера шины оконечному устройству с адресом 1 из подадреса 1. Второе командное слово задаётся равным нулю и никак не влияет на транзакцию. В регистре управления устанавливается бит BCMODE, что соответствует режиму работы КШ, а также устанавливается бит BCSTART, что инициирует начало транзакции, выбирается канал А для передачи (TRA=1), а также устанавливается делить частоты 40, что соответствует частоте работы ядра 40 МГц.

Для того чтобы инициировать приём в этом примере, необходимо только установить бит 10 равным единице в командном слове 1.

Если транзакция завершена успешно (признак VALMESS установился в единицу), то полученные СД или слово данных команды управления могут быть прочитаны. В противном случае устанавливается один из флагов ошибки. Сброс этих флагов осуществляется установкой битом MR или инициированием новой транзакции битом BCSTART.

25.10 Прерывания

Для уменьшения потерь времени программы на опрос флагов контроллера, введено одно прерывание, генерируемое при установке любого из флагов контроллера. Прерывание может генерировать установка одного из четырёх флагов:

- флаг ошибки;
- флаг успешного завершения транзакции в канале;
- флаг приёма достоверного КС, ОС или слова данных команды управления;
- флаг неактивности контроллера.

Каждый из флагов может быть маскирован битами разрешения прерывания по какому-либо флагу.

25.11 Описание регистров

Таблица 212 – Регистры контроллера МКПД (MIL_STD)

Базовый адрес	Смещение	Название	Описание
0x8000_6000		MIL-STD-1553B0	
0x8000_7000		MIL-STD-1553B1	
	0x000-0x3FF	DATA	Память передаваемых/принимаемых СД
	0x400	CONTROL	Регистр управление контроллером
	0x401	STATUS	Регистр состояния контроллера
	0x402	ERROR	Регистр ошибок контроллера
	0x403	CommandWord1	Регистр командного слова 1
	0x404	CommandWord2	Регистр командного слова 2
	0x405	ModeData	Слово данных команды управления
	0x406	StatusWord1	Регистр ответного слова 1

	0x407	StatusWord2	Регистр ответного слова 2
	0x408	INTEN	Регистр разрешения прерываний
	0x409	MSG	Регистр декодирования сообщений

25.11.1 Регистр управления CONTROL

Таблица 213 – Регистр управления контроллером CONTROL

Номер	31….22	21	20	19	18	17	16
Доступ	U	R/W	R/W	R/W	R/W	R/W	R/W
Сброс		0	0	0	0	0	0
	-	**AUTOTUNE**	**ENFILTER**	**INVTR**	**RERR**	**DIV6**	**DIV5**

Номер	15	14	13	12	11	10	9	8
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	0
	DIV4	**DIV3**	**DIV2**	**DIV1**	**DIV0**	**RTA4**	**RTA3**	**RTA2**

Номер	7	6	5	4	3	2	1	0
Доступ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0	0	1
	RTA1	**RTA0**	**TRB**	**TRA**	**RTMODE**	**BCMODE**	**BCSTART**	**MR**

Таблица 214 – Описание бит регистра CONTROL

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31…22	-	Зарезервировано
21	AUTOTUNE	Бит автоматической подстройки середины битовых интервалов (с ревизии 2) 0 – автоподстройка разрешена 1 – автоподстройка запрещена
20	ENFILTER	Бит разрешения фильтрации потока NRZ (с ревизии 2) 1 – фильтрация разрешена 0 – фильрация запрещена В случае применения драйверов с неккоректной скважностью и длительностью импульсов NRZ кода, необходимо устанавливать бит в единицу для корректного приёма. В этом случае контроль длительностей импульсов NRZ не осуществляется.
19	INVTR	Разрешение инверсии сигналов управления шинными формирователями (сигнал EN для 5559ИН1У) PRD_PRMA, PRD_PRMB 1 – инверсия 0 – прямой выход
18	RERR	Сброс ошибок в режиме ОУ и М 1 – ошибки могут быть сброшены только битом MR 0 – сброс ошибок происходит автоматически, после установки бита IDLE
17…11	DIV6-DIV0	Делитель частоты SOCCLK до 1 МГц Содержит значение, на которое необходимо поделить частоту SOCCLK, чтобы получить 1 МГц. Частота SOCCLK обязательно должна быть не более 120 МГц и кратна 8. Если SOCCLK не кратна 8, то DIV[6:3]=(SOCCLK/8)+1, DIV[2:0]=0, но стабильность приёма не гарантируется.
10…6	RTA4-RTA0	Адрес оконечного устройства

		Содержит адрес, который присвоен устройству, если контроллер работает в режиме оконечного устройства RTMODE=1; BCMODE=0
5	TRB	Блокировка передатчика резервного канала. 1 – передатчик разблокирован 0 – передатчик заблокирован
4	TRA	Блокировка передатчика основного канала. 1 – передатчик разблокирован 0 – передатчик заблокирован
3…2	RTMODE BCMODE	Выбор режима работы контроллера 10 – режим оконечного устройства 01 – режим контроллера шины 11 – режим неадресуемого монитора
1	BCSTART	Инициирует передачу сообщения в канал в режиме КШ. 1 – старт сообщения. 0 – стоп сообщения. Сбрасывается в ноль автоматически по завершению сообщения.
0	MR	Сброс контроллера. 1 – контроллер сбрасывается в исходное состояние 0 – разрешение работы контроллера

25.11.2 Регистр состояния STATUS

Таблица 215 – Регистр состояния STATUS

Номер	31…6	5	4	3	2	1	0
Доступ	U	RO	RO	RO	RO	RO	RO
Сброс		0	0	0	0	0	1
	-	**RCVB**	**RCVA**	**ERR**	**VALMESS**	**RFLAGN**	**IDLE**

Таблица 216 – Описание бит регистра STATUS

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...6	-	Зарезервировано
5	RCVB	Признак активности резервного канала 0 – канал неактивен 1 – канал активен
4	RCVA	Признак активности основного канала 0 – канал неактивен 1 – канал активен
3	ERR	Ошибка в сообщении. 0 – нет ошибок 1 – в последней транзакции возникла ошибка В режиме ОУ и М, если сброшен бит RERR, автоматически сбрасывается не менее чем через 4 мкс после установки.
2	VALMESS	Успешное завершение транзакции в канале. 0 – транзакция завершена с ошибкой, либо транзакции нет в канале 1 – транзакция завершена успешно В режиме ОУ и М автоматически сбрасывается не менее чем через 4 мкс после установки.

1	RFLAGN	Получено достоверное слово из канала. 0 – нет достоверных слов в канале 1 – в режиме КШ получено достоверное ответное слово 1 – в режиме ОУ или М получено достоверное командное слово, ответное слово или слово данных в команде управления Между сообщениями бит автоматически сбрасывается в ноль.
0	IDLE	Состояние контроллера. 1 – контроллер в неактивном состоянии 0 – контроллер в состоянии обмена сообщениями

25.11.3 Регистр ошибок ERROR

Таблица 217 – Регистр ошибок ERROR

Номер	31...7	6	5	4	3	2	1	0
Доступ	U	RO	RO	RO	RO	RO	RO	RO
Сброс	0	0	0	0	0	0	0	0
	-	**PROERR**	**CONERR**	**GAPERR**	**CSYCERR/ SEQERR**	**DSYCERR/ SYNCERR**	**MANERR**	**NORCV**

Таблица 218 – Описание бит регистра ERROR

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...7	-	Зарезервировано
6	PROERR	Ошибка в протоколе в режиме КШ. 1 – недопустимое слово обнаружено на шине во время обмена сообщениями 0 – нет ошибок
5	CONERR	Ошибка непрерывности сообщения. 1 – передача сообщения не непрерывная 0 – нет ошибок
4	GAPERR	Недопустимая активность на шине. 1 – обнаружена активность на шине в интервале 4 мкс после успешного завершения сообщения 0 – нет ошибок
3	CSYCERR/ SEQERR	Ошибка синхронизации команды в режиме КШ (CSYCERR). 1 – ожидался синхроимпульс команды, а получен синхроимпульс данных 0 – ошибок нет Ошибка после приёма команды в режиме ОУ (SEQERR). 1 – обнаружена пауза после приёма командного слова с битом 10 равным нулю или обнаружены слова данных после приёма командного слова с битом 10 равным единице 0 – ошибок нет Ошибка в режиме М (SEQERR). 1 – обнаружено отсутствие ожидаемых данных в сообщении или пауза при приеме первого слова 0 – ошибок нет

2	DSYCERR/ SYNCERR	Ошибка синхронизации данных в режиме КШ (DSYCERR). 1 – ожидался синхроимпульс данных, а получен синхроимпульс команды 0 – ошибок нет Ошибка синхронизации в режиме ОУ и М (SYNCERR). 1 – ожидался синхроимпульс команды, а получен синхроимпульс данных или наоборот 0 – ошибок нет
1	MANERR	Ошибка декодирования NRZ кода. 1 – ошибка в количестве принятых бит или ошибка в бите контроля чётности 0 – ошибок нет
0	NORCV	Ошибка приёма. 1 – не получено ответное слово в интервале 14 мкс или не получены ожидаемые данные 0 – ошибок нет

25.11.4 Регистр команды 1 CommandWord1

Таблица 219 – Регистр команды 1 CommandWord1

Номер	31...16	15…11	10	9…5	4…0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	-	**Адрес ОУ**	**Приём / Передача**	**Подадрес / Режим управления**	**Число СД / Код команды**

Таблица 220 – Описание бит регистра CommandWord1

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...16	-	Зарезервировано
15...11	Адрес ОУ	Адрес оконечного устройства, которому предназначено командное слово
10	Приём/передача	Бит приёма/передачи 1 – режим работы ОУ-КШ 0 – режим работы КШ-ОУ
9..5	Подадрес / Режим управления	Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД. В случае передачи команды, содержит код 00000 или 11111
4..0	Число СД / Код команды	Содержит количество принимаемых или передаваемых слов данных. В случае передачи команды содержит код команды из Таблицы 1 ГОСТ Р52070-2003

Примечание – В режиме ОУ и М регистр доступен только на чтение, в режиме КШ только на запись.

25.11.5 Регистр команды 2 CommandWord2

Таблица 221 – Регистр команды 2 CommandWord2

Номер	31...16	15…11	10	9…5	4…0
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	
	-	**Адрес ОУ**	**Прием/ передача**	**Подадрес / Режим управления**	**Число СД / Код команды**

Таблица 222 – Описание бит регистра CommandWord2

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано
15...11	Адрес ОУ	Адрес оконечного устройства, которому предназначено командное слово
10	Приём/передача	Бит приёма/передачи 1 – режим работы ОУ-ОУ 0 – командное слово не используется
9..5	Подадрес	Содержит подадрес, по которому в памяти располагаются принимаемые или передаваемые СД
4..0	Число СД	Содержит количество принимаемых или передаваемых слов данных

Примечание – В режиме ОУ и М регистр доступен только на чтение и содержит второе командное слово транзакции ОУ-ОУ. В режиме КШ регистр доступен только на запись и используется для транзакции ОУ-ОУ, если установлен в единицу бит 10.

25.11.6 Слово данных команды управления ModeData

Таблица 223 – Регистр слова данных команды управления ModeData

Номер	31...16	15…0
Доступ	U	R/W
Сброс	0	0
	-	**Слово данных команды управления**

Таблица 224 – Описание бит регистра ModeData

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31..16	-	Зарезервировано
15..0	ModeData	Содержит принятое или передаваемое слово данных в команде управления

25.11.7 Ответное слово 1 StatusWord1

Таблица 225 – Регистр ответного слова 1 StatusWord1

Номер	31...16	15...11	10	9	8
Доступ	U	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0
	-	**Адрес ОУ**	**Ошибка в сообщ.**	**Пер.ОС**	**Запр. На обсл.**

Номер	7…5	4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W	R/W
Сброс	0	0	0	0	0	0
	-	**Прин ГК**	**Абонзан.**	**Неисп. Абон.**	**Прин упр. Инт.**	**Неисп ОУ**

Таблица 226 – Описание бит регистра StatusWord1

№	Функциональное имя бита	Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений
31...16	-	Зарезервировано

15...11	Адрес ОУ	Адрес ОУ, от которого принято ответное слово в режиме КШ. Адрес ОУ, которое передаёт ответное слово в режиме ОУ
10	Ошибка в сообщ.	Ошибка в сообщении
9	Пер.ОС	Передача ответного слова
8	Запр. На обсл.	Запрос на обслуживание
7...5	-	Зарезервировано
4	Прин. ГК	Принята групповая команды
3	Абон. Зан.	Абонент занят
2	Неисп. Абон.	Неисправность абонента
1	Прин. Упр. Инт.	Принято управление интерфейсом
0	Неисп. ОУ	Неисправность ОУ

Примечания:

1 Для режима КШ по чтению регистр содержит первое принятое ответное слово, а в случае транзакции ОУ-ОУ по чтению содержит ОС принятое от принимающего ОУ (второе ОС на шине).

2 Для режима М по чтению регистр содержит первое ОС в транзакции на шине.

3 Для режима ОУ по чтению регистр содержит ОС принятое от второго ОУ в транзакции ОУ-ОУ (это может быть как принимающее, так и передающее ОУ).

25.11.8 Ответное слово 2 StatusWord2

Таблица 227 – Регистр ответного слова 2 StatusWord2

Номер	31…16	15…11	10	9	8	7..5	4	3	2	1	0
Доступ	U	RO	RO	RO	RO	U	RO	RO	RO	RO	RO
Сброс	0	0	0	0	0	0	0	0	0	0	0
	-	**Адрес ОУ**	**Ошибка в сообщ.**	**Пер. ОС**	**Запр. На обсл.**	-	**ПринГК**	**Абон зан.**	**Неисп. абон.**	**Прин упр. Инт.**	**Неисп ОУ**

Таблица 228 – Описание бит регистра StatusWord2

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31...16	-	Зарезервировано
15...11	Адрес ОУ	Адрес ОУ, передающего данные в транзакции ОУ-ОУ
10	Ошибка в сообщ.	Ошибка в сообщении
9	Пер. ОС	Передача ответного слова
8	Запр. На обсл.	Запрос на обслуживание
7…5		Зарезервировано
4	Прин. ГК	Принята групповая команды
3	Абон. Зан.	Абонент занят
2	Неисп. Абон.	Неисправность абонента
1	Прин. Упр. Инт.	Принято управление интерфейсом
0	Неисп. ОУ	Неисправность ОУ

Примечания:

1 Для режима КШ доступен только на чтение и в случае транзакции ОУ-ОУ содержит ОС принятое от передающего ОУ (первое ОС на шине).

2 Для режима М доступен только на чтение и содержит второе ОС в транзакции ОУ-ОУ на шине.

3 В режиме ОУ регистр не используется.

25.11.9 Регистр разрешения прерываний INTEN

Таблица 229 – Регистр разрешения прерываний INTEN

Номер	31…4	3	2	1	0
Доступ	U	R/W	R/W	R/W	R/W
Сброс		0	0	0	0
	-	**ERRIE**	**VALMESSIE**	**RFLAGNIE**	**IDLEIE**

Таблица 230 – Описание бит регистра INTEN

№	**Функциональное имя бита**	**Расшифровка функционального имени бита, краткое описание назначения и принимаемых значений**
31..4		Зарезервировано
3	ERRIE	Прерывание при возникновении ошибки в сообщении. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при возникновении ошибок в сообщении
2	VALMESSIE	Прерывание при успешном завершении транзакции в канале. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при успешном завершении обмена данными в канале
1	RFLAGNIE	Прерывание при приёме достоверного слова. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание при приёме достоверного ОС в режиме КШ или достоверного КС, ОС или слова данных команды управления в режиме ОУ
0	IDLEIE	Прерывание неактивности контроллера. 0 – прерывание маскировано 1 – прерывание разрешено, это позволяет генерировать прерывание по переходу контроллера в неактивное состояние

25.11.10 Регистр декодирования сообщений MSG

Таблица 231 – Регистр декодирования сообщений MSG

Номер	15..14	13..0
Доступ	U	RO
Сброс		0
	-	**Код сообщения**

Примечание – Регистр содержит код сообщения, полученного в режиме ОУ или М, и доступен только на чтение. В режиме КШ регистр не используется. Регистр обновляется каждый раз при получении нового достоверного КС.

Таблица 232 – Коды сообщений регистра MSG

Код сообщения	CommandWord1	CommandWord2
Команды обмена данными	15:11 10 9:5 4:0	15:11 10 9:5 4:0
0001 Команда приёма КШ-ОУ, не групповая	RTA 0 00001-11110 XXXXX	
0080 Команда приёма КШ-ОУ, групповая	11111 0 00001-11110 XXXXX	
0004 Команда приёма ОУ-ОУ, не групповая	RTA 0 00001-11110 XXXXX	XXXXX 1 00001-11110 XXXXX
0100 Команда приёма ОУ-ОУ, групповая	11111 0 00001-11110 XXXXX	не RTA 1 00001-11110 XXXXX
0402 Команда передачи ОУ-КШ	RTA 1 00001-11110 XXXXX	
1008 Команда передачи ОУ-ОУ, не групповая	не F 0 00001-11110 XXXXX	RTA 1 00001-11110 XXXXX
0200 Команда передачи ОУ-ОУ, групповая	11111 0 00001-11110 XXXXX	RTA 1 00001-11110 XXXXX
Команда управления		
0410 Код 0-15 K=1 нет данных, не групповая	RTA 1 00000 11111 0XXXX	
0400 Код 0-15 K=1 нет данных, групповая	11111 1 00000 11111 0XXXX	
2420 Код 16-31 K=1 с данными, не групповая	RTA 1 00000 11111 1XXXX	
0040 Код 16-31 K=0 с данными, не групповая	RTA 0 00000 11111 1XXXX	
0800 Код 16-31 K=0 с данными, групповая	1111 0 00000 11111 1XXXX	

25.11.11 Память принимаемых/передаваемых данных DATA

Память принимаемых/передаваемых данных**.**

Примечание – Данные читаются из памяти или записываются в память в соответствии с подадресом (биты с 9 по 5) достоверного командного слова. Каждому подадресу соответствует 32x16 ячеек памяти на приём и 32x16 ячеек памяти на передачу. Общий объём памяти данных 2Kx16.

26 Модуль цифрового смесителя

Модуль цифрового смесителя (далее ЦС) предназначен для приема информации от внешнего RF-приемника сигналов систем GPS/Glonass, предварительной обработки принятых данных и передачи данных во внутреннюю память процессора. ЦС может принимать сигналы от двух внешних RF-приемников. В качестве внешней микросхемы RF-приемника может быть использована, например, gp2015 фирмы Zarlink Semiconductors Inc. Внутренняя структура ЦС представляет собой набор из 16-ти идентичных каналов обработки, которые могут работать параллельно.

Назначение внешних выводов модуля ЦС приведено в Таблице 233.

Таблица 233 – Назначение внешних выводов модуля ЦС

Обозначение вывода	Назначение вывода модуля ЦС	Тип	Функциональное назначение
PC[24]	GPS0_CLKO	I/O	Интерфейс GPS0. Выходной синхросигнал
PC[25]	GPS0_MAG	I/O	Интерфейс GPS0. Амплитуда
PC[27]	GPS0_SIGN	I/O	Интерфейс GPS0. Знак
PC[28]	GPS1_CLKO	I/O	Интерфейс GPS1. Выходной синхросигнал
PC[29]	GPS1_MAG	I/O	Интерфейс GPS1. Амплитуда
PC[31]	GPS1_SIGN	I/O	Интерфейс GPS1. Знак
L0DATIP[6]	GPS0_CLKIP	-	GPS0. Вход синхросигнала (+)
L0DATIN[6]	GPS0_CLKIN	-	GPS0. Вход синхросигнала (-)
L0DATIP[7]	GPS1_CLKIP	-	GPS1. Вход синхросигнала (+)
L0DATIN[7]	GPS1_CLKIN	-	GPS1. Вход синхросигнала (-)

Внутренний интерфейс модуля ЦС представлен в Таблице 234. Описанные сигналы и шины представляют собой интерфейс внутреннего модуля в системе на кристалле.

Таблица 234 – Интерфейс ЦС

Сигнал	Тип	Функция
PCLK	Вход	Синхросигнал периферийной шины
PRESET	Вход	Сброс (активный 0)
PSEL	Вход	Выбор модуля ЦС (активная 1)
PADDR [11:0]	Вход	Адрес регистра
PWRITE	Вход	Признак записи
PWDATA [31:0]	Вход	Шина данных для записи
PRDATA [31:0]	Выход	Шина данных для чтения
DCLK0	Вход	Синхросигнал RF-приемника 0
SAT0[1:0]	Вход	Данные (sign+magn) приемника 0
DCLK1	Вход	Синхросигнал RF-приемника 1
SAT1[1:0]	Вход	Данные (sign+magn) приемника 1
DMA_WE	Выход	Строб записи данных в контроллер DMA
DMA_busy	Вход	Признак занятости DMA
DMA_WA	Выход	Адрес для DMA

[19:0]		
DMA_WD [127:0]	Выход	Данные для DMA
INT	Выход	Запрос прерывания от модуля ЦС

Для сигналов, описанных в таблице, внешними сигналами микросхемы являются только входы от RF-приемников. Регистры модуля доступны посредством интерфейса периферийной шины. Обработанные данные поступают в канал 12 контроллера DMA, далее они направляются в буфер, запрограммированный каналом. Канал 12 DMA задает базовый адрес буфера приема. Смещение внутри буфера формируется каналом ЦС и передается в DMA вместе с данными. Модуль ЦС может формировать запрос прерывания в контроллер прерываний.

26.1 Регистры ЦС

В таблице 235 приведен список регистров ЦС. Регистры доступны со стороны шины периферийных устройств и имеют базовый адрес 0x80005000.

Таблица 235 – Регистры ЦС

Смещение	Имя	Описание	сброс
0-7	CH0	Регистры канала 0	
8-15	CH1	Регистры канала 1	
…	…	Регистры канала 2--14	
0x78-0x7F	CH15	Регистры канала 15	
0x80	CR[31:0]	Регистр управления	0
0x81	SR[31:0]	Регистр состояния	0
0x82	LEN[7:0]	Регистр количества (биты 7:0) принимаемых данных. Задает количество квадрослов данных которые каждый из каналов выдает в контроллер DMA до момента переключения в другой режим. Одинаково для всех каналов. Каждое квадрослово содержит 8 пар (два числа по 8 бит) отсчетов.	
0x83	TST[31:0]	Регистр тестовых данных (биты 31:0). Запись в этот регистр вызывает подачу на корреляторы 8-ми пар (2-х бит) данных. Биты 31:16 канал 1, биты 15:0 канал 0	0
0x84	IRQ[15:0]	Регистр запросов прерываний (биты 15:0). Информирует о переключении канала из одного состояния в противоположное. Биты могут быть сброшены записью 1-го значения.	
0x85	CFG[3:0]	Регистр конфигурации модуля ЦС	

Каждый из каналов может работать с одним из двух буферов. После выгрузки заданного количества квадрослов входных отсчетов (задается регистром LEN) канал переключается на работу с другим буфером. При этом он использует другой набор регистров.

Соответственно, при работе с другим буфером канал может выбрать другой источник сигнала, значение задержки, а также значения шагов.

Длина выгружаемых данных для всех каналов одинакова.

При переключении канала вырабатывается флаг прерывания. Канал может быть обслужен и ему можно задать новые параметры.

26.1.1 Регистр управления CR

Регистр позволяет задать основные режимы работы модуля. Назначение бит регистра приведено в таблице ниже.

Таблица 236 – Регистр управления

Бит	Название	Функция
0	EN0	Разрешение работы канала 0 (1)
…		
15	EN15	Разрешение работы канала 15 (1)
16	IE0	Разрешение прерывания от канала 0 (1 – разрешено)
…		
31	IE15	Разрешение прерывания от канала 15 (1 – разрешено)

После сброса значение регистра управления равно 0.

26.1.2 Регистр состояния SR

Позволяет анализировать состояние модуля. Назначение бит регистра приведено в таблице ниже.

Таблица 237 – Регистр состояния

Бит	Название	Функция
0	PPS0	1 – канал 0 работает с буфером 1 0 – канал 0 работает с буфером 0
…	…	..
15	PPS15	1 – канал 15 работает с буфером 1 0 – канал 15 работает с буфером 0
16	Err0	1 – ошибка в работе канала 0 (переполнение) 0 – нет ошибок
..	…	..
31	Err15	1 – ошибка в работе канала 15(переполнение) 0 – нет ошибок

26.1.3 Регистр количества LEN

Позволяет задавать количество отгружаемых каналом квадрослов данных до момента переключения в другой режим. Назначение бит регистра приведено в таблице ниже.

Таблица 238 – Регистр LEN

Бит	Название	Функция
7:0	LEN	Количество квадрослов данных минус 1
31:8	-	Не используются. При чтении всегда ноль

26.1.4 Регистр конфигурации CFG

Позволяет задавать режим работы модуля ЦС. Назначение бит регистра приведено в таблице ниже.

Таблица 239 – Регистр TSM

Бит	Название	Функция
0	TESTM	Разрешение тестового режима: 1 – разрешено.

Бит	Название	Функция
		0 – запрещено. Нормальный режим работы.
1	NPS0	Выбор активного фронта синхросигнала приемника 0. 0 – переход из 0 в 1 1 – переход их 1 в 0
2	NPS1	Выбор активного фронта синхросигнала приемника 1. 0 – переход из 0 в 1 1 – переход их 1 в 0
3	-	Бит общего назначения
31:4	-	Не используются. При чтении всегда ноль

26.1.5 Регистр тестовых данных TST

Позволяет задавать данные для работы модуля ЦС без использования внешних приемников. Регистр имеет 32 разряда и младшие 16 разрядов используются как данные от приемника 0, а старшие как данные от приемника 1. Как известно разрядность данных приемника равна 2-м битам. В модуле ЦС эти данные поступают на линию задержки имеющей 8 стадий. Использование 16 разрядов данных тестового регистра позволяет задать состояние всей линии задержки конкретного приемника. Каждый канал обрабатывает 2-хбитные входные данные с конкретной стадии линии задержки. Номер стадии программируется.

26.1.6 Регистр запросов прерываний IRQ

Отражает состояние запросов прерываний от каждого из каналов ЦС. Модуль ЦС имеет один общий запрос прерывания в контроллере прерываний, поэтому для определения номера канала, вызвавшего запрос прерывания может использоваться регистр IRQ. Биты с 0 по 15 регистра соответствуют каналам с 0 по 15 модуля ЦС. Старшие 16 бит не используются и при чтении равны 0. Регистр запросов доступен по чтению. При записи возможен только сброс бит регистра. Если при записи какой-то из младших 16-ти разрядов данных равен 1, то соответствующий ему разряд регистра IRQ будет сброшен в ноль. Нулевое значение разряда данных не влияет на бит регистра.

26.2 Регистры канала

Таблица 240 – Регистры канала

Смещение	Имя	Описание	Сброс
0	SC0_CNT	27 бит регистр-счетчик таблицы синуса-косинуса	
1	SC0_STEP	32 бит регистр приращения счетчика SC0_CNT	
2	DZ0_CNT	27 бит регистр-счетчик дециматора	
3	DZ0_STEP	32 бит регист приращения счетчика DZ0_CNT	
4	SC1_CNT	27 бит регистр-счетчик таблицы синуса-косинуса	
5	SC1_STEP	27 бит регист приращения счетчика SC1_CNT	
6	DZ1_CNT	27 бит регистр-счетчик дециматора	
7	DZ1_STEP	27 бит регист приращения счетчика DZ1_CNT	

26.2.1 Регистры SCx_CNT и SCx_STEP

Регистр SC_CNT содержит значение указателя на таблицу синусов-косинусов. Начальное значение указателя можно задать путем записи в регистр. Каждый раз

после чтения константы из таблицы (обработка одного принятого значения), значение указателя наращивается на значение в регистре SC_STEP, т.е. выполняется операция SC_CNT=SC_CNT + SC_STEP[26:0]. Таблица значений синусов-косинусов имеет 8 элементов. Для выбора одного из 8-ми значений используются только биты 26:24 регистра SC_CNT. Значения синусов и косинусов приведены в таблице ниже.

Таблица 241 – Значение таблицы синусов косинусов

SC_CNT	SIN (SATi)				COS(SATi)			
[26:24]	00	10	01	11	00	10	01	11
0	1	-1	3	-3	2	-2	6	-6
1	2	-2	6	-6	1	-1	3	-3
2	2	-2	6	-6	-1	1	-3	3
3	1	-1	3	-3	-2	2	-6	6
4	-1	1	-3	3	-2	2	-6	6
5	-2	2	-6	6	-1	1	-3	3
6	-2	2	-6	6	1	-2	3	-3
7	-1	1	-3	3	2	-1	6	-6

Поскольку только 27 младших бит используются для значения приращения, оставшиеся биты используются для задания других функций канала. Бит SC_STEP[31] выбирает источник сигнала (0 – приемник 0, 1 – приемник 1). Биты SC_STEP[30:28] представляют собой 3-х битовое значение, которое используется для выбора стадии линии задержки, с которой выполняется прием данных для обработки. Пара регистров 0 используется для работы в текущем режиме, а пара 1 для работы в следующем режиме.

26.2.2 Регистрs DZ_CNT и DZ_STEP

Данные регистры используются для задания коэффициента децимации. Каждое входное значение с приемника сигнала, после обработки с использований значений синуса и косинуса, поступает в аккумулятор сигналов, который выполняет накапливание поступающих значений. Значение регистра DZ_CNT позволяет задать количество значений, которые поступают на аккумулятор, для формирования итоговой суммы. Счетчик после каждого такта накопления увеличивает свое значение на приращение DZ_STEP, т.е. выполняется операция DZ_CNT = DZ_CNT + DZ_STEP. Момент переполнения счетчика DZ_STEP, т.е. загрузка в DZ_CNT значения которое имеет старший бит (26-й) равным нулю в момент, когда значение бита DZ_CNT[26] равно 1, вызывает окончание текущего накопления, формирование 2-х отсчетов и начало новых накоплений. Накопленная пара отсчетов поступает в выходной буфер. Когда будут сформированы 8-мь пар отсчетов (128-бит квадрослово), канал выполнит запрос к контроллеру DMA и передаст подготовленное значение. При этом канал также подготовит значение смещение адреса, по которому должна быть выполнена запись данных. Формирование смещения происходит следующим образом. После старта канала его внутренний счетчик QW_CNT начинает отсчет переданных квадрослов. Значение счетчика изменяется от нуля до значения, заданного в регистре LEN, т.е. максимальный диапазон значений от 0 до 255. Состав смещения показан в таблице 18-9.

Таблица 242 – Структура смещения канала

Биты	Значение	Описание
1:0	00	Адрес слова в квадрослове
9:2	QW_CNT	Номер квадрослова
13:10	CH_N	Номер канала

14	PPS	Выбор одного из 2-х буферов. Значение, с которым работает буфер, задается в регистре состояния.
19:15	00000	Всегда нули.

Таким образом, зная номер канала и номер буфера с которым он работает, а также зная максимальное количество формируемых отсчетов, можно определить местонахождение данных в оперативной памяти и выполнить дальнейшую обработку.

После выгрузки заданного значения отсчетов канал переключается на работу со вторым буфером и первый становится доступным для обработки. При этом для второго буфера канал использует новый набор регистров управления.

Факт окончания выгрузки данных в текущий буфер отражается в регистре запросов прерывания и может использоваться для выполнения процессором действий по обслуживанию канала.

Все сформированные данные передаются во внутреннюю оперативную память через DMA канала номер 12. В канале 12 задается базовый адрес массива данных модуля ЦС. Скорость работы канала должна быть достаточной для обслуживания всех 16 каналов ЦС. Если в процессе обслуживания какой-то канал не сможет передать свои данные, будет сформирован признак ошибки.

27 Модуль цифровой обработки UP/DOWN

Модуль цифровой обработки UP/DOWN может быть сконфигурирован для работы в двух режимах:

- режим децимации входного сигнала (DOWN);
- режим интерполяции выходного сигнала (UP).

В режиме DOWN модуль (см. Рисунок 111):

- Принимает информацию от приемника LINK-порта и помещает её во входной буфер. Входная информация представляет собой 16-разрядные числовые данные, упакованные в 128-разрядные слова.
- Извлекает информацию из буфера, производит перемножение данных на значения синусов и косинусов Ti_s, Ti_c из таблицы. Каждое 16-разрядное число умножается на два коэффициента и получается одно комплексное число с 16-разрядными мнимой и действительной частями.
- Перемноженные данные (комплексное число) поступают на входной фильтр (filter_ADD). На данном этапе производится накопление данных.
- После входного фильтра данные поступают на выходной фильтр (filter_SUB).
- Результат выходного фильтра поступает на сдвигатель, где из 86-разрядного входного данного формируется 16-разрядный результат. Два 16-разрядных результата образуют одно комплексное число, которое помещается в выходной буфер. В буфере числа упаковываются в 128-разрядные слова.
- Данные с выходного буфера могут быть выданы на периферийную шину обмена при чтении с использованием процессора либо контроллера DMA. Выходной буфер имеет возможность формирования запроса на обслуживание к контроллеру DMA.

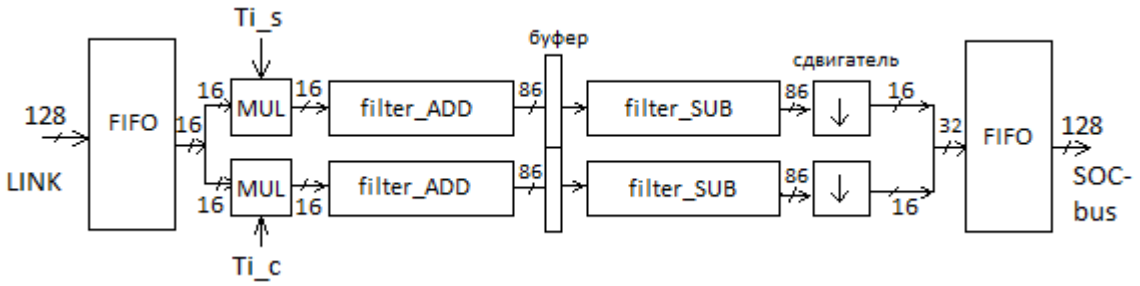


Рисунок 111 – Структура модуля в режиме DOWN

Структура входного и выходного фильтров для режима DOWN приведена на Рисунке 112. На рисунке представлен случай длины фильтра равной 3.

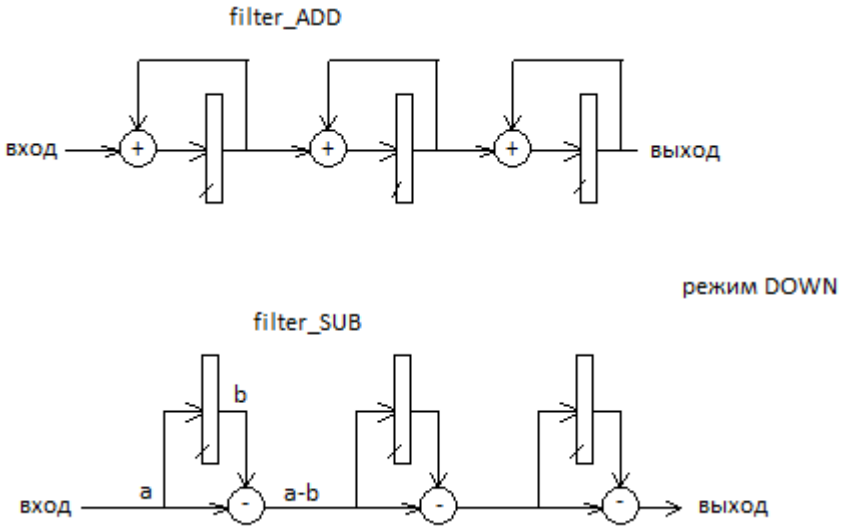


Рисунок 112 – Структура фильтров для режима DOWN

Вообще, итоговая передаточная характеристика каскада фильтров H(z) имеет следующий вид

$$\left(\frac{1}{M} * \frac{1 - Z^{-M}}{1 - Z^{-1}}\right)^{N}, \qquad (1)$$

где M – коэффициент децимации (см. определение поля Kdelay регистра CR);
N – порядок фильтра, определяемый полем FLEN регистра CR.

Фильтр filter_ADD тактируется синхросигналом процессорного ядра и за один такт может производить обработку двух отсчетов. В связи с этим коэффициент децимации, задаваемый параметром Kdelay, может быть только четным. Фильтр filter_SUB тактируется синхросигналом периферийной шины. Выдача одного отсчета производится при передаче одного результата с первичной стадии обработки (от фильтра filter_ADD). Например, если значение Kdelay равно 4, то каждые 5 тактов процессорной частоты будет происходить обработка 10-ти входных отсчетов и передача одного результата на фильтр filter_SUB. Соответственно, каждые 5 тактов процессорной частоты фильтр filter_SUB будет посылать в выходной буфер одно комплексное 32-разрядное число и каждые 20 процессорных тактов (5 тактов * 4 числа) будет формироваться одно квадрослово данных, и посылаться запрос к контроллеру DMA на пересылку данных. Приемник порта связи передает в модуль обработки данные упакованными 128-разрядными словами. При обработке данных со скоростью два 16-разрядные отсчета за один процессорный такт, скорость поступления данных от приемника порта связи не должна превышать одного квадрослова за 4 процессорных такта. При работе с 8-ми разрядной шиной данных, порт связи за каждый такт частоты приемника принимает один 16-разрядный отсчет. Таким образом, за 8 тактов частоты приема порт связи сформирует одно квадрослово. Частота приемника LINK-порта, в этом случае, не должна превышать частоту процессора более, чем в 2 раза. При работе с 16-разрядной шиной данных частота приемника порта связи не должна превышать частоту процессорного ядра, т.к. в противном случае, может произойти переполнение входного буфера модуля обработки.

В режиме UP модуль (Рисунок 113):

- Принимает информацию с шины периферийных устройств (при выполнении операции записи процессором или контроллером DMA) и помещает ее во входной буфер. Информация представлена в виде 32-

разрядного комплексного числа, мнимая и действительная части разрядностью 16 бит. Числа упакованы в 128-разрядные слова.

- Извлекает информацию из буфера и подает их на входной фильтр (filter_SUB). Каждый такт извлекается и обрабатывается одно комплексное число.
- После входного фильтра данные поступают на выходной фильтр (filter_ADD).
- Результат выходного фильтра предварительно сдвигается вправо на заданное число разрядов, с целью выделения значащей 16-разрядной части.
- Результаты работы сдвигателя перемножаются на коэффициенты из таблицы, суммируются и помещаются в выходной буфер. В выходной буфер помещаются 16-разрядные результаты, которые упаковываются в 128-разрядные слова.
- Данные с выходного буфера могут быть выданы в передатчик LINK-порта. При этом инициатором чтения данных из буфера является передатчик LINK-порта.

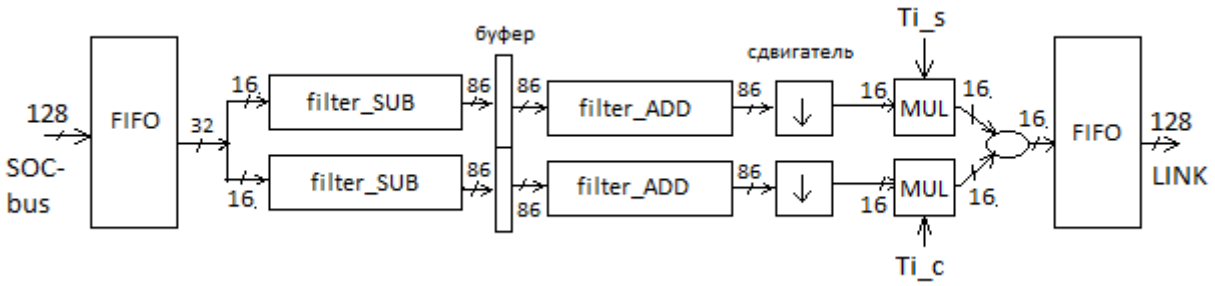


Рисунок 113 – Структура модуля в режиме UP

Структура фильтров filter_SUB и filter_ADD для режима UP аналогична структуре этих же фильтров для режима DOWN.

1 При готовности данных во входном FIFO и готовности промежуточного буфера принять данные, фильтр filter_SUB формирует один отсчет и помещает его в буфер.
2 Получив данные от первичного фильтра filter_SUB, вторичный фильтр filter_ADD формирует несколько выходных отсчетов в соответствии с заданным коэффициентом интерполяции Kdelay.
3 Каждый отсчет (комплексная и мнимая части) поступают на сдвигатель для выделения 16-разрядного результата, затем умножается на коэффициенты синусов и косинусов Ti_s, Ti_c из таблицы синусов и косинусов.
4 Результаты перемножения частей комплексного числа складываются, и получается один 16-разрядный выходной отсчет.
5 Выходные отсчеты упаковываются в 128-разрядные квадрослова.
6 Как только полное 128-разрядное число готово, оно может быть считано передатчиком порта связи и передано во внешнее устройство.

Фильтр filter_SUB работает на частоте периферийной шины, фильтр filter_ADD работает на частоте процессорного ядра и может формировать два выходных отсчета за один такт. Таким образом, каждые 4 такта процессорного клока в выходное FIFO поступает одно квадрослово данных для передачи. Скорость считывания из выходного FIFO не должна превышать скорости записи. В противном случае будет передано предыдущее значение. Скорость продвижения конвейера

обработки определяется степенью наполнения выходного буфера и наличием данных во входном буфере. Если в выходном буфере имеется свободное место, модуль производит обработку и формирует выходной отсчет. Естественно, при наличии входных данных от процессора во входном буфере.

Таблицы синусов и косинусов для DOWN и UP преобразований зашиты в ПЗУ. Таблица SC_table содержит значения синуса-косинуса в одном квадранте (по 256 16-ти битных значений). Таблица CORR_table – коррекция (линейная аппроксимация). 14-битное значение текущего указателя угла ugol[13:0] формирует итоговое значение синуса-косинуса в соответствии со следующим алгоритмом:

- биты ugol[13:6] выбирают грубое значение в квадранте из таблицы SC_table: coarse_sin = SC_table(ugol[13:6]);
- эти же биты выбирают значение угла наклона аппроксимирующей прямой: corr_tan = CORR_table(ugol[13:6]);
- результирующее значение синуса-косинуса будет: sinus = coarse_sin + corr_tan*ugol[5:0];
- само 16-ти битное значение текущего указателя угла ugol формируется из предыдущего прибавлением значения регистра STEP.

27.1 Регистры модуля

В Таблице 243 приведен список всех регистров контроллера. Регистры контроллера доступны со стороны шины периферийных устройств и имеют базовый адрес 0x80000280, 0x80000260, 0x800002A0, 0x800002C0 для модулей UP/DOWN0, -1, -2, -3, соответственно).

Таблица 243 – Регистры модуля

Смещение	Имя	Тип	Значение по сбросу	Описание
0	CR	R/W	0	Регистр управления
1	SR	R	0	Регистр состояния
2	STEP	R/W	0	Регистр шага по таблице коэффициентов
3	-	-	-	Зарезервировано
7-4	DR	R/W	-	Регистр данных
8	RCNT	R/W	0	Счетчик отсчетов
9	XCR	R/W	0	Регистр управления инициализацией от внешних выводов
10	RCNT_STEP	R	0	Составной регистр формата {RCNT_buf, 0x0000, STEP}

27.1.1 Регистр управления CR

Регистр позволяет задать основные режимы работы модуля. Назначение бит регистра приведено в Таблице 244.

Таблица 244 – Регистр управления

Бит	Название	Значение по сбросу	Функция
0	EN	0	Разрешение работы 1 – работа разрешена 0 – работа запрещена

Бит	Название	Значение по сбросу	Функция
1	LINK	0	Источник данных в режиме DOWN 0 – прием данных с APB шины (тестовый режим) 1 – прием данных с ЛИНК порта
2	ROUND	0	1 – округление выходного результата 0 – нет округления
3	SAT	0	1 – насыщение для выходного результата 0 – нет насыщения
4	ROUNDM	0	1 – округление после перемножения 0 – нет округления
5	TBD	0	0 – первый 16-бит отсчет находится в младших битах 128-бит слова 1 – в старших
6	IQ_QI	0	Выдача результата из двух 16-разрядных половин 1 – формат выдачи данных {I[15:0]}, Q[15:0]} 0 – формат выдачи данных {Q[15:0]}, I[15:0]}
7	DAM	0	Режим DOWN либо UP: 0 – DOWN – данные принимаются из линк порта 1 – UP – данные выдаются в линк порт
16:8	Kdelay	0	Коэффициент передачи между фильтрами. Определяет коэффициент интерполяции/децимации, который равен (Kdelay+1)*2. Допустимые значения Kdelay > 0
23:17	SHFR	0	Сдвиг вправо выходного 86-разрядного данного для получения 16-разрядного результата SHFR = FLEN*log2((Kdelay+1)*2)
24	INT_BLK	0	Блокировка прерывания при обнулении RCNT 1 – прерывание запрещено 0 – прерывание разрешено
27:25	Резерв	0	–
29:28	FLEN	0	Длина фильтров 00 – 3 ступени 01 – 5 ступеней 10 – 7 ступеней
30	RCNT_ON	0	Разрешение работы счетчика 1 – работа разрешена 0 – работа запрещена
31	LNKUSE	0	Номер Link-порта для приема (DOWN) или передачи данных (UP)

После сброса значение регистра управления равно 0. Назначение бит регистра зависит от выбранного режима (DOWN или UP).

27.1.1.1 Режим DOWN

Режим работы DOWN задается, если бит DAM равен нулю. В этом случае модуль принимает данные и производит децимацию. Необходимо задать источник данных. Для выбора Link-порта необходимо установить бит LINK. В этом случае входное FIFO находится под управлением приемника Link-порта и способно принимать 128-разрядные данные.

В процессоре имеется два Link-порта. Бит LNKUSE позволяет выбрать 0-й (если 0) или 1-й (если 1) порт. В модуле имеются фильтры обработки данных, для которых можно задать требуемую длину. Биты FLEN позволяют выбрать количество ступеней обработки для фильтров. В фильтре filter_ADD происходит накопление информации. Параметр Kdelay позволяет задать количество накоплений, которые

произведет первый фильтр до того, как полученный результат передаст во второй фильтр. Данный параметр является коэффициентом децимации.

Принятые 16-разрядные данные (упакованные в 128-разрядные слова) представляют собой действительные числа в формате fractional, т.е. знаковые дробные числа K со значением в диапазоне -1<K<1. Принятое число перемножается на 16-разрядные коэффициенты из таблицы. Один коэффициент представляет собой значение синуса, а второй – значение косинуса. При перемножении получается 32-разрядное число MUL, но итоговый результат 16-разрядный.

Бит ROUNDM позволяет задать режим обработки результата умножения MUL:

1 если бит равен нулю, то результат равен MUL[30:15];

2 если бит равен 1, то выполняется округление к ближайшему из указанных бит результата. При этом факт переполнения после округления не отслеживается.

При извлечении 16-разрядного числа из упакованного 128-разрядного порядок следования отсчетов (начиная со старших слов или с младших) в 128-разрядном слове определяет бит TBD.

При выполнении обработки чисел с помощью встроенных фильтров итоговый результат имеет разрядность 86 бит. Для того чтобы преобразовать конечный результат обратно в формат 16-разрядного числа используется сдвиговое устройство, которое выполняет сдвиг входного 86-разрядного результата вправо на значение, заданное полем SHFR.

После сдвига для выходного результата можно задать режим округления с помощью бита ROUND:

1 если бит равен нулю, то в качестве результата используется 16 младших бит результата сдвига;

2 если бит равен 1 – используется округление. Тип округления – biased, т.е. к старшему выдвинутому биту при округлении добавляется значение 1. Если установлен в 1 бит SAT, то при формировании результата учитывается факт его переполнения:

- если число больше, чем максимальное 16-разрядное положительное число, то итоговый результат равен 0x7FFF;
- если число меньше максимального отрицательного числа, то итоговый результат равен 0x8000.

Два полученных 16-разрядных результата образуют одно 32-разрядное комплексное число, которое упаковывается в 128-разрядные слова и записывается в выходной буфер. При этом существует возможность с помощью бита IQ_QI задать расположение мнимой и действительной частей в 32-разрядном слове. Выходной буфер формирует запрос к контроллеру ПДП на обслуживание.

В режиме DOWN можно задать бит источника данных LINK равным нулю. В этом случае данные могут быть загружены с шины периферийных устройств процессором. Этот режим является тестовым и используется для проверки алгоритма обработки.

После определения всех параметров модуль необходимо включить, установив бит EN в 1.

27.1.1.2 Режим UP

Режим работы UP задается, если бит DAM равен единице. В этом случае модуль принимает данные от процессора (КПДП), производит процедуру интерполяции и выдает данные в передатчик Link-порта. Назначение бит в основном

аналогично назначениям при работе в режиме DOWN. Параметр Kdelay в режиме UP является коэффициентом интерполяции. Значение коэффициента определяет количество отсчетов, сформированных выходным фильтром (filter_ADD) до момента обновления его входного буфера результатом из входного фильтра. Бит TBD также позволяет определить, с какого 32-разрядного слова, упакованного в 128-разрядное квадрослово, нужно начинать обработку. Бит выбора Link-порта LNKUSE задет номер порта, в который осуществляется выдача информации.

27.1.2 Регистр состояния SR

Регистр состояния SR позволяет анализировать состояние модуля. Назначение бит регистра приведено в Таблице 245.

Таблица 245 – Регистр состояния

Бит	Название	Функция
0	T_req	Флаг готовности модуля к приему данных 1 – модуль готов принять новые данные 0 – модуль не готов принять новые данные
1	R_req	Флаг готовности модуля выдать данные 1 – модуль готов выдать данные 0 – модуль не готов выдать данные
2	Ferr	Флаг состояния выходного FIFO 1 – переполнение 0 – нет переполнения
3	overf	Флаг состояния входного FIFO: 1 – перепполнение 0 – нет переполнения

Флаг T_req формируется модулем в случае, когда входное FIFO может принять входные данные в режиме UP. В ответ на запрос канал DMA может выполнить пересылку нового квадрослова в приемный буфер.

Флаг R_req формирует запрос к контроллеру DMA в случае, когда в его выходном FIFO (в режиме DOWN) имеются данные. Это позволяет каналу DMA, прочитать данные и отослать их в память процессора.

Флаг Ferr указывает на то, что при работе в режиме DOWN выходное FIFO переполнилось, т.е. скорость потока обработанных данных выше, чем скорость считывания данных контроллером DMA.

Флаг overf устанавливается, если поток данных от приемника Link-порта превышает скорость обработки данных в модуле децимации.

27.1.3 Регистр шага STEP

Регистр шага STEP – 16-разрядный регистр, который содержит приращение счетчика для выбора констант таблицы коэффициентов. Начальное значение регистра равно нулю. Значение регистра модифицируется как записью непосредственно в него, так и записью в регистр RCNT (см. ниже).

Рекомендуемое значение STEP = $65536*F_{пч}/F_{sc}$, где

$F_{пч}$ – промежуточная частота;

F_{SC} – частота дискретизации.

27.1.4 Регистр счетчика отсчетов RCNT

Запись в регистр RCNT всегда осуществляется 64-разрядным словом. Младшие 16 бит слова записываются в регистр STEP, а старшие 32 разряда записываются в регистр RCNT. Регистр RCNT имеет буферный регистр и счетный регистр. При записи по адресу 8, значение записывается сразу в RCNT_cnt и RCNT_buf. Счетчик RCNT_cnt может осуществлять вычитание 1 из своего значения по сигналу модуля. Вычитание происходит в следующих случаях:

- при работе в режиме DOWN в момент, когда в выходное FIFO записывается 128-разрядное слово;
- при работе в режиме UP в момент считывания 128-разрядного слова из входного FIFO для обработки.

В момент, когда значение счетчика становится равным нулю, происходит формирование запроса на прерывание к процессору. Таким образом, модуль может проинформировать процессор о том, что заданное количество квадрослов принято или отправлено. Если бит RCNT_ON регистра управления установлен, после достижения счетчиком значения ноль счетчик перезагружается начальным значением из RCNT_buf и начинает обратный отсчет. Поскольку модуль всегда работает с каналом DMA, можно также использовать прерывание от контроллера DMA для информирования процессора о передаче заданного количества данных.

При чтении по адресу 8 будет возвращено значение {RCNT, 0x0000, STEP}.

27.1.5 Регистр XCR

Назначение бит регистра приведено в Таблице 246.

Таблица 246 – Регистр XCR

Бит	Название	Значение по сбросу	Функция
0	SYNC_SEL	0	Выбор канала синхронизации и инициализации модулятора 0 – сигналы L0ACKI, L0BCMPI 1 – сигналы L1ACKI, L1BCMPI
1	EN_COG_SYNC	0	Разрешение внешней синхронизации модулятора от вывода L#ACKI 1 – синхрониация разрешена 0 – синхронизация запрещена
2	EN_SYNC_RES	0	Разрешение сброса интеграторов/дециматоров от вывода L#BCMPI 1 – сброс разрешен 0 – сброс запрещен
3	EN_COG_PACK	0	Разрешение работы канала DMA: 1 – DMA разрешен 0 – DMA запрещен
4	EN_INT_ACK	0	Разрешение прерывания по входу L#ACKI 1 – прерывание разрешено 0 – прерывание запрещено
5	EN_INT_BCMP	0	Разрешение прерывания по входу L#BCMPI 1 – прерывание разрешено 0 – прерывание запрещено
6	EN_INT_RFIN	0	Разрешение прерывания по обнулению счетчика RCNT 1 – прерывание разрешено 0 – прерывание запрещено

Бит	Название	Значение по сбросу	Функция
7	-	0	Зарезервирован
8	SC_CLR_ACK	0	Разрешение сброса цифрового смесителя по L#ACKI
9	ADD_CLR_ACK	0	Разрешение сброса сумматора по L#ACKI
10	SUB_CLR_ACK	0	Разрешение сброса фильтра по L#ACKI
11	-	0	Зарезервирован
12	SC_CLR_BCMP	0	Разрешение сброса цифрового смесителя по сигналу L#BCMPI
13	ADD_CLR_BCMP	0	Разрешение сброса сумматора по сигналу L#BCMPI
14	SUB_CLR_BCMP	0	Разрешение сброса фильтра по сигналу L#BCMPI
15	EN_LOAD_BCMP	0	Разрешение загрузки начального значения счетчика RCNT внешним сигналом: 1 – L#BCMPI или L#ACKI 0 – L#ACKI

Регистр предоставляет возможность синхронизировать несколько модулей UP/DOWN при построении системы, включающей в себя несколько АЦП или даже процессоров. Используя внешние выводы L0ACKI, L0BCMPI, L1ACKI, L1BCMPI можно инициировать захват данных одновременно на нескольких процессорах в нужное время. Для использования этой возможности необходимо установить в 1 бит EN_COG_PACK и выбрать нужную пару сигналов L#ACKI, L#BCMPI посредством бита SYNC_SEL и один из сигналов L#ACKI, L#BCMPI посредством бита EN_LOAD_BCMPI. Далее необходимо в нужное пользователю время подать импульсы длительностью не менее одного такта SOS-шины, что приведет, в зависимости от значения регистра XCR, к загрузке начальных значений в счетчик RCNT, указатель коэффициентов смесителя, обнулению сумматоров и дифференциаторов фильтров. Следует иметь в виду, что при этом указатель в буфере порта связи не обнуляется, поэтому данные для обработки на DOWN-конвертеры выровнены с точностью до одного квадрослова.

27.2 Подключение модулей UP/DOWN в системе

С помощью модуля можно принимать данные и производить предварительную обработку. Входные данные могут быть получены только с использованием приемников Link-портов. В процессоре имеется 4 модуля UP/DOWN. Структура системы при работе в режиме DOWN представлена на Рисунке 114.

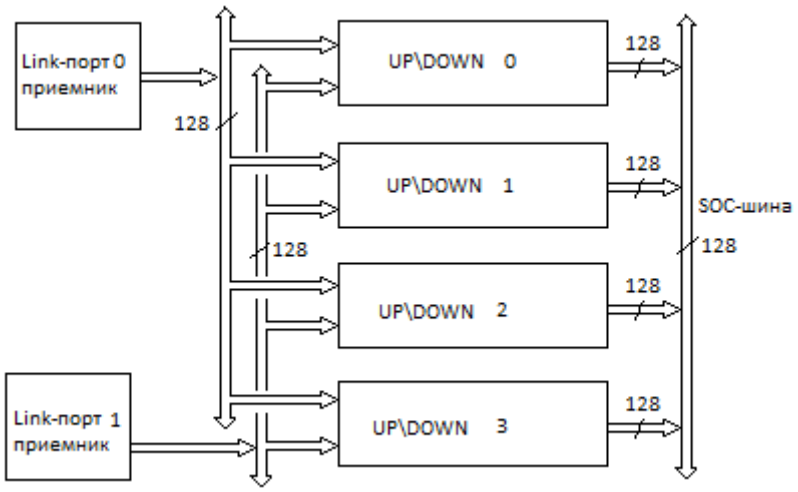


Рисунок 114 – Подключение модуля в режиме DOWN

Как следует из Рисунка 113, каждый из приемников Link-портов может передавать данные модулю для обработки. Выбор конкретного порта осуществляется программированием регистра управления CR. При этом сам модуль Link-порта настраивается на прием данных не в собственный буфер, а в модуль UP/DOWN.

При включении системы желательна следующая последовательность действий:

- выполнение настроек и включение модуля UP/DOWN;
- программирование канала DMA на работу с модулем UP/DOWN;
- включение приемника Link-порта.

Модуль обработки находится в постоянном ожидании данных от приемника Link-порта. Как только приемник включается и начинает принимать данные, модуль начинает обработку принятых данных. При этом с одним и тем же приемником могут работать несколько модулей UP/DOWN, которые могут иметь различные настройки.

При работе модуля в режиме UP структура связей (Рисунок 115) отражает прием данных от контроллера DMA, обработку данных и передачу в передатчик порта связи.

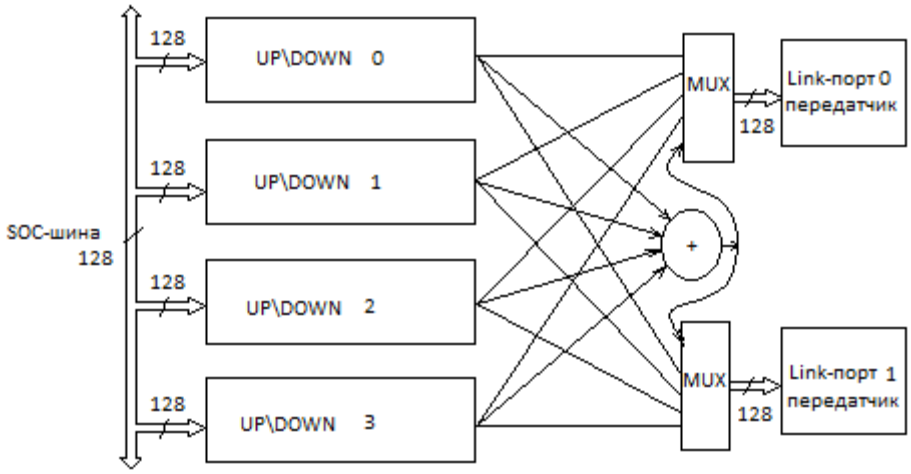


Рисунок 115 – Подключение модуля в режиме UP

Каждый из четырех модулей может быть подключен к передатчику одного из портов связи. Выбор номера порта связи выполняется в регистре управления. Одновременно к передатчику может быть подключен только один модуль. Если ошибочно несколько модулей сконфигурированы для подключения к одному и тому же передатчику, преимущество в подключении имеет модуль 0, затем 1, 2 и 3. Особым образом обрабатывается случай, когда все четыре модуля одновременно подключены к одному и тому же передатчику. В этом случае на передатчик подается сумма выдаваемых значений из всех четырех модулей. Значение суммы округляется и восемь 16-разрядных значений подаются в передатчик выбранного порта. Непосредственное подключение данных от модулей к передатчику выполняется при программировании Link-порта.

28 Порт JTAG и интерфейс отладки

Процессор поддерживает IEEE Standard 1149.1 Joint Test Action Group (JTAG) порт для тестирования системы. Этот стандарт определяет метод последовательного сканирования состояния входов-выходов для каждого компонента системы. Последовательный порт JTAG также используется эмулятором для получения доступа к встроенным возможностям отладки процессора.

Описываемые ниже функции реализованы в аппаратной части процессора для того, чтобы позволить программе-отладчику и операционным системам выполнять более сложные задачи. Функциональность разбита на группы по нескольким уровням. Эти уровни выстроены друг над другом, что позволяет отладчику или ядру ОС осуществлять более качественный контроль и анализ процессора.

Отладчик может осуществлять контроль процессора двумя путями:

- используя отладочный монитор (работающий в режиме супервизора);
- используя встроенный эмулятор (ICE).

При отладке с помощью программного монитора в режиме супервизора, некоторые системные ресурсы используются только монитором (например, память, прерывание, флаги и т. п.).

При отладке с использованием встроенного эмулятора, используется резервный коммуникационный канал (порт доступа к тестированию JTAG) для контроля за процессором.

Порт JTAG является подчиненным устройством на SOC-шине. Многие отладочные функции являются востребованными как при использовании программного метода отладки, так и при аппаратном.

Функции и возможности процессора для поддержки отладки программ включают:

- **Точки наблюдения**

  Позволяет пользователю указывать конкретный диапазон адресов точки наблюдения и условия, которые останавливают процессор. После остановки, пользователь может проанализировать состояние процессора, восстановить прежнее состояние и продолжить выполнение команд. Программные точки останова и одношаговые операции также выполняются с помощью точки наблюдения.

- **Запись истории**

  Специальный буфер трасс позволяет проследить за программным счетчиком с целью восстановления хода программы. Буфер трассировки на кристалле сохраняет последние 32 перехода (все дискретные значения счетчика программ) выполненные устройством управления ядра, позволяя восстановить последний программный путь.

- **Счетчик тактов**

  Обеспечивает функцию подсчета количества процессорных тактов для всех функций программного кода.

- **Мониторинг производительности**

  Позволяет осуществлять мониторинг некоторых внутренних событий ядра процессора при выполнении программного кода. Отладчик может точно указать, какое из внутренних событий нуждается в мониторинге.

- **Доступ к защищенным регистрам**
  Защищенные регистры доступны только в режиме супервизора или ICE. К ним нет доступа в режиме пользователя.

- **Счетчики точки наблюдения**
  Позволяет пользователю искать **n**-е возникновение события до остановки программы.

- **Исключение**
  Событие, которое отменяет выполнение всех последующих команд.

28.1 Рабочие режимы

Процессор работает в одном из трех режимов: режим эмулятора, супервизора или пользователя. В режимах пользователя и супервизора все команды выполняются стандартно. Тем не менее, в режиме пользователя ограничен доступ к регистрам. В режиме эмуляции имеются ограничения на использование некоторых типов команд.

28.2 Ресурсы отладки

В процессоре предусмотрены несколько ресурсов отладки, включая:
- специальные команды;
- точки наблюдения;
- буфер трассировки адреса команды.

28.2.1 Специальные команды

Процессор поддерживает специальные команды, которые используются для помощи в отладке системы. Данные команды нужны для выполнения программных остановов в отладчиках и вызовов операционной системы.

28.2.2 Точки наблюдения

Точки наблюдения адреса позволяют пользователю проверять состояние процессора во всех случаях, когда происходит выполнение условий доступа к памяти, определенной пользователем. Существует три точки наблюдения, которые работают параллельно. Во время работы каждой точки наблюдения пользователь может определять условия срабатывания при доступе к шине и действия, которые должен выполнить процессор.

Точка наблюдения имеет два адресных регистра: WPiL и WPiH. Когда точка наблюдения запрограммирована на один адрес, используется только регистр с младшим адресом (WPiL). Если точка наблюдения запрограммирована на диапазон адресов, в WPiL содержится младший адрес, а в WPiH – старший адрес. Необходимо установить регистры указателя адреса до того, как установлен регистр управления WPiCTL, который переводит точку наблюдения в активное состояние. Адресные регистры точки наблюдения не могут быть изменены, пока точка наблюдения активна.

Работа точки наблюдения определяется ее регистром управления WPiCTL, (где: i – это 0, 1 или 2 в зависимости от номера точки наблюдения).

Следующая информация должна быть запрограммирована в регистрах до начала поиска:

- адрес или диапазон адресов;
- отсчет.

Когда регистр управления устанавливается в активное состояние, поле отсчета устанавливается в начало отсчета. С этого момента аппаратная часть точки наблюдения отслеживается внутренние шины с целью совпадения адресов передач, что указывается в регистрах точек наблюдения.

При каждом совпадении значение счетчика уменьшается. После того, как счетчик достигает значение нуля, инициируется исключение: программное исключение или вызов эмулятора. Выбор действий определяется разрядами 3-2 регистра WPiCTL.

С этого момента регистр состояния указывает на завершение работы точки наблюдения. Чтобы начать новый поиск, пользователю необходимо снова записать регистр управления. Три точки наблюдения связаны с тремя различными шинами.

Точка наблюдения 0 используется для мониторинга на I-шине. Точка наблюдения 0 поддерживает пошаговую опцию (SSTP), которая используется при эмуляции для выполнения одношаговых операций.

Точка наблюдения 1 используется для наблюдения за доступом к данным через J- и K-шины. При этом мониторинг может выполняться для одной из шин или для обеих сразу. Точка наблюдения 1 поддерживает также выбор типа доступа: только чтение, только запись, чтение и запись.

Точка наблюдения 2 используется для наблюдения за доступом к данным через S-шину. Точка наблюдения 2 поддерживает также выбор типа доступа: только чтение, только запись, чтение и запись.

Регистр WPiSTAT указывает на текущее состояние точки наблюдения. Он хранит текущее значение счетчика поиска и режима работы точки наблюдения. Описание разрядов регистра состояния приведено ниже (Таблица 247).

Таблица 247 – Описание разрядов регистра WPiSTAT

Разряды	Имя	Опиание
15–0	VALUE	Значение точки наблюдения
		Определяет текущее значения счетчика точки наблюдения
17–16	EX	Разряд выполнения
		Выполнение точки наблюдения 00 – точка наблюдения запрещена 01 – поиск совпадений 11 – окончание работы счетчика точки наблюдения
31–18	Reserved	Резервные

28.2.3 Буфер трассировки адреса команды (TBUF)

Поскольку программный счетчик недоступен вне кристалла в реальном времени в процессе выполнения команд, то буфер трассировки используется для помощи в отладке программы.

Буфер трассировки включает в себя 32 регистра, в которых хранится история последних от 16 до 32-х переходов, выполненных устройством управления ядра процессора, что дает возможность пользователю полностью восстановить путь выполнения программы в течение последних переходов.

Переходы циклично записываются в регистры буфера трассировки. Первый записывается в TRCB0, следующий в TRCB1, и т. д. до TRCB31. Тридцать второй переход снова записывается в TRCB0, и так далее. Для того, чтобы определить какой из регистров был записан последним, пользователь должен обратиться к регистру указателя буфера трассировки, который указывает на последний записанный вход.

Буфер трассировки всегда запоминает адрес команды, вызвавшей переход. Если адрес перехода вычислялся с использованием дополнительного регистра (не только PC и смешение в коде команды), то кроме адреса команды перехода в буфере трасс запоминается и адрес перехода.

Для того, чтобы различать какой тип информации записан в линии буфера трасс, используется регистр маски TRCBMASK. Каждый разряд в RCBMASK связан с соответствующим регистром TRCBi. Если он сброшен, TRCBi хранит адрес команды перехода. Если же он установлен, TRCBi содержит адрес перехода.

28.3 Мониторинг производительности

Целью мониторинга производительности является оптимизация функционирования рабочих приложений. Аппаратные возможности процессора обеспечивают подсчет числа циклов, которые были затрачены на выполнение кода программы или подсчет числа заданных событий при выполнении программы. Мониторинг обеспечивается с помощью трех регистров: счетчик циклов, маска монитора производительности и счетчик монитора производительности.

Счетчики циклов и монитора производительности должны быть установлены пользователем в ноль перед тем, как начнется определенный период работы. Соотношение между общим счетчиком и счетчиком монитора производительности и дает требуемое значение.

Когда счетчик монитора производительности переполняется, это вызывает исключительную ситуацию.

28.3.1 Регистр маски монитора производительности (PRFM)

Регистр маски монитора производительности (PRFM) определяет, какие события в процессоре отслеживаются и подсчитываются счетчиком монитора производительности (PRFCNT). Значением сброса регистра PRFM является 0x0000 0000. Подробное описание разрядов регистра приведено ниже (Таблица 248).

Таблица 248 – Биты регистра PRFM

Бит	Имя	Назначение
7:0	-	Резерв
8	Jexe	Подсчет команд выполненных на линии конвейера J 1 – включен 0 – выключен
9	Kexe	Подсчет команд выполненных на линии конвейера К 1 – включен 0 – выключен
10	Xexe	Подсчет команд выполненных на линиях конвейера X1 и X2 1 – включен 0 – выключен
11	Yexe	Подсчет команд выполненных на линиях конвейера У1 и У2 1 – включен 0 – выключен

12	Sexe	Подсчет команд выполненных на линии конвейера S 1 – включен 0 – выключен
15:13	-	резерв
16	STALL	Подсчет команд остановов конвейера вызванных ожиданием данных или другими блокирующими конвейер ситуациями 1 – включен 0 – выключен
17	BTB_true	Подсчет количества правильных предсказаний буфера переходов 1 – включен 0 – выключен
18	ABORT	Подсчет количества программных исключительных ситуаций 1 – включен 0 – выключен
19	Uexe	Подсчет количества линий команд выполненных в режиме пользователя 1 – включен 0 – выключен
20	BTB_false	Подсчет количества ошибочных предсказаний буфера переходов 1 – включен 0 - выключен
21	BTB_load	Подсчет количества загрузок в буфер переходов 1 – включен 0 – выключен
31:22	-	Резерв

Все события, которые отслеживаются монитором производительности суммируются по «ИЛИ» и при их наступлении происходит увеличение счетчика PRFM на 1. Логичным является разрешение подсчета одного условия из многих.

28.3.2 Регистр счетчика монитора производительности (PRFCNT)

Назначение счетчика производительности – увеличивать свое значение на 1 каждый раз, когда происходит отслеживаемое событие. Отслеживаемое событие определяется регистром маски монитора производительности (PRFM). Счет прекращается, когда процессор переходит в режим эмуляции. Регистр очищается после сброса.

28.3.3 Регистры счетчика циклов (CCNTx)

Длина счетчика циклов равна 64 разряда, но доступен он как два универсальных 32-разрядных регистра – CCNT0 и CCNT1. После сброса значение счетчика равно нулю.

В связи с тем, что счетчик 64 бита, а доступ возможен только к частям регистра, необходимо соблюдение специальной процедуры чтения и записи регистра. При чтении сначала считывается нижняя часть (CCNT0), а затем верхняя часть (CCNT1). При чтении нижней части верхняя копируется в буфер и это гарантирует ее корректное значение. При записи сначала пишется нижняя часть (она попадает в буфер), а затем верхняя. При записи верхней полное 64-битное значение попадает в счетчик.

Счет прекращается, когда процессор переходит в режим эмуляции.

28.3.4 Регистр данных точки наблюдения 1 (WPDR)

Точка наблюдения 1 имеет дополнительные возможности по мониторингу обмена данными ядра процессора с памятью. Выше описывался механизм срабатывания исключительной ситуации в случае совпадения адресов точки наблюдения с адресами шин J и K. Однако имеется возможность дополнить сравнение адресов еще и сравнением данных. При чтении имеется возможность контролировать то, какие данные считывает процессор. Если дополнительно к совпадению адресов происходит совпадение и прочитанных данных – точка срабатывает. Также можно контролировать и процесс записи данных. Необходимые для контроля данные записываются в регистр WPDR. Необходимо отметить, что шина чтения или записи имеет разрядность 128 бит. Разрядность регистра данных только 32 бита. Поэтому сравнение выполняется только для 32-разрядного слова шины данных определяемого младшими битами шины адреса, независимо от того выполняется чтение или запись 64-х или 128-разрядного слова.

28.3.5 Регистр маски точки наблюдения 1 (WPMR)

При использовании шины данных в описании точки наблюдения 1, не все биты шины данных могут понадобиться при анализе читаемых или записываемых данных. Регистр маски позволяет установить, какие биты регистра данных WPDR должны участвовать в сравнении. Если бит установлен в 1, соответствующий ему бит данных сравнивается с битом шины данных. После сброса регистр маски равен нулю и данные не участвуют в работе точки наблюдения.

28.4 JTAG интерфейс

Процессор поддерживает порт доступа к средствам тестирования, соответствующий стандарту IEEE 1149.1.

28.4.1 Выводы JTAG порта

Таблица 249 описывает выводы процессора, используемые в аппаратной части эмуляции JTAG. Подчеркнутые снизу названия выводов и выводы с чертой над названием указывают на активные низкие сигналы. Для устойчивой работы JTAG-интерфейса необходимо, чтобы частота TCK была как минимум в два раза ниже частоты SOCCLK (минимум в четыре раза меньше частоты CCLK).

Таблица 249 – Выводы порта связи и эмуляции устройства ввода/вывода

Сигнал	Тип	Описание
TRST	вход	Сброс JTAG Сбрасывает интерфейс JTAG. Сигнал TRST должен быть активирован после включения питания, чтобы убедиться в правильной работе JTAG. Вход TRST имеет внутренний повышающий резистор
TCK	вход	Синхронизация JTAG Обеспечивает тактовый сигнал для работы JTAG
TDI	вход	Входной сигнал данных JTAG Имеет внутренний повышающий резистор
TDO	выход	Выходной сигнал данных JTAG
TMS	вход	Выбор режима тестирования JTAG Управляет машиной состояний тестирования. Имеет внутренний повышающий резистор
EMU	выход	Эмуляция Признак перехода процессора в режим эмуляции

В дополнение к стандартным функциям JTAG, выводы TMS и EMU выполняют следующие функции отладки:

- Вход TMS (когда разряд TEME установлен в регистре EMUCTL) выполняет функцию запроса к процессору на переход в режим эмуляции. Исключение эмуляции формируется при каждом нарастающем фронте сигнала TMS. Данная функция является дополнительной к функциональности TMS в JTAG.
- Выходной сигнал EMU является сигналом с открытым стоком, который активизируется, только если разряд EMUOE установлен в регистре EMUCTL. Вывод EMU указывает на следующее:
  - Произошло срабатывание точки наблюдения, но регистр управления имеет в поле типа (поле EXTYPE в WPiCTL) значение ноль. Уровень сигнала EMU в этом случае снижается на 20 циклов CCLK.
  - Каждый раз, когда процессор находится в режиме эмуляции и готов к выполнению новой линии команд, подаваемой в регистр EMUIR, уровень сигнала на выводе EMU снижается до тех пор, пока такая команда добавляется.

28.4.2 Группа регистров эмулятора JTAG

Базовый адрес регистров 0x8000_03A0. Список доступных регистров приведен в таблице 250.

Таблица 250 – Группа регистров JTAG

Имя	Описание	Смещение	Значение по умолчанию
EMUCTL	Управление	0x00	0x0000 0000
EMUSTAT	Состояние	0x01	0x0000 0002
EMUDAT	Данные	0x02	Не определено
IDCODE	Код ID	0x03	0x427A F0CB
EMUIR:0	Команда 0	0x04	0x33C0 0000
EMUIR:1	Команда 1	0x05	0x33C0 0000
EMUIR:2	Команда 2	0x06	0x33C0 0000
EMUIR:3	Команда 3	0x07	0xB3C0 0000
-	-	0x08	-
OSPID	OSPID	0x09	Не определено
-	-	0x0A– 0x1F	-

28.4.2.1 Регистр управления (EMUCTL)

Регистр EMUCTL устанавливает параметры эмуляции. Значением сброса регистра EMUCTL является 0. Большинство разрядов регистра EMUCTL используются эмулятором и не должны модифицироваться программой. Единственными разрядами, предназначенными для использования программой, являются SWRST (программный сброс) и BOOTDIS (запрет загрузки). Подробное описание разрядов регистра приведено ниже (Таблица 251).

Таблица 251 – Регистр EMUCTL

Бит	Имя	Назначение
0	EMEN	Включение эмулятора 1 – включен 0 – выключен
1	TEME	Разрешение генерации запроса к процессору по входу TMS: 1 – разрешено 0 – запрещено
2	EMUOE	Разрешение активной выдачи информации на выход TDO 1 – разрешено 0 – запрещено
3	-	Бит общего назначения
4	SWRST	Запрос программного сброса: 1 – сброс 0 – рабочий режим
5	BOOTDSBL	Запрещение загрузки 1 – запрещено 0 – разрешено Установка данного бита запрещает загрузку из внешнего EPROM.
31:6	-	Всегда ноль

28.4.2.2 Регистр состояния эмуляции (EMUSTAT)

Регистр EMUSTAT отражает состояние эмуляции. Значением сброса регистра EMUCTL является 0x0000 0002. Подробное описание разрядов регистра приведено ниже (Таблица 252).

Таблица 252 – Регистр EMUSTAT

Бит	Имя	Назначение
0	EMUMOD	1 – процессор в состоянии эмуляции 0 – нормальный режим работы
1	IRFREE	Флаг готовности команды в регистре EMUIR 1 – команда готова 0 – нет
2	INRESET	Флаг сброса эмулятора
31:3	-	Всегда ноль

28.4.2.3 Регистр типа кристалла и кода ID (IDCODE)

Регистр IDCODE определяет ID модели процессора и типа кристалла. Регистр доступен только по чтению и имеет значение 0x427AF0CB.

28.4.3 Регистр команды JTAG

Регистр команды JTAG имеет длину, равную пяти разрядам, и позволяет контроллеру доступа порта (TAP) выбрать любой из сканируемых регистров данных. Младший значащий бит первым выдвигается на TDO. Кодирование команд приведено ниже (Таблица 253).

Таблица 253 – Декодирование команды доступа порта JTAG

Если значение IR4–0 равно...	Соответствующая команда...	Выполнить...
00000	EXTEST	выполнить EXTEST
10000	Sample/Preload	Выполнить сэмплирование и предзагрузку
01000	EMUIR	Сканирование в регистре EMUIR
10100	EMUDAT	Сканирование в регистре EMUDAT
01110	IDCODE	Сканирование кода регистра ID
11110	SAMPLEPC	Сэмплирование ПК, к которому подключен процессор
10001	EMUTRAP	Выполнить перехват эмуляции
10011	OSPID	Сканирование регистра OSPID
00100	EMUCTL	Сканирование регистра EMUCTL
01100	EMUSTAT	Сканирование регистра EMUSTAT
11111	BYPASS	Параллельный режим

28.4.4 Регистры данных

Регистры данных выбираются через регистр команд. Как только соответствующее значение регистра данных записывается в регистр команд, и состояние TAP изменяется на SHIFT-DR, данные выбранного регистра начинают выдвигаться на TDO и приниматься с TDI. Больше деталей в спецификации стандарта IEEE 1149.1.

Регистры данных:

- **Обход (Байпас)**
  Регистр, отвечающий требованиям IEEE 1149.1, является одноразрядным регистром.

- **Пограничные регистры**
  Регистры, используемые многими командами JTAG. Все три команды JTAG соответствуют требованиям IEEE 1149.1

- **EMUIR**
  Команда JTAG загружает 48 разрядов, из которых 32 наименьших значащих разряда загружаются в регистр EMUIR. EMUIR является 128-разрядным регистром. В режиме эмуляции регистр загружается четыре раза последовательно (одной командой JTAG) или загружается до тех пор, пока в загружаемом слове не установится разряд EX (тогда остальные слоты EMUIR считаются командами NOP). Каждый раз, когда данные загружаются в EMUIR, сначала загружаются разряды 31–0, затем загружаются разряды 63–32 и так далее.

Когда четыре загрузки EMUIR завершены (или линия команд была завершена установленным разрядом EX) команды, загруженные в JTAG, передаются в устройство управления ядра и выполняются.
Когда счетверенное слово загружается в EMUIR, оно должно включать одну полную линию команд. Линия команд не должна продолжаться в следующем счетверенном слове. Лишние слоты должны быть заполнены NOP командами.
Запись в EMUIR является недопустимой операцией, когда процессор не находится в режиме эмуляции.

- **EMUDAT**
  Это 48-разрядный регистр, в котором 32 старших значащих разряда используются для сканирования Ureg с таким же именем. Эмулятор использует этот регистр для чтения и записи всех остальных регистров в процессоре.
- Эмулятор также использует данный регистр для доступа к памяти. Должна быть команда, использующая IALU для генерации адреса доступа к памяти, назначением или источником которой является EMUDAT. Когда эмулятор выполняет серию доступов к памяти, процессор должен вставить циклы удержания для того, чтобы завершить все доступы (как во время нормальной работы).

29 Модуль управления синхронизацией и энергопотреблением

Процессор содержит специальный модуль (CMU) для задания различных режимов работы системы. Возможно индивидуальное задание частоты работы периферийных устройств, частоты процессора, частоты шин обмена. Возможна реализация режимов с пониженным энергопотреблением.

Данный модуль подключен к шине периферийных устройств и имеет базовый адрес **0x800001C0**.

Набор регистров управления/состояния представлен ниже (Таблица 254).

Таблица 254 – Регистры модуля CMU

Номер	Обозначение	Разрядность	Тип	Назначение
0	CFG1	32	r/w	Регистр конфигурации периферийных модулей
1	CFG2	16	r/w	Управление синтезатором частот CPU_PLL
2	CFG3	16	r/w	Управление синтезатором частот BUS_PLL
3	CFG4	2	r/w	Управление выбором частот процессора
4	CFG5	32	r/w	Управление синтезатором частот LINK_PLL
5	SYS_STS	16	r/w	Флаги состояния / Сброс флагов состояния
7-6	-			
10-8	CFG8	32	r/w	Управление синхронизацией периферийных устройств

Процессор использует для своей работы внешний вход тактовой частоты XTI. Данная частота используется блоком RTC в качестве рабочей частоты, а также в качестве опорной для внутренних умножителей частоты CPU_PLL, BUS_PLL и LINK_PLL. CPU_PLL формирует рабочую частоту для ядра процессора, которая далее делится, образуя частоту SOC-шины и периферийных устройств, а также частоту работы контроллеров. BUS_PLL формирует частоту работы внешнего интерфейса (SRAM, SDRAM). LINK_PLL формирует рабочую частоту для передатчиков портов связи (LINK-портов). Следует иметь в виду, что для достижения желаемой выходной частоты работы LINK-портов F_{LINK} следует запрограммировать LINK_PLL на выдачу частоты $2*F_{LINK}$. Алгоритм программирования всех PLL одинаков и более подробно описан в подразделе 29.2.1 «Регистры CFG2, CFG3, CFG5».

Структура формирования частот процессора показана ниже (Рисунок 116). Значение коэффициента умножения для каждой PLL задается посредством программирования соответствующих регистров конфигурации. Периферийные устройства используют в качестве базовой частоты частоту SOCCLK. Далее внутри периферийного устройства с помощью регистров конфигурации может быть получена требуемая частота работы устройства.

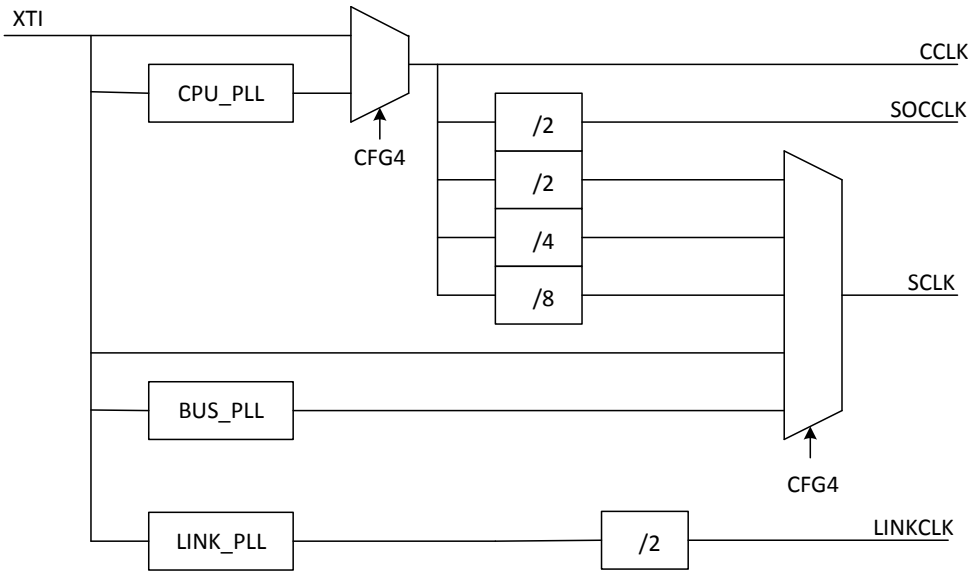


Рисунок 116 – Источники тактовой частоты

После сброса модули PLL отключены, и процессор начинает работу на частоте XTI. Далее программа должна задать необходимые параметры частоты работы для PLL, выждать требуемое время необходимое для стабилизации выходной частоты и затем переключиться на частоту PLL.

Таким образом, процессор имеет несколько частотных доменов, т.е. определенных модулей, функционирующих на собственной частоте:

- процессорное ядро;
- шина периферийных устройств, работающая всегда на половине частоты ядра;
- интерфейс внешней памяти, работающий на собственной частоте;
- часы реального времени, работающие от внешней базовой частоты XTI;
- порты связи, работающие на программируемой частоте.

29.1 Регистр конфигурации периферийных модулей CFG1

Регистр позволяет задавать различные параметры работы периферийных модулей процессора.

Таблица 255 – Регистр CFG1

Биты	Обозначение	По сбросу	Назначение
0	-	0	
1	-	0	
2	LINK_BW	0	Ширина шины линк-портов 0 – ширина шины задается битами регистра управления порта связи. 1 – ширина шины порта связи равна 4 битам
3	SYS_WE	0	Разрешение многократной записи регистров SYSCON,SDRCON 0 – однократная запись 1 – многократная запись
4	ARINC_T0_EN	0	Подключение передатчика 1 контроллера ARINC к внешним контактам: 1 – подключено
5	ARINC_T1_EN	0	Тоже для передатчика 2
6	ARINC_T2_EN	0	Тоже для передатчика 3

Биты	Обозначение	По сбросу	Назначение
7	ARINC_T3_EN	0	Тоже для передатчика 4
8	MIL_DIS	0	Отключение передатчика МКПД от внешних контактов: 1 – отключен
11-9	-	0	
12	DMA_HP_EN	0	Разрешение высокоприоритетному каналу DMA иметь преимущество перед процессором при доступе к внешней памяти: 1 – разрешено 0 – процессор всегда имеет высший приоритет
13	H_OFF	0	Запрещение работы хост-интерфейса 0 – хост-интерфейс разрешен 1 – хост интерфейс запрещен
14	SPI1_EN	0	Разрешение работы интерфейса SPI1 0 – интерфейс запрещен 1 – интерфейс разрешен
15	SPI2_EN	0	Разрешение работы интерфейса SPI2 0 – интерфейс запрещен 1 – интерфейс разрешен
16	I2C_ALT	0	Коммутация интерфейса I2C 0 – SCL = PC[12], SDA = PC[11] 1 – интерфейс I2C отключен
18-17	NAND_ALT	00	Альтернативное подключение контроллера NAND Flash на порты общего назначения PA 00 – NAND не подключен 01 – запрещенная комбинация 10 – подключен, но пользоваться нельзя. Интерфейсы SPI могут работать. NAND удерживает неактивными сигналы выборки и чтения-записи. 11 – подключен и может использоваться
19	GTMR0_EN	0	Разрешение работы таймера 0 с функцией Захвата/ШИМ 0 – таймер запрещен 1 – таймер разрешен
20	GTMR1_EN	0	Разрешение работы таймера 1 с функцией Захвата/ШИМ 0 – таймер запрещен 1 – таймер разрешен

29.2 Регистры управления внутренними PLL

Процессор имеет синтезатор тактовой частоты CPU_PLL для формирования синхросигналов процессорного ядра, синтезатор тактовой частоты BUS_PLL для формирования синхросигнала контроллера внешней памяти, синтезатор тактовой частоты LINK_PLL для формирования синхросигнала портов связи. Генераторы использует базовую частоту синхронизации XTI.

29.2.1 Регистры CFG2, CFG3, CFG5

Регистры доступны по чтению и записи, позволяют задавать различные параметры работы синтезаторов частоты CPU_PLL, BUS_PLL, LINK_PLL. Запись в

регистры разрешена только при нулевом значении бит CPPL_SEL, BPLL_SEL и LPLL_SEL, соответственно (регистр CFG4).

Таблица 256 – Регистр CFG2/ CFG3/ CFG5

Биты	Обозначение	По сбросу	Назначение
3:0	DIVR	0	Вход, задающий коэффициент деления делителя опорной частоты 0000 коэф. деления 1 0001 коэф. деления 2 1111 коэф. деления 16
10:4	DIVF	0	Вход, задающий коэффициент деления делителя обратной связи 0000000 коэф. деления 1 0000001 коэф. деления 3 0000010 коэф. деления 4 1111111 коэф. деления 129
13:11	DIVQ	0	Вход, задающий коэффициент деления выходного делителя 000 коэф. деления 1 001 коэф. деления 2 010 коэф. деления 4 011 коэф. деления 8 100 коэф. деления 16 101 коэф. деления 32
16:14	RANGE	0	Вход подстройки фильтра в зависимости от частоты ЧФД 001 = 10-16 МГц 010 = 16-25 МГц 011 = 25-40 МГц 100 = 40-65 МГц 101 = 65-100 МГц
20	bypass	0	Вход включения режима «bypass» при значении «1» на выход PLLOUT подается сигнал со входа REF.
22:21	S	0	Вход управления выходным мультиплексором, передающим сигнал на выход MUXOUT. 00 – подается сигнал с выхода PLL 01 – подается сигнал с выхода делителя опорной частоты 10 – подается сигнал с выхода делителя в обратной связи

Структурная схема PLL приведена ниже (Рисунок 117). Внешний вход частоты XTI подключен ко входу REF.

Значения коэффициентов подбираются таким образом, чтобы частота f_{VCO} находилась в диапазоне 800 – 1600 МГц, а частота f_{IN} была равна частоте f_{FB} и находилась в диапазоне, определяемом значением RANGE.

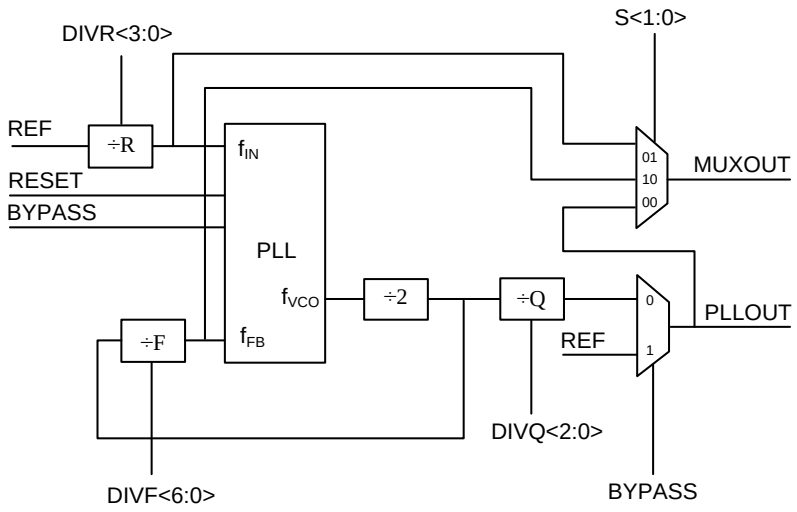


Рисунок 117 – Структура PLL

Процедура работы с CPU_PLL может быть следующей:

1 После сброса системы, синтезатор CPU_PLL выключен, и процессор работает на входной частоте XTI.
2 Посредством регистра CFG2 пользователь производит установку рабочих параметров PLL.
3 После задержки, необходимой для стабилизации частоты PLL, пользователь устанавливает соответствующий бит в регистре CFG4 и переключается на частоту PLL.
4 При необходимости изменить параметры рабочей частоты, пользователь выполняет переключение на частоту XTI. Далее выполняются шаги 2 и 3.

Аналогичные действия должны выполняться и при изменениях частоты внешней шины.

29.2.2 Регистр CFG4

Регистр позволяют задавать источник частоты различных модулей процессора.

Таблица 257 – Регистр CFG4

Биты	Обозначение	По сбросу	Назначение
0	CPLL_SEL	0	Задание частоты ядра процессора: 0 – XTI. Разрешение записи в регистр CFG2 1 – PLLOUT CPU_PLL. Запрещение записи в регистр CFG2
1	BPLL_SEL	0	Задание частоты интерфейса внешней шины: 0 – XTI. Разрешение записи в регистр CFG3 1 – BCLK (см. биты CFG4[5:4]). Запрещение записи в регистр CFG3
2	DIS_CC	0	1 – останов синхронизации ядра
3	DIS_BC	0	1 – останов синхронизации внешнего интерфейса
5:4	BCLK_SEL	00	Выбор источника синхронизации BCLK внешнего интерфейса 00 – частота PLLOUT BUS_PLL 01 – частота PLLOUT /2 CPU_PLL 10 – частота PLLOUT /4 CPU_PLL 11 – частота PLLOUT /8 CPU_PLL

Биты	Обозначение	По сбросу	Назначение
6	-	0	
7	LPLL_SEL	0	Задание частоты портов связи 0 – разрешение записи в регистр CFG5 1 – запрещение записи в регистр CFG5

Переключение процессора с одной частоты на другую требует выполнения определенных действий. Подробно процесс переключения излагается в подразделе «Переключение частоты процессора».

Также в процессоре предусмотрена возможность останова частот синхронизации с целью обеспечения реализации режима пониженного энергопотребления. Подробно процедура реализации данного механизма описана в подразделе «Реализация «спящего режима»».

29.3 Регистр состояния системы SYS_STS

Регистр доступен только по чтению. Описание битов регистра состояния SYS_STS приведено ниже (Таблица 258).

Таблица 258 – Регистр SYS_STS

Бит	Обозначение	Назначение
0	BOOT_0	Состояние входов BOOT[2:0] (они же PC[6:4]) в момент завершения сигнала внешнего сброса или внутреннего сброса по включению питания. Хранят вариант начального старта процессора.
1	BOOT_1	
2	BOOT_2	
3	-	
4	L0_BCMPO	Состояние входа PC[26] в момент завершения сигнала внешнего сброса или внутреннего сброса по включению питания.
5	L1_BCMPO	Состояние входа PC[30] в момент завершения сигнала внешнего сброса или внутреннего сброса по включению питания. Вход используется для задания режима работы контактной площадки внешнего осциллятора.
6	RST	Флаг устанавливается по сигналу внешнего сброса, сброса по включению питания или сброса от сторожевого таймера. Флаг может быть сброшен записью любого значения в регистр SYS_STS.
7	POR	Флаг устанавливается по сигналу внешнего сброса или сброса по включению питания. Флаг может быть сброшен записью любого значения в регистр SYS_STS.
31:8	-	Всегда 0

После сброса регистр состояния позволяет определить был ли сброс вызван внешним источником или включением питания (RST==1, POR==1) или сброс был вызван срабатыванием сторожевого таймера (RST==1, POR==0). В регистре состояние отражается вариант начального старта, который используется внутренним ПЗУ для определения алгоритма загрузки. Если бит L1_BCMPO равен 1, в качестве опорной частоты используется внешний вход XTI. Если значение бита L1_BCMPO равно 0, в качестве опорной частоты используется встроенный внутренний осциллятор.

29.4 Регистр управления синхронизацией периферийных устройств CFG8

Регистр может использоваться для разрешения или останова синхронизации различных периферийных устройств. Нулевое значение бита разрешает синхронизацию соответствующего периферийного устройства. Единичное значение останавливает синхронизацию.

Таблица 259 – Регистр CFG8

Биты	Обозначение	По сбросу	Разрешение синхронизации для модуля
0	I2S0_en	0	I2S 0 интерфейс
1	I2S1_en	0	I2S 1 интерфейс
2	VCAM_en	0	VCAM
3	SPI_en	0	SPI интерфейс
4	NAND_en	0	NAND контроллер
5	ARINC_en	0	ARINC
6	MIL0_en	0	МКПД 0
7	MIL1_en	0	МКПД 1
8	ADDA0_en	0	UP\DOWN 0
9	ADDA1_en	0	UP\DOWN 1
10	ADDA2_en	0	UP\DOWN 2
11	ADDA3_en	0	UP\DOWN 3
12	GPS0_en	0	Цифровой коррелятор
13	GPS1_en	0	Цифровой коррелятор
14	LCD_en	0	LCD контроллер
15	UART1_en	0	UART1
16	UART2_en	0	UART2
17	SPI1_en	0	SPI1 интерфейс
18	SPI2_en	0	SPI2 интерфейс
19	GTMR0_en	0	Таймер 0 с функцией Захвата/ШИМ
20	GTMR1_en	0	Таймер 1 с функцией Захвата/ШИМ
21	I2C_en	0	I2C интерфейс
31:22		0	Зарезервировано

Регистр может использоваться для задания режима пониженного энергопотребления системы. В случае если устройство в данный момент не используется, можно блокировать его частоту синхронизации. После блокировки, доступ к устройству невозможен. После сброса значение регистра равно нулю и на все периферийные устройства поступает сигнал синхронизации.

Примечание – Тактирование таймеров с функцией Захвата/ШИМ осуществляется, когда в регистре CFG8 установлено разрешение синхронизации сразу двух таймеров: GTMR0_en и GTMR1_en. См. описание ошибки 0007 в документе “1967BH044 Errata Notice”.

29.5 Переключение частоты процессора

В процессоре предусмотрена возможность гибкого выбора необходимой частоты функционирования. После сброса процессор всегда работает от входного источника синхронизации XTI. С помощью имеющихся модулей PLL можно задать требуемую для выполнения задачи частоту процессорного ядра и переключиться на заданную частоту.

Переключение частот выполняется с помощью регистра конфигурации CFG4. Биты регистра номер 7, 1 и 0 выполняют выбор источника синхронизации для портов связи, внешней шины и ядра соответственно. Переключение частот для портов связи и внешнего порта выполняется простой установкой или сбросом соответствующих бит регистра. Главное требование при смене данных частот это обеспечение отсутствия в момент переключения каких-либо действий в передатчиках порта связи или во внешнем интерфейсе.

Перечень необходимых подготовительных действий для смены, например, частоты внешнего интерфейса зависит от конфигурации внешней шины. Если, например, подключена внешняя динамическая память, нужно не забывать о периоде регенерации динамической памяти. Если смена частоты происходит после сброса процессора и динамическая память еще не инициализирована или нет необходимости обеспечивать сохранность информации в данной памяти, то переключение частоты можно выполнять без подготовительных работ. Если же процессор уже использовал динамическую память и в ней хранится информация, смена частот (например, с высокой на среднюю) будет требовать следующих действий:

- перевод динамической памяти в режим саморегенерации;
- переключение частоты внешней шины на входную частоту XTI;
- подготовка новой частоты в модуле BUS_PLL;
- корректировка параметров периода регенерации динамической памяти и других временных параметров SDRAM;
- переключение на частоту BUS_PLL с помощью бита CFG4[1];
- выход SDRAM из режима саморегенерации.

Возможны и другие действия в зависимости от подключенных внешних периферийных устройств и их требований в обслуживании.

Процесс переключения частоты ядра также требует выполнения целого комплекса работ в зависимости от ситуации в которой требуется переключение частот. Рассмотрим самый простой случай, когда процессор переключается на заданную частоту сразу после сброса. В этом случае должны быть выполнены следующие действия:

- с помощью регистра CFG2 задается требуемая конфигурация CPU_PLL для получения необходимой частоты. Выдерживается необходимое время для стабилизации выхода PLL.
- процессор переводится в режим останова синхронизации с помощью установки бита CFG4[2].
- сразу же за установкой бита CFG4[2] выполняется установка бита CFG4[0]. Процессор переключает частоту в момент, когда частоты ядра, периферийных устройств и внешней шины блокированы.
- через заданный интервал времени частота процессорного ядра разрешается и процессор продолжает выполнение программы с новой частотой.

При переключении частоты ядра используется функция останова синхронизации («спящий режим»). Подробно реализация данного режима и его особенностей приведена в подразделе «Реализация «спящего режима»».

Возможность переключения частоты процессора позволяет гибко выбирать необходимую производительность процессора в зависимости от решаемой задачи и таким образом сокращать потребляемую мощность системы.

29.6 Реализация «спящего режима»

В процессоре возможна реализация режима, в котором частоты ядра, периферийной шины и внешнего интерфейса могут быть остановлены. Данная функция введена для обеспечения режима наименьшего потребления в случае ожидания внешнего или внутреннего события. Ниже показан формирователь частот с учетом функции спящего режима (Рисунок 118). Специальные gate-модули G0, G1, G2 выполняют блокировку выбранных частот.

Установка бита CFG4[3] приведет к останову синхронизации на внешней контактной площадке SCLK и к отсутствию синхронизации внешнего интерфейса. Данная функция может быть использована, когда выводы внешнего интерфейса выполняют GPIO функции, и нет необходимости выполнять адресуемые обмены по внешней шине. Возобновление синхронизации внешнего интерфейса выполняется программно установкой данного бита в ноль.

Останов синхронизации ядра реализован немного сложнее. При установке бита CFG4[2] в единицу произойдет блокировка прохождения частоты на ядро, шину периферийных устройств и на внешний интерфейс (хотя на внешнем контакте SCLK частота будет присутствовать). После останова синхронизации ядра никакие программы уже невозможно выполнять и выход из данного режима можно выполнить только аппаратно. При этом источники формирующие сигнал пробуждения wake_up должны использовать другие источники синхронизации.

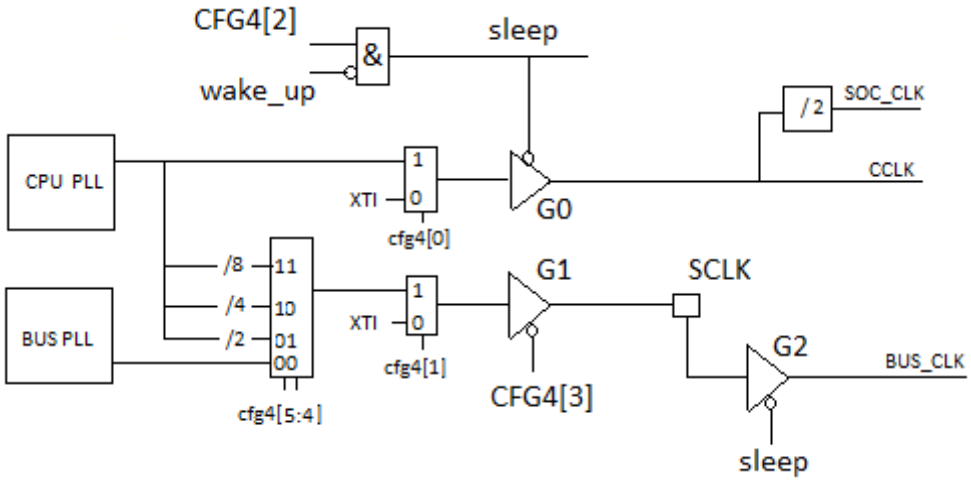


Рисунок 118 – Схема блокировки синхронизации

В качестве событий, которые могут вывести процессор из спящего режима могут использоваться:

- Запрос прерывания по любому из внешних входов портов общего назначения PA, PB, PC. Прерывания должны быть запрограммированы с выбором функции «по уровню».
- Запрос от события «тик-импульс» часов реального времени.
- Запрос от события «прерывание от сторожевого таймера» часов реального времени.
- Запрос от события «будильник» часов реального времени.

Модуль часов реального времени использует для своей работы вход внешней частоты XTI. В связи с этим при останове синхронизации ядра, модуль RTC продолжает работу и позволяет формировать требуемы сигналы.

Переходя в спящий режим, пользователь может запрограммировать выход из состояния сна с помощью необходимого ему события.

События wake_up, выводящие процессор из состояния сна можно охарактеризовать двумя параметрами: периодом следования и длительностью активного уровня. Если выбран внешний источник порта общего назначения, период следования и длина импульса неизвестны. Если выбран один из источников модуля RTC, параметры данных импульсов известны.

После сброса тик-импульсы формируются с периодом 256 тактов внешней частоты и имеют длительность периода внешней частоты. Период следования можно поменять при помощи значения регистра TIC_VAL.

Счетчик сторожевого таймера после сброса имеет значение 15, поэтому сторожевой таймер после сброса сформирует первый свой импульс wake_up через 15 периодов тик-импульса. Дальше период следования импульсов пробуждения от сторожевого таймера будет равен 256 периодам тик-импульса (если сторожевой таймер не обслуживается).

Будильник часов реального времени будет формировать свой wake_up запрос каждый раз, когда значение счетчика секунд RTC_CNT будет совпадать со значением регистра RTC_MR. Длительность импульса будет равна одной псевдосекунде.

Длительность импульса wake_up имеет значение, т.к. после пробуждения процессор не сможет опять вернуться в режим сна, если импульс пробуждения не завершился.

Запросы на выход из режима сна от портов общего назначения могут быть блокированы путем записи в регистр маскирования прерываний значения ноль. Запросы от источников модуля RTC могут быть блокированы с помощью разрядов 10, 9, 8 регистра управления RTC_CR.

Из рисунка видно, что процессор может выйти из режима сна, используя запрограммированную частоту, т.е. может возобновить свою работу на частоте XTI или частоте CPU_PLL. Дополнительное выключение модулей PLL для обеспечения пониженного энергопотребления также может быть использовано. Источник внешней синхронизации XTI не имеет функции выключения.

Каждый раз, переходя в режим сна, должен обеспечивать сохранность информации во внешней динамической памяти (если она используется), отсутствие работающих периферийных устройств (их синхронизация будет остановлена), а также должен позаботиться о способе выхода из режима сна.

30 Начальный старт процессора

Разряды порта PC[6:4] (они же BOOT[2:0]) во время сброса процессора задают режим начального старта процессора. Процессор может выполнить один из 8-ми вариантов начального старта. В любом из выбранных вариантов, процессор начинает работу с выполнения кода программы, размещенного во внутреннем загрузочном ПЗУ. Описание режимов загрузки приведено в Таблице 260.

Таблица 260 – Режимы загрузки

BOOT[2:0]	Режим загрузки
000	Старт загрузки по адресам, выбираемым одним из входов nIRQ. Срабатывание по уровню. 16-ти разрядная внешняя шина. nIRQ[0] -> 0x30000000 (MS0) nIRQ[1] -> 0x38000000 (MS1) nIRQ[3] -> 0x00000000 (internal memory)
001	Старт загрузки по адресам, выбираемым одним из входов nIRQ. Срабатывание по уровню. 32-ти разрядная внешняя шина. nIRQ[0] -> 0x30000000 (MS0) nIRQ[1] -> 0x38000000 (MS1) nIRQ[3] -> 0x00000000 (internal memory)
010	Старт из параллельной 8-ми разрядной FLASH-памяти (аналогично TS-300)
011	Старт из последовательной SPI FLASH-памяти (выборка по CS0)
100	Загрузка из NAND FLASH-памяти (выводы PD)
101	Загрузка через LINK-порты (любой) в 1-битовом режиме
110	Загрузка из NAND FLASH-памяти (альтернативные выводы PA)
111	Зарезервировано. Выполнение служебной программы

Процессор считывает код старта из регистра состояния и выполняет переход на одну из меток, соответствующую номеру начального старта. Далее будут подробно рассмотрены действия, которые выполняются при выборе того или иного варианта старта. Файл def1967VN044.h с параметрами, используемыми в программе начального старта, является частью поставки среды разработки программ CM-LYNX.

```
#include "def1967VN044.h"

.section program;
.global _boot_start;

_boot_start:
        /*
                Default clock configuration:
                        xti = input
                        core clk (cclk) = xti
                        SOC bus clock (sclk) = xti/2
                        External bus (memory) clock = xti
                        NAND operating frequency is configured to the slowest by
hardware after reset
                DSP starts in SV mode, GIE bit is set, all interrupts are disabled
(masks = 0)
                */
```

```
        // Enable multiple writes to SYSCON/SDRCON
        j9 = base_CMU;;
        j0 = 8;;
        cjmp = 0x00;;

        [j9 + 0] = j0;;  // CMU_CFG1_LOC

    // Define boot mode
        j1 = [j9 + 5];; // CMU_SYS_STS_LOC
        j1 = j1 and 7;;

        comp(j1, 7);;
        if jeq, jump start_XROM(np);;
        comp(j1, 6);;
        if jeq, jump start_xNAND(np);;
        comp(j1, 5);;
        if jeq, jump start_LINK1(np);;
        comp(j1, 4);;
        if jeq, jump start_NAND(np);;
        comp(j1, 3);;
        if jeq, jump start_SPIF(np);;
        comp(j1, 2);;
        if jeq, jump start_XPF(np);;
        comp(j1, 1);;
        if jeq, jump start_IRQ(np);          j10 = 0x00;;  // 32-bit mode
        j10 = SYSCON_MEM_WID64;;                                    //     16-bit
mode


        //-------------------------------------------------------//
        // Passive nIRQ boot
        //
        //-------------------------------------------------------//
    start_IRQ:
        // Initialize SYSCON register
        j0 = j10 or SYSCON_MS0_WT3          |
                    SYSCON_MS0_SLOW         |
                    SYSCON_MS0_IDLE         |

                    SYSCON_MS1_WT3          |
                    SYSCON_MS1_SLOW         |
                    SYSCON_MS1_IDLE;;
        syscon = j0;;

        // Setup GPIO
        // Alternate function for nBMS, nRD, MS1 and MS0 signals
        j0 = (PF_ALT << 20) | (PF_ALT << 21) | (PF_ALT << 18) | (PF_ALT << 19);;
        [j31 + GPC_ALT_LOC] = j0;;

        // Alternate function for DATA and ADDR bus
        j0 = PX_ALT_PDB0 | PX_ALT_PDB1;;                                //
data [15:0]
```

```
                comp(j10, 0x00);;
                if jeq; do, j0 = j0 or PX_ALT_PDB2 | PX_ALT_PDB3;;          //      data
[31:16]
                j0 = j0 or PX_ALT_PAB0 | PX_ALT_PAB1 | PX_ALT_PAB2;;        // addr
                [j31 + PX_ALT_LOC] = j0;;

                // Setup vectors
                j8 = 0x30000000;;
                IVIRQ0 = j8;;
                j8 = 0x38000000;;
                IVIRQ1 = j8;;
                j8 = 0x00000000;;
                IVIRQ3 = j8;;

                j8 = 0x0F;;
                intctl = j8;;     // nIRQ3:0 by level

                // Enable interrupts by external IRQs and vector interrupt (for host)
                j8 = INT_IRQ0 | INT_IRQ1 /*| INT_IRQ2*/ | INT_IRQ3 | INT_VIRPT;;
                IMASKH = j8;;

                // SQCTL[GIE] bit is set by reset, so we're done!
        stay_idle:
                nop;; nop;; nop;; nop;;
                idle;; nop;; nop;; nop;;
                jump stay_idle(np);;


                //-------------------------------------------------------//
                // Serial SPI FLASH at #CS0
                //
                //-------------------------------------------------------//
        start_SPIF:

                // Setup SPI
                j0 = (199 << 16) | SPCR0_DSS_8BIT;;                         //   8-bit   mode,
SPI clock = (SCLK_in / 2) / 200
                [j31 + SPCR0_LOC] = j0;;
                j0 = (1<<SPCR1_HOLDCS_P) | (1<<SPCR1_SPE_P);;       //   Enable   SPI,
hold #CS
                [j31 + SPCR1_LOC] = j0;;

                // Setup GPIO
                j0 = (PF_ALT << 4) | (PF_ALT << 5) | (PF_ALT << 6) | (PF_ALT << 7);;
                [j31 + GPA_ALT_LOC] = j0;;

                k0 = SPDR_LOC;;
                k1 = SPSR_LOC;;
                xr9 = 0x0808;;              // Initial FDEP arg
                xr10 = 0x0800;;             // FDEP arg modifier
                j10 = 0x00000000;; // target address
                j11 = 0x00;;
                lc0 = 256;;
```

```
        // Issue READ command
        xr0 = 0x03;;
        [k0 + 0] = xr0;                 xr1 = 0x00;;
        [k0 + 0] = xr1;;        // A[23:16]
        [k0 + 0] = xr1;;        // A[15:8]
        [k0 + 0] = xr1;;        // A[7:0]

loop_SPIF:
        // Write 4 bytes
        [k0 + 0] = xr1;;
        [k0 + 0] = xr1;;
        [k0 + 0] = xr1;;
        [k0 + 0] = xr1;;

        // Wait 4 bytes to be read
wait_SPIF:
        j0 = [k1 + 0];;
        j0 = j0 and (1<<SPSR_RFS_P);;
        if jeq, jump wait_SPIF;;

        // Read received data and compose 32-bit word
        xr8 = xr9;;
        xr4 = [k0 + 0];;
        xr0 = [k0 + 0];;
        xr4 += fdep r0 by r8;;
        xr0 = [k0 + 0];                 xr8 = r8 + r10;;
        xr4 += fdep r0 by r8;;
        xr0 = [k0 + 0];                 xr8 = r8 + r10;;
        xr4 += fdep r0 by r8;;

        // First read is dummy due to command and address shift
        [j10 += j11] = xr4;;
        if nlc0e, jump loop_SPIF;  j11 = 0x01;;

        // There are 4 more bytes in the FIFO - read them
        comp(j10, 0xFF);;
        if jeq, jump wait_SPIF(np);lc0 = 0x01;;

        // Done! Release #CS0
        j0 = /*(1<<SPCR1_HOLDCS_P) |*/ (1<<SPCR1_SPE_P);;
        [j31 + SPCR1_LOC] = j0;;

        // Jump to target address
        cjmp(abs)(np);;



   //-------------------------------------------------------//
        // Parallel 8-bit boot EEPROM
        //
        //-------------------------------------------------------//
start_XPF:
```

```
        // Setup GPIO
        // Alternate function for nBMS and nRD signals
        j0 = (PF_ALT << 20) | (PF_ALT << 21);;
        // Alternate function for MS1 and MS0 signals
        j0 = j0 or (PF_ALT << 18) | (PF_ALT << 19);;
        [j31 + GPC_ALT_LOC] = j0;;

        // Alternate function for DATA and ADDR bus
        j0 = PX_ALT_PDB0;;                                  // data (8-
bit)
        j0 = j0 or PX_ALT_PAB0 | PX_ALT_PAB1;;          // addr (16-bit)
        [j31 + PX_ALT_LOC] = j0;;

        // Setup DMA
        xr0 = 0x00;;
        xr1 = (256 << 16) | 4;;
        xr2 = 0x00;;
        xr3 = TCB_EPROM | TCB_NORMAL;;

        xr4 = 0x00;;
        xr5 = (256 << 16) | 1;;
        xr6 = 0x00;;
        xr7 = TCB_INTMEM | TCB_NORMAL;;

        DCS0    = xr3:0;;
        DCD0    = xr7:4;;

wait_XPF:
        j1:0 = DSTAT;;
        j0 = j0 and 0x7;;
        comp(j0, DSTAT_DONE);;
        if njeq, jump wait_XPF;;

        // Jump to target address
        cjmp(abs)(np);;

   //-------------------------------------------------------//
        // Link Ports slave boot
        //
        //-------------------------------------------------------//
start_LINK1:

        j0 = LRX_DSIZE_1BIT;;

start_LINK:
        j1 = j31 + j0;;
        j0 = j0 or LRX_EN | LRX_CLK_L0;;
        j1 = j1 or LRX_EN | LRX_CLK_L1;;

        // Setup GPIO
        // Alternate function for LINK0 and LINK1 control lines
        j4 = (PF_ALT<<24) | (PF_ALT<<25) | (PF_ALT<<26) | (PF_ALT<<27) | \
```

```
                (PF_ALT<<28) | (PF_ALT<<29) | (PF_ALT<<30) | (PF_ALT<<31);;
        [j31 + GPC_ALT_LOC] = j4;;

        // Enable LINK clock PLL bypass
        j4 = (1<<PLL_BYPASS_P);;
        [j31 + PLL_LINK_CFG_LOC] = j4;;

        // Setup DMA
        xr0 = 0x00;;
        xr1 = (256 << 16) | 0x04;;
        xr2 = 0x00;;
        xr3 = TCB_INTMEM | TCB_QUAD;;

        // Enable LINK port receivers
        LRCTL0 = j0;;
        LRCTL1 = j1;;

        // Start DMA
        DC8 = xr3:0;;
        DC9 = xr3:0;;

        // Wait for DMA (any of two channels)
wait_LINK:
        j1:0 = DSTAT;;
        j2 = j1 and (0x7 << 0);;
        j3 = j1 and (0x7 << 3);;
        comp(j2, DSTAT_DONE << 0);;
        if jeq, jump wait_LINK_done(np);;
        comp(j3, DSTAT_DONE << 3);;
        if jeq, jump wait_LINK_done(np);;
        jump wait_LINK;;

wait_LINK_done:
        // Jump to target address
        cjmp(abs)(np);;


start_XROM:  cjmp = 0x80008000;;
         cjmp(abs)(np);;

   //-------------------------------------------------------//
        // NAND FLASH boot
        //
        //-------------------------------------------------------//

start_xNAND:  j1 = 0x60008;; // bit 18-17
                        xr0 = 0x6FF7C;;
                    [j9+0] = j1;;
          [j31+GPA_ALT_LOC] = xr0;;
                        jump  boot_xNAND (NP);;

start_NAND:
```

```
        // Setup GPIO
        // Alternate function for DATA and ADDR bus
        j0 = PX_ALT_PDB2 | PX_ALT_PDB3;;  // PXD[31:16] = alternate functions
        j0 = j0 or PX_ALT_PDB23F0;;                  // PXD[31:16] = NAND FLASH
        [j31 + PX_ALT_LOC] = j0;;

        // NAND controller is active after reset. It's settings are:
        //      - page size = 2K
        //      - 3 bytes to select a page (row)
        //      - 2 bytes to select a column
        //      - 5 address bytes and 2 read commands

boot_xNAND:

        // Setup number of 32-bit words to read
        lc0 = 256;;
        [j31 + NAND_CNTR_LOC] = lc0;;
        j4 = 0x00;;             // start address

read_NAND:
        j0 = [j31 + NAND_SR_LOC];;
        j0 = j0 and NAND_SR_EMPTY;;
        if njeq, jump read_NAND;;
        xr0 = [j31 + NAND_DR_LOC];;
        [j4 += 0x01] = xr0;;
        if nlc0e, jump read_NAND;;

        // Jump to target address

        cjmp(abs)(np);;
        nop;;nop;;nop;;nop;;
        nop;;nop;;nop;;nop;;


_boot_start.end:
```

30.1 Загрузка из внешней NAND флэш-памяти

Код программы выполняет загрузку 256 слов данных из внешней NAND флэш-памяти во внутреннюю память, начиная с адреса 0. После завершения загрузки управление передается на адрес 0.

30.2 Загрузка EPROM

Выбор данного варианта соответствует загрузке процессора из внешней параллельной памяти типа EPROM. При выборе режима загрузки EPROM процессор инициализирует канал 0 DMA для передачи 256-ти 32-разрядных слов кода из загрузочного EPROM в ячейки 0x00-0xFF блока 0 внутренней памяти процессора. После инициализации DMA канала процессор переходит в состояния проверки завершения передачи блока данных из внешней памяти во внутреннюю. Как только передача завершится, процессор передает управление по адресу 0.

30.3 Загрузка с использованием порта связи

Процессор программирует два канала DMA для приема данных из портов связи. Каналы DMA портов связи инициализируются для передачи блока 256 слов в адреса внутренней памяти с 0 по 255. После программирования передачи, процессор выполняет ожидание завершения пересылки и затем передает управление по адресу 0.

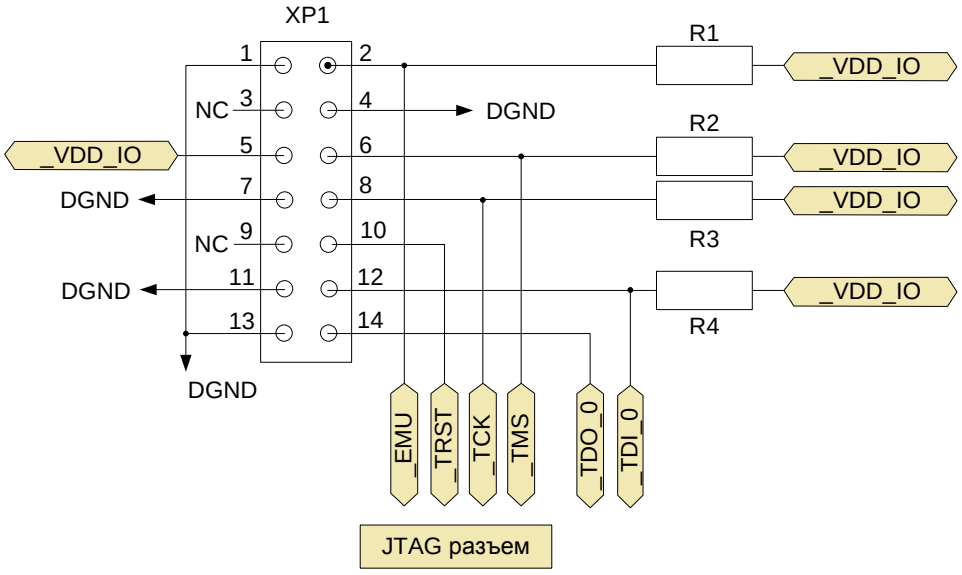
30.4 Старт по запросу прерывания

В данном варианте старта процессор определяет ширину внешней шины в 32 бита (вариант 1) или 16 бит (вариант 0), конфигурирует контроллер прерываний для обслуживания запросов прерываний по входам IRQ0, IRQ1, IRQ3, предварительно определив для них значения векторов прерываний 0x3000_0000, 0x3800_0000 и 0. Также разрешается векторное прерывание. Определив конфигурацию, процессор переходит в режим ожидания прерывания. При возникновении одного из определенных ранее прерываний, процессор перейдет на его обработку. В данном режиме старта внешний хост через последовательный интерфейс может загрузить код программы во внутреннюю память и, воспользовавшись векторным прерыванием, запустить процессор на выполнение загруженной программы.

30.5 Выбор источника синхросигнала

Вывод общего назначения PC[30] во время сброса определяет способ подачи внешнего синхросигнала на процессор. Если во время сброса на этом выводе удерживается низкий уровень, то к выводам XTI, XTO должна быть подключена схема запуска внутреннего генератора с кварцевым резонатором. В противном случае – высокий уровень на PC[30] или неподключенный вывод PC[30] (за счет внутреннего резистора подтяжки к питанию) требуют подключения генератора синхросигнала на вывод XTI.

30.6 Схема подключения отладчика JEM-LYNX



R1 – R4 – резисторы сопротивлением 47 кОм ± 5 %

Рисунок 119 – Схема подключения отладчика JEM-LYNX

Назначение выводов разъема на рисунке 119 приведено в таблице 261.

Таблица 261 – Описание выводов разъема

Номер вывода	Обозначение	Назначение
1	GND	Общий
2	/EMU	Сигнал готовности процессора к приему отладочной инструкции
3	NC	Не используется
4	GND	Общий
5	VDDIO	Проверка наличия питания и согласование уровней сигналов
6	TMS	Изменение состояния TAP-контроллера
7	GND	Общий
8	TCK	Тактовая частота TAP-контроллера
9	/RST	Синхронный аппаратный сброс процессоров
10	/TRST	Сброс TAP-контроллера
11	JG_MX	Управление JG_MX
12	TDI	Запись команд и данных в TAP-контроллер
13	GND	Общий
14	TDO	Чтение данных из TAP-контроллера

31 Часы реального времени RTC

Модуль RTC представляет собой набор счетчиков, которые используют для своей работы входную частоту процессора XTI. В связи с отсутствием частоты 32,768 КГц обычно используемой для часов реального времени, в данном модуле используется предварительный делитель входной частоты XTI с помощью которого можно задать опорную частоту, выполняющую такие же функции, что и частота 32,768 КГц.

Модуль RTC включает набор регистров, позволяющих управлять работой модуля, а также отслеживать его состояние. Регистры подключены к шине обмена периферийных устройств и имеют базовый адрес **0x800001E0**. Краткое описание регистров приведено ниже (Таблица 262).

Таблица 262 – Регистры модуля RTC

Номер	Название	Бит	Сброс	Регистр
0	RTC_CNT	32	0	Счетчик секунд
1	RTC_MR	32	max	Регистр сравнения
2	WDT_CNT	8	15	Текущее значение сторожевого счетчика
3	RTC_TDIV	20	0	Текущее значение делителя тик-импульса
4	RTC_CR	8	0	Регистр управления
5	RTC_SDIV	16	0	Текущее значение делителя секунды
6	TIC_VAL	20	255	Период тик-импульса минус 1
7	SEC_VAL	16	3	Количество тик-импульсов в секунде
8	RTC_BUSY	1	0	Флаг занятости интерфейса

Счетчики модуля RTC также могут быть использованы как дополнительные таймерные модуля, используемые для генерации различных временных интервалов.

31.1 Формирователь «тик-импульсов»

Входная частота XTI поступает на счетчик RTC_TDIV, который по каждому импульсу входного синхросигнала увеличивает свое значение на 1. Значение счетчика RTC_TDIV непрерывно сравнивается со значением регистра TIC_VAL. Если значение счетчика RTC_TDIV становится равным значению регистра TIC_VAL, то с приходом следующего импульса XTI значение счетчика RTC_TDIV становится равным нулю и к контроллеру прерывания посылается запрос прерывания – «тик-импульс». Регистр TIC_VAL задает период следования тик-импульсов и содержит значение реального периода уменьшенное на 1. Тик-импульсы могут быть использованы операционной системой в качестве базового периода для работы программ в системе разделения времени. Период формирования тик-импульсов импульсов равен

T_тик = XTI / (TIC_VAL+1).

Полученные тик-импульсы используются модулем RTC для генерации импульсов секунд (точнее – псевдосекунд) и для работы сторожевого таймера.

31.2 Формирователь импульсов секунд

Модуль RTC содержит счетчик RTC_SDIV, используемый для формирования импульсов секунд. Счетчик тактируется с помощью синхросигнала XTI, однако увеличивает значение RTC_SDIV на 1, только в момент прихода тик-импульса. Значение регистра RTC_SDIV постоянно сравнивается со значением регистра SEC_VAL, который задает период работы счетчика RTC_SDIV в терминах тик-импульсов. Если значение счетчика RTC_SDIV равно значению регистра SEC_VAL, то с приходом следующего тик-импульса будет сформирован импульс секунды. Значение счетчика RTC_SDIV станет равным нулю, и он начнет повторный счет. Таким образом генератор секунд формирует импульсы секунд согласно формуле

T_сек = T_тик/(SEC_VAL+1).

Секундные импульсы поступают на счетчик секунд RTC_CNT.

31.3 Счетчик секунд RTC_CNT

В данном счетчике содержится время в секундах прошедшее с момента включения часов. Счетчик может быть скорректирован. Счетчик тактируется синхросигналом XTI, однако увеличивает свое значение на 1 только в момент прихода импульса секунд. Регистр RTC_CNT имеет 32 разряда. Его значение непрерывно сравнивается со значением регистра RTC_MR. В случае если значение счетчика RTC_CNT равно значению регистра RTC_MR, то в момент прихода следующего секундного импульса будет сформирован запрос прерывания ALARM к процессору. Регистр RTC_MR позволяет реализовать функции будильника и должен содержать значение времени срабатывания в секундах (минус 1 от действительного значения).

31.3.1 Регистр сравнения RTC_MR

Регистр сравнения используется для реализации функций будильника. Его значение непрерывно сравнивается со значением регистра счетчика секунд. В момент равенства значений устанавливается соответствующий флаг и вырабатывается сигнал прерывания.

31.3.2 Регистр управления RTC_CR

Регистр управляет включением или выключение различных функций часов реального времени. Биты регистра кратко описаны ниже (Таблица 263). По сигналу внешнего сброса все биты регистра устанавливаются в ноль.

Таблица 263 – Регистр управления

Бит	Имя	Тип	Назначение
1:0			
2	WDT_EN	R/W	Разрешение работы сторожевого таймера 0 – выключен 1 – включен
3	FREEZ	R/W	Блокировка сторожевого таймера. После установки этого бита невозможно выключить сторожевой таймер.

Бит	Имя	Тип	Назначение
6:4	WDT_SEL	R/W	Выбор значения загружаемого в счетчик сторожевого таймера. 0 – 1, 1 – 2, 2 – 4, 3 – 8, 4 – 16, 5 – 32, 6 – 64, 7 – 128
7	LOCK	R/W	Управление доступом к модулю RTC 0 – чтение-запись 1 – только чтение (для RTC_CR – чтение-запись)
8	DIS_ATIC	R/W	Когда 1 запрещает генерацию асинхронного прерывания от импульса «тик»
9	DIS_ADOG	R/W	Когда 1 запрещает генерацию асинхронного прерывания от сторожевого таймера
10	DIS_AMR	R/W	Когда 1 запрещает генерацию асинхронного прерывания от регистра сравнения

Функции регистра управления заключаются в задании параметров работы модуля RTC. Биты регистра управления позволяют включить сторожевой таймер (WDT_EN), при необходимости заблокировать возможность его случайного выключения(FREEZ). С помощью разрядов WDT_SEL можно задать значение константы, которое будет загружаться в сторожевой таймер при его инициализации. Также возможно задание блокировки доступа к регистрам RTC для защиты от случайной записи (бит LOCK). В этом случае ко всем регистрам RTC (кроме регистра управления) возможен доступ только по чтению. В процессоре реализован режим пониженного энергопотребления (спящий режим), в котором все системные частоты могут быть остановлены. Активной может быть только частота XTI. В этом случае для выхода из спящего режима могут быть использованы асинхронные запросы прерываний от тик-импульса, будильника или сторожевого таймера. Соответствующие биты регистра управления разрешают эти функции.

31.3.3 Регистр счетчика сторожевого таймера WDT_CNT

Этот 8-разрядный регистр осуществляет вычитание 1 из своего содержимого при поступлении тик-импульса. При достижении значения WDT_CNT==1 и в момент прихода следующего тик-импульса, сторожевой таймер вырабатывает запрос на прерывание. При этом значение WDT_CNT становится равным нулю. Запрос прерывания от сторожевого таймера должен рассматриваться как запрос на повторную инициализацию сторожевого таймера. Если значение WDT_CNT равно нулю и приходит очередной тик-импульс, сторожевой таймер сформирует сигнал сброса системы длительностью в один период тик-импульса. Косвенным образом можно записать новое значение в счетчик сторожевого таймера. Для этого вначале нужно записать соответствующее значение в биты WDT_SEL регистра управления, а затем записать константу 0x04072013 по адресу счетчика сторожевого таймера. В регистре WDT_CNT автоматически установится требуемое значение.

31.3.4 Регистр занятости интерфейса часов RTC_BUSY

Это 1-битный регистр, доступный только по чтению. Проблема в работе с модулем RTC заключается в том, что все его внутренние регистры тактируются частотой XTI. В тоже время процессор обращается к модулю через периферийную шину, частота которой может значительно превосходить частоту XTI. В связи с этим в модуле реализован согласующий буфер между двумя доменами синхронизации. Любая запись в часы приводит к записи в буфер. Значение из буфера в дальнейшем передается в другой домен и там происходит запись. Пока вся процедура не выполнена, флаг занятости равен 1 и новая запись невозможна. А именно новая запись приведет к ошибке. Чтение регистров реализовано следующим образом. При

чтении регистра модуля часов происходит копирование значения регистра в буфер периферийной шины. В этом же такте старое значение буфера передается в процессор. Таким образом, чтобы прочитать текущее значение какого-то регистра необходимо выполнить два последовательных чтения.

Отметим, что при чтении регистра ***RTC_BUSY*** считывается непосредственное значение флага, и нет необходимости выполнять чтение повторно.

31.4 Ограничения модуля RTC

Модуль использует для своей работы входную тактовую частоту XTI. При генерации запросов прерывания от тик-импульсов, будильника, сторожевого таймера, импульсы запросов синхронизируются частотой SOC-шины. Для надежной генерации прерывания необходимо чтобы частота SOC-шины была выше внешней входной частоты XTI.

31.5 Состояние модуля RTC после сброса

Процессор может быть сброшен в исходное состояние следующими сигналами:

- внутренний сигнал сброса от модуля контроля включения питания процессора nPOR;
- активным уровнем по внешнему входу nRST_in;
- сигналом от сторожевого таймера WDT_RES.

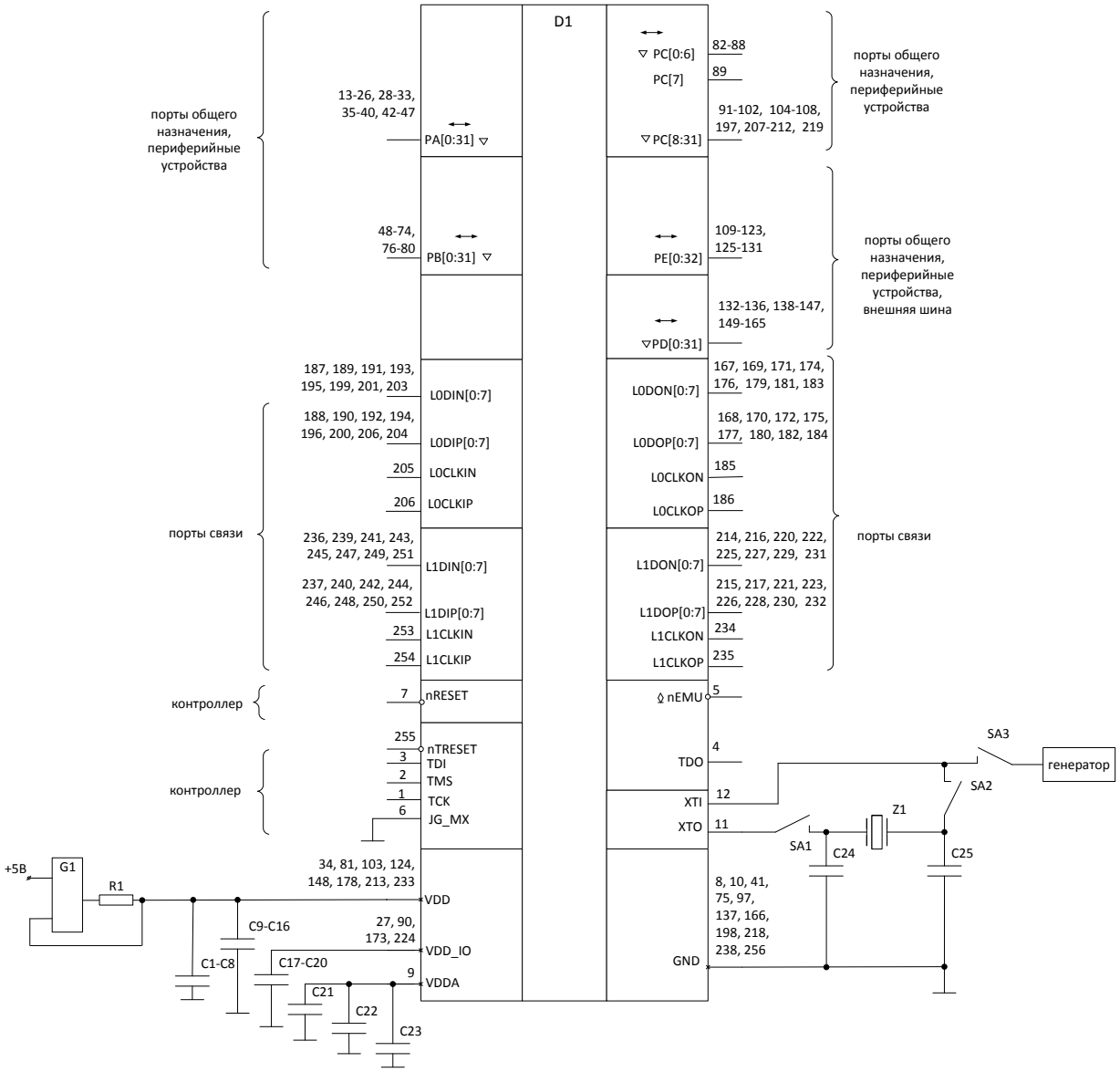
Часть ресурсов модуля RTC сбрасывается сигналом nPOR только при включении питания процессора. Этими ресурсами являются регистры: RTC_CNT, RTC_MR, RTC_TDIV, RTC_SDIV, TIC_VAL, SEC_VAL. Все другие регистры и флаги модуля сбрасываются любым из трех вышеназванных источников. Таким образом, если пользователь определил конфигурацию часов реального времени, значение часов будет корректным до тех пор, пока питание процессора в норме. Никакие внешние события или внутренний сброс от сторожевого таймера не изменяют значение часов. В модуле CMU имеется флаг, который позволяет различить сброс сторожевого таймера от сброса из других источников. Однако нет возможности различить сброс по включению питания от внешнего сброса. Между тем факт внешнего сброса может быть вычислен косвенным образом. Например, если после сброса значение счетчика секунд не равно нулю, значит произошел сброс от внешнего сигнала и программа не должна выполнять процедуру настройки часов повторно.

Регистр управления модулем RTC_CR сбрасывается в любом из трех случаев.

32 Инструкция по программированию

Раздел находится в разработке.

33 Типовая схема включения



D1 – включаемая микросхема;
G1 – генератор питания;
C1 ÷ C8, C17 ÷ C21 – конденсаторы емкостью 100 нФ;
C9 ÷ C16, C22 – конденсаторы емкостью 10 нФ;
C23 – конденсатор емкостью 100 пФ;
C24, C25 – конденсатор емкостью 15 пФ;
R1 – резистор сопротивлением не менее 2,1 Ом;
SA1 ÷ SA3 – переключатели;
Z1 – кварцевый резонатор.

Рисунок 120 – Типовая схема включения микросхем

34 Схема формирования внутреннего сброса по включению питания

В микросхеме реализована схема формирования внутренних сигналов сброса по включению питания как по домену U_{CCIO} – сигнал por_b_33, так и по домену U_{CC} – сигнал por_b_12. Низкий уровень сигнала por_b_33 удерживает в выключенном состоянии приемопередатчики портов связи, высокий уровень разрешает их работу. Низкий уровень сигнала por_b_12 приводит все триггеры микросхемы в начальное состояние и удерживает схему в сбросе. Высокий уровень разрешает работу микросхемы. Алгоритм формирования этих сигналов следующий:

1 В начальный момент значение сигналов por_b_12, por_b_33 равно нулю;
2 При достижении напряжений питания U_{CCIO} и U_{CC} значений пороговых уровней U_{CCIO_th} и U_{CC_th} соответственно начинает работать внутренний счетчик;
3 Через промежуток времени T_{ON} внутренним счетчиком устанавливаются сигналы por_b_12 и por_b_33 в состояния U_{CC} и U_{CCIO} соответственно.

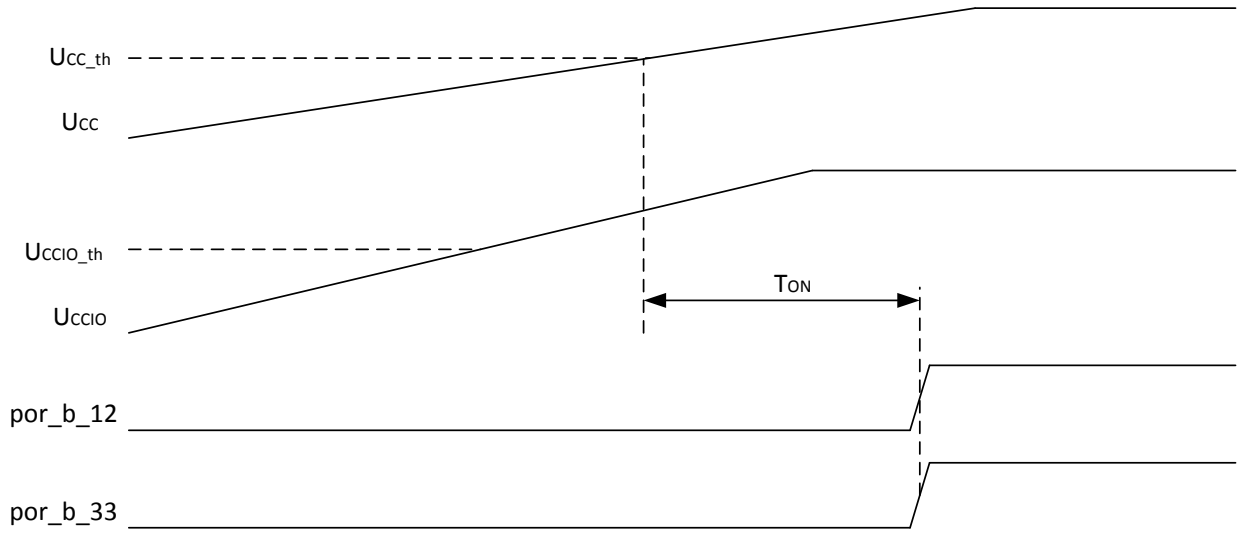


Рисунок 121 – Формирование внутренних сигналов сброса по включению питания

Таблица 264 – Параметры формирования внутренних сигналов сброса

Буквенное обозначение параметра, единица измерения	Норма параметра		Наименование параметра
	не менее	не более	
U_{CCIO_th}, В	2,75	2,81	Пороговое значение питания U_{CCIO}
U_{CC_th}, В	0,98	1,0	Пороговое значение питания U_{CC}
T_{ON}, мс	12	28	Задержка включения схемы

35 Типовые зависимости

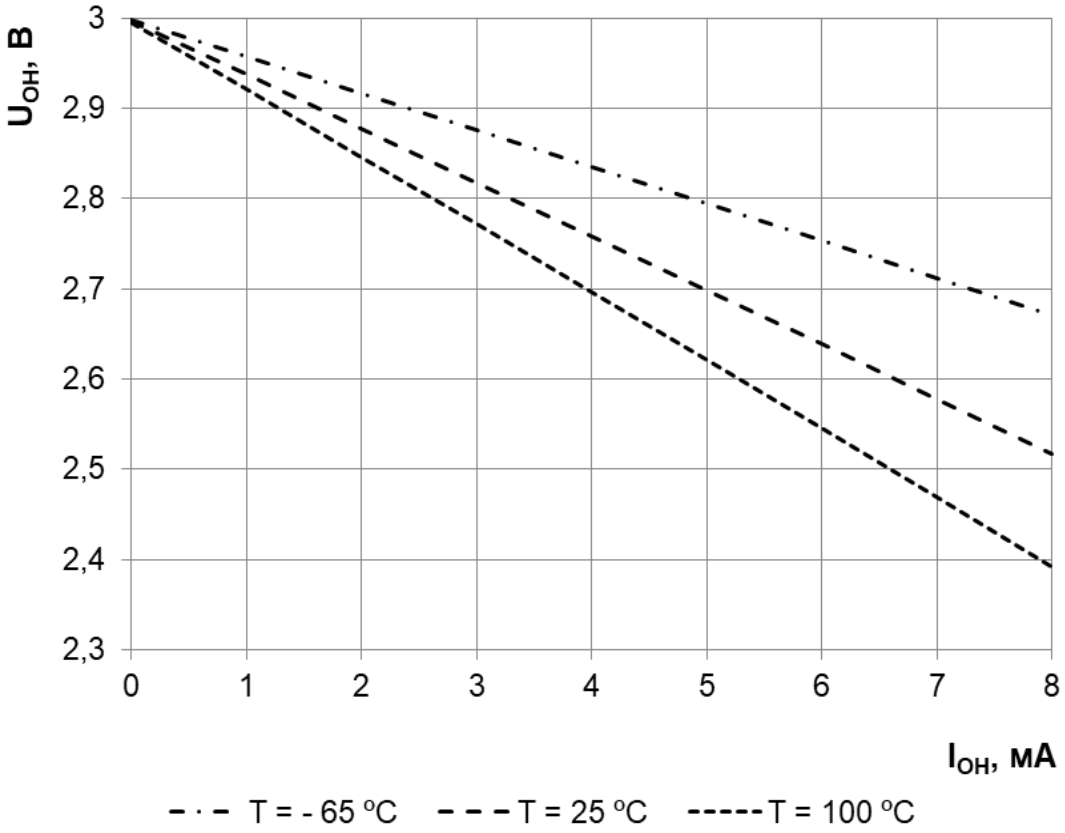


Рисунок 122 – Зависимость выходного напряжения высокого уровня U_{OH} от выходного тока высокого уровня при U_{CC} = 1,08 В, U_{CCIO} = 3,0 В

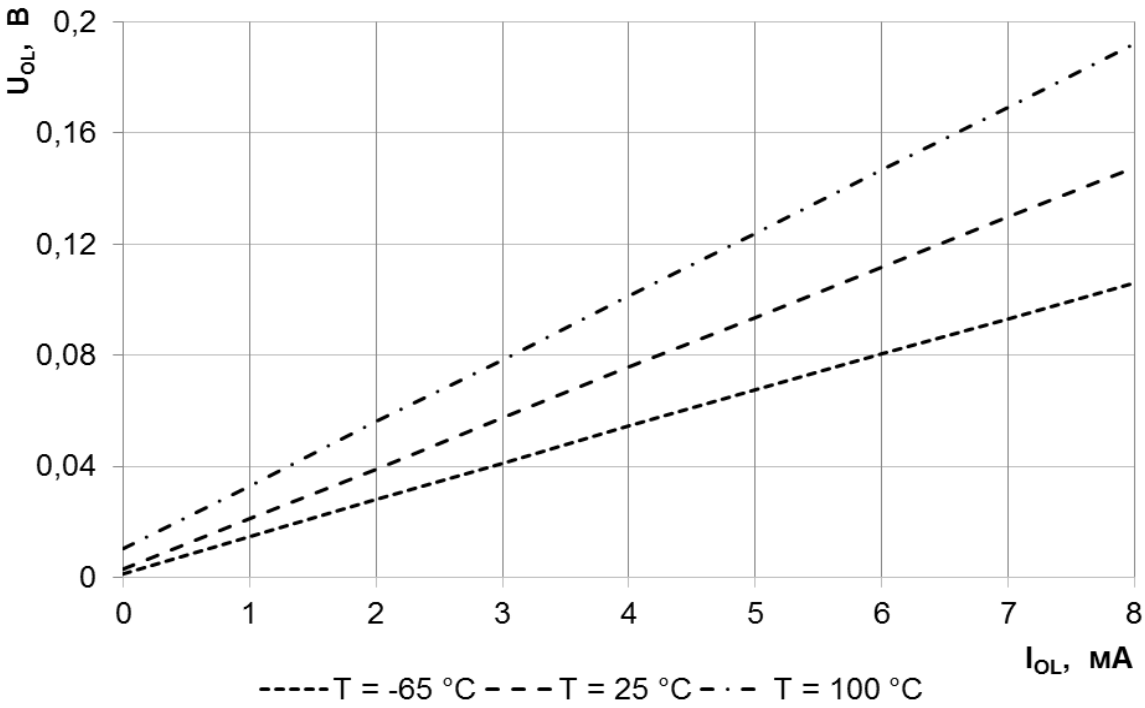


Рисунок 123 – Зависимость выходного напряжения низкого уровня U_{OL} от выходного тока низкого уровня при U_{CC} = 1,08 В, U_{CCIO} = 3,0 В

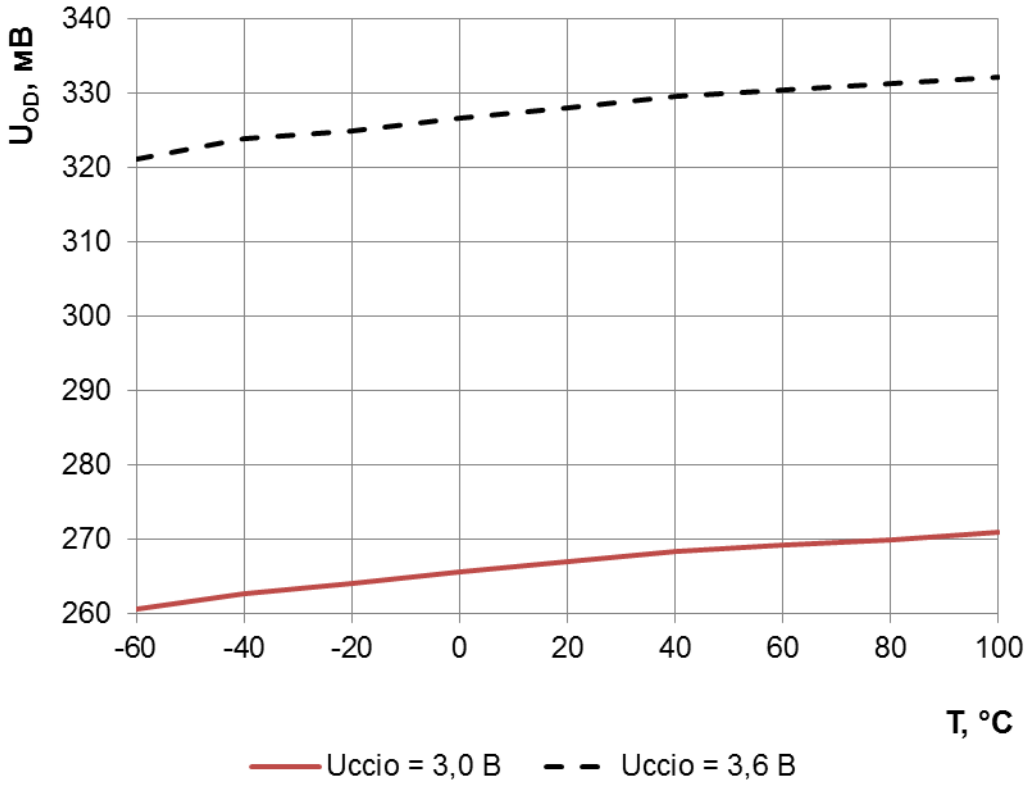


Рисунок 124 – Зависимость дифференциального выходного напряжения LINK-портов U_{OD} от температуры при R_L=100 Ом

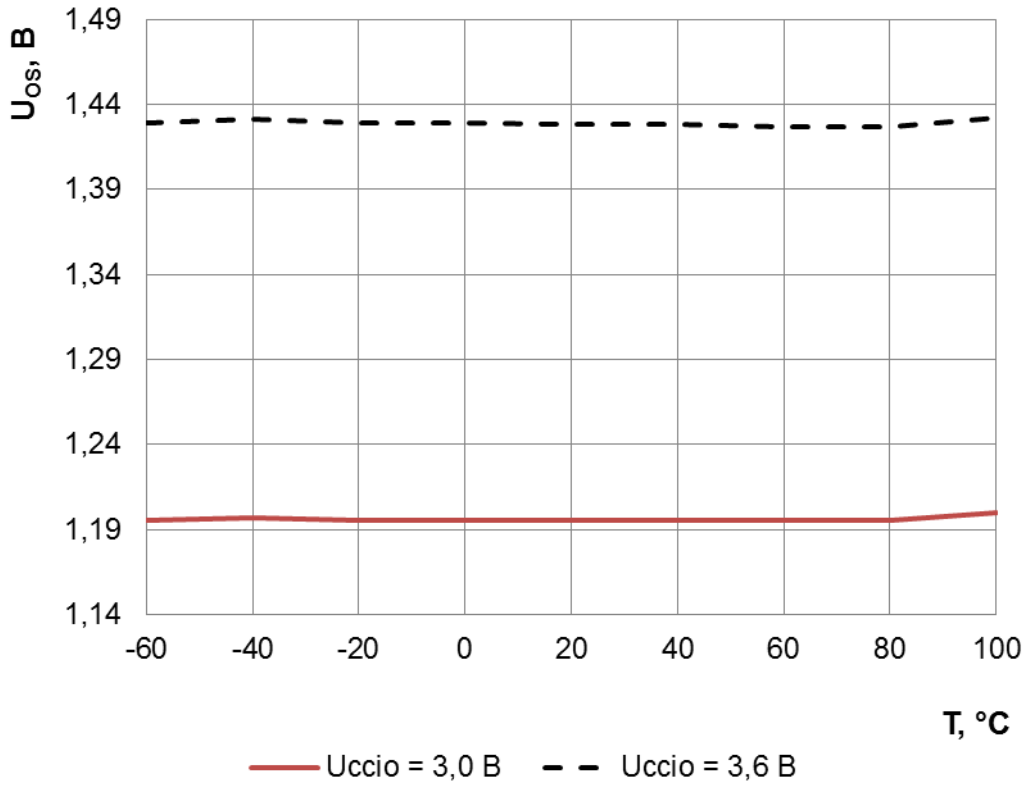


Рисунок 125 – Зависимость синфазного выходного напряжения LINK портов U_{OS} от температуры при R_L=100 Ом

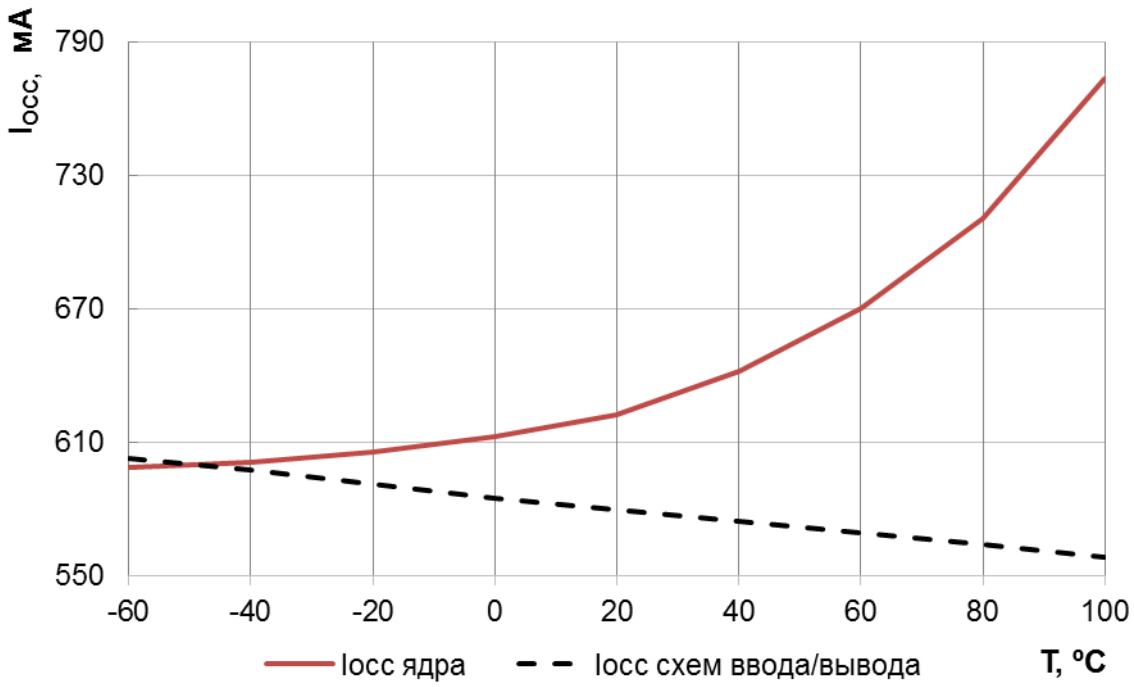


Рисунок 126 – Зависимость динамического тока потребления I_{OCC} ядра и схем ввода/вывода от температуры при U_{CC} = 1,32 В, U_{CCIO} = U_{CCA} =3,6 В, f_C = 230 МГц

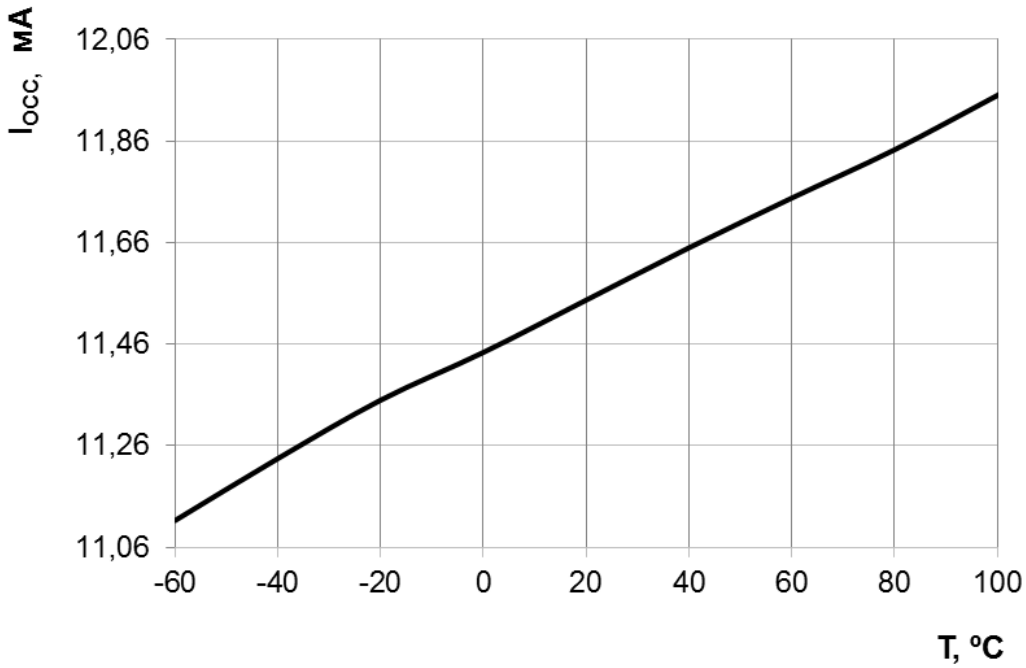


Рисунок 127 – Зависимость динамического тока потребления I_{OCC} аналоговых блоков от температуры при U_{CC} = 1,32 В, U_{CCIO} = U_{CCA} =3,6 В

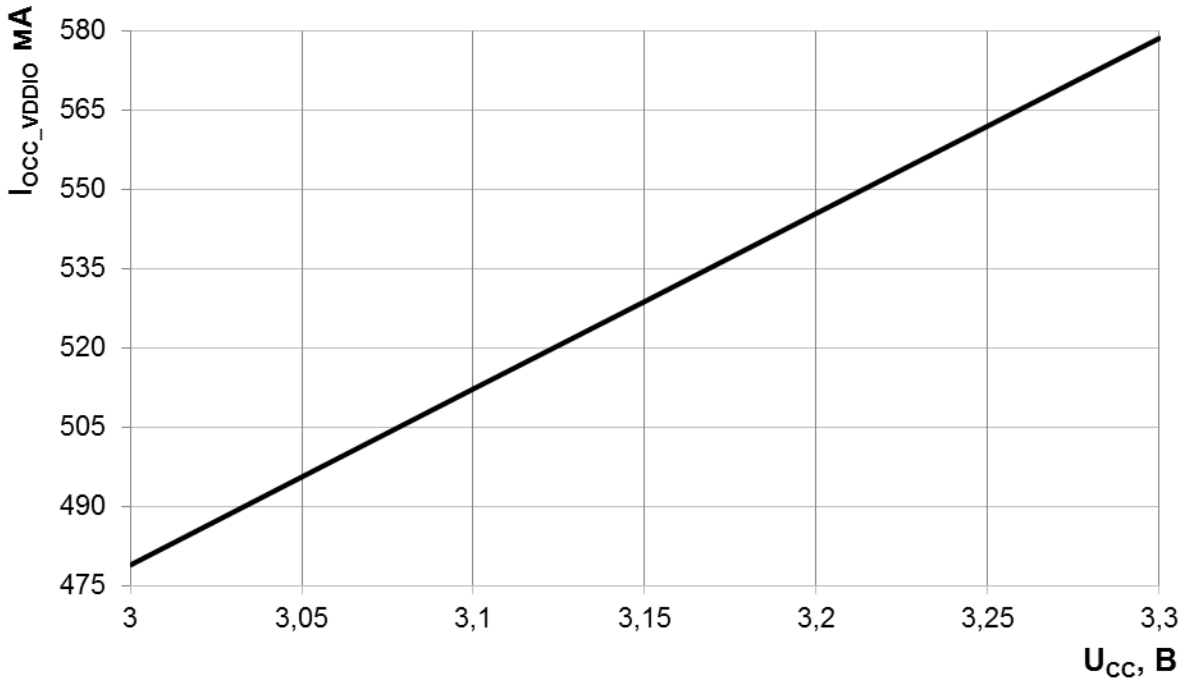


Рисунок 128 – Зависимость динамического тока потребления I_{OCC} схем ввода/вывода от напряжения питания при f_C = 230 МГц, T = 25 ºC

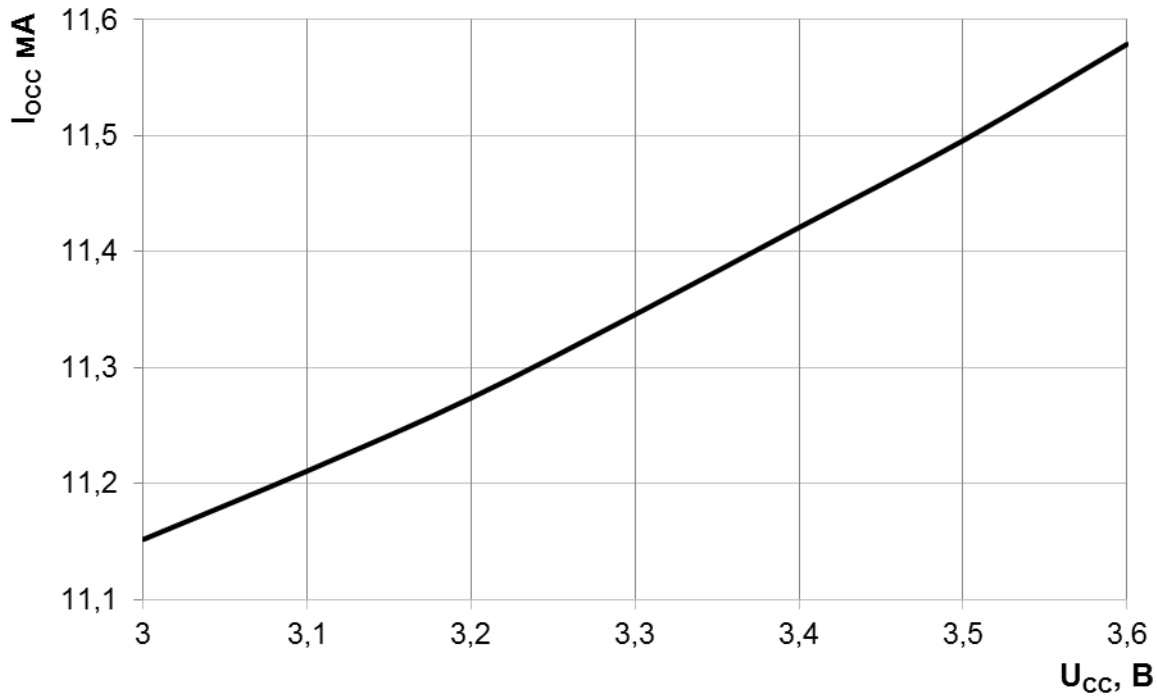


Рисунок 129 – Зависимость динамического тока потребления I_{OCC} аналоговых блоков от напряжения питания при f_C = 230 МГц, T = 25 ºC

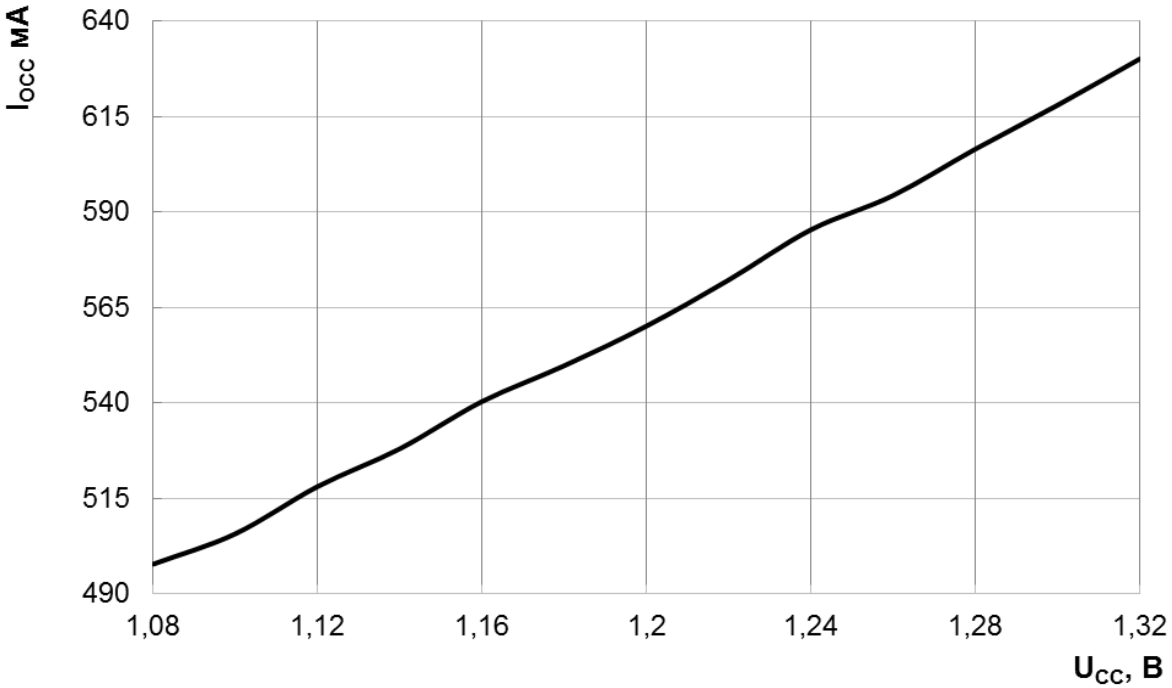


Рисунок 130 – Зависимость динамического тока потребления I_{OCC} ядра от напряжения питания при f_C = 230 МГц, T = 25 ºC

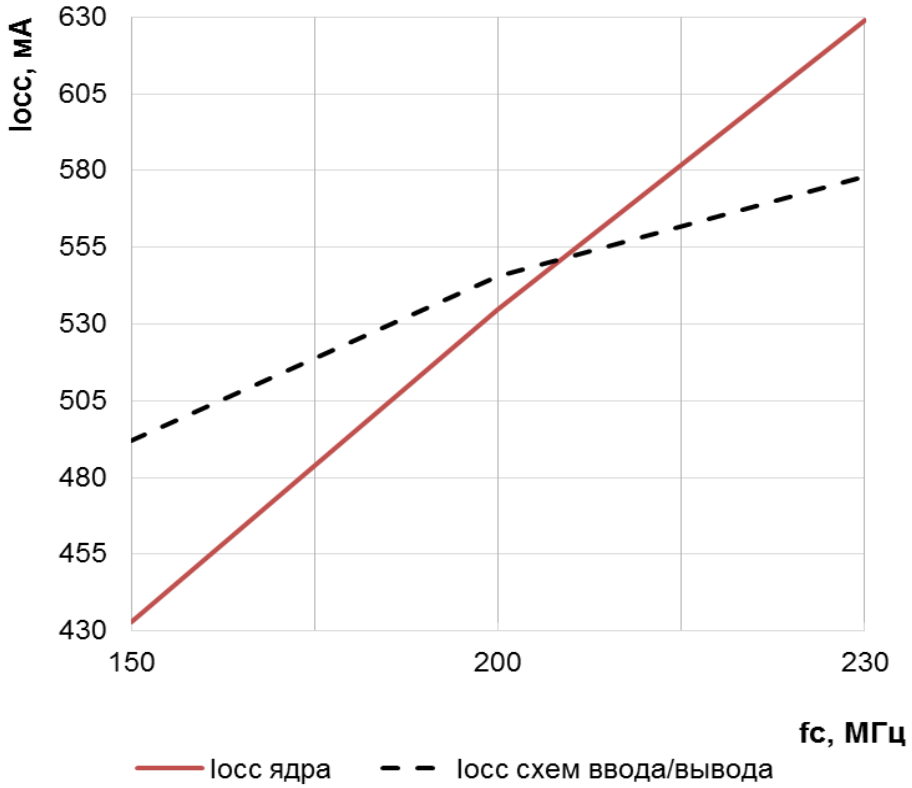


Рисунок 131 – Зависимость динамического тока потребления I_{OCC} ядра и схем ввода/вывода от тактовой частоты процессора f_C при U_{CC} = 1,32 В, U_{CCIO} = U_{CCA} = 3,6 В, T = 25 ºC

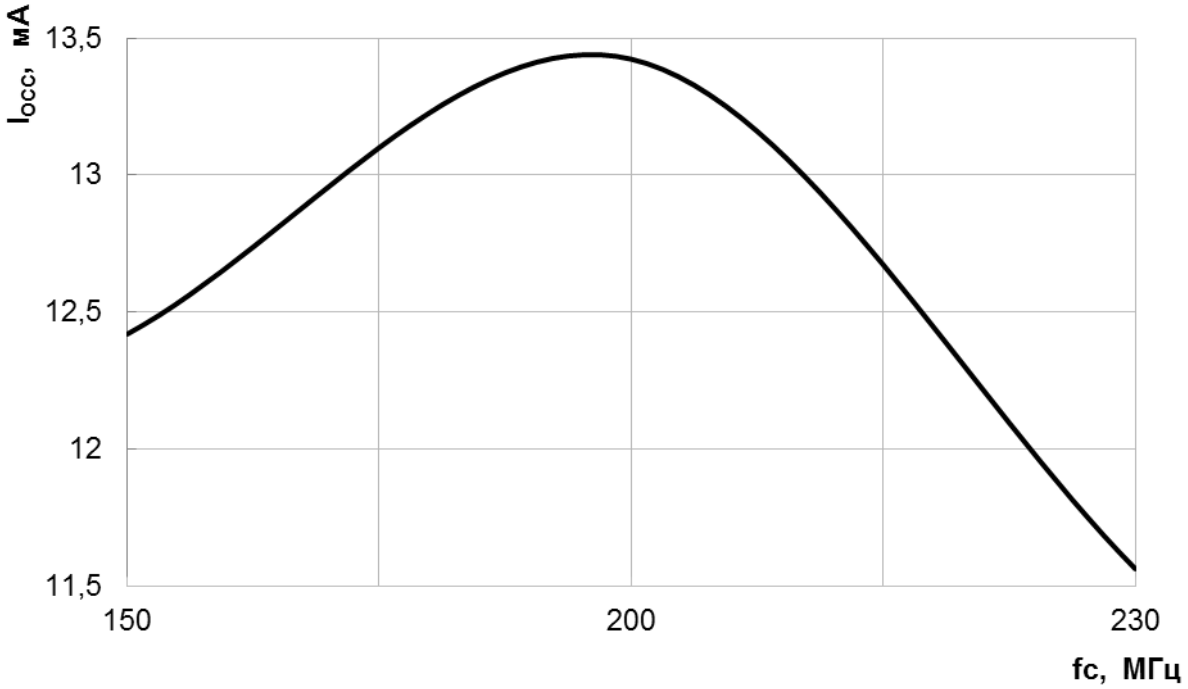


Рисунок 132 – Зависимость динамического тока потребления аналоговых блоков I_{OCC} от тактовой частоты процессора f_C при U_{CC} = 1,32 В, U_{CCIO} = U_{CCA} = 3,6 В, T = 25 ºС

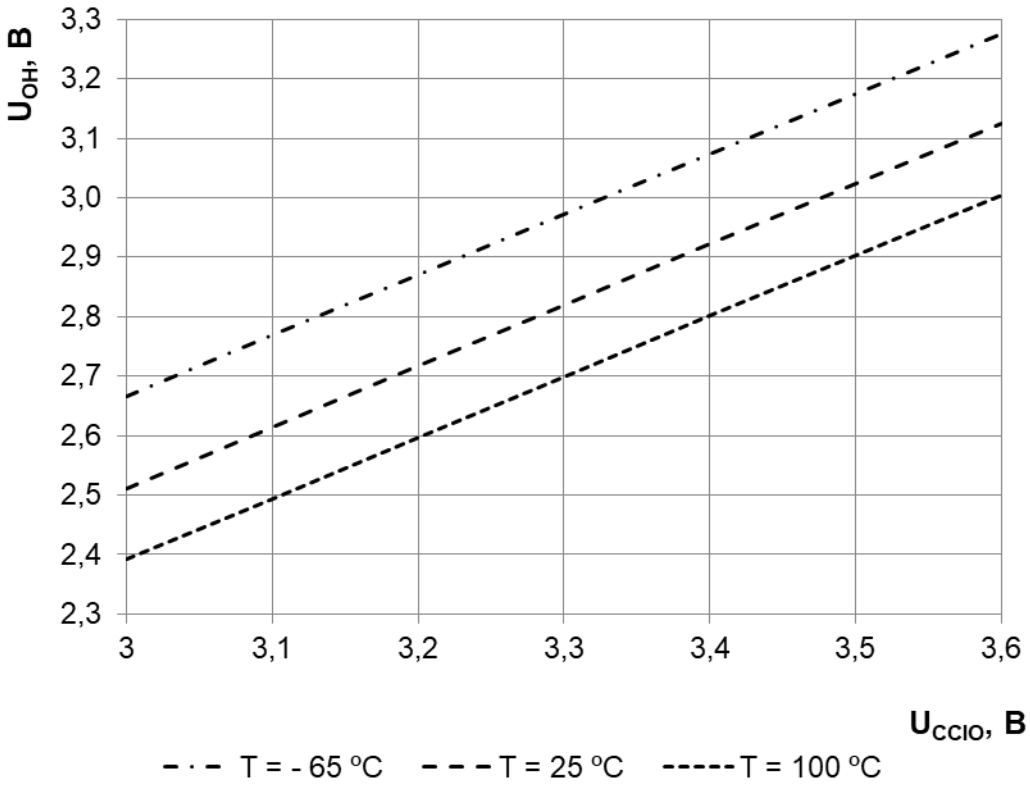


Рисунок 133 – Зависимость выходного напряжения высокого уровня U_{OH} от напряжения питания схем ввода/вывода U_{CCIO} при U_{CC} = 1,08 В, I_{OH} = – 4 мА

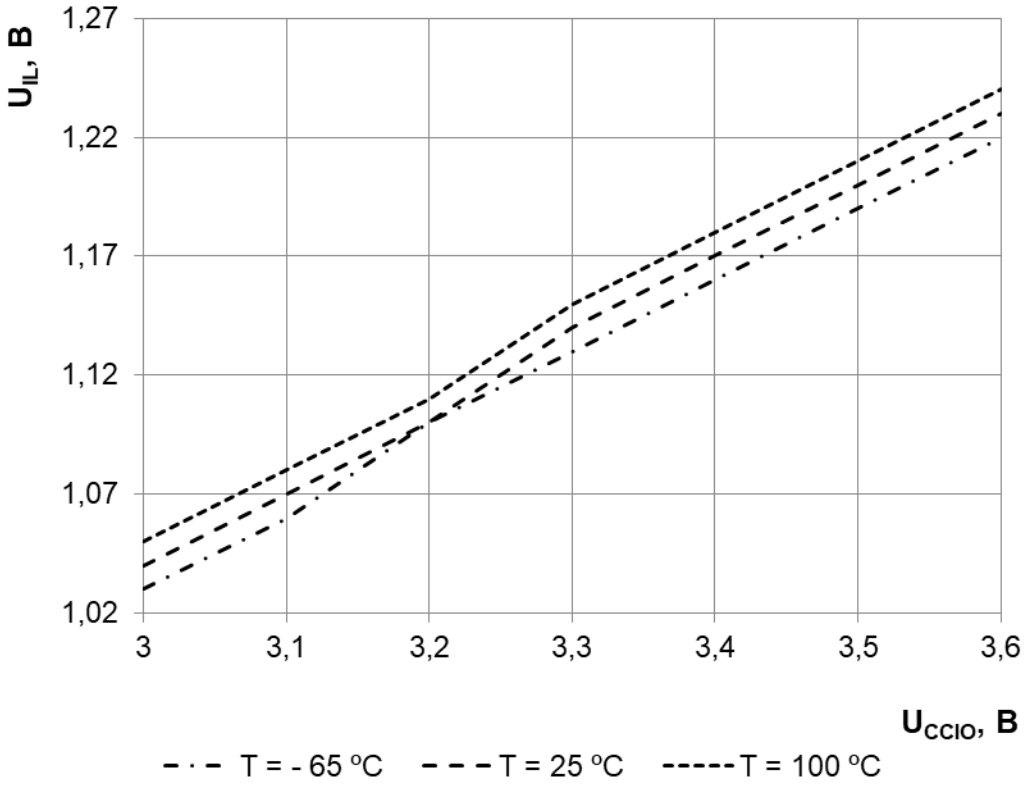


Рисунок 134 – Зависимость входного напряжения низкого уровня U_{IL} от напряжения питания схем ввода/вывода U_{CCIO} при U_{CC} = 1,08 В

36 Предельно-допустимые режимы

Таблица 265 – Предельно-допустимые режимы эксплуатации и предельные электрические режимы микросхем

Наименование параметра, единица измерения	Буквенное обозначение параметра	Предельно-допустимый режим		Предельный режим	
		не менее	не более	не менее	не более
Напряжение питания, В:					
– ядра на выводах VDD;	U_{CC}	1,08	1,32	–	1,4
– схем ввода/вывода на выводах VDD_IO;	U_{CCIO}	3,0	3,6	–	4,0
– аналоговых блоков на выводе VDDA	U_{CCA}	3,0	3,6	–	4,0
Входное напряжение высокого уровня, В	U_{IH}	2,0	U_{CCIO}	–	U_{CCIO} + 0,3
Входное напряжения низкого уровня, В	U_{IL}	0	0,4	– 0,3	0
Входное напряжение дифференциальное LINK портов, В, на парах выводов L0DIN[0:7]-L0DIP[0:7], L0CLKIN- L0CLKIP, L1DIN[0:7], L1DIP[0:7], L1CLKIN, L1CLKIP	U_{ID}	0,2	1,2	–	–
Входное напряжение синфазное LINK портов, В, на парах выводов L0DIN[0:7]-L0DIP[0:7], L0CLKIN-L0CLKIP, L1DIN[0:7]-L1DIP[0:7], L1CLKIN- L1CLKIP	U_{IS}	0,6	1,8	–	–
Напряжение, прикладываемое к выходу микросхемы в состоянии «Выключено», В	U_{OZ}	0	U_{CCIO}	– 0,3	U_{CCIO} + 0,3
Выходной ток высокого уровня, мА, на выводах портов A, B, C, D, E	I_{OH}	– 8*	0	– 24	–
Выходной ток низкого уровня, мА, на выводах портов A, B, C, D, E	I_{OL}	0	8	–	24
Тактовая частота процессора, МГц	f_C	–	230	–	–
Тактовая частота LINK портов, МГц	f_{LINK}	–	180	–	–
Входная частота PLL, МГц	f_{C_PLL}	10	100	–	–
Емкость нагрузки каждого выхода, пФ	C_L	–	15	–	–

* Суммарный выходной ток высокого уровня должен быть не более 800 мА.

Примечания:

1 Допускается использование разных источников для питания схем ввода/вывода и аналоговых блоков, при этом разница между напряжениями питания не должна превышать 200 мВ.

2 Не допускается одновременное воздействие двух и более предельных режимов

37 Электрические параметры

Таблица 266 – Электрические параметры микросхем при приемке и поставке

Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °С
		не менее	не более	
Выходное напряжение высокого уровня, В	U_{OH}	2,0	–	25, 100, – 60
Выходное напряжение низкого уровня, В	U_{OL}	–	0,4	
Дифференциальное выходное напряжение LINK портов уровней логического «0» и логической «1», мВ, R_L* = 100 Ом	U_{OD}	220	650	
Синфазное выходное напряжение LINK портов уровней логического «0» и логической «1», В, R_L = 100 Ом	U_{OS}	1,10	1,55	
Ток утечки высокого уровня на входе, мкА	I_{ILH}	– 20	20	
Ток утечки низкого уровня на входе, мкА	I_{ILL}	– 20	20	
Ток утечки высокого уровня на входе с внутренним резистором доопределения до питания, мкА, $U_{IH} = U_{CCIO}$	I_{ILH_U}	– 20	20	
Ток утечки низкого уровня на входе с внутренним резистором доопределения до нуля, мкА, U_{IL} = 0 В	I_{ILL_D}	– 20	20	
Входной ток высокого уровня на выводе с внутренним резистором доопределения до нуля, мкА	I_{IH_D}	20	150	
Входной ток низкого уровня на выводе с внутренним резистором доопределения до питания, мкА	I_{IL_U}	– 150	– 20	
Выходной ток высокого уровня в состоянии «Выключено», мкА, $U_{OZ} = U_{CCIO}$	I_{OZH}	– 20	20	
Выходной ток низкого уровня в состоянии «Выключено», мкА, U_{OZ} = 0 В	I_{OZL}	– 150	– 20	
Динамический ток потребления, мА:	I_{OCC}			
– ядра, при f_C = 230 МГц;		–	1000	
– аналоговых блоков;		–	20	
– схем ввода/вывода		–	800	

* R_L – резистор, подключаемый между выходами дифференциальной пары

Микросхемы устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В.

38 Габаритный чертеж микросхемы

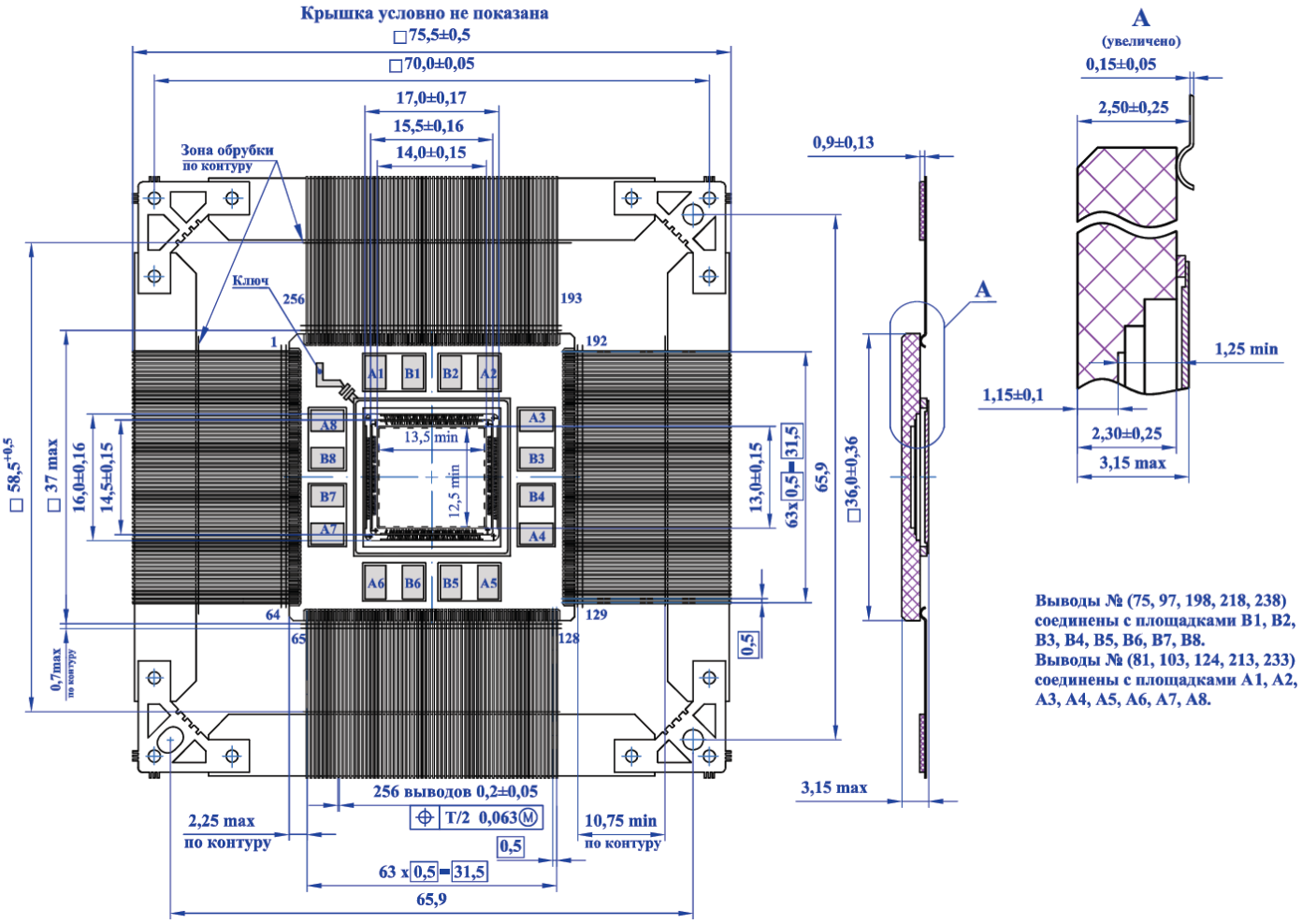


Рисунок 135 – Микросхема в корпусе 4244.256-3

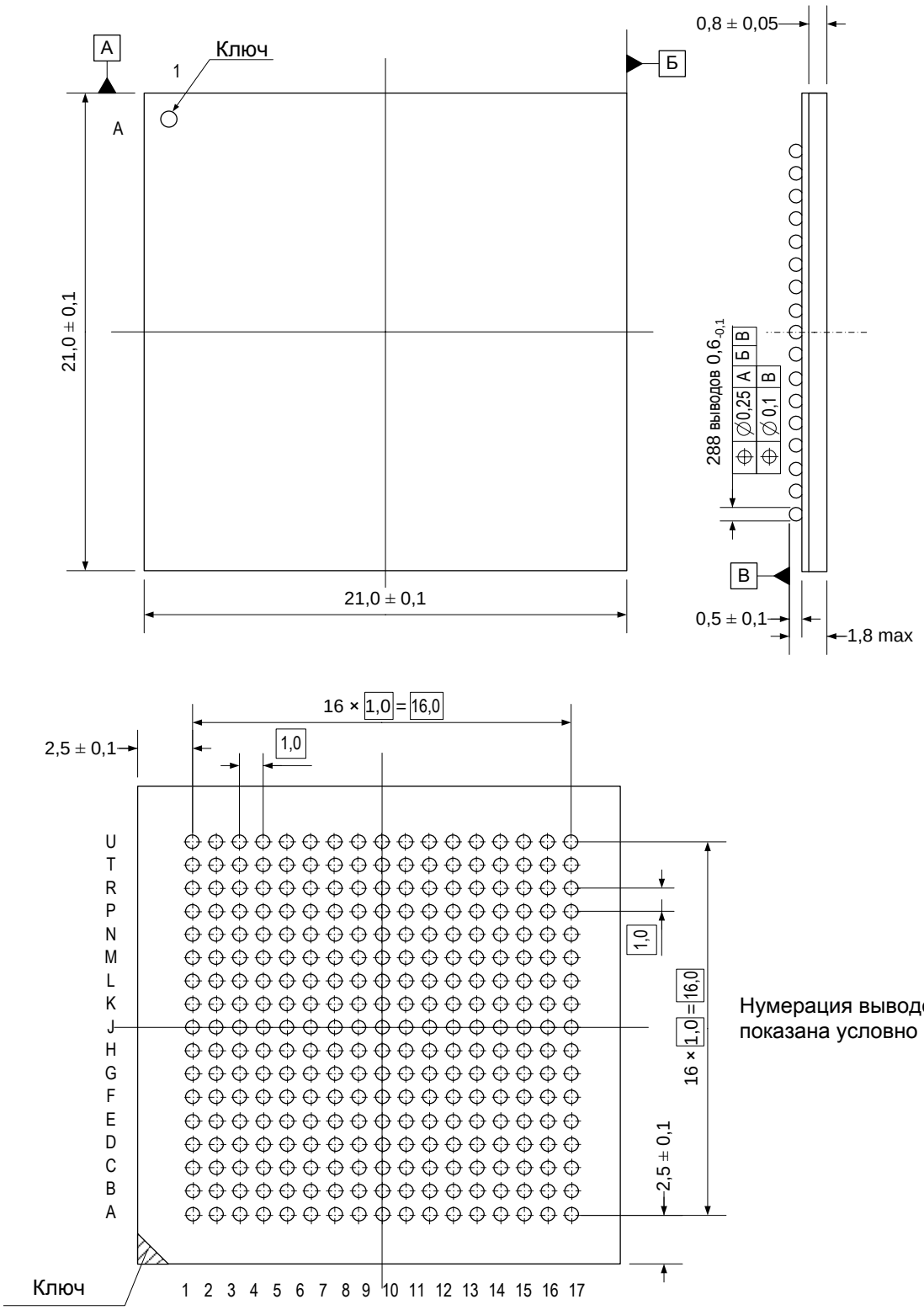
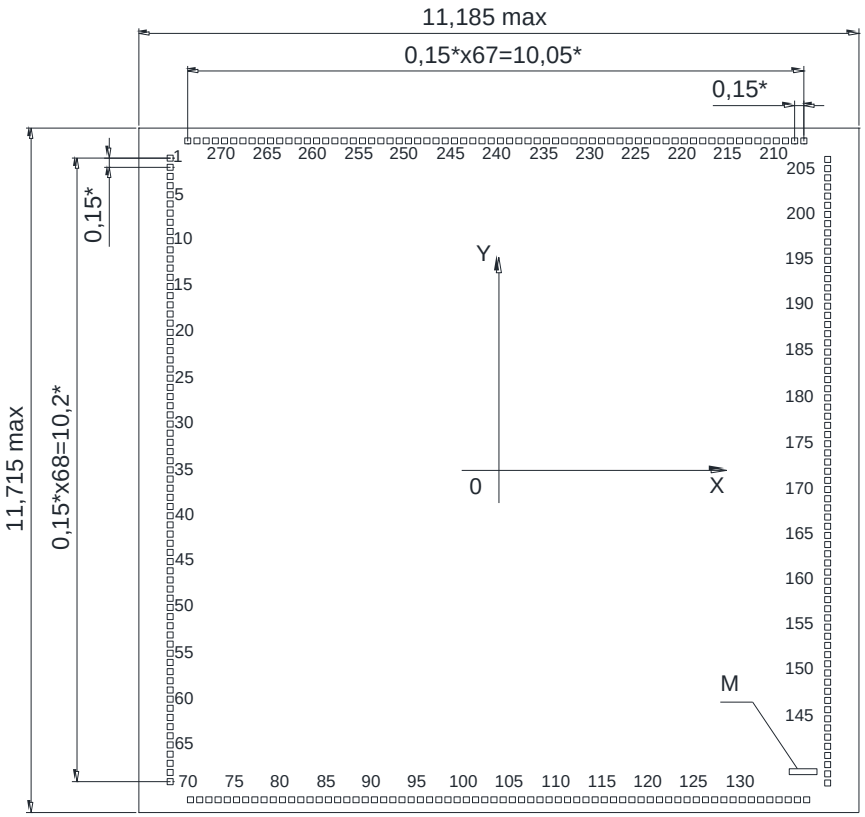


Рисунок 136 – Микросхема в корпусе BGA288



1. * Размеры для справок.
2. Размеры контактных площадок (КП) - 97 x 97 мкм.
3. Номера КП присвоены условно и их расположение соответствует топологическому чертежу.
4. Координаты КП смотри в таблице ниже.
5. Толщина кристалла (0,460±0,015) мм.
6. M - маркировка кристалла MLDR140.

Рисунок 137 – Кристалл (бескорпусное исполнение)

Таблица 267 – Координаты КП кристалла

№ КП	Обозначение КП	Координаты КП		№ КП	Обозначение КП	Координаты КП	
		X	Y			X	Y
1	TCK	-5341,640	5100,760	18	SPI_CLK	-5341,640	2550,520
2	TMS	-5341,640	4950,680	19	SPI_DO	-5341,640	2401,000
3	TDI	-5341,640	4800,600	20	SPI_DI	-5341,640	2250,920
4	TDO	-5341,640	4650,520	21	SPI_CS[0]	-5341,640	2100,840
5	nEMU	-5341,640	4501,000	22	SPI_CS[1]	-5341,640	1950,760
6	JG_MX	-5341,640	4350,920	23	SPI_CS[2]	-5341,640	1800,680
7	nRESET	-5341,640	4200,840	24	SPI_CS[3]	-5341,640	1650,600
8	GND	-5341,640	4050,840	25	SPI_CS[4]	-5341,640	1500,520
9	VDDA	-5341,640	3900,840	26	SPI_CS[5]	-5341,640	1351,000
10	GND	-5341,640	3750,840	27	SSI0_TCLK	-5341,640	1200,920
11	GND	-5341,640	3600,840	28	VDD_IO	-5341,640	1050,920
12	XTO	-5341,640	3451,000	29	VDD_IO	-5341,640	900,920
13	XTI	-5341,640	3300,920	30	VDD_IO	-5341,640	750,920
14	U0_TXD	-5341,640	3150,840	31	SSI0_TFS	-5341,640	600,600
15	U0_RXD	-5341,640	3000,760	32	SSI0_TXD	-5341,640	450,520
16	U1_TXD	-5341,640	2850,680	33	SSI0_RCLK	-5341,640	301,000
17	U1_RXD	-5341,640	2700,600	34	SSI0_RFS	-5341,640	150,920

№ КП	Обозначение КП	Координаты КП		№ КП	Обозначение КП	Координаты КП	
		X	Y			X	Y
35	SSI0_RXD	-5341,640	0,840	83	LC_VSYNC	-3075,240	-5367,960
36	SSI1_TCLK	-5341,640	-149,240	84	LC_HSYNC	-2925,160	-5367,960
37	VDD	-5341,640	-299,240	85	LC_CLK	-2775,080	-5367,960
38	VDD	-5341,640	-449,240	86	nDMAR3	-2625,000	-5367,960
39	SSI1_TFS	-5341,640	-599,480	87	VDD	-2475,000	-5367,960
40	SSI1_TXD	-5341,640	-749,000	88	FLAG[0]	-2325,400	-5367,960
41	SSI1_RCLK	-5341,640	-899,080	89	FLAG[1]	-2175,320	-5367,960
42	SSI1_RFS	-5341,640	-1049,160	90	FLAG[2]	-2025,240	-5367,960
43	SSI1_RXD	-5341,640	-1199,240	91	FLAG[3]	-1875,160	-5367,960
44	VC_CLK	-5341,640	-1349,320	92	BOOT[0]	-1725,080	-5367,960
45	GND	-5341,640	-1499,320	93	BOOT[1]	-1575,000	-5367,960
46	GND	-5341,640	-1649,320	94	BOOT[2]	-1424,920	-5367,960
47	VC_VSYNC	-5341,640	-1799,000	95	SCLK	-1275,400	-5367,960
48	VC_HSYNC	-5341,640	-1949,080	96	VDD_IO	-1125,400	-5367,960
49	VC_DATA[0]	-5341,640	-2099,160	97	VDD_IO	-975,400	-5367,960
50	VC_DATA[1]	-5341,640	-2249,240	98	VDD_IO	-825,400	-5367,960
51	VC_DATA[2]	-5341,640	-2399,320	99	SD_CKE	-675,080	-5367,960
52	VC_DATA[3]	-5341,640	-2549,400	100	MSSD[0]	-525,000	-5367,960
53	VC_DATA[4]	-5341,640	-2699,480	101	MSSD[1]	-374,920	-5367,960
54	VC_DATA[5]	-5341,640	-2849,000	102	MSSD[2]	-225,400	-5367,960
55	VC_DATA[6]	-5341,640	-2999,080	103	MSSD[3]	-75,320	-5367,960
56	VC_DATA[7]	-5341,640	-3149,160	104	SD_nCAS	74,760	-5367,960
57	LC_B[0]	-5341,640	-3299,240	105	GND	224,760	-5367,960
58	LC_B[1]	-5341,640	-3449,320	106	GND	374,760	-5367,960
59	LC_B[2]	-5341,640	-3599,400	107	SD_nRAS	525,000	-5367,960
60	LC_B[3]	-5341,640	-3749,480	108	SD_nWE	675,080	-5367,960
61	LC_B[4]	-5341,640	-3899,000	109	SD_DQM	824,600	-5367,960
62	LC_B[5]	-5341,640	-4049,080	110	SD_A10	974,680	-5367,960
63	LC_G[0]	-5341,640	-4199,160	111	nMS[1]	1124,760	-5367,960
64	LC_G[1]	-5341,640	-4349,240	112	VDD	1274,760	-5367,960
65	LC_G[2]	-5341,640	-4499,320	113	nMS[0]	1424,920	-5367,960
66	LC_G[3]	-5341,640	-4649,400	114	nBMS	1575,000	-5367,960
67	LC_G[4]	-5341,640	-4799,480	115	nRD	1725,080	-5367,960
68	LC_G[5]	-5341,640	-4949,000	116	nWR	1874,600	-5367,960
69	LC_R[0]	-5341,640	-5099,080	117	ACK	2024,680	-5367,960
70	LC_R[1]	-5025,160	-5367,960	118	ADDR[21]	2174,760	-5367,960
71	LC_R[2]	-4875,080	-5367,960	119	ADDR[20]	2324,840	-5367,960
72	LC_R[3]	-4725,000	-5367,960	120	ADDR[19]	2474,920	-5367,960
73	LC_R[4]	-4574,920	-5367,960	121	ADDR[18]	2625,000	-5367,960
74	LC_R[5]	-4425,400	-5367,960	122	ADDR[17]	2775,080	-5367,960
75	LC_T[0]	-4275,320	-5367,960	123	ADDR[16]	2924,600	-5367,960
76	LC_T[1]	-4125,240	-5367,960	124	ADDR[15]	3074,680	-5367,960
77	LC_T[2]	-3975,160	-5367,960	125	ADDR[14]	3224,760	-5367,960
78	LC_T[3]	-3825,080	-5367,960	126	ADDR[13]	3374,840	-5367,960
79	LC_PWM	-3675,000	-5367,960	127	ADDR[12]	3524,920	-5367,960
80	GND	-3525,000	-5367,960	128	ADDR[11]	3675,000	-5367,960
81	GND	-3375,000	-5367,960	129	ADDR[10]	3825,080	-5367,960
82	LC_DRDY	-3225,320	-5367,960	130	ADDR[9]	3974,600	-5367,960

№ КП	Обозначение КП	Координаты КП		№ КП	Обозначение КП	Координаты КП	
		X	Y			X	Y
131	ADDR[8]	4124,680	-5367,960	179	L0DON[0]	5341,640	1041,880
132	ADDR[7]	4274,760	-5367,960	180	L0DOP[0]	5341,640	1191,960
133	VDD	4424,760	-5367,960	181	L0DON[1]	5341,640	1342,040
134	ADDR[6]	4574,920	-5367,960	182	L0DOP[1]	5341,640	1492,120
135	ADDR[5]	4725,000	-5367,960	183	L0DON[2]	5341,640	1642,200
136	ADDR[4]	4875,080	-5367,960	184	L0DOP[2]	5341,640	1792,280
137	ADDR[3]	5024,600	-5367,960	185	VDD_IO	5341,640	1942,280
138	ADDR[2]	5341,640	-5108,040	186	VDD_IO	5341,640	2092,280
139	ADDR[1]	5341,640	-4957,960	187	VDD_IO	5341,640	2242,280
140	ADDR[0]	5341,640	-4807,880	188	L0DON[3]	5341,640	2392,040
141	DATA[31]	5341,640	-4657,800	189	L0DOP[3]	5341,640	2542,120
142	DATA[30]	5341,640	-4508,280	190	L0DON[4]	5341,640	2692,200
143	DATA[29]	5341,640	-4358,200	191	L0DOP[4]	5341,640	2842,280
144	DATA[28]	5341,640	-4208,120	192	VDD	5341,640	2991,800
145	DATA[27]	5341,640	-4058,040	193	L0DON[5]	5341,640	3141,880
146	GND	5341,640	-3908,040	194	L0DOP[5]	5341,640	3291,960
147	GND	5341,640	-3758,040	195	L0DON[6]	5341,640	3442,040
148	GND	5341,640	-3608,040	196	L0DOP[6]	5341,640	3592,120
149	DATA[26]	5341,640	-3458,280	197	L0DON[7]	5341,640	3742,200
150	DATA[25]	5341,640	-3308,200	198	L0DOP[7]	5341,640	3892,280
151	DATA[24]	5341,640	-3158,120	199	L0CLKON	5341,640	4041,800
152	DATA[23]	5341,640	-3008,040	200	L0CLKOP	5341,640	4191,880
153	DATA[22]	5341,640	-2857,960	201	L0DIN[0]	5341,640	4341,960
154	DATA[21]	5341,640	-2707,880	202	L0DIP[0]	5341,640	4492,040
155	DATA[20]	5341,640	-2557,800	203	L0DIN[1]	5341,640	4642,120
156	DATA[19]	5341,640	-2408,280	204	L0DIP[1]	5341,640	4792,200
157	DATA[18]	5341,640	-2258,200	205	L0DIN[2]	5341,640	4941,720
158	DATA[17]	5341,640	-2108,120	206	L0DIP[2]	5341,640	5091,800
159	VDD	5341,640	-1958,120	207	L0DIN[3]	4968,040	5367,960
160	DATA[16]	5341,640	-1807,960	208	L0DIP[3]	4817,960	5367,960
161	DATA[15]	5341,640	-1657,880	209	L0DIN[4]	4667,880	5367,960
162	DATA[14]	5341,640	-1507,800	210	L0DIP[4]	4517,800	5367,960
163	DATA[13]	5341,640	-1358,280	211	L0ACKO	4368,280	5367,960
164	DATA[12]	5341,640	-1208,200	212	GND	4218,280	5367,960
165	DATA[11]	5341,640	-1058,120	213	GND	4068,280	5367,960
166	DATA[10]	5341,640	-908,040	214	L0DIN[5]	3918,040	5367,960
167	DATA[9]	5341,640	-757,960	215	L0DIP[5]	3767,960	5367,960
168	DATA[8]	5341,640	-607,880	216	L0DIN[6]	3617,880	5367,960
169	DATA[7]	5341,640	-457,800	217	L0DIP[6]	3467,800	5367,960
170	DATA[6]	5341,640	-308,280	218	L0DIN[7]	3318,280	5367,960
171	DATA[5]	5341,640	-158,200	219	L0DIP[7]	3168,200	5367,960
172	DATA[4]	5341,640	-8,120	220	L0CLKIN	3018,120	5367,960
173	DATA[3]	5341,640	141,960	221	L0CLKIP	2868,040	5367,960
174	DATA[2]	5341,640	292,040	222	L0ACKI	2717,960	5367,960
175	DATA[1]	5341,640	442,120	223	L0BCMPO	2567,880	5367,960
176	DATA[0]	5341,640	592,200	224	L0BCMPI	2417,800	5367,960
177	GND	5341,640	742,200	225	L1ACKO	2268,280	5367,960
178	GND	5341,640	892,200	226	L1ACKI	2118,200	5367,960

№ КП	Обозначение КП	Координаты КП		№ КП	Обозначение КП	Координаты КП	
		X	Y			X	Y
227	L1BCMPO	1968,120	5367,960	251	VDD	-1631,400	5367,960
228	VDD	1818,120	5367,960	252	L1CLKON	-1781,640	5367,960
229	L1DON[0]	1668,520	5367,960	253	L1CLKOP	-1931,720	5367,960
230	L1DOP[0]	1518,440	5367,960	254	L1DIN[0]	-2081,800	5367,960
231	L1DON[1]	1368,360	5367,960	255	L1DIP[0]	-2231,880	5367,960
232	L1DOP[1]	1218,280	5367,960	256	GND	-2381,880	5367,960
233	GND	1068,280	5367,960	257	L1DIN[1]	-2532,040	5367,960
234	L1BCMPI	918,120	5367,960	258	L1DIP[1]	-2682,120	5367,960
235	L1DON[2]	768,600	5367,960	259	L1DIN[2]	-2832,200	5367,960
236	L1DOP[2]	618,520	5367,960	260	L1DIP[2]	-2982,280	5367,960
237	L1DON[3]	468,440	5367,960	261	L1DIN[3]	-3131,800	5367,960
238	L1DOP[3]	318,360	5367,960	262	L1DIP[3]	-3281,880	5367,960
239	VDD_IO	168,360	5367,960	263	L1DIN[4]	-3431,960	5367,960
240	VDD_IO	18,360	5367,960	264	L1DIP[4]	-3582,040	5367,960
241	VDD_IO	-131,640	5367,960	265	L1DIN[5]	-3732,120	5367,960
242	L1DON[4]	-281,400	5367,960	266	L1DIP[5]	-3882,200	5367,960
243	L1DOP[4]	-431,480	5367,960	267	L1DIN[6]	-4031,720	5367,960
244	L1DN[5]	-581,560	5367,960	268	L1DIP[6]	-4181,800	5367,960
245	L1DOP[5]	-731,640	5367,960	269	L1DIN[7]	-4331,880	5367,960
246	L1DON[6]	-881,160	5367,960	270	L1DIP[7]	-4481,960	5367,960
247	L1DOP[6]	-1031,240	5367,960	271	L1CLKIN	-4632,040	5367,960
248	L1DON[7]	-1181,320	5367,960	272	L1CLKIP	-4782,120	5367,960
249	L1DOP[7]	-1331,400	5367,960	273	nTRESET	-4932,200	5367,960
250	VDD	-1481,400	5367,960	274	GND	-5082,280	5367,960

39 Информация для заказа

Обозначение	Маркировка	Тип корпуса	Температурный диапазон
1967ВН044	1967ВН044	4244.256-3	минус 60 – 100 °С
К1967ВН044	К1967ВН044	4244.256-3	минус 60 – 100 °С
К1967ВН044К	К1967ВН044●	4244.256-3	0 – 70 °С
К1967ВН04BG	MDR32S2BG	BGA288	минус 40 – 85 °С

Примечание – Микросхемы в бескорпусном исполнении поставляются в виде отдельных кристаллов, получаемых разделением пластины. Микросхемы поставляются в таре (кейсах) без потери ориентации. Маркировка микросхемы в бескорпусном исполнении – 1967ВН04Н4, К1967ВН04Н4 – наносится на тару.

Микросхемы с приемкой «ВП» маркируются ромбом.
Микросхемы с приемкой «ОТК» маркируются буквой «К».

Лист регистрации изменений

№ п/п	Дата	Версия	Краткое содержание изменения	№№ изменяемых листов
1	02.06.2017	0.1.0	Введена вновь	–
2	09.06.2017	0.1.1	Исправлена маркировка типономинала на рисунке, приведены в соответствиие с ТУ и КД таблицы приемки-поставки, предельных режимов, таблица выводов, блок-схема, УГО	1, 11-27, 436, 437
3	07.08.2017	0.2.0	Введение бескорпусной микросхемы К1967ВН04Н4 Внесение правок и дополнений главного конструктора	По тексту
4	23.01.2018	0.2.1	Исправлена опечатка	440, 441
5	06.04.2018	1.0.0	Исправлено значение тактовой частоты процессора в таблицах 258, 259, на рисунках 122, 123	1, 431, 433, 434
6	13.06.2018	1.1.0	Внесение правок и дополнений главного конструктора	92 – 124, 392 По тексту
7	13.02.2019	2.0.0	В таблице 1 уточнено описание вывода 6; Исправлены ошибки в описании дополнительных назначений выводов 38, 39, 42; добавлена сноска для дополнительных назначений выводов. В таблице 90 уточнено описание бита TY. В таблице 97 исправлено описание бита 17; исправлены ошибки в описаниях протокола медленного устройства и конвейерного протокола для шины. В подразделе 15,2 исправление ошибки. В таблице 255 исправлено обозначение битов 19, 20; в подразделе 28.4 добавлено примечание	13, 15, 25 170 204, 205 245 418
8	17.06.2019	2.1.0	Исправлена опечатка на рисунке 1. Исправлено описание битов 63-58 в таблице 76. Исправлена опечатка в таблице 110. Исправлено рекомендуемое значение STEP. Добавлена схема подключения отладчика JEM-LINX. Добавлен раздел Схема формирования внутреннего сброса по включению питания. Таблицы электрических параметров и предельно-допустимых режимов приведены в соответствие с ТУ	11 149 232 399 431 437 442, 443
9	26.05.2020	2.2.0	Введение исполнения в пластиковом корпусе. Плановая корректировка. Исправлена ошибка в подключении интерфейса I2C (таблицы 1, 2, 194). Добавлен п. 23.4.6 с рекомендациями по подключению. Изменено описание бита I2C_ALT (таблица 255)	По тексту 20, 34, 35 365 427
10				